Image Copy-Move Forgery Detection in the Social Media Based on a Prior Density Clustering and the Point Density

Cong Lin[1,3(✉)], Hai Yang[2], Ke Huang[2], Yufeng Wu[1], Yamin Wen[1,3], and Yuqiao Deng[1]

[1] Applied Laboratory of Dig Data and Education Statistics, School of Statistics and Mathematics, Guangdong University of Finance and Economics, Guangzhou 510320, China
lincong0310@gmail.com

[2] School of Information, Guangdong University of Finance and Economics, Guangzhou 510320, China

[3] Guangdong Provincial Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou 51006, China

Abstract. Copy-move forgery is one common image manipulation technique. Many images are compressed by the social media, but most of the existing copy-move forgery detection schemes are proposed to deal with the uncompressed version of images. To handle this problem, in this paper, a copy-move forgery detection scheme for social media is proposed based on the point density and a prior density clustering. Firstly, the concept of "point density" is proposed, and it is combined with feature extraction to improve the extraction effect. Secondly, the hierarchical matching is adopted, and the keypoints are grouped according to the pixel value. Thirdly, a prior-based density clustering is proposed, called prior-DBSCAN. In this scheme, the matching pairs are divided into start points and end points for clustering, respectively, and the prior region of each cluster is obtained. Then, the clustering with the new cluster radius is performed. Finally, an iterative localization technique is used to obtain the final localization results. Considering the compression of images during their transmission through the social media, the proposed scheme is made more suitable for real-world scenarios. The experimental results demonstrate that the proposed scheme based on a prior density clustering and the point density, which is better and more robust than the state-of-the-art schemes on publicly available datasets.

Supported by Characteristic Innovation Project of Regular Institutions of Higher Learning of Guangdong Province (Natural Science) (2022KTSCX041); Basic and Applied Basic Research Project of Guangzhou Science and Technology Program (202102080316); Opening Project of Guangdong Province Key Laboratory of Information Security Technology (No. 2020B1212060078); Science and Technology Program of Guangzhou Haizhu District (海科工商信计2022-45).

J. Cai et al. (Eds.): SPNCE 2023, LNICST 525, pp. 239–257, 2025.
https://doi.org/10.1007/978-3-031-73699-5_17

Keywords: Multimedia forensics · Image forensics · Copy-move forgery · Density clustering · Point density

1 Introduction

With the increasing popularity of chat tools and image editing software, people can freely publish modified images on the Internet through simple operations and extremely low costs, which has caused serious social integrity problems. Therefore, it is necessary to study digital image forgery detection. The main forgery schemes are splicing, inpainting and copy-move. Image copy-move forgery refers to one or several parts of an image are copied and pasted into another region of the same image. At present, many excellent CMFD schemes have been proposed. Most of them can be divided into three categories: block-based schemes, keypoint-based schemes and deep learning-based schemes.

The main idea of block-based schemes is to divide the image into several overlapping rectangular blocks or circular blocks, and features are extracted from each block for matching. The main difference among the block-based schemes is the block features. Fridrich et al. [10] proposed to use Discrete Cosine Transform (DCT) as the block feature, the blur moment invariants were proposed by Mahdian et al. [19], Muhammad et al. [21] chose to use Discrete Wavelet Transform (DWT) as the features, Ryu et al. [26,27] proposed to use the Zernike moment feature, Li et al. [15] used Polar Cosine Transform (PCT) as the block feature, Qin et al. [25] introduced the features of radial harmonic Fourier moments, Lai et al. [13] introduced the features of Exponential-Fourier moments, and Emam et al. [9] proposed to use Polar Complex Exponential Transform (PCET) as the block features, Meena et al. [20] proposed the scheme of using Tetrolet transform. Wang et al. [34] used a scheme combining PCET with Singular Value Decomposition (SVD). The above block-based schemes have shown certain robustness under various attack schemes. However, the image is divided into many pixel blocks, it brings high computational complexity. In order to solve this problem, a new forgery detection scheme based on improved PatchMatch was proposed by Cozzolino et al. [8], and the efficiency of the algorithm was greatly improved. In summary, the block-based schemes robustness is poor for large-scale geometric transformations, such as rotation and scaling. Based on this, the keypoint-based schemes were proposed and became another research hotspot.

The basic steps of the keypoint-based schemes include feature extraction, matching, clustering and post-processing. For feature extraction, the Scale-Invariant Feature Transform (SIFT) algorithm is widely used as features in [2,14,24,31]. To obtain sufficient and uniform SIFT keypoints, Li et al. [14] introduced to reduce the contrast threshold and resize the input image, Gan et al. [11] designed an improved SIFT structure with inherent scale invariance and removed the contrast threshold, and Wang et al. [31] proposed strategies to normalize images and remove contrast thresholding. In addition, SURF [28,29], KAZE [36], and LIOP [5,16,18] are also adopted as CMFD features. For feature matching, Pan et al. [24] introduced a 2-Nearest Neighbor (2NN) matching

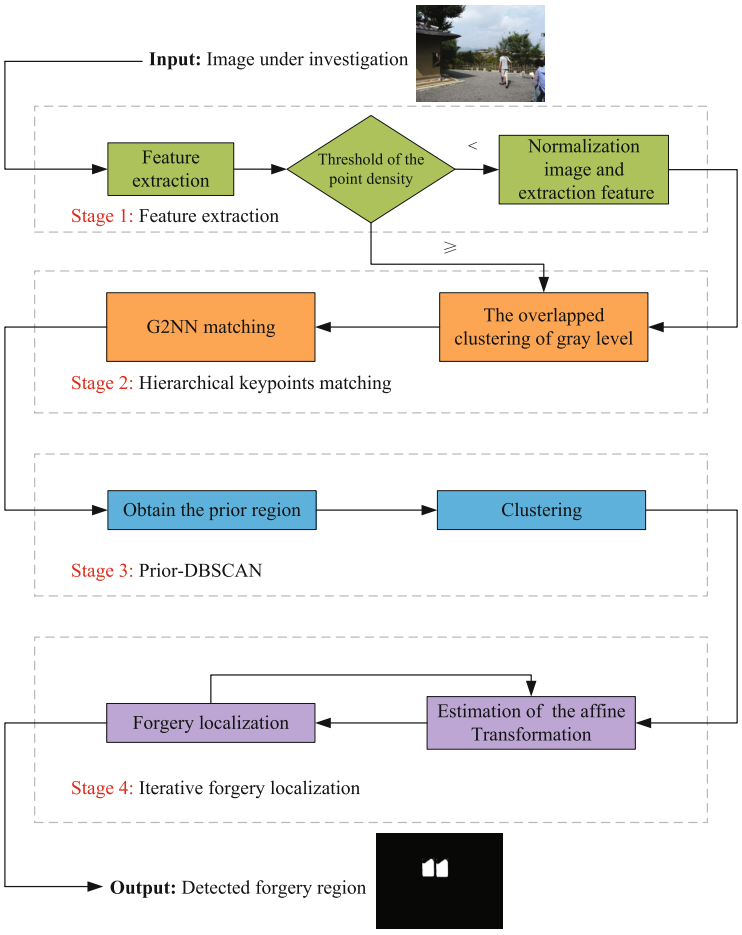

Fig. 1. Framework of the proposed copy-move forgery detection scheme.

method. However, this method cannot handle the case of multiple matching points. Therefore, Amerini et al. [2] proposed the generalized 2-nearest neighbor (G2NN) method. Subsequently, the reverse G2NN (RG2NN) method was proposed by Wang et al. [32]. The double matching approach was used in [18,31]. Li et al. [14] designed a hierarchical matching strategy, and magnitude hierarchical matching was adopted by Niu et al. [23] for feature matching. A Feature Label Matching (FLM) method was proposed by Gan et al. [11]. To detect possible duplicate regions, keypoint clustering is adopted by many keypoint-based schemes. The agglomerative hierarchical clustering was used in [2]. Afterwards, Amerini et al. [3] adopted the JLinkage algorithm [30] and Lyu et al. [18] proposed to use DBSCAN to classify matching points. Yang et al. [37] proposed a grid-based and cluster-based two-stage filtering scheme. For post-processing, to estimate the geometric transformation between the original region and the

forged region, the RANdom SAmple Consensus (RANSAC) algorithm is used in the work of [2,14,18,24,36,38]. Wang et al. [33] proposed to apply Progressive sample Consensus (PROSAC) in the field of CMFD. For localization, correlation coefficient of the pixel is calculated by [18,24,36]. Li et al. [14] performed forgery localization by matching adjacent segmented regions of keypoints. In summary, most keypoint-based schemes extract too few keypoints in smooth or small tampered regions, the detection results will be affected.

In recent years, with the development of artificial intelligence and the maturity of deep learning technology, many CMFD schemes based on deep learning have been proposed. The end-to-end Dense-InceptionNet and AR-Net were introduced by Zhong et al. [40] and Zhu et al. [41], respectively. Jindal et al. [1] used deep convolutional neural networks and semantic segmentation to detect copy-move and splicing tampered images. Liu et al. [17] proposed a two-stage CMFD scheme framework with backbone self deep matching network and Proposal SuperGlue. An end-to-end High-Resolution Dilation convolutional Attention Network (HRDA-Net) detection scheme was proposed by Zhu et al. [42] to handle the case of multiple tampering. Nazir et al. [22] used DenseNet-41 to extract features, and then forgery localization was performed through mask region-based convolutional neural network (RCNN). A deep convolutional neural network (VI-NET) hybrid with VGG and inception v3 was proposed by Kumar et al. [12]. Aria et al. [4] introduced an image quality-independent deep learning scheme (QDL-CMFD) for CMFD. A U-Net-like tampered region related framework (UCM-Net) was proposed by Weng et al. [35]. Barni et al. [6] used Multi-Branch Convolutional Neural Networks (Multi-Branch CNNs) and Zhang et al. [39] proposed a convolutional neural network-based generative adversarial network to detect the source and target regions of the tampered image. In summary, most of the deep learning-based schemes are able to handle different types of tampering. But those schemes require a lot of data, time and computing resources.

In the era of social media, most of the images that people can contact with are compressed by the social media. The original version of the image is often dealt with by the previous schemes. Therefore, the detection of images compressed by social media has more theoretical value and practical significance. In this paper, a novel copy-move forgery detection scheme for social media is proposed. The main contributions of this paper are as follows:

1) The concept of "point density" is proposed. Combining "point density" with feature extraction, the effect of feature extraction is improved;
2) The concept of "prior" is proposed. The approximate region of the tampering target is obtained by using "prior", resulting in an enhancement of keypoints clustering and the forgery localization effectiveness;
3) A novel density clustering algorithm is proposed, which is called prior-DBSCAN. The prior region with arbitrary shape is fitted by a circle of equal area, thus, the clustering effect is enhanced, the grouping of keypoints and the filtering of false matches are improved effectively.

The remainder of this paper is organized as follows. Section 2 describes the proposed scheme in detail; Sect. 3 shows the experimental results of copy-move forgery detection; Finally, a brief summary is made in Sect. 4.

(a) (b)

Fig. 2. Keypoints detection: (a) The SIFT under common setting.(b) The SIFT by combining the point density.

2 Proposed Scheme

This section describes the proposed copy-move forgery detection scheme in detail. Figure 1 shows the framework of the proposed scheme. First, the SIFT feature are extracted from the input image, and the better effect is obtained by combining the point density with feature extraction. Then, the extracted keypoints are grouped by the gray values, and the keypoints in different groups are matched. Subsequently, a prior-DBSCAN method is proposed. The prior region of each cluster is obtained by clustering the matching pairs, and then clustering combined with prior region is performed. The possible tampered regions are obtained and mismatched pairs are removed. Finally, the iterative forgery localization method is used to detect the forgery region from the input image.

2.1 Feature Extraction

Since the SIFT feature is invariant to scaling, rotation and translation, and has good robustness to illumination changes and affine transformations. Similar to many existing copy-move forgery detection schemes, the SIFT is adopted for detecting keypoints with corresponding interest regions.

The number of keypoints is closely related to the detection performance. The sufficiency of the number of keypoints should not only be determined by their quantity, but also combined with the image size for comprehensive evaluated. Therefore, in this paper, the concept of "point density" is introduced to measure

whether the number of extracted keypoints is sufficient or not. The definition of "point density" ρ is as follows:

$$\rho = \frac{n}{N_I \times M_I} \tag{1}$$

where n means the number of the SIFT keypoints; $N_I \times M_I$ is the resolution for the input image I. Image point density denotes the number of keypoints per unit area. By using the point density, the interest region in the image can be selected, the real region and the forged region can be distinguished. For the keypoint-based on CMFD scheme, not only the total number of keypoints is sufficient, but also enough keypoints can be contained in the critical area. However, with the development of social media, a large number of images will be compressed and transmitted through social network channels. After the image is compressed by social media, the resolution is reduced and the number of keypoints extracted is decreased. In order to balance the efficiency and performance of the proposed scheme, if Eq. (2) is satisfied, the image resolution will be increased:

$$(n \leqslant N_p) \cup (\rho \leqslant T_\rho) \tag{2}$$

where $N_p = 10000$ and $T_\rho = 0.005$ in our implementation. For images that do not satisfy Eq. (2), it means that the number of keypoints is sufficient.

Combining point density with feature extraction, n keypoints $\{k_1, k_2, \ldots, k_n\}$ are generated for the given image I, where $k = (x, y, \sigma, \theta)$, x, y are the coordinates of the keypoint, σ, θ are the scale and direction of the keypoint, respectively. The descriptor corresponding to each keypoint is $\{f_1, f_2, \ldots, f_n\}$, where f is a 128-dimensional vector. Two examples of the keypoint detection using SIFT features extracted under common settings and combined point density are shown in Fig. 2. As can be seen that, after combining the point density, the keypoints are more uniformly and densely distributed in the image, the number is increased, so that the copy-move regions can be covered by extracted keypoints.

2.2 Keypoint Hierarchical Matching

In the copy-move forgery detection scenario, the gray value between the corresponding positions of the tampered object will be the same or similar. Based on this, in this paper, the keypoints are grouped by their gray values [14], and matched over each group. After the keypoints are grouped, the number of keypoints in each group is reduced, which speeds up the matching process of keypoints.

Suppose that n keypoints are extracted, for each keypoint k_i, the Euclidean distance between it and the remaining $(n-1)$ keypoints are calculated. Denote vector $D = \{d_1, d_2, \ldots, d_{n-1}\}$ as the Euclidean distances in an increasing order. Then the keypoint k_i is matched if and only if:

$$d_1/d_2 < T_d \tag{3}$$

where the threshold T_d is fixed to 0.6. If Eq. (3) is satisfied, the keypoint corresponding to the minimum Euclidean distance d_1 is considered as the matching

point of k_i. This is the traditional 2NN matching method. Since copy-move forgery may happen that the same region is cloned repeatedly, a keypoint can be matched to multiple keypoints. Therefore, in this paper, the G2NN matching method proposed by Amerini et al. [2] is adopted. Multiple matching points of the keypoint k_i can be found only if:

$$d_j/d_{j+1} < T_d, j \in (2, \ldots, n-2) \tag{4}$$

is satisfied. Increase the value of j until the Eq. (4) is not satisfied, then the keypoints corresponding to $\{d_1, d_2, \ldots, d_{j-1}\}$ are the matching points of k_i.

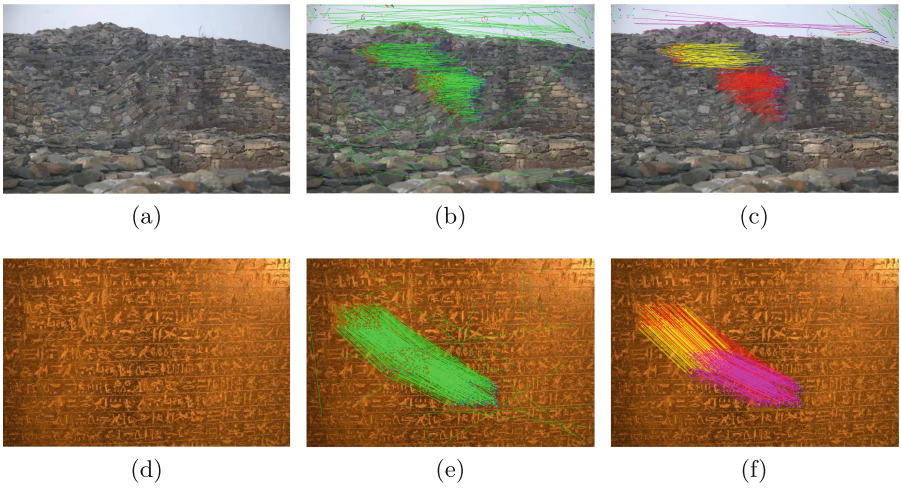

Fig. 3. The results after matching and clustering. Column 1: The input images. Column 2: The results after feature matching. Column 3: The results after clustering.

The iterative operation is performed on all keypoints, and a set of each keypoint and its corresponding matching points is obtained. The second column of Fig. 3 shows that the matching of the two images. The results show that there are some keypoints of mismatching outside the region with high density of matching pairs. If the matching pairs obtained in this step is directly used for forgery localization, the localization accuracy will be affected. Therefore, the following clustering step will be used to reduce false matching, and improve localization efficiency and localization accuracy.

2.3 Prior-DBSCAN

After keypoints matching, the density of matching keypoints in the tampered regions is often significantly greater than other mismatched regions. Therefore, to identify possible tampered regions, density clustering is performed after matching. Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

is a representative density clustering algorithm. It has the advantages of being insensitive to noisy data and can cluster dense datasets of any shape. However, the traditional DBSCAN has the disadvantages that it is difficult to distinguish the regions that are close to each other and sensitive to parameters.

To overcome the shortcomings of the DBSCAN algorithm. In this section, a new density clustering algorithm called Prior-DBSCAN is proposed. First, the prior region is obtained by each cluster after an adaptive radius clustering; Second, the prior region is fitted and the clustering is executed again.

Obtain the Prior Region. Considering that the matching keypoints should belong to different regions, the better clustering effect can be achieved by dividing the matching points into two parts and clustering them separately. Therefore, a unified direction for the matching pairs is obtained, and the matching pairs are divided into the start points and the end points to cluster separately. Suppose m pairs of matching keypoints $(K, K^{'})$ are detected, the keypoints $K = \{k_1, k_2, \ldots, k_m\}$, $K^{'} = \{k_1^{'}, k_2^{'}, \ldots, k_m^{'}\}$ and

$$\{k_i = (x, y, \sigma, \theta), k_i^{'} = (x^{'}, y^{'}, \sigma^{'}, \theta^{'}), k_i \in K, k_i^{'} \in K^{'}, i \in (1, \ldots, m)\} \quad (5)$$

where $x_i, y_i, x_i^{'}, y_i^{'}$ are the coordinate of the keypoints, $\sigma_i, \sigma_i^{'}, \theta_i, \theta_i^{'}$ are the scale and direction information of the keypoints, respectively. In this paper, the direction of matched pair $< k_i, k_i^{'} >$ is specified from k_i to $k_i^{'}$, the start point k_i and the end point $k_i^{'}$ of the matching keypoints are determined by the coordinate of the keypoint. In a matching pair, the keypoint with the smaller x coordinate value is considered as the start point, and the larger one is the end point. If the x coordinate values are the same, the y coordinate values are compared, and the smaller y value is regarded as the start point, and the larger value is regarded as the end point. If the initial direction does not satisfy the above direction conditions, the exchange is performed:

$$\{swap(k_i, k_i^{'}) | (x_i > x_i^{'}) \cup (x_i = x_i^{'}, y_i > y_i^{'}), k_i \in K, k_i^{'} \in K^{'}\} \quad (6)$$

where the $swap(k_i, k_i^{'})$ means to exchange keypoint k_i and $k_i^{'}$. Finally, the direction of each matching pair is adjusted.

The results of density clustering are affected by the clustering radius. If a fixed radius is used for all images, it is not satisfactory for clustering effects of some image. Therefore, an adaptive clustering radius is selected for each image, which reduces the objective preferences, and improves the accuracy and robustness of the DBSCAN results.

The adaptive clustering radius is determined by the mean of the Euclidean distance between the matching points:

$$R = \alpha \cdot aver(\sum_{i=1}^{m} \sqrt{(x_i - x_i^{'})^2 + (y_i - y_i^{'})^2}) \quad (7)$$

where m is the number of the matching keypoints, $x_i, x_i^{'}, y_i, y_i^{'}$ are the coordinates of the two matching keypoints, $aver(\cdot)$ refers to mean distance between matching

points, $\alpha = 1/10$ in our implementation. After the adaptive clustering radius is obtained, the clustering is performed on the start point set Z and the end point set $Z^{'}$, and the clusters with more than 4 points are selected:

$$\begin{aligned} C &= DBSCAN(Z, R) \\ and \quad C^{'} &= DBSCAN(Z^{'}, R) \end{aligned} \tag{8}$$

where C is clustering result of the start point set Z, denoted as $C = (c_1, \ldots, c_p)$, $C^{'}$ is clustering result of the end point set $Z^{'}$, denoted as $C' = (c_1^{'}, \ldots, c_q^{'})$. After clustering, the number of clusters at both ends may be different. It is necessary to find the matching cluster according to the number of matching keypoints in the two clusters. If the number of matching keypoints between the two clusters is more than 4 pairs, then the two clusters are considered to be matched clusters. In order to intuitively identify the matching clusters, in the following sections, the matching results of each cluster are uniformly represented by $C = (c_1, \ldots, c_s)$, $C^{'} = (c_1^{'}, \ldots, c_s^{'})$, and there are s matching clusters in total.

Figure 3 shows the effect after clustering. The images in the second column contains many irregular matching pairs. The red points represent the start points, and the blue points represent the end points. The direction of the matching pairs is red to blue. After clustering, the effect of filtering the wrong matching pairs can be achieved. The results are shown in the third column. The connecting lines of different colors represent different matching clusters after clustering.

After the adaptively selected clustering radius is used for clustering, and s pairs of matching clusters are obtained. The prior region of each cluster can be obtained according to the matching keypoints in each cluster. Suppose $k_0 = (x_0, y_0, \sigma_0, \theta_0)$ is a keypoint in cluster c_1, then the region s_0 composed of the keypoint:

$$s_0 = \{(x, y) | \sqrt{(x - x_0)^2 + (y - y_0)^2} \leqslant \beta\sigma_0, (x, y) \in S\} \tag{9}$$

where s_0 is a circular region centered at (x_0, y_0), S is the set of all pixels of the whole image, (x, y) is the coordinates of the pixels in S, (x_0, y_0) are the coordinates of the k_0; σ_0, θ_0 are the scale and direction of k_0 respectively, $\beta = 14$ in our implementation. The prior region of cluster c_1 is obtained by superimposing the circular region corresponding to each keypoint in c_1. According to the Eq. (9), each cluster in C, $C^{'}$ is selected for the same calculation. The prior region corresponding to each cluster are obtained.

Clustering. After the previous step, s pairs of matching clusters and their corresponding prior regions have been obtained. Considering that there may be multiple pairs tampered regions in a forgery image, and the distribution of keypoints of tampered regions in an image may also be different. To optimize the matching pairs, in this section, a circle is used to fit the prior region to obtain the new cluster radius, and the clustering is performed again according to the new radius to obtain a more accurate tampering region.

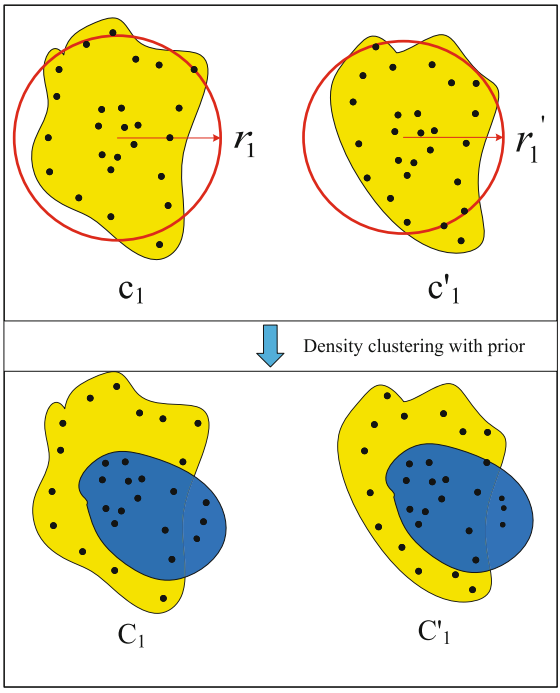

Fig. 4. The Prior-DBSCAN procedure. The yellow region indicates the prior region after clustering. The red circle is the fitting circle corresponding to the yellow region. The blue region represents the region after the prior-DBSCAN. (Color figure online)

Firstly, for each cluster, a circle is adopted to fitting with the prior region, denote Ac as the area of the circle, Ap as the area of the prior region, where $Ac = Ap$. The circle radius r can be represented as:

$$r = \sqrt{Ac/\pi} \tag{10}$$

Then, the new cluster radius is:

$$R^{'} = \gamma r \tag{11}$$

where $\gamma = 1/3$ in our implementation.

As shown in Fig. 4, the clusters $c_1, c_1^{'}$ are a pair of matching clusters, and the keypoints in them are one-to-one matching. First, the keypoints in clusters $c_1, c_1^{'}$ are used to obtain the respective prior regions (yellow regions), and according to Eq. (10), the circles (red circles) can be obtained respectively. Then, in the cluster $c_1, c_1^{'}$, a central keypoint (the center point of all keypoints contained in the cluster $c_1, c_1^{'}$) is selected as the start point. According to the circle radius r_1 and $r_1^{'}$, the cluster radius R_1 and $R_1^{'}$ can be obtained by Eq. (11). The density clustering is performed, and the new clusters $C_1, C_1^{'}$ can be obtained (points within the blue region).

2.4 Iterative Forgery Localization

In this section, an iterative localization method is used to locate the forgery image.

Step 1): Estimation of affine matrix;

Step 2): Correlation coefficients are used for forgery localization.

Table 1. Precision, Recall, and F_1 Scores (%) for Translation on FAU Dataset.

Methods	*Precision*	*Recall*	F_1
Cozzolino [8]	**58.22**	51.73	52.91
Li [14]	48.59	32.34	37.22
Lyu [18]	13.78	2.75	4.07
Proposed	55.72	**63.44**	**55.56**

Table 2. Precision, Recall, and F_1 Scores (%) for Translation on MICC-F600 Dataset.

Methods	*Precision*	*Recall*	F_1
Cozzolino [8]	**60.25**	53.34	54.84
Li [14]	53.52	35.91	41.19
Lyu [18]	16.86	3.64	5.29
Proposed	57.59	**64.47**	**56.63**

Estimation of Affine Transform. The affine matrix H is defined as follows:

$$\begin{pmatrix} x^{'} \\ y^{'} \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, H = \begin{pmatrix} t_{11} & t_{12} & t_x \\ t_{21} & t_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix} \tag{12}$$

where (x, y), $(x^{'}, y^{'})$ is the coordinates of keypoints and points after affine transformation, $t_{11}, t_{12}, t_{21}, t_{22}$ is the rotation and scaling parameter, t_x, t_y is the translation parameter.

To obtain a unique solution to the transform parameters, at least 3 matched pairs are required to calculate the affine matrix H. However, the obtained results are inaccurate due to the mismatched keypoints. To obtain a more accurate affine matrix, we employ the RANSAC algorithm to select inliers and remove outliers:

1) Randomly select more than 3 matching pairs that are not collinear from a pair of matching clusters, and an affine matrix is calculated;

2) Select a matching pair $(k, k^{'})$ in the matching cluster successively. If the following conditions are satisfied:

$$\|Hk - k^{'}\|_2^2 \leqslant T_1 \tag{13}$$

where $T_1 = 9$, the matching pair is regarded as the inlier corresponding to the affine matrix, otherwise it is regarded as the outlier. Then the number of inliers corresponding to the affine matrix is recorded;

3) Steps 1) and 2) are repeated Q times ($Q = 50$ in our implementation), and the affine matrix corresponding to the largest number of inliers is selected as the optimal affine matrix;

4) If there are multiple affine matrices obtained in step 3). Then, according to the Eq. (14), an affine matrix is selected as the optimal affine matrix:

$$H = \arg\min_{H} \sum_{i=1}^{n} \|k_i^{'} - Hk_i\|_2^2 \tag{14}$$

where n is the number of inliers, $(k_i, k_i^{'})$ is the inlier of the affine matrix.

Forgery Localization. It is assumed that there is a pair of duplicated regions (S_1, S_2), and the affine transformation matrix between the two regions is obtained, which satisfies the following:

$$S_1 \xrightarrow{H} S_2 \tag{15}$$

where H is the affine matrix from region S_1 to S_2, Select a pixel e in S_1 successively, and it is matched to e^* through the affine transformation H:

$$e^* = He, e \in S_1 \tag{16}$$

Let Ω_1 as the 5×5 pixels neighbor field centered at e, Ω_2 as the 5×5 pixels neighbor field centered at e^*. The correlation coefficient between two pixels is computed as:

$$C_e = \frac{\sum_{p \in \Omega_1, q \in \Omega_2} I(p)I(q)}{[\sum_{p \in \Omega_1} I^2(p)][\sum_{q \in \Omega_2} I^2(q)]} \tag{17}$$

where $I(\cdot)$ refers to the gray value, p is the pixel of Ω_1, q is the pixel of Ω_2.

After calculating the correlation coefficient of each pixel in a pair of duplicated regions, then the forgery localization is performed. Firstly, all the pixel values of the image are set to 0. If $C_e \geqslant T_2$ ($T_2 = 0.5$ in our implementation), the pixel value corresponding to the pixel e is set to 1. In this way, until all the pixels in the region are processed. The binary image can be obtained.

The same operation is performed on all matching clusters, and the binary images obtained from each matching cluster are merged. The preliminary localization result is obtained. Then the obtained localization result is post-processed, such as removing small regions, filling with the internal region and other morphological operations. Finally, the localization result of the input image is obtained.

In order to obtain the more accurate results, our scheme choose to perform N iterations of the two steps ($N = 10$ in our implementation). Then N binary images are merged to get the final results.

3 Experimental Results

In this section, the proposed scheme will be evaluated through experiments. The operating system of the PC running the algorithm is Microsoft Windows10, the PC used for testing has Intel i5-4200H CPU and 8 GB RAM, and the software is MATLAB 2021a.

Table 3. Precision, Recall, and F_1 Scores (%) for Scale on MICC-F600 Dataset.

Methods	*Precision*	*Recall*	F_1
Cozzolino [8]	55.60	43.37	46.00
Li [14]	31.69	16.70	20.44
Lyu [18]	4.91	0.80	1.38
Proposed	**62.50**	**55.63**	**53.45**

Table 4. Precision, Recall, and F_1 Scores (%) for Multiple Copy-Move on MICC-F600 Dataset.

Methods	*Precision*	*Recall*	F_1
Cozzolino [8]	51.52	37.66	42.33
Li [14]	38.18	18.27	22.66
Lyu [18]	19.27	4.39	6.28
Proposed	**52.23**	**61.24**	**53.62**

3.1 Datasets and Evaluation Criteria

Two public copy-move forgery test datasets, i.e., FAU [7] and MICC-F600 [3] are used to demonstrate the effectiveness of the proposed scheme.

The FAU is built by Christlein et al. [7]. This dataset consists of 48 original images and 1440 tampered images which include a variety of tampering types:

Table 5. Precision, Recall, and F_1 Scores(%) for Rotation on MICC-F600 Dataset.

Methods	*Precision*	*Recall*	F_1
Cozzolino [8]	56.80	50.24	51.96
Li [14]	57.73	38.26	43.74
Lyu [18]	12.22	1.52	2.42
Proposed	**63.48**	**59.87**	**55.80**

Table 6. Precision, Recall, and F_1 Scores(%) for Comprehensive Results on MICC-F600 Dataset.

Methods	*Precision*	*Recall*	F_1
Cozzolino [8]	56.05	46.15	48.78
Li [14]	45.28	27.29	32.01
Lyu [18]	13.32	2.59	3.84
Proposed	**58.95**	**60.30**	**54.88**

240 rotated images with a rotation angle range of $[0 : 0.02 : 0.1]$; 240 images are added with 5 different level of noise, and the range of added noise is $[0° : 2° : 10°]$; 432 JPEG compressed images with a quality factor range of $[20 : 10 : 100]$; 480 images are scaled by different scales, the range is $[0.91 : 0.02 : 1.09]$; There are also 48 images with plain copy-move forgery, this form of tampering means that only a simple copy-move operation is used on the tampered region. All images are of high quality, with an average resolution of 3000×2300.

The MICC-F600 was proposed by Amerini et al. [3]. This dataset consists of 600 images in total, of which 440 are original images and 160 are tampered images. The image resolution of the dataset ranges from 800×533 to 3888×2592. The tampered image is obtained from 40 basic images through the following 4 operations:

- **Translation:** The target region in the image is copied once and moved to another location in the image;
- **Rotation:** Rotate the duplicated region of the image by 30°;
- **Multiple:** Copy the target region in the image two or three times and move them to other location of the image;
- **Scale:** The duplicated region of the image is rotated by 30° and scale by 120%.

Nowadays, social media has become an indispensable tool to obtain information in human life. In order to save network resource, images transmitted on social media are often compressed, so most of the images that people contact at ordinary times are not the original version of the images. In this case, it is necessary to propose a CMFD scheme to detect images compressed by social media. Therefore, in our experiment, 48 plain copy tampered images of FAU and 160 tampered images of MICC-F600 are compressed by WeChat, and then the performance of the proposed scheme is tested. After compression, the average resolution of the dataset is about 800×500.

To evaluate the performance of the proposed scheme, *recall*, *precision* and F_1 scores are calculated at the pixel level to evaluate performance of the proposed scheme. They are calculated as:

$$precision = \frac{TP}{TP + FP} \tag{18}$$

$$recall = \frac{TP}{TP + FN} \tag{19}$$

$$F_1 = \frac{2 * precision * recall}{precision + recall} \tag{20}$$

where TP is the True Positive, which represents the number of correctly detected tampered pixels. FP is the False Positive, which are the number of falsely detected true pixels. FN is the False Negative, which represents the number of undetected tampered pixels. F_1 score is the comprehensive evaluation of the *precision* and the *recall*.

3.2 Comparison Results and Analysis

In order to verify the effectiveness of the proposed scheme, it is compared with some state-of-the-art CMFD schemes. The schemes are proposed by Cozzolino et al. [8], Li et al. [14] and Lyu et al. [18].

Experimental Results Under Plain Copy-Move. Tables 1 and 2 show the results of our scheme and several other schemes on two datasets. From the table, it can be seen that on FAU and MICC-F600 datasets, the *recall* and F_1 scores of our scheme are the best, and the F_1 score are 2.65% and 1.79% higher than those of the second best schemes on FAU and MICC-F600 datasets respectively. This indicates that the proposed scheme is better than those of the other schemes. The scheme is proposed by Cozzolino [8] has the highest *precision* and second in F_1 score and *recall*.

One of the main reasons why the F_1 score of the proposed scheme is better than other schemes may be that our scheme combines point density in the feature extraction stage, the sufficient keypoints are extracted. Another main reason is that this scheme optimizes the matching keypoints by using prior-DBSCAN combined with prior regions, which reduces the keypoints of mismatching. Finally, the localization accuracy is improved.

Experimental Results Under Different Transforms. In this subsection, the performance of the proposed scheme is tested with images under different transforms, which are more challenging than the plain copy tampered images. Overall, the *precision*, *recall* value and F_1 value of our scheme are higher than those of the other three schemes. It shows that the detection performance of the proposed scheme is better than that of the other three schemes.

Table 3 shows that the proposed scheme obtains the best detection results for scaled tampered images on MICC-F600 dataset. As you can see from the table, the *precision*, *recall* and F_1 score of the proposed scheme are 6.90%, 12.26% and 7.45% higher than those of the second best scheme. It indicates that our scheme is robust to scaling attacks. Table 4 shows the detection results for images where the target region is copied multiple times on the MICC-F600 dataset. From the data in the Table 4, the proposed scheme obtained the best

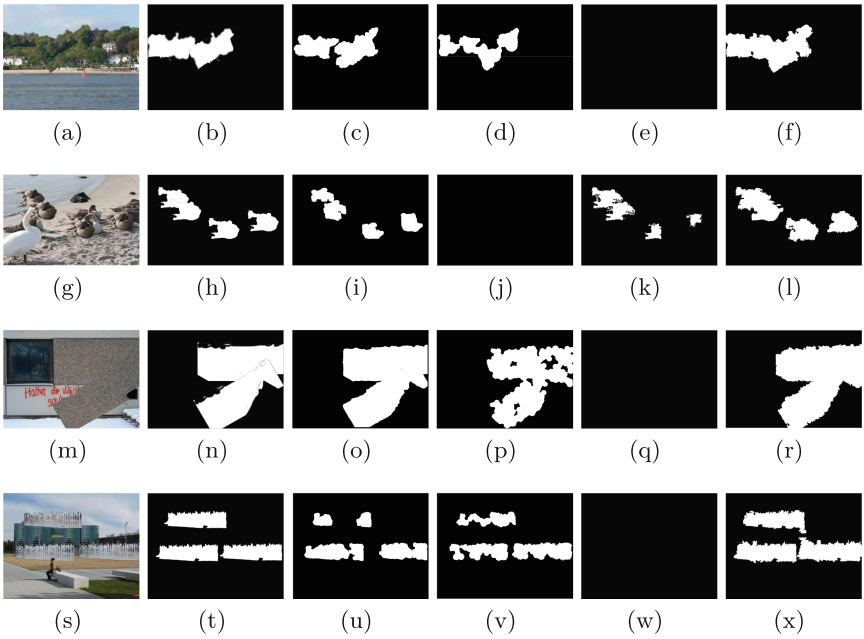

Fig. 5. Some challenging examples of copy-move forgery detection. From the first column to the last: the forged images, the ground-truth masks, the corresponding detection results by Cozzolino [8], by Li [14], by Lyu [18] and by our proposed scheme.

result. The *precision*, *recall* and F_1 score of the proposed scheme are 0.71%, 23.58%, 11.29% higher than those of the second best scheme. Table 5 shows the detection results of rotated tampered images on MICC-F600 dataset. It can be seen from the experimental results in the table that the *precision*, *recall* and F_1 score obtained by our scheme are the highest among several schemes, reaching to 63.48%, 59.87% and 55.80% respectively, which are 5.75%, 9.63%, 3.84% higher than those of the second best scheme. The results show that the proposed scheme is robust to rotation attacks. Table 6 shows the overall detection results of each scheme on MICC-F600 dataset. The proposed scheme obtains the best results. The *precision*, *recall* and F_1 score of the proposed scheme are 2.90%, 14.15%, 6.10% higher than those of the second best scheme. It shows that the proposed scheme has significant advantages in detecting images processed by the social media. The experimental results of the Lyu's scheme are relatively bad. The reasons are as follows. First, the Difference of Gaussians (DoG) keypoints and LIOP descriptors are extracted from the input image, then the Delaunay triangles are obtained and matched. Second, expanded triangle set are used to match again. Third, after the images are processed by social media, the results of feature extraction and feature matching of this scheme are not ideal. Then the forgery detection results of Lyu's are bad.

Performance of Challenging Copy-Move Forgery Detection. In this Section, some challenging copy-move forgery detection examples are shown in Fig. 5. In this figure, Fig. 5(a) are scaled tampered image, Fig. 5(g) and Fig. 5(s) are copied multiple times images, and Fig. 5(m) is a rotated tampered image.

As can be expected, the tampered region detected by Cozzolino in the third image is roughly the same as its ground-truth mask, and the tampered region of the other images is great different from the ground-truth mask. In the second image, no tampered regions were detected by Li. Lyu did not detect tampered regions in the first, third and fourth images, and in the second images, the detected tampered regions were relatively small. In contrast, the proposed scheme in this paper detects tampered regions in all four images, and the difference of tampered regions are relatively small compared with the ground-truth masks. As we can see, the robustness of the proposed scheme is the best, and good detection results are still achieved under those challenging conditions.

4 Conclusion

The existing schemes cannot effectively solve the problem that copy-move forgery images are processed by the social media. In order to solve this problem, a CMFD scheme based on the point density and the prior-DBSCAN is proposed. First of all, the concept of the point density is proposed. Considering that images transmitted through the social media are generally compressed. This paper measures the point density of the input image, and normalizes the image to get enough keypoints. Then, the keypoints are grouped by the pixel value to optimize the matching speed. Secondly, a novel prior-DBSCAN is proposed in this paper, a unified direction is appointed to matching pairs, and they are divided into starting points and ending points for DBSCAN. Then the corresponding prior region is obtained for each cluster after clustering. The prior region is fitted by a circle, the area of the circle is equal to that of the prior region, and the new clustering radius is obtained. The clustering is performed again to optimize the clustering effect. Finally, tampered regions are iteratively located by computing multiple affine matrices of the matched clusters. Experiments on public datasets show that the proposed scheme performs better than state-of-the-art schemes, under various challenging conditions, such as translation, scale, rotation, etc.

References

1. Abhishek, Jindal, N.: Copy move and splicing forgery detection using deep convolution neural network, and semantic segmentation. Multimedia Tools Appl. **80**(3), 3571–3599 (2021)
2. Amerini, I., Ballan, L., Caldelli, R., Bimbo, A.D., Serra, G.: A SIFT-based forensic method for copy-move attack detection and transformation recovery. IEEE Trans. Inf. Forensics Secur. **6**(3), 1099–1110 (2011)
3. Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., Del Tongo, L., Serra, G.: Copy-move forgery detection and localization by means of robust clustering with J-Linkage. Signal Process. Image Commun. **28**(6), 659–669 (2013)

4. Aria, M., Hashemzadeh, M., Farajzadeh, N.: QDL-CMFD: a quality-independent and deep learning-based copy-move image forgery detection method. Neurocomputing **511**, 213–236 (2022)
5. Aydın, Y.: A new copy-move forgery detection method using LIOP. J. Vis. Commun. Image Represent. **89**, 103661 (2022)
6. Barni, M., Phan, Q.T., Tondi, B.: Copy move source-target disambiguation through Multi-Branch CNNs. IEEE Trans. Inf. Forensics Secur. **16**, 1825–1840 (2020)
7. Christlein, V., Riess, C., Jordan, J., Riess, C., Angelopoulou, E.: An evaluation of popular copy-move forgery detection approaches. IEEE Trans. Inf. Forensics Secur. **7**(6), 1841–1854 (2012)
8. Cozzolino, D., Poggi, G., Verdoliva, L.: Efficient dense-field copy-move forgery detection. IEEE Trans. Inf. Forensics Secur. **10**(11), 2284–2297 (2015)
9. Emam, M., Han, Q., Niu, X.: PCET based copy-move forgery detection in images under geometric transforms. Multimedia Tools Appl. **75**(18), 11513–11527 (2016)
10. Fridrich, J., Soukal, D., Lukáš, J.: Detection of copy-move forgery in digital images. In: Proceeding of Digital Forensic Research Workshop (DFRW), pp. 19–23. Cleveland, OH, USA (2003)
11. Gan, Y., Zhong, J., Vong, C.: A novel copy-move forgery detection algorithm via feature label matching and hierarchical segmentation filtering. Inf. Process. Manag. **59**(1), 102783 (2022)
12. Kumar, S., Gupta, S.K., Kaur, M., Gupta, U.: VI-NET: a hybrid deep convolutional neural network using VGG and inception V3 model for copy-move forgery classification. J. Vis. Commun. Image Represent. **89**, 103644 (2022)
13. Lai, Y., Huang, T., Jiang, R.: Image region copy-move of forgery detection based on exponential-fourier moments. J. Image Graph. **20**(9), 1212–1221 (2015)
14. Li, Y., Zhou, J.: Fast and effective image copy-move forgery detection via hierarchical feature point matching. IEEE Trans. Inf. Forensics Secur. **14**(5), 1307–1322 (2019)
15. Li, Y.: Image copy-move forgery detection based on polar cosine transform and approximate nearest neighbor searching. Forensic Sci. Int. **224**(1–3), 59–67 (2013)
16. Lin, C., et al.: Copy-move forgery detection using combined features and transitive matching. Multimedia Tools Appl. **78**(21), 30081–30096 (2019)
17. Liu, Y., Xia, C., Zhu, X., Xu, S.: Two-stage copy-move forgery detection with self deep matching and proposal superglue. IEEE Trans. Image Process. **31**, 541–555 (2021)
18. . Lyu, Q., Luo, J., Liu, K., Yin, X., Liu, J., Lu, W.: Copy move forgery detection based on double matching. J. Vis. Commun. Image Represent. **76**(1), 103057 (2021)
19. Mahdian, B., Saic, S.: Detection of copy-move forgery using a method based on blur moment invariants. Forensic Sci. Int. **171**(2), 180–189 (2007)
20. Meena, K.B., Tyagi, V.: A copy-move image forgery detection technique based on tetrolet transform. J. Inf. Secur. Appl. **52**, 102481 (2020)
21. Muhammad, G., Hussain, M., Bebis, G.: Passive copy move image forgery detection using undecimated dyadic wavelet transform. Digit. Investig. **9**(1), 49–57 (2012)
22. Nazir, T., Nawaz, M., Masood, M., Javed, A.: Copy move forgery detection and segmentation using improved mask region-based convolution network (RCNN). Appl. Soft Comput. **131**, 109778 (2022)
23. Niu, P.P., Wang, C., Chen, W., Yang, H., Wang, X.: Fast and effective keypoint-based image copy-move forgery detection using complex-valued moment invariants. J. Vis. Commun. Image Represent. **77**, 103068 (2021)
24. Pan, X., Lyu, S.: Region duplication detection using image feature matching. IEEE Trans. Inf. Forensics Secur. **5**(4), 857–867 (2010)

25. Qin, J., Li, F., Xiang, L., Yin, C.: Detection of image region copy-move forgery using radial harmonic Fourier moments. J. Image Graph. **18**(8), 919–923 (2013)
26. Ryu, S.J., Kirchner, M., Lee, M.J., Lee, H.K.: Rotation invariant localization of duplicated image regions based on Zernike moments. IEEE Trans. Inf. Forensics Secur. **8**(8), 1355–1370 (2013)
27. Ryu, S.-J., Lee, M.-J., Lee, H.-K.: Detection of copy-rotate-move forgery using Zernike moments. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 51–65. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16435-4_5
28. Shivakumar, B., Baboo, S.S.: Detection of region duplication forgery in digital images using SURF. Int. J. Comput. Sci. Issues (IJCSI) **8**(4), 199–205 (2011)
29. Silva, E., Carvalho, T., Ferreira, A., Rocha, A.: Going deeper into copy-move forgery detection: exploring image telltales via multi-scale analysis and voting processes. J. Vis. Commun. Image Represent. **29**, 16–32 (2015)
30. Toldo, R., Fusiello, A.: Robust multiple structures estimation with J-linkage. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5302, pp. 537–547. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88682-2_41
31. Wang, C., Huang, Z., Qi, S., Yu, Y., Shen, G., Zhang, Y.: Shrinking the semantic gap: spatial pooling of local moment invariants for copy-move forgery detection. IEEE Trans. Inf. Forensics Secur. **18**, 1064–1079 (2023)
32. Wang, X.Y., Li, S., Liu, Y.N., Niu, Y., Yang, H.Y., Zhou, Z.L.: A new keypoint-based copy-move forgery detection for small smooth regions. Multimedia Tools Appl. **76**(22), 23353–23382 (2017)
33. Wang, X., Chen, W., Niu, P., Yang, H.: Image copy-move forgery detection based on dynamic threshold with dense points. J. Vis. Commun. Image Represent. **89**, 103658 (2022)
34. Wang, Y., Kang, X., Chen, Y.: Robust and accurate detection of image copy-move forgery using PCET-SVD and histogram of block similarity measures. J. Inf. Secur. Appl. **54**, 102536 (2020)
35. Weng, S., Zhu, T., Zhang, T., Zhang, C.: UCM-Net: a U-Net-like tampered-region-related framework for copy-move forgery detection. IEEE Trans. Multimedia 1–14 (2023)
36. Yang, F., Li, J., Lu, W., Weng, J.: Copy-move forgery detection based on hybrid features. Eng. Appl. Artif. Intell. **59**, 73–83 (2017)
37. Yang, J., Liang, Z., Gan, Y., Zhong, J.: A novel copy-move forgery detection algorithm via two-stage filtering. Digit. Signal Process. **113**, 103032 (2021)
38. Zandi, M., Mahmoudi-Aznaveh, A., Talebpour, A.: Iterative copy-move forgery detection based on a new interest point detector. IEEE Trans. Inf. Forensics Secur. **11**(11), 2499–2512 (2016)
39. Zhang, Y., et al.: CNN-Transformer based generative adversarial network for copy-move source/target distinguishment. IEEE Trans. Circuits Syst. Video Technol. **33**(5), 2019–2032 (2022)
40. Zhong, J.L., Pun, C.M.: An end-to-end dense-InceptionNet for image copy-move forgery detection. IEEE Trans. Inf. Forensics Secur. **15**, 2134–2146 (2019)
41. Zhu, Y., Chen, C., Yan, G., Guo, Y., Dong, Y.: AR-Net: adaptive attention and residual refinement network for copy-move forgery detection. IEEE Trans. Industr. Inf. **16**(10), 6714–6723 (2020)
42. Zhu, Y., Yu, Y., Guo, Y.: HRDA-Net: image multiple manipulation detection and location algorithm in real scene. J. Commun. **43**(1), 217–226 (2022)