Hindawi
Security and Communication Networks
Volume 2021, Article ID 6688393, 13 pages
https://doi.org/10.1155/2021/6688393

Research Article

CF Model: A Coarse-to-Fine Model Based on Two-Level Local Search for Image Copy-Move Forgery Detection

Fang Mei,[1,2] **Tianchang Gao**,[2,3] **and Yingda Lyu**[2,4]

[1]*College of Computer Science and Technology, Jilin University, Changchun 130012, China*
[2]*Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China*
[3]*College of Software, Jilin University, Changchun 130012, China*
[4]*Public Computer Education and Research Center, Jilin University, Changchun 130012, China*

Correspondence should be addressed to Yingda Lyu; ydlv@jlu.edu.cn

Received 10 December 2020; Revised 27 March 2021; Accepted 12 April 2021; Published 4 May 2021

Academic Editor: Honghao Gao

Copy-move forgery is the most predominant forgery technique in the field of digital image forgery. Block-based and interest-based are currently the two mainstream categories for copy-move forgery detection methods. However, block-based algorithm lacks the ability to resist affine transformation attacks, and interest point-based algorithm is limited to accurately locate the tampered region. To tackle these challenges, a coarse-to-fine model (CFM) is proposed. By extracting features, affine transformation matrix and detecting forgery regions, the localization of tampered areas from sparse to precise is realized. Specifically, in order to further exactly extract the forged regions and improve performance of the model, a two-level local search algorithm is designed in the refinement stage. In the first level, the image blocks are used as search units for feature matching, and the second level is to refine the edge of the region at pixel level. The method maintains a good balance between the complexity and effectiveness of forgery detection, and the experimental results show that it has a better detection effect than the traditional interest-based copy and move forgery detection method. In addition, CFM method has high robustness on postprocessing operations, such as scaling, rotation, noise, and JPEG compression.

1. Introduction

With the rapid development of technology worldwide, there are many ways to obtain and process images [1]. Evolutions in computer technology, the Internet, and image applications have allowed individuals to tamper easily with image content. Copy-move is the most common means of image forgery, in which a copy of a region is inserted into the same image. Two examples are shown in Figure 1, where the copy-move forgeries are used to enrich image content. Considering scenarios involving the court, news, and so on, it is of paramount importance to determine whether an image is tampered. The purpose of digital image forensics is to verify the authenticity of an image.

As one of the most common means of image tampering, copy-move forgeries may be accompanied by certain postprocessing, including JPEG compression, noise addition, and blurring, to change the image content and confuse the information recipient [2]. In particular, the copied area is often geometrically transformed (rotated, scaled, etc.). Therefore, the passive forensics of copy-move tampered images faces great technical challenges and has a strong practical application value. This paper studies the corresponding passive forensic techniques for copy-move operations.

Our main contributions can be summarized as follows:

(1) This paper proposes a coarse-to-fine model for detecting forged regions by the affine transformation matrix (CFM). The localization of the forged regions from sparse to accurate is achieved.

(2) To further extract the forgery region accurately, a two-stage local search algorithm is designed in the refinement stage to better maintain the balance

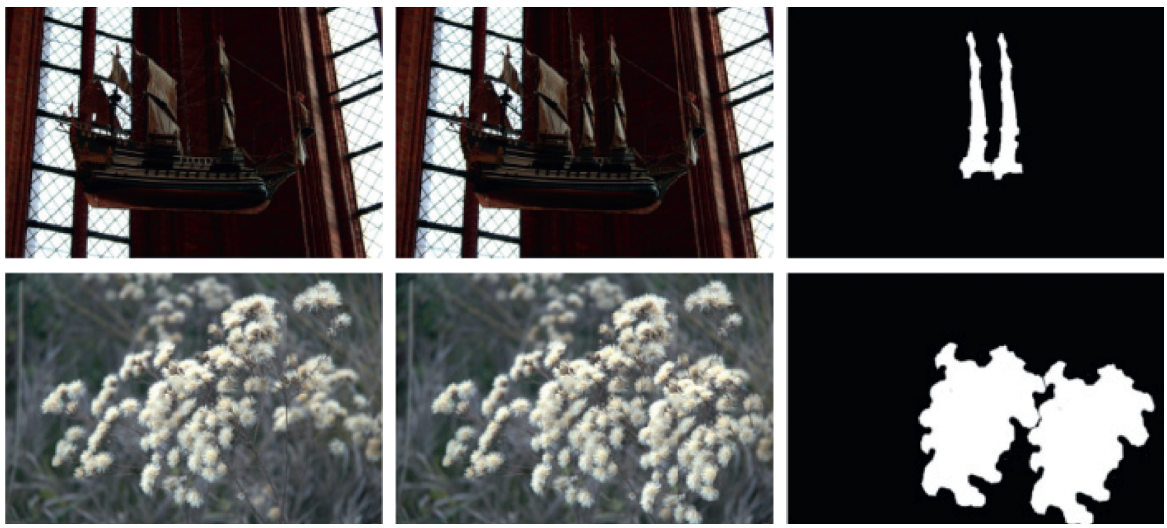


Figure 1: Two examples of the copy-move forgery. Left to right: original images, forged images through copy-move operations, and copy-move regions.

between complexity and effectiveness of forgery detection.

(3) The method has better detection results and higher robustness to postprocessing operations such as scaling, rotation, noise, and JPEG compression.

2. Related Work

Numerous methods for copy-move forgery detection (CMFD) have been proposed in the last decade, which are traditionally categorized into two classes: block-based and interest point-based methods.

2.1. Block-Based CMFD. In 2003, Fridrich [3] proposed the first CMFD algorithm which divided an input image into overlapping blocks to yield similar block pairs and used discrete cosine transform (DCT) to describe image blocks. LBP is a grey-scale texture operator which is used to describe the spatial structure of the image texture. Wang et al. [4] extracted Quaternion Exponent Moment (QEM) moduli from each overlapped circular color block. The main limitation of this method is the higher computational complexity, which can be reduced by applying super pixel theory. Chen et al. [5] proposed a scheme to detect copy-move regions through the invariant features extracted from each block, and each block was only compared with other blocks under the intersection of closed mean and variance features. Mahmood et al. [6] divided the approximation sub-band of the shift invariant stationary wavelet transform into overlapping blocks. Distinct features extracted from the overlapping blocks were used to expose tampered regions forged in digital images. The features of these algorithms can be classified as follows: invariant moments, dimension reduction, textural features, and polar transform. Matching techniques include dictionary sorting and Euclidean distance [7]. However, most algorithms based on image blocks do not perform well in resisting affine transformation attacks.

2.2. Interest Point-Based CMFD. Different from block-based algorithms, interest point-based CMFD algorithms are more robust against affine transformations. Unlike dividing an image, this method extracts interest points on the image, and image features are then extracted around the interest points. He et al. [8] used PCA on the feature vector to reduce computational complexity. Mohamadian and Pouyan [9] combined SIFT and Zernike moments to reduce the potential of being unable to detect tampered regions in flat regions. Pun et al. [10] proposed a novel CMFD scheme using adaptive oversegmentation and feature point matching, which integrates block-based and interest point-based forgery detection methods. Pandey et al. [11] proposed a fast and effective copy-move forgery detection algorithm through hierarchical feature point matching. Due to the high stability of intermediate and postprocessing operations, the SIFT method has been widely used in CMFD. To improve SIFT performance, Bay et al. [12] initially proposed the speeded-up robust features (SURF) technique. The SURF operator maintains the excellent performance of the SIFT operator but addresses the shortcomings of high computational complexity and time consumption. Bo et al. [13] proposed a CMFD technique based on SURF and extended the dimensions of Bay's techniques to 128 to reduce false matching. Many scholars have only used this technique in interest point detection to produce feature points, after which local features were employed to describe an interest point to achieve satisfactory results [14, 15]. Mishra et al. [16] presented a detection method based on the combination between speeded-up robust features (SURF) and hierarchical agglomerative clustering (HAC). Zandi et al. [17] proposed a new interest point detector that leverages the advantages of block-based and traditional interest point-based methods and uses improved strategies to implement

the algorithm. However, because the interest points are comparatively few and scattered, interest point-based detection methods can encounter difficulties in locating a precise forged region.

The block-based CMFD algorithm and interest point-based CMFD algorithm each have a similar framework as depicted in Figure 2 [18].

(i) Preprocessing: its main purpose is to eliminate irrelevant information in the image and restore useful real information; the most common approach is to convert the image from an RGB version to a grayscale image

(ii) Feature extraction: local image information is extracted from an image block or interest point represented by a feature descriptor

(iii) Matching: similar pairs of image blocks or points are determined during the matching process

Most existing algorithms based on image blocks suffer from some attacks, such as scaling, rotation, and noise addition, and interest point-based methods cannot locate the tampered region precisely. To solve these problems, a hybrid two-level method combining image blocks and interest points is proposed in this paper. We chose the SIFT as the feature descriptor to represent the interest point. Then, the adaptive oversegmentation method is used to improve the matching process and calculate the affine transformation matrix. Finally, the proposed local search algorithm is applied to image block level and pixel level, respectively, to locate the tampered region accurately.

3. Proposed Detection Algorithm

In this paper, an accurate CMFD method based on interest point and local search algorithm is proposed. The process is illustrated in Figure 3.

The main flow of the proposed algorithm is as follows: (1) feature extraction: interest points are detected in the input image represented by a feature descriptor, after which accurate interest point matches are obtained via a matching process; (2) affine transformation calculation: utilize a random verification algorithm to calculate the affine transformation matrix; (3) forgery region extraction: local search algorithm is applied to the image block level and the pixel level. The image block level realizes the location of the tampering region, and the pixel level is used to refine the tampering region boundary.

The image-level detection and pixel-level detection of the proposed model on the testing dataset show promising results. Our main contributions are as follows:

(i) A method combining image blocks and pixels is proposed. Based on the block, the forged region can be located, and the pixel points are used to make the area boundary more refined. This method can make up for the poor performance of only extracting tampered areas with points of interest, thereby improving detection performance.

(ii) Considering the balance between algorithm complexity and performance, design a two-level local search algorithm. In the first stage, the image is divided into small blocks by rectangular blocks. If the image block contains the point of interest, it is marked as a forgery unit and calculated by affine transformation. The search algorithm matches the result to get the forgery region. In the second stage, the boundary of the forged area is extracted at the pixel level, and a secondary search algorithm is used for improvement to further improve the accuracy of model detection.

(iii) Four different postprocessing operations were performed on the test dataset, and the experimental results show that our model still exhibits high robustness.

In the rest of this section, we present the process of this detection algorithm as illustrated in Figure 3. The details of our proposed algorithm are reflected in the following sections: Section 3.1 presents the feature extraction and description along with image segmentation using the adaptive oversegmentation algorithm to prepare for the next matching process. Section 3.2 outlines the feature-matching process using the two nearest neighbor (2NN) algorithm [19]. And then, the affine transformation is calculated. Section 3.3 introduces the local search algorithm. In Section 3.4, two-level local search algorithm using affine transformation matrix is utilized to locate the tampered region accurately. In the first stage, the image blocks are used as search units for feature matching, and the second stage is at the pixel level to refine the edge of the region.

3.1. Feature Extraction and Adaptive Oversegmentation. The first phase of the proposed algorithm involves interest point detection and feature extraction based on SIFT features, referring to local features of an image. SIFT remains invariant to rotation, scaling, and light intensity and maintains stable robustness to changes in the viewing angle, affine transformation, and noise. The interest points and their corresponding descriptors are obtained. Based on these results, the proposed algorithm performs a matching operation to identify similar local regions.

To obtain good performance in matching and calculation of the affine transformation matrix, the adaptive oversegmentation method is adopted [10]. Next, we find corresponding interest point pairs via the feature matching process. In our proposed method, the segmentation algorithm is simple linear iterative clustering (SLIC). SLIC algorithm can generate compact and nearly uniform superpixel, and has high comprehensive evaluation in terms of operation speed, object contour preservation, and superpixel shape, which is more in line with the expected segmentation effect. When the SLIC segmentation method is used, the balance between computational cost and detection precision must be guaranteed. Therefore, the adaptive over segmentation algorithm is adopted to adaptively define the size of superpixels according to the texture of the test images.

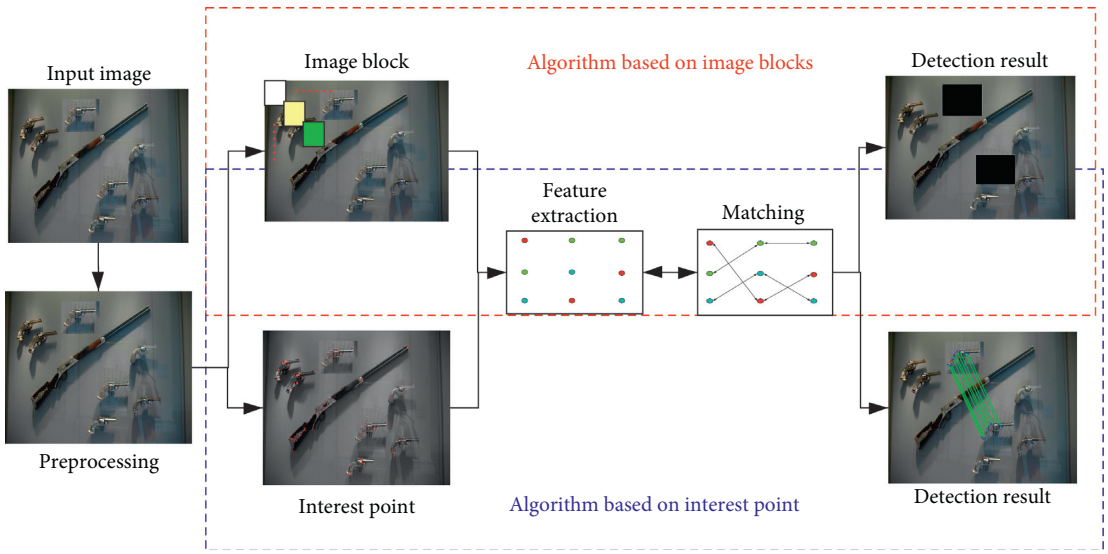


Figure 2: Common copy-move forgery detection framework-based CMFD.

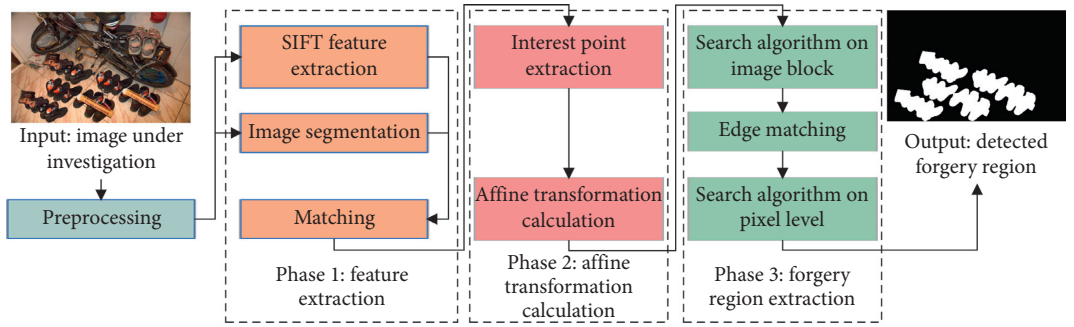


Figure 3: Framework of the proposed copy-move forgery detection method.

Next, a segmented image builds the image blocks set $B = (B_1, B_2, \ldots B_i, \ldots B_{NB})$, where NB is the total number of image blocks; the interest points and feature descriptors in the i^{th} image block are stored in B_i. Figure 4 depicts the relationship of the block set. Then, we find the corresponding interest point pairs via the feature-matching process.

3.2. Interest Point Matching and Affine Transformation Calculation. The 2NN algorithm utilizes the ratio of the distance between the nearest neighbor and the second nearest neighbor. If image blocks B_i and B_j must match, for any feature point, where is the k^{th} point in block B_i, the calculation is as follows:

$$T_b > \frac{T_1}{T_2}, \tag{1}$$

where T_b is the similarity threshold, d_1 is the closest neighbor, and d_2 is the second closest neighbor. The distance d_m is calculated as

$$d_m = \left\| f_i^k - f_j^m \right\|_2, \tag{2}$$

where d_m denotes the distance between point p_i^k and point p_i^m. p_j^m is the m^{th} point in P_j, and f_i^k and f_j^m are the corresponding feature descriptors.

In our experiment, T_b is set to 0.2. If constraint (1) is satisfied, then the inspected interest point p_i^k is matched with p_j^m (p_i^k and p_j^m denotes the interest pairs).

We iterate the 2NN process in different image blocks in our experiment until all blocks have been traversed, resulting in a dataset: $MP = \left\{M_1^2, M_1^3, M_1^4, \ldots M_i^j, \ldots M_{m-1}^m\right\}$, where the interest pairs between B_i and B_j are stored in M_i^j.

Matching operations between image blocks can avoid failed matching due to the proximity of points to coordinates. To further prevent match failure, assuming that M_i^j exists in MP, if the number of point pairs in M_i^j is too small, then the point pairs between the image blocks B_i and B_j are considered a failure and must be deleted. As such,

$$T_p \geq \text{size}(MP[x]), \tag{3}$$

where size () represents the number of point pairs in MP $[x]$ and the threshold T_p is set to 3 to filter the failed pairs. Thus, most missed matches are filtered.

To better display the tampered region, affine transformation matrix T is used to describe the relationship between

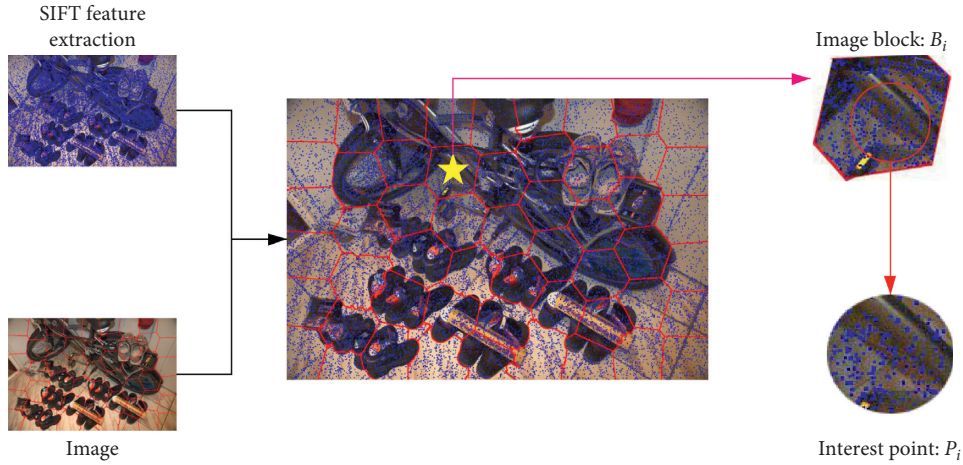


Figure 4: Framework of the proposed copy-move forgery detection method.

the source region and replication regions. The traditional method of estimating affine transformation is not suitable for the algorithm in this paper. In our proposed method, we propose a more efficient matrix estimation algorithm. If M_i^j exists in MP, then we randomly extract three point pairs M_i^j and store them in $C_{\text{matrix}} = \{(p_1, p_1'), (p_2, p_2'), (p_3, p_3')\}$. The affine transformation matrix T is described as follows:

$$\left|y'\right| = T|y|, \tag{4}$$

where the affine transformation matrix T is represented as

$$T = \left| a_3 \;\; a_4 \;\; t_v \right|, \tag{5}$$

where t_x and t_y denote translations and a_1, a_2, a_3, and a_4 are associated with scaling and rotation. C_{matrix} can obtain the affine transformation matrix T.

To verify the accuracy of matrix T, all point pairs in M_i^j must be tested using this matrix. For any interest point pairs (p, p') in M_i^j, point p can obtain the corresponding interest point p' using the following equation:

$$p' = T * p. \tag{6}$$

We verify the matrix accuracy based on the distance between $p_,$ and p'.

$$D_p = \left|x' - x_,\right| + \left|y' - y_,\right| < T_d, \tag{7}$$

where x', y', and x', y' are the coordinates of $p_,$ and p'. T_d is the similarity threshold of the matrix ($T_d = 1.5$ in our experiment). Then, we obtain the number of right point pairs count in M_i^j.

When rate is greater than 0.5, the matrix T is considered correct. In this case,

$$\text{rate} = \frac{\text{count}}{\text{size}}\left(M_i^j\right), \tag{8}$$

where size (M_i^j) is the amount of all point pairs in M_i^j.

In most cases, the source region and replication region may be covered by many image blocks. Many affine transformation matrices can be obtained through MP. We propose an algorithm to deal with this problem. Whenever any set M in MP must be calculated, we must examine the relationship between point pairs in M and existing matrix using formulas (6)–(8). If the label *rate* is more than 0.5, the set M is not to be calculated. Finally, the matrix set is described as follows:

$$T_{\text{end}} = \{T_1, T_2, T_3, \ldots\}. \tag{9}$$

Next, we will display the tampered region in the search algorithm.

3.3. Local Search Algorithm. Extracting the tampered region using only the interest point results in poor performance. By considering the balance between algorithm complexity and performance to more accurately extract the forgery region, we propose a local search algorithm that can be applied at the image block level and pixel level. The role of the local search algorithm is described in Figure 5, where the grid is used to replace the test image, the region outlined in red is the forged region, and the blue small block is the forged unit; when the first search algorithm is used, the forged unit is an image block, and the second forged unit is a pixel. Details of the search algorithm are provided in the following section.

The detection unit can find a corresponding unit via the affine transformation matrix, which is key to the local search algorithm. The detected unit can find corresponding unit through the matrix. Before executing the search algorithm, the forged units must be collated and added to the forgery region set (TR). Then, the local search algorithm is executed; steps are shown in Algorithm 1.

TR_{cnt} is the result of the current detection, D_{nei} is the set of neighborhood $p_{\text{nei}} = \{p_1, p_2, p_3, p_4\}$, and (1, 2, 3, 4) denotes four angles (0°, 90°, 180°, and 270°). Notably, the detection unit in p_{nei} may be the detected element; therefore, the detected elements in p_{nei} must be deleted. Then, the corresponding unit p_i is calculated by matrix T, and feature descriptors are used to measure the similarity. These descriptors are explained in detail in the following section.

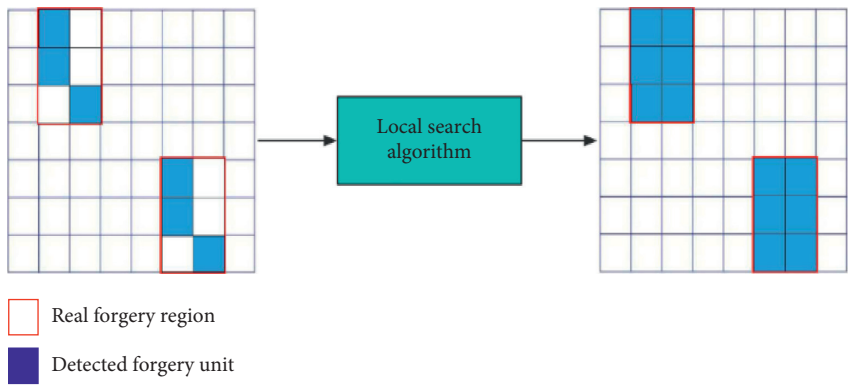


Figure 5: Role of local search algorithm. The region outlined in red is the forged region, and the blue small block is the forged unit.

The successfully matched unit pairs are added to $TR_{\text{cnt}+1}$. This operation is iterated until all elements in TR_{cnt} have been detected. Finally, the test result $TR_{\text{cnt}+1}$ is combined with the original result $TR_{\text{cnt}+1}$, and we obtain the final result $TR_{\text{cnt}+1} = TR_{\text{cnt}+1}$ U TR_{cnt}. To understand the algorithm flow and prove the validity of the local search algorithm, a flow chart is used for descriptive purposes (Figure 6).

Figure 6 presents the ordinary flow of the local search algorithm. There are only six forged units (a, b, c, a′, b′, and c′) at the beginning of the algorithm; the forged region is not completely covered. Implementation steps of the algorithm are described in Figure 6, where the blue blocks are forged units, green tags stand for detecting units, red blocks are nonforged units, and white blocks are units that have not been detected. Assume that there is only one affine transformation matrix T, and the final result was shown.

3.4. Tampered Region Localization. To balance the complexity and accuracy of the algorithm, the two-stage local search algorithm is proposed: the image block level. And, the second stage is at the pixel level to refine the edge of the tampered region. The framework of the algorithm is displayed in Figure 7.

3.4.1. The First Stage. In our method, interest points in the *MP* are extracted and stored in P_{right}. First, a small, nonoverlapping rectangular block is used to cover the host image, and all image blocks are scanned. If the image block contains interest points in P_{right}, the block is marked as a forged unit. Then, the image blocks as a detection unit are added to TR_0, and the search algorithm is employed on the image block level. Corresponding image blocks are calculated by the affine transformation T. Assume that image block B_i calculates corresponding image block B_i; in this case, image block B_i cannot reach the center of another block (B_i) and needs to extract the true matching image block B_i, so feature comparison must be executed between B_i and B_i. Then, the ZNCC (zero-based normalized cross-correlation) should be calculated between B_i and B_i as follows:

$$C(x) = \frac{\sum_{u\in B_i}(I(u) - \overline{I})(I''(u) - \overline{I''})}{\sqrt{\sum_{u\in B_i}(I(u) - \overline{I})^2}\sqrt{\sum_{u\in B_i''}(I''(u) - \overline{I''})^2}}, \quad (10)$$

where $I(u)$ and $I''(u)$ denote pixel intensities at location u, and I and I are the average pixel intensities of B_j and B. We apply a Gaussian filter of 7×7 pixels with a standard deviation of 0.5 to reduce noise; the threshold (T_{RD}) is set up to obtain similar image block pairs:

$$C(x) \geq T_{R\,D}. \quad (11)$$

In our work, T_{RD} is set to 0.55 once formula (11) has been calculated. The two image blocks $(B_i$ and $B_j)$ are similar, and the results of the search algorithm are stored in TR_1.

A filtering algorithm is used to render the test results more accurate. For each forged unit in TR_1, the neighbor of detection element D must be extracted, and the neighboring blocks are defined as $D_{\text{nei}} = \{d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7\}$. In our experiment, if the number of forged units in D_{nei} is less than 2, the detection element D is deleted.

3.4.2. The Second Stage. It is challenging to extract the forgery region at the image block level, and the algorithm does not have good performance at the edge of the tampered region. Thus, the edge of TR_1 is extracted, and we obtain an edge region ER_0 and a center region CR_1 on the image block level, where ER_0 is considered inaccurate and CR_1 is accurate. In matrix T, all pixels in ER_1 must be calculated. For the obtained pixel pairs, the ZNCC algorithm is used to measure similarities, and the threshold (TDR) is set to 0.55. The matching result is saved in ER_1, from which, forgery region TR_2 is obtained by combining the center region CR_1 and the matching result ER_1. To improve the edge of the forged region, the edge of ER_2 is extracted at pixel level in TR_2, and ER_2 is used to execute

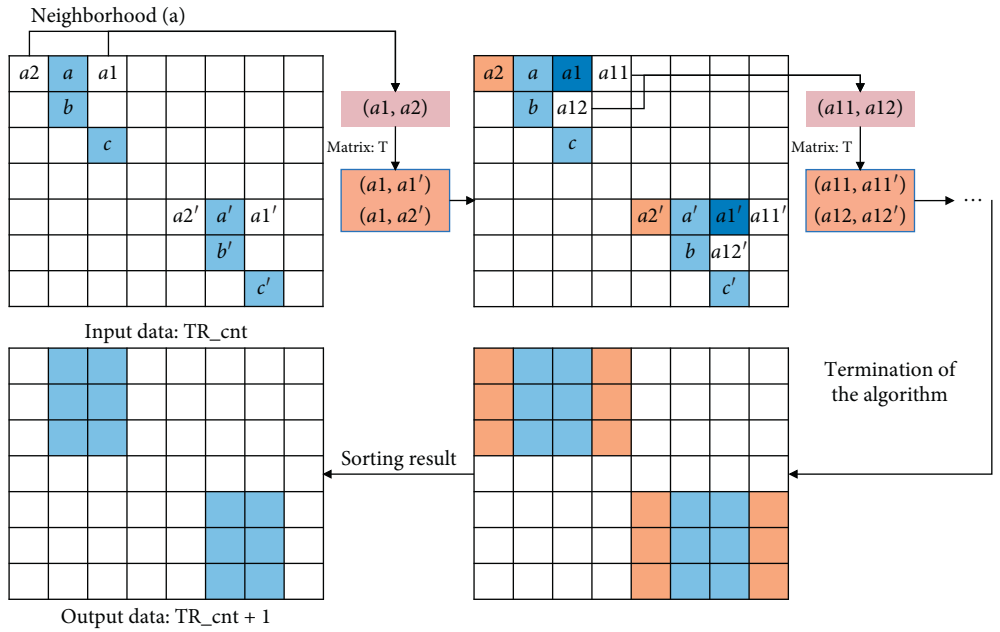


Figure 6: Framework of the local search algorithm. The blue blocks are forged units, green tags stand for detecting units, red blocks are nonforged units, and white blocks are nondetected units.

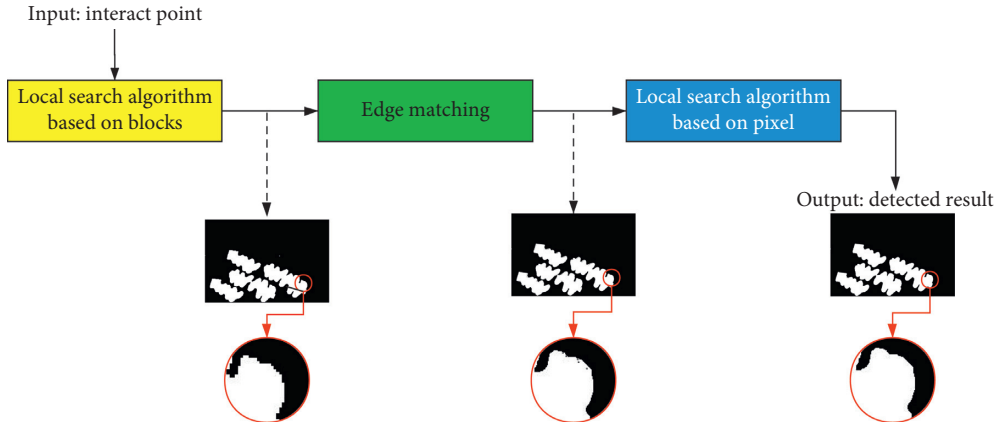


Figure 7: Framework of two-stage tampered region localization algorithm.

local search algorithm. Assume that we get (I, I') by matrix T; the color feature should be extracted, respectively, between I and I' as follows:

$$F_I = \frac{R(E_I) + G(E_I) + B(E_I)}{3},$$
$$F'_I = \frac{R(E_{I'}) + G(E_{I'}) + B(E_{I'})}{3}, \tag{12}$$

where R (), G (), and B () are three color channels of the detected image unit; F_I, $F_{I'}$ are the color features of I and I'; and if feature F_I and $F_{I'}$ conform to formula (11), matching is successful between unit I and I'.

$$\left|F_I - F_{I'}\right| \le T_{RD^2}, \tag{13}$$

where T_{RD^2} is the degree of similarity between I and I'. In our work, T_{RD^2} is 0.5. Results are stored in ER_3.

The tampered region TR_2 is obtained by combining ER_3 and center region CR_2. After the filtering step, the morphological close operation is applied to TR_3 to eliminate small gaps, after which the tampered region TR_{end} is generated. The algorithm is evaluated in the following section to demonstrate its effectiveness.

4. Experimental Results

In this section, a series of experiments are conducted to evaluate the performance of the proposed CMFD method. Section 4.1 introduces the image dataset used in our experiments and the evaluation criteria used to evaluate the performance of the proposed method. Section 4.2 shows the experimental results of the proposed algorithm. In section 4.3, the experimental results of the proposed CMFD method were finally compared with existing state-of-the-art CMFD

methods under different transforms, and the results of comparative analysis were outlined.

4.1. Datasets and Evaluation Criteria. In the following experiments, a benchmark database [20] that includes realistic copy-move forgeries was used to test the proposed scheme. This image dataset included 48 source images along with manually prepared per image, semantically meaningful regions to be copied. Each image measured 3000×2300 pixels. Forgery regions comprised approximately 10% of each image. The copied regions belonged to the categories of living, natural, artificial, and mixed textures ranging from smooth to complex. Transformed images, such as those that underwent rotation, scaling, JPEG artifacts, and added noise, were also included in the image dataset.

To quantitatively evaluate the detection performance, we adopted two metrics: precision and recall. Precision is the fraction of pixels identified as forgery that are truly forgery, defined as the ratio of the number of correctly detected forged pixels to the total number of detected forged pixels. Recall refers to the fraction of forged pixels that are correctly classified, defined as the ratio of the number of correctly detected forged pixels to the number of forged pixels in the ground truth forgery image. Precision and Recall are calculated using (14) and (16), where Ω denotes the set of the detected forged regions in forged images with the CMFD method at the pixel level and Ω' denotes the forged regions of the ground-truth of forged images. We provide the F_i score as a measure that combines precision and recall in a single value.

Using these metrics, we show how precisely the CMFD algorithms identified tampered regions. To reduce the effects of random samples, the average precision and recall were computed for all images in the dataset.

4.2. Experimental Results of the Proposed Algorithm

4.2.1. Experimental Results on Plain Copy-Move Forgery. Plain copy-move forgery is a kind of one-to-one copy-move method that does not involve other transformation operations. It is to cut the local area of the target image and then paste it into the target image again through rotation, scaling, and other operations to generate a new tampered image. We experimented on 48 plain copy-move forgery images in total. Figure 8 displays eight copy-move forgery detection results for the plain copy-move forgery, and the forgery content is either smooth (e.g., sky), rough (e.g., rocks), or structured (typically man-made buildings). From top to bottom are test images and corresponding ground-truth forged regions, and the final row is forged region detected by the CFModel. As can be seen from the figure, the proposed model obtains fine prediction masks and even in small forgery region. These groups can be used as categories for CMFD images.

$$\text{precision} = \frac{\left|\Omega \cap \Omega'\right|}{\Omega'}, \tag{14}$$

$$\text{recall} = \frac{\left|\Omega \cap \Omega'\right|}{\Omega'}, \tag{15}$$

4.2.2. Experimental Results under Various Attacks. In addition to one-to-one copy-move forgery, we also experimented on the various attacks to verify the effectiveness of the proposed algorithm.

(i) Scale: the tampered region is rescaled to between 91% and 109% of their original size with 2% step length.

In total, $48 \times 10 = 480$ images are experimented. Figure 9 displays eight copy-move forgery detection results for the scaling, and some scale resizing parameters are included: 91%, 93%, 95%, 97%, 103%, 105%, 107%, and 109%.

(ii) Rotation: the tampered region is rotated at a rotation angle varying from 2° to 10° with a step length of 2°. In total, $48 \times 5 = 240$ images are experimented. Figure 10 shows eight copy-move forgery detection results for the rotation, and some rotation angles, i.e., 2°, 4°, 6°, 8°, and 10°, are considered.

(iii) Gaussian noise: the image intensities of the tampered region is normalized between 0 and 1 with added zero-mean Gaussian noise with standard deviations of 0.02 to 0.10 and a step length of 0.02. In total, $48 \times 5 = 240$ images are experimented. Figure 11 illustrates eight copy-move forgery detection results for noise, and noise standard deviations are included: 0.02, 0.04, 0.06, 0.08, and 0.1.

(iv) JPEG compression: the forged image is JPEG compressed with quality factors varying between 100 and 20 and a step length of 10. In total, $48 \times 9 = 432$ images are experimented. Figure 12 shows eight copy-move forgery detection results for the JPEG, quality factor (QF) which included: 20, 30, 40, 50, 60, 70, 80, and 90.

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{16}$$

4.3. Comparative Analysis of Algorithms. This section presents the comparison results between CFModel and the existing methods, and experiments on the dataset proposed in [20] including 1488 tampered images. Three recent methods based on SIFT [20] and SURF [20] along with iterative CMFD [17] were selected for comparison.

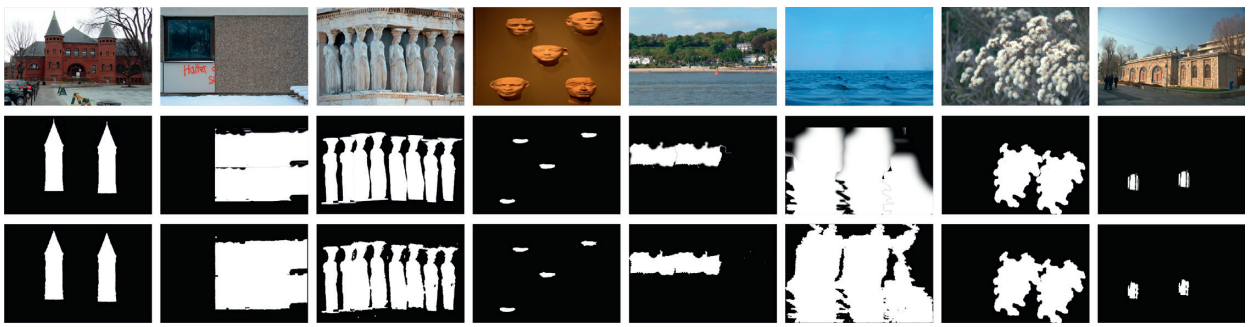


Figure 8: Copy-move forgery detection results under plain copy-move forgery. Top to bottom: test images from dataset, ground-truth forged regions, and forged regions detected by CFModel.

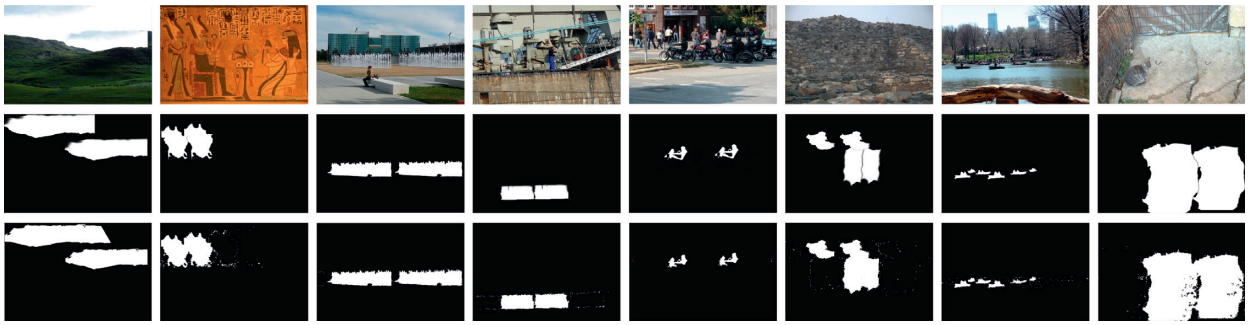


Figure 9: Copy-move forgery detection results for the scale. Top to bottom: test images from dataset, ground-truth forged regions, and forged regions detected by the proposed algorithm. Left to right represents the scale of 91%, 93%, 95%, 97%, 103%, 105%, 107%, and 109%.

Figure 10: Copy-move forgery detection results for the rotation. Top to bottom: test images from dataset, ground-truth forged regions, and forged regions detected by the proposed algorithm. Left to right, the angle of rotation is 2°, 4°, 4°, 6°, 6°, 8°, 8°, and 10°.

Figure 11: Copy-move forgery detection results for noise. Top to bottom: test images from dataset, ground-truth forged regions, and forged regions detected by the proposed algorithm. Left to right: the standard deviation is 0.02, 0.04, 0.04, 0.06, 0.06, 0.08, 0.08, and 0.1.

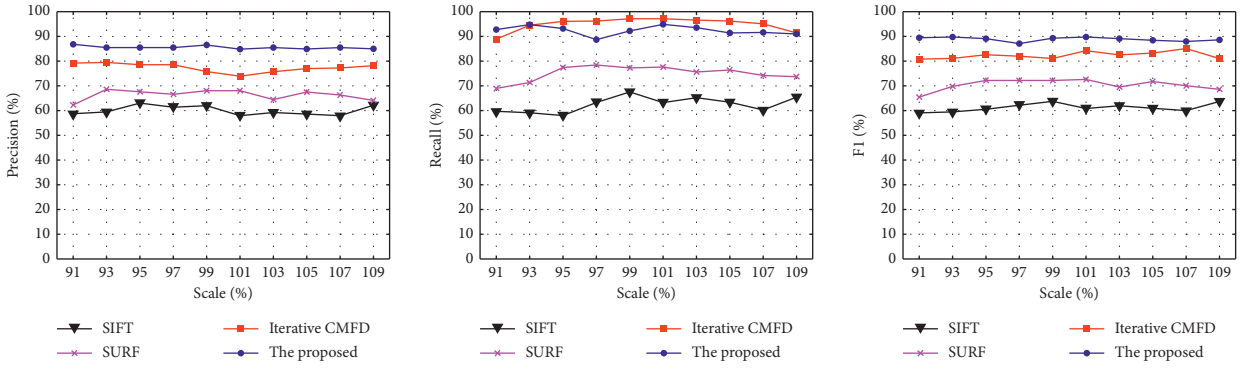
Figure 12: Copy-move forgery detection results for the JPEG. Top to bottom: test images from dataset, ground-truth forged regions, and forged regions detected by the proposed algorithm. Left to right, the quality factor is 20, 30, 40, 50, 60, 70, 80, and 90.

Input: forgery region set (TR_{cnt}) (block or pixel), affine transformation matrix (T_{end})
Output: forgery region set ($TR_{\text{cnt}+1}$)
(1) Detection unit p selected from TR_{cnt} and obtain the neighborhood p_{nei}; elements that have been detected in p_{nei} are deleted. p_{nei} is added to the set D_{nei}.
(2) Nondetection unit p_i is removed from D_{nei} and obtain detection unit p_i' by T. Calculate the similarity between p_i and p_i'; if successful, p_i and p_i' are added to $TR_{\text{cnt}+1}$, and the neighborhood of p_i is added to D_{nei}. Continue to execute step 2 until D_{nei} is empty.
(3) Iterate steps 1 and 2 until all elements in TR_{cnt} have been detected.

Algorithm 1: Local search algorithm.

Table 1: Detection results of plain copy-move forgery at image level.

Image level	Precision (%)	Recall (%)	F_1 (%)
SIFT [20]	88.37	79.17	83.52
SURF [20]	91.47	89.58	90.52
Iterative [17]	67.14	97.91	79.66
Hybrid [21]	78.33	97.92	87.04
CFModel	**97.82**	**93.75**	**95.74**



(a) (b) (c)

Figure 13: Continued.

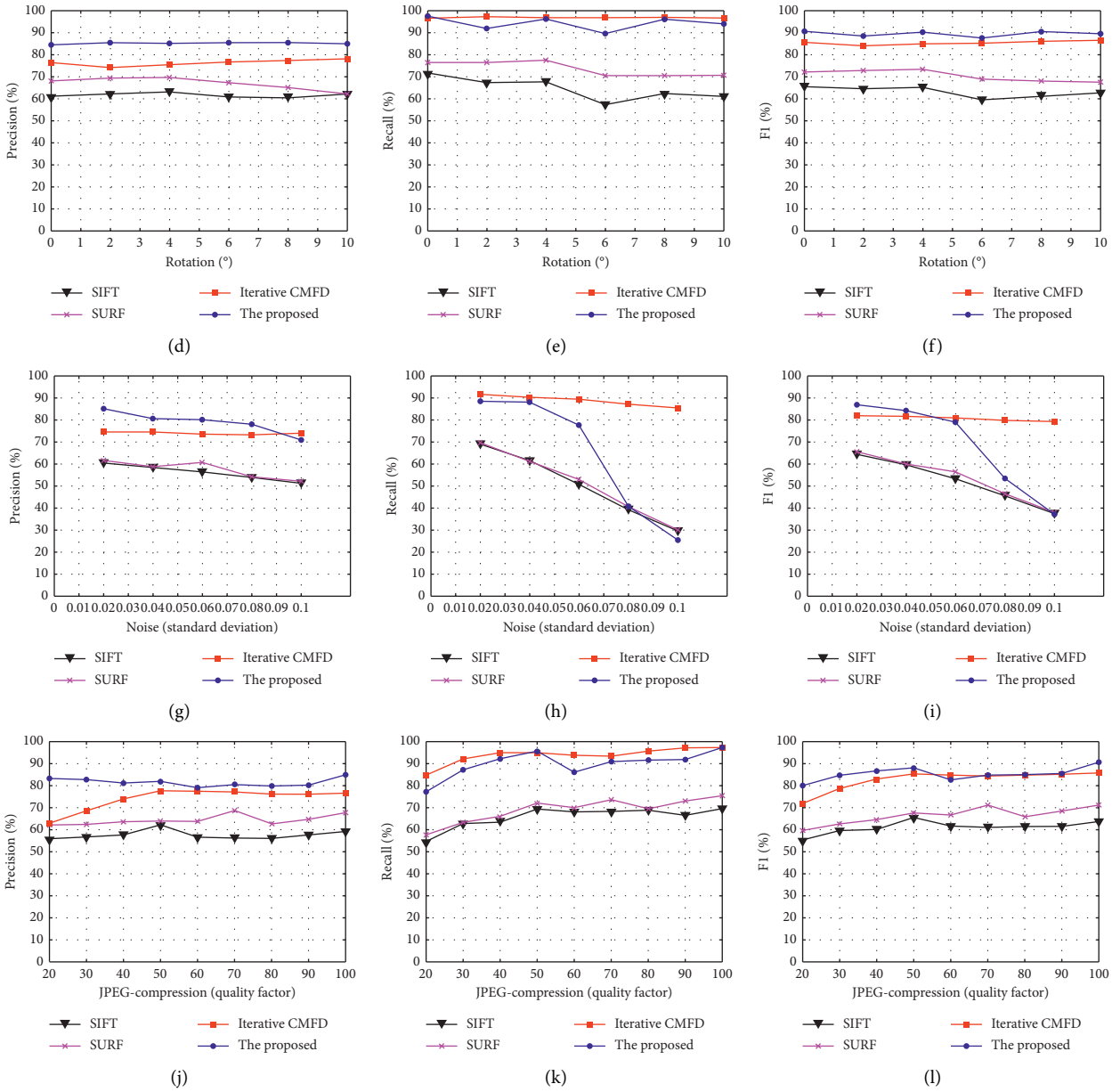


Figure 13: Detection results under various attacks. Top to bottom: scale, rotation, Gaussian noise addition, and JPEG compression. Left to right: precision, recall, and F_1.

4.3.1. Detection Results under Plain Copy-Move Forgery. We first evaluated our algorithm under plain copy-move forgery attack. We experimented on 48 original images and 48 forged images, which are tampered by one-to-one copy-move forgery. Tables 1 and 2 present the results of the evaluation at the image level and pixel level.

As noted in Table 1, the CFModel achieved 97.82% precision and 93.75% recall, better than the most state-of-the-art methods at image level. Our scheme also achieved better performance at the pixel level. As indicated in Table 2, the CFModel achieved up to 84.58% precision and up to 97.41% recall, surpassing most state-of-the-art methods. Compared to Bi [22] and Chen [23], *F1* score is slightly lower than them. The possible reason is that the proposed model is based on block and interest point, which focuses more on recall rate (whether the forged pixel is checked completely and correctly). These results show that the proposed method is more effective than others. Figure 8 also provides the representative results of eight examples. As is shown in the figure, we can see that our proposed algorithm can accurately locate the tampered region even in those small or smooth copy-move regions.

4.3.2. Detection Results under Various Attacks. In order to obtain a more detailed assessment of the discriminative properties of the method, the detailed data of copy-move forgery detection results, experimented on 1392 tampered images under various attacks in total, are shown in Figure 13.

Table 2: Detection results of plain copy-move forgery at pixel level.

Pixel level	Precision (%)	Recall (%)	Fi (%)
SIFT [20]	60.80	71.48	65.71
SURF [20]	68.13	76.43	72.04
Iterative [17]	72.23	96.46	82.61
Hybrid [21]	90.27	78.61	84.04
Bi [22]	—	—	92.87
Chen [23]	—	—	93.92
CFModel	**84.58**	**97.41**	**90.54**

"/" denotes that corresponding results are not provided in the literature.

We use 1392 images in total under different attacks. Figure 13 provides all qualitative results: top to bottom—scale attack, rotation attack, Gaussian noise addition, and JPEG compression; left to right—precision rate, recall rate, and $F1$ score.

As shown in the figure, the precision rate and recall rate of our scheme reached a higher level than other methods, the $F1$ score was particularly prominent under scale indicating that our method provides a good balance of precision and recall. The main reason is that our method proposes a two-stage local search algorithm, which can not only locate the tampered region at the image block level but also locate the edge at the pixel level. In other words, our scheme performed better than most state-of-the-art methods in most cases; however, our method has a very low score when the standard deviation exceeds 0.6, and we will address this deficiency in subsequent work.

5. Conclusion

With the development of digital technology, digital images can be easily forged using image processing software. Forged images must be identified given the potential legal and other implications. In this paper, we propose a copy-move forgery detection algorithm using SIFT as the interest point and feature extraction method. The affine transformation matrix was then calculated, followed by a local search algorithm to locate the forged region. Experimental results show that the proposed scheme performs much better than state-of-the-art copy-move forgery detection algorithms and demonstrates good performance under various attacks. However, performance was poor when images contained noise; we will focus on this image type in later work.

Future research is mainly as follows:

(1) To address the problem that the method cannot adapt to noisy operations, future plans are to incorporate richer texture feature information to achieve better robustness

(2) In future work, we will focus on detection tasks with multiple copy-move tampered regions at the same image to realize practical applications of the detection algorithm

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Key Research and Development Program of China (2018YFB080402 and 2018YFB0804203), Regional Joint Fund of NSFC (U19A2057), the National Natural Science Foundation of China (61672259 and 61876070), Jilin Province Science and Technology Development Plan Project (20190303134SF and 20180201064SF), CERNET Innovation Project (NGII20190802), and Undergraduate Innovation and Entrepreneurship Training Program of Jilin University (202010183389).

References

[1] Y. Yin, Z. Cao, Y. Xu, H. Gao, R. Li, and Z. Mai, "QoS prediction for service recommendation with features learning in mobile edge computing environment," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1136–1145, 2020.

[2] Z. Shi, X. Shen, H. Chen, and Y. Lyu, "Global semantic consistency network for image manipulation detection," *IEEE Signal Processing Letters*, vol. 27, pp. 1755–1759, 2020.

[3] A. J. Fridrich, B. D. Soukal, and A. J. Lukáš, "Detection of copy-move forgery in digital images," in *Proceedings of the 2003 Digital Forensic Research Workshop*, Cleveland, OH, USA, August 2003.

[4] X.-Y. Wang, Y.-N. Liu, H. Xu, P. Wang, and H.-Y. Yang, "Robust copy-move forgery detection using quaternion exponent moments," *Pattern Analysis and Applications*, vol. 21, no. 2, pp. 451–467, 2018.

[5] C.-C. Chen, H. Wang, and C.-S. Lin, "An efficiency enhanced cluster expanding block algorithm for copy-move forgery detection," *Multimedia Tools and Applications*, vol. 76, no. 24, pp. 26503–26522, 2017.

[6] T. Mahmood, Z. Mehmood, M. Shah, and Z. Khan, "An efficient forensic technique for exposing region duplication forgery in digital images," *Applied Intelligence*, vol. 48, no. 7, pp. 1791–1801, 2018.

[7] H. Gao, C. Liu, Y. Li, and X. Yang, "V2VR: reliable hybrid-network-oriented V2V data transmission and routing considering RSUs and connectivity probability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–14, 2020.

[8] H. He, X. Huang, and J. Kuang, "Exposing copy move forgeries based on a dimension reduced SIFT method,"

Information Technology Journal, vol. 12, no. 14, pp. 2957–2979, 2013.

[9] Z. Mohamadian and A. A. Pouyan, "Detection of duplication forgery in digital images in uniform and nonuniform regions," in *Proceedings of the 2013 UKSim 15th International Conference on Computer Modelling and Simulation*, pp. 455–460, Cambridge, UK, April 2013.

[10] C. M. Pun, X. C. Yuan, and X. L. Bi, "Image forgery detection using adaptive over-segmentation and feature point matching," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1705–1716, 2015.

[11] R. C. Pandey, S. K. Singh, K. K. Shukla, and R. Agrawal, "Fast and robust passive copy-move forgery detection using SURF and SIFT image features," in *Proceedings of the 2014 2014 9th International Conference on Industrial and Information Systems (ICIIS)*, pp. 1–6, Gwalior, India, December 2014.

[12] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: speeded up robust features," in *Computer Vision-ECCV 2006*, pp. 404–417, Springer, New York, NY, USA, 2006.

[13] X. Bo, W. Junwen, L. Guangjie, and D. Yuewei, "Image copy-move forgery detection based on SURF," in *Proceedings of the 2010 International Conference on Multimedia Information Networking and Security*, pp. 889–892, Nanjing, Jiangsu, China, November 2010.

[14] H. Gao, W. Huang, Y. Duan et al., "The cloud-edge-based dynamic reconfiguration to service workflow for mobile ecommerce environments: a QoS prediction perspective," *ACM Transactions on Internet Technology*, vol. 21, no. 1, pp. 1–23, 2020.

[15] X. Yin, M. Cao, and S. Zhou, "An approach to alleviate the sparsity problem of hybrid collaborative filtering based recommendations: the product-attribute perspective from user reviews," *Mobile Networks Applications*, vol. 25, no. 2, pp. 376–390, 2020.

[16] P. Mishra, N. Mishra, S. Sharma, and R. Patel, "Region duplication forgery detection technique based on SURF and HAC," *The Scientific World Journal*, vol. 2013, no. 17, pp. 1–8, 2013.

[17] M. Zandi, A. Mahmoudi-Aznaveh, and A. Talebpour, "Iterative copy-move forgery detection based on a new interest point detector," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2499–2512, 2016.

[18] M. Zandi, A. Mahmoudi-Aznaveh, and A. Mansouri, "Adaptive matching for copy-move forgery detection," in *Proceedings of the 2014 IEEE International Workshop on Information Forensics and Security. (WIFS)*, pp. 119–124, Atlanta, GA, USA, December 2014.

[19] X. Pan and S. Lyu, "Region duplication detection using image feature matching," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 6, pp. 857–867, 2010.

[20] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1841–1854, 2012.

[21] F. Yang, J. Li, W. Lu, and J. Weng, "Copy-move forgery detection based on hybrid features," *Engineering Applications of Artificial Intelligence*, vol. 59, pp. 73–83, 2017.

[22] X. Bi and C.-M. Pun, "Fast copy-move forgery detection using local bidirectional coherency error refinement," *Pattern Recognition*, vol. 81, pp. 161–175, 2018.

[23] B. Chen, M. Yu, Q. Su, H. J. Shim, and Y.-Q. Shi, "Fractional quaternion Zernike moments for robust color image copy-move forgery detection," *IEEE Access*, vol. 6, pp. 56637–56646, 2018.