Received: 4 March 2022 | Revised: 8 June 2022 | Accepted: 20 June 2022

DOI: 10.1049/ipr2.12578

IET Image Processing

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

SMDAF: A novel keypoint based method for copy-move forgery detection

Guangyu Yue[1,3] | **Qing Duan**[1,2] | **Renyang Liu**[3,4] | **Wenyu Peng**[1,3] | **Yun Liao**[1,2] | **Junhui Liu**[1,2]

[1]National Pilot School of Software, Yunnan University, Kuming, China

[2]Key Laboratory in Software Engineering of Yunnan Province, Yunnan University, Kuming, China

[3]Engineering Research Center of Cyberspace, Yunnan University, Kuming, China

[4]School of Information Science and Engineering, Yunnan University, Kuming, China

Correspondence
Qing Duan, National Pilot School of Software, Yunnan University, Kuming 650000, China.
Email: qduan@ynu.edu.cn

Funding information
Open Foundation of Key Laboratory of in Software Engineering of Yunnan Province, Grant/Award Number: 2020SE307; the Yunnan Province Science Foundation for Youths, Grant/Award Number: No.202005AC160007; Scientific Research Fund of Yunnan Provincial Education Department, Grant/Award Number: 2021J0007; National Natural Science Foundation of China, Grant/Award Numbers: 62101480, 62162067, 61762089, 61763048

Abstract

Copy–move forgery poses a significant threat to social life and has aroused much attention in recent years. Although many copy-move forgery detection (CMFD) methods have been proposed, the most existing CMFD methods are short of adaptability in detecting images, which leads to the limitation on detection effects. To solve this problem, the paper proposes a novel keypoint-based CMFD method: second-keypoint matching and double adaptive filtering (SMDAF). Motivated by image matching based on keypoint, the second-keypoint matching method is designed to match keypoints extracted from copy–move forgery images, which can be used for both the single-CMFD and the multiple-CMFD. Then, a double adaptive filter (DAF) based on the AdaLAM algorithm and the KANN-DBSCAN clustering algorithm to filter wrong keypoint matches adaptively are proposed, according to the distinct distribution of keypoints in each image. Finally, the forgery regions are presented by finding their convex hulls and padding them. Compared with existing methods, extensive experiments show that the SMDAF method significantly provides more efficiency in detecting images under simulated real-world conditions, has better robustness when facing images with different post-treatment attacks, and is more effective in distinguishing images that look copy–move forged but are real.

1 | INTRODUCTION

The expanding and flourishing of IoT (Internet of things) has brought us into the era of big data. It is no doubt that different kinds of multimedia data are produced, transferred, and modified each day. People are not unusual to observe that powerful image editing tools can easily manipulate a certain image [1, 2] so as for beautification and entertainment. However, this may bring potentially severe consequences for misleading personal judgment, imposing negative impacts on society and disturbing digital forensics [3, 4]. Therefore, image forgeries [5–7], including splicing, retouching, and copy–move, has drawn broad concern in several essential image application fields.

Compared with other image forgeries, copy–move forgery is one of the most common manners, which copy certain areas and paste them into other parts in the same image, as is shown in Figure 1. Since the clone region with any shape can be located at any region, it is infeasible to search for all possible forgery parts. In addition, it is uneasy to detect by looking for feature differences between forged and other areas because the copy–pasted area comes from the same image, where the features (e.g. colour and noise) are compatible. Hence, there are still remaining challenges to perform copy–move forgery detection [8–10] (CMFD). In general, there are two goals for CMFD:

(1) binary classification: to judge whether an image is forged or not.

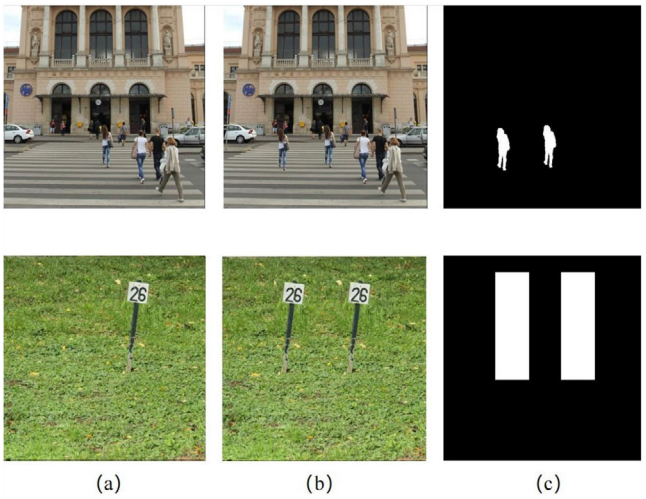

FIGURE 1 From left to right: original image (a), copy–move forged image (b), and binary groundtruth overlying copy–move area (c)

(2) forgery locating: to precisely label or locate elaborate designed copy–move parts from the whole image.

In the past few years, many copy–move forgery detection methods have been proposed, which can be roughly divided into block-based methods [11–15], deep learning-based methods [16–21], and feature point-based methods [22–30]. The block-based algorithms are inefficient and do not perform well in detecting the copy–move regions with different geometric and post-processing attack operations. Although deep learning-based methods have gained state-of-the-art success in many other fields, one interesting fact here is that these methods need many training data, and they cannot achieve good performance in CMFD. Compared with block-based and deep learning-based methods, the keypoint-based methods have achieved relatively superior performance concerning detection consequences and running speed [29, 31]. However, in the existing keypoint-based methods, the descriptors extracted from keypoints may not all meet the needs. The results contain a large number of false matching and reduce the matching accuracy. To overcome this weakness, researchers proposed some algorithms for filtering the wrong matching keypoint pairs [27–29]. Chen et al. [27] aggregated the feature and colour of the extracted keypoints, then used reverse-G2NN and outpoint filtering algorithm to filter the keypoints. Wang et al. [28] used candidate clustering based on offset information to remove wrong matching pairs and used RANSAC and ZNCC to locate duplicate regions. And Yang et al. [29] proposed a two-stage(mesh-based and cluster-based) filter to select most of the keypoint pairs of wrong matching and then applied the Delaunay triangulation algorithm for image matting.

The results are still not satisfying because they often use a general model to filter different keypoint sets. However, these methods are lack of adaptability because the keypoint set extracted from each image has its distinctive distribution, which leads to the poor effect when filtering some special sets. To solve this problem, different from the works mentioned above, we propose a novel copy–move forgery detection method called SMDAF, based on second-keypoint matching (SM) and double adaptive filtering (DAF).

The main contributions of this paper are as follows:

- This paper proposes a novel keypoint-based method for CMFD called SMDAF. It is more adaptable to filter false keypoint matching and the locating of the forgery areas is more accurate.
- The proposed second-keypoint matching method (SM) is used to obtain more keypoint matching, which can be used for both single and multiple CMFD problems.
- The proposed double adaptive filter (DAF) shows effective adaptability in filtering false keypoint matching and clustering internal keypoints. It also makes the location of the forgery areas more accurate.

Extensive experiments show that the SMDAF method is significantly more efficient in detecting images under simulated real-world conditions, has better robustness against images with different post-processing attacks, and is more effective in discriminating images that look copy–move forged but are real.

The rest of this article is organised as follows: The related work is given in Section 2, the proposed method is explained in Section 3, the experiments shown in Section 4, and the whole paper is concluded in Section 5.

2 | RELATED WORK

Copy–move forgery detection methods can be roughly divided into block-based methods, deep learning-based methods and keypoint-based methods. In this section, we introduce related researches in these areas.

2.1 | Block-based methods

Block-based methods divide the image into overlapping blocks and map them based on their features. Different methods use different transformations to represent the features of blocks. In ref. [11], Ouyang et al. proposed a method based on the pyramid model and Zernike moments. In ref. [12], Mahmood et al. combined DWT and Hu Moments together. Warif et al. proposed CMF-iteMS that extracted feature by using polar cosine transform (PCT) [13]. Li et al. used polar sine transform (PST) to tramsform features of blocks. And Emam et al. used the polar complex exponential transform (PCET) [15] kernels to represent each block. However, high computational complexity is the common defect of the above methods, and it is not suitable for solving large-scale rotation scaling and typical image processing operations like noise.

2.2 | Deep learning-based methods

In recent years, some deep learning-based methods have been applied to solve copy–move forgery detection problems

and proved to have great potential based on their excellent performance. Wu et al. [17] proposed an end-to-end deep neural network to detect copy–move forgery. In ref. [16], they improved their work as BusterNet with dual-branch architecture. Then, in ref. [19], Zhong et al. proposed the end-to-end Dense-InceptionNet based on the current popular convolutional neural network. In ref. [18], Zhu et al. developed AR-Net based on adaptive attention and residual refinement. Different from the above methods that are concentrate on CMFD, Abhishek et al. [20] used a deep convolutional neural network and semantic segmentation to detect copy–move and splicing forged images, and Yohanna et al. [21] analysed two deep learning methods, explored the detecting effect of using different neural network structures. However, these deep learning-based methods generally require a fixed image size and a vast number of training and test data. Therefore, the detection results of deep learning-based methods entirely depend on the training data's quality. They failed to detect copy–move forgery images when faced with unpredictable characteristics generated in a new way. In short, the existing methods based on deep learning are still not well perform for copy–move forgery detection.

2.3 | Keypoint-based methods

Some researchers use the local invariance technology for image forensics, such as scale-invariant feature transform (SIFT) [22–25, 27, 32], speeded-up robust features (SURF) [26, 33] and oriented fast and rotated brief (ORB) [34, 35]. Amerini et al. [22] and Pan et al.[23] proposed methods based on SIFT for image copy–move forgery detection and achieved a satisfying solution. Dhivya s et al. [26] used SURF to extract features and then trained the SVM for classification. Moreover, Tian et al. [35] used ORB to extract features that can significantly lessen detection time. For improving the number of keypoints, Yang et al. [24] proposed an algorithm based on SIFT that is robust to image transform to extract keypoints from images. In ref. [25], Li et al. discovered that changing the image contrast and scaling the image can increase the number of SIFT feature points, based on this, they optimised the performance of small or smooth copy–move areas in their method. However, due to the limitation of descriptors extracted from keypoints, they often return false matches, which affects the detecting results.

To improve the effects of CMFD, we propose a novel method called SMDAF, which can filter false matches effectively and locate the forgery areas correctly. Experimental results show that SMDAF has competitive results compared with the existing methods.

3 | PROPOSED METHOD

In this section, we first briefly introduce the related concepts and fundamental knowledge for a better understanding of our later analysis. Then we present a step by step description of our method. As shown in Figure 2, the method can be divided into the following four steps: keypoint extraction, second-keypoint matching, double adaptive filtering, and forgery locating.

3.1 | Preliminaries

3.1.1 | AdaLAM outpoint filter

As a fast and accurate outlier filter, AdaLAM is mainly used in image matching [36] based on keypoint. The methods in keypoint-based image matching often form the nearest neighbour matching set to draw correspondences between images, as is shown in Figure 3(a). The main steps of AdaLAM algorithm can be summarised as follows:

(1) It takes a large number of matched keypoints as input, then selects a limited number of seed points that are confident and well-distributed based on neighbouring compatible correspondences in this algorithm.
(2) By running highly parallel RANSAC with sample-adaptive inlier thresholds, it verifies local affine consistency in neighbourhoods of each seed point.
(3) It outputs the union of all the inliers of the seed points with strong enough support within each one's specific inlier threshold.

Similar to image matching, copy–move forgery detection based on keypoint correspond matchings in one image, as is shown in Figure 3(b). Motivated by this, our proposed method uses the AdaLAM algorithm for first outpoint filtering.

3.1.2 | KANN-DBSCAN

The KANN-DBSCAN algorithm is based on DBSCAN [37], a density-based clustering method that is sensitive to Eps and MinPts. Each parameter can be explained in Figure 4.

To adaptively find the optimum parameter pair (Eps, MinPts) that can be set for getting a better clustering effect, the KANN-DBSCAN algorithm is proposed. It first generates the candidate Eps list by using the proposed K-ANN based on the average nearest neighbour algorithm [38, 39]. As each Eps corresponding many MinPts, the algorithm calculates the mathematical expectation of them to get the MinPts list. Then it combines the Eps list and the Minpts list to a parameter pair list, successively sets them in pairs as the parameters of DBSCAN, and observe the number of clusters. Once the number remains the same after setting three consecutive parameter pairs, the algorithm records it as the best called N. After that, it continues DBSCAN clustering until the cluster number changed, and the last pair of parameters (Eps_{best}, $MinPts_{best}$) setted before changing is the optimal.

3.2 | Keypoint extraction

We use SIFT to extract the keypoints. Then, to provide enough basic keypoints, as Li [25] mentioned, we extract and combine the keypoint sets (KP_o and KP_z) from the original image and the scaled image separately, to obtain the union set called KP.

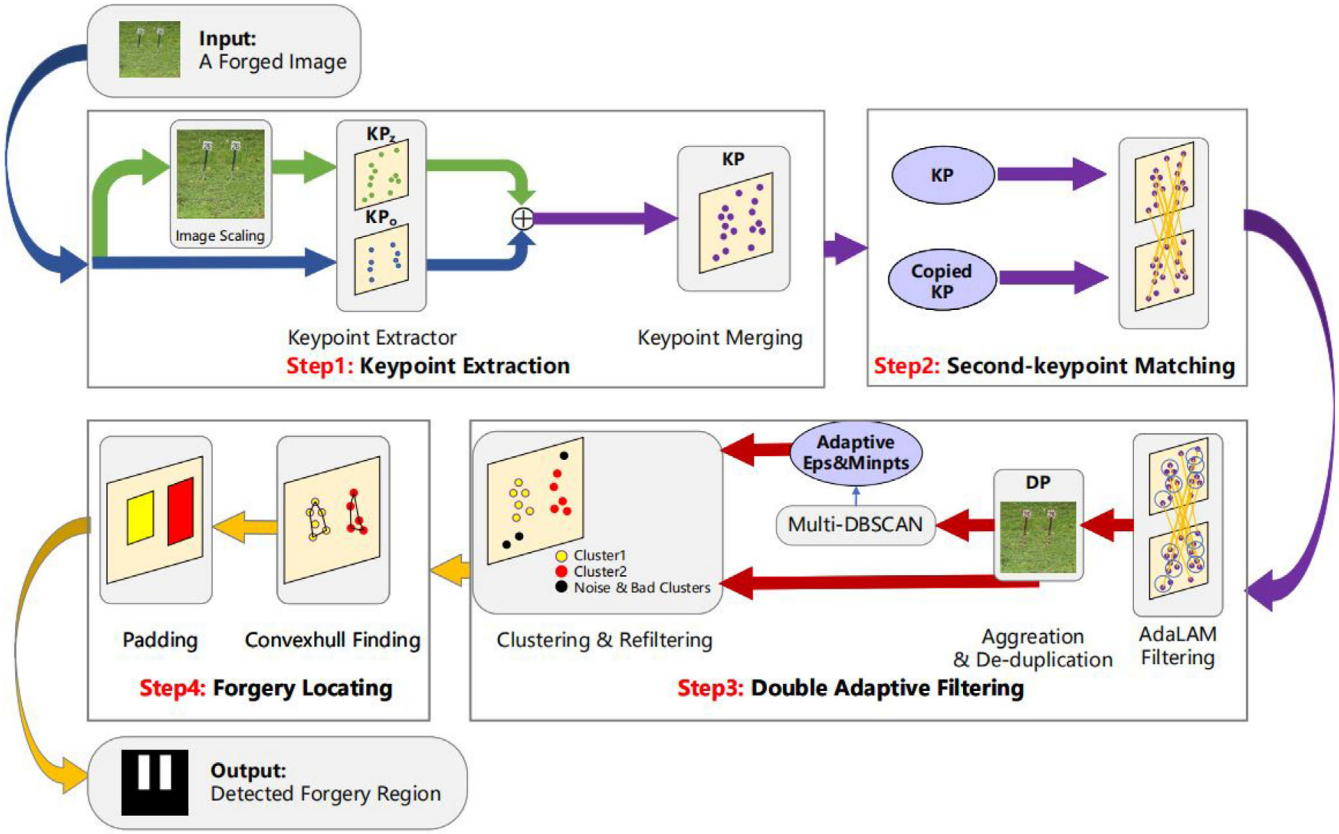

FIGURE 2 Framework of SMDAF. In the first step, the SIFT algorithm is used to extract keypoints for a given original image. To gain more keypoints, we extract them from both the original and the magnified for each image. In the second step, we copy the keypoint set containing descriptors extracted in step 1 and input them into the proposed second-keypoint matching method. As shown in step 3, we use AdaLAM as the first outpoint filter; after aggregating and de-duplicating the output, we get filtered discrete points (DP). For the unique filtered set generated from each image, we use the modified KANN-DBSCAN algorithm called Multi-DBSCAN to find the most suitable hyperparameters(i.e. Eps and Minpts) and execute DBSCAN clustering. After that, we filter out the clusters whose results are labelled as noise or the clusters with too few points. In step 4, we finally got the mask, which covered the copy–move part, by finding the convex hulls of remained clusters and padding them

(a) (b)

FIGURE 3 Typical examples of image matching (a) and copy–move forgery detection (b)

Finally, for each keypoint P_j in the keypoint set KP, where $j \in \{1, 2..., s\}$, we extract the local binary pattern features, which weight are rotation invariance and uniformity, to form the 132-dimensional keypoint feature descriptors.

The details are shown in Formula (1) and Formula (2), where S is the number of keypoints, and j $\in$ (0, S):

$$\mathrm{KP} = \{P_1, P_2, P_3, \ldots, P_S\} \tag{1}$$

$$P_j = \{PT_j, A_j, S_j, D_j\} \tag{2}$$

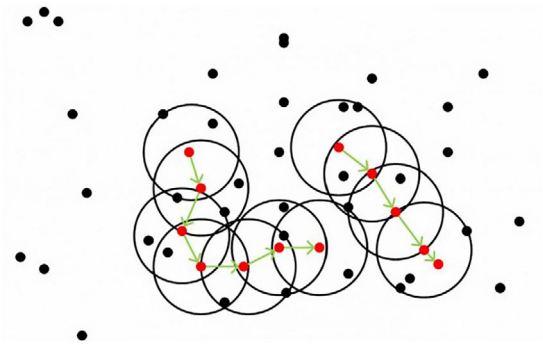

FIGURE 4 Explain of parameters in DBSCAN clustering. The circle represents ϵ-neighbourhood, which is limited by its radius Eps. The red dot ensures that the points in the circle exceed MinPts, the minimum number of points that can be chosen as a seed in the ϵ-neighbourhood. These dots are connected by green lines, meaning they are in the same point set

In Formula (2), PT_j is a 1×2 matrix, indicating 2D coordinates. A_j is a 1×1 matrix representing the gradient direction, which is calculated by SIFT around the neighbourhood of keypoints. S_j is a 1×1 matrix means the importance of the

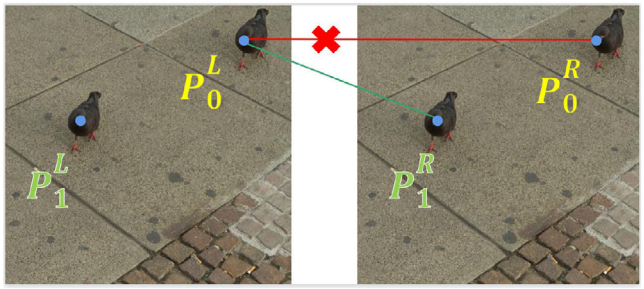

FIGURE 5 Keypoint matching for single-CMFD

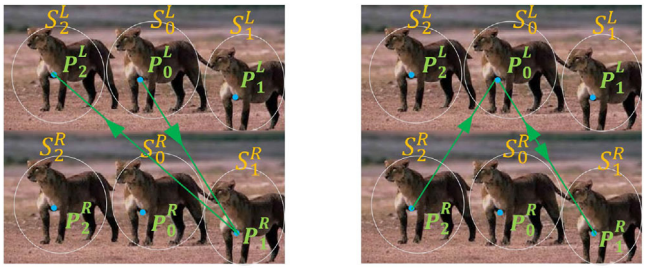

FIGURE 6 Keypoint matching for multiple-CMFD

keypoint. Finally, D_j is a 1×128 eigenvector calculated by SIFT.

3.3 | Second-keypoint matching

After the previous step, the image features are extracted as keypoints. Inspired by image matching based on keypoint, we proposed the second-keypoint matching method. It matches two keypoint sets from identical images: for each keypoint in the first set, we match the second similar keypoint corresponding to another set.

To make the matching method suitable for CMFD, which is divided into the single-CMFD and the multiple-CMFD, we analysed the distribution of keypoints and found that regions with dense keypoints often cover the copy–move areas regardless of single or multi copy–move forgery. The reason can be explained like this: As the copy–pasted regions must be more similar than other unforged regions if we ignore distinguishing them, the feature descriptors of the keypoints extracted from these regions must be very close. As a result, for a keypoint extracted from a random copy–move region, we can always find another keypoint closest to the current in other copy–move areas. This "another keypoint" is embodied as the second similar in the proposed method.

For single-CMFD, as shown in Figure 5, comparing the similarity by feature descriptors, the point P_0^{R} in the right picture is the most similar to the current point P_0^{L} in the left picture, as shown by the red line. However, it is not suitable for CMFD because the descriptors of P_0^{L} and P_0^{R} must be the same. Since the points extracted by SIFT are not sensitive to various post-processing, that is to say, the descriptors corresponding to the point after post-processing transformation are very close to the original point's descriptors. Hence, the second similar descriptor mentioned above has a high probability of being the copy-moving part's corresponding point. In our work, we match such a second similar keypoint corresponding to the current point from the right picture, and the matching results are P_0^{L} and P_1^{R}.

Things are different for multiple-CMFD, the most common situation in the CMFD task. It takes multiple forgeries from the same source region. As is shown in Figure 6, we use S_x^y to represent copy–move areas, where the subscript x can be 0, 1 and 2 to represent a certain area separately, and the superscript L or R denotes the left image or right image. A keypoint P is selected in each S_x^y, which superscript and subscript representations are the same as S_x^y. Unlike the single one, there are two following situations for matching keypoints in multi-copy–move forgery:

(a) $P_0^{\mathrm{L}} \rightarrow P_1^{\mathrm{R}}$: P_0^{L} match P_1^{R} as the second similar keypoint, but not vice versa.
(b) $P_0^{\mathrm{L}} \leftrightarrow P_1^{\mathrm{R}}$: The two keypoints match with each other.

To detect multiple-CMFD, in our work, we regard both (a) and (b) as matching and record all corresponding points.

3.4 | Double adaptive filtering

Unlike the other CMFD works, we use AdaLAM for first adaptive outpoint filtering, separating each copy–move region by clustering discrete points and refilter by deleting unsatisfied clusters.

3.4.1 | First adaptive filtering

For the first adaptive outpoints filtering, as shown in Figure 7, we first take a comprehensive set of hypothesis second matches from the two similar keypoint sets as input, and each match is represented as the yellow line in Figure 7(a). Then select the reasonable propagation hypothesis corresponding to the rough areas displayed as blue circles in Figure 7(b). As shown in Figure 7(c), for the set of all hypothesis matches, each considered region is consistent with the corresponding hypothesis of the same region. Moreover, we only retain the locally consistent correspondence with sufficient support for affine transformation, and the results can be seen in Figure 7(d).

As the method outputs two discrete point sets of coordinates, we overlap the two images to unify the coordinate values. Shown in Figure 8, the blue dots represent the points in copy–move regions, and the red points indicate the outliers to be filtered. After de-duplicate, we finally get a set of discrete points.

3.4.2 | Adaptive clustering

The output of the step 2 is a set of discrete points called DP. As is shown in Figure 9, the red dots indicates DP. There is

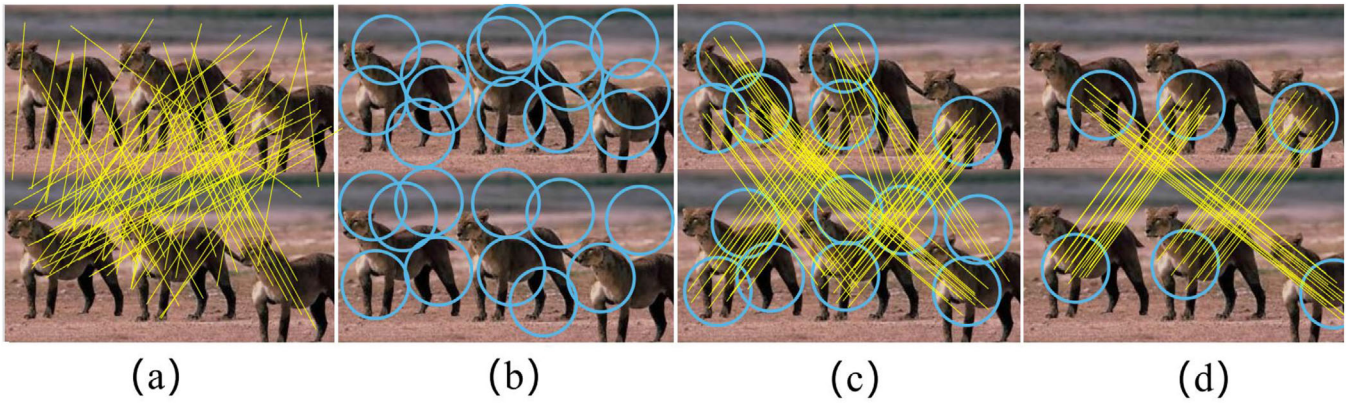

FIGURE 7 The procession of first adaptive outpoints filtering

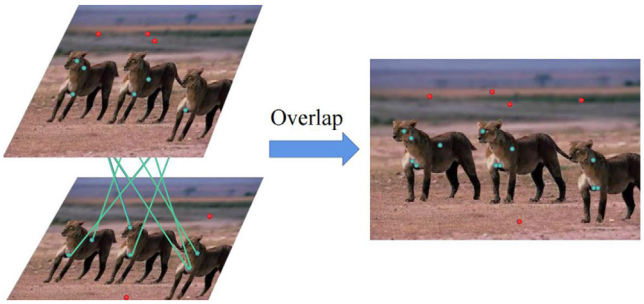

FIGURE 8 Overlap after first filtering

FIGURE 9 The image marked with DP

an interesting phenomenon discovered that these points in DP are concentrated, and they exactly cover the whole copy–move region of an image. Based on this, we intend to separate each copy–move region by clustering discrete points. An easy way to realise this is by K-means [40]. However, the K-means needs to specify the number of clustering K, as the number of image copy–move regions in CMFD cannot be fixed entirely. Inspired by the hotspot location mining based on given coordinate's density in the data mining field, we use DBSCAN to cluster discrete points. This algorithm can divide the current discrete points into flexible standard clusters (marked 1,2,3...) and a noise cluster (marked −1).

Since each image has its most suitable clustering pattern, referring to the KANN-DBSCAN algorithm that is introduced in Section 3.1.2, we modify the part that generates the candidate Eps and Minpts lists, develop Multi-DBSCAN to find the hyperparameter pair list called H, which can be written as

$$H = \{(\mathrm{Eps}_1, \mathrm{MinPts}_1), (\mathrm{Eps}_2, \mathrm{MinPts}_2) \ldots \ldots (\mathrm{Eps}_L, \mathrm{MinPts}_L)\} \quad (3)$$

For the reason that the algorithm needs to give the range of cluster numbers, we set the range to be 2–10 after observing various datasets. Then, we find all hyperparametric cases as candidates that can achieve 2–10 clustering, arranged from small to large. Next, we extract the candidate Eps_{ada} items and MinPts_{ada} items from the H and average them to obtain the adaptive hyperparameters. Mathematically,

$$\mathrm{Eps}_{ada} = \frac{\sum_{n=1}^{L} I_{\mathrm{Eps}}^{n}}{L} \quad (4)$$

$$\mathrm{MinPts}_{ada} = \frac{\sum_{n=1}^{L} I_{\mathrm{MinPts}}^{n}}{L} \quad (5)$$

It is straightforward but practical to specify the unique hyperparameter pair and the most likely cluster number during average because the average result is bound to be biased towards a cluster with the best parameter pairs in H. Finally, we set the obtained parameters for DBSCAN clustering.

3.4.3 | Refiltering

In this subsection, we find the convex hulls [41] of clusters in Section 3.4.2 to help us refine the results. Take Figure 10 for an example: the red polygon is called the convex hull, when it wraps all the points and ensures it is convex.

All know that the clusters should be filtered if they cannot satisfy the criteria, so it is needed to re-filter these clusters in Section 3.4.2. There are two following situations: One is the cluster composed of less than three points, and the other is that the whole points in the cluster are collinear. In addition, we also

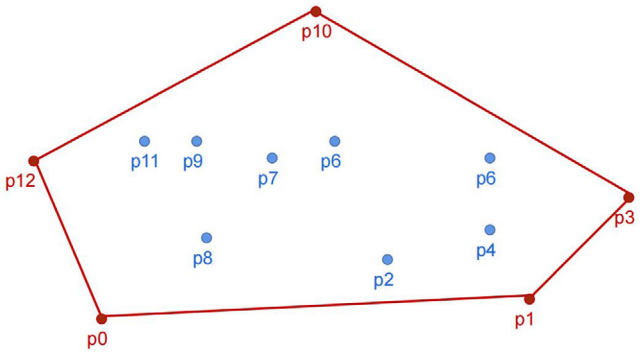

FIGURE 10 The illustration of convex hull

need to filter the lousy cluster results treated as noises, which usually contain the outpoints that were not filtered out.

Based on these preliminaries above, we find the convex hulls and refiltering the unsatisfied. Figure 11 shows an experiment comparing the effects of RANSAC, first adaptive filtering, and double adaptive filtering. As shown in Figure 11, after applying RANSAC, noticeably, too few keypoints are left (column (b)).The first adaptive filtering procedure removes most false-matching pairs but remains some wrong keypoints (column (c)).We do not care about matching pairs during refiltering but the distribution of keypoints. We gather all the points together, cluster them, and then filter the unsatisfied clusters. Finally, the proposed filter removes all false keypoints in this experiment (column (d)).

3.5 | Forgery locating

To locate the copy–move forged areas properly, we directly fill the convex hull areas to cover the copy–move areas. The results are illustrated in Figure 13, Figures 15 and 16. These figures display the predicted area's superposition and masks at the pixel level. The meaning of colours are as follows: background (blue areas), correctly detected pixels (green areas), undetected pixels (red areas), and falsely detected pixels (yellow areas). The reason is that the convex hull of the point set will also wholly draw out the edge area of the current point set, which can cover the copy–move forged area.

4 | EXPERIMENT

4.1 | Setup

4.1.1 | Datasets

In order to verify the competitive performance of the method, three benchmark datasets are used in this experiment: CASIA-CMFD dataset [42], MICC-F220 dataset [43], CoMoFoD dataset [44], and Coverage dataset [45]. Table 1 shows the detailed project information for the dataset.

1. CASIA-CMFD dataset: The dataset has 3274 copy–move forged images and 7491 authentic images, including various styles like animals, plants, and patterns. And the forged images are randomly selected and manipulated from authentic images.
2. MICC-F220 dataset: The dataset consists of 220 images: 110 are tampered images and 110 are originals. The image resolution varies from 722×480 to 800×600 pixels and the size of the forged patch covers, on the average, 1.2% of the whole image.
3. CoMoFoD dataset: The complete CoMoFoD database contains 200 small image categories (512×512 pixels) and 60 large image categories (3000×2000 pixels). Each fake image underwent six post-processing attacks, including JPEG compression, image blurring, adding noise, brightness changes, colour reduction, and contrast adjustments. In our experiment, we choose the 200 small image categories to evaluate the detection capability of the algorithm against various post-processing attacks, with 5000 images in total. Specific parameters are shown in Table 3.
4. Coverage dataset: The dataset has 100 pairs of images: 100 standard images similar to copy–move forged images and 100 forged images. Moreover, the forged picture has the groundtruth of the copy–move area. The purpose of the dataset is to highlight and solve the ambiguity caused by natural images' self-similarity in the popular forgery detection methods.

4.1.2 | Metrics

Since most state-of-the-art approaches use the same metrics, including Precision, Recall, and F_1 score, these metrics are used to objectively evaluate the CMFD algorithm's performance, which are respectively defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

Depending on the specific experiments, these indicators will be used at image-level and pixel-level, which separately measure the effect of binary classification and forgery locating. We focus on correctly identifying the image as forged or authentic on the image-level. At that time, Precision, Recall, and F_1 score in the above definition respectively represent whether the method can correctly distinguish positive and negative samples. At the pixel level, we obtain a schematic image by superimposing the predicted mask with the groundtruth image, which is used to analyse the performance of forgery locating accuracy. Unlike the image level, Precision, Recall, and F_1 score in the above definition only applies to negative samples (forged image). In addition, we averaged the Precision and Recall of each image

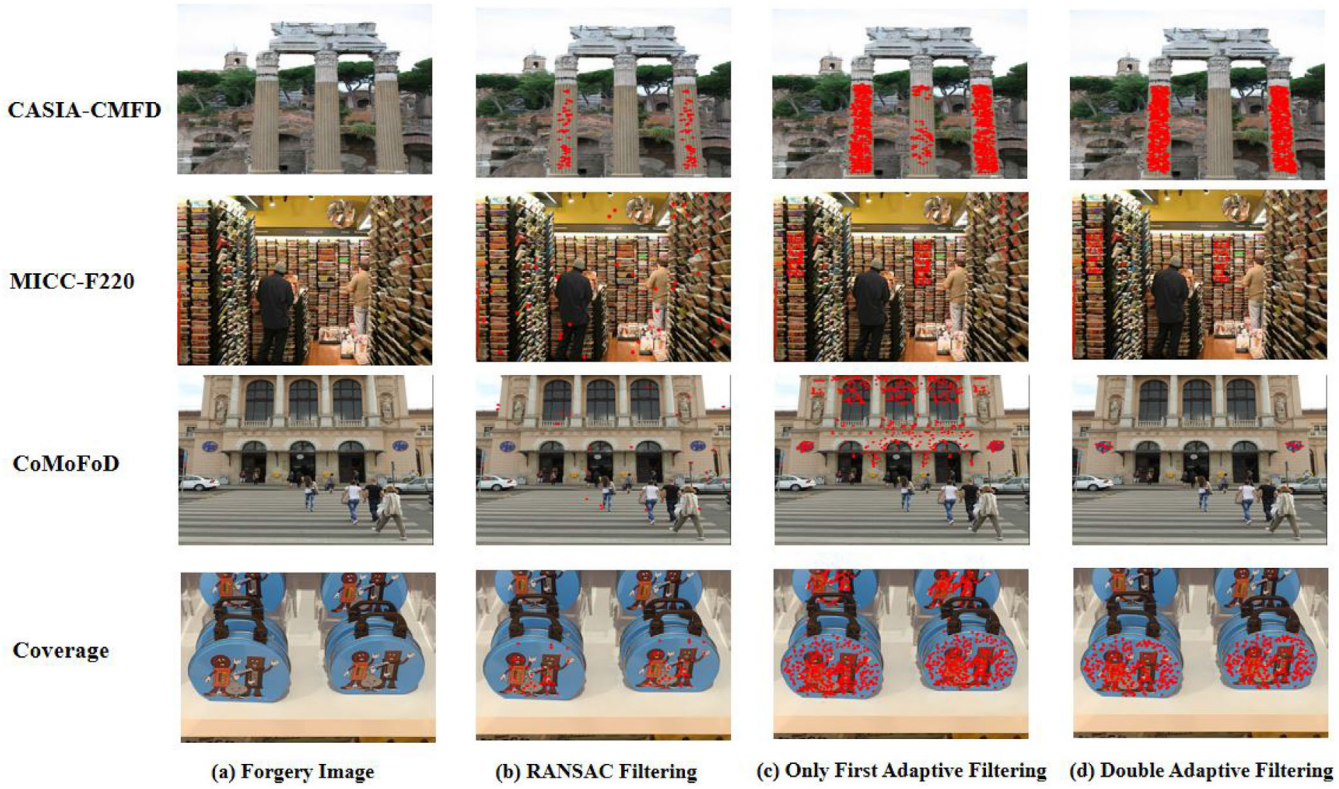

FIGURE 11 Comparing the filtering results of different filters. (a) Forgery image; (b) RANSAC filtering; (c) only first adaptive filtering; (d) double adaptive filtering

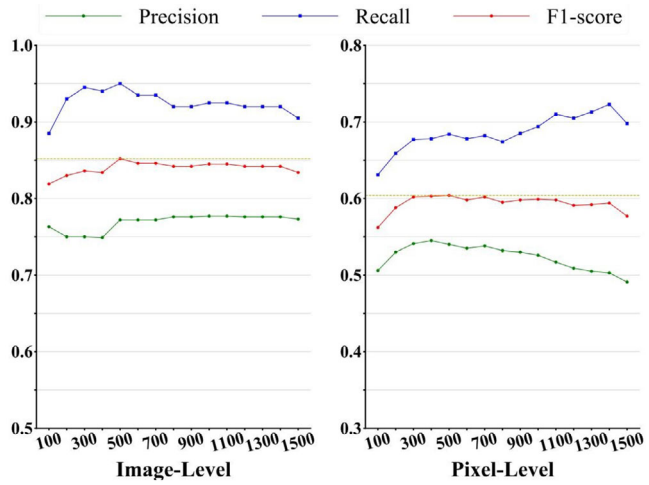

FIGURE 12 Ablation experiments using CoMoFoD dataset. Showing curves of Precision, Recall, and F_1.

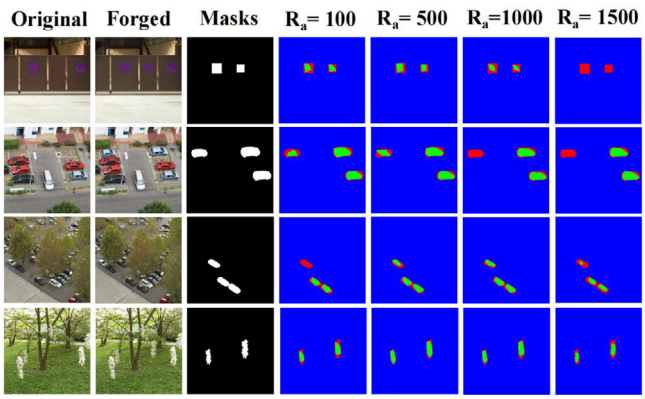

FIGURE 13 Examples from pixel-level ablation experiments

during calculation and then used the formula to calculate the final F_1 score.

4.1.3 | Settings

All experiments are conducted on a personal computer with Intel(R) Core (TM) i7-10700k 3.80 GHz CPU, 32 GB RAM and NVIDIA GTX 2070 GPU. The code is implemented with Python 3.7 and OpenCV 3.3.1.

4.2 | Ablation experiment

In order to improve the effect of first adaptive filtering in Section 3.3, we examine the influence of parameters set for AdaLAM on the CoMoFoD dataset at the image-level and pixel-level. Our experiment found that changing the area-ratio (R_a), which is responsible for the number of seeds generated during keypoint matching, has the most significant influence. Generally

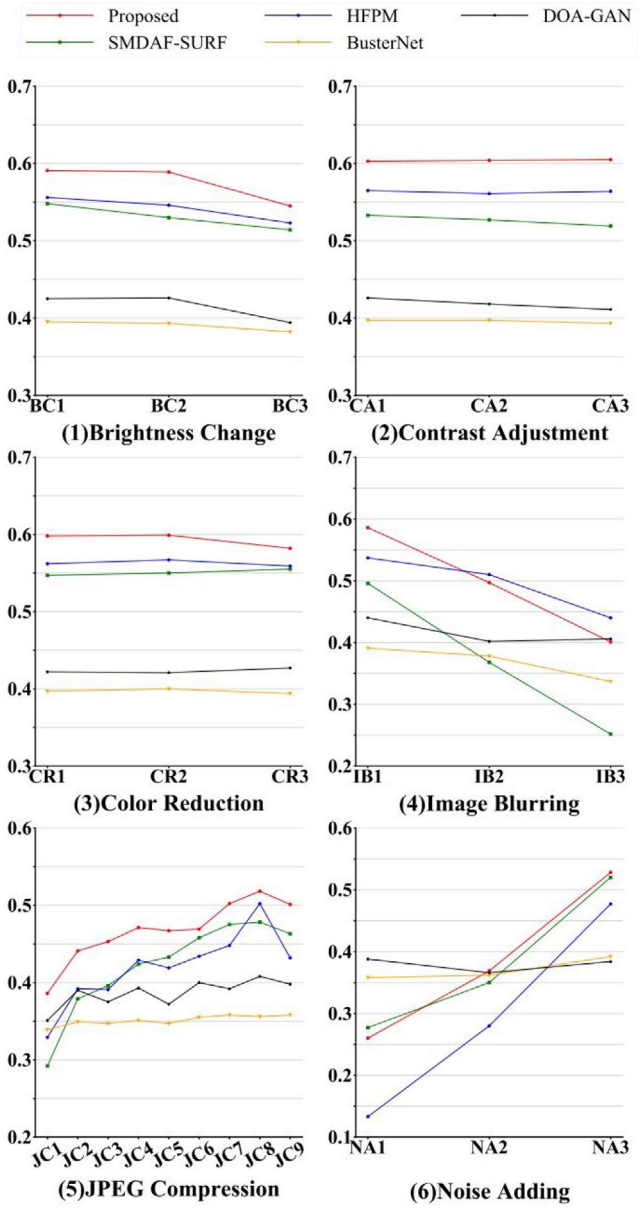

FIGURE 14 The F_1 curves using different algorithms with the CoMoFoD dataset

speaking, the higher the R_a value, the smaller the neighbourhood, the more seeds generated, and the more meticulous the matching. But the results overturned this conclusion. As can be seen in Figure 12, no matter in image-level or pixel-level, as the R_a increases from 100 to 1500, the Precision, Recall, and F_1 score all rise first and then fall. The reason can be explained that if the ratio value is too high, the generated seeds will lack representativeness, and they will be removed during filtering, thus affecting the prediction results. Then it is observed that the F_1 curve (red line) reaches the maximum when R_a is 500, the score is 0.852 at the image-level and 0.604 at the pixel-level. The results in Figure 13 also prove it, as the correctly predicted green part almost covers all the copy–movement areas. After ablation, we set 500 as the final r_a in our experiment.

In the following sections, the post-ablation method is used to compare with some existing methods containing BusterNet method (2018) [16], HFPM method (2018) [25], and DOA-GAN (2020) [46]. Moreover, we add the SMDAF-SURF method for comparing, which uses SURF as the keypoint extraction algorithm, but the other processions are the same as our proposed method.

4.3 | Simulating real-world conditions

To simulate detecting under the actual situation, which usually contains extensive data and multiple complexities, we evaluate the metrics on the CASIA-CMFD dataset and MICC-F220 dataset. We choose these datasets for the following reasons: First, the images in these datasets include various objects like animals, plants and patterns. The variety is relatively complex and closer to the real world; Then, the magnitude is relatively large compared with some other public datasets for CMFD. Due to the vital metric of detecting a relatively large dataset in the real world is its classification success rate, and the lack of groundtruth, we only focus on the image-level in the experimental part. Finally, to ensure the same cardinality of Precision, Recall, and F_1 score when calculating, we regard the image that will report program error as prediction error here.

From the results in Table 2, the BusterNet methods makes poor performance at the image-level. The HFPM method grows to 0.500 and 0.901 of the F_1 score. However, it occurs program errors when detecting 1313 authentic and 838 forgery images. Another deep learning-based method called DOA-GAN, it gets better results on CASIA-CMFD dataset, which F_1 score is 0.629. However, the F_1 score on MICC-F220 dataset is as disappointing as BusterNet, which only reaches 0.746. After using the SMDAF-SURF and the proposed method, the F_1 scores significantly improve at the image-level. On the CASIA-CMFD dataset, the result of SMDAF-SURF improves from 0.629 in DOA-GAN to 0.682, and the proposed method is even more excellent, achieving 0.714. On the MICC-F220 dataset, the F_1 score of the proposed method can reach 0.904, which is held a slender lead to HFPM (0.901) and SMDAF-SURF (0.869).

All in all, in simulating an actual situation to detect copy–move forged images, the proposed method is more practical as it has apparent advantages at the image-level.

4.4 | Different post-treatment attacks

For testing the effect of methods after post-processing, we take experiences on the CoMoFoD dataset, which contains brightness change (BC), contrast adjustments (CA), colour reduction (CR), image blurring (IB), JPEG compression (JC), and noise adding (NA).

As can be seen from Table 3, the number of detected images by using SMDAF-SURF and the proposed method dramatically exceeds that of other methods. In contrast, the proposed method outperforms the SMDAF-SURF method in most

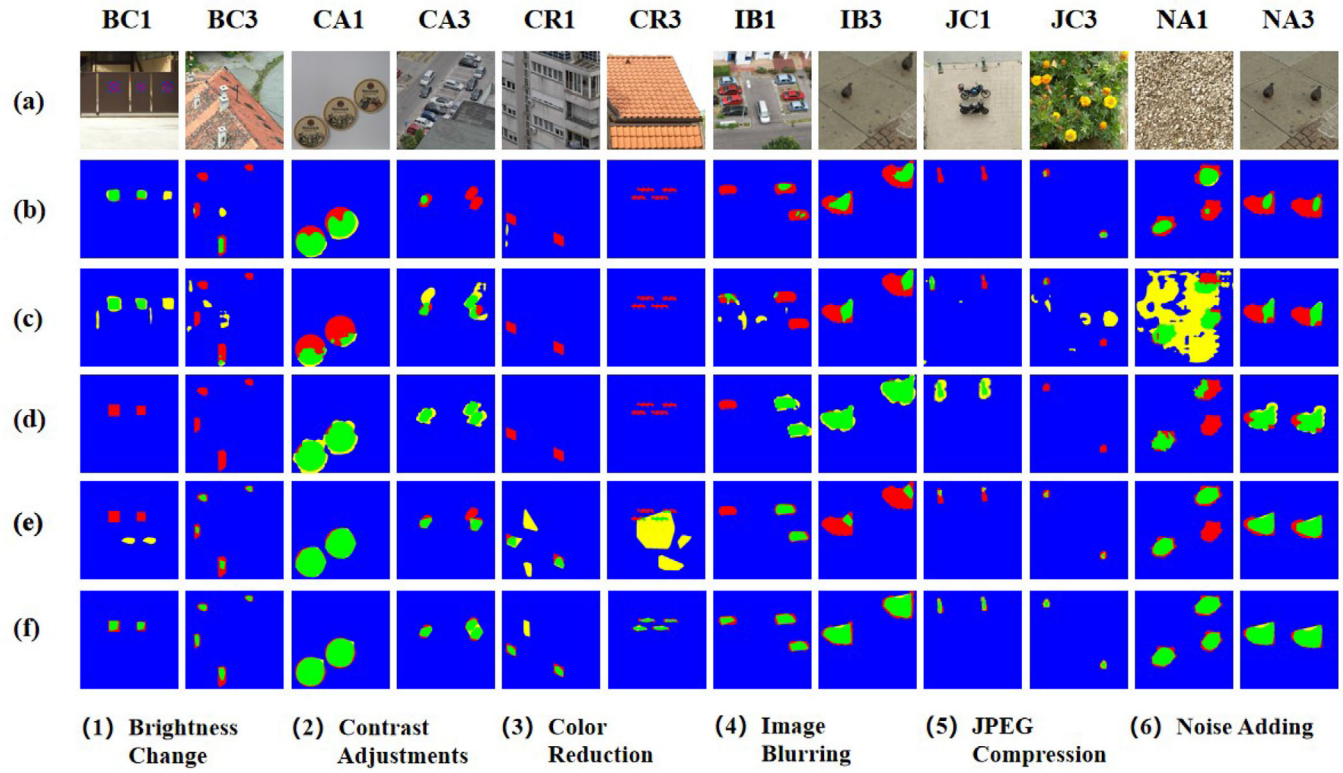

FIGURE 15 Examples of using different algorithms to predict the post-processing attack image in the CoMoFoD dataset. From top to bottom: (a) forged images. (b) BusterNet [16]. (c) DOA-GAN [46]. (d) HFPM [25]. (e) SMDAF-SURF. (f) Proposed method

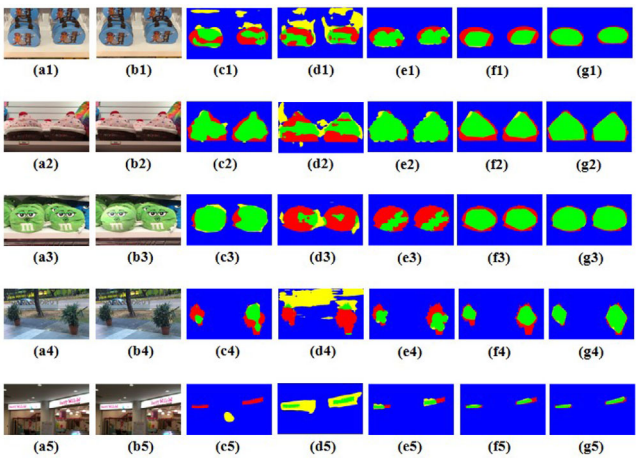

FIGURE 16 Examples obtained on the coverage dataset. From left to right: original images (a1)∼(a5), forged images (b1)∼(b5), masks output by BusterNet [16] (c1)∼(c5), by DOA-GAN method [46] (d1)∼(d5), by HFPM method [25] (e1)∼(e5), by SMDAF-SURF method (f1)∼(f5) and by our proposed method (g1)∼(g5)

TABLE 1 Datasets

Dataset	Operation	Sum
CASIA-CMFD	None	7491
	Forgery	3274
MICC-F220	None	110
	Forgery	110
CoMoFoD	None	200
	Forgery	200
	Forgery+JPEG compression	1800
	Forgery+image blurring	600
	Forgery+noise adding	600
	Forgery+brightness change	600
	Forgery+colour reduction	600
	Forgery+contrast adjustments	600
Coverage	None	100
	Forgery	100

post-processing attacks, expecting several attacks (NA1, NA2, NA3, BC3, JC6, JC9). Moreover, the proposed method achieves the highest recall rate and F_1 score at the pixel-level.

The F_1 curves at the pixel-level are shown in Figure 14, which exposes the superiority of the proposed method (red line) in the face of most post-treatment attacks. Nevertheless, it points out that for image blurring, as the size of average filter increasing, from 3×3 to 7×7, the F_1 score using keypoint-based methods decline over 15% on average (24.4% for SMDAF-SURF, 18.5% for the proposed and 9.7% for HFPM). In contrast, the BusterNet only declines 5.4% and the DOA-GAN declines 4.8%. Then, adding noises at different scales also affects the keypoint-based methods a lot. Take the proposed method as an example,

TABLE 2 Using different algorithms for Image detection results of CASIA-CMFD dataset and MICC-F220 dataset(The results are divided into this: CASIA-CMFD/MICC-F220)

	Image-level			
Methods	**Detected images**	***P***	***R***	F_1
BusterNet[16]	**10,765/220**	0.554/0.664	0.453/0.863	0.498/0.751
HFPM[25]	8614/**220**	0.529/0.853	0.474/**0.954**	0.500/0.901
DOA-GAN[46]	**10,765/220**	0.585/0.679	0.680/0.863	0.629/0.746
SMDAF-SURF	**10,765/220**	0.775/0.811	0.609/0.936	0.682/0.869
Proposed	**10,765/220**	**0.807/0.867**	**0.640**/0.945	**0.714/0.904**

the F_1 score increases rapidly as the noise addition reduces, from 0.26 to 0.528, while the deep learning-based methods are stable. BusterNet is from 0.358 to 0.392, and the DOA-GAN is from 0.388 to 0.384.

From the examples in Figure 15, after using the deep learning-based methods (BusterNet and DOA-GAN), there are many undetected pixels (red areas) and falsely detected pixels (yellow areas) in the pictures, meaning this method lacks at the pixel-level. For the keypoint based methods, like HFPM, several pictures display only blue and red, which indicates that the method failed to detect the forged images at the image-level, and there are many falsely detected pixels (yellow areas) in the pictures meaning over-locating, which declines the F_1 score performance at the pixel-level. As for the SMDAF-SURF method, the performance is close to the proposed method but still unsatisfying.

In general, the proposed method is more robust than others in the face of post-processing attacks. Although the

TABLE 3 Specific parameters of CoMoFoD dataset and the number of correctly detected images using different algorithms

					Methods				
Operations		**Parameters**	**Flags**	**Num**	**Buster Net [16]**	**HFPM [25]**	**DOA-GAN [46]**	**SMDAF-SURF**	**Proposed**
Brightness change		Brightness ranges (0.01, 0.95)	BC1	200	135	145	158	186	**188**
		Brightness ranges = (0.01, 0.9)	BC2	200	135	142	151	181	**186**
		Brightness ranges = (0.01, 0.8)	BC3	200	132	135	132	**178**	176
Contrast adjustments		Adjustment ranges = (0.01, 0.95)	CA1	200	136	152	163	185	**192**
		Adjustment ranges = (0.01, 0.9)	CA2	200	137	148	169	185	**192**
		Adjustment ranges = (0.01, 0.8)	CA3	200	135	148	169	183	**189**
Colour reduction		Intensity levels per each colour channel = 32	CR1	200	135	148	151	184	**189**
		Intensity levels per each colour channel = 64	CR2	200	139	148	151	186	**189**
		Intensity levels per each colour channel = 128	CR3	200	134	148	146	**187**	**187**
Image blurring		Average filter = 3 × 3	IB1	200	137	142	173	176	**192**
		Average filter = 5 × 5	IB2	200	133	138	125	158	**179**
		Average filter = 7 × 7	IB3	200	123	116	128	128	**162**
JPEG compression		Quality factor = 20	JC1	200	144	106	118	148	**159**
		Quality factor = 30	JC2	200	139	114	132	162	**171**
		Quality factor = 40	JC3	200	130	110	121	**167**	**167**
		Quality factor = 50	JC4	200	130	120	121	**177**	172
		Quality factor = 60	JC5	200	132	117	118	**176**	170
		Quality factor = 70	JC6	200	136	122	115	**179**	169
		Quality factor = 80	JC7	200	131	129	129	181	**183**
		Quality factor = 90	JC8	200	137	143	132	183	**187**
		Quality factor = 100	JC9	200	133	122	126	**183**	182
Noise adding		Mean value $\mu = 0$, variance $\sigma^2 = 0.009$	NA1	200	139	96	**154**	132	121
		Mean value $\mu = 0$, variance $\sigma^2 = 0.005$	NA2	200	133	78	140	**148**	143
		Mean value $\mu = 0$, variance $\sigma^2 = 0.0005$	NA3	200	137	124	131	**162**	173
None		-	F	200	136	147	155	184	**190**
Sum		-	-	5000	3368	3181	3508	4319	**4418**
Pixel-level	Precision	-	-	-	0.302	**0.571**	0.415	0.420	0.468
	Recall	-	-	-	0.494	0.398	0.393	0.510	**0.572**
	F_1-score	-	-	-	0.373	0.468	0.404	0.457	**0.511**

TABLE 4 Results of coverage dataset

Methods	Detected images	Image-level			Pixel-level		
		P	*R*	F_1	*P*	*R*	F_1
BusterNet[16]	**200**	0.508	**0.940**	0.660	**0.624**	0.652	0.638
HFPM[25]	194	**0.635**	0.750	0.688	0.595	0.595	0.595
DOA-GAN[46]	**200**	0.509	0.810	0.625	0.591	0.530	0.559
SMDAF-SURF	**200**	0.568	0.870	0.687	0.490	0.684	0.571
Proposed	**200**	0.583	0.910	**0.711**	0.606	**0.768**	**0.678**

keypoint-based methods are not as stable enough as the deep learning-based methods when facing image blurring and noise adding, the proposed method is superior to other existing keypoint-based methods.

4.5 | Distinguishing the similar but authentic images

To measure whether the algorithms can correctly detect authentic images similar to a copy–move image and accurately point out the forged part, we take experiences on the Coverage dataset containing copy–move forged images and their originals with similar but genuine objects.

It can be seen from Figure 16 that the effect in pixel-level using different methods on the Coverage dataset is similar to the effect on the CoMoFoD dataset. It shows the superiority of the proposed method on different datasets. From Table 4, the BusterNet method achieves the highest recall rate and accuracy rate at the image level, but there are problems in the detection accuracy rate at the image level. The accuracy at the image level of 0.508 indicates that this method cannot distinguish forged images from raw images that are very similar to copy–move. The HFPM method has flaws in detecting images, which can only detect 97 pairs out of 100 pairs, and program errors may occur when detecting No.9, No.13, and No.36 image pairs. When comparing with other methods on image-level and pixel-level, it is needed to keep up with the number of detected images in total by different methods. Thus the three image pairs are processed as undetected when calculating the Precision, Recall, and F_1 score. Moreover, although the HFPM method has the highest value in image-level accuracy, the proposed method is superior to the HFPM method in other aspects. After using the DOA-GAN, the result at the image level is similar to BusterNet but gets worse at the pixel level. And the performance of SMDAF-SURF are inferior to the proposed method in all metrics.

In conclusion, the proposed method can better distinguish confusing images at image-level and pixel-level after taking experiments.

5 | CONCLUSION

The paper proposes a novel keypoint-based method for CMFD called SMDAF, consisting of (SM) and (DAF), which adaptively filter false keypoint matches according to the distinctive distribution of each image. The proposed second-keypoint matching method (SM) is used to match more SIFT keypoints and can solve both the single and multiple CMFD problems. The proposed DAF can filter the false keypoint matches more adaptively and locate the forgery areas more precisely than existing filtering methods. Compared with three SOTA methods for CMFD on four different benchmark datasets, the SMDAF method significantly provides competitive results. First, the proposed method is more practical as it reaches the highest F1 scores for binary classification on the CASIA-CMFD and MICC-F220 datasets. Second, in the experiments on the CoMoFoD dataset, although the deep learning-based method is more stable when facing post-processing attacks, the proposed method has better performance on the entire dataset, and the F1-score at the pixel level achieves 0.511, while the SOTA methods cannot exceed 0.5. Besides, it can also better distinguish authentic images similar to copy–move images on the Coverage dataset. Extensive experiments show that this approach has the advantages of filtering false keypoint matches more adeptly and locating the forgery areas more precisely.

In future work, we will make improvements in the following two aspects. One is to study the extraction of keypoints, and the other is to design better forgery localisation algorithms to locate the copy–paste regions precisely. In addition, the stability of deep learning-based methods against post-processing attacks has attracted our attention. We will also focus on the combination of keypoint-based and deep-learning methods.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 62162067, 62101480, 61762089, 61763048, in part by the Yunnan Province Science Foundation for Youths under Grant No.202005AC160007, in part by the Open Foundation of Key Laboratory in Software Engineering of Yunnan Province under Grant 2020SE307, and in part by the Scientific Research Fund of Yunnan Provincial Education Department under Grant 2021J0007.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analysed in this study.

ORCID

Guangyu Yue https://orcid.org/0000-0003-4663-962X
Renyang Liu https://orcid.org/0000-0002-7121-1257
Junhui Liu https://orcid.org/0000-0003-3230-5653

REFERENCES

1. Verdoliva, L.: Media forensics and deepfakes: an overview. IEEE J. Sel. Top. Signal Process. 14(5), 910–932 (2020)

2. Pasquini, C., Amerini, I., Boato, G.: Media forensics on social media platforms: A survey. EURASIP J. Inf. Secur. 2021(1), 1–19 (2021)
3. Hajialilu, S.F., Azghani, M., Kazemi, N.: Image copy-move forgery detection using sparse recovery and keypoint matching. IET Image Process. 14(12), 2799–2807 (2020)
4. Ferreira, A., Felipussi, S.C., Alfaro, C., Fonseca, P., Vargas Munoz, J.E., Dos Santos, J.A., et al.: Behavior knowledge space-based fusion for copy–move forgery detection. IEEE Trans. Image Process. 25(10), 4729–4742 (2016)
5. Bi, X., Wei, Y., Xiao, B., Li, W.: RRU-Net: the ringed residual U-Net for image splicing forgery detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 30–39. IEEE, Piscataway, NJ (2019)
6. Li, H., Luo, W., Qiu, X., Huang, J.: Image forgery localization via integrating tampering possibility maps. IEEE Trans. Inf. Forensics Secur. 12(5), 1240–1252 (2017)
7. Zhou, P., Han, X., Morariu, V.I., Davis, L.S.: Learning rich features for image manipulation detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1053–1061. IEEE, Piscataway, NJ (2018)
8. Soni, B., Das, P.K.: Image Copy-move Forgery Detection: New Tools and Techniques. Vol. 1017, Springer Nature, London (2022)
9. Dixit, R., Naskar, R.: Review, analysis and parameterisation of techniques for copy–move forgery detection in digital images. IET Image Process. 11(9), 746–759 (2017)
10. 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)
11. Ouyang, J., Liu, Y., Liao, M.: Robust copy-move forgery detection method using pyramid model and Zernike moments. Multimedia Tools Appl. 78(8), 10207–10225 (2019)
12. Mahmood, T., Nawaz, T., Shah, M., Khan, Z., Ashraf, R., Habib, H.A.: Copy-move forgery detection technique based on DWT and Hu moments. Int. J. Comput. Sci. Inf. Secur. 14(5), 156–161 (2016)
13. Warif, N.B.A., Idris, M.Y.I., Wahab, A.W.A., Salleh, R., Ismail, A.: CMF-items: an automatic threshold selection for detection of copy-move forgery. Forensic Sci. Int. 295, 83–99 (2019)
14. Li, L., Li, S., Zhu, H., Wu, X.: Detecting copy-move forgery under affine transforms for image forensics. Comput. Electr. Eng. 40(6), 1951–1962 (2014)
15. 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)
16. Wu, Y., Abd Almageed, W., Natarajan, P.: Busternet: detecting copy-move image forgery with source/target localization. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 168–184. Springer, Cham (2018)
17. Wu, Y., Abd Almageed, W., Natarajan, P.: Image copy-move forgery detection via an end-to-end deep neural network. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1907–1915. IEEE, Piscataway, NJ (2018)
18. 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. Ind. Inf. 16(10), 6714–6723 (2020)
19. 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)
20. Jindal, N., et al.: Copy move and splicing forgery detection using deep convolution neural network, and semantic segmentation. Multimedia Tools Appl. 80(3), 3571–3599 (2021)
21. Rodriguez.Ortega, Y., Ballesteros, D.M., Renza, D.: Copy-move forgery detection (CMFD) using deep learning for image and video forensics. J. Imaging 7(3), 59 (2021)
22. Amerini, I., Ballan, L., Caldelli, R., Del.Bimbo, A., 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)
23. Pan, X., Lyu, S.: Region duplication detection using image feature matching. IEEE Trans. Inf. Forensics Secur. 5(4), 857–867 (2010)
24. Yang, B., Sun, X., Guo, H., Xia, Z., Chen, X.: A copy-move forgery detection method based on CMFD-SIDT. Multimedia Tools Appl. 77(1), 837–855 (2018)
25. 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 (2018)
26. Dhivya, S., Sangeetha, J., Sudhakar, B.: Copy-move forgery detection using surf feature extraction and SVM supervised learning technique. Soft Comput. 24, 14429–14440 (2020)
27. Chen, H., Yang, X., Lyu, Y.: Copy-move forgery detection based on keypoint clustering and similar neighborhood search algorithm. IEEE Access 8, 36863–36875 (2020)
28. Wang, X.y., Wang, C., Wang, L., Yang, H.y., Niu, P.p.: Robust and effective multiple copy-move forgeries detection and localization. Pattern Anal. Appl. 24, 1025–1046 (2021)
29. Yang, J., Liang, Z., Gan, Y., Zhong, J.: A novel copy-move forgery detection algorithm via two-stage filtering. Digital Signal Process. 113, 103032 (2021)
30. Zhang, W.h., Zheng, J.w., Guo, M.x., Tao, Z.h., He, Y.k.: Tampering image detection algorithm of multi-distance feature matching. J. Xi'an Univ. Sci. Technol. (2019)
31. Gan, Y., Zhong, J., Vong, C.: A novel copy-move forgery detection algorithm via feature label matching and hierarchical segmentation filtering. Inf. Process. Manage. 59(1), 102783 (2022)
32. Ng, P.C., Henikoff, S.: Sift: Predicting amino acid changes that affect protein function. Nucleic Acids Res. 31(13), 3812–3814 (2003)
33. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European Conference on Computer Vision, pp. 404–417. Springer, Berlin, Heidelberg (2006)
34. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: an efficient alternative to SIFT or SURF. In: 2011 International Conference on Computer Vision, pp. 2564–2571. IEEE, Piscataway, NJ (2011)
35. Tian, X., Zhou, G., Xu, M.: Image copy-move forgery detection algorithm based on orb and novel similarity metric. IET Image Process. 14(10), 2092–2100 (2020)
36. Remondino, F., Spera, M.G., Nocerino, E., Menna, F., Nex, F.: State of the art in high density image matching. Photogramm. Record 29(146), 144–166 (2014)
37. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. Proc. KKD 96, 226–231 (1996)
38. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. on Intell. Syst. Technol. 2(3), 1–27 (2011)
39. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Mach. Learn. 46(1), 389–422 (2002)
40. Morissette, L., Chartier, S.: The k-means clustering technique: general considerations and implementation in mathematica. Tut. Quant. Methods Psychol. 9(1), 15–24 (2013)
41. Shin, S.Y., Woo, T.C.: Finding the convex hull of a simple polygon in linear time. Pattern Recognit. 19(6), 453–458 (1986)
42. Dong, J., Wang, W., Tan, T.: CASIA image tampering detection evaluation database. In: 2013 IEEE China Summit and International Conference on Signal and Information Processing, pp. 422–426. IEEE, Piscataway, NJ (2013)
43. Amerini, I., Ballan, L., Caldelli, R., Del.Bimbo, A., 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)
44. Tralic, D., Zupancic, I., Grgic, S., Grgic, M.: CoMoFo—new database for copy-move forgery detection. In: Proceedings ELMAR-2013, pp. 49–54. IEEE, Piscataway, NJ (2013)

45. Wen, B., Zhu, Y., Subramanian, R., Ng, T.T., Shen, X., Winkler, S.: Coverage-a novel database for copy-move forgery detection. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 161–165. IEEE, Piscataway, NJ (2016)
46. Islam, A., Long, C., Basharat, A., Hoogs, A.: DOA-GAN: dual-order attentive generative adversarial network for image copy-move forgery detection and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4676–4685. IEEE, Piscataway, NJ (2020)

How to cite this article: Yue, G., Duan, Q., Liu, R., Peng, W., Liao, Y., Liu, J.: SMDAF: A novel keypoint based method for copy-move forgery detection. IET Image Process. 16, 3589–3602 (2022). https://doi.org/10.1049/ipr2.12578