applied sciences

MDPI

Article

Structural Correlation Based Method for Image Forgery Classification and Localization

Nam Thanh Pham [1], Jong-Weon Lee [2] and Chun-Su Park [3,*]

1 Department of Digital Contents, Sejong University, Seoul 05006, Korea; pham@sju.ac.kr
2 Department of Software, Sejong University, Seoul 05006, Korea; jwlee@sejong.ac.kr
3 Department of Computer Education, Sungkyunkwan University, Seoul 03063, Korea
* Correspondence: cspk@skku.edu

Received: 12 June 2020; Accepted: 26 June 2020; Published: 28 June 2020

Abstract: In the image forgery problems, previous works has been chiefly designed considering only one of two forgery types: copy-move and splicing. In this paper, we propose a scheme to handle both copy-move and splicing image forgery by concurrently classifying the image forgery types and localizing the forged regions. The structural correlations between images are employed in the forgery clustering algorithm to assemble relevant images into clusters. Then, we search for the matching of image regions inside each cluster to classify and localize tampered images. Comprehensive experiments are conducted on three datasets (MICC-600, GRIP, and CASIA 2) to demonstrate the better performance in forgery classification and localization of the proposed method in comparison with state-of-the-art methods. Further, in copy-move localization, the source and target regions are explicitly specified.

Keywords: image forgery; copy-move; splicing; classification; localization

1. Introduction

In an era of globalization, social networks such as Facebook, Twitter, and Instagram are widely used in our daily lives and a huge number of photos are uploaded to these networks everyday. Further, it becomes easy even for unpracticed users to manipulate digital images without leaving any perceptible trace. Copy-move and image splicing are two most popular image manipulation methods. In the copy-move forgery (CMF), one or more regions are copied from an authentic image and then pasted into other regions of that image. The authentic image used to compose the copy-move image is called the host image. On the other hand, in image splicing, some regions are copied from a source image (the donor image) and pasted into a target image (the host image) [1]. Examples of copy-move and spliced images are given in Figure 1.

In the image forgery scenario, a tampered region might not be exactly the same as the original region since it usually undergoes a sequence of post-processing operations such as rotation, scaling, edge softening, blurring, denoising, and smoothing for a better visual appearance [1]. Therefore, human beings may easily be deceived by tampered images and it is difficult to manually verify the authenticity of images.

Many researchers have put considerable effort into detecting and localizing tampered regions of image forgery. However, in most cases, forgery detection and localization algorithms were designed considering only one of two forgery types, copy-move and image splicing. In this paper, we propose an image forgery detection and localization algorithm that can handle both types of image forgeries simultaneously. The proposed method utilizes the bag-of-features (BOF) image representation and Hamming Embedding (HE) based image retrieval. The image forgery clustering algorithm classifies input images into distinct clusters, each of which consists of one authentic image and all the spliced and

copy-move images which were composed using that authentic image as the host image. The algorithm also determines the authentic image based on structure and luminance similarity between images and assigns it as the centroid of the cluster. The cluster centroid is used to classify the image forgeries and localize the tampered regions. The experimental results show that the proposed method outperforms state-of-the-art techniques in image forgery classification and localization accuracy. In addition, we distinguish the source and target regions in copy-move tampering localization.

Figure 1. Examples of image forgery in the top row, the corresponding host images are in the middle row, and the groundtruth images of forged regions are in the bottom row. Copy-move images are shown in (**a**,**b**), while spliced images are shown in (**c**,**d**).

The further part of this paper is organized as follows. Section 2 provides a brief review of image splicing and copy-move detection and localization methods. In Section 3, we present the image retrieval algorithm based on HE and BOF. The proposed image clustering algorithm is introduced in Section 4. Section 5 presents the image forgery detection and localization. The experimental results are discussed in Section 6. Finally, Section 7 concludes the paper.

2. Related Works

In the literature, image splicing forgery detection problem has been addressed efficiently [2–4]. In recent years, a substantial attention has been paid to deep learning based approaches [5,6] for localizing image splicing [7–15] wherein convolutional neural network (CNN) has been widely used [8–12]. Bondi et al. [8] extracted and employed features capturing characteristic traces from different camera models to localize a tampered mask by an iterative clustering algorithm. Region proposal network and condition random field are the main components of the model developed in Chen et al. [10]. The noise levels difference between spliced and original regions was utilized to find the splicing traces [11,13]. Non-linear camera response function was used in Yao et al. [11] and was combined with noise level function to exploit the strong relationship between two functions to localize the forged edges using a CNN. Mayer et al. [12] used a similarity network and a CNN-based feature extractor to determine whether image patches contain different traces or being captured by different camera models. Zeng et al. [13] estimated the noise levels using the principal component analysis and then clustered using *k*-means algorithm to localize the spliced regions. Matern et al. [15] utilized the gradient-based illumination descriptor to detect the illumination inconsistency and object color change, which helped localize image splicing traces. Wang et al. [16] used gamma transformation to detect splicing forgery and localize spliced region by estimating the probabilities of sliding window based overlapping blocks being gamma transformed.

CMF detection is the problem of detecting the tampered regions in copy-move images, is called CMF localization (CMFL) in this paper, to distinguish from CMF classification. CMFL has also been actively studied in many researches, which can be divided into three categories: block-based methods [17–22], keypoints-based methods [23–28], and segmentation-based methods [29–32].

Park et al. [17] introduced the upsampled log-polar Fourier descriptor, which is invariant to rotation and scaling, to robustly detect various types of tampering attacks. Wu et al. [18] proposed a two-branch deep neural network to detect potential manipulation via visual artifacts and visually similar regions, which helps specify the copied and pasted regions. Park et al. [19] used the scale space representation of scale-invariant feature transform (SIFT) to handle different geometric transformation. PatchMatch, an algorithm used to search for approximate nearest neighbors, was combined with Zernike moments to detect copy-move attacks in [23,24] whereas SIFT was utilized in [25–27]. In segmentation-based approaches, the input image was semantically segmented into non-overlapped regions [29–32]. Li et al. [29] developed two stages of matching to detect the copy-move regions. Firstly the affine transformation matrix between segmented regions was roughly estimated and then iteratively refined by using an expectation-maximization algorithm-based probability model. However, the major disadvantage of this method is its high computational complexity. Zheng et al. [30] classified smooth regions and non-smooth regions (keypoint regions) to be apply two different techniques. On the one hand, SIFT was used in a keypoint-based method to detect forgery in non-smooth regions. On the other hand, Zernike moments were extracted in a block-based method to handle smooth regions. The CMFL was effectively performed by the fusion of above-mentioned techniques.

3. Bag-of-Features and Hamming Embedding Based Image Retrieval

In image retrieval, images are represented by descriptive features. The features are used to evaluate similarity or dissimilarity between images. In the image forgery, since the forged regions may be rotated, scaled, and translated in different manners, the features of the images should be invariant to these transformations. The features generated by SIFT [33] have such noteworthy characteristics and the proposed algorithm utilizes the SIFT features to represent images [34,35].

In this section, we briefly review the image retrieval method based on BOF [36–38] and HE encoding [38,39]. Suppose that a query image $\mathbf{Q}$ is represented by a set of N descriptors, $\mathbb{X}^{\mathbf{Q}} = \left\{x_1^{\mathbf{Q}}, x_2^{\mathbf{Q}}, \ldots, x_N^{\mathbf{Q}}\right\}$. All of these descriptors are mapped into a visual vocabulary set $\mathbb{W} = \{w_1, w_2, \ldots, w_K\}$ by a K-means vector quantizer q. For example, q maps $x_n^{\mathbf{Q}}$ $(n = 1, 2, \ldots, N)$ to the closest visual word w_k $(k = 1, 2, \ldots, K)$, where $q(x_n^{\mathbf{Q}}) = w_k \in \mathbb{W}$. We define a set of descriptor indexes, which assigns descriptors of $\mathbf{Q}$ to a particular visual word w_k as $\mathbb{I}_k^{\mathbf{Q}} = \left\{n \,\middle|\, q\left(x_n^{\mathbf{Q}}\right) = w_k\right\}$.

A matching model HE is used to estimate the matching of descriptors to a visual word. HE represents each descriptor as D-dimensional binary signatures [38]. Let $b_{n_d}^{\mathbf{Q}}, 1 \leq d \leq D$, is a single bit binary code used to represent $x_n^{\mathbf{Q}}$, then $b_n^{\mathbf{Q}} = \left\{b_{n_1}^{\mathbf{Q}}, b_{n_2}^{\mathbf{Q}}, \ldots, b_{n_d}^{\mathbf{Q}}, \ldots, b_{n_D}^{\mathbf{Q}}\right\}$ is a binary signature of descriptor $x_n^{\mathbf{Q}}$. The Hamming distance between two descriptors, $x_m^{\mathbf{Q}}$ and $x_n^{\mathbf{Q}}$, is computed using their binary signatures as follows:

$$h\left(b_m^{\mathbf{Q}}, b_n^{\mathbf{Q}}\right) = \sum_{d=1}^{D} \left| b_{m_d}^{\mathbf{Q}} - b_{n_d}^{\mathbf{Q}} \right|. \tag{1}$$

Let us denote $\mathbb{X}^{\mathbf{P}}$ be the set of descriptors of the database image $\mathbf{P}$. The probability that two sets of descriptors, $\mathbb{X}^{\mathbf{Q}}$ and $\mathbb{X}^{\mathbf{P}}$ are assigned to the same visual word w_k is defined as:

$$P_k(\mathbb{X}^{\mathbf{Q}}, \mathbb{X}^{\mathbf{P}}) = \sum_{i \in \mathbb{I}_k^{\mathbf{Q}}} \sum_{j \in \mathbb{I}_k^{\mathbf{P}}} f\left(h\left(b_i^{\mathbf{Q}}, b_j^{\mathbf{P}}\right)\right), \tag{2}$$

where the weighting function for a Hamming distance h is calculated as a Gaussian function [38]:

$$f(s) = \begin{cases} e^{-h^2/\sigma^2}, & \text{if } h \leq 3\sigma/2, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The number of dimensions for the binary signatures is typically set to $D = 64$, and the Gaussian bandwidth parameter [38,40] is set to $\sigma = D/4 = 16$.

In order to retrieve images, an inverted index file is built in the image indexing process. The inverted file consists a list of entries. In each entry, a visual word is stored along with the identifier of associated images, descriptors of those images which are assigned to the visual word and the HE used for matching measurement. When the query of $\mathbf{Q}$ is performed, the entries of visual words associated to $\mathbf{Q}$ are searched in the inverted file. The score of a database image $\mathbf{P}$ in this query is calculated by accumulating the Hamming distances between two sets of descriptors' signatures for all the shared visual words of two images. Specifically, the similarity between $\mathbf{Q}$ and $\mathbf{P}$ is defined as follows:

$$\Omega_{\mathbf{Q}}^{\mathbf{P}} = \frac{\sum\limits_{w_k \in \mathbb{W}} \alpha_k P_k(\mathbb{X}^{\mathbf{Q}}, \mathbb{X}^{\mathbf{P}})}{\sqrt{\sum\limits_{w_k \in \mathbb{W}} \alpha_k P_k(\mathbb{X}^{\mathbf{Q}}, \mathbb{X}^{\mathbf{Q}})} \sqrt{\sum\limits_{w_k \in \mathbb{W}} \alpha_k P_k(\mathbb{X}^{\mathbf{P}}, \mathbb{X}^{\mathbf{P}})}}, \tag{4}$$

where the constant α_k is the inverse document frequency [41] of a visual word w_k in $\mathbb{W}$. Suppose that $p(w_k)$ is the probability of w_k occurring in $\mathbb{W}$, then $\alpha_k = -\log p(w_k)$.

4. Image Forgery Clustering

In this section, we give an exposition of the proposed image forgery clustering algorithm. Suppose that we have an input dataset including authentic and tampered images. The proposed algorithm classifies images into separate clusters, where each cluster consists of tampered images which were composed using an identical host image and that host image. Subsequently, the proposed algorithm finds the host image to be the centroid of each image cluster. The details of images clustering and centroid determination are provided in Algorithm 1.

Firstly, we randomly select a query image $\mathbf{Q}$ in the dataset. The ranking score of a database image $\mathbf{P}$ in the query of $\mathbf{Q}$ is denoted as $\Omega_{\mathbf{Q}}^{\mathbf{P}}$ and calculated as the similarity between two images according to Equation (4). The retrieval results are a list of images arranged in descending order of ranking scores. A cut-off threshold θ is set to obtain the set of images. Let us denote $\ddot{\mathbf{Q}}$ as the host image of the image $\mathbf{Q}$ in the dataset. An authentic image is considered as the host image of itself. We need to retrieve all the relevant images $\mathbf{R}$ to the query $\mathbf{Q}$ satisfying $\ddot{\mathbf{R}} = \ddot{\mathbf{Q}}$. To this end, we set the threshold θ to a relatively low value. This low threshold value leads to the case where also some irrelevant images may be retrieved together. Note that, the irrelevantly retrieved images will be discarded in the last step of the iteration. Due to the insignificant processing time of these operations, we can easily handle the case of a large number of images in a cluster. Further, we perform an additional query to ensure that all the relevant images to $\mathbf{Q}$ are retrieved. Notice that the top ranked image in retrieved list $\mathbb{L}_1$, image $\mathbf{D}_1$ is identical to the query image $\mathbf{Q}$. Therefore, the second highest ranked result in $\mathbb{L}_1$, image $\mathbf{D}_2$, is selected as the query image. The score threshold θ is also used in this query, then we obtain the set of retrieved images $\mathbb{L}_2$.

The image cluster $\mathbb{C}$ is the union of two sets of retrieved images, i.e., $\mathbb{C} = \mathbb{L}_1 \cup \mathbb{L}_2$. The centroid of $\mathbb{C}$ is determined based on two criteria which measure the correlations in structure and luminance among images in the cluster. In this work, we extract SIFT features [33] in images and use Random Sample Consensus [42] to find the matching. Let $\mathbb{K}_{\mathbf{UV}} = \{(\mathbf{k}_{\mathbf{U}}^1, \mathbf{k}_{\mathbf{V}}^1), (\mathbf{k}_{\mathbf{U}}^2, \mathbf{k}_{\mathbf{V}}^2), \dots\}$ denote the set of matched keypoints between two images $\mathbf{U}$ and $\mathbf{V}$ in $\mathbb{C}$ where $(\mathbf{k}_{\mathbf{U}}^i, \mathbf{k}_{\mathbf{V}}^i)$ is a pair of keypoints. Then $s_{\mathbf{UV}} = |\mathbb{K}_{\mathbf{UV}}|$ is the number of matching keypoints between $\mathbf{U}$ and $\mathbf{V}$. We denote by $\mathbf{c}_{\mathbf{U}}^i$ the pixel coordinates of $\mathbf{k}_{\mathbf{U}}^i$ in $\mathbf{U}$. The number of matching keypoints in the corresponding positions of $\mathbf{U}$ and $\mathbf{V}$, denoted by $\hat{s}_{\mathbf{UV}}$, is calculated as follows:

$$\hat{s}_{\mathbf{UV}} = \sum_{i=1}^{s_{\mathbf{UV}}} \delta(\mathbf{c}_{\mathbf{U}}^i, \mathbf{c}_{\mathbf{V}}^i), \tag{5}$$

where δ is the Kronecker delta function:

$$\delta(a,b) = \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{if } a \neq b. \end{cases} \tag{6}$$

We define the ratio $\hat{s}_{\mathbf{UV}}/s_{\mathbf{UV}}$ as the structural similarity between $\mathbf{U}$ and $\mathbf{V}$.

Algorithm 1: Image forgery clustering

Input: Image set $\mathbb{S}$
Output: Clusters of relevant images
`/* Each image is classified into only 1 cluster */`
function *Query* $(\mathbf{Q}, \mathbb{S}, \theta)$
  Query $\mathbf{Q}$ in $\mathbb{S}$
  return $\left\{\mathbf{P} | \mathbf{P} \in \mathbb{S}, \Omega_{\mathbf{Q}}^{\mathbf{P}} \geq \theta\right\}$ `// ` $\Omega_{\mathbf{Q}}^{\mathbf{P}}$`: similarity score of P and Q defined in (4)`
procedure *Cluster* $(\mathbb{S})$
  repeat
    Randomly select an image $\mathbf{Q}$ in $\mathbb{S}$
    $\mathbb{L}_1 \leftarrow$ *Query* $(\mathbf{Q}, \mathbb{S}, \theta)$
    Select $\mathbf{D}_2$, the second highest ranking image in the list $\mathbb{L}_1$
    $\mathbb{L}_2 \leftarrow$ *Query* $(\mathbf{D}_2, \mathbb{S}, \theta)$
    $\mathbb{C} \leftarrow \mathbb{L}_1 \cup \mathbb{L}_2$
    foreach $\mathbf{U} \in \mathbb{C}$ **do**
      foreach $\mathbf{V} \in \mathbb{C}, \mathbf{V} \neq \mathbf{U}$ **do**
        Compute $s_{\mathbf{UV}}$ `// number of keypoints matching`
        Compute $\hat{s}_{\mathbf{UV}}$ `// number of keypoints matching at relevant position`
        Compute $l_{\mathbf{UV}}$ `// luminance similarity index`
    $\mathbf{T} \leftarrow \arg\max_{\mathbf{U} \in \mathbb{C}} \sum_{\mathbf{V} \in \mathbb{C}} \left(l_{\mathbf{UV}} + \frac{\hat{s}_{\mathbf{UV}}}{s_{\mathbf{UV}}} \right)$ `// Determine centroid T of cluster` $\mathbb{C}$
    foreach $\mathbf{V} \in \mathbb{C}$ **do**
      if $\hat{s}_{\mathbf{TV}} < s_{\mathbf{TV}}/2$ **then**
        $\mathbb{C} \leftarrow \mathbb{C} \setminus \mathbf{V}$
    $\mathbb{C}$ is a new image cluster with centroid $\mathbf{T}$
    $\mathbb{S} \leftarrow \mathbb{S} \setminus \mathbb{C}$
  until $\mathbb{S} = \emptyset$

In addition, we denote by $\mathbf{U}_Y(x,y)$ the luminance value of image $\mathbf{U}$ at pixel (x,y), which can be calculated as follows [43]:

$$\mathbf{U}_Y(x,y) = 0.299\mathbf{U}_R(x,y) + 0.587\mathbf{U}_G(x,y) + 0.114\mathbf{U}_B(x,y), \tag{7}$$

where $\mathbf{U}_R(x,y)$, $\mathbf{U}_G(x,y)$, and $\mathbf{U}_B(x,y)$ are the red, green, and blue color values of $\mathbf{U}$ at pixel (x,y), respectively. We define $l_{\mathbf{UV}}$, the luminance similarity image between $\mathbf{U}$ and $\mathbf{V}$ as follows:

$$l_{\mathbf{UV}} = \frac{\sum_{x=1}^{H} \sum_{y=1}^{W} \phi\left[\mathbf{U}_Y(x,y) - \mathbf{V}_Y(x,y)\right]}{HW}, \tag{8}$$

where H and W are height and width of $\mathbf{U}$, respectively and

$$\phi(\gamma) = \begin{cases} 1, & \text{if } |\gamma| \leq 3, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

We determine $\mathbf{T}$, the centroid of image cluster $\mathbb{C}$ as follows:

$$\mathbf{T} = \arg\max_{\mathbf{U} \in \mathbb{C}} \sum_{\mathbf{V} \in \mathbb{C}} \left(l_{\mathbf{UV}} + \frac{\hat{s}_{\mathbf{UV}}}{s_{\mathbf{UV}}} \right). \tag{10}$$

Afterwards, we refine the image cluster by discarding irrelevant images $\mathbf{V}$ to $\mathbf{T}$ where $\ddot{\mathbf{V}} \neq \mathbf{T}$ as follows

$$\frac{\hat{s}_{\mathbf{TV}}}{s_{\mathbf{TV}}} < 0.5. \tag{11}$$

Therefore, all the retrieved authentic images, with the exception of the centroid image $\mathbf{T}$, are discarded from the cluster. In other words, $\mathbf{T}$ is the unique authentic image in $\mathbb{C}$. Figure 2 illustrates an example of discarding an image from the cluster according to Equation (11).

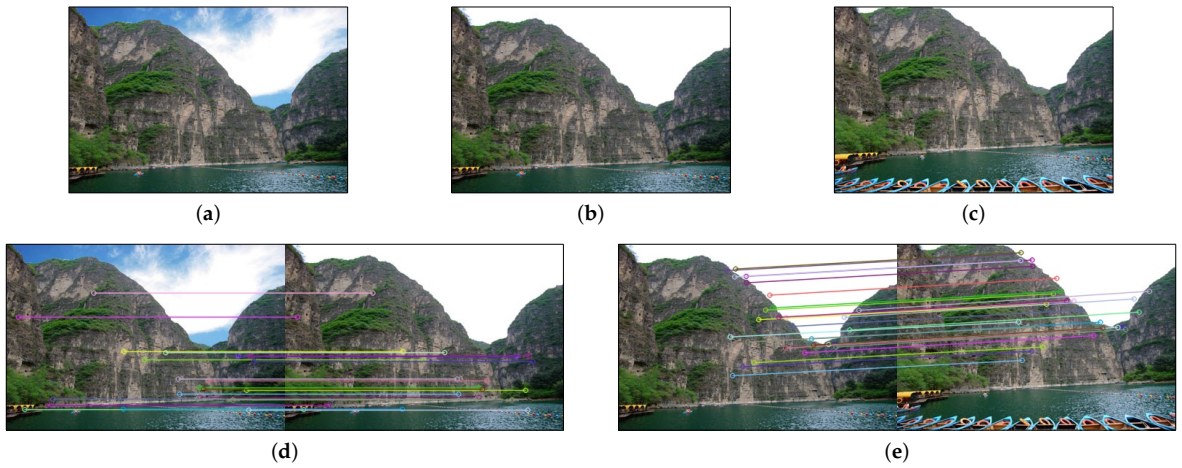

Figure 2. Image (**d**,**e**) are the illustrations of keypoints matching between couples of images (**a**,**b**), (**b**,**c**), respectively. Although (**b**,**c**) have many matching keypoints, the positions of those keypoints in relative images are different. In this example, image (**c**) is discarded from the cluster.

Figure 3 depicts an example of image database indexing and one iteration of the proposed image forgery clustering algorithm to obtain one image cluster with. After each iteration of the proposed clustering algorithm, all the images of the new cluster are excluded from the image database. We repeatedly perform querying and clustering process until the database $\mathbb{S}$ is empty. Finally, each input image belongs to only one cluster.

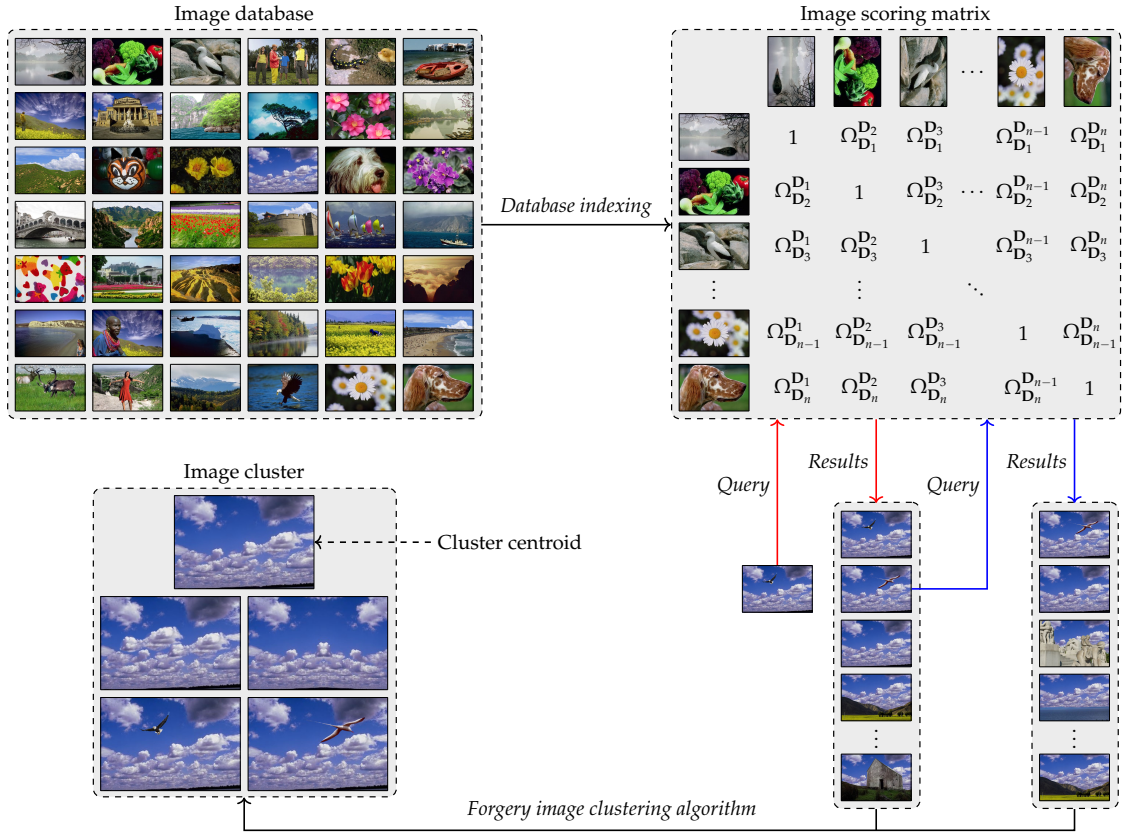

Figure 3. The illustration of image database indexing and one iteration of the proposed image forgery clustering algorithm.

5. Image Forgery Classification and Localization

Given the centroid $\mathbf{T}$ and an image $\mathbf{U}_i$ in the cluster, we can easily estimate the mask of forged regions of $\mathbf{U}_i$ based on $\mathbf{T}_Y - \mathbf{U}_{iY}$. Specifically, $\mathbf{U}_i \cap \mathbf{T}$ denotes the image region including all image pixels that $\mathbf{U}_i$ and $\mathbf{T}$ jointly have, and $\mathbf{U}_i \setminus \mathbf{T}$ denotes the image region in $\mathbf{U}_i$ but not in $\mathbf{T}$.

$$[\mathbf{U}_i \cap \mathbf{T}](x,y) = \begin{cases} \mathbf{U}_i(x,y), & \text{if } |\mathbf{U}_{iY}(x,y) - \mathbf{T}_Y(x,y)| \leq 3, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

$$[\mathbf{U}_i \setminus \mathbf{T}](x,y) = \begin{cases} \mathbf{U}_i(x,y), & \text{if } |\mathbf{U}_{iY}(x,y) - \mathbf{T}_Y(x,y)| > 3, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Consequently, two image regions $\mathbf{U}_i \cap \mathbf{T}$ and $\mathbf{U}_i \setminus \mathbf{T}$ are extracted as shown in Figure 4. These image regions are refined by using median filter to remove salt and pepper noise.

We use SIFT to find the matched regions of $\mathbf{U}_i \cap \mathbf{T}$ and $\mathbf{U}_i \setminus \mathbf{T}$. 3 pairs of matched keypoints are utilized to calculate the affine transformation matrix, and subsequently, a warped image is generated for each transformation matrix. To localize the duplicated regions, the zero mean normalized cross-correlation method is adopted [19]. If we can find such regions, the image $\mathbf{U}_i$ is classified as a copy-move image; otherwise, the tampered image is classified as a spliced image. In Figure 4, images $\mathbf{U}_1$ and $\mathbf{U}_2$ are classified as copy-move images and the detected forgery regions are illustrated in the last column. The previously detected regions, $\mathbf{U}_i \cap \mathbf{T}$, are the target regions in white, and the newly found matched regions are the source of the copy-move operation, which are represented in green. In the last two examples of Figure 4, the spliced regions of images $\mathbf{U}_3$ and $\mathbf{U}_4$, are highlighted in white.

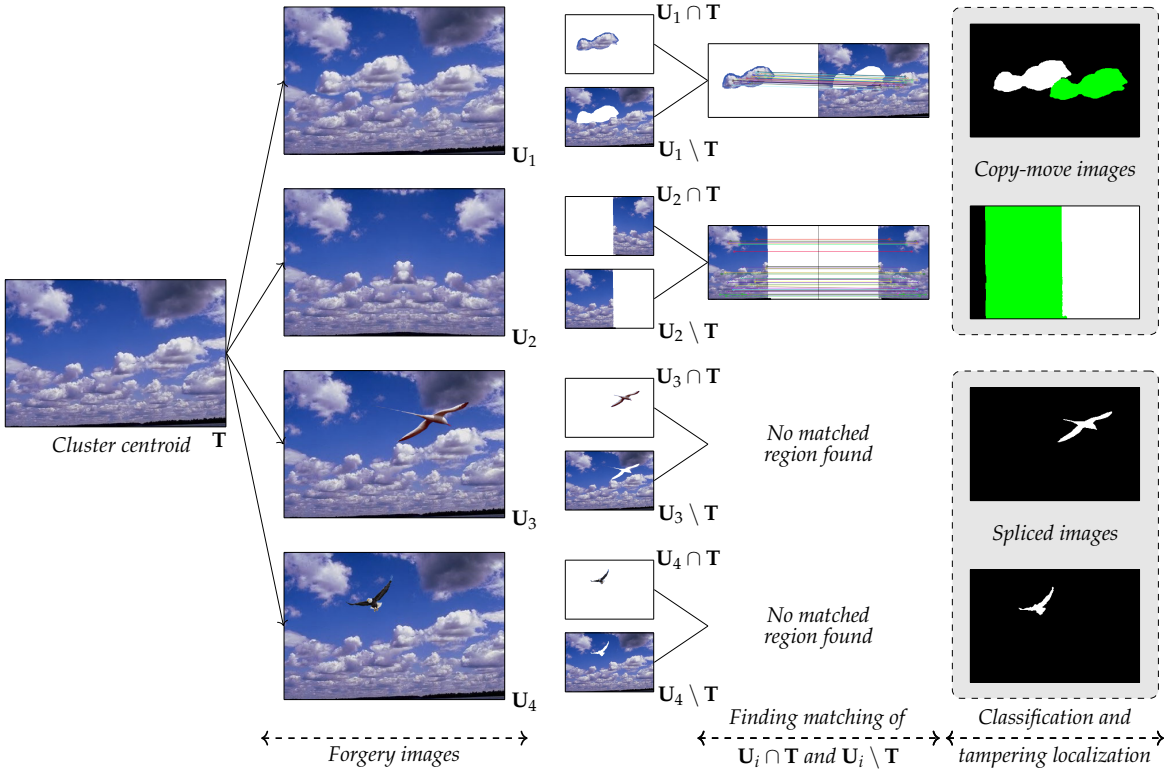

Figure 4. Image forgery classification and localization in an image cluster.

6. Experimental Results

6.1. Datasets

There exist several benchmarking datasets for evaluating the performance of image forgery detection algorithms. In our experiments, we used three challenging datasets MICC-600 [25], GRIP [23], and CASIA 2.0 [44] for the evaluation.

6.1.1. MICC-600

MICC-600 is a dataset of 600 high resolution images with various sizes from to pixels. There are 440 original images and 160 copy-move images. As shown in Figure 5, multiple scenarios of copy-move operations were performed in this dataset:

- Single source region and single target region—Figure 5**c**,**d**,**f**
- Single source region and multiple target regions—Figure 5**b**
- Multiple source regions and multiple target regions—Figure 5**a**,**e**
- Target regions were rotated 30 degree counter-clockwise—Figure 5**a**,**f**
- Target regions were scaled by 120%—Figure 5**a**

6.1.2. GRIP

GRIP is a small dataset with copy-move and original images. All the images in this dataset have either resolution 1024 × 768 or 768 × 1024. The target regions in copy-move images were composed using different attacks, such as compression, noise addition, rotation, scaling.

6.1.3. CASIA 2

CASIA 2 is a big dataset with more than 12,000 images in three categories: authentic, spliced and copy-move images. The images in this dataset are in low resolution with the sizes vary from 240 × 160

to 900 × 600 pixels. Among three datasets in our simulations, CASIA 2 is the only dataset which has both types of forgery: splicing and copy-move.

(a) (b) (c) (d) (e) (f)

Figure 5. Examples of CMFL in MICC-600 dataset. First row: original images, second row: copy-move images, third row: ground truth images, and fourth row: source regions (green) and target regions (white) detected by the proposed method.

6.2. Evaluation Metrics

In the experiments, we evaluate the performance of image retrieval and image forgery classification and localization.

6.2.1. Metrics for Image Retrieval

To evaluate the performance of the proposed image forgery clustering algorithm, we use the mean average precision (MAP) metric used in image retrieval problem. For a query q, let us denote N_q the number of retrieved images, M_q the number of relevant images, and $\mathrm{Rel}_q(k)$ the number of relevant images in top k retrieved results. The precision and recall of query q at cut-off k, denoted by $P_q(k)$ and $R_q(k)$, are calculated as follows:

$$P_q(k) = \frac{\mathrm{Rel}_q(k)}{k}. \tag{14}$$

$$R_q(k) = \frac{\mathrm{Rel}_q(k)}{M_q}. \tag{15}$$

Then, the average precision for query q is computed as follows:

$$AP_q = \sum_{k=1}^{N_q} P_q(k)\Delta R_q(k), \tag{16}$$

where $\Delta R_q(k) = R_q(k) - R_q(k-1)$ is the change in recall from items $k-1$ to k. Note that $R_q(0) = 0$. Finally, MAP for all the queries is defined as follows:

$$MAP = \frac{\sum_{q=1}^{Q} AP_q}{Q}, \tag{17}$$

where Q is the number of queries.

6.2.2. Metrics for Image Forgery Classification and Localization

Since we concurrently classify image forgery types and localize the forged regions, the evaluation is performed in both image and pixel levels.

To quantitatively evaluate the performance of forgery localization, we adopt two metrics for tampered regions of a classified tampered image [19], localization precision L_P and localization recall L_R, which are defined as follows:

$$L_P = \frac{\text{\# correctly detected pixels}}{\text{\# all detected pixels}}. \tag{18}$$

$$L_R = \frac{\text{\# correctly detected pixels}}{\text{\# all tampered pixels}}. \tag{19}$$

Similarly, we define the classification precision C_P, and recall C_R at image level:

$$C_P = \frac{\text{\# correctly detected tampered images}}{\text{\# all detected tampered images}}. \tag{20}$$

$$C_R = \frac{\text{\# correctly detected tampered images}}{\text{\# all tampered images}}. \tag{21}$$

In order to balance between precision and recall, we consider both of these quantities by computing their harmonic mean, called localization F-measure, as follows:

$$L_F = \frac{2L_P L_R}{L_P + L_R}. \tag{22}$$

$$C_F = \frac{2C_P C_R}{C_P + C_R}. \tag{23}$$

The metrics precision, recall, and F-measure at pixel level are used for all 3 datasets in this work. Nevertheless, the metrics at image level are only used to evaluate performance of the proposed method in MICC-600 and GRIP datasets. To evaluate the classification performance in CASIA 2, which contains 3 classes, we use confusion matrix.

6.3. Image Retrieval Results

To evaluate the performance of the proposed image forgery clustering algorithm, we carry out the experiments to estimate MAP of image retrieval in 3 different scenarios related to cluster formation of Algorithm 1. In the first case, only one query is performed to compose the cluster. In the second case, the second query is performed to augment the retrieved results. In the third case, the cluster refinement using structural correlation of Algorithm 1 is conducted after two queries to form the image forgery cluster. We denote these cases by case A, case B, and case C, respectively. Figure 6 shows the retrieval performance of 3 above-mentioned cases in 3 datasets. It is clear that MAP significantly increases from case A to case C in all 3 datasets to prove the efficiency of the proposed image retrieval based clustering algorithm.

We present the average ratios that the host image of the query is retrieved in 3 cases in Table 1. The results ensure that by using image forgery clustering algorithm, we can generally retrieve the host images of query images into the clusters.

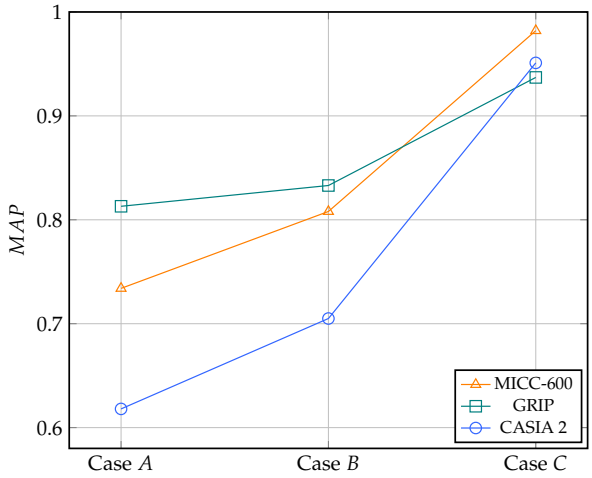

Figure 6. The mean average precision obtained by 3 scenarios of image retrieval performed by the proposed clustering algorithm.

Table 1. Average ratios when the host image of the query image is retrieved (%).

	Case *A*	**Case *B***	**Case *C***
MICC-600	87.6	98.3	98.3
GRIP	100	100	100
CASIA 2	97.5	99.6	99.4

6.4. Forgery Detection and Localization Results on MICC-600 Dataset

Table 2 presents the performance of the proposed method in comparison with state-of-the-art on MICC-600 dataset. Our classification F-measure outperforms Li. et al. [29] and is slightly lower than Li. et al. [27]. In term of localization performance, our method surpasses other methods with $L_F = 93.1\%$. Visual examples of CMFL are shown in Figure 5 where we distinguish the source and target regions in green and white, respectively.

Table 2. Performance comparison on MICC-600 dataset (%).

	C_P	C_R	C_F	L_P	L_R	L_F
Li et al. [29]	69.8	88.1	77.9	86	88	87
Jin et al. [26]	-	-	-	90.2	93.7	91.9
Li et al. [27]	97.5	86.2	**91.5**	-	-	91.8
Proposed method	88.6	92.5	90.5	90.8	95.5	**93.1**

6.5. Forgery Detection and Localization Results on GRIP Dataset

Table 3 summarizes the performance on GRIP dataset where the proposed method exceeds other methods in both classification and localization indexes. The evaluations in different types of copy-move situations are also considered. Specifically, 4 attacks includes Gaussian noise addition and JPEG compression to the copy-move images, rotation and scaling to the copied regions. Figure 7a indicates that our method is better than other methods in term of localization F-measure with different levels of Gaussian noise added to the copy-move images. Figure 7b shows that different CMFL methods handling JPEG compression situation with a slight difference. In case the copied regions are rotated or scaled, Chen et al. [20], Chen et al. [32], and our method sequentially perform better than the rest (Figure 7c,d). The proposed method performs better than other methods when the changes of the copied regions are small. On the contrary, its performance declines for larger rotation angle and scaling factor. Figure 8 illustrates the CMFL examples of the proposed method on GRIP dataset.

Table 3. Performance comparison on GRIP dataset (%).

	C_P	C_R	C_F	L_P	L_R	L_F
Chen et al. [20]	-	-	-	-	-	95.33
Cozzolino et al. [24]	-	-	94.61	-	-	94.06
Li et al. [27]	100	100	**100**	-	-	94.66
Bi et al. [28]	-	-	96.63	-	-	92.98
Chen et al. [32]	-	-		-	-	95.77
Proposed method	96.3	98.8	97.5	96.2	97.4	**96.8**

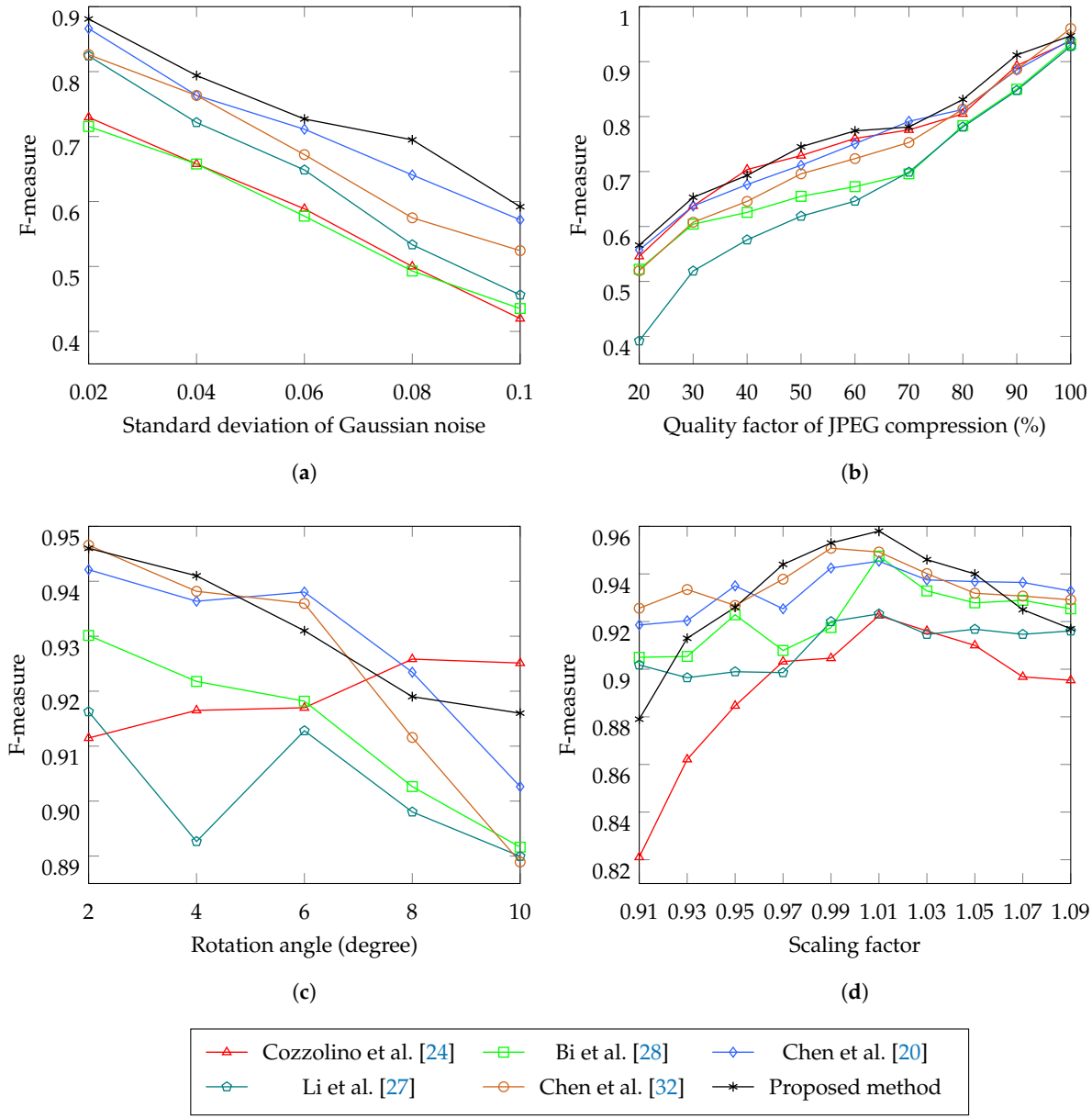

Figure 7. Pixel level *F*-measure comparison in the GRIP dataset with different types of copy-move attacks.

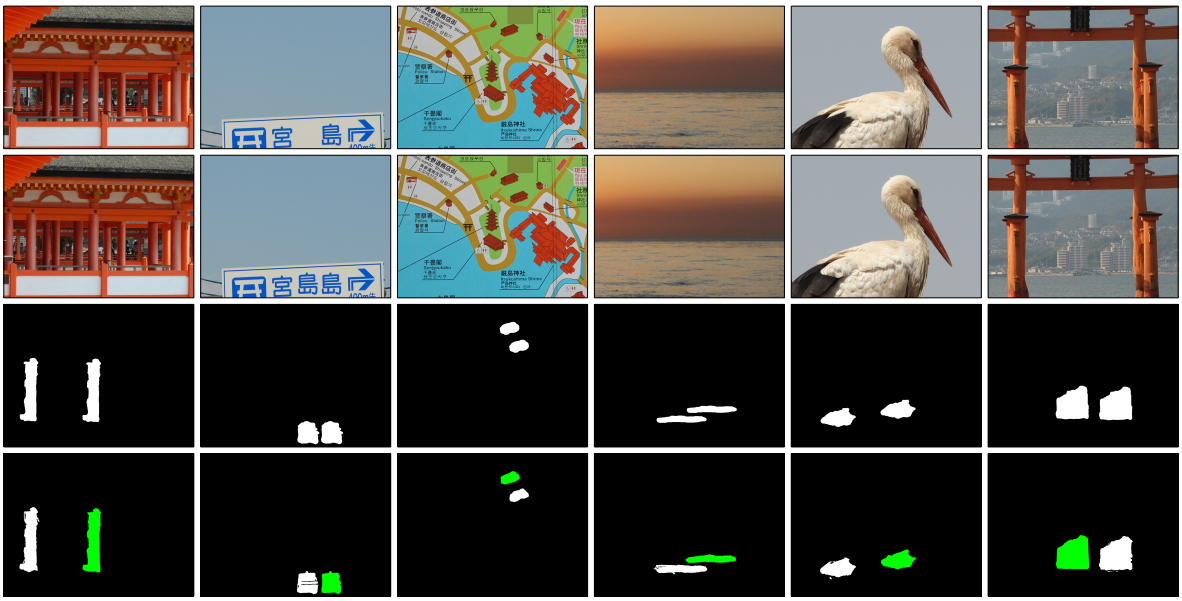

Figure 8. Examples of copy-move forgery detection in GRIP dataset. First row: original images, second row: copy-move images, third row: ground truth images, and forth row: source regions (green) and target regions (white) detected by the proposed method.

6.6. Forgery Detection and Localization Results on CASIA 2 Dataset

Table 4 summarizes the 3-class classification results of the proposed method on the CASIA 2 dataset. To the best of our knowledge, all of the previous researches on forgery detection of this dataset are binary classification. Therefore, only the results of our work are reported. The detection accuracy of authentic images achieve 96.9%, which is higher than two image forgery types. 6.7% of copy-move images are classified as spliced images. By contrast, 4.4% of spliced images are mistakenly detected as copy-move images.

Table 5 and Table 6 compare the proposed method with other researches on localization performance of spliced images and copy-move images of CASIA 2 dataset, respectively. Examples of CMFL results of the proposed method are shown in Figure 9. Since CASIA 2 is the most challenging dataset in our experiments with many small and smooth tampered regions, the proposed method occasionally fails to search for matching regions.

Table 4. Performance of image forgery classification in CASIA 2 dataset (%).

		Actual Class		
		Authentic	**Copy-Move**	**Splicing**
Predicted class	Authentic	96.9	1.8	1.9
	Copy-move	1.0	91.5	4.4
	Splicing	2.1	6.7	93.7

Table 5. Performance of image splicing localization on CASIA 2 (%).

	L_P	L_R	L_F
Shi et al. [9]	77	51	62
Chen et al. [10]	-	-	73.88
Proposed method	80.3	70.9	**75.3**

Table 6. Performance of copy-move images localization on CASIA 2 (%).

	L_P	L_R	L_F
Abd-Almageed et al. [18]	77.38	59.15	67.05
Cozzolino et al. [24]	81.87	61.34	70.13
Wu et al. [22]	67.83	85.69	75.72
Proposed method	79.2	73.8	**76.4**

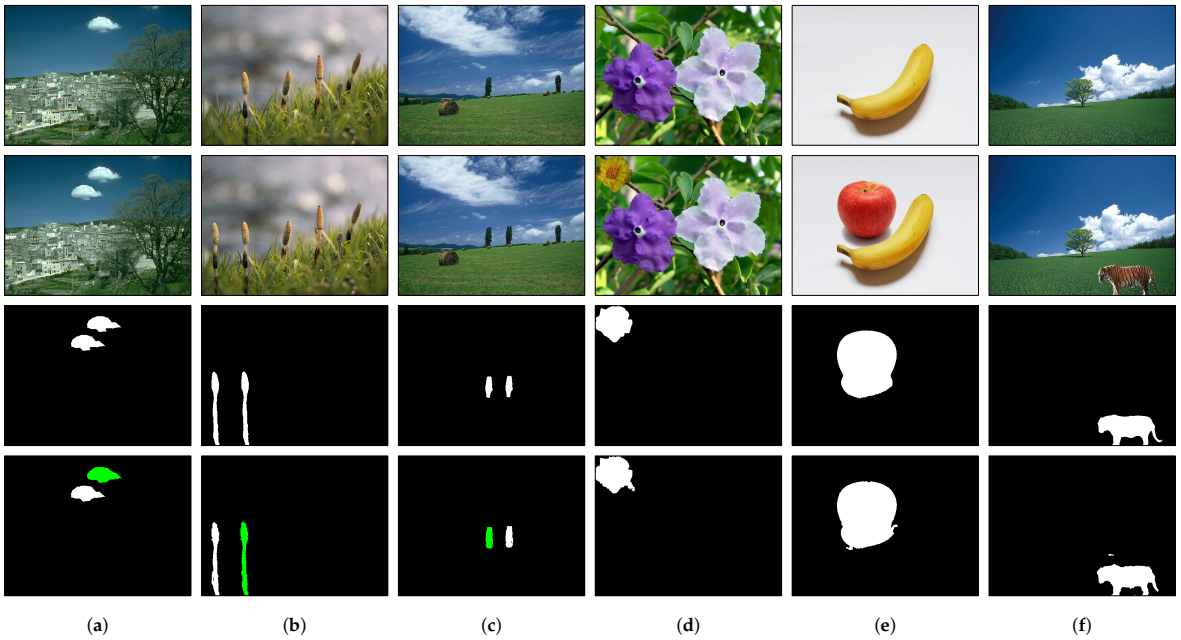

Figure 9. Examples of splicing forgery detection in CASIA 2 dataset. Columns (**a**–**c**): copy-move images, columns (**d**–**f**): spliced images. First row: original images, second row: spliced images, third row: ground truth images, and forth row: tampered regions detected by the proposed method.

7. Conclusions

This paper introduces a novel method to detect and localize authentic images and two types of tampered images: copy-move and spliced images. We propose a robust algorithm to divide relevant images into cluster using BOF and HE based image retrieval. From image clusters, by exploiting the structural correlation between images, the proposed algorithm determines the cluster centroid, which is the only authentic image in the cluster. Afterwards, the image forgery are classified, and the forged regions are localized. The experimental results show that this method achieves higher performance in both forgery detection and localization in comparison with state-of-the-art methods. Notably, the proposed method can indicate the source and target regions of copy-move images.

Author Contributions: Conceptualization, N.T.P., J.-W.L., and C.-S.P.; methodology, N.T.P.; software, N.T.P.; formal analysis, N.T.P.; resources, J.-W.L.; data curation, N.T.P.; writing—original draft preparation, N.T.P.; writing—review and editing, J.-W.L. and C.-S.P.; visualization, N.T.P.; supervision, J.-W.L. and C.-S.P.; project administration, J.-W.L. and C.-S.P.; funding acquisition, J.-W.L. and C.-S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2016-0-00312) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation). This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2019R1F1A1055593).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CMF	Copy-Move Forgery
CNN	Convolutional Neural Network
CMFL	Copy-Move Forgery Localization
SIFT	Scale Invariant Feature Transform
BOF	bag-of-features
HE	Hamming Embedding
MAP	mean average precision

References

1. Pham, N.T.; Lee, J.W.; Kwon, G.-R.; Park, C.-S. Hybrid Image-Retrieval Method for Image-Splicing Validation. *Symmetry* **2019**, *11*, 83. [CrossRef]
2. Chen, B.; Qi, X.; Sun, X.; Shi, Y.-Q. Quaternion pseudo-Zernike moments combining both of RGB information and depth information for color image splicing detection. *J. Vis. Commun. Image Represent* **2017**, *49*, 283–290. [CrossRef]
3. He, Z.; Lu, W.; Sun, W.; Huang, J. Digital image splicing detection based on Markov features in DCT and DWT domain. *Pattern Recognit.* **2012**, *45*, 4292–4299. [CrossRef]
4. Pham, N.T.; Lee, J.W.; Kwon, G.-R.; Park, C.-S. Efficient image splicing detection algorithm based on Markov features. *Multimedia Tools Appl.* **2018**, *78*, 12405–12419. [CrossRef]
5. Vo, A.H.; Le, T.; Vo, M.T.; Le, T. A Novel Framework for Trash Classification Using Deep Transfer Learning. *IEEE Access* **2019**, *7*, 178631–178639. [CrossRef]
6. Le, T.; Vo, M.T.; Kieu, T.; Hwang, E.; Rho, S.; Baik, S. Multiple Electric Energy Consumption Forecasting Using a Cluster-Based Strategy for Transfer Learning in Smart Building. *Sensors* **2020**, *20*, 2668. [CrossRef]
7. Zampoglou, M.; Papadopoulos, S.; Kompatsiaris, Y.; Kompatsiaris, I. Large-scale evaluation of splicing localization algorithms for web images. *Multimedia Tools Appl.* **2016**, *76*, 4801–4834. [CrossRef]
8. Bondi, L.; Lameri, S.; Güera, D.; Bestagini, P.; Delp, E.; Tubaro, S. Tampering Detection and Localization Through Clustering of Camera-Based CNN Features. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1855–1864.
9. Shi, Z.; Shen, X.; Kang, H.; Lv, Y. Image Manipulation Detection and Localization Based on the Dual-Domain Convolutional Neural Networks. *IEEE Access* **2018**, *6*, 69472–69480. [CrossRef]
10. Chen, B.; Qi, X.; Wang, Y.; Zheng, Y.; Shim, H.J.; Shi, Y.-Q. An Improved Splicing Localization Method by Fully Convolutional Networks. *IEEE Access* **2018**, *6*, 69472–76453. [CrossRef]
11. Yao, H.; Wang, S.; Zhang, X.; Qin, C.; Wang, J. Detecting Image Splicing Based on Noise Level Inconsistency. *Multimedia Tools Appl.* **2017**, *76*, 12457–124797. [CrossRef]
12. Mayer, O.; Stamm, M.C. Forensic Similarity for Digital Images. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 1331–1346. [CrossRef]
13. Zeng, H.; Zhan, Y.; Kang, X.; Lin, X. Image splicing localization using PCA-based noise level estimation. *Multimedia Tools Appl.* **2017**, *76*, 4783–4799. [CrossRef]
14. Zheng, Y.; Cao, Y.; Chang, C.-H. A PUF-Based Data-Device Hash for Tampered Image Detection and Source Camera Identification. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 620–634. [CrossRef]
15. Matern, F.; Riess, C.; Stamminger, M. Gradient-Based Illumination Description for Image Forgery Detection. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 1303–13170. [CrossRef]
16. Wang, P.; Liu, F.; Yang, C.; Luo, X. Blind forensics of image gamma transformation and its application in splicing detection. *J. Vis. Commun. Image Represent.* **2018**, *55*, 80–90. [CrossRef]
17. Park, C.-S.; Kim, C.; Lee, J.; Kwon, G.-R. Rotation and scale invariant upsampled log-polar fourier descriptor for copy-move forgery detection. *Multimedia Tools Appl.* **2016**, *75*, 16577–16595. [CrossRef]
18. 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), Munich, Germany, 8–14 October 2018; pp. 170–186.

19. Park, C.-S.; Choeh, J.Y. Fast and robust copy-move forgery detection based on scale-space representation. *Multimedia Tools Appl.* **2017**, *77*, 16795–16811. [CrossRef]
20. Chen, B.; Yu, M.; Su, Q.; Shim, H.J.; Shi, Y.-Q. Fractional Quaternion Zernike Moments for Robust Color Image Copy-Move Forgery Detection. *IEEE Access* **2018**, *6*, 56637–56646. [CrossRef]
21. Zhong, J.-L.; Pun, C.-M. An End-to-End Dense-InceptionNet for Image Copy-Move Forgery Detection. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2134–2146. [CrossRef]
22. Wu, Y.; Abd-Almageed, W.; Natarajan, P. Image Copy-Move Forgery Detection via an End-to-End Deep Neural Network. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1907–1915.
23. Cozzolino, D.; Poggi, G.; Verdoliva, L. Copy-move forgery detection based on PatchMatch. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Beijing, China, 6–10 July 2013; pp. 5312–5316.
24. Cozzolino, D.; Poggi, G.; Verdoliva, L. Efficient Dense-Field Copy–Move Forgery Detection. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 2284–2297. [CrossRef]
25. Amerini, I.; Ballan, L.; Caldelli, R.; Del Bimbo, A.; Del Tongo, L.; Serra, G. Copy-move forgery detection and localization by means of robust clustering with J-Linkage. *Signal Process. Image Commun.* **2013**, *28*, 659–669. [CrossRef]
26. Jin, G.; Wan, X. An improved method for SIFT-based copy–move forgery detection using non-maximum value suppression and optimized J-Linkage. *Signal Process. Image Commun.* **2017**, *57*, 113–125. [CrossRef]
27. Li, Y.; Zhou, J. Fast and Effective Image Copy-Move Forgery Detection via Hierarchical Feature Point Matching. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1307–1322. [CrossRef]
28. Bi, X.; Pun, C.-M. Fast copy-move forgery detection using local bidirectional coherency error refinement. *Pattern Recognit.* **2018**, *81*, 161–175. [CrossRef]
29. Li, J.; Li, X.; Yang, B.; Sun, X. Segmentation-Based Image Copy-Move Forgery Detection Scheme. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 507–518.
30. Zheng, J.; Liu, Y.; Ren, J.; Zhu, T.; Yan, Y.; Yang, H. Fusion of block and keypoints based approaches for effective copy-move image forgery detection. *Multidimens. Syst. Signal Process.* **2016**, *27*, 989–1005. [CrossRef]
31. Pun, C.-M.; Yuan, X.; Bi, X.-L. Image Forgery Detection Using Adaptive Oversegmentation and Feature Point Matching. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1705–1716.
32. Chen, B.; Yu, M.; Su, Q.; Li, L. Fractional quaternion cosine transform and its application in color image copy-move forgery detection. *Multimedia Tools Appl.* **2018**, *78*, 8057–8073. [CrossRef]
33. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
34. Manzo, M. Graph-Based Image Matching for Indoor Localization. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 785–804. [CrossRef]
35. Manzo, M.; Pellino, S. Bag of ARSRG Words (BoAW). *Mach. Learn. Knowl. Extr.* **2019**, *1*, 871–882. [CrossRef]
36. Tolias, G.; Avrithis, Y.; Jégou, H. To Aggregate or Not to aggregate: Selective Match Kernels for Image Search. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Sydney, NSW, Australia, 1–8 December 2013; pp. 1401–1408.
37. Arandjelović, R.; Zisserman, A. DisLocation: Scalable Descriptor Distinctiveness for Location Recognition. In Proceedings of the Asian Conference on Computer Vision (ACCV), Singapore, 1–5 November 2014; pp. 188–2044.
38. Jegou, H.; Douze, M.; Schmid, C. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In Proceedings of the European Conference on Computer Vision (ECCV), Marseille, France, 12–18 October 2008; pp. 304–317.
39. Sattler, T.; Havlena, M.; Schindler, K.; Pollefeys, M. Large-Scale Location Recognition and the Geometric Burstiness Problem. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1582–1590.
40. Jegou, H.; Douze, M.; Schmid, C. On the burstiness of visual elements. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 1169–1176.
41. Robertson, S. Understanding inverse document frequency: On theoretical arguments for IDF. *J. Doc.* **2004**, *60*, 503–520. [CrossRef]

42. Fischler, M.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
43. Gonzalez, R.C.; Woods, R.E. Image Compression. In *Digital Image Processing*, 3rd ed.; Pearson Prentice Hall: Upper Saddle River, NJ, USA, 2008; pp. 608.
44. Dong, J.; Wang, Y.; Tan, T. CASIA Image Tampering Detection Evaluation Database. In Proceedings of the 2013 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), Beijing, China, 6–10 July 2013; pp. 422–426.