Neural Computing and Applications (2021) 33:4713–4727
https://doi.org/10.1007/s00521-020-05433-1

S.I. : HIGHER LEVEL ARTIFICIAL NEURAL NETWORK BASED INTELLIGENT SYSTEMS

Copy-move forgery detection technique based on discrete cosine transform blocks features

Esteban Alejandro Armas Vega[1] · Edgar González Fernández[1] · Ana Lucila Sandoval Orozco[1] · Luis Javier García Villalba[1]

Received: 1 September 2020 / Accepted: 7 October 2020 / Published online: 20 October 2020

Abstract
With the increasing number of software applications that allow altering digital images and their ease of use, they weaken the credibility of an image. This problem, together with the ease of distributing information through the Internet (blogs, social networks, etc.), has led to a tendency for information to be accepted as true without its veracity being questioned. Image counterfeiting has become a major threat to the credibility of the information. To deal with this threat, forensic image analysis is aimed at detecting and locating image forgeries using multiple clues that allows it to determine the veracity or otherwise of an image. In this paper, we present a method for the authentication of images. The proposed method performs detection of copy-move alterations within an image, using the discrete cosine transform. The characteristics obtained from these coefficients allow us to obtain transfer vectors, which are grouped together. Through the use of a tolerance threshold, it is possible to determine whether there are regions copied and pasted within the analysed image. The results obtained from the experiments reported in this paper demonstrate the effectiveness of the proposed method. For the evaluation of the proposed methods, experiments were carried out with public databases of falsified images that are widely used in the literature.

Keywords Cope-move forgery · Digital images · Discrete cosine transforms · Forgery detection

1 Introduction

As the famous saying goes, "A picture is worth a thousand words" and this has never been more true than in today's visually oriented society. Currently, images are used in many common areas such as teaching, journalism, jurisprudence, medicine, advertising, art, etc.

Driven by social media networks and instant messaging applications, multimedia content is the primary source of internet traffic. Besides all of this, the continuous improvement of the cameras incorporated in mobile devices together with the evolution of the image editing tools has made it easier to manipulate an image with excellent results and shared it with the world through the Internet in no time. Although manipulated images have been around for decades and are present in many sectors (politics, cinema, press, legal, etc.), nowadays the availability of resources to share information makes image tampering

✉ Luis Javier García Villalba
javiergv@fdi.ucm.es

Esteban Alejandro Armas Vega
esarmas@ucm.es

Edgar González Fernández
edggonza@ucm.es

Ana Lucila Sandoval Orozco
asandoval@fdi.ucm.es

[1] Group of Analysis, Security and Systems (GASS), Department of Software Engineering and Artificial Intelligence (DISIA), Faculty of Computer Science and Engineering, Office 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, Spain

dangerous, making people think that what they are seeing is the truth.

Regarding the legal use of multimedia content, photograph and video evidence can be extremely useful in legal proceedings. Potential evidence is everywhere, thanks to the proliferation of surveillance cameras, smartphones, tablets, and social media networks. Nevertheless, any image and video to be admissible in court must meet two basic requirements [15]: relevance and authenticity. For the evidence to be relevant, it must be sufficiently useful to prove something important in a trial, which means that it must either support or undermine the truth during the legal proceedings. And, to be authenticated, the evidence must accurately represent its subject.

Over the years, image editing tools have been perfected, offering better results and simplifying their functionality. It is now relatively simple to make more realistic tampered multimedia files, such as images and videos, without leaving any noticeable shreds of evidence . This leaves a challenging task for the forensic analyst to validate the authenticity of images as it is almost impossible for the human naked eye to distinguish between the forged image and the real one.

In this matter, this paper proposes a method which performs detection of copy-move alterations in an image using the discrete cosine transform. The characteristics obtained from these coefficients allow us to obtain transfer vectors, which are grouped together and, through the use of a tolerance threshold, it is possible to determine whether or not there are regions copied and pasted within the analyzed image. Our experiments were carried out with manipulated images from public databases widely used in the literature and they demonstrate the efficiency of the proposed method.

The rest of the work is organized as follows: Section 2 details the types of techniques and tools used in the manipulation of digital images, focusing on the most relevant techniques from the passive approach. The details of the proposed detection technique are presented in Sect. 3. Section 4 analyses the results of the experiments carried out and, finally, the conclusions of the work are included in Sect. 5.

2 Related work

Image forgery analysis can be divided into two main approaches [8, 23]: active and passive. These two approaches use different methods and techniques to achieve their goals. The methods and techniques used in the active approaches are mainly based on the analysis of digital watermarks, and signatures left by the device during the image generation. Although, one of the major drawbacks of this type of approach is that many cameras do not have the ability to incorporate such watermarks or signatures, so their scope is limited. On the other hand, passive approaches analyse the content and characteristics of the image without any prior information. The techniques based on this approach focus in the analysis of any inconsistencies in the image features such as noise, camera response function (CRF), colour filter array (CFA), etc.

The approaches and their techniques used in digital image forgery detection are shown in Fig. 1. Because the algorithm proposed in this paper is part of the passive approach, techniques principally used in the passive analysis will be described below.

2.1 JPEG compression properties

JPEG is one of the most popular and commonly used compression formats for digital images [6]. In the early years of digital photography, the majority of digital cameras exported their pictures in a JPEG format, and nowadays, almost all devices with a built-in digital camera generate and save their images using this format. Moreover, when a forgery technique such as splicing or copy-move is applied over an image, the double JPEG compression is inevitable. Then, identify whether a JPEG image has been re-compressed or not is an important matter when a forensic analysis is conducted.

Huang et al. [18], proposed a method to detect JPEG double compression on images with the same quantization matrix. Their approach is based on the observation of how the JPEG coefficients monotonically decrease between the first and second JPEG compression of the image. The authors use a random perturbation strategy to discriminate the difference between a single and a double compressed image, especially when the image is compressed with a relatively high-quality factor. Their experiments show a direct relationship between the accuracy and the quality factor of the test image.

The intrinsic property of decreasing the number of JPEG coefficients, leave by two consecutively compressed images with the same quantization matrix, can be used to detect double JPEG compression [18]. Nevertheless, if the JPEG images have been compressed with low-quality factors, then detecting double compression in this scenario becomes challenging to achieve. Niu et al. [25] proposed an approach that enhances Huang et al.'s method and makes it capable to detect double compression on images compressed with a low-quality factors. The main difference between [18] and Niu et al.'s method lies on the random selection of $+/-1$ valued JPEG coefficient of the recompressed image. The experiments show that the method increase up to 1.74% the accuracy compared with previous

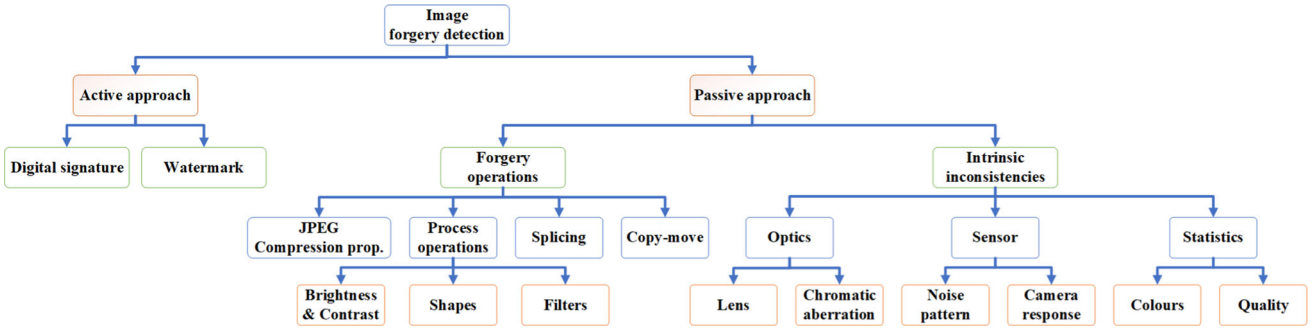


Fig. 1 Image forgery detection approaches and techniques

algorithms, specially when the quality factor is less than 90.

2.2 Process operations

Image processing is the manipulation of images using digital computers [30]. Its use has been growing in the last decades, and its purposes vary from medicine to entertainment. The action domain of this type of image manipulation is to rotate, to scale, to filter or to adjust brightness and contrast of an image.

Kee et al. [21] proposed an algorithm to detect the use of image post-processing retouches to enhance the images in magazine covers. To do that, the authors use a dataset composed by 468 images and introduced a metric (range 1–5) for quantifying alterations of the image done by digital photo-editing techniques, depending on the amount of image alteration. The photometric and geometrical modifications of the original and the retouched photograph were calculated for each picture. Then, eight statistics were extracted—four statistics from the mean and standard deviation of the motion magnitude calculated individually on the face and body and four statistics from the means and standard deviations of both the spatial boundaries of local smoothing/sharpening filters and the Structural Similarity Index Metric (SSIM)—incorporating the degree of photographic retouch to calculate the correlation with the evaluation of each photograph. In the experiments a support vector machine (SVM) was used to calculated the degree of modification of the image. The absolute prediction error was below 0.5 and 1.0 for 81.4% and 99.1% of the images, respectively.

The detection of image sharpening is one of the main topics in image forensics, and the most popular sharpening method is the unsharp mask (USM) [24]. Ding et al. [13] proposed a technique, to detect image USM sharpening, which used the overshoot artifacts left by the USM algorithm on the image's edge pixels. The author's method extract features from the edge perpendicular binary coding histogram to train an SVM, and then to define whether an image was sharper using USM or not. Despite that, the experimental results show that Ding et al.'s method outperforms existing methods; it still remains at a low level for weak image sharpening. To improve the results from their previous work, Ding et al. [12] proposed a new method that is capable of detecting weak USM sharpening. The new method differs from the previous one, by introducing further steps such as the edge direction, the edge areas definition using interpolation algorithms, and the local threshold calculation before building the histogram and extract the features to feed the SVM to trained. The experimental results show a 97.85% of accuracy, which is a superior performance compared with similar algorithms.

When a retouched image is compared with the original one, the face identification is considerably degraded. Due to this, Bharati et al. [5] proposed a supervised algorithm that use Boltzmann machine to detect retouching in face images. Moreover, to evaluate the proposed approach the authors introduce two face image databases with unaltered and retouched images (ND-IIITD). The authors compare their supervised method with similar state-of-the-art algorithm proposed by Kee et al. [21] and the obtained results show a better performance by Bharati et al.'s approach. While Kee et al. proposal gets a correct classification accuracy of slightly less than 50%, the Boltzmann machine supervised algorithm proposed by Bharati et al. gets an 87.1% which is a great improvement in the accuracy detection.

2.3 Splicing detection

Image splicing is a common and relatively easy task to perform, and many modern tools provide ways to conceal the modification by applying further post-processing operations, leaving no visible traces. Splice detection can be addressed in many ways, for example, by detecting signal differences in the original background and the spliced fragment, or by detecting post-processing operations applied to borders.

When splice manipulation is performed, the local distribution of the edge micro-patterns is altered by introducing new micro-patterns into the pasted region. Therefore, it changes its regularity and the local frequency distribution. All of the methods discussed below differ only in the way they model the structural changes caused by counterfeiting. The success of a method depends on the representation of these changes, which will be the characteristics with which the different images will be trained and classified.

Most algorithms for splicing detection are split into subprocesses, which are common to each other, as shown in Fig. 2.

Shi et al. [28] presented a splice forgery detection method that extracts two features, statistical features from the image and the features of a multi-size block discrete cosine transform (MBDCT). These two features are the input vector for the SVM classifier. The author's experiments show a great detection rate, up to 90% of accuracy using the public dataset "Columbia [10]".

Zhang et al. [35] proposed an algorithm to identified spliced images. This method uses the features obtained from the 2D matrices resulting from applying MBDCT [28]—the author's work used as features, the image quality metrics (IQM). The resulting vector is the input of the SVM. The dataset used was "Columbia" and the accuracy was up to 87.10%

Wang et al. [33] proposed a tampering detection method. This method is based on modelling edge information. The author's algorithm converts the image from RGB to YCbCr and to extract the edge information use the Cb and Cr components. Once that the features are extracted, a vector with nine components is built and used as the SVM classifier input. The results have shown that the algorithm is useful for tampering detection. The accuracy obtained was up to 95.6% using "CASIA TIDE v2.0 [14]" public dataset.

Zhao et al. [38] used different space colours to detect image splicing. The authors analyse the YCrCb space colour versus the commonly utilized RGB. The algorithm extracts a four grey level run-length run-number (RLRN) vectors. Later the characteristic extraction, the resulting vector becomes the input of an SVM. During the experiments, the authors used "CASIA TIDE v1.0" and "Columbia" public datasets. The detection rate was up to 94.7% of precision. Therefore, the YCrCb space colour is more effective that RGB to detect splice manipulations within images.

Xia et al. [34], proposed an algorithm to recognize counterfeit within fingerprints images. To obtain the required features to create the input vector for the classifier, Xia et al.'s method uses the discrete wavelet transform (DWT) and the local binary pattern (LBP). The precision achieved by the experiments was up to 92%. The dataset used was the "LivDet" [22].

Alahmadi et al. [1] introduced a technique based on discrete cosine transform (DCT) and the local binary pattern (LBP) to identify splicing and copy-move forgeries. The first step of the proposed algorithm is to change the space colour to the YCbCr. Next, split the Cb and Cr components into overlapping blocks. Then, apply LBP to each block; each block is converted into the DCT domain and extract the DCT coefficients to create the features vector for the SVM classifier. The results obtained by the algorithm show a precision of 97.77%. The dataset used was "CASIA TIDE v2.0".

2.4 Copy-move forgery detection techniques

The copy-move technique is another popular method used today for image forgery, where a region of an image is used to hide another region from the same image. The existence of two identical regions is not ordinary in natural images; thus, this property can be used to detect this type of manipulations. Even after applying some post-processing processes, such as edge smoothing, blurring, and adding noise to eliminate visible traces of manipulation, there will be two extremely similar regions in the manipulated image.

In the literature a large number of copy-move forgery detection methods have been proposed. Nevertheless, all of these methods can be classified into two main categories: block-based and keypoint-based methods [26, 31]. Of all those, one of the most used to detect copy-move forgery is the method that use a block matching algorithm. In this algorithm, the image is divided into overlapping blocks, and the blocks are compared to find the duplicated region.

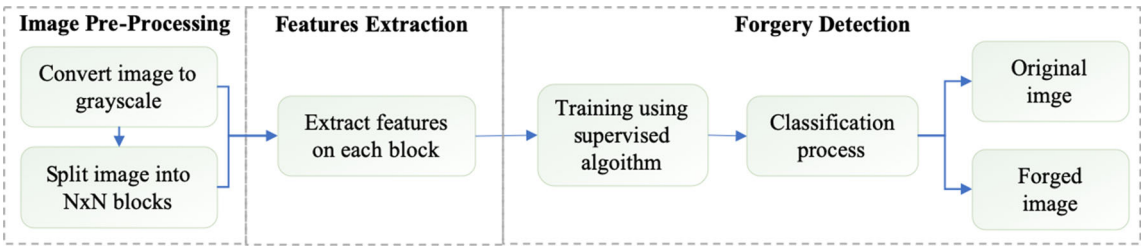


Fig. 2 Process diagram of splicing detection techniques

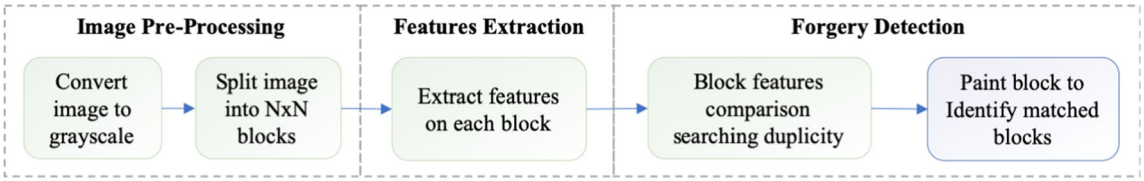


Fig. 3 Diagram of copy-move forgery detection techniques

Figure 3 shows a general scheme of the block matching algorithm.

Fridrich et al. [16] proposed a method based on the discrete cosine transform (DCT) to identify copy-move forgery. The method split the image into overlapping blocks of 16×16. Then, the DCT coefficient characteristics are extracted from each block and then these coefficients are classified lexicographically. After the lexicographical classification, comparable squares are distinguished, and the duplicated regions are found. Fridrich et al. introduced one of the first techniques that use DCT to identify copy-move forgeries on images.

Popescu et al. [27] introduced a technique to recognize duplicate regions within images. Popescu's algorithm employs principal components analysis (PCA) rather than DCT. The algorithm uses PCA on small fixed-size image blocks, and then each block is lexicographically ordered. This method has proved great efficiency to recognize copy-move forgeries.

Kang et al. [20] used singular value decomposition SVD to distinguish the modified areas in a picture. By applying SVD, a feature vector is extracted, and the dimensions reduced. Then, identical blocks were identified by the use of a lexicographic classification. Kang's method demonstrated to be robust and effective. The results of the experiment prove the efficacy of the method.

Huang et al. [19] introduced a method to identify copy-move manipulation over digital images applying SIFT algorithm. The authors showed the SIFT calculation algorithm using the block matching function. This method gives great results even when the image is noisy or compressed.

In [7] a scheme based on speeded up robust features (SURF) was proposed, which have key point characteristics better than SIFT because they work better with post-processing techniques such as brightness and blur variations. However, the methods based on key points present a problem of visual output because the copied and pasted regions consist of lines and points that do not show a clear and intuitive visual effect.

Amerini et al. [3], proposed a method based on SIFT. The proposed method can identify copied regions in images. Also, the method proposed can detect which geometric transformation was applied. Due to the copied region of the image looks the same as the original, the key points extracted in the duplicated region will be identical to those in the original. This method is also useful with low-quality factor compressed images.

Table 1 presents a summary of the copy-move detection techniques analysed by comparing their results in terms of accuracy.

3 Proposed image authentication scheme

In this work, an improved algorithm for copy-move forgery detection is proposed. The algorithm is based on the technique introduced by Fridrich [16]. A diagram presenting the main processes of the detection algorithm can be found in Fig. 4. Further details will be provided along the rest of the section.

Summary detection algorithm are described below:

1. Transform the image to grayscale.
2. Divide the image into small overlapping blocks of size $B \times B$ from top-left to bottom-right.
3. Compute the DCT transformation of every block, sort the coefficients in a zig-zag fashion and truncate the list to contain the first k elements.
4. Make a lexicographic sort of the truncated coefficient lists, and for each list, compute a similitude measure between its nearest neighbours. If the similarity is lower than the threshold, blocks are considered as identical.
5. For every pair of identical blocks, the translation vector is computed. If the number of vectors in a given direction exceed a predefined quantity, every block is considered as part of the copy-move tampering.

The block size and thresholds chosen for the algorithm should be dependent mainly on the size of the image and the expected size of the modification. Some recommendation for the parameter selection will be given according to results obtained in the experiments detailed in Sect. 4. For now, let us explore each step of the aforementioned technique in detail.

The input parameters and the expected output after algorithm execution are the following:

- INPUT: Suspicious image I.

Table 1 Comparison of copy-move forgery detection techniques

Work	Used method	Observations	Accuracy
[16]	DCT coefficients and lexicographic classification	Robust to image retouching	Not available
[27]	PCA, *Eigen* values and lexicographical classification applied	Good results against compressed or noisy images	70.97%
[20]	SVD and lexicographic classification	Validity against blur, noise and compression filters	Not available
[19]	SIFT calculation algorithm using the block matching function	Good results against compressed or noisy images	Not available
[7]	Based on SURF feature descriptors	Works well with post-processing techniques such as brightness and blur variations	Not available
[3]	Extraction of key points with SIFT algorithm	Effective in compressed images with a low quality factor	93.42%
[37]	DCT coefficients and singular value decomposition (SVD) features	Effective over Gaussian blurring, additive white Gaussian noise, and JPEG compression	92%
[26]	Extraction of key points with SIFT algorithm	Effective over several post-processing transformations such as rotation, scaling, JPEG compression, and additive white Gaussian noise	80%

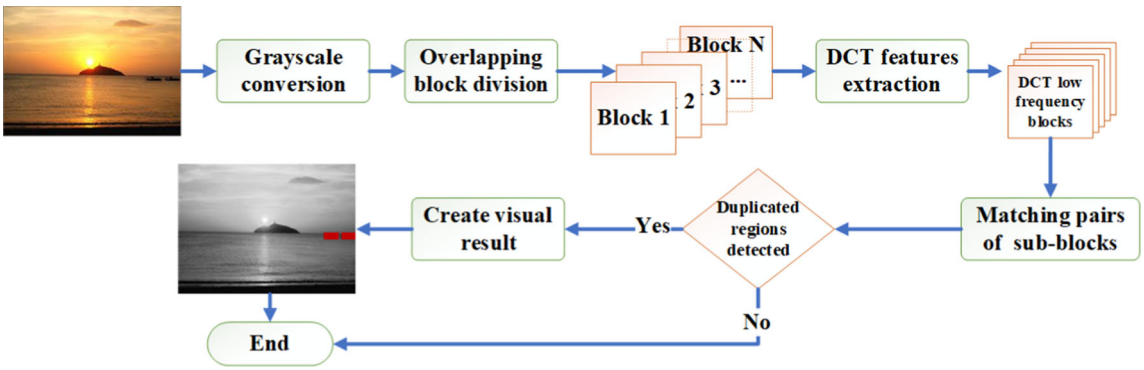


Fig. 4 Processes of the copy-move detection algorithm

- OUTPUT: greyscale image with the original and probably tampered blocks painted in black.

First, images are commonly regarded as a combination of the red, green and blue (RGB) channels. These values are commonly provided by Bayer filters in many image devices. The grayscale image, also known as *luminance*, is obtained by combining the RGB components according to the following linear transformation.

$$Y = 0.2125R + 0.7154G + 0.0721B \quad (1)$$

Subsequently a blocks size B is established to obtain the overlapping division of the image in blocks. Starting with the top-left corner, subsequent blocks are gathered by sliding left one pixel. Starting with the top-left corner, subsequent blocks are gathered by sliding left one pixel. Once extracted all these blocks, the process continues with one pixel down. The total number of blocks for an image of size $M \times N$ at the end of this process must be $(M - B + 1)(N - B + 1)$. Good results have been obtained with $B = 8$, which is the default value considered in what follows.

The next step consists in applying the DCT transform on every block to obtain a list of coefficients. DCT is chosen since many of the coefficients have values near 0, specifically, those corresponding to highest frequencies, located near the bottom right corner of the coefficient matrix [17]. This reason leads to sort the coefficients following a zig-zag pattern as it has been mentioned before, as shown in Fig. 5.

The zig-zag sorted lists are now truncated to k elements. The value of k is assigned according to the block size B (bigger blocks should consider more elements and vice versa). A truncation factor $0 < f_t < 1$ is fixed to compute k as shown in Eq. (2).

$$k = \left\lfloor f_{\mathrm{t}} B^2 \right\rfloor \tag{2}$$

To accelerate the sorting process, only small integer values (k) will be considered. After truncation, the remaining values are quantized using a quantization factor f_{q}. Quantization is achieved by first dividing every value of the list by f_{q} and rounding the result. This process is specified in Eq. (3). Values $a_{i1}, \ldots, a_{ik}$ denote the original coefficients of the i-th truncated list.

$$\mathbf{a_i} = \left(\left\lfloor \frac{a_{i1}}{f_{\mathrm{q}}} \right\rfloor, \left\lfloor \frac{a_{i2}}{f_{\mathrm{q}}} \right\rfloor, \ldots, \left\lfloor \frac{a_{ik}}{f_{\mathrm{q}}} \right\rfloor \right) \tag{3}$$

With the truncated and quantized lists of coefficients, the next step is to sort them in lexicographic order. With this process, it is expected that very similar blocks provide similar quantized coefficient vectors.

After sorting, a matrix is produced with rows corresponding to similar blocks close together, but a similarity measure will indicate whether two blocks are duplicate of each other. The precise process to decide if two blocks are the same is provided by the following steps:

- Let $A = (a_{ij})$ be the sorted matrix of coefficients and $\mathbf{a_i}$ the i-th row of A. The first step is to define N as the maximum number of rows to be compared with $\mathbf{a_i}$. This is, $\mathbf{a_i}$ will be compared with $\mathbf{a}_{i+l}$ for $l = 1, \ldots, N$.
- Next, decide whether two blocks are identical using the pseudocode defined as Algorithm 1.

Algorithm 1 Pseudocode to decide when two blocks are identical

```
r_max ← −∞
r_min ← ∞
c ← 0
for l = 1, ..., k do
    if a_lj = 0 and |a_li| > 0 then
        c ← c + 1
    else
        r ← a_il / a_jl
        if r > r_max then
            r_max ← r
        if r < r_min then
            r_min ← r
if r_max − r_min > 0 then
    c → c + 1
```

- After this process is completed, verify if $c = 0$ In this case a_i and a_j are marked as a copy.

Since contiguous blocks are generally very similar, we should discard marking them as possible copies by selecting only blocks that are far enough. For this, we consider the coordinates of the upper left pixel of a pair of similar blocks, namely, $v_i = (x_i, y_i)$ and $v_j = (x_j, y_j)$, corresponding to a_i, a_j, respectively. If the Euclidean distance of v_i and v_j, computed following Eq. (4), exceeds a given distance threshold (T_d), the blocks are considered for the subsequent steps of the process, otherwise they are discarded.

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} > T_{\mathrm{d}} \tag{4}$$

The final step of the detection algorithm consists in computing the frequencies for the transference vectors of every pair of suspicious blocks. The transference vector of two blocks starting at positions (x_i, y_i) and (x_j, y_j), respectively, is $\mathbf{v} = (x_i - x_j, y_i - y_j)$. If the frequency of a given transference vector is high, it is more likely that a big zone of the image has been displaced to the same region. If this frequency exceeds a given frequency threshold (T_{f}), all the pairs of suspicious blocks associated with the transference vector, both blocks are marked.

If the analysis is successful, a well-defined area must be perfectly visible, showing the original and copied regions. Unfortunately, the original region is not distinguished from the copied region. When false positives are found, it is easy to determine that the analysis has been successful, since the blocks do not show a well-defined area.

4 Experimental results

Throughout this section, all the experiments that have been carried out to evaluate the effectiveness of training-based manipulation identification algorithms will be described.

4.1 Datasets

The Image Manipulation Dataset [9] (hereinafter referred as **D1**) is a ground truth database for benchmarking the detection of image tampering artifacts. It includes 48 base images, separate snippets from these images, and a

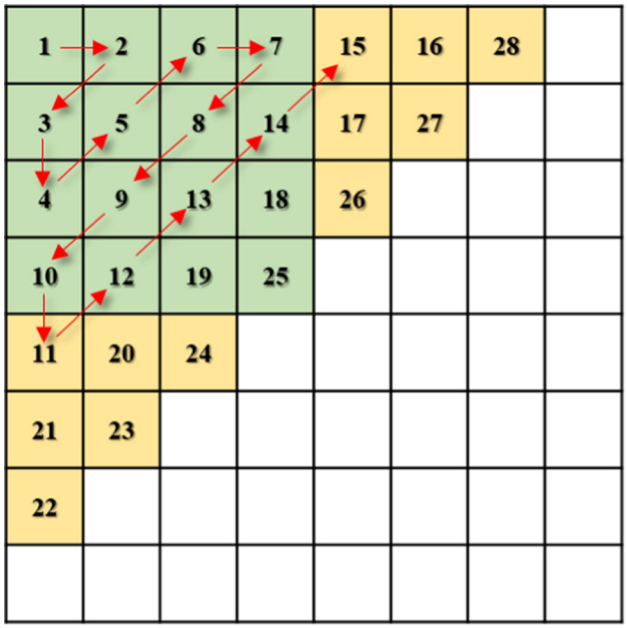


Fig. 5 Zig-zag sorting of DCT coefficients

software framework for creating ground truth data. The idea is to "replay" copy-move forgeries by copying, scaling and rotating semantically meaningful image regions.

The CMFD GRIP Dataset by Cozzolino et al. [11] (hereinafter referred as **D2**) is a dataset composed by 80 images, with realistic copy-move forgeries. All these images have size 768×1024 pixels, while the forgeries have arbitrary shapes, aimed at obtaining visually satisfactory results.

The CoMoFoD database [32] (hereinafter referred as **D3** and **D4**) has 260 image sets, 200 images in small image category (512×512), and 60 images in large image category (3000×2000). In both categories, following transformations are applied:

- *Translation* a copied region is only translated to the new location without performing any transformation,
- *Rotation* a copied region is rotated and translated to the new location,
- *Scaling* a copied region is scaled and translated to the new location,
- *Distortion* a copied region is distorted and translated to the new location

The distortion added to the dataset's images can be noise adding, image blurring, brightness change, colour reduction, contrast adjustments or the combination of two or more distortions on a copied region before moving it to the new location.

Ardizzone et al. [4] make a copy-move forgery dataset (hereinafter referred as **D5**) which contain a medium sized images, almost all 1000×700 or 700×1000. This dataset contains 50 not compressed images with simply translated copies and 46 not compressed images with 11 different types of rotation around the angle zero in the range of $[-25°, 25°]$ with step 5° and 11 scaling factors in the range of [0.75, 1.25] with step 0.05.

The CMH dataset (hereinafter referred as **D6**) was created by [29] and comprises 108 realistic cloning images. Each image is stored in the PNG format (which does not modify pixel values), and has a resolution varying from 845×634 pixels (the smallest) to 1296×972 pixels (the biggest). The dataset contains four groups of images:

- 23 images where the cloned area was only copied and then moved (simple case);
- 25 images with a rotation of the duplicated area (orientations in the range of 90° and 180°);
- 25 images with a resizing of the cloned area (scaling factors between 80 and 154%);
- 35 images involving rotation and resizing altogether.

4.2 Experiments setup

In all the experiments carried out, *Python* has been used as a programming language, due to its great flexibility to perform data analysis. For the evaluation of the proposed algorithm, the experiments carried out in this paper used the public datasets described before and Table 2 shows the main features of each dataset.

The characteristics of the equipment in which the experiments were carried out are presented in Table 3. These are an essential factor to take into account since the execution times of the different tests vary according to the resources available.

4.3 Experiments

This section will show the experiments that have been performed to evaluate the effectiveness of the algorithm for identifying the duplicated region with copy-move techniques. Throughout the tests carried out, it has been possible to verify that the algorithm works with any format, such as JPEG, PNG, BMP, among others. It should also be noted that the image size does not influence the accuracy of the results; it only produces variations in the processing time, as shown in Sect. 4.3.3.

4.3.1 Truncation factor evaluation

This experiment assesses and verifies the effectiveness of the proposed algorithm. This algorithm makes use of different configurable parameters, depending on the assigned value, the results may vary significantly. In [36] proposed a copy-move forgery detection algorithm based on DCT,

which produced excellent results in the identification of copy-move manipulations. To perform their experiments, they made comparisons between the parameters used by other investigations. The values established by them are used in this research to initialize the parameters of the algorithm. Table 4 shows each of the parameters used and their corresponding values.

The parameter that improves the results is the frequency threshold or T_f. This parameter sets the value under which a block of the image can be considered a valid manipulation. If a block appears several times in the image as a duplicate, and the frequency of appearance exceeds the one established by the threshold T_f, it will be considered as part of the manipulation. Because the image is segmented into overlapping blocks, it is possible to analyse the frequency of appearance of the duplicated blocks.

When T_f is high, the final results are more refined, removing the areas identified as manipulated that are false positives. In the experiment, the parameter T_f is set to three values: 50, 100, and 150. Figure 6 shows the results of the detection for these three T_f values.

From Fig. 6, it is evident that at a higher value of T_f, the results present less noise, that is, the black areas that are not part of the forgery. In the first image, the manipulation is identified at $T_f = 50$ (Fig. 6c), with a higher value, the algorithm does not find any duplicate block that meets the frequency of appearance established by T_f. On the contrary, the other two images show that noise produced by false positives is removed when $T_f > 50$. This difference happens because, in both images, the forged areas occupy a significant proportion of the image so the frequency of appearance of the duplicated blocks would be much higher than the overlap.

However, the algorithm fails with a specific type of manipulation. For example, when parts of a duplicated block are modified from the initial block as it is shown in Fig. 7. In Fig. 7a, the tree located in the central part has been duplicated. This tree has gaps between the branches which have been edited in the duplication so that it integrates perfectly with the background, that is why the algorithm treats both trees as different objects and is not able to detect the forgery.

Table 3 Features of the experimentation equipment

Resources	Features
Operating system	Ubuntu 17.04
Memory RAM	12 GB
Processor	Intel®Core™ i5-6200U CPU @ 2.30GHzx4
Graphic card	Intel®HD Graphics 520 (Skylake GT2)
Storage	64 GB

4.3.2 Texture influence on the success rate

In the second experiment, we checked the accuracy of the algorithm to detect the copy region on textures with similar patterns. In this type of images, the manipulation goes unnoticed due to its excellent integration with the original background as is the case of images with the same colour pattern. Two tests were carried out with this type of images, and the parameter T_f was adjusted to the value 150 to reduce the noise of black dots in the results.

In the first test, images with multiple colours and details but similar patterns were used; this makes the duplicated area difficult to detect. Three examples of identification in this type of images are shown in Fig. 8. As noted, the algorithm achieves remarkable accuracy.

For the second test, we used images where the duplicated region was moved to an area with the same colour as other regions of the image. In this type of images, it is also difficult to detect the duplicated region since it can be confused with another original region that has the same colour. Figure 9 shows three examples where the algorithm has an excellent performance in this type of manipulation.

4.3.3 Image resolution influence

In this experiment, we analysed the efficiency of the algorithm in large and high-resolution images. We

Table 2 Used dataset's features

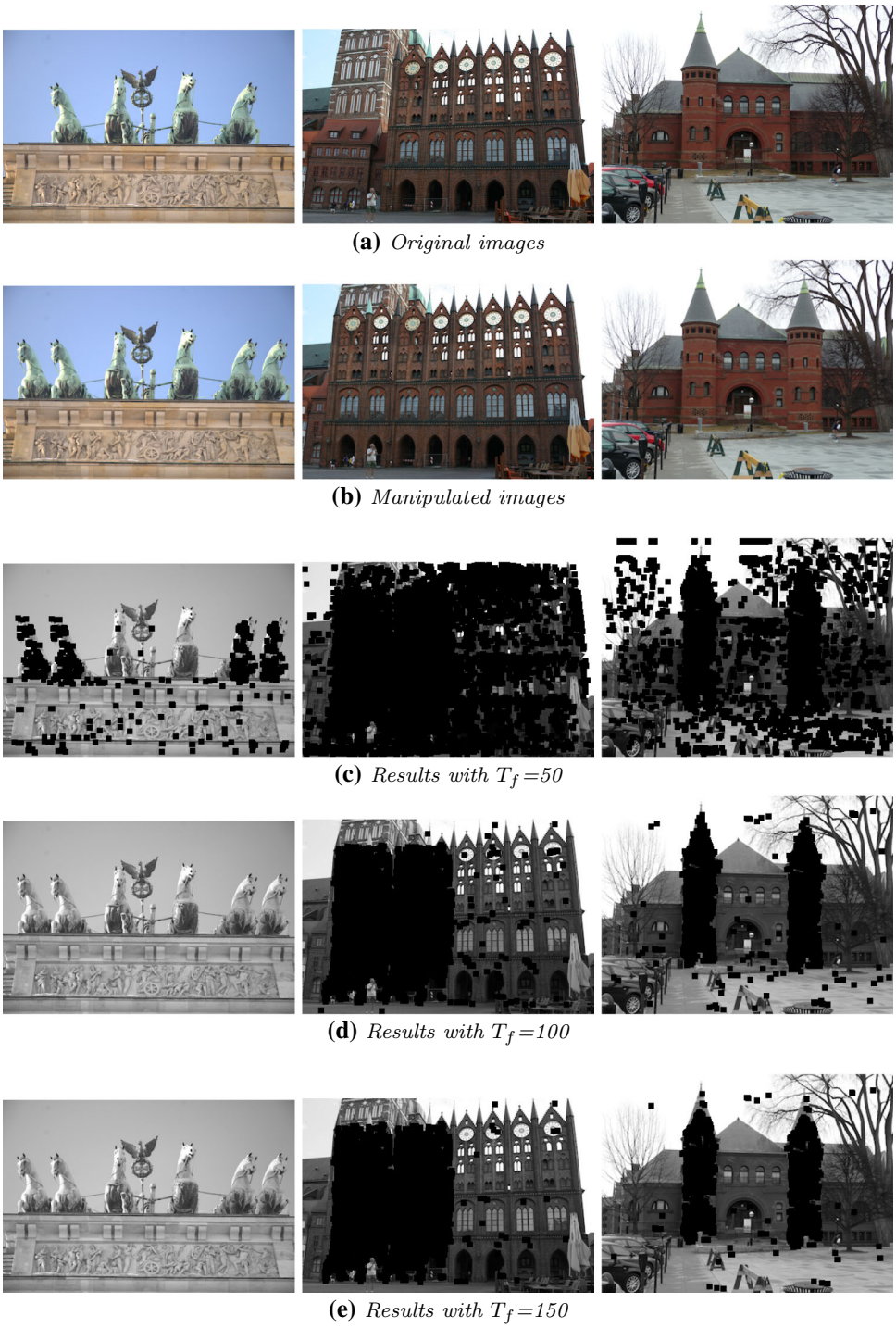
Datasets	No. images	Resolutions	Transformations	Formats
D1 [9]	48	2362×1581, 3888×2592	None	jpg/png
D2 [11]	80	1024×768	None	png
D3 [32]	960	3000×2000	Rotation, scaling, distortion	png/jpg
D4 [32]	200	512×512	Rotation, scaling, distortion	png/jpg
D5 [4]	96	1024×768	Rotation, scaling, distortion	bmp
D6 [29]	108	845×634, 1296×972	Rotation, scaling, distortion	png

Table 4 Configurable parameters of the copy-move algorithm

Parameter	Name	Assigned value
f_t	Truncation factor	0.25
f_q	Quantification factor	4
N_a	Comparable neighbouring rows	3
T_f	Frequency threshold	50
T_d	Distance of vectors	20

observed that when scaling an image to a smaller size, the accuracy of the algorithm remains high without undergoing significant changes. Thus, allowing to perform scaling of large images before the algorithm processes them, which increases efficiency without losing quality in the results.

Figure 10 shows an example of a modified image by the copy-move forgery. In this image, the bird above the grass has been copied and placed on the cow's head. The original



(a) *Original images*

(b) *Manipulated images*

(c) *Results with* $T_f = 50$

(d) *Results with* $T_f = 100$

(e) *Results with* $T_f = 150$

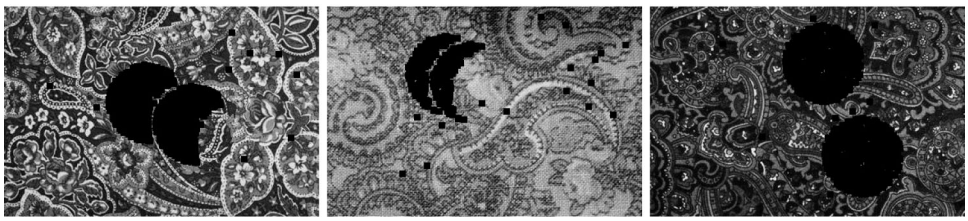
Fig. 6 Detection of copy-move manipulations

(a) *Original Image* **(b)** *Tampered Image* **(c)** *Results with* $T_f = 50$ **(d)** *Results with* $T_f = 100$ **(e)** *Results with* $T_f = 150$

Fig. 7 Duplicate area with details of the original image

(a) *Original Images*

(b) *Tampered Images*



(c) *Detection results*

Fig. 8 Copy-move detection over images with similar patterns textures

image size is 1080×854 pixels (c), and the size of the scaled image is 640×427 pixels. The execution time it took for the algorithm to process the original image was 160 s, while the scaled image took 48 s. From Fig. 10, it is apparent that the manipulation has been correctly detected in both images, so it is possible to perform the scaling without affecting the accuracy of the algorithm and considerably improving the execution time.

4.4 Experiment 4

In order to test our algorithm, we use the six datasets described before, and the metrics used to quantify its accuracy were the precision, recall, and F1 scores. The precision, P, is the ratio of the probability that a detected region is accurate, and the formula to calculate the precision is the following:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{5}$$

where TP is the number of true positives pixels and FP the number of false positives pixels detected by the algorithm.

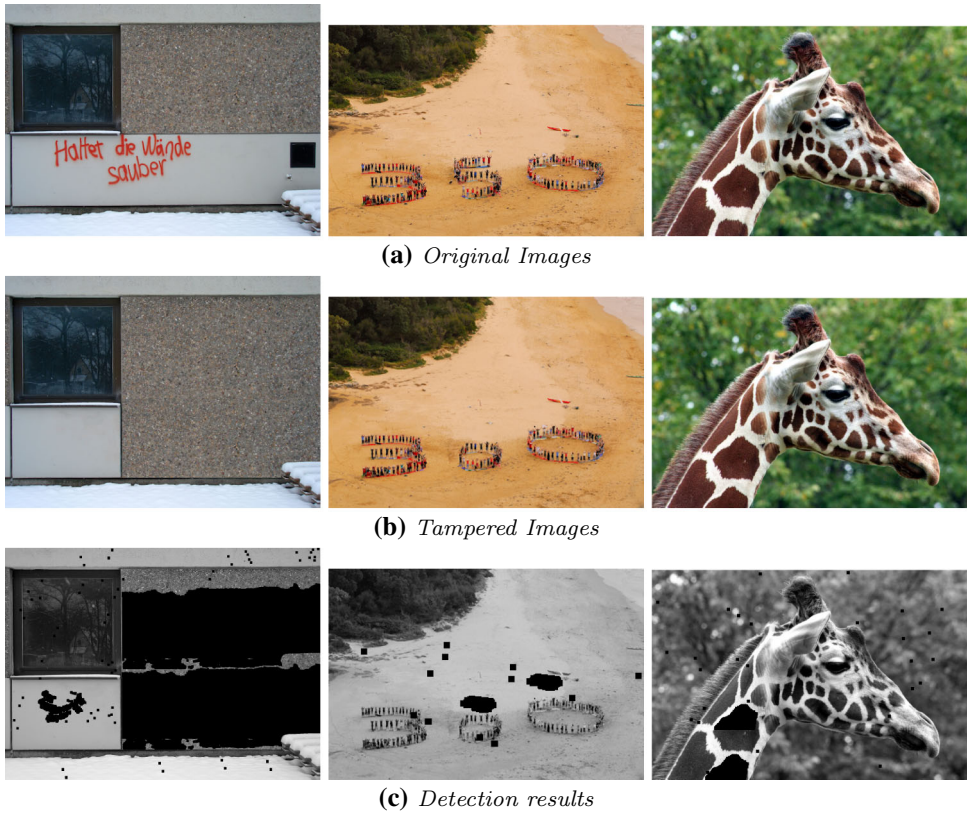
The recall is the True Positive Rate (TPR) component. These are given by the recall, R, is the true positive ratio which measure the ability of the algorithm to find all the positive samples, its formula is the following:

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{6}$$

where TP is the number of true positives and FN the number of false negatives

The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:

$$F1 = \frac{2 * (P * R)}{P + R} \tag{7}$$



(a) *Original Images*

(b) *Tampered Images*

(c) *Detection results*

Fig. 9 Copy-move detection over images with areas of the same colour (colour figure online)

(a) *Original Image* **(b)** *Tampered Image* **(c)** *Image of* 1280×854 **(d)** *Image of* 640×427

Fig. 10 Copy-move identification in scaled images

Table 5 Results obtained by proposed algorithm

Datasets	Precision (%)	Recall (%)	F1 (%)
D1 [9]	95.51	69.74	78.7
D2 [11]	90.61	58.71	70.31
D3 [32]	95.35	70.99	79.51
D4 [32]	86.66	57.25	67.06
D5 [4]	96.58	**81.06**	**87.94**
D6 [29]	**97.52**	27.49	42.52
Average	93.71	60.87	71.01

Number in bold are the best result

where P is the precision and R the recall obtained by the algorithm over each analysed image.

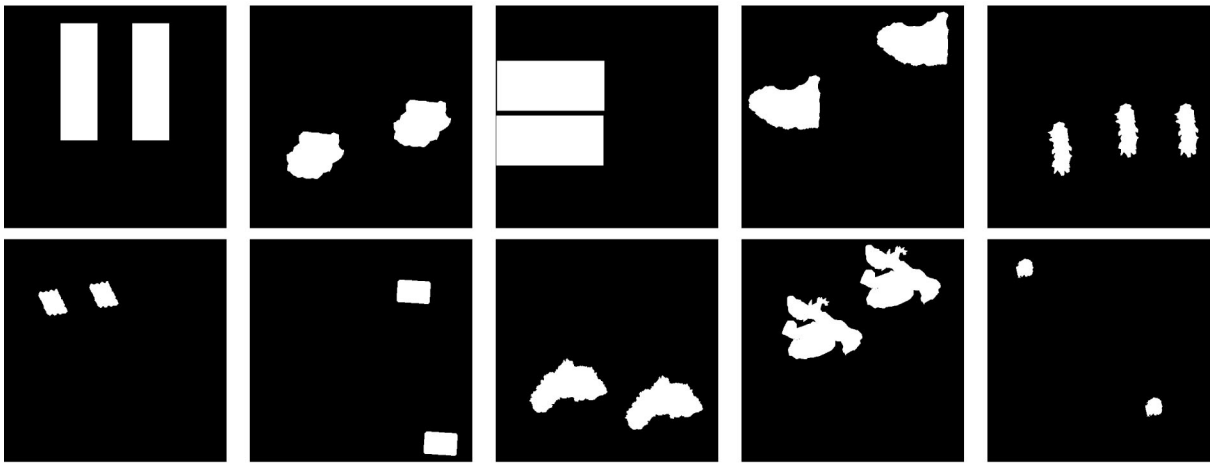
The results summarized in Table 5 shows a high accuracy precision and recall over all the tested datasets, specially the datasets **D5**. Our proposed algorithm gets a precision average over the 93% even over images that contain distortion, such as an increase or decrease of brightness and/or contrast, and small geometrical transformations like slight degree rotation.

To validate our results, it is essential to compare our algorithm to a related method, such as the one proposed by Alkawaz et al. [2] in which the authors get a 96% of recall and a 64.52% of precision using a block size of 8×8. Figure 11 and Table 6 show the outputs and the results and the outputs.

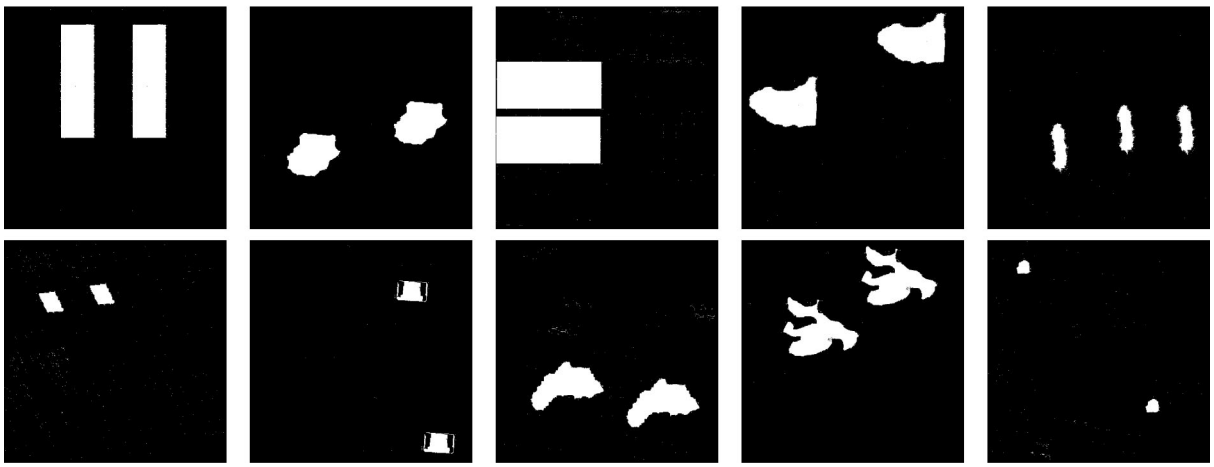
040.png 029.png 024.png 015.png 027.png

016.png 017.png 013.png 028.png 011.png

(a) *Tampered Images*



(b) *Ground Truth*



(c) *Detection results*

Fig. 11 Copy-move detection over images with areas of the same colour (colour figure online)

Table 6 Results obtained by proposed algorithm

Images	Our algorithm		[2]	
D4 [32]	Precision (%)	Recall (%)	Precision (%)	Recall (%)
040.png	99.93	86.95	**100**	97.53
029.png	**99.96**	80.58	95.19	96.25
024.png	99.59	**91.31**	49.80	**100**
015.png	99.41	85.22	87.62	99.48
027.png	98.71	63.07	63.32	88.47
016.png	91.78	67.07	62.53	97.3
017.png	99.35	55.15	59.51	99.86
013.png	99.01	81.19	59.25	98.68
028.png	99.86	71.79	41.49	93.37
011.png	91.29	58.11	26.58	94.85
Average	97.88	74.04	64.53	96.58

Number in bold are the best result

5 Conclusions

As at the beginning of this paper says the famous saying goes, "A picture is worth a thousand words". Therefore, having faster and reliable algorithms to analyse the integrity of an image is needed. Nowadays, thanks to the fast and easy way to share images plus the easiness of use professional image editing tools make it harder to detect forgeries.

During the development of this work, experiments were performed using the proposed algorithm against six different datasets widely used in the literature. This group of images contained different types of formats, sizes, and additional transformations to the copy-move manipulation. The results obtained by the algorithm proposed in this work showed an precision of more than 97% on the dataset **D6** [29] and the overall average on all the datasets used was 93.71%. In addition, the proposed algorithm demonstrated a high accuracy in the analysis of images with additional transformations, similar algorithms have many difficulties to identify the copy-move manipulation in images with geometric transformations and filters. On the other hand, we compared our algorithm with the one proposed by Alkawaz et al. [2] and used the same group of images as the authors of [2]. Our algorithm showed an average precision of 97.88% compared to 64.53% obtained in [2].

In this work, an exhaustive study on existing forgery detection techniques has been carried out, emphasising on copy-move detection. Also, a new approach for forgery detection was presented. The experiments carried out with the proposed algorithm have shown their robustness and efficiency in the results obtained. The algorithm can detect and locate with high precision the duplicate zone in the image. Besides its accuracy, the algorithm proved been a fast method to analyse even high-resolution photographs in a short time.

Acknowledgements This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 700326. Website: http://ramses2020.eu

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Alahmadi A, Hussain M (2017) Passive detection of image forgery using DCT and local binary pattern. Signal Image Video Process 11(1):81–88. https://doi.org/10.1007/s11760-016-0899-0
2. Alkawaz MH, Sulong G, Saba T, Rehman A (2018) Detection of copy-move image forgery based on discrete cosine transform. Neural Comput Appl 30(1):183–192
3. Amerini I, Ballan L, Caldelli R, Del Bimbo A, Serra G (2011) A SIFT-based forensic method for copy-move attack detection and transformation recovery. IEEE Trans Inf Forensics Secur 6(3):1099–1110
4. Ardizzone E, Bruno A, Mazzola G (2015) Copy-move forgery detection by matching triangles of keypoints. IEEE Trans Inf Forensics Secur 10(10):2084–2094
5. Bharati A, Singh R, Vatsa M, Bowyer KW (2016) Detecting facial retouching using supervised deep learning. IEEE Trans Inf Forensics Secur 11(9):1903–1913. https://doi.org/10.1109/TIFS.2016.2561898
6. Birajdar GK, Mankar VH (2013) Digital image forgery detection using passive techniques: a survey. Digit Investig 10(3):226–245
7. Bo X, Junwen W, Guangjie L, Yuewei D (2010) Image copy-move forgery detection based on SURF. In: 2010 international conference on multimedia information networking and security. Nanjing, China, pp 889–892. https://doi.org/10.1109/MINES.2010.189
8. Chen M, Fridrich J, Goljan M, Lukas J (2008) Determining image origin and integrity using sensor noise. IEEE Trans Inf Forensics Secur 3(1):74–90. https://doi.org/10.1109/TIFS.2007.916285
9. Christlein V, Riess C, Jordan J, Riess C, Angelopoulou E (2012) An evaluation of popular copy-move forgery detection approaches. IEEE Trans Inf Forensics Secur 7(6):1841–1854
10. Columbia University: Columbia DVMM image splicing datasets (2011). http://www.ee.columbia.edu/ln/dvmm/newDownloads.htm
11. Cozzolino D, Poggi G, Verdoliva L (2015) Efficient dense-field copy-move forgery detection. IEEE Trans Inf Forensics Secur 10(11):2284–2297
12. Ding F, Zhu G, Dong W, Shi YQ (2018) An efficient weak sharpening detection method for image forensics. J Vis Commun Image Represent 50:93–99
13. Ding F, Zhu G, Yang J, Xie J, Shi Y (2015) Edge perpendicular binary coding for USM sharpening detection. IEEE Signal Process Lett 22(3):327–331. https://doi.org/10.1109/LSP.2014.2359033
14. Dong J, Wang, W (2019) CASIA TIDE v1.0 - v2.0. http://forensics.idealtest.org/
15. Freeman L (2017) Digital evidence and war crimes prosecutions: the impact of digital technologies on international criminal investigations and trials. Fordham Int Law J 41(2):283

16. Fridrich J, Soukal D, Lukas J (2003) Detection of copy move forgery in digital images. In: Proceedings of the digital forensic research workshop. Binghamton, New York, pp 5–8
17. Fu Q, Zhou X, Wang C, Jiang B (2016) Mathematical relation between APBT-based and DCT-based JPEG image compression schemes. J Commun 11:84–92
18. Huang F, Huang J, Shi YQ (2010) Detecting double JPEG compression with the same quantization matrix. IEEE Trans Inf Forensics Secur 5(4):848–856. https://doi.org/10.1109/TIFS.2010.2072921
19. Huang H, Guo W, Zhang Y (2008) Detection of copy-move forgery in digital images using SIFT algorithm. In: 2008 IEEE Pacific-Asia workshop on computational intelligence and industrial application, vol 2, pp 272–276. https://doi.org/10.1109/PACIIA.2008.240
20. Kang X, Wei S (2008) Identifying tampered regions using singular value decomposition in digital image forensics. In: 2008 international conference on computer science and software engineering, vol 3, pp 926–930. https://doi.org/10.1109/CSSE.2008.876
21. Kee E, Farid H (2011) A perceptual metric for photo retouching. Natl Acad Sci 108(50):19907–19912. https://doi.org/10.1073/pnas.1110747108
22. Listverse: Top 15 photoshopped photos that fooled us all (2007) http://listverse.com/2007/10/19/top-15-manipulated-photographs/
23. Mahdian B, Saic S (2010) A bibliography on blind methods for identifying image forgery. Signal Process Image Commun 25(6):389–399
24. Malin DF (1977) Unsharp masking. Am Astron Soc Photo Bull 16:10–13
25. Niu Y, Li X, Zhao Y, Ni R (2019) An enhanced approach for detecting double JPEG compression with the same quantization matrix. Signal Process Image Commun 76:89–96
26. Park CS, Choeh JY (2018) Fast and robust copy-move forgery detection based on scale-space representation. Multimed Tools Appl 77(13):16795–16811
27. Popescu AC, Farid H (2004) Exposing digital forgeries by detecting duplicated image regions. Department of Computer Science, vol 646
28. Shi YQ, Chen C, Chen W (2007) A natural image model approach to splicing detection. In: Proceedings of the 9th workshop on multimedia and security. Dallas, Texas, pp 51–62. https://doi.org/10.1145/1288869.1288878
29. Silva E, Carvalho T, Ferreira A, Rocha A (2015) Going deeper into copy-move forgery detection: exploring image telltales via multi-scale analysis and voting processes. J Vis Commun Image Represent 29:16–32
30. da Silva EAB, Mendonça GV (2005) 4—digital image processing. In: Chen WK (ed) The electrical engineering handbook. Academic Press, Burlington, pp 891–910
31. Teerakanok S, Uehara T (2019) Copy-move forgery detection: a state-of-the-art technical review and analysis. IEEE Access 7:40550–40568. https://doi.org/10.1109/ACCESS.2019.2907316
32. Tralic D, Zupancic I, Grgic S, Grgic M (2013) Comofod–new database for copy-move forgery detection. In: Proceedings ELMAR-2013. IEEE, pp 49–54
33. Wang W, Dong J, Tan T (2010) Image tampering detection based on stationary distribution of Markov chain. In: 2010 IEEE international conference on image processing. Hong Kong, China, pp. 2101–2104. https://doi.org/10.1109/ICIP.2010.5652660
34. Xia Z, Yuan C, Sun X, Sun D, Lv R (2016) Combining wavelet transform and LBP related features for fingerprint liveness detection. IAENG Int J Comput Sci 43(3):290–298
35. Zhang Z, Kang J, Ren Y (2008) An effective algorithm of image splicing detection. In: 2008 international conference on computer science and software engineering, vol 1, pp 1035–1039. https://doi.org/10.1109/CSSE.2008.1621
36. Zhang Z, Wang D, Wang C, Zhou X (2017) Detecting copy-move forgeries in images based on DCT and main transfer vectors. KSII Trans Internet Inf Syst 11:4567–4587
37. Zhao J, Guo J (2013) Passive forensics for copy-move image forgery using a method based on DCT and SVD. Forensic Sci Int 233(1):158–166
38. Zhao X, Li J (2011) Detecting digital image splicing in chroma spaces. In: Digital watermarking. Springer, Berlin, vol 6526, pp 12–22