




# Copy-move forgery detection using local tetra pattern based texture descriptor

Sagnik Ganguly<sup>1</sup> · Sanmit Mandal<sup>1</sup> · Samir Malakar<sup>2</sup>  · Ram Sarkar<sup>1</sup>

Received: 4 May 2021 / Revised: 8 June 2022 / Accepted: 3 December 2022 /

Published online: 18 January 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

In modern era it has become increasingly easier to manipulate and tamper digital images, one of the primary reasons being the boon of commonplace availability of powerful image editing tools and software. These tools become a bane when used for malicious reasons as users can possibly add or remove important features from an image without leaving any obvious marks of tampering. Hence the need of forgery detection techniques which show high accuracy in detection arises. One of the most prevalent forms of image tampering is the copy-move forgery attack. In this type of forgery, a part of an image is copied and then pasted somewhere else in the same image with the intent to hide key features of the image. This paper introduces a new copy-move image forgery detection technique which relies on a texture feature descriptor called Local Tetra Pattern (LTrP) for block level image comparison used to localize tampered region(s). Initially, the input image is divided into overlapping blocks, then LTrP features are extracted from each block to form a single feature vector. Next, the feature vectors of all image blocks are sorted lexicographically, and then similar blocks are identified by matching the features from neighboring blocks. Finally, blocks matched falsely due to the presence of homogeneous color information like sea, field, and sky are removed using a shift vector aided outlier removal method. Experiments have been conducted on two standard datasets - GRIP and CoMoFoD. We have obtained 0.9834 and 0.9093 average  $F_1$  scores at pixel-level for GRIP and CoMoFoD datasets respectively. The experimental results demonstrate that the proposed technique has been able to detect the forged regions with higher accuracy as compared to many state-of-the-art copy-move forgery detection methods. Moreover, experimental results on CoMoFoD dataset show that the method is able to correctly detect the forgery even after various post-processing attacks. The source code of proposed method is available at <https://github.com/RollingThunderSagnik/LTrP-Copy-Move-Forgery-Detection>.

**Keywords** Copy-move forgery · Digital image forensics · GRIP · CoMoFoD · Duplicated region detection · Local tetra pattern

---

✉ Samir Malakar  
malakarsamir@gmail.com

# 1 Introduction

In the past few years, digitally prepared multimedia contents like text, images, and videos are largely used for communication and conveying information in day to day lives. However, with the exponential rise of easy availability of handy digital devices and freely downloadable apps/software with build-in sophisticated manipulation techniques to modify digitally prepared contents specially images, in the past few years, tampering of images has become so easy that anyone with minimum digital literacy can forge a multimedia content [24, 31]. Consequently, digitally doctored images have now become increasingly common which give rise to serious threats especially in journalism and the entertainment industry, and sometimes matters of national safety. Fake news along with digitally forged images have disastrous effect on almost every sphere of life. Almost every passing day, with the advancement of technologies, the visual quality of synthetic multimedia contents is becoming close to the real ones. Consequently, identification of synthetic images by bare human eyes becomes difficult, which in turn, hampers their trustworthiness [24, 31]. Thus, it becomes a pressing need to develop a competent method to detect such images before they used to convey some information [23, 25].

Copy-move forgery is one of the most widespread forms of digital image forgery because it is very easy to create such forged images. With advancements in the image-editing software, images can now be forged in such a manner that the changes made in them cannot be distinguished by the naked eye. Copy-move forged images can be prepared using two approaches [36]: (i) intra-image copy-move and (i) inter-image copy-move or splicing. In the first approach, one(multiple) region(s) is(are) copied and pasted within the same image, while in the second approach, region(s) is(are) copied from one image and pasted into another image. It is worth mentioning here that a copied region may contain object(s), portion of some objects, and background etc. At times, such copy-move forgery may go unnoticed to the human eye. Hence, an efficient copy-move forgery detection (CMFD) technique is required to identify such type of forgery in digital images. Here we aim to design a new CMFD technique by discussing and analyzing the issues related to detection of forgery in digital images. In this paper, a block based scheme using the Local Tetra Pattern (LTrP) feature descriptor is proposed for the detection of intra-image forgeries.

## 1.1 Major contributions

First, this paper has analysed and summarized some of the key problems related to copy-move forgery detection. To overcome some of these issues, a novel approach based on texture analysis has been proposed in this paper. The LTrP feature descriptor due to its direction-sensitive property, is more robust than other feature descriptors in detecting copy-move forgery. To the best of our knowledge, this is the first time this feature descriptor has been applied in the field of image forgery detection. Second, the proposed method uses an efficient shift-vector based outlier removal technique, which greatly improves the localization accuracy of the final copy-move forged region(s). Third, the effectiveness of the proposed method has been analysed and demonstrated through rigorous testing and evaluation on standard datasets. The proposed method also yields promising results in both plain copy-move forgery as well as post-processed forged images. Performance comparison of the proposed method with various state-of-the-art methods has also been presented.

## 1.2 Organization of the paper

The rest of the paper has been organized as follows: Section 2 presents the related work and brief introduction of popular forgery detection techniques. In Section 3, the LTrP feature descriptor is briefly introduced, which lays the foundation of its application in the proposed CMFD method which is described in details in Section 4. In Section 5, the experimental results and comparative studies with the existing benchmark methods in this domain have been presented. In Section 6, the conclusion and future scope of the work has been discussed.

## 2 Related work

Many passive CMFD methods have been proposed to detect and localize tampered region(s). In CMFD solutions, researchers have mostly focused on extracting different features to improve the detection capability of their models. Thus, here we have categorized the existing methods into three main groups depending on the use of the features: (i) Key-point based methods, (ii) Block based methods, and (iii) Deep learning based methods. A key point based algorithm scans an image and extracts the high entropy regions which are then used to extract features. The extracted features are processed to identify the copied regions. In a block based algorithm, first an image is partitioned into many overlapping/non-overlapping blocks and then these blocks are further analysed to expose the forged region(s). To locate the forged regions(s), at first features are extracted from these blocks and then blocks with identical features are used to locate the forged area(s). A better feature means better identification of forged regions in this case. These two categories of methods, in general, extract hand-crafted features first and then detect the forgery in an unsupervised manner. Thus no training samples are needed. However, recently many researchers [6, 42] have provided deep learning based solutions. Before testing, these methods trained their network models with voluminous forged images on a resource heavy machine. A detailed discussion on these categories of methods is provided in the following subsections.

### 2.1 Keypoint based methods

Keypoints such as edges, corners and hinges are used for detecting the forgery. Pan et al. [34] designed a keypoint based technique where scale invariant feature transform (SIFT) method was used to detect the keypoints and subsequently forged pixels were detected. This keypoint method was employed to resist different types of geometric distortion. Amerini et al. [2] used the SIFT method with comprehensive analysis, and used hierarchical clustering and geometric transformation estimation in their proposed scheme. After the keypoint descriptors were extracted, they used an iterated generalized 2 nearest neighbor (g2NN) method for matching and finally to identify the forged area(s), hierarchical clustering was used. Later, in another work, Amerini et al. [3] used J-Linkage algorithm which performed robust clustering in the space of geometric transformation instead hierarchical clustering used in [2] on the extracted keypoint descriptors and obtained better performance. Costanzo et al. [12] proposed three novel forensic detectors based on anomalies in the distribution of interest points following manipulation, they could identify images whose SIFT keypoints had been

locally or globally removed. These methods are largely affected by the erroneous detection of matched keypoint pairs. Thus, performance of methods can be improved if better keypoint pairs are identified or better keypoints are extracted. Consequently, many researchers tried some alternative methods to extract better keypoints or to reduce the extracted keypoints intelligently to left with only salient keypoints.

Chen et al. [11] utilized Harris corner point detector to extract image keypoints, and collected the feature vectors by calculating the sector statistic of a small circular region around each Harris corner point. In [4], Ardizzone et al. presented a new CMFD method that analysed triangles of local keypoints generated by SIFT method instead of one used in other methods. Zandi et al. [45] presented a new interest point detector that automatically adjusts the density of interest points to make it more practical in the CMFD purpose and it was easier to generate suspicious regions. In another research work, Wang et al. [43] made use of scale-invariant feature detector with error resilience (SIFER) and fast quaternion radial harmonic Fourier moments (FQRHFM). In this method, the features were matched using coherency sensitive hashing method. These methods were able to improve the model performance with added computational cost. However, the objective of obtaining better keypoint pairs remains an open research problem. As a result, researchers are trying to propose new methods to remove outlier keypoints efficiently.

To filter out some keypoints produced by the SIFT method, Yang et al. [44] proposed a two-stage filtering mechanism. In the first stage, a grid based filtering technique was employed to remove the key point pairs that contain one outlier keypoint i.e., not belonging to a source or a target copied region. In the second stage, a clustering based outlier keypoints removal method was employed where 2 nearest neighbors (2NN) clustering method was used. Kumar and Meenpal [20] also followed a similar approach like [44]. But in this method, keypoints were detected using SIFT and KAZE features first and then salient keypoints from these two sets of keypoints were detected using a ranker. For ranking features, the authors made use of distinctiveness, repeatability, and detectability properties of the keypoints. In another work, to extract better set of keypoints using SIFT method, Tahaoglu et al. [38] employed rotation invariant local binary pattern (Rot-LBP) operator to transform the input image that represents the texture of the image more prominently. To deal with multiple (more than two) cloning of copied regions and forgery regions only involves small or smooth regions, Niu et al. [33] first extracted a dense set of keypoints using modified SIFT method and then represented them by robust and discriminative moment invariants. Next, non-important keypoints were removed using complex-valued invariant feature matching technique specially designed for fast keypoint matching. In a recent work, Gan et al. [17] modified the existing SIFT based keypoint extraction method to obtain better keypoints and then employed the feature label matching method to downsample keypoints detected in homogeneous regions. Finally, a hierarchical segmentation filtering was employed to prune the outlier keypoints.

In general, the keypoint based methods can locate the matched keypoints in lesser CPU time than the block based and deep learning based methods. But the use of above mentioned filtering methods increase the CPU time by a significant amount as well. These methods perform well when geometric transformation like scaling and rotation are employed on copied regions. But most of these techniques fail to locate the regions very accurately as they usually do not work well when the objects are hidden in smooth background areas. Besides, a small duplicated regions with a small number of pixels may fail to get detected. Moreover, in many cases, these techniques are unable to distinguish between copy-move

forged regions and naturally similar regions, and hence are less accurate in marking the forged regions properly.

## 2.2 Block based methods

Most of the methods of this category follow similar steps with slight variations as described by Fridrich et al. [16]. In this work, the authors first divided an input image into a number of overlapping blocks and then extracted discrete cosine transform (DCT) coefficients as features. Such features were then sorted lexicographically. The Euclidean distance between feature vectors was calculated and then the copy-move forgery was detected by checking feature vectors exceeding a certain threshold value. Since in the processing steps followed by block based CMFD methods, in general, same as steps mentioned by Fridrich et al. [16] thus a trend of using better features is observed. The performance of these methods mostly relied on the used feature descriptor i.e., a better feature descriptor will end up with better performance. Ryu et al. [35] utilized Zernike moments as features. In another work, Mahmood et al. [27] used local binary pattern (LBP) and stationary wavelets features to detect forgery. Bi et al. [8] performed feature extraction by applying multilevel dense descriptor (MLDD). False regions were detected by looking at the texture of neighboring pixel groups, not matching feature vectors. They aimed to reduce the error margin using adaptive distance and orientation based filtering. Meena and Tyagi [28, 29] used tetrolet transform and Gaussian-Hermite moments based features in the work [29] and [28] respectively. In another work, Kumar et al. [21] used coefficients generated by employing DCT and singular value decomposition (SVD) on image as features. However, in both the cases they used less number of coefficients as compared to state-of-the-art methods.

Apart from aforementioned block based CMFD methods, a few methods [9, 13, 14] tried patchmatch technique proposed by Barnes et al. [5] to estimate similarity between two patches (here blocks). Cozzolino et al. [13] modified the existing patchmatch algorithm to deal with rotation made during pasting the copied region and accordingly developed a randomized approach to detect similar patches in an image. Recently, Chen et al. [9] used a modified version of the patchmatch algorithm and extracted features using fractional quaternion cosine transform to detect forgery. Li et al. [22] also tried to find similarity among semantically independent patches using a two-step process. First, they found the suspicious pairs that may contain forged regions and estimated the affine transformation matrix. Second, they refined the transformation matrix using an expectation maximisation (EM) algorithm and confirmed the existence of forgery. Apart from these, Meena and Tyagi [30] proposed a CMFD technique that coupled the benefits of block based and keypoint based approaches. In this work, the authors used Fourier-Mellin moment based features for block matching and SIFT method to extract keypoints within the detected similar blocks.

The block based methods, in general, perform satisfactorily against the copy-move attacks when images are post processed using operations like blurring, color reduction, brightness change. These methods also perform well for detecting plain copy-move forgery. They produce good results as the extracted features try to represent the key characteristics of the blocks and measure using on all pixels' information in the block. However, these methods have inherent limitations like they, in general, require relatively higher CPU time to test an image as compared to a keypoint based method and sometimes fail to perform satisfactorily to address copy-moved regions that are transformed using some geometric

operations like large scaling and shape-deformation. Apart from these, the performance of a block based method largely depends on the used feature descriptor.

### 2.3 Deep learning based methods

The methods described above used handcrafted features to represent characteristics of a block or a keypoint. However, recently, in many real-life applications deep learning based models have proved to be beneficial. Hence, some researchers have proposed deep-learning based methods for CMFD to overcome the issues that may arise due to the limitations handcrafted features. For example, Liu et al. [26] proposed a convolutional kernel network which may be considered as a data-driven local descriptor with the deep convolutional architecture. In this work, the authors first over-segmented an input image using an adaptive semantic segmentation method and then tried to locate the forged areas in it with the help of the convolutional kernel network. Wu et al. [42] proposed an end-to-end trainable parallel deep neural network scheme called BusterNet for CMFD. One network localizes potential forging regions with the help of visual artifacts and the other network detects the copy-move regions by assessing visual similarities. Later, Chen et al. [10] improved the BusterNet model by introducing two serially placed sub-networks: the copy-move similarity detection network (CMSDNet) and the source/target region distinguishment network (STRDNet) inside the parallel networks of the BusterNet. To explore self-correlation features for CMFD, Zhu et al. [47] and Zhong et al. [46] introduced AR-Net and Dense-InceptionNet. Abhishek and Jindal [1] proposed another semantic segmentation based deep convolution neural network to detect the copy-move and splicing forgery images. A Convolutional Long Short-Term Memory (CovLSTM) network was proposed by Elaskily et al. [15] while Barni et al. [6] designed a multi-branch convolutional neural network (MBCNN) for CMFD. Jaiswal and Srivastava [19] proposed a multi-scale multi-stage deep learning model where the authors used pixel based classification technique for detecting copied and pasted regions. The model is facilitated with the powerful encoder-decoder model.

Deep learning based algorithms definitely open up a new research direction and the performance of these methods is competent. Their relatively novel approach of classifying pixels into forged and real ones shows some great potentials for CMFD. However, in their present state one can easily observe some notable limitations. The most notable one is that the performance of state-of-the-art deep learning based methods is not at par compared to state-of-the-art keypoint or block based methods. Moreover, the performance of such methods is largely controlled by the quality and amount of the training data available. Here, it is noteworthy to mention that preparing the quality training data is not an easy task as the type of copy-move forged images evolves over the time due to technological advancements. Furthermore, most of these methods possess a technical limitation. An image-based deep learning model considers all input images with fixed size and therefore, images of varying dimensions need to resize to make it a fixed-sized input which may in turn hide the information these methods look for.

## 3 Motivation

From the above discussion it is clear that block based methods are more robust compared to the keypoint based or deep learning based methods as they often fail to achieve satisfactory results. Hence, in this work, we have followed the block based approach to design a new CMFD method. Since the performance of block based CMFD methods largely depends on

used feature descriptor and thus, here we made use of the LTrP features. LTrP features are texture based features which help in detecting the forged regions that have been changed in brightness, contrast and color aspects. From the above discussion, it has also been observed that developing the resistance to these types of modifications have not been explored much in many of state-of-the-art methods till date.

### 4 Local tetra pattern: an overview

The idea of LTrP is originally proposed by Murala et al. [32] for texture classification. LTrP describes the spatial structure of the local texture based on the direction of the pixels using horizontal and vertical derivatives. Let, for a given an image  $I$ ,  $G_{\theta}^1(p_n)|_{\theta=0^{\circ}, 90^{\circ}}$  denotes the first-order derivatives along  $0^{\circ}$  and  $90^{\circ}$  directions. The original version of the LTrP considers two horizontal neighbors and two vertical neighbors of a center pixel, thus resulting in 4 combinations. However, we stick to using only one combination. Let  $p_c$  denote the center pixel in  $I$ ,  $p_h$  denote the horizontal neighbor, i.e., the pixel to the left of  $p_c$ , and  $p_v$  denote the vertical neighbor, i.e., the pixel above the  $p_c$  respectively. Now, the first-order derivatives at  $p_c$  can be written as

$$G_{0^{\circ}}^1(p_c) = I(p_h) - I(p_c) \tag{1}$$

$$G_{90^{\circ}}^1(p_c) = I(p_v) - I(p_c) \tag{2}$$

The direction value at  $p_c$  (say,  $G_{\theta}^1(p_c)$ ) is calculated as

$$G_{\theta}^1(p_c) = \begin{cases} 1, & G_{0^{\circ}}^1(p_c) \geq 0 \quad \text{and} \quad G_{90^{\circ}}^1(p_c) \geq 0 \\ 2, & G_{0^{\circ}}^1(p_c) < 0 \quad \text{and} \quad G_{90^{\circ}}^1(p_c) \geq 0 \\ 3, & G_{0^{\circ}}^1(p_c) < 0 \quad \text{and} \quad G_{90^{\circ}}^1(p_c) < 0 \\ 4, & G_{0^{\circ}}^1(p_c) \geq 0 \quad \text{and} \quad G_{90^{\circ}}^1(p_c) < 0 \end{cases} \tag{3}$$

From (3), it is evident that any one of four different  $G_{\theta}^1(p_c)$  values can be assigned to each  $p_c$  (i.e.,  $M_{\theta}^1(p_c) \in \{1, 2, 3, 4\}$ ) and eventually, the image is represented by four direction values. For a  $3 \times 3$  neighborhood of a center pixel, there are 8 neighbors. The second order LTrP coefficients (say,  $LTrP_{coeff}$ ) for a block of image are obtained from these 8 neighbors using (4).

$$LTrP_{coeff} = [f(M_{\theta}^1(p_c), M_{\theta}^1(p_1)), f(M_{\theta}^1(p_c), M_{\theta}^1(p_2)), f(M_{\theta}^1(p_c), M_{\theta}^1(p_3)), \dots, f(M_{\theta}^1(p_c), M_{\theta}^1(p_8))] \tag{4}$$

where  $p_1, p_2, \dots, p_8$  are the neighbors of  $p_c$  and  $f$  is the function defined in (5)

$$f(G_{\theta}^1(p_c), G_{\theta}^1(p_n)) = \begin{cases} 0, & G_{\theta}^1(p_c) = G_{\theta}^1(p_n) \\ G_{\theta}^1(p_n), & \text{Otherwise} \end{cases} \tag{5}$$

In (5),  $p_n$  represents the  $n^{th}$  neighbor ( $n=1, 2, \dots, 8$ ) of  $p_c$ .

The original LTrP approach, proposed by Murala et al. [32], goes on to segregate the second order LTrP into three binary patterns first for each of 4 directions resulting in total of 12 binary patterns, after that a 13th binary pattern based on the magnitude component is also added to the feature list. But there was negligible change in the accuracy after this



addition, hence to reduce both time and implementation complexity we have used the LTrP feature descriptor with the 8 features as obtained in (4).

## 5 Proposed scheme

This section describes our proposed CMFD method which addresses intra-image copy-move forgery. Since the copied regions come from the same image, at the end of the detection process, the forged area and the original area will be relatively similar. It is to be noted that often some post-processing operations such as color reduction, retouching, blurring, and contrast enhancement are performed to make the forged image more realistic. These operations can hamper the correlation between copy-moved regions up to a great extent, thereby making it difficult to detect forged areas. However, a robust feature descriptor can successfully identify such forged regions. The detection of such forgery will, therefore, consist in finding relatively similar areas in an image. An exhaustive search process cannot be a viable option because it is computationally costly and hence effective only for small sized images. Keeping this fact in mind, we use a very stable approach that involves dividing the suspected image into a number of overlapping blocks. After dividing the image into blocks, LTrP texture features are extracted from the blocks. Finally, the features are sorted lexicographically to make a sufficiently reliable decision based on the similarity of consecutive blocks. Complete workflow of the proposed scheme is shown in Fig. 1

### 5.1 Preprocessing

After tampering the image, the forger tries to hide traces of the forgery - which is usually done in RGB color space. Many authors [18, 40, 41] have found that the chrominance spaces

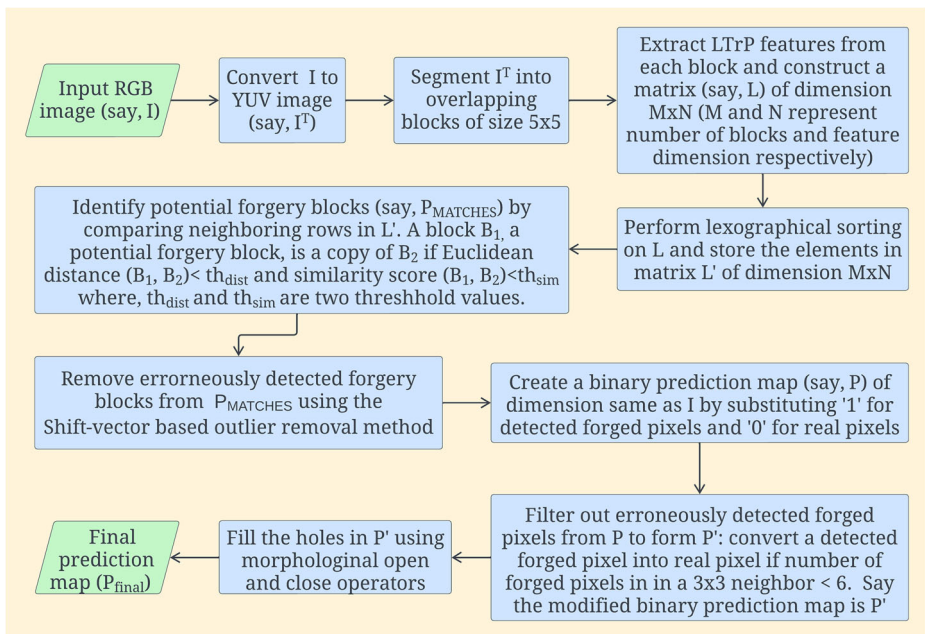
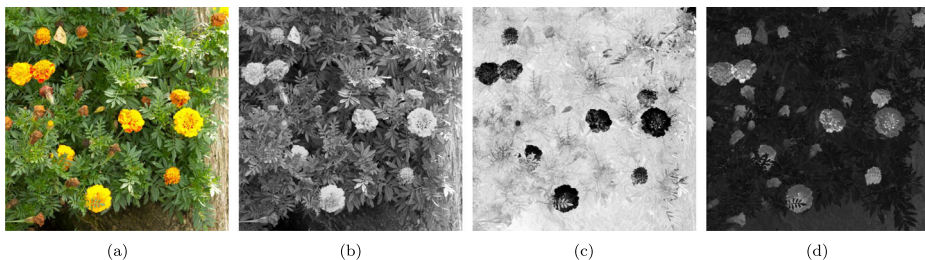


Fig. 1 Workflow of the proposed copy-move forgery detection scheme





**Fig. 2** An RGB image and its corresponding Y, U, V channels in YUV color space: (a) RGB image, (b) Luminance/brightness component (i.e., Y channel) of the RGB image, (c) Chrominance component (i.e., U channel) of blue channel of the RGB image, (d) Chrominance component of red channel of the RGB image (i.e., V channel)

are much more effective at detecting copy-move forgery. Therefore, the input image which is in RGB color space, is first converted into YUV color space to increase the robustness of the proposed method. The luminance component (Y), blue chrominance component (U), and red chrominance component (V) of an image in YUV color space are calculated using (6).

$$\begin{cases} Y = 0.299 * R + 0.587 * G + 0.114B \\ U = -0.14713 * R - 0.28886 * G + 0.436 * B \\ V = 0.615 * R - 0.51498 * G - 0.10001 * B \end{cases} \quad (6)$$

In (6), R, G and B are the color components that represent the red, green and blue color information of a pixel in RGB image respectively. In Fig. 2, we have shown an RGB image (see Fig. 2a) and the gray-scale images representing luminance (see Fig. 2b), blue chrominance (see Fig. 2c), and red chrominance (see Fig. 2d) information of the RGB image. In other words, Fig. 2b represents how bright the RGB image is or the corresponding gray-level image of the RGB image. However, Fig. 2c and d represent dominance of blue and red channels pixel intensity respectively over the pixel intensities of other two color channels. To be specific, in Fig. 2c, a black color pixel represents absence of blue color intensity (i.e.,  $B = 0$ ) or impact less blue color intensity (i.e.,  $0.14713 * R + 0.28886 * G \geq 0.436 * B$ ) in the corresponding pixel of the RGB image while a white pixel represents the absence of red and green color in the RGB image.

### 5.2 Dividing image into blocks

The YUV image is decomposed into overlapping square fixed-size blocks of size  $5 \times 5$  pixels. Thus, if the spatial resolution of an input image is  $M \times N$  pixels, then the image generates  $(M-4) \times (N-4)$  number of blocks.

### 5.3 LTrP feature extraction

Now, each block is of size  $5 \times 5$  pixels and contains three channels Y, U and V. From each block, we extract LTrP features for each channel which gives us 8 LTrP coefficients. We then get a feature vector of length 24 from those 8 coefficients from each of the 3 channels. Therefore, for an image of size  $M \times N$  pixels, a feature matrix (say,  $F$ ) of  $(M-4) \times (N-4)$  rows and 24 columns, where each row indicates a block in the image, is constructed.

## 5.4 Feature matching similar blocks

It has been observed that even after undergoing tampering and image-level different processing on the tampered image, the blocks of the duplicate patches do not lose their similarity entirely. As a result, the features extracted from similar blocks still remain similar to a great extent. We sort  $F$  lexicographically such that all the rows are arranged in lexicographic order. As a result, the rows of  $F$  corresponding to the similar blocks end up in adjacent locations (i.e., rows-wise) in the sorted matrix. We analyse each row (corresponding to each block) in  $F$ , first with its immediately adjacent succeeding neighbor and then with the row after that. When two blocks are being analysed, the proposed technique considers them as potentially forged blocks if: (a) the absolute difference between them is less than a similarity threshold  $\eta_{simi}$ , and (b) the Euclidean distance between them is less than a distance threshold  $\eta_{dist}$  - whose value we have experimentally set as 20. The proposed technique includes all such potentially forged blocks to a match list  $P\_MATCHES$ .

## 5.5 Removal of false matches

Certain objects which are homogeneous in the color space like sea, field, sky etc. may be present in the image. As a result, the blocks which make up such self-similar objects are often matched falsely. Such erroneous matches should effectively be removed. In this work, we use a technique based on shift vectors to remove the outliers, as it is a robust and popular method used by many authors in the past [16]. A frequency matrix (say,  $V$ ) which acts as a counter for the shift vectors is initialized to zero before the process starts. For each pair of potential blocks in  $P\_MATCHES$ , we calculate the shift vector (say,  $\sigma$ ) between them and the counter  $V$  is incremented for that particular  $\sigma$ . The  $\sigma$  between the two potential blocks whose coordinates are  $(x_1, y_1)$  and  $(x_2, y_2)$  is calculated as  $\sigma = (\sigma_x, \sigma_y) = (x_1 - x_2, y_1 - y_2)$ . It is to be noted that  $-\sigma$  and  $+\sigma$  indicate the same shift between similar blocks, hence we normalize the  $\sigma$ s by taking the absolute values. The normalized  $\sigma$  counter i.e.,  $V$  is incremented by one for each matching pair of potential blocks like so:  $V(\sigma_x, \sigma_y) = V(\sigma_x, \sigma_y) + 1$ .

After the matching process ends,  $V$  contains the frequencies with which each normalized  $\sigma$  occurrences. Thereafter, we find the highest frequency ( $V_{max}$ ) from  $V$ . Consequently, we discard all the normalized  $\sigma$ s, whose frequency is less than a shift frequency threshold (say,  $\eta_{shift}$ ). The value of  $\eta_{shift}$  is obtained dynamically from  $V_{max}$  by multiplying it with a factor of 0.25 (determined after experimentation). For all the remaining shift vectors, corresponding pair of matching blocks which contribute to that specific shift are considered as duplicate blocks. Hence, the patches that might have been forged comprise of these blocks. The entire outlier removal process is described in Algorithm 1.

## 5.6 Post-processing on prediction map

A prediction binary map  $P_1$  with all equal to zero and of size equal to input image is created. The coordinates of the duplicate blocks are then marked as white in  $P_1$  to indicate the forged areas.

After this we need proper post-processing because there may be some voids (false negatives) and islands (false positives) in the prediction map. Two types of post-processing methods are applied on the prediction binary map to make the final result more accurate. The first one involves checking the neighborhood of each pair of blocks marked as duplicate. We filter out all those pixels in the prediction map  $P_1$  whose 3x3 neighborhood does

---

```

input : P_MATCHES, M, N
output: Duplicate blocks
 $V \leftarrow \text{int}[M, N]$ ;
for  $i \leftarrow 1$  to  $M$  do
  | for  $j \leftarrow 1$  to  $N$  do
  | |  $V[i, j] \leftarrow 0$ 
  | end
end
for  $A, B$  in  $P\_MATCHES$  do
  |  $\text{shift}_x \leftarrow \text{abs}(A_x - B_x)$ ;
  |  $\text{shift}_y \leftarrow \text{abs}(A_y - B_y)$ ;
  |  $V[\text{shift}_y][\text{shift}_x] \leftarrow V[\text{shift}_y][\text{shift}_x] + 1$ 
end
 $V_{\max} \leftarrow \max(V)$ ; // get maximum frequency in V
 $\eta_{\text{shift}} \leftarrow V_{\max} * 0.25$ ;
 $V[V \leq \eta_{\text{shift}}] \leftarrow 0$ ;
 $V[V > \eta_{\text{shift}}] \leftarrow 1$ ;
 $Duplicates \leftarrow []$ ;
for  $A, B$  in  $P\_MATCHES$  do
  |  $\text{shift}_x \leftarrow \text{abs}(A_x - B_x)$ ;
  |  $\text{shift}_y \leftarrow \text{abs}(A_y - B_y)$ ;
  | if  $V[\text{shift}_y][\text{shift}_x] == 1$  then
  | |  $Duplicates.append((A, B))$ ; // append the pair to list
  | | "Duplicates"
  | end
end

```

---

**Algorithm 1** Outlier removal using shift vectors.

not contain at least 6 forged pixels. These (false detection) noisy pixels could be removed by morphological close operation too but by doing that, we may lose the shape of the main forged areas to a great extent. To remove holes and discontinuities that might still be remaining in the detection map, we further apply morphological operations erosion and dilation in suitable combination to achieve the desired results. After this, we get the final detection binary map  $P_{final}$ .

## 5.7 Computational complexity analysis

The proposed method is comprised of several subprocesses. Hence, we first define the time complexity for each subprocess and then define the complexity of the entire model. We discuss the time complexity using the asymptotic notation 'big-o' ( $O$ ). The time complexity of preprocessing where we convert a color image from RGB color space into corresponding YUV color space is  $O(M * N)$ , where  $M$ , and  $N$  represent height and width of an image since (6) is employed for each pixel of the input image. Next, we divide the images into multiple overlapping blocks. This process can be performed using  $O((M - 4) * (M - 4))$  or  $O(M * N)$  time complexity as we are dividing the images into  $(M - 4) * (M - 4)$  subimages and the division process takes some constant time. Next, from each subimage, we extract the LTrP features. The time taken by this LTrP features extraction process is dependent on

the dimension of the image used for feature extraction which is here  $5 \times 5$  and hence in our case, it takes constant time say,  $c$ . Thus the overall feature extraction process can be performed using  $O(c * (M - 4) * (M - 4))$  or  $O(M * N)$ . The length of the LTrP features for each block is 24 and number of such feature vectors for an image is  $(M - 4) * (M - 4)$  and the time complexity of the shorting process is  $O((M - 4) * (M - 4) * 24)$  or  $O(M * N)$ . In a similar way, the process: feature matching and removal of false matches can be performed using  $O(M * N)$  time complexity. Post processing is performed on the entire image and thus it can be done using  $O(M * N)$  time complexity. Hence, the overall time complexity of our algorithm is  $O(M * N)$  which means the dimension of the input image controls the time complexity of the proposed method.

## 6 Experimental evaluation

### 6.1 Image datasets

To evaluate the performance of the proposed technique, a dataset is required which contains original images and forged images along with the corresponding ground truth masks. We use two standard datasets which are publicly available: GRIP [14] and CoMoFoD [39], for evaluation of the proposed technique.

The GRIP dataset contains 80 forged and their 80 original counterparts, each of dimensions  $768 \times 1024$  pixels. The forged images have duplicate regions of various shapes and sizes, thus making it very difficult to detect the forgeries.

The second dataset CoMoFoD contains total 10,400 images, each of spatial resolution  $512 \times 512$  pixels. There are 200 sets of images. They can be classified into five categories (each containing 40 sets of images) - translation, rotation, scaling, distortion, combination. However, we have done all experiments on the first 40 sets of images only, which are forged images involving only translation. Now each set of images has 52 images. Of these there are 25 forged images, 25 original images, 1 binary ground truth map, and 1 RGB ground truth mask. Of the 25 forged, and original images, there are 1 image with no post-processing, 3 images with brightness change, 3 images with contrast adjustment, 3 images with color reduction, 3 images with blurring, 9 images with JPEG compression, 3 images with noise addition. Detailed attack descriptions and settings can be found in [39].

### 6.2 Evaluation metrics

To evaluate the performance of the proposed CMFD method, we consider three popularly used parameters: precision (say,  $Pr$ ), recall (say,  $Re$ ) and  $F_1$  score. Let,  $T_P$  be the number of forged pixels that are detected as forged pixels i.e., the number of true positive pixels,  $F_P$  be the number of non-forged pixels that are erroneously detected as forged pixels i.e., the number of false positive pixels and  $F_N$  be the number of forged pixels that are detected as non-forged pixels i.e., number of false negative pixels. Now, the values of  $Pr$  and  $Re$  are calculated as

$$Pr = \frac{T_P}{T_P + F_P} \quad Re = \frac{T_P}{T_P + F_N} \quad (7)$$

Here,  $Pr$  represents the probability that a detected forgery is truly forged, and  $Re$  represents the probability that an actual forgery is successfully detected. From the values of

$Pr$  and  $Re$  in (7),  $F_1$  score, which is the harmonic mean of  $Re$  and  $Pr$ , is calculated using (8)

$$F_1score = 2 * \frac{Pr * Re}{Pr + Re} \quad (8)$$

For pixel-level evaluation, we compare the final prediction map (i.e.,  $P_{final}$ ) of an image with its ground truth to calculate  $T_P$ ,  $F_P$  and  $F_N$  pixel counts for each image in the datasets in use. From these counts, two different protocols could be followed to calculate the performance of a method at pixel-level-

- **Protocol A:**  $T_P$ ,  $F_P$  and  $F_N$  counts are made over the all images in a dataset first and then from those counts, we can calculate the values of  $Pr$ ,  $Re$  and  $F_1$  score.
- **Protocol B:** We calculate the values of  $Pr$ ,  $Re$  and  $F_1$  score for each image in the dataset first and then we find out the average of those scores.

Protocol A is better suited to describe the overall performance of our algorithm, whereas the protocol B is efficient in describing the localization performance since  $F_1$  score is undefined when  $T_P$  and  $F_P$  and/or  $F_N$  are zero. Almost all authors in the past have followed protocol B for evaluation at pixel-level, hence we also assess our proposed technique at pixel-level using protocol B.

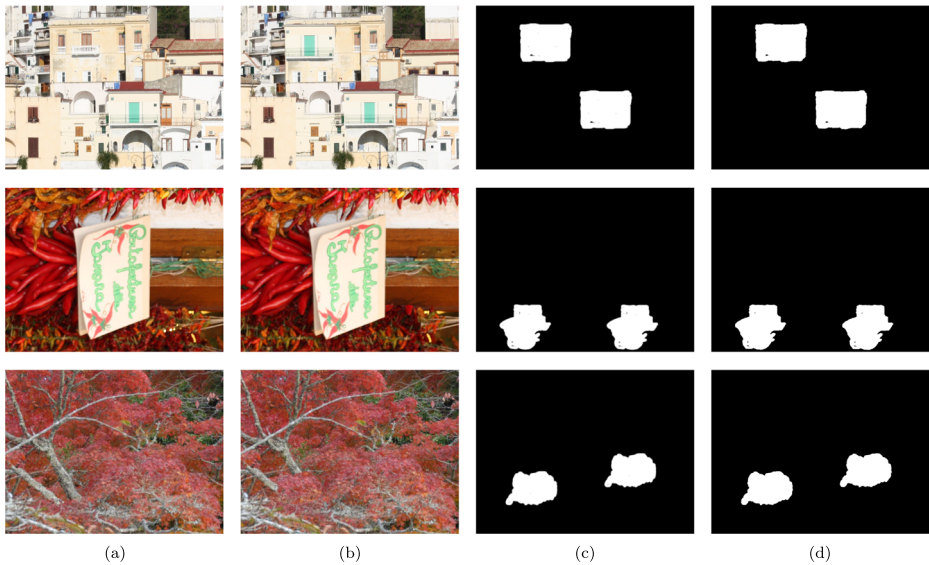
At image-level,  $T_P$  is the total number of forged images that are correctly detected as forged images,  $F_P$  is the total number of original images that are wrongly identified as forged and  $F_N$  is the total number of forged images that have been erroneously considered as real. From these, we calculate the values of image-level  $Pr$ ,  $Re$  and  $F_1$  score using the formulas mentioned above.

### 6.3 Image forgery detection results

The aim of this paper is not only detection of copy-move forgery but the localization of the forged regions as well. Therefore, experiments at image-level are necessary to assess the ability of the technique to distinguish between original and forged images, whereas experiments at pixel-level are equally important since they quantify how efficient the method is in localizing the forged regions in forged images. Image-level as well as pixel-level experiments have been conducted on both the datasets CoMoFoD and GRIP as they both contain forged images and their original counterparts.

#### 6.3.1 Performance under plain copy-move forgery

When a part of the image is duplicated within the same image without any further tampering or post-processing operation, it is considered as a plain copy-move forgery. The dataset GRIP consists of 80 plain copy-move forged images and their 80 original images. Our proposed technique correctly detects all the forged images and does not falsely detect any original image as forged, which implies it obtains the image-level values of  $Pr$ ,  $Re$  and  $F_1$  score as 1 on this dataset. Further, the proposed technique also obtains a promising  $F_1$  score of 0.9834 at pixel-level on the GRIP dataset. Detection results under GRIP dataset are shown in Fig. 3. The dataset CoMoFoD contains 40 forged images and their 40 original counterparts fall under plain copy-move forgery, hence at image-level the proposed technique achieves impressive values of  $Pr$ ,  $Re$  and  $F_1$  score of 1. At pixel-level also, the proposed technique achieves a very high  $F_1$  score of 0.9430. Detection results under plain copy-move forgery obtained by the proposed method are illustrated in the first row of Fig. 4.



**Fig. 3** Copy-move forgery detection results on the GRIP dataset: (a) Original image, (b) Forged image, (c) Ground truth image, (d) Detection result of proposed technique

We can see from this result that the present method can detect multiple copy-move forgeries in the same image.

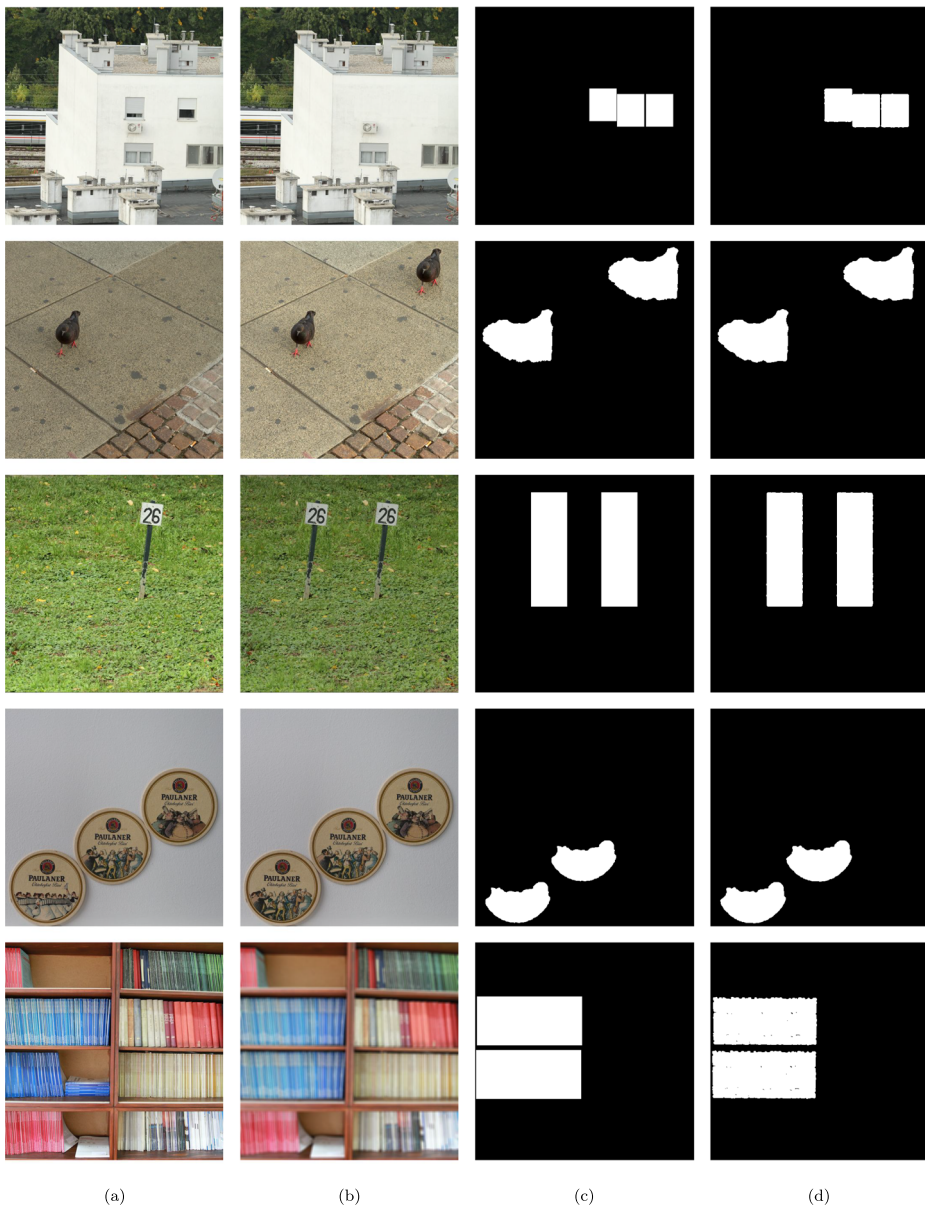
### 6.3.2 Performance against brightness change on copy-move forged images

To analyse the performance of our proposed technique against brightness change of the images, experiments are performed on all 120 images (numbered from 001\_F\_BC1 to 040\_F\_BC3) of the CoMoFoD dataset. Results shown in the second row of Fig. 4 demonstrate that the proposed technique can locate duplicated regions effectively even when the image is post-processed with brightness change. From Fig. 5a, we observe that there is a negligible change in performance in range [0.01, 0.95] and range [0.01, 0.90]. This is because the LTrP feature descriptor is texture based and hence changes in brightness do not change the feature extracted. However, brightness change in range [0.01, 0.80] almost bleaches those images which were already bright before post-processing, hence in these images, the texture of the bright areas is lost to a great extent, and due to this the LTrP feature descriptor does not perform as efficiently. As a result, the performance falls, although very slightly, in the range [0.01, 0.80].

### 6.3.3 Performance against contrast adjustments on copy-move forged images

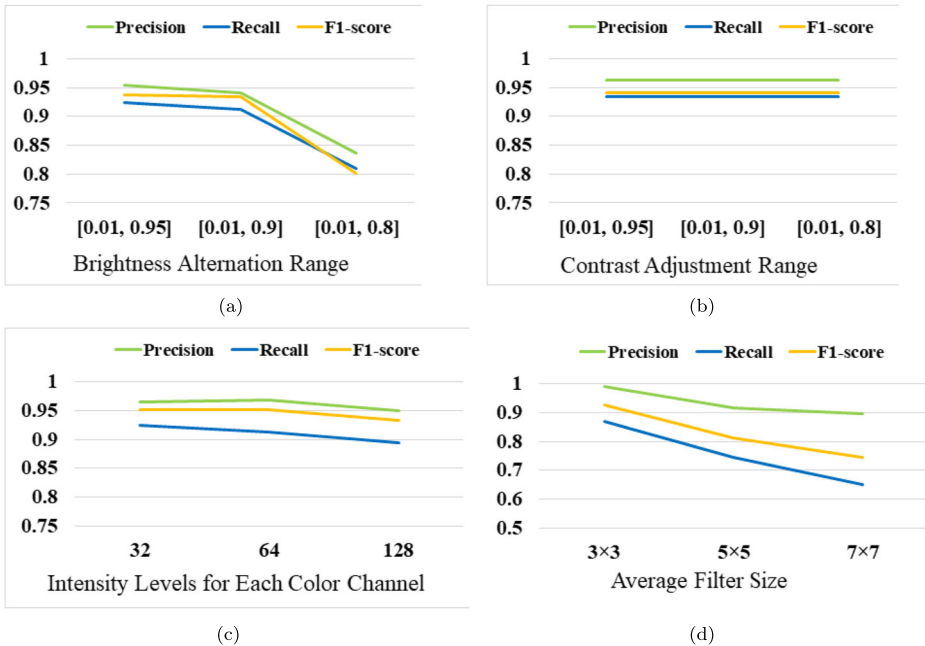
The CoMoFoD dataset also has 120 forged images with various levels of contrast adjustments - 001\_F\_CA1 to 040\_F\_CA3. Images are post-processed with contrast adjustments in the range [0.01, 0.95] - images whose names end with \_CA1, in the range [0.01, 0.90] - images whose names end with \_CA2 and in the range [0.01, 0.80] - images whose names end with \_CA3 [39]. From the almost flat lines in Fig. 5b, it is clearly evident that our proposed technique is resistant to contrast adjustment. Third row of Fig. 4 shows forgery detection results with an image that has been contrast adjusted.





**Fig. 4** Copy-move forgery detection results on the CoMoFoD dataset: (a) Original image (b) Forged image (c) Ground truth image (d) Copy-move forgery detection result of the proposed technique. First row: plain copy-move forgery image, Second row: brightness changed [0.01, 0.8] forgery image, Third row: Contrast adjusted (adjustment factor [0.01, 0.8]) forgery image, Fourth row: color reduced (32 colors per channel) forgery image, Fifth row: blurred ( $7 \times 7$  averaging filter) forgery image





**Fig. 5** Average values of  $Pr$ ,  $Re$  and  $F1$  score at pixel-level under various post-processing operations: (a) brightness change (b) contrast adjustment (c) color reduction, and (d) image blurring

### 6.3.4 Performance against color reduction on copy-move forged images

The performance of our proposed technique against color reduction is tested by considering all 120 images - 001\_F\_CR1 to 040\_F\_CR3 of the CoMoFoD dataset. The forged images are post-processed with the number of different intensity levels reduced from 256 to 32 (images whose names end with \_CR1), 64 (images whose names end with \_CR2) or 128 (images whose names end with \_CR3) levels [39]. Copy-move forgery detection result on the image that is post-processed with color reduction is demonstrated in the fourth row of Fig. 4. The graph in Fig. 5c indicates that the proposed technique is very robust against color reduction.

### 6.3.5 Performance against blurring on copy-move forged images

There are 120 images in CoMoFod from 001\_F\_IB1 to 040\_F\_IB3, which have been blurred after copy-move forgery involving only translation. Images whose names end with suffix \_IB1 are blurred with an averaging filter of  $3 \times 3$ , those ending with suffix \_IB2 are blurred with an averaging filter of  $5 \times 5$ , and those ending with suffix \_IB3 are blurred with an averaging filter of  $7 \times 7$  [39]. Experiments have been performed on all these 120 images from the CoMoFoD dataset. Sample results on the forged blurred images with  $5 \times 5$  averaging filters are shown in the fifth row of Fig. 4. Figure 5d shows that the proposed technique is not that affected by image blurring and can still detect forgery quite efficiently.

**Table 1** IOU analysis for the proposed method under the GRIP and CoMoFoD datasets

Dataset	Post-processing	IoU
GRIP	Plain copy-move	0.9676
	Plain copy-move	0.8959
	Brightness change in range [0.01, 0.95]	0.8857
	Brightness change in range [0.01, 0.90]	0.8766
	Brightness change in range [0.01, 0.80]	0.7560
	Contrast adjustment in range [0.01, 0.95]	0.8949
CoMoFoD	Contrast adjustment in range [0.01, 0.90]	0.8961
	Contrast adjustment in range [0.01, 0.80]	0.8939
	Image blurring using averaging filter of size $3 \times 3$	0.8606
	Image blurring using averaging filter of size $5 \times 5$	0.7279
	Image blurring using averaging filter of size $7 \times 7$	0.6422
	Color reduction - number of colors per channel = 32	0.8950
	Color reduction - number of colors per channel = 64	0.8934
Color reduction - number of colors per channel = 128	0.8665	

### 6.3.6 IoU analysis

Table 1 shows the intersection over union (IoU) results between the ground truth and detected forged regions. The studies show the high IOU overlap with GRIP dataset and satisfactory IOU overlap with several subparts of the CoMoFoD dataset that facilitates good detection.

### 6.3.7 Comparison with state-of-the-art methods

In this section we analyse the performance of the proposed model with performance of some state-of-the-art methods on both the datasets used here. Table 2 shows  $F_1$  scores of the proposed method at image-level and pixel-level on GRIP dataset in comparison with the some state-of-the-art methods proposed by Cozzolino et al. [13], Cozzolino et al. [14], Bi et al. [8], Bi et al. [7], Wu et al. [42], Wang et al. [43], Meena and Tyagi [28], Meena and Tyagi [29], Meena and Tyagi [30], Barni et al. [6], Gan et al. [17], and Tahaoglu et al. [38]. The detection results outlined in this table show that the proposed method outperforms the other methods at both image-level and pixel-level on the GRIP dataset.

The performances of the proposed CMFD method are compared with performances of some of the past methods proposed by Kumar and Meenpal [20], Kumar et al. [21], Li et al. [22], Liu et al. [26], Meena and Tyagi [28], Meena and Tyagi [29], and Silva et al. [37]. The comparisons are made considering plain copy-move forgery as well as under forgery with various postprocessing operations like no postprocessing, image blurring, Brightness change, Color reduction, and Contrast adjustment as discussed earlier. The comparative results on CoMoFoD dataset are recorded in Table 3. From this table, we can observe that the method proposed by Meena and Tyagi [28] outperforms the proposed method in terms of overall accuracy in terms of the  $F_1$  score. However, this method decided an image as a real image based on the number of detected regions being larger than a preset threshold value. By doing so they eliminated the method-level failure on original images (i.e.,  $F_1$  score = 1.00) and thus the improved performance. But in reality, a forged image can

**Table 2** Comparison of the pixel-level and image-level performances of the proposed method with various existing methods on GRIP dataset

Work Ref.	Feature used	$F_1$ score (image-level)	$F_1$ score (pixel-level)
Cozzolino et al. [13]	Patchmatch	0.9467	0.8867
Cozzolino et al. [14]	Zernike and FMT	0.9340	0.9267
Bi et al. [8]	Multi-level dense descriptor	0.9277	0.8788
Bi et al. [7]	Local bidirectional coherency error refinement	0.9663	0.9298
Wu et al. [42]	BusterNet: a CNN architecture	–	0.2115
Wang et al. [43]	Compact color content descriptor	0.9630	0.9601
Meena and Tyagi [28]	Gaussian-Hermite moments	<b>1.0000</b>	0.9805
Meena and Tyagi [29]	Tetrolet transform	0.9876	-
Meena and Tyagi [30]	Fourier-Mellin and SIFT	–	0.9697
Barni et al. [6]	Multi-Branch CNN	–	0.8405
Gan et al. [17]	improved SIFT	–	0.9211
Tahaoglu et al. [38]	Rotation Invariant Local Binary Pattern and SIFT	0.9625	0.9200
Proposed method	LTrP feature descriptor	<b>1.0000</b>	<b>0.9834</b>

In the case of image-level measure, for an image if we obtain pixel-level  $F_1$  score  $\geq 0.5$  then the image is considered as correctly detected. The bold-faced number represents the best score

**Table 3** Forged region detection results on the CoMoFoD dataset at pixel-level under plain copy-move (forged images with no post-processing) and differently post-processed forged images (brightness change in range [0.01, 0.90], contrast adjustment in range [0.01, 0.80], color reduction to number of colors per channel = 32, and blurring using averaging filter of size  $3 \times 3$ )

Work ref.	Post-processing on forged image	# correctly detected images	$Pr$	$Re$	$F_1$ score
Li et al. [22]	Plain copy-move	17	0.4180	0.8327	0.5566
	Image blurring	13	0.3186	0.9206	0.4734
	Brightness change	15	0.3957	0.7942	0.5283
	Color reduction	15	0.3940	0.8361	0.5357
	Contrast adjustment	14	0.4031	0.8706	0.5511
	Overall	74	0.3859	0.8508	0.529
Silva et al. [37]	Plain copy-move	20	0.4921	0.7754	0.6021
	Image blurring	19	0.4842	0.7653	0.5932
	Brightness change	16	0.4157	0.7429	0.5331
	Color reduction	19	0.4872	0.7884	0.6023
	Contrast adjustment	18	0.5156	0.7382	0.6072
	Overall	92	0.4790	0.7620	0.5876

**Table 3** (continued)

Work ref.	Post-processing on forged image	# correctly detected images	<i>Pr</i>	<i>Re</i>	<i>F</i> <sub>1</sub> score
Liu et al. [26]	Plain copy-move	16	0.4547	0.8023	0.5805
	Image blurring	16	0.4547	0.8023	0.5805
	Brightness change	14	0.4128	0.7848	0.5411
	Color reduction	15	0.4502	0.8448	0.5874
	Contrast adjustment	13	0.4101	0.7218	0.5231
	Overall	74	0.4365	0.7912	0.5625
Meena and Tyagi [28]	Plain copy-move	40	0.9544	0.9567	0.9555
	Image blurring	40	0.9828	0.8724	0.9243
	Brightness change	39	0.9004	0.9076	0.9040
	Color reduction	40	0.9507	0.9273	0.9389
	Contrast adjustment	40	0.9454	0.9582	<b>0.9518</b>
	Overall	199	0.9467	0.9244	<b>0.9349</b>
Meena and Tyagi [29]	Plain copy-move	40	<b>0.9920</b>	0.9216	0.9564
	Image blurring	38	0.7737	<b>0.9403</b>	0.8433
	Brightness change	38	0.7974	<b>0.9174</b>	0.8282
	Color reduction	39	0.8912	<b>0.9598</b>	0.9211
	Contrast adjustment	39	0.8977	<b>0.9692</b>	0.9299
	Overall	194	0.8704	<b>0.9417</b>	0.8958
Kumar and Meenpal [20]	Plain copy-move	–	0.9790	<b>0.9366</b>	<b>0.9573</b>
	Image blurring	–	0.9487	0.9208	<b>0.9313</b>
	Brightness change	–	0.9210	0.8510	0.8940
	Color reduction	–	0.9491	0.9266	0.9355
	Contrast adjustment	–	0.9480	0.8990	0.9230
	Overall	–	0.9492	0.9068	0.9282
Kumar et al. [21]	Plain copy-move	–	0.9100	0.8175	0.8442
Proposed method	Plain copy-move	40	0.9557	0.9365	0.9424
	Image blurring	40	<b>0.9870</b>	0.8712	0.9197
	Brightness change	37	<b>0.9625</b>	0.9119	<b>0.9279</b>
	Color reduction	40	<b>0.9555</b>	0.9365	<b>0.9422</b>
	Contrast adjustment	40	<b>0.9559</b>	0.9342	0.9411
	Overall	197	<b>0.9633</b>	0.9181	0.9346

40 forged images in each category are used for comparison. All scores provided in the form of average score for the corresponding forgery type. In the case of image-level measure, for an image if we obtain pixel-level *F*<sub>1</sub> score  $\geq 0.5$  then the image is considered as correctly detected. Overall performances are calculated by taking average of category-wise performances. The bold-faced number represents the best score

have several copies regions, and therefore setting such a threshold value may be harmful in such cases. Moreover, they used a strong outlier detection method for the images that were decided as forged. All of these helped the authors to obtain better performance over our method. However, their next work [29] failed to outperform the proposed CMFD method.

In summary, the performance of the proposed method is comparable with other methods when considering each evaluation metric used here for comparison.

## 7 Conclusion

In this paper, we have proposed a new method for detecting and localizing copy-move forgery. Experimental outcomes on two publicly available datasets - GRIP and CoMoFoD have demonstrated the superior performance of the proposed method over many state-of-the-art methods. Our method is also robust against various known post-processing attacks. The main reason for this is the LTrP feature descriptor which is indifferent to brightness change, contrast change, and change in color as it is texture based. Comparisons with some other authors' algorithms attest to this, as no other approach we have come across is tolerant of postprocessing attacks. Moreover, experimental results show that it can detect even multiple copy-move forgeries in the same image also.

In spite of this success, the performance of the proposed method in detecting copy-move forgeries still needs to be improved. We have observed that the proposed method gives not-so-good results when it comes to detecting copy-move forgery in the images that have been tampered with by rotating and scaling the copied regions before pasting the same. This is because the LTrP feature descriptor fails to find similarities between such regions, and also the shift vector based outlier removal technique shows its limitation in this context. We plan to improve the CMFD method in the future so that it becomes rotational and scale invariant.

**Data Availability Statement** The datasets used during the current study are GRIP and CoMoFoD datasets that are publicly available from the links <https://www.grip.unina.it/research/83-image-forensics/90-copy-move-forgery.html> and <https://www.vcl.fer.hr/comofod/comofod.html> respectively.

## Declarations

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

## References

1. Abhishek JN (2021) Copy move and splicing forgery detection using deep convolution neural network, and semantic segmentation. *Multimed Tools Appl* 80(3):3571–3599
2. 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 Inform Forens Secur* 6(3):1099–1110
3. Amerini I, Ballan L, Caldelli R, Del Bimbo A, Del Tongo L, Serra G (2013) Copy-move forgery detection and localization by means of robust clustering with j-linkage. *Signal Process Image Commun* 28(6):659–669
4. Ardizzone E, Bruno A, Mazzola G (2015) Copy-move forgery detection by matching triangles of keypoints. *IEEE Trans Inform Forens Secur* 10(10):2084–2094
5. Barnes C, Shechtman E, Finkelstein A, Goldman DB (2009) Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans Graph* 28(3):24
6. Barni M, Phan QT, Tondi B (2021) Copy move source-target disambiguation through multi-branch cnns. *IEEE Trans Inform Forens Secur* 16:1825–1840
7. Bi X, Pun CM (2018) Fast copy-move forgery detection using local bidirectional coherency error refinement. *Pattern Recogn* 81:161–175
8. Bi X, Pun CM, Yuan XC (2016) Multi-level dense descriptor and hierarchical feature matching for copy-move forgery detection. *Inf Sci* 345:226–242


9. Chen B, Yu M, Su Q, Li L (2019) Fractional quaternion cosine transform and its application in color image copy-move forgery detection. *Multimed Tools Appl* 78(7):8057–8073
10. Chen B, Tan W, Coatrieux G, Zheng Y, Shi YQ (2020) A serial image copy-move forgery localization scheme with source/target distinguishment. *IEEE Trans Multimedia* 23:3506–3517
11. Chen L, Lu W, Ni J, Sun W, Huang J (2013) Region duplication detection based on harris corner points and step sector statistics. *J Vis Commun Image Represent* 24(3):244–254
12. Costanzo A, Amerini I, Caldelli R, Barni M (2014) Forensic analysis of sift keypoint removal and injection. *IEEE Trans Inform Forens Secur* 9(9):1450–1464
13. Cozzolino D, Poggi G, Verdoliva L (2014) Copy-move forgery detection based on patchmatch. In: 2014 IEEE international conference on image processing (ICIP). IEEE, pp 5312–5316
14. Cozzolino D, Poggi G, Verdoliva L (2015) Efficient dense-field copy-move forgery detection. *IEEE Trans Inform Forens Secur* 10(11):2284–2297
15. Elaskily MA, Alkinani MH, Sedik A, Dessouky MM (2021) Deep learning based algorithm (convlstm) for copy move forgery detection. *J Intell Fuzzy Systems* 40(3):4385–4405
16. Fridrich AJ, Soukal BD, Lukáš AJ (2003) Detection of copy-move forgery in digital images. In: *Proceedings of digital forensic research workshop*. Citeseer
17. Gan Y, Zhong J, Vong C (2022) A novel copy-move forgery detection algorithm via feature label matching and hierarchical segmentation filtering. *Information Processing & Management* 59(1):102783
18. Hussain M, Saleh SQ, Aboalsamh H, Muhammad G, Bebis G (2014) Comparison between wld and lbp descriptors for non-intrusive image forgery detection. In: 2014 IEEE International symposium on innovations in intelligent systems and applications (INISTA) Proceedings. IEEE, pp 197–204
19. Jaiswal AK, Srivastava R (2022) Detection of copy-move forgery in digital image using multi-scale, multi-stage deep learning model. *Neural Process Lett* 54(1):75–100
20. Kumar N, Meenpal T (2022) Salient keypoint-based copy-move image forgery detection. *Australian Journal of Forensic Sciences* 1–24
21. Kumar S, Mukherjee S, Pal AK (2022) An improved reduced feature-based copy-move forgery detection technique. *Multimed Tools Appl* 1–26
22. Li J, Li X, Yang B, Sun X (2014) Segmentation-based image copy-move forgery detection scheme. *IEEE Trans Inform Forens Secur* 10(3):507–518
23. Liao X, Yu Y, Li B, Li Z, Qin Z (2019) A new payload partition strategy in color image steganography. *IEEE Trans Circuits Syst Video Technol* 30(3):685–696
24. Liao X, Li K, Zhu X, Liu KR (2020a) Robust detection of image operator chain with two-stream convolutional neural network. *IEEE J Sel Top Signal Process* 14(5):955–968
25. Liao X, Yin J, Chen M, Qin Z (2020b) Adaptive payload distribution in multiple images steganography based on image texture features. *IEEE Transactions on Dependable and Secure Computing*
26. Liu Y, Guan Q, Zhao X (2018) Copy-move forgery detection based on convolutional kernel network. *Multimed Tools Appl* 77(14):18269–18293
27. Mahmood T, Mehmood Z, Shah M, Khan Z (2018) An efficient forensic technique for exposing region duplication forgery in digital images. *Appl Intell* 48(7):1791–1801
28. Meena KB, Tyagi V (2019) A copy-move image forgery detection technique based on gaussian-hermite moments. *Multimed Tools Appl* 78(23):33505–33526
29. Meena KB, Tyagi V (2020a) A copy-move image forgery detection technique based on tetrolet transform. *J Inform Secur Appl* 52:102481
30. Meena KB, Tyagi V (2020b) A hybrid copy-move image forgery detection technique based on fourier-mellin and scale invariant feature transforms. *Multimed Tools Appl* 79(11):8197–8212
31. Mohiuddin S, Malakar S, Sarkar R (2021) Duplicate frame detection in forged videos using sequence matching. In: *International conference on computational intelligence in communications and business analytics*. Springer, pp 29–41
32. Murala S, Maheshwari R, Balasubramanian R (2012) Local tetra patterns: a new feature descriptor for content-based image retrieval. *IEEE Trans Image Process* 21(5):2874–2886
33. Niu P, Wang C, Chen W, Yang H, Wang X (2021) Fast and effective keypoint-based image copy-move forgery detection using complex-valued moment invariants. *J Vis Commun Image Represent* 77: 103068
34. Pan X, Lyu S (2010) Region duplication detection using image feature matching. *IEEE Trans Inform Forens Secur* 5(4):857–867
35. Ryu SJ, Kirchner M, Lee MJ, Lee HK (2013) Rotation invariant localization of duplicated image regions based on zernike moments. *IEEE Trans Inform Forens Secur* 8(8):1355–1370
36. Sharma S, Dhavale SV (2016) A review of passive forensic techniques for detection of copy-move attacks on digital videos. In: 2016 3rd international conference on advanced computing and communication systems (ICACCS), vol 1. IEEE, pp 1–6

37. 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
38. Tahaoglu G, Ulutas G, Ustubioglu B, Ulutas M, Nabiye V (2022) Ciratefi based copy move forgery detection on digital images. *Multimed Tools Appl* 1–36
39. 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
40. Ustubioglu B, Ulutas G, Ulutas M, Nabiye V, Ustubioglu A (2016) Lbp-dct based copy move forgery detection algorithm. In: *Information Sciences and Systems 2015*. Springer, pp 127–136
41. Wang Y, Tian L, Li C (2017) Lbp-svd based copy move forgery detection algorithm. In: *2017 IEEE international symposium on multimedia (ISM)*, IEEE, pp 553–556
42. Wu Y, Abd-Almageed W, Natarajan P (2018) Busternet: Detecting copy-move image forgery with source/target localization. In: *Proceedings of the European conference on computer vision (ECCV)*, pp 168–184
43. Xy W, Lx J, Xb W, Hy Y, Pp N (2019) Copy-move forgery detection based on compact color content descriptor and delaunay triangle matching. *Multimed Tools Appl* 78(2):2311–2344
44. Yang J, Liang Z, Gan Y, Zhong J (2021) A novel copy-move forgery detection algorithm via two-stage filtering. *Digital Signal Processing* 113:103032
45. Zandi M, Mahmoudi-Aznavah A, Talebpour A (2016) Iterative copy-move forgery detection based on a new interest point detector. *IEEE Trans Inform Forens Secur* 11(11):2499–2512
46. Zhong JL, Pun CM (2019) An end-to-end dense-inceptionnet for image copy-move forgery detection. *IEEE Trans Inform Forens Secur* 15:2134–2146
47. Zhu Y, Chen C, Yan G, Guo Y, Dong Y (2020) Ar-net: Adaptive attention and residual refinement network for copy-move forgery detection. *IEEE Trans Indust Inform* 16(10):6714–6723

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Affiliations

Sagnik Ganguly<sup>1</sup> · Sanmit Mandal<sup>1</sup> · Samir Malakar<sup>2</sup>  · Ram Sarkar<sup>1</sup>

Sagnik Ganguly  
sagnikg2013@gmail.com

Sanmit Mandal  
sanmitmandal17@gmail.com

Ram Sarkar  
raamsarkar@gmail.com

<sup>1</sup> Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

<sup>2</sup> Department of Computer Science, Asutosh College, Kolkata, India