Disentangling Copy-Moved Source and Target Areas

Ludovic Darmet, Kai Wang, and François Cayre

Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France

Abstract

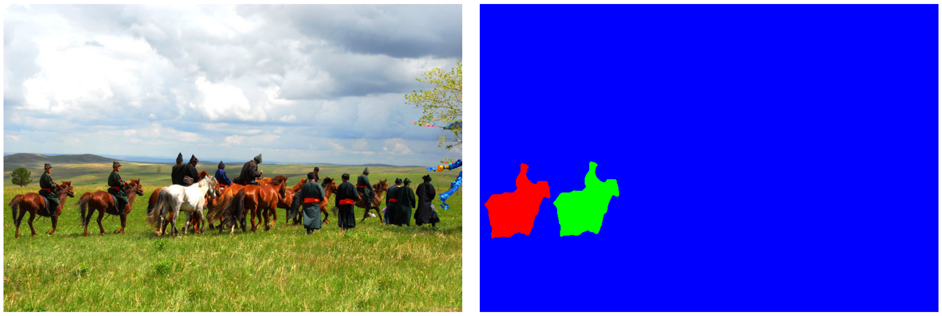
Copy-move is a very popular image falsification where a semantically coherent part of the image, the *source* area, is copied and pasted at another position within the same image as the so-called *target* area. The majority of existing copy-move detectors search for matching areas and thus identify the source and target zones indifferently, while only the target really represents a tampered area. To the best of our knowledge, at the moment of preparing this paper there has been only one published method called BusterNet that is capable of performing source and target disambiguation by using a specifically designed deep neural network. Different from the deep-learning-based BusterNet method, we propose in this paper a source and target disentangling approach based on local statistical model of image patches. Our proposed method acts as a second-stage detector after a first stage of copy-move detection of duplicated areas. We had the following intuition: even if no manipulation (*e.g.,* scaling and rotation) is added on target area, its boundaries should expose a statistical deviation from the pristine area and the source area; further, if the target area is manipulated, the deviation should appear not only on the boundaries but on the full zone. Our method relies on machine learning tool with Gaussian Mixture Model to describe likelihood of image patches. Likelihoods are then compared between the pristine region and the candidate source/target areas as identified by the first-stage detector. Experiments and comparisons demonstrate the effectiveness of the proposed method.

Keywords: Image forensics, Copy-move detection, Image statistics, Gaussian Mixture Model, Likelihood

1. Introduction

The majority of existing methods for copy-move detection output a binary mask indicating without distinction the two types of copy-moved areas, *i.e.,*

Email address: {ludovic.darmet,kai.wang,francois.cayre}@gipsa-lab.grenoble-inp.fr (Ludovic Darmet, Kai Wang, and François Cayre)



(a) (b)

Figure 1: Example of copy-move image from CASIA2 [1] database. (a) Rider on the left has been copied. (b) Ground-truth mask of the tampering, where green indicates the source area and red the target area of the copy-move.

source (original) and *target* (tampered) ones. However, to allow a better understanding of the semantics of the falsified image, it is desirable that we are able to identify the original and the tampered region. For instance in Figure 1, one of the rider has been copied probably to make a bigger group. One could be interested to know what the group really looked like and thus identify the source and target areas. This kind of inferred information about the real position of person in the original image may be very important especially for images used in the court. For the image shown in Figure 1, visually it is difficult to identify the original rider and the tampered one. In addition, the task of visual inspection will be very tedious and time-consuming when there are a lot of suspicious images to be examined. Therefore, it is interesting and necessary to design automatic forensic methods to disentangle the two kinds of areas.

The aim of this work is illustrated in Figure 3 (page 14). On the first row are some images with copy-move forgery. On the second row of ground-truth masks, the red, green and blue color represents respectively the target, source and *pristine* areas. We show on the third row typical outputs of a copy-move detector with the source and target zones identified indifferently. The *objective* of our proposed method is to properly analyze the given image and refine the mask in the third row, so as to produce a mask only highlighting the target, tampered area as illustrated in the fourth row of Figure 3.

To the best of our knowledge, at the moment of conducting research studies presented in this paper, BusterNet [2] has been the only published method which can carry out source and target disentanglement in a copy-move forgery. As presented in subsection 2.3, it relies on the detection of manipulation on target area (*e.g.,* scaling and rotation) for source/target localization. Our proposed method is based on a weaker and more general assumption. We focus on the *statistical deviation* of the candidate source/target areas at both area interiors and boundaries, by taking the identified pristine region (*i.e.,* black part of masks in the third row of Figure 3) as reference. Source area should have more natural boundaries than target area. In addition, if the target area has been manipulated, as it is assumed by BusterNet, statistical deviation would be even bigger. Intuitively, source and target discrimination in copy-move forgeries would be a more challenging task than the detection of other image falsifications such as splicing and inpainting. In splicing forgery, one part of an image is copied and pasted into another host image, so the spliced region would have different statistical properties from the host image. For an image with inpainting falsification, the inpainted area can be considered as a kind of new content inserted into the image. Although inpainting algorithms endeavor to mimic the visual appearance of authentic part of the image when generating inpainted content, it is in general difficult to keep good statistical consistency between authentic and inpainted areas (*e.g.*, existence of subtle differences of textures and noises). Existing splicing [3, 4, 5] and inpainting [6, 7, 8] forensic methods usually make use of such differences or inconsistencies to carry out forgery detection. In contrast, the source and target disentangling in copy-move forgeries appears more challenging because the target area is entirely from the authentic part of the same image. This may explain why there are only few methods proposed in the literature for source and target discrimination (*cf.*, subsection 2.3).

The remainder of this paper is organized as follows. We first briefly review existing methods for copy-move detection in section 2. Then in section 3 we present our proposed method for source/target disentanglement which is based on Gaussian Mixture Model (GMM). We show in section 4 experimental results and comparisons on images with copy-move forgeries from CASIA2 [1] and CoMoFoD [9] datasets. We draw conclusions and suggest some future working directions in section 5.

2. Related Work

Different kinds of copy-move detection methods have been proposed in the literature among the image forensics research community. The most recent ones are naturally based on *deep learning*. However, as we show later in this paper, classical *feature-based* methods are still competitive.

2.1. Feature-based approaches

There are in general two types of feature-based approaches, extracting features respectively from image blocks and key-points [10]. For the block-based methods, images are divided into blocks, *i.e.,* overlapping patches. Features are then extracted from these patches and matched across the image. Finally a step of post-processing is often performed to reduce false alarms. Early features used for copy-move detection were based on Discrete Cosine Transform (DCT) on image patches [11]. Other features were then considered such as Zernike Moment [12, 13] and Tetrolet transform [14]. They are to some extend invariant to scaling and rotation of the copied area, thus can provide more robustness to geometric transformations than features based on DCT. Following this feature extraction stage, a matching of blocks is performed. With a naive, exhaustive search, the cost of matching would be cubic in the number of patches. It would quickly become unpractical with regular sized images or with too much overlap of patches. Therefore strategies of approximate nearest-neighbor search have been utilized in copy-move detectors, such as the kd-tree used by [15] or the PatchMatch algorithm [16] used by the method of [13]. The last stage of post-processing usually consists of filtering of small areas and areas that are too close to each other. Morphological operations may also be performed.

In order to reduce the computational cost, it is possible to work only on key-points, instead of working on the full image as considered in bloc-based methods described above. Features are extracted but only on key-points. Subsequent steps of matching and post-processing are then quite similar but with less samples. Typical key-points used for copy-move detection are based on Scale-Invariant Feature Transform (SIFT) [17] or Speeded Up Robust Features (SURF) [18]. They are, by construction, scale-invariant and therefore provide some robustness against resizing. The main drawback of these methods is that most of key-points are usually to be found around high-entropy region of image. Thus a copy-move performed in a uniform area, *i.e.,* an area potentially without key-points, would be hardly detected. Recently, authors of [19] proposed a hybrid approach which fused block-based and key-point-based methods. A comparison of some representative features-based methods is presented in [10].

2.2. Deep-learning-based approaches

Beside approaches based on matching of features extracted from either blocks or key-points, recently deep-learning-based copy-move detection methods have also been proposed. Ouyang *et al.* presented in [20] one of the first copy-move detectors in the literature that make use of deep neural networks. In their method, synthetic copy-moved images were generated and used to train a Convolutional Neural Network (CNN) for copy-move detection. Liu *et al.* [21] proposed a CNN-based approach that takes image key-points (*e.g.*, SIFT or SURF) as network input for detecting copy-move forgeries.

An important deep-learning-based method of copy-move detection is the so-called BusterNet [2]. This network is composed of two branches: *Mani-Det* and *Simi-Det*. The former is trained to detect manipulations on target area, while the latter aims to detect similar copy-moved areas in an image by using a well-designed self-correlation module. Both branches output a heatmap of the same size as the input image. A fusion of the two branches is finally performed. The source and target discrimination in BusterNet is based on the assumption that the target area has been manipulated, *e.g.,* scaled or rotated. Regarding implementation, branches are first trained separately. This separated training is performed on a synthetic dataset of manipulated copy-move images composed of 100000 samples. The network is quite large so there is no existing realistic dataset that is big enough to train on. In addition, the training requires the use of ground-truth information of source and target in the copy-move which is not always available. The *Mani-Det* branch is trained also with realistic image falsification datasets which are of relatively small scale and typically include a few hundred or thousand tampered images (IEEE IFS-TC dataset and Wild

Web dataset [22]). Then the fusion module is trained alone, with both branches frozen, and finally the full network is fine-tuned in an end-to-end manner.

Zhong and Pun [23] proposed to use DenseNet instead of CNN to detect copy-move tampering. Their method also relies on a module of self-correlation computation. However, unlike BusterNet, this network is not able to identify the source and target areas. Alternatively, authors of BusterNet proposed to use a CNN initially designed for splicing detection and localization [24] to expose copy-move forgeries. With some adaptations of the network training and testing, the adapted CNN can detect well duplicated areas but is not able to discern the source and the target. Recently, Zhu *et al.* [25] designed a CNN with an adaptive attention mechanism for copy-move detection. This special mechanism can guide the network to focus its attention on certain important neurons. Authors reported satisfying performance for localization of duplicated areas but the method does not offer the capability of source and target discrimination.

2.3. Source and target disambiguation

At the moment of preparing and submitting this paper, to our knowledge BusterNet [2] has been the only published method which is able to carry out disentangling of copy-moved source and target areas. As mentioned above, BusterNet is an end-to-end deep-learning-based approach to copy-move detection. The source and target discrimination in BusterNet is based on manipulation detection on target areas. Indeed, BusterNet is built with two branches: one branch extracts deep-features to be matched (for detecting duplicated areas) and the other branch detects manipulation (for locating target area).

We have made efforts to search for pre-prints that study this research problem of source/target disambiguation and found two relevant references. The first method was proposed by Salehi and Mahmoodi-Aznaveh [26]. In this method, authors assumed that the boundary of the target area was smoothed to hide the potential visual discontinuity between the copied fake region and the neighboring pristine part. For a pair of candidate source and target areas, two histograms of local binary pattern (LBP) [27] descriptors were constructed respectively for the boundary of these two areas. Source and target classification was realized by a simple comparison between the standard deviation values of the two histograms. In the second arXiv pre-print that we found [28], Barni *et al.* proposed a deep-learning-based method for source and target disambiguation that relies on a multi-branch CNN. The proposed network is composed of a first branch which compares inside areas of two duplicated zones and a second one comparing boundaries of the two zones. Similar to BusterNet, the authors of [28] also used dataset of synthetic images for the network training.

3. Identification of Source and Target

As explained earlier, most existing methods are powerful in identifying the duplicated areas but not able to perform source/target discrimination. Therefore, we intend to develop a new detector which first makes use of such efficient method to produce a mask of duplicated areas, and then disentangles the source and the target.

3.1. Gaussian Mixture Model (GMM)

An essential machine learning tool used by our method to carry out source and target disentangling is the Gaussian Mixture Model. It is a statistical model based on the most used law of probability in statistics, *i.e.,* the Gaussian distribution. Let $X = (X_1, \dots, X_p) \in \mathbb{R}^p$ be a continuous random vector, in our case representing vectorized image patch of dimension $p = 8 \times 8 = 64$, the probability density function (PDF) of p-dimensional Gaussian distribution is:

$$f_X^{(\mathcal{N})}(x) = \mathcal{N}(x; \mu, \Sigma) = \frac{1}{(2\pi)^p \sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^t \Sigma^{-1}(x-\mu)\right), x \in \mathbb{R}^p, \tag{1}$$

with Σ the covariance matrix of size $p \times p$ which is symmetric and positive definite and μ the mean vector of size p. $X \sim \mathcal{N}(\mu, \Sigma)$ means that random vector X follows a Gaussian distribution of parameters μ and Σ. This distribution is very simple and unimodal, with a unique peak.

To approach more complex distributions, for instance distributions with multiple modes, mixture models are used. It is a weighted sum of simple probability densities. The non-negative weights sum to 1, so that the combination is convex and the mixture has a valid probability density function. The PDF of a Gaussian Mixture Model with N components is defined as:

$$f_X(x) = \sum_{k=1}^{N} \pi_k \mathcal{N}\left(x; \mu_k, \Sigma_k\right), \tag{2}$$

with $x \in \mathbb{R}^p$, π_k the weights (non-negative values summing up to 1), and μ_k and Σ_k parameters of the N Gaussian components. GMM is a universal approximator, which means that it can approximate exactly any density when $N \to \infty$.

Parameters of the GMM are usually learned with the Expectation-Maximization (EM) procedure. Let $\Theta = (\theta_1, \dots, \theta_l)$ be a collection of l observations that we want to approximate with a GMM. The aim is to maximize the log-likelihood of the model for the dataset Θ. With the classical assumption that entries of the dataset $\theta_{j, j=1,\dots,l}$ are independent and identically distributed, the log-likelihood to be maximized is:

$$\ln(p(\Theta|\boldsymbol{\mu}, \boldsymbol{\Sigma})) = \sum_{j=1}^{l} \ln\left(\sum_{k=1}^{N} \pi_k \mathcal{N}(\theta_j; \mu_k, \Sigma_k)\right), \tag{3}$$

with $\boldsymbol{\mu} = \{\mu_1, \dots, \mu_N\}$ and $\boldsymbol{\Sigma} = \{\Sigma_1, \dots, \Sigma_N\}$.

In the EM algorithm, we need to compute the derivative of this log-likelihood with respect to the GMM parameters, for example with respect to the mean μ_k

as shown below:

$$\frac{\partial \ln(p(\Theta|\boldsymbol{\mu}, \boldsymbol{\Sigma}))}{\partial \mu_k} = \sum_{j=1}^{l} \left(\frac{\pi_k \mathcal{N}(\theta_j; \mu_k, \Sigma_k)}{\sum_{s=1}^{N} \pi_s \mathcal{N}(\theta_j; \mu_s, \Sigma_s)} \right) \Sigma_k^{-1} (\theta_j - \mu_k). \tag{4}$$

The term after the sum sign inside the parenthesis in the above equation is named *responsibility*. Responsibilities are latent variables r_{jk} such that $r_{jk} = 1$ if the component k has generated the sample θ_j (and 0 otherwise). It can be shown using Bayes theorem that:

$$P(k|\theta_j) = \frac{P(k)P(\theta_j|k)}{P(\theta_j)} = \frac{\pi_k \mathcal{N}(\theta_j; \mu_k, \Sigma_k)}{\sum_{s=1}^{N} \pi_s \mathcal{N}(\theta_j; \mu_s, \Sigma_s)} = r_{jk}. \tag{5}$$

These very same responsibilities also appear in the partial derivative of the log-likelihood with respect to the weights π_k or the covariance matrices Σ_k. Therefore, to be able to *maximize* log-likelihood and find optimal values for GMM parameters π_k, μ_k and Σ_k, a first step of calculating responsibilities has to be performed. This step is called *Expectation* step (*E* step). During this *E* step π_k, μ_k and Σ_k are assumed to be known. Then, during the *M* step (*Maximization* step), responsibilities are assumed to be known and model parameters are updated. These two steps are performed alternatively until reaching a convergence criterion. This criterion is usually based on a minimum threshold for improvement of the log-likelihood. Main limitation of the EM algorithm is the sensitivity to the initialization of π_k, μ_k and Σ_k. Initialization can be performed either randomly or with the help of an auxiliary algorithm such as k-means. In both cases multiple initializations are performed to find and keep only the one with the highest log-likelihood. GMM contains a number of parameters to be estimated. More precisely, we have $p \times (p+1)/2$ independent parameters for each covariance matrix, p parameters for each mean vector and N weights, so a total of $N \times \left(\frac{1}{2}p^2 + \frac{3}{2}p + 1\right)$ independent parameters for a GMM with N components of p-dimensional Gaussian distributions. A common trick to reduce the number of parameters is to enforce a particular structure to covariance matrices (diagonal, spherical, *etc.*) with a trade-off on the loss of the richness of the captured correlation relationships.

In our method, we use GMM as a statistical model to describe the likelihood of small image patches of 8×8 pixels (*cf.* Equation 2). For source/target disentangling, we compare empirical distribution of log-likelihood, on patches from the identified pristine (*i.e.,* background) region and those from copy-moved areas (*i.e.,* candidate source and target areas). This is detailed in the next subsection.

3.2. The proposed method

A graphical overview of the main steps of our method is shown in Figure 2. Input of our method is the given image to be analyzed and the binary mask produced by a first-stage detector in which duplicated areas are identified indifferently. To discern source and target areas, we first extract patches from

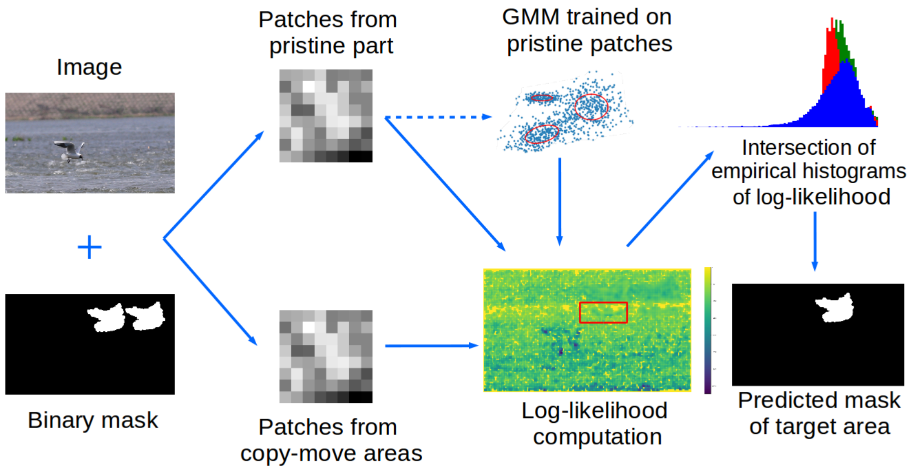


Image
+
Binary mask
Patches from pristine part
Patches from copy-move areas
GMM trained on pristine patches
Log-likelihood computation
Intersection of empirical histograms of log-likelihood
Predicted mask of target area

Figure 2: Graphical summary of the main steps of our method to disentangle copy-moved source and target areas. The red histogram of target area, which is corrected identified in the final mask, has less intersection with the blue histogram of pristine part than the source area (green histogram). This is also somehow partially reflected by the small blue band close to the boundary of target area in the log-likelihood map, as highlighted by the red rectangle (zoom-in of the digital version is recommended).

the identified pristine region (black region in binary mask) and train a GMM to represent the statistics of pristine patches. The next step is to construct the empirical histogram of patch log-likelihood of all the connected components (CCs) in the binary mask, including the pristine region as well as the candidate source/target areas (white CCs in binary mask). We then compute the *intersection* between the histogram of pristine region and that of each candidate source/target area. At last, we consider the candidate area with the largest intersection as the source area and the other(s) as target.

One important intuition motivating the design of our method is that the boundaries of target area would expose locally some slight statistical deviation from rest of image. Authors of BusterNet [2] make the assumption that target area has been manipulated and they rely on manipulation detection to discriminate the two zones. We make a weaker and more general assumption which measures the statistical deviation of local image patches. We believe that by comparisons of likelihoods of duplicated areas we are able to distinguish source and target. First, assuming no manipulation on target area, the log-likelihood values for interior patches of copy-moved areas would intersect largely with the values of the pristine region, whereas log-likelihood values on boundaries for the target area would deviate. Further, when target area is manipulated, the statistical deviation would be even bigger for patches from the whole target area, including both interiors and boundaries. So the area with the log-likelihood values the closest to the pristine area is considered as the original (source) one and the one that deviates the most would be the target.

Our approach relies on Gaussian Mixture Model because it is a very good candidate for modelling natural image statistics [29]. It is inspired by the one used for image manipulation detection in our previous work [30]. Here in this paper, a single GMM, instead of two GMMs in [30], is trained on 8×8 centered patches (DC component is removed, resulting in continuous numbers) coming from a specific image. The patches are extracted from the identified pristine region outside of the duplicated areas as predicated by first-stage copy-move detector. The GMM is trained by using the EM algorithm which maximizes the model's log-likelihood of Equation 3 on the pristine patches. Therefore, this GMM should capture the regular local statistics of the image. In our work, GMM is simpler and has less parameters with 25 components compared to the 200 components for manipulation detection in [30]. Experimentally, it is not necessary to have a very high capacity of description because now a single image is considered and not a big image database as in [30, 29]. Each GMM component has a full covariance matrix as we intend to capture the subtle dependence between pixels in patches, but not only simplified dependence as diagonal covariance matrices would do.

The idea is then to measure the statistical deviation of each duplicated area from the regular statistics of the image as described by the GMM trained on pristine patches. In practice, we compute and compare the distance between empirical distribution of log-likelihood on patches in each candidate source/target area and the distribution in the identified pristine part. For the distance computation and comparison, we have used a simple but effective distance measure which is borrowed from [31] and which is based on the number of elements in the intersection of two normalized histograms. This intersection-based metric was to our knowledge first proposed in [31] with the name of *overlapping coefficient* because it attempts to measure the overlapping between two distributions. The metric is later used in various applications, for instance the well-known color-based image indexing method of [32]. Concretely, the adopted histogram intersection metric is defined as:

$$intersect(h^c, h^p) = \sum_{i=1}^{n} \min(h_i^c, h_i^p), \tag{6}$$

with h^c the normalized histogram of patch log-likelihood for one candidate source/target area, h^p the normalized histogram for the pristine part, and n the number of bins. As histograms are normalized, it is not necessary to (re)normalize this intersection score which is naturally between 0 and 1. The area with the largest intersection with the predicted pristine part is considered as source and the other one(s) as target. Width of histogram bins is set automatically to have 75 bins within the range of log-likelihood values of patches of the whole image, from both pristine and copy-moved areas. The number of 75 bins has been set empirically, based on the observation of histograms like those in the last row of Figure 3. The method is not very sensitive to this number, a larger number of bins does not improve significantly the results, while a rough histogram with less than 50 bins starts decreasing the performance. It is worth mentioning that there are a lot of metrics for measuring the distance between histograms, or between distributions in a broad sense: Bhattacharyya distance, Kolmogorov-Smirnov statistic, Kullback-Leibler divergence, Hellinger distance, and Chi-Squared distance, just to name a few. The intersection-based distance of Equation 6, which is adopted in our method, is probably one of the simplest metrics and has been commonly used in the literature [31, 32]. We have explored experimentally two other measures of Kolmogorov-Smirnov statistic and Chi-Squared distance. They do not provide clearly better empirical results but are more complex and more costly to compute. We have also tried to compare empirical normalized histograms through comparisons of straightforward statistics such as the mean or the median. However these statistics are in some cases not strong enough to capture the differences. Therefore, the adopted metric of histogram intersection appears to be a good technical choice in terms of both simplicity and effectiveness.

Some examples of log-likelihood histograms and output masks of target area on CASIA2 dataset [1] are shown in Figure 3, with the well-known dense-field copy-move forgery detector (hereafter abbreviated as DF-CMFD) [13] as the first-stage detector. In the last column of Figure 3, histograms for the two areas are very close which led to an error of classification. When the target area has been manipulated, for example in the third column a small deformation has been applied according to the specification of the CASIA2 dataset, the difference in histogram intersections can be more visible.

4. Experiments

4.1. Metrics and datasets

We designate as "Discernible" all the samples with at least two white connected components (CCs) in the binary mask, and these CCs should intersect with source and target areas in ground-truth mask. Remaining samples are in the "Indiscernible" subset, because technically it is not possible for a method to disentangle source and target for samples in this subset. This subset mainly comprises samples for which the binary mask has no white CC (*i.e.*, no duplicated area detected) or the CCs do not intersect with source or target area in ground-truth mask (*i.e.*, false positives of wrongly detected duplicated areas). Samples with only one white CC cannot be disentangled neither so they are also considered as "Indiscernible". To measure the discerning capability of source and target, we compute the *overall accuracy* and the accuracy on the "Discernible" subset, as the ratio of number of images with correct source/target classification to the number of images in the whole dataset or in the "Discernible" subset.

Authors of BusterNet [2] consider a "Miss" subset composed of predicted masks that do not intersect with the source and target area in ground-truth mask. This "Miss" subset is included in our "Indiscernible" subset defined above. In [2] authors also define another subset of images they call "Opt-Out". It includes samples with final predicted masks in which all duplicated CCs receive a same label (source or target) as given by BusterNet. We consider that

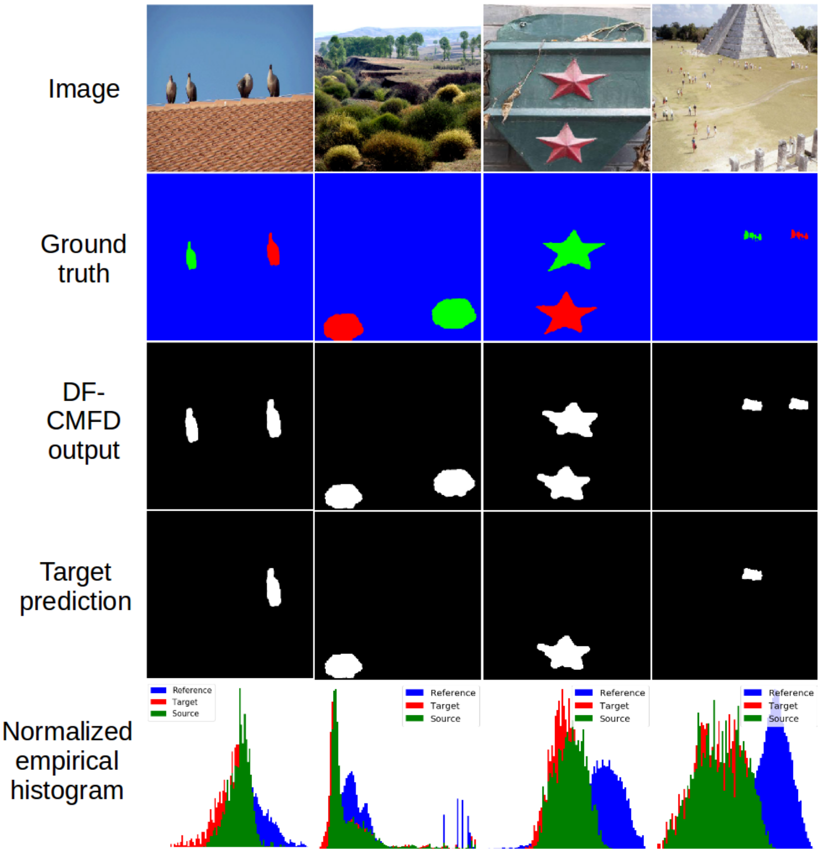


Figure 3: Example results of our method on CASIA2 dataset [1] with DF-CMFD [13] as first stage detector. Red in ground-truth masks indicates target area, green the source one, and blue the pristine part. Example in the last column is a miss-classification.

labeling all CCs with same label as a miss-classification for discerning source and target. In our setting such samples are naturally in the "Discernible" subset defined in the last paragraph, and performance is evaluated on them as well as other images in the subset. Finally, in [2], "Opt-In" is the subset comprising all remaining images except those in the "Miss" and the "Opt-Out". Authors report performance on these "Opt-In" samples for which BusterNet attributes two distinct labels to ground-truth source and target regions. We think that the "Opt-In" samples in [2] represent a subset where BusterNet classifier performs the best. Therefore, for a fair comparison, we consider this subset of images as a "*Favorable Subset*". For our method we compute a ranking to extract images where the classification is the most certain to also obtain a "Favorable Subset". This ranking is based on the descending order of a score (a kind of Michelson contrast) computed on images with binary masks of two duplicated zones, as:

$$score = \frac{|intersect(h^{c1}, h^{p}) - intersect(h^{c2}, h^{p})|}{intersect(h^{c1}, h^{p}) + intersect(h^{c2}, h^{p})}, \tag{7}$$

with $intersect(.,.)$ as defined in Equation 6, and h^{c1}, h^{c2} and h^{p} the normalized empirical histogram for respectively the first zone, the second one and the pristine part. A higher score implies that one duplicated area exposes a considerably bigger deviation from the pristine background when compared to the other one. It could be that the target area has been manipulated to create a visually convincing copy-move forgery. We set a threshold on the score in order to have the same number as the "Opt-In" images defined by BusterNet, *e.g.,* 190 for CASIA2 dataset. A ranking is particularly interesting from an operational point of view, it makes possible to determine the most suspicious images among a large dataset. Finally, as expected, images with a high ranking score are less likely to be from the "Indiscernible" subset of images. For instance, on CASIA2 dataset for our GMM-based method with DF-CMFD binary masks, among the 200 images with the highest scores of Equation 7 only 5 are in the "Indiscernible" subset. Still from an operational point of view, it could help us to detect miss-detection of the first-stage copy-move detector.

It is worth mentioning that although we define the "Discernible" subset and the "Favorable Subset" and will report later in this section the performance of source and target disambiguation on these subsets, the two most important and intuitive metrics are the number of images with correct source/target classification and the corresponding overall accuracy on the whole dataset. In addition, we think that performance on the "Opt-In" subset as defined in the BusterNet paper [2] would not be a very fair metric because such subset depends on the technical details of the source/target classification method. In the BusterNet method, this "Opt-In" subset only considers images with ground-truth source and target regions that receive distinct source/target labels as given by BusterNet (in fact BusterNet in quite a few cases attributes same label to both source and target areas, *i.e.*, no decision). We consider such subset as a "Favorable Subset" as explained in the last paragraph. In contrast, the performance on the "Discernible" subset that we have defined at the beginning of this subsection appears to be fairer and more objective, which reflects how well a source/target classification method behaves on a subset of samples for which it is in theory possible to achieve a correct source and target disambiguation.

We test our proposed method on copy-move images from CASIA2 [1] and CoMoFoD [9] datasets. We use ground-truth masks from authors of BusterNet[1] for the two datasets. These ground-truth masks provide information on source and target areas of copy-move: the red channel of masks contains information about target area, the green channel on source area, and the blue one on pristine part of images (*cf.,* the second row of Figure 3).

[1] `https://github.com/isi-vista/BusterNet`

Table 1: Analysis and comparisons of copy-move detection and localization performance on CASIA2 dataset (source and target are identified indifferently). "HP" means hyper-parameters. Please refer to [2] for details of the three evaluation protocols. For each protocol and evaluation metric, the highest value among the compared methods is highlighted in bold.

	Methods						**DF-CMFD [13]**	
	[2]	[12]	[10]	[24]	[23]	[25]	**Default HP**	**Tuned HP**
	Image Level Evaluation Protocol							
Precision	78.22%	97.01%	68.49%	66.37%	-	-	**97.64%**	71.44%
Recall	73.89%	24.47%	67.82%	73.59%	-	-	56.81%	**88.80%**
F1-score	75.95%	39.08%	68.15%	69.80%	-	-	71.83%	**79.18%**
	Pixel Level Evaluation Protocol – A							
Precision	77.38%	**94.46%**	64.84%	17.06%	-	-	86.20%	92.47%
Recall	59.15%	25.05%	0.17%	10.60%	-	-	64.43%	**65.85%**
F1-score	67.05%	39.59%	0.34%	13.08%	-	-	71.62%	**76.92%**
	Pixel Level Evaluation Protocol – B							
Precision	55.71%	22.71%	37.09%	23.97%	**70.85%**	58.32%	48.18%	67.32%
Recall	43.83%	13.36%	0.14%	13.79%	58.85%	37.33%	49.40%	**73.93%**
F1-score	45.56%	16.40%	0.23%	14.64%	64.29%	45.52%	47.88%	**67.97%**

4.2. DF-CMFD as first-stage detector

With the aim to produce a better pipeline for copy-move detection with source and target disentangling, we look for the best possible first-stage copy-move detector available. Performances for some state-of-the-art detectors on CASIA2 dataset are reported in Table 1. Reported performances are in terms of precision, recall, and F1-score, with F1-score being a more informative metric computed as the harmonic mean of precision and recall. Results for methods [12], [10] (SIFT-based methods) and [24] are extracted from the BusterNet paper [2]. Details of transformation of [24] from splicing to copy-move detector can be found in [2]. We also report results extracted from the original papers of two recent deep-learning-based methods: the DenseNet-based method in [23] and the method in [25] which is based on the so-called AR-Net (Adaptive attention and Residual refinement Network). We were unable to reproduce the lower scores reported in [2] for DF-CMFD [13], but our results (the column of "DF-CMFD – Default HP" in Table 1) are coherent with reported performance of DF-CMFD in [23] (Table III of that paper, column of "PM"). We have used the implementation of DF-CMFD by original authors which is available on-line[2].

To perform source/target discrimination, it is obviously better to have a detector that correctly identifies a large number of copy-move images. This is mainly reflected in Table 1 by the F1-scores at "Image Level" and under the "Pixel Level Evaluation Protocol – A". Following this criterion, DF-CMFD [13] appears a good candidate to be first-stage detector. However, one drawback is that default parameters of DF-CMFD are better suited for big images as considered in the method's original paper [13]. For instance minimum size considered for clones is 1200 pixels. This represents a quite big area for images of sizes

[2] `http://www.grip.unina.it/research/83-multimedia_forensics/90-copy-move-forgery.html`

from 240 × 160 to 900 × 600 pixels of CASIA2 dataset. Features are extracted in circle regions of diameter of 16 pixels, which is also large for small images. Therefore this setting is unfavorable because predicted copy-move images and masks are not very accurate. As reported in Table 1, F1-score for "Pixel Level Evaluation Protocol – B" (localization performance on images with copy-move forgery) for CASIA2 dataset is only of 47.88%. Therefore, to obtain a higher score, it is better to tune the hyper-parameters of DF-CMFD.

We tune empirically the hyper-parameters with regard to the size of images. We reduce the feature size, the minimum size of clones and distance between clones. For instance for CASIA2, we have empirically selected a feature size of 12, a minimum size of 225 pixels and a minimum distance of 160 pixels. Results of this tuned DF-CMFD can be found in the last column of "DF-CMFD – Tuned HP" in Table 1, which has the best F1-score for all the three evaluation protocols. In particular, scores for "Pixel Level Evaluation Protocol – B" are largely improved, reflecting a more accurate localization. More accurate masks imply that the ground-truth boundaries and interiors of copy-moved areas are more surely located, which is beneficial for our method. It is possible to obtain even better score by cross-validation on the hyper-parameters. However, to follow a realistic operational scenario, we use the empirical tuning and set the tuned DF-CMFD as first-stage copy-move detector in the following experiments.

4.3. Results of source and target disentangling

Results of BusterNet method for discerning source and target on CASIA2 and CoMoFoD datasets are presented in the second block of Table 2. We extract the number of images with correct source/target disambiguation (146 and 33 for respectively CASIA2 and CoMoFoD) and the corresponding overall accuracy on whole dataset (11.11% and 16.50% for respectively CASIA2 and CoMoFoD) from Table 5 of the original paper of BusterNet [2], and report these results in the columns of "Number of images – Cor." and "Accuracy – Overall" in Table 2. As mentioned in subsection 4.1, they are the most important evaluation metrics for source and target discrimination; for the sake of clarity the corresponding two columns in Table 2 are shaded, and the best values are highlighted in bold. As explained earlier also in subsection 4.1, the "Opt-In" subset in Table 5 of [2] corresponds here to a subset we call "Favorable Subset" ("F.S." in Table 2) on which BusterNet has its best performance. We extract the number of images in this "Opt-In" subset (190 images for CASIA2 and 41 images for CoMoFoD) and the corresponding "Opt-In" accuracy (76.84% for CASIA2 and 80.49% for CoMoFoD) from Table 5 of [2], and report them respectively in parentheses of the column "Number of images – Cor.(/F.S.)" and the column of "Accuracy – F.S." in Table 2. The "Miss" and "Opt-Out" in Table 5 of [2] are subsets on which BusterNet is not able to make correct source and target classification (so the accuracy is 0% on "Miss" and "Opt-Out" for BusterNet); here we omit these two subsets because they are complementary information to the "Opt-In" (*i.e.*, "F.S.") subset. In addition, we downloaded the resulting masks of BusterNet shared on-line by original authors (*cf.*, link in footnote 1 on page 17) which correspond to results in Table 5 of [2], and have used these masks to

Table 2: Source/target discernibility performances of our GMM-based method, BusterNet [2], multi-branch CNN [28], and the LBP-based method [26]. "Indisc." means the "Indiscernible" subset, "Disc." the "Discernible" subset, "Cor." images with correct classification of source and target, "F.S." the "Favorable Subset". Please refer to subsection 4.1 for details of evaluation metrics used in the table. In the column of "Cor.(/F.S.)", numbers are presented in the following format: NumberOfCorrectlyClassifiedInF.S.(/NumberInF.S.). The two columns of "Number of images – Cor." and "Accuracy – Overall" are shaded to show that these are the two most important evaluation metrics for source and target disambiguation. The highest values for theses two metrics among the four methods are highlighted in bold.

		Number of images					**Accuracy**		
	Dataset	**Total**	**Indisc.**	**Disc.**	**Cor.**	**Cor.(/F.S.)**	**Overall**	**Disc.**	**F.S.**
Our method with tuned DF-CMFD	**CASIA2**	1313	489	824	**549**	149(/190)	**41.81**%	66.63%	78.42%
	CoMoFoD	200	87	113	**69**	33(/41)	**34.50**%	61.06%	80.49%
BusterNet [2]	**CASIA2**	1313	557	756	146	146(/190)	11.11%	19.31%	76.84%
	CoMoFoD	200	78	122	33	33(/41)	16.50%	27.05%	80.49%
Multi-branch CNN [28]	**CASIA2**	1313	489	824	536	-	40.82%	65.05%	-
	CoMoFoD	200	87	113	62	-	31.00%	54.87%	-
LBP-based method [26]	**CASIA2**	1313	489	824	487	-	37.09%	59.10%	-
	CoMoFoD	200	87	113	52	-	26.00%	46.02%	-

compute the number of samples in the "Discernible" and "Indiscernible" subsets for BusterNet as shown respectively in the column of "Number of images – Disc." and "Number of images – Indisc." in Table 2. We also calculate and report the accuracy on the "Discernible" subset in the column of "Accuracy – Disc." in Table 2. As explained in subsection 4.1, the accuracy on "Discernible" subset appears to be a more objective metric than accuracy on "F.S.", because the "Discernible" subset comprises samples on which it is in theory possible to carry out correct source and target disambiguation. From Table 2 it can be observed that on CASIA2 dataset, BusterNet has an accuracy of 19.31% on the "Discernible" subset and an overall accuracy of 11.11% which are rather low, with in total 146 images with correct source and target discrimination. These 146 correctly classified images are naturally all in the "F.S." subset for BusterNet, therefore the method has a high accuracy of 76.84% on the 190 favorable samples in the "F.S." subset which is satisfactory.

Results of our GMM-based method with binary masks from the tuned DF-CMFD are presented in the first block of Table 2. We have competitive results on "Favorable Subset" against BusterNet: an accuracy of 78.42% on CASIA2, and 80.49% on CoMoFoD. More importantly, on the two datasets, the number of images with correct source/target disentanglement of our method is significantly higher than that of BusterNet (549 vs. 146 on CASIA2, and 69 vs. 33 on CoMoFoD). The overall accuracy is also much better for our method (41.81% vs. 11.11% on CASIA2, and 34.50% vs. 16.50% on CoMoFoD). As mentioned in subsection 4.1, these are the two most important metrics when assessing a source/target disambiguation method. In addition, we can notice that the size of "Discernible" subset is rather comparable between our method and BusterNet and that performances on this subset are largely upgraded in our method (please compare the columns of "Number of images – Disc." and "Accuracy – Disc." for the first two blocks in Table 2).

We also provide results for the deep-learning-based method from the recent arXiv pre-print [28]. We use code from the authors[3]. For a fair comparison, we feed to the CNN of [28] the same masks from tuned DF-CMFD as used by our method. It should be favorable for the CNN to use a more accurate first-stage detector. Indeed, with masks produced by tuned DF-CMFD the multi-branch CNN method [28] is able to correctly identify source and target on a much larger number of images on CASIA2 than the result reported in the original pre-print [28]. Since the number of images with correctly classified source and target areas is one of the most important evaluation metrics for source/target disambiguation, therefore in Table 2 we compare with a stronger version of Barni *et al.*'s CNN-based method [28] than its original version, which is favorable for their method. More precisely, according to Table VII of the pre-print [28], we can compute the number of images with correctly classified source and target as $482 \times 74.04\% \approx 357$, with 482 being the number of samples in a customized "Opt-In" subset and 74.04% being the accuracy on this subset. The "Opt-In" subset of the multi-branch CNN method [28] is different from "Opt-In" of BusterNet but is also related to technical details of the source and target discrimination method (please refer to [28] for details). In this paper and our experiments, with binary masks of tuned DF-CMFD, now the stronger version of multi-branch CNN method can correctly discern source and target on 536 images on CASIA2 as shown in the third block of Table 2, column of "Number of images – Cor.". This number is much higher than 357 correctly classified images as achieved in the original arXiv pre-print [28] where a weaker first-stage detector was used. Accordingly, the overall accuracy of the multi-branch CNN method is also much higher for the stronger version considered and used in our experiments. Regarding the comparison between our method and the method of [28], there is no significant difference in the number of correct images on the two datasets, *i.e.,* 549 vs. 536 and 69 vs. 62, our method being slightly better. The performance is lower for both methods on CoMoFoD, probably due to the higher difficulty of this dataset. In CoMoFoD, contrary to CASIA2, there is no manipulation, *e.g.,* scaling, on the target area.

At last, we carry out comparison with the LBP-based method proposed in another arXiv pre-print [26]. As shown in the last block of Table 2, the performance (in terms of number of images with correctly classified source/target and overall accuracy) of this feature-based method is lower than our GMM-based method and the multi-branch CNN method, especially on CoMoFoD. Our explanation is that the assumption of the LBP-based method, *i.e.*, boundary of target area has been smoothed, is rather restrictive. This smoothing operation is not always applied on target boundaries, *e.g.*, on CoMoFoD images. Sometimes the target area can be "naturally" inserted at a target position without boundary smoothing, for instance by leveraging the visual masking effect induced by rich textures at neighboring regions which is compliant to the free-energy principle of human brains [33]. In addition, even with the presence of boundary smoothing

[3] `https://github.com/andreacos/MultiBranch_CNNCopyMove_Disambiguation`

of target zone, it cannot be guaranteed that the proposed LBP-based feature can always achieve a correct discrimination between source and target areas.

Additionally, with curiosity, we did some post-experimental analysis of our GMM-based method and found that we are able to reach almost 100% of accuracy for source and target disentangling when the predicted mask from DF-CMFD shares at least 50% of duplicated areas with the ground-truth. We only mention this result as illustration of the importance of the localization accuracy of the first-stage detector. It is not a realistic scenario as it requires access to ground-truth masks. In contrast, although some images are in theory "Discernible", the source/target disentangling is very difficult on them because of poorly identified duplicated areas and areas of false alarms in the mask of first-stage detector. This results in a success rate lower than 50% on such difficult images for both our method and [28], *e.g.,* mistakenly attributing reversed labels or two same labels to ground-truth source and target areas. More efforts shall be devoted to the study of these difficult cases.

4.4. Discussion

We were able to improve largely, when compared to BusterNet, the overall accuracy for source and target disentangling on CASIA2 and CoMoFoD datasets. One possible explanation is that contrary to BusterNet that relies on manipulation detection to locate and distinguish between source and target, in our method we assume that the boundary of the target area would expose statistical deviation. This deviation would be even larger if the interior of the target area is manipulated. A major limitation of BusterNet is that it tends to attribute same label for both source and target areas, *i.e.,* no decision, while our method produces decision on more images. In fact, BusterNet produces a decision only for 14.5% of the CASIA2 images. At last, BusterNet method [2] does not enforce pixel correspondence in source and target. DF-CMFD imposes such prior which is beneficial for the method based on it, ours and [28]. Beside that, BusterNet uses 256×256 images as input due to memory limitation, therefore test images are resized prior to being fed to the network. We can consider that this resizing acts as a post-processing, which would remove part of fingerprints left by manipulation on target area or abnormal transitions between target and pristine areas. In contrast, our method is able to process full-sized images to discriminate source and target.

Comparable results on CASIA2 dataset of our method and [28] could be explained by a similarity of the two methods: GMM-based or CNN-based approach driven by information from the copy-moved zones and boundaries. It seems, according to the results on CoMoFoD dataset, that our method is slightly better when no manipulation has been added to target area, *i.e.,* when statistical deviation is smaller. The method in [28] needs a large synthetic dataset for network training, while our GMMs are specific for each image. This probably allows us to capture more subtle differences. This *single image setting* also makes the development lighter as the training process is simpler and requires less resources. By contrary, CNN-based methods need a large amount of training data. For instance, authors of BusterNet [2] have produced 100000 synthetic samples for the training of their network. Two additional third-party image tampering datasets were also used for training. The network of [28] has been trained on a synthetic dataset of 900000 samples. We can observe that contrary to the two deep-learning-based methods, no labels are used in the training of our method with the statistical machine learning tool of GMM. Loosely speaking, our approach could be considered like an unsupervised method working on a single image basis. This provides additional flexibility compared to the two CNN-based methods. Usually classical machine learning tools achieve lower performance than recent deep-learning approaches. Here we do not observe this trend and we explain this mainly by the adaptability of our method to the individual given image. This interesting point is worth further investigation.

5. Conclusion

We propose a simple method to discriminate source and target areas in copy-move forgeries. Our approach acts as a second-stage detector and takes as input the binary mask produced by a first-stage copy-move detector. The basic idea is to measure and compare the statistical deviation of duplicated areas by using a Gaussian Mixture Model trained on identified pristine patches. We show that our method outperforms the only other published detector capable of such disentanglement, BusterNet [2], on two different datasets (CASIA2 and CoMoFoD). Another advantage of our method is the possibility to rank predictions among a dataset to extract images with the most surely distinguished source and target areas. As discussed in subsection 4.4, we consider that bringing additional flexibility and adaptability with the single image framework compared to the training of deep-learning methods on large synthetic datasets may be a key factor for good performance. Usually deep-learning methods are able to achieve (much) better results than classical machine learning tools, such as GMMs, but this is not the case here. It is thus a promising line of research to find solutions to make deep-learning methods more flexible and adaptive, probably in a single image setting and using few labeled samples.

As future work, other statistical models and histogram distance measures could be considered. Another possibility would be an approach without statistical modeling (*i.e.,* the Gaussian Mixture Model), but directly with the pixel values. This would then probably require more advanced distance measures such as Wasserstein or MMD (Maximum Mean Discrepancy). A metric could also be learned with a neural network. Siamese networks seem especially promising, such as those used in [28]. We plan to test our method combined with more first-stage detectors to study their impact. There is also room for improvement in the post-processing of first-stage detector by using useful information provided by our method, *e.g.,* the ranking scores. It could help to identify and discard the false alarms of the first-stage detector.

Acknowledgment

This research work is financially supported by French National Research Agency (DEFALS ANR-16-DEFA-0003, ANR-15-IDEX-02).

References

[1] J. Dong, W. Wang, T. Tan, CASIA image tampering detection evaluation database, in: Proc. of the IEEE China Summit and International Conference on Signal and Information Processing, 2013, pp. 422–426.

[2] Y. Wu, W. Abd-Almageed, P. Natarajan, BusterNet: Detecting copy-move image forgery with source/target localization, in: Proc. of the European Conference on Computer Vision, 2018, pp. 170–186.

[3] M. Huh, A. Liu, A. Owens, A. A. Efros, Fighting fake news: Image splice detection via learned self-consistency, in: Proc. of the European Conference on Computer Vision, 2018, pp. 101–117.

[4] T. Pomari, G. Ruppert, E. Rezende, A. Rocha, T. Carvalho, Image splicing detection through illumination inconsistencies and deep learning, in: Proc. of the IEEE International Conference on Image Processing, 2018, pp. 3788–3792.

[5] Y. Rao, J. Ni, H. Zhao, Deep learning local descriptor for image splicing detection and localization, IEEE Access 8 (2020) 25611–25625.

[6] X. Zhu, Y. Qian, X. Zhao, B. Sun, Y. Sun, A deep learning approach to patch-based image inpainting forensics, Signal Processing: Image Communication 67 (2018) 90–99.

[7] H. Li, J. Huang, Localization of deep inpainting using high-pass fully convolutional network, in: Proc. of the IEEE International Conference on Computer Vision, 2019, pp. 8301–8310.

[8] X. Wang, S. Niu, H. Wang, Image inpainting detection based on multi-task deep learning network, IETE Technical Review (2020) 1–9.

[9] D. Tralic, I. Zupancic, S. Grgic, M. Grgic, CoMoFoD – New database for copy-move forgery detection, in: Proc. of the International Symposium on Electronics in Marine, 2013, pp. 1–6.

[10] V. Christlein, C. Riess, J. Jordan, C. Riess, E. Angelopoulou, An evaluation of popular copy-move forgery detection approaches, IEEE Trans. on Information Forensics and Security 7 (6) (2012) 1841–1854.

[11] J. Fridrich, D. Soukal, J. Luk, Detection of copy-move forgery in digital images, in: Proc. of the Digital Forensic Research Workshop, 2003, pp. 1–10.

[12] S.-J. Ryu, M.-J. Lee, H.-K. Lee, Detection of copy-rotate-move forgery using Zernike moments, in: Proc. of the International Workshop on Information Hiding, 2010, pp. 51–65.

[13] D. Cozzolino, G. Poggi, L. Verdoliva, Efficient dense-field copymove forgery detection, IEEE Trans. on Information Forensics and Security 10 (11) (2015) 2284–2297.

[14] K. B. Meena, V. Tyagi, A copy-move image forgery detection technique based on Tetrolet transform, Journal of Information Security and Applications 52 (2020) 102481:1–9.

[15] A. Langille, M. Gong, An efficient match-based duplication detection algorithm, in: Proc. of the Canadian Conference on Computer and Robot Vision, 2006, pp. 64:1–64:8.

[16] C. Barnes, E. Shechtman, A. Finkelstein, D. B. Goldman, Patchmatch: a randomized correspondence algorithm for structural image editing, ACM Trans. on Graphics 28 (3) (2009) 24:1–24:10.

[17] X. Pan, S. Lyu, Region duplication detection using image feature matching, IEEE Trans. on Information Forensics and Security 5 (4) (2011) 857–867.

[18] B. Shivakumar, S. Baboo, Detection of region duplication forgery in digital images using SURF, International Journal of Computer Science Issues 8 (4) (2011) 199–205.

[19] K. B. Meena, V. Tyagi, A hybrid copy-move image forgery detection technique based on Fourier-Mellin and scale invariant feature transforms, Multimedia Tools and Applications 79 (2020) 8197–8212.

[20] J. Ouyang, Y. Liu, M. Liao, Copy-move forgery detection based on deep learning, in: Proc. of the International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, 2017, pp. 1–5.

[21] Y. Liu, Q. Guan, X. Zhao, Copy-move forgery detection based on convolutional kernel network, Multimedia Tools and Applications 77 (14) (2018) 18269–18293.

[22] M. Zampoglou, S. Papadopoulos, Y. Kompatsiaris, Detecting image splicing in the wild (web), in: Proc. of the IEEE International Conference on Multimedia & Expo Workshops, 2015, pp. 1–6.

[23] J. Zhong, C. Pun, An end-to-end Dense-InceptionNet for image copy-move forgery detection, IEEE Trans. on Information Forensics and Security 15 (2019) 2134–2146.

[24] Y. Wu, W. Abd-Almageed, P. Natarajan, Deep matching and validation network: An end-to-end solution to constrained image splicing localization and detection, in: Proc. of the ACM International Conference on Multimedia, 2017, pp. 1480–1502.

[25] Y. Zhu, C. Chen, G. Yan, Y. Guo, Y. Y. Dong, AR-Net: Adaptive attention and residual refinement network for copy-move forgery detection, IEEE Trans. on Industrial Informatics 16 (10) (2020) 6714–6723.

[26] S. Salehi, A. Mahmoodi-Aznaveh, Discriminating original region from duplicated one in copy-move forgery, arXiv:1903.07044 CoRR (2019) 1–6.

[27] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recognition 29 (1) (1996) 51–59.

[28] M. Barni, Q.-T. Phan, B. Tondi, Copy move source-target disambiguation through multi-branch CNNs, arXiv:1912.12640 CoRR (2019) 1–15.

[29] D. Zoran, Y. Weiss, From learning models of natural image patches to whole image restoration, in: Proc. of the IEEE International Conference on Computer Vision, 2011, pp. 479–486.

[30] W. Fan, K. Wang, F. Cayre, General-purpose image forensics using patch likelihood under image statistical models, in: Proc. of the IEEE International Workshop on Information Forensics and Security, 2015, pp. 1–6.

[31] H. F. Inman, E. L. Bradley Jr, The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities, Communications in Statistics − Theory and Methods 18 (1989) 3851–3874.

[32] M. Swain, D. Ballard, Color indexing, International Journal of Computer Vision 7 (1991) 11–32.

[33] K. Friston, The free-energy principle: a unified brain theory?, Nature Reviews Neuroscience 11 (2) (2010) 127–138.