IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 19, 2019, accepted February 25, 2020, date of publication March 16, 2020, date of current version March 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2980992

GRAFT: Unsupervised Adaptation to Resizing for Detection of Image Manipulation

LUDOVIC DARMET, (Student Member, IEEE), KAI WANG, AND FRANÇOIS CAYRE
GIPSA-Lab, CNRS, Grenoble INP, Université Grenoble Alpes, 38000 Grenoble, France

Corresponding author: Ludovic Darmet (ludovic.darmet@gipsa-lab.grenoble-inp.fr).

This work was supported in part by the French National Research Agency (ANR) under projects DEFALS ANR-16-DEFA-0003 and ANR-15-IDEX-02.

ABSTRACT A large number of methods for forensics of image manipulation relies on detecting fingerprints in residuals or noises. Therefore, these detection methods are bound to be sensitive to noise generated by the image acquisition process, as well as any *pre-processing*. We show that a difference in pre-processing pipelines between training and testing sets induces performance losses for various classifiers. We focus on a particular pre-processing: *resizing*. It corresponds to a typical scenario where images may be resized (*e.g.*, downscaled to reduce storage) prior to being manipulated. This performance loss due to pre-resizing could be troublesome but has been rarely investigated in the image forensics field. We propose a new and effective adaptation method for one state-of-the-art image manipulation detection pipeline, and we call our proposed method Gaussian mixture model Resizing Adaptation by Fine-Tuning (GRAFT). Adaptation is performed in an *unsupervised* fashion, *i.e.*, without using any ground-truth label in the pre-resized testing domain, for the detection of image manipulation on very small patches. Experimental results show that the proposed GRAFT method can effectively improve the detection accuracy in this challenging scenario of unsupervised adaptation to resizing pre-processing.

INDEX TERMS Image forensics, manipulation detection, Gaussian mixture model, covariance matrices, likelihood maximization, domain adaptation.

I. INTRODUCTION

The broad availability of smartphones and digital cameras, combined with social platforms, has made sharing images easier than ever. Moreover, image editing software has become easy to use, even for the layman, and it is now available on smartphones. This situation both accounts for a tremendous revolution for communication and makes it harder than ever to trust images, as their integrity may be altered easily. Accordingly, forensics tools have been developed to help assess the source and integrity of digital images.

We list in Table 1 (Section V) some basic operations for image manipulation: Gaussian blurring, image sharpening, JPEG compression, *etc.* These basic operations are often *(i)* combined to build more advanced modifications or *(ii)* used to cover malicious operations like splicing or copy-move. For instance, the popular ''Photoshop Clone Stamp Tool'' uses Gaussian blurring to produce smoother fusion and transitions. Being able to detect these operations is a step toward detecting more complex and challenging tampering. Hence, this work focuses on these elementary operations.

A handful of effective detectors have been developed in the last decade [1], [2], targeting various basic image manipulation operations. These classifiers often rely on image statistical models, which may be altered by *a resizing prior to manipulation*. Most of public databases (both train and test sets) for image forensics are full-sized images, which may not reflect some typical scenarios in the wild. For example, the image to be manipulated may originate from a social media platform, where images are typically resized to save storage space. Note that similar concerns were first investigated in the steganalysis community [3], [4] under the name *cover-source mismatch*, though it is a broader problem than pre-resizing as it considers development process in general. It is very likely that image forensics researchers also need to tackle these issues. In this context, our first contribution in this paper is to provide evidence that such a mismatch of pre-resizing causes considerable loss in the detector

The associate editor coordinating the review of this manuscript and approving it for publication was Pierluigi Gallo.

performances. As an example, we observed a drop of 22% in accuracy when using one of the state-of-the-art methods [5] to detect JPEG compression (quality factor $Q = 90$) trained on regular patches but tested on samples that have undergone resizing with a factor of 1.25 before JPEG compression.

In a typical image forensics scenario, only part of an image would undergo a modification, which conveys a semantic meaning. Accordingly, forensics analysts are interested in obtaining a fine-grained forensic decision on patches, preferably on very small patches, *e.g.*, of 8×8 pixels. Hence, our *objective* in this paper is to resolve the issue caused by resizing pre-processing for the detection of image manipulation operations on such small patches.

In order to solve this problem, one would use a naive approach which trains a classifier with samples from various resizing factors. This approach is very resource consuming and has low flexibility. Moreover, performance drops as samples from different resizing factors are mixed together for training (results in Section V-E). Therefore, our work intends to provide adaptation shortcuts, especially in a scenario where no label is available for samples in the pre-resized target domain, *i.e.*, the so-called *unsupervised adaptation*. It is different from the *semi-supervised* or *weakly-supervised* frameworks where few labeled samples are supposed to be available in the pre-resized target domain. In *semi-supervised* framework unlabeled data on target domain are leveraged jointly with labeled ones, while in *weakly-supervised* framework only labeled data are used. *Unsupervised* framework is more challenging.

Contributions of this paper are summarized as follows:[1]

- We demonstrate performance loss due to resizing as a pre-processing operation prior to the basic image manipulations that we want to detect;
- We perform statistical tests on patches to better understand the cause of this performance loss;
- We propose an unsupervised method to adapt an already trained classifier to a new target pre-resized domain.

Section II presents related work on image forensics, steganalysis, and domain adaptation. Statistical tests about pre-resized images are reported and analyzed in Section III. Our proposed method GRAFT (Gaussian mixture model Resizing Adaptation by Fine-Tuning) is detailed in Section IV. Experimental results and comparisons are presented in Section V. Finally, we draw conclusions in Section VI.

II. RELATED WORK

A. IMAGE FORENSICS

Image forensics relates to security of digital images and acts as a passive image authentication approach. Image forensics problems include camera identification, image manipulation detection, identification of synthetic images, splicing detection, copy-move detection, *etc.* In this paper, we focus on the detection of elementary image manipulation operations (median filtering, sharpening, JPEG compression, *etc.*, see Table 1 of Section V for considered operations). Two types of methods have been developed for this problem:

1) Methods aimed at detecting a targeted and specific manipulation [7]–[13];
2) More recently, universal detectors of image manipulations (*i.e.* same algorithmic pipeline for every manipulations) have been proposed [5], [14]–[16].

In order to tackle the second and more difficult problem of universal detection, three approaches have been followed:

1) Using explicit image models to capture variations on small patches of the image (*e.g.*, a Gaussian Mixture Model, GMM) and carrying out comparison of likelihood values between GMMs trained on manipulated and pristine patches [5];
2) Computing features borrowed from steganalysis (mainly SPAM [17] and SRM [18]) as statistics of an implicit image model and using a classifier trained on the feature space [14], [15];
3) Using deep learning classifiers with constraints or specific processing for the first layer of the convolutional neural network (CNN) to extract residuals [16].

Our work aims at designing a universal manipulation detector in a challenging setting, where the testing set has undergone resizing as a pre-processing operation before image manipulation. Another challenging aspect is that the method intends to work with very small patches of 8×8 pixels, so as to have a fine-grained forensic decision. In addition, from a research point of view, it is also interesting to test limits of different methods by using smallest patches possible. We focus on the GMM-based approach of Fan *et al.* [5] because *(i)* the explicit image modeling can be carried out on 8×8 patches, much smaller than considered patches in other approaches [14]–[16], and *(ii)* as shown later in this paper the original GMM-based method [5] has the best overall baseline performance (*i.e.*, without resizing pre-processing) on 8×8 patches. We have recently explored this small patch setting and adaptation to pre-resizing in [19], although in a less difficult scenario of weakly-supervised adaptation (*i.e.*, a small number of labeled samples are available for the pre-resized target domain) and with different algorithms. Comparisons with [19] are presented in the experimental study of Section V.

At last, we have lately become aware of an arXiv preprint on domain adaptation in detecting synthetic images [20]. The authors of [20] designed an embedding that preserves generalization for an auto-encoder-based detector. Their method uses a few labeled samples in target domain to extend classification to a new class. Therefore, the goal of [20] and that of our work are related but different, and approaches remain distinct and not directly comparable, *i.e.* weakly-supervised with deep learning for detecting full-sized synthetic images for [20] *vs.* unsupervised with an explicit image model for

[1] A preliminary and short version of our method was informally presented in a local workshop (with a 4-page short summary in French, [6] in reference list). There are no formal published proceedings for this workshop.

detecting image manipulations on small patches for ours. These are to our knowledge the only weakly/semi-supervised methods proposed so far in image forensics.

B. STEGANALYSIS

As mentioned in Section I, researchers in steganalysis have carried out studies on the effect of downsampling [3]. They show some serious drops in performance when using a classifier on testing images that have been subject to a resizing before embedding a hidden message. These concerns over performance drops are known as cover-source mismatch and have been studied also in [4], [21], [22], in a more general setting of considering the effects of the whole image development process, not only resizing. The main differences with our adaptation approach are:

- The above steganalysis methods use handcrafted features, while we use explicit modeling of image data;
- Steganalysis tries to produce a decision for an image as a whole, while we work on very small patches;
- Existing solutions in steganalysis mainly consist in training a classifier with a bigger diversity of sources, combining different detectors, or training multiple classifiers and using the most appropriate one during testing, while we explicitly adapt classification model to the target domain in an unsupervised fashion.

It is worth mentioning that authors of [4] show that better results are obtained if they consider specifically the consistency between training and testing data, when compared to the simple solution of blindly training with a bigger diversity of the data.

C. DOMAIN ADAPTATION

Basic assumption of machine learning is that training set (in *source/train domain*) and testing set (in *target/test domain*) are supposed to be close in terms of distribution. In the case of detecting image manipulations (Gaussian blurring, sharpening, *etc.*), variations in the pre-processing pipeline introduce slight discrepancies between the train and test domains. However, some relationships, *i.e.*, the manipulation traces, should have been partially preserved between the two domains. Accordingly, forensic classification of pre-resized test samples can be considered as a problem of domain adaptation instead of a whole new problem to train on. In this paper, we focus on the adaptation of classification model, *i.e.*, in our case the Gaussian Mixture Model [5], to cope with the changes between source and target domains.

The best known technique for model adaptation is the fine tuning of a convolutional neural network pre-trained on ImageNet [23]. Classical CNNs need millions of images to train on, as in ImageNet, and a considerable amount of computing time and resources. For other problems, these conditions may not be satisfied. One possible shortcut is to use pre-trained weights on ImageNet as an initialization and then only fine-tune the last few layers. While this technique is effective, it needs access to labels in target domain to be applied. The fine-tuning of a dense neural network is used for image manipulation detection in our previous work [19] to adapt to pre-resizing in a weakly-supervised setting. We focus on the adaptation of GMM whose key parameters include the covariance matrices of Gaussian components. Therefore, here we briefly present previous works on transformation of covariance matrix. In [24], [25], the authors use covariance matrices between signals of several electrodes placed on the head of the subject as features. These features are used to distinguish movements made by the subject. Their objective is to adapt features between experiments to avoid re-calibration, in a semi-supervised setting. Domain adaptation is performed through covariance matrix transformations. More precisely, in [24], [25] they compute centers of mass with geodesic distance (*i.e.* the geometric mean in the language of Riemannian geometry) for the training and the testing ensembles of covariance matrices and minimize the geodesic distance between the two. It is worth pointing out that for us, covariance matrices are part of the classification model and not the features, and that we consider the challenging unsupervised setting and propose a new adaptation method as presented later in the paper.

III. ANALYSIS OF STATISTICAL DIFFERENCES BETWEEN TRAINING AND TESTING DATA

A. RESEARCH PROBLEM AND MOTIVATION

Primary motivation of this work is depicted in Fig. 1. A detector based on GMMs [5] is trained on patches of original-sized images to classify between manipulated and pristine patches (details of the classification pipeline are given in Section IV-A). This detector is then used on a testing set that has undergone resizing (bi-cubic interpolation) as a pre-processing operation, prior to applying a manipulation. Drops in accuracy are observed for several resizing factors. It is quite intuitive to assume that dependencies and transitions between neighboring pixels are altered by resizing. For example, downscaling usually induces sharper transitions as fewer pixels contribute to each transition. Consequently,

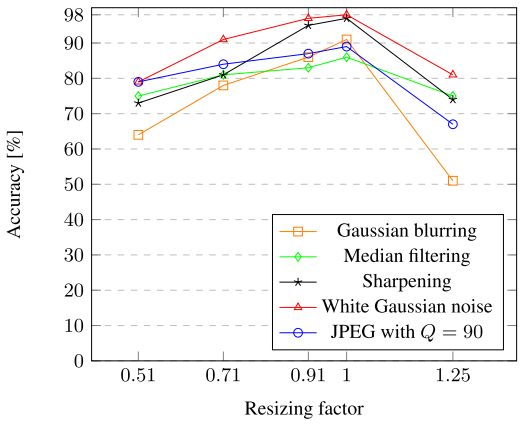


FIGURE 1. Accuracy of GMM-based method [5] under different resizing factors (bi-cubic interpolation) for several manipulations. We do not use factors like 0.5 to avoid the potential side effect of such factors (*i.e.* pixels sub-sampling instead of interpolation). 1 means no resizing.

alteration on pixels distribution for resized images would reduce descriptive power of GMMs and therefore forensic performances. We observe similar performance drops for other methods, *e.g.*, the method based on SPAM steganalytic features [14] and the CNN-based method [16] (results and discussions are presented in Section V). Finding solutions to avoid this performance drop is an important problem for enhancing the practical applicability of forensic detectors; however, to our knowledge only few efforts have been made in the literature to solve this kind of problem.

In this work, *source* denotes the training data (patches) from original-sized images. Half of these data have undergone a manipulation. *Target* denotes the testing data that have undergone resizing as a pre-processing operation, *i.e.*, patches from pre-resized images. Then, like for the training set, half of these testing data are manipulated. Our objective is to distinguish between original (*i.e.*, pristine) *vs.* manipulated patches. Accuracy should not drop because of pre-resizing for an ideal classifier. We first study how well (or how bad) a classifier trained on source data performs on target data (*e.g.*, results in Fig. 1), and secondly we seek an unsupervised method to enhance performances on target data without retraining the detector from scratch. Indeed, we intend to propose a method which is flexible, adaptive and resource-friendly, and which requires no ground-truth label in the target domain (*i.e.*, unsupervised adaptation).

In this study we mainly consider bi-cubic interpolation as pre-resizing, because it is probably the most commonly used resizing operation due to high visual quality of resized images. Our method also works with other interpolations. Additional results are presented in Supplementary Material.

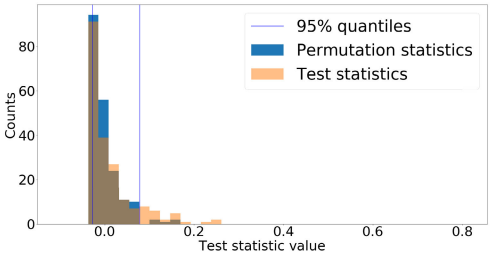
B. STATISTICAL TESTS

Decreases in performances seem to indicate a difference between source and target distributions. GMM trained on source data exhibits lower likelihood on target data. However GMM is a parametric model and this could also indicate that the problem is the parameterization and not the data. Non-parametric test is suited to investigate differences between source and target. Work from Gretton *et al.* [26] on two-sample problem with Maximum Mean Discrepancy (MMD) is a good tool for this purpose. MMD is defined as:

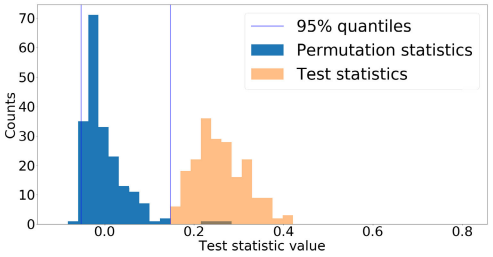
$$MMD[\mathcal{F}, p, q] = \sup_{f \in \mathcal{F}} (\mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{y\sim q}[f(y)]), \quad (1)$$

with p and q distribution of respectively source and target, x and y source and target samples, $\mathcal{F}$ a unit ball in a universal Reproducing Kernel Hilbert Space (RKHS) (in our case we choose Gaussian kernel with $\sigma = 2$), sup for supremum, and $\mathbb{E}$ for expectation. If $p = q$ then we have $MMD = 0$. This approach aims at finding the biggest difference between any moments of the two distributions: if it is zero then the two distributions are equal.

We have implemented a permutation test with MMD to test H_0 hypothesis, *i.e.*, train and test have same distribution. More details about it can be found in the Supplementary Material. Typical results are presented in Fig. 2. For original-sized testing images (Fig. 2.(a)), permutation statistics and test statistics about MMD follow very close empirical histograms, so in this case H_0 cannot be rejected. For pre-resized testing images, there is clear difference between permutation and test statistics (Fig. 2.(b)), therefore here H_0 can be rejected. The MMD test indicates that pre-resizing could change the distribution of patches. This is quite intuitive, nevertheless here we provide a statistical evidence of it.



(a) 0



(b) 0

FIGURE 2. Histograms of MMD between training and testing data (both sets contain half of sharpened patches): (a) with patches of original-sized testing images, and (b) with patches of testing images subject to resizing pre-processing (bi-cubic interpolation with factor of 0.53).

After that, we have considered Hilbert-Schmidt independence criterion (HSIC) test. Test is described in [27]. HSIC test is basically an MMD two-sample test between joint law of source and target ($P_{X,Y}$) and the product of marginals ($P_X * P_Y$). H_0 hypothesis is: $P_{X,Y} = P_X * P_Y$, which would indicate that X and Y are independent. Results are shown in Fig. 3. Test statistic values are clearly outside permutation statistic values, so we can reject the null hypothesis. Discarding H_0 is a hint in favor of dependence between source and target domains. This implies that it could be possible to adapt knowledge of source on target.

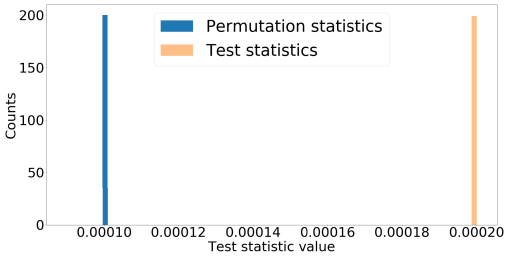


FIGURE 3. HSIC test for the case with sharpening as manipulation and pre-resizing factor of 0.53.

To summarize, the first test has given us some non-parametric arguments in favor of differences in statistical distributions of original-sized images (source) and pre-resized images (target) which could explain the observed drops in performances as shown in Fig. 1; the possible dependency between the two kinds of images (as reflected by the second test) has motivated our domain adaptation approach presented in the next section.

IV. PROPOSED APPROACH

A. CLASSIFICATION PIPELINE

Classification pipeline is inspired by work of Fan *et al.* [5]. Authors' pipeline uses GMM to model zero-mean centered patches (DC component removed). A comparison of likelihoods between model trained on original patches and model trained on manipulated patches is made to produce a classification decision. As mentioned earlier, we choose the GMM-based pipeline mainly because of its good performance on very small patches, better than other methods as show in Section V. Another reason is that GMM's parameters are closely related to empirical statistics of data, which may facilitate the adaptation. Technically, GMMs are intended to describe local dependencies and variations of pixels. Likelihood for a sample x_i (an 8×8 patch vectorized) under a GMM is expressed as:

$$\mathcal{L}(x_i|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k), \tag{2}$$

with π_k, μ_k and Σ_k respectively the weight, mean and multivariate (full) covariance matrix for k-th component in mixture of K. One GMM is trained only on manipulated patches and the other one only on original patches. Expectation Maximization (EM) algorithm is used to train the GMMs. Then for each test patch x_j, likelihood values of both GMMs are computed to form a ratio as:

$$r(x_j) = \frac{\mathcal{L}_{GMM_{manip}}(x_j)}{\mathcal{L}_{GMM_{ori}}(x_j)}. \tag{3}$$

If $r(x_j) > 1$ then the test patch x_j is predicted as manipulated, otherwise it is predicted as original.

Fan *et al.* [5] showed through an experimental study on leading eigenvectors that covariance matrices of Gaussian components in GMM are discriminative elements. This has motivated us to adapt GMM's covariance matrices to target data. It is also possible to work on the adaptation of weights of GMM, as in our recent work [19]. Details about that method and some comparisons are presented in Section V.

B. GRAFT: GMM RESIZING ADAPTATION BY FINE-TUNING

1) OBJECTIVE AND MAIN IDEA

Our goal is to come up with a simple method to adapt quickly GMM covariance matrices to target dataset (*i.e.* pre-resized), starting from a model learned on the source dataset (*i.e.* without pre-resizing). Inputs of our method are:

- $\mathcal{C}_1$: a set of $N = 2 \times K$ covariance matrices from two GMMs trained on the source dataset (the two GMMs, each having K components, are trained respectively on original patches and manipulated patches);
- $\mathcal{C}_2$: empirical estimations of the covariance matrices for original and manipulated patches on target pre-resized dataset.[2]

We seek to obtain $\mathcal{C}_1^{adp}$, a set of N covariance matrices adapted to the pre-resized target domain. $\mathcal{C}_1^{adp}$, along with the weights of original GMMs in the source domain, constitute two adapted GMMs which have improved forensic performance on the target domain. $\mathcal{C}_1^{adp}$ will be obtained by transforming $\mathcal{C}_1$ using information from $\mathcal{C}_2$. Of course, as an unsupervised adaptation approach, our method does not use any ground-truth label of the target domain, neither for the estimation of $\mathcal{C}_2$ nor for the transformation of $\mathcal{C}_1$.

Formally we have:

$$\begin{aligned}\mathcal{C}_1 &= \{C_1^{(1),ori}, \ldots, C_K^{(1),ori}, C_1^{(1),mnp}, \ldots, C_K^{(1),mnp}\},\\ \mathcal{C}_2 &= \{C_1^{(2),ori}, \ldots, C_{\frac{M}{2}}^{(2),ori}, C_1^{(2),mnp}, \ldots, C_{\frac{M}{2}}^{(2),mnp}\}.\end{aligned} \tag{4}$$

The idea is to find some transformation of $|\mathcal{C}_1|$ to bring it "closer" to $|\mathcal{C}_2|$, under the constraint of increasing the *likelihood* of adapted GMMs on the target pre-resized dataset. Here, we have been inspired by the work of Rodrigues *et al.* [25] in the brain-computer interface field. Authors propose the Riemannian Procrustes Analysis (RPA) method, an adaptation between sets of covariance matrices. In their work, and unlike in ours, covariance matrices are features and not parameters of a probabilistic model. They perform feature adaptation in a semi-supervised manner while we would like to adapt our GMMs in an unsupervised scenario. The objectives are not comparable and it is not possible to directly use their method for our problem. However, our GRAFT method retains RPA spirit by using a set of basic geometrical transformations of covariance matrices. So here we briefly present the RPA procedure which comprises three main steps (mathematical details can be found in [25]):

1) Translate source and target sets to obtain the identity as geometric mean (re-centering): $\mathcal{C}_1^{ctr} = T_1(\mathcal{C}_1)$ with $\mathcal{C}_1$ the first set of covariance matrices. Let V be the geometric mean of $\mathcal{C}_1 = \{C_1^1, \ldots, C_K^1\}$. Then T_1 is defined as $C_i^{1,(ctr)} = V^{-\frac{1}{2}} C_i^1 V^{-\frac{1}{2}}$. Similarly, we have T_2 for second set $\mathcal{C}_2$;
2) Perform rescaling on each axis to get unit variance: $\mathcal{C}_1^{str} = S_1(\mathcal{C}_1^{ctr})$ and $\mathcal{C}_2^{str} = S_2(\mathcal{C}_2^{ctr})$. S_1 is defined such that $C_i^{1,(str)} = \left(C_i^{1,(ctr)}\right)^p$ where p is a proper scaling factor for the variance normalization;
3) Minimize the geodesic distance between $\mathcal{C}_1^{str}$ and $\mathcal{C}_2^{str}$ by finding an optimal rotation U around the origin (the rotation is applied to $\mathcal{C}_1^{str}$ only, not on $\mathcal{C}_2^{str}$). $C_i^{1,(str)}$ are

[2] In our algorithm, M estimations are computed in order to improve robustness against variance of empirical estimation. Details on how to obtain $\mathcal{C}_2$ are presented later in this section.

modified so that $C_i^{1,(rot)} = UC_i^{1,(str)}U^T$ with $UU^T = \mathcal{I}$ (*i.e.*, the product is identity matrix).

With the RPA procedure, both sets of features become comparable: a classifier learned with C_1 features is also meaningful with C_2 features. RPA uses geodesic distance between matrices to derive the transformation, while GMMs use likelihood to carry out classification (see Section IV-A for GMM-based classification pipeline). Hence, in GRAFT we shall use the group of elementary geometrical transformations of covariance matrices, but combine and optimize them in a different and appropriate manner.

2) TRANSFORMATION AND INTERPOLATION

We in the first place transform our $\mathcal{C}_1$ and $\mathcal{C}_2$ in Eq. (4) by using the three geometrical transformations from RPA procedure (see above). More precisely, we first translate $\mathcal{C}_1$ and $\mathcal{C}_2$ so that they both have the identity as barycenter. Then stretching and rotation are performed so that $\mathcal{C}_1$ and $\mathcal{C}_2$ get closer in the transformed space in terms of geodesic distance. The set of transformed covariance matrices corresponding to $\mathcal{C}_1$ are denoted by $\mathcal{C}_1^{RPA}$. We know from the analysis at the end of last subsection that this transformation is not optimum for GMM-based classification pipeline, mainly because RPA does not use likelihood as criterion during the transformation. Therefore, additional steps are needed.

We then consider another transformation which translates $\mathcal{C}_1$ toward $\mathcal{C}_2$, so that the translated version, denoted by $\mathcal{C}_1^{trg}$, has the same center of mass as $\mathcal{C}_2$. With a little abuse of notation, for original patches this means satisfying the following equation (similarly for manipulated patches):

$$\sum_{i=1}^{K} \pi_i \times \mathcal{C}_i^{trg,ori} = \frac{2}{M} \sum_{j=1}^{M/2} C_j^{(2),ori}, \tag{5}$$

where $\mathcal{C}_i^{trg,ori}$ is the i-th covariance matrix for original patches in $\mathcal{C}_1^{trg}$, π_i are weights of GMM of original patches in source domain, and $C_j^{(2),ori}$ is the j-th estimated covariance of original patches in target domain. This transformation considers the fit of second-order statistics (*i.e.*, the covariance matrix) to the target dataset. In fact, it can be shown, with the assumption of a perfect fit of GMM and centered patches, that the weighted sum of Gaussian components' covariance matrices of a GMM (the left-hand side of Eq. (5)) is equal to the covariance matrix of the data (the right-hand side of Eq. (5) is an estimation of data covariance).[3] Indeed, as in [28], [5] and [19], our patch samples are centered to have zero mean (*i.e.*, DC component is removed for each patch), and accordingly Gaussian components in GMM also each have zero mean. As proven in [19], the cross terms in covariance computation disappear with zero-mean Gaussian components. Then we have [19]:

$$\mathrm{cov}(X_i, X_j) = \sum_{k=1}^{K} \pi_k \times \Sigma_k^{(i,j)}, \tag{6}$$

where X is the 64-dimensional random variable of pixel values from the vectorized patches, cov(., .) is the covariance function, Σ_k and π_k are respectively the covariance and the weight for the k-th component in GMM, and i, j and (i, j) are element indices within the corresponding vector and matrix.

Technically, similar to the translation step in RPA [25] (see step 1) of RPA in the last subsection), the translation to $\mathcal{C}_1^{trg}$ is realized by simple matrix multiplications on $\mathcal{C}_1$.

Neither $\mathcal{C}_1^{RPA}$ nor $\mathcal{C}_1^{trg}$ are optimal in terms of GMM likelihood (see Fig. 6 in Section V for a concrete example). The former has the identity as geometric mean, while the latter only considers fit of covariance matrices on average but not the GMM likelihood. However, both $\mathcal{C}_1^{RPA}$ and $\mathcal{C}_1^{trg}$ get closer to the target domain to some extent, and a natural and simple idea is to *interpolate* between the two to get a better solution. This is a heuristics based approach but it has good intuition and is experimentally effective. Another advantage is that a simple interpolation between two valid covariance matrices still leads to a valid solution of symmetric positive semi-definite matrix.

The interpolation is naturally driven and governed by the *maximization of likelihood* of GMMs. The rationale behind is the fact that GMMs rely heavily on the descriptive capability (high likelihood) to carry out correct classification. Two distinct (regarding respectively original and manipulated classes of patches) interpolation coefficients are computed between $\mathcal{C}_1^{RPA}$ and $\mathcal{C}_1^{trg}$:

- α_1 is used to interpolate between $\mathcal{C}_1^{RPA,ori}$ and $\mathcal{C}_1^{trg,ori}$ (original patches);
- α_2 is used to interpolate between $\mathcal{C}_1^{RPA,mnp}$ and $\mathcal{C}_1^{trg,mnp}$ (manipulated patches).

Optimal values for α_1 and α_2 are computed by maximizing the log-likelihood respectively on the two GMMs. This will be shown to enhance both the descriptive power by maximizing the log-likelihood and the discriminative capability by choosing separate coefficients for original/manipulated classes. This trade-off between adapting to the new domain while simultaneously preserving the discriminative power is also motivated by the theoretical study of [29].

To estimate the covariance matrix on 8×8 patches of pre-resized testing images, usual empirical estimator is used:

$$\sigma_{ij} = \frac{1}{N_P} \sum_{l=1}^{N_P} (x_{il} - \mu_i)(x_{jl} - \mu_j), \tag{7}$$

where i is the row, j is the column, N_P is the number of patches, and μ_i, μ_j are respectively the empirical mean of row i and column j over the N_P samples. This estimator is unbiased, yet it suffers from a high variance. As explained earlier, in order to counterbalance the variance of estimation, we perform M estimations of empirical covariance matrices

[3] Detailed proof can be found in our previous work [19] (preprint link is given in the reference list).

on separate subsets of the target dataset of patches from pre-resized images. It is worth mentioning that we estimate matrices separately for the two original/manipulated classes, by using the so-called pseudo-labels in the target domain. Pseudo-labels are an important feature of GRAFT and are described in the next subsection.

3) PSEUDO-LABELS

In the unsupervised framework of GRAFT, we do not have access to ground-truth labels of target domain. However, it turns out that the accuracy of the GMM-based classifier does not drop to 50%, *i.e.* random guessing. It means that classifier is still able to label accurately some pre-resized test samples. From these samples, we derive *pseudo-labels*. We explain in the following how these pseudo-labeled samples are selected.

At the beginning of the optimization of α_1 and α_2, their initial values are drawn randomly to compute the first two sub-optimal, interpolated GMMs. We assume that even with this sub-optimal interpolation, the GMMs can still label correctly some testing samples. Hence, we need to select almost surely original and almost surely manipulated patches. Almost surely original (*resp.* manipulated) patches are selected from the 5–15 (*resp.* 85–95) percentile of the likelihood ratio, leading to 20% patches with reliable pseudo-labels because their likelihood ratio is farthest from 1. The GMMs are originally trained on the source domain, so most surely classified target samples are very likely to be closer to the source domain than to the target domain. It means that they are not enough representative of the effect of pre-resizing. Therefore, extreme likelihood ratio values (percentile 0–5 and 95–100) are discarded.

To validate this approach, we have computed the accuracy of pseudo-labeled samples and it is typically above 95%. Of course, ground-truth labels on the target domain are not used in the GRAFT method and have been only used here to validate our hypothesis on pseudo-labels. In the end, 20% (10% almost surely original + 10% almost surely manipulated) of the test data are pseudo-labeled. The method is robust regarding the chosen percentages. Selecting percentiles 5–20 and 80–95 (or 10–15 and 85–90) would not significantly impact the final accuracy. However, the classification accuracy starts to drop when more than 30% of the test data are pseudo-labeled because their accuracy drops.

One detail is that we use directly $\mathcal{C}_1$ to get the pseudo-labels used for the estimation of empirical covariance matrices $\mathcal{C}_2$ on test samples (see steps 2 and 3 of Algorithm 1). This is different from the derivation of pseudo-labels via sub-optimal interpolation mentioned above, but remains reasonable because at the stage of estimating $\mathcal{C}_2$ it is impossible to use interpolation to get pseudo-labels. In fact, the estimation of $\mathcal{C}_2$ serves to obtain $\mathcal{C}_1^{trg}$, one end point of the interpolation.

4) SUMMARY

The main steps of the unsupervised adaptation method of GRAFT are illustrated in Fig. 4 and its pseudo-code is presented in Algorithm 1. The red line in Fig. 4 represents translation toward identity, stretching and rotation. Sets of covariance matrices within the orange dashed contour have the identity matrix ($\mathcal{I}_n$) as geometric mean. In Algorithm 1, α_1 and α_2 are the coefficients of an interpolation represented by the green dotted line in Fig. 4. Formally, the adapted sets of covariance matrices are obtained as:

$$\begin{aligned} \mathcal{C}_{adp}^{ori} &= \mathcal{C}_1^{trg,ori} * (1-\alpha_1) + \mathcal{C}_1^{RPA,ori} * \alpha_1, \\ \mathcal{C}_{adp}^{mnp} &= \mathcal{C}_1^{trg,mnp} * (1-\alpha_2) + \mathcal{C}_1^{RPA,mnp} * \alpha_2. \end{aligned} \quad (8)$$

Algorithm 1 GRAFT Algorithm

Input: Source and target data, two GMMs trained respectively on original and manipulated patches of source data

Output: Adapted GMMs

1: Concatenate the two sets of covariance matrices from the two trained GMMs on source data to form $\mathcal{C}_1$
2: Compute pseudo-labels based on ratio of likelihood with $\mathcal{C}_1$ as covariance matrices
3: Use these pseudo-labels to compute estimations of covariance matrices of original and manipulated patches on target data to get $\mathcal{C}_2$
4: Recentering: $\mathcal{C}_1^{ctr} = T_1(\mathcal{C}_1)$ and $\mathcal{C}_2^{ctr} = T_2(\mathcal{C}_2)$
5: Rescaling: $\mathcal{C}_1^{str} = S_1(\mathcal{C}_1^{ctr})$ and $\mathcal{C}_2^{str} = S_2(\mathcal{C}_2^{ctr})$
6: Rotation: $\mathcal{C}_1^{RPA}$ and $\mathcal{C}_2^{RPA}$
7: Translation of $\mathcal{C}_1$ to have the same center of mass as $\mathcal{C}_2$ (separately for original and manipulated patches): $\mathcal{C}_1^{trg}$
8: Initialization of $\alpha_1 \sim \mathcal{U}[0.1, 0.9]$ and $\alpha_2 \sim \mathcal{U}[0.1, 0.9]$
9: Perform a first sub-optimal interpolation between $\mathcal{C}_1^{RPA}$ and $\mathcal{C}_1^{trg}$ with these random values of α_1 and α_2
10: Only keep most confident samples (see text) and construct two pseudo-labeled sets: almost surely original and almost surely manipulated testing patches
11: Find optimal interpolation coefficients α_1 and α_2 in Eq. (8) based on maximization of sum of log-likelihood on two sets of pseudo-labeled patches: $\mathcal{C}_{adp}^{ori}$ and $\mathcal{C}_{adp}^{mnp}$
12: Repeat steps 8 to 11 for five times and keep the adapted GMMs with highest sum of log-likelihood
13: **return**

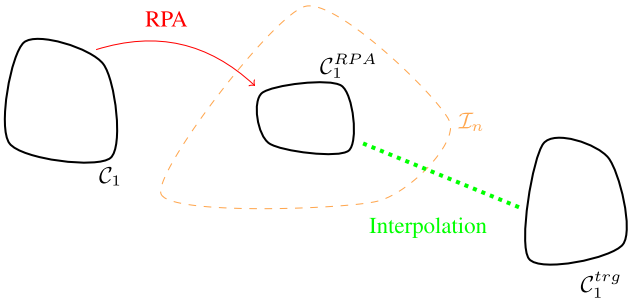


FIGURE 4. Transformation and interpolation in GRAFT. $\mathcal{I}_n$ is the identity matrix of dimension $n \times n$, in our case 64 × 64, the size of the covariance matrix of vectorized 8 × 8 patches. The interpolation is essential in GRAFT which maximizes the log-likelihood of GMMs with regard to the target domain.

α_1 and α_2 are computed such that they maximize the sum of log-likelihood of two GMMs:

$$\underset{\alpha_1,\alpha_2}{\operatorname{argmax}} \mathcal{LL}^{ori}_{adp}(\alpha_1) + \mathcal{LL}^{mnp}_{adp}(\alpha_2). \quad (9)$$

Log-likelihood $\mathcal{LL}^{ori}_{adp}$ is computed on almost surely original patches of the testing set, with adapted covariances and corresponding original weights. Similarly, $\mathcal{LL}^{mnp}_{adp}$ is computed on manipulated pseudo-labeled patches of the testing set with the adapted GMM covariances. Optimal values for α_1 and α_2 depend on the manipulation and resizing factor, so the full interval [0, 1] is searched. Like for the EM algorithm, performance depends on the initial random values of α_1 and α_2 for the first sub-optimal interpolation. Therefore, initialization is performed multiple times (five times is experimentally a good trade-off between computation time and performances), and the one with highest log-likelihood on pseudo-labeled test samples is selected as the final adapted GMMs. To summarize, in GRAFT, in order to adapt GMMs to the new target domain, simple operations (translation, scaling, rotation and interpolation) are performed to adjust covariance matrices in an unsupervised manner, and the procedure is driven by GMMs likelihood maximization on pseudo-labeled target samples.

V. EXPERIMENTS

A. IMPLEMENTATION DETAILS OF GRAFT

Code is available online[4] to reproduce experiments. We use 8×8 patches. They are flattened (vectorized) and centered (mean of each patch is removed), as in [5]. We use Scikit-learn [30] implementation of Gaussian Mixture Model. Each GMM has 75 components. This number is a good trade-off between model complexity for training phase and performance on testing phase. In order to counterbalance weaknesses of EM algorithm, we perform five times of initialization for mixture weights π_k and covariance matrices Σ_k. Initialization is done using K-means. The GMM means μ_k are initialized to zeros and forced again to be zero after training [5], [28]. We keep among the five trials the initialization with the highest log-likelihood on training data. Grid searches for interpolation parameters α_1 and α_2 are performed for speed purposes with black-box optimization technique from [31]. It does not necessarily produce better results than regular grid search but is quicker as the more interesting sub-spaces are selected automatically to then perform grid search. For empirical covariance matrices on target domain, we have $M = 2 \times 30 = 60$ estimations. The number of 30 has been set experimentally to have a stable geometric mean around different runs. Method is robust against this number and could work with less.

B. DATA

Images are from Dresden database [32]. It is a relatively big source of images with about 1200 images, in RAW format, coming from 5 different cameras. Images are in relatively high resolution (typically of 2000×3000 pixels), with quite diverse exposures and contents. Some image examples can be found in Figure 5. We randomly select 30% of images for testing purposes. These images are resized, before applying considered manipulation. The remaining 70% of images are for training and selection of hyper-parameters. This set is not resized at any stage. As mentioned in Section I, we have selected a number of manipulation operations that we consider as elementary. They are summarized and described in Table 1. We apply manipulations to every training images, and it allows us to learn on pair of patches: original *vs.* manipulated. To prepare experimental data, raw Dresden images are demosaicked using rawpy (https://github.com/letmaik/rawpy), then converted to grayscale with Python interface of OpenCV (https://opencv.org/). Finally we draw randomly 400000 8×8 patches of each class, which makes 800000 training patches (balanced case) for each considered binary classification problem. We also extract same amount of testing patches.

(a) Contrasted scenery (b) Bright indoor scenery

(c) Outdoor scenery (d) Bleak scenery

FIGURE 5. Some examples of original Dresden images.

TABLE 1. List of basic image manipulation operations to be detected.

ORI	No image modification
GF	Gaussian filtering with 3×3 kernel and $\sigma = 0.5$
MF	Median filtering with 3×3 kernel
USM	Unsharp masking with window size 3×3, and parameter 0.5 for the Laplacian filter to generate the sharpening filter kernel
WGN	White Gaussian noise addition with $\sigma = 2$
JPEG	JPEG compression with $Q = 90$

C. DETAILS OF COMPARISON METHODS

1) WEAKLY-SUPERVISED GMM ADAPTATION WITH WEIGHTS ADJUSTMENT

We have recently considered in [19] a classification pipeline that shares similarities with pipeline of this work. Similarly, two GMMs are trained; however, instead of computing ratio of likelihood, in [19] features are extracted from GMMs (the so-called "responsibilites" of Gaussian components) and used by a Dense Neural Network (DNN) to distinguish between original and manipulated patches.

[4] https://forge.uvolante.org/darmet/GRAFT

Likewise, objective of [19] is to adapt to pre-resizing. It is achieved through adaptation of GMMs' weights to better fit empirical covariance of target domain (feature extraction adaptation) and then fine-tuning of the DNN for the new features (classification adaptation). The main difference is that in GRAFT adaptation is performed in the challenging unsupervised scenario, while method [19] requires some labels from target domain. Despite the differences in methodologies and scenarios, we compare GRAFT with the weakly-supervised method of [19] and its unsupervised variant.

2) SPAM-BASED METHOD

A branch of methods to detect image manipulation are based on steganalytic features, typically the SPAM [17] and SRM [18] features. In this work, We compare with a SPAM-based method. SPAM features are extracted with code from http://dde.binghamton.edu/download/feature_ extractors, translated in Python, with a quantization order $T = 4$ and two-neighbor co-occurrences. We perform classification with the popular XGBoost framework [33]. It is worth mentioning that although existing methods based on steganalytic features consider larger patches, such methods can directly cope with 8×8 patches because the feature extraction and classifier training still work without any problem on small patches. This actually leads to acceptable baseline performances as shown later in this section.

3) CNN-BASED METHOD

To the best of our knowledge, no existing CNN-based methods consider and report results on small patches of 8×8 pixels. Nevertheless, for comparisons purposes and similar to our previous study in [19], we improve and adapt the state-of-the-art deep-learning-based method from Bayar and Stamm [16] so that the CNN can work with very small patches. Indeed, with the four pooling layers in the original CNN of [16] and 8×8 patches as input, output of these pooling layers drop to 1×1 and following 2D convolution cannot be computed anymore. It is technically possible to keep one or two of the four pooling layers to work with 8×8 patches, but experimentally we have obtained better results without any pooling layer retained. We think that pooling as spatial reduction may have a good effect for big patches of for example 256×256 pixels, but not anymore on 8×8 patches where we should probably avoid information loss caused by pooling. Learning rate has also been tuned for better accuracy from 10^{-3} (value suggested in the original paper [16]) to 10^{-4}. We use Caffe implementation from authors (https://gitlab.com/MISLgit/constrained-conv-TIFS2018) on exactly the same training and testing patches as GMM (same number as well).

D. PERFORMANCE AND ANALYSIS OF ADAPTATION

1) GRAFT METHOD

Table 2 presents the testing accuracy, without any adaptation, under different resizing factors for GMM-based method

TABLE 2. Testing accuracy (in %) without any adaptation for GMM-based method. The performance drop compared to the case without pre-resizing (*i.e.* the row of $\times 1$) is given in parentheses. The last column gives average values of the accuracy for five considered manipulations.

	GF	MF	USM	WGN	JPEG	AVG
$\times 1$	91	86	97	98	89	92
$\times 0.51$	64 (-27)	75 (-11)	73 (-24)	79 (-19)	79 (-10)	74 (-18)
$\times 0.71$	78 (-13)	81 (-5)	81 (-16)	91 (-7)	84 (-5)	83 (-9)
$\times 0.91$	86 (-5)	83 (-3)	95 (-2)	97 (-1)	87 (-2)	90 (-2)
$\times 1.25$	51 (-40)	75 (-11)	74 (-23)	81 (-17)	67 (-22)	70 (-22)

(same results as those shown in Fig. 1 in Section III), where $\times 1$ stands for the case of original-sized testing images without any resizing pre-processing. We can see that there are obvious accuracy drops for both downscaling and upscaling. Our objective is to improve the accuracy on pre-resized testing data, in an unsupervised manner, by using the proposed GRAFT algorithm.

Results of our approach GRAFT are presented in last row of Table 3 (pre-resizing factor of 0.51), Table 4 (pre-resizing factor of 0.71) and Table 5 (pre-resizing factor of 1.25, upscaling). We also show results of another method: retraining new GMMs from scratch by using 10% testing samples with ground-truth labels. It corresponds to a scenario where few target pre-resized data are available with labels. This scenario is also a good indicator of the difficulty of the task. Indeed, if a detector trained with few labeled target data performs bad on target domain, the problem should be difficult. From the tables, we can see that GRAFT method gives performance improvement in all cases, except for two situations, *i.e.* MF and JPEG under resizing of 0.71. However, in the first case of MF, even the retraining method gives no improvement, which implies that forensics problem becomes more difficult for GMM-base method. In addition, under many testing scenarios there is no big performance gap between unsupervised GRAFT and the retraining method which does use ground-truth labels from target domain. In certain cases, retraining GMMs can lead to big accuracy increase when compared with GRAFT (*e.g.*, JPEG with resizing of 0.71), though at the expense of higher computational cost and with the assumption of gaining access to true labels.

The improved average accuracy of GRAFT (last number at bottom right in Tables 3, 4 and 5) is quite satisfying, around 85%. For resizing factor of 0.51 (Table 3), we are able to gain in average $+10\%$ of testing accuracy with GRAFT method. The performance improvement of GRAFT depends on the manipulation operation, ranging from the smallest increase of $+4\%$ for MF and JPEG to the biggest increase of $+15\%$ for GF and USM. The improvement is to some extent correlated with the drop in performances induced by pre-resizing. For resizing factor of 0.71 (Table 4), we can see that even with the retraining method only limited improvements can be achieved. This case is more difficult than resizing $\times 0.51$ or $\times 1.25$, which explains the moderate gains in performance with our GRAFT method, though the average accuracy after adaptation remains satisfying. Our method is

TABLE 3. Testing accuracy (in %) with adaptation of GMMs (resizing $\times 0.51$). The improved accuracy, compared to the case of "without adaptation", is given in parentheses.

	GF	MF	USM	WGN	JPEG	AVG
Resizing $\times 0.51$ (without adaptation)	64	75	73	79	79	74
Resizing $\times 0.51$ (retraining with 10%)	77 (+13)	79 (+4)	82 (+9)	84 (+5)	81 (+2)	81 (+7)
Resizing $\times 0.51$ (unsupervised, GRAFT)	79 (+15)	79 (+4)	88 (+15)	89 (+10)	83 (+4)	84 (+10)

TABLE 4. Testing accuracy (in %) with adaptation of GMMs (resizing $\times 0.71$), with improved accuracy in parentheses.

	GF	MF	USM	WGN	JPEG	AVG
Resizing $\times 0.71$ (without adaptation)	78	81	81	91	84	83
Resizing $\times 0.71$ (retraining with 10%)	82 (+4)	81 (+0)	93 (+12)	93 (+2)	89 (+5)	88 (+5)
Resizing $\times 0.71$ (unsupervised, GRAFT)	79 (+1)	81 (+0)	91 (+10)	92 (+1)	84 (+0)	85 (+2)

TABLE 5. Testing accuracy (in %) with adaptation of GMMs (resizing $\times 1.25$), with improved accuracy in parentheses.

	GF	MF	USM	WGN	JPEG	AVG
Resizing $\times 1.25$ (without adaptation)	51	75	74	81	67	70
Resizing $\times 1.25$ (retraining with 10%)	70 (+19)	80 (+5)	95 (+21)	89 (+8)	80 (+13)	83 (+13)
Resizing $\times 1.25$ (unsupervised, GRAFT)	83 (+32)	84 (+9)	83 (+9)	96 (+15)	88 (+21)	87 (+17)

TABLE 6. Testing accuracy (in %) with adaptation of the GMM-based method by GRAFT for the case of mixed pre-resizing factors. The pre-resizing factors are drawn following uniform law within the specified interval.

	GF	MF	USM	WGN	JPEG	AVG
Resizing $\times [0.48, 0.72]$ (without adaptation)	71	81	76	72	87	77
Resizing $\times [0.48, 0.72]$ (unsupervised, GRAFT)	80 (+9)	82 (+1)	89 (+13)	77 (+5)	89 (+2)	83 (+6)
Resizing $\times [1.12, 1.27]$ (without adaptation)	53	78	81	83	74	74
Resizing $\times [1.12, 1.27]$ (unsupervised, GRAFT)	63 (+10)	79 (+1)	89 (+8)	95 (+12)	76 (+2)	80 (+6)

not limited to downscaling and also provides good results with upscaling, as shown in Table 5 with the case of resizing factor of 1.25. A considerable average accuracy improvement of $+17\%$ is achieved by our GRAFT method. At last, it can be observed from Table 3 and 5 that sometimes GRAFT can achieve higher accuracy than the retraining method. One possible explanation is that the knowledge gained from original-sized source domain is beneficial to improve the performance of the same task in the new pre-resized domain.

In Table 6 target data are not pre-resized with a specific factor but a random factor drawn in some interval. As depicted in the table, GRAFT approach is able to cope with a testing set composed of a mix of pre-resizing factors. It is a more difficult problem as the target domain is more diverse. Hence, performance gain in average is $+6\%$. An explanation is that our method relies on likelihood maximization and with a mix of factors likelihood is optimized on average. Therefore when testing with a patch at a specific factor it is not as optimal as previously. Yet our method remains useful in particular for unsharp masking (USM) detection, and the average accuracy after the improvement of GRAFT is higher than or equal to 80% for both cases.

In order to show the effectiveness of interpolation of covariance matrices in GRAFT, Fig. 6 illustrates the evolution of testing accuracy as a function of the interpolation coefficients α_1 and α_2. For this case of adapting a detector of GF to a resizing of $\times 0.51$ as pre-processing, the derived optimal values are around 0.2 for both α_1 and α_2. The upper right corner $(1, 1)$ corresponds to C^{RPA} and the lower left corner $(0, 0)$ to C^{trg}. None of these two points are optimal in terms of accuracy or likelihood, and the interpolation between the two can improve the detection performance.

At last, we would like to mention that our method is much less time consuming than the 150 iterations of

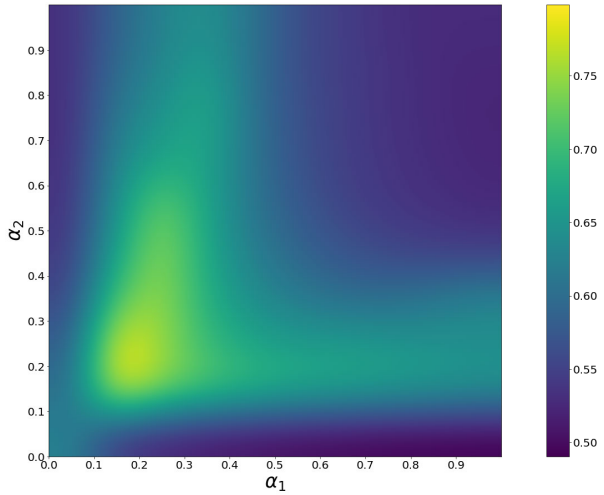


FIGURE 6. Testing accuracy as a function of interpolation coefficients α_1 and α_2. The manipulation is Gaussian blurring and the resizing factor is 0.51.

EM algorithm for the GMM training. We were actually looking for shortcuts to perform lightweight and flexible adaptation as in the proposed GRAFT algorithm. Regarding computation time, on our machine with Intel Xeon E5-2630 CPU, it takes approximately 15min for GRAFT procedure, while it needs about 5h to train one GMM with 400000 samples. The weakly-supervised GMM adaptation takes about 3min (1min for the optimization problem and 2min for DNN fine-tuning). Training of the CNN-based method of Bayar and Stamm takes around 2h for each binary problem on an Nvidia 1080 Ti GPU (it would be much longer on CPU). Finally, extraction of SPAM features is quite fast (about 1h) as it can be done in parallel on the 40 cores of our CPU. However training of the classifier is about 10h as feature dimension is very large.

2) COMPARISONS WITH WEAKLY-SUPERVISED GMM ADAPTATION WITH WEIGHTS ADJUSTMENT AND ITS UNSUPERVISED VARIANT

As explained in Section V-C, our recent method of [19] requires some labels on target domain, while the proposed GRAFT method in this paper performs unsupervised adaptation. Results of weakly-supervised method [19] are presented in Table 7, on the row of "weakly sup. [19]", with comparisons to GRAFT. It can be observed that the GRAFT method achieves comparable or even slightly better performances though considering a more challenging setting of unsupervised adaptation. Average accuracy after applying method of [19] with a pre-resizing factor of $\times 0.51$ is 80% while with GRAFT method it is 84%. Average accuracy for upscaling of $\times 1.25$ is also in favor of GRAFT with 87% compared to 83% for weakly-supervised adaptation.

Instead of using few labels as in the original method of [19] for adaptation, we have implemented an unsupervised version that uses pseudo-labels instead. Pseudo-labeled samples are determined similarly as in our GRAFT method. Results of this unsupervised variant are presented in Table 7, on the row of "unsup. version of [19]". The unsupervised version has lower accuracy than the original weakly-supervised version, which is understandable because less information on target domain is available. GRAFT method outperforms the unsupervised variant of [19]. The better performance of GRAFT may be due to the formulation of adaptation as GMM likelihood maximization, instead of only considering the fit of covariance statistics in [19].

TABLE 7. Testing accuracy (in %) of the weakly-supervised method from [19] and its unsupervised variant, with comparisons with GRAFT. "sup." and "unsup." stand respectively for "supervised" and "unsupervised."

	GF	MF	USM	WGN	JPEG	AVG
Resizing $\times 0.51$ (weakly sup. of [19])	78	76	92	70	86	80
Resizing $\times 0.51$ (unsup. version of [19])	71	75	82	65	84	75
Resizing $\times 0.51$ (unsupervised, GRAFT)	79	79	88	89	83	84
Resizing $\times 1.25$ (weakly sup. of [19])	66	80	95	95	78	83
Resizing $\times 1.25$ (unsup. version of [19])	64	75	91	92	68	78
Resizing $\times 1.25$ (unsupervised, GRAFT)	83	84	83	96	88	87

3) RESULTS OF SPAM-BASED AND CNN-BASED METHODS

SPAM features have been originally designed for the steganalysis field, though they are also very powerful for image forensics problems. Results of SPAM-based method are presented in Table 8. GMM-based method provides better base score (*i.e.*, accuracy on testing set without pre-resizing) than SPAM-based method. It is +8% on average for GMM (92% for GMM *vs.* 84% for SPAM in terms of average accuracy, see the row of "$\times 1$" in Tables 2 and 8). We also observe performance drop under pre-resizing for SPAM-based method, although not as big as for GMM approach. However, the decreased average accuracy is comparable between the two methods (please compare results in Table 2 and Table 8). It is interesting to notice that after applying GRAFT adaptation, GMM approach has improved accuracy which is as competitive as the base score of SPAM (the average improved accuracy of GRAFT is around 85%, see Tables 3–5, compared with 84% for SPAM without pre-resizing). We also notice that SPAM-based method is quite robust for MF, therefore fusing results of the two approaches could be an interesting solution to be explored in the future.

As described in Section V-C, we have modified Bayar and Stamm's CNN [16] so that the network is now able to detect manipulation on 8×8 patches. We would like to emphasize here that manipulations considered in this work are more challenging than in their paper [16] (smaller magnitude, smaller kernel for median filtering, *etc.*) and that patches are much smaller. Therefore performances of the network are reduced compared to performances reported in Bayar and

TABLE 8. Testing accuracy (in %) for SPAM-based method. The performance drop compared to the case without pre-resizing (*i.e.* the row of ×1) is given in parentheses.

	GF	MF	USM	WGN	JPEG	AVG
×1	73	95	89	87	78	84
×0.51	60 (-13)	95 (-0)	78 (-11)	68 (-19)	70 (-8)	74 (-10)
×0.71	68 (-5)	96 (+1)	83 (-6)	76 (-11)	73 (-5)	79 (-5)
×0.91	71 (-2)	95 (-0)	87 (-2)	83 (-4)	73 (-5)	82 (-2)
×1.25	56 (-17)	90 (-5)	84 (-5)	86 (-1)	70 (-8)	77 (-7)

TABLE 9. Testing accuracy (in %) for the CNN-based method. The performance drop compared to the case without pre-resizing (*i.e.* the row of ×1) is given in parentheses.

	GF	MF	USM	WGN	JPEG	AVG
×1	79	85	91	86	79	84
×0.51	68 (-11)	82 (-3)	80 (-11)	66 (-20)	76 (-3)	74 (-10)
×0.71	73 (-6)	83 (-2)	89 (-2)	80 (-6)	79 (-0)	81 (-3)
×0.91	61 (-18)	78 (-7)	81 (-10)	88 (+2)	71 (-8)	76 (-8)
×1.25	55 (-22)	72 (-13)	74 (-17)	82 (-4)	55 (-22)	68 (-16)

Stamm's paper [16]. Accuracy of the CNN-based method is presented in Table 9. CNN's base performance without resizing is lower than GMM, −8% in average (84% for CNN *vs.* 92% for GMM). CNN also suffers from accuracy decrease under pre-resizing. Similar to the comparison with SPAM, GMM with GRAFT achieves after adaptation to pre-resizing average accuracy values (around 85%) comparable to the base average score of CNN when there is no pre-resizing.

We have made efforts to carry out unsupervised adaptation to pre-resizing for SPAM- and CNN-based detectors (to our knowledge no such methods exist in the image forensics literature). We have applied the adaptation methods based on optimal transport [34] and Transfer Component Analysis [35] to SPAM and tried to use pseudo-labels for both SPAM and CNN, but for each case there is no noticeable accuracy improvement after adaptation. One possible explanation is that it seems more difficult to perform unsupervised adaptation for discriminative classification pipelines such as SPAM and CNN than for pipeline based on generative model such as GMM. For example, pseudo-labeled samples have very little impact on SPAM- and CNN-based classifiers because pseudo labels are determined by these classifiers and in turn refining classifiers with these samples in the discriminative training setting has limited effect. By contrary, GMM's descriptive capability for pseudo-labeled samples can be effectively improved, which leads to performance improvement. We leave for future work the development of successful adaptation algorithms for SPAM- and CNN-based methods.

E. TRAINING WITH A MIX OF RESIZED DATA

As discussed in Section I, a method that comes firstly in mind to mitigate this problem of pre-resizing is to mix a lot of different resizing factors in the training set. However, results in Table 10 seem to indicate that this is not the optimal solution for both CNN- and GMM-based methods. CNN- and GMM-based detectors have been trained on Dresden database, with each image (both train and test) having been resized with a factor randomly selected in [0.48, 0.57, 0.63, 0.68, 0.72, 0.79, 0.85, 0.91, 1, 1.12, 1.17, 1.22, 1.27] where 1 indicates no resizing. For detectors trained with patches of resized images, the testing accuracy is 11% lower on average for GMM-based and 7% lower for CNN-based method, when compared with training on patches of original-sized images (see Table 10). In addition, we can notice that the improved accuracy after unsupervised GRAFT adaptation to pre-resizing is higher than the testing accuracy when we train GMMs on mixed data (around 85% for GRAFT in Tables 3–5 *vs.* 81% for "GMM on mix" in Table 10). We also keep in mind that the solution of training "GMM on mix" is more computationally expensive and less flexible than GRAFT adaptation. Moreover, we assume that there would be more accuracy decrease for "GMM on mix" if we consider more scaling factors and more interpolation algorithms during training. One explanation is that training on a mix of resized data is harder as it introduces a lot of variability in patches. This variability has then to be disentangled from manipulation fingerprints. Authors of [4] notice similar behaviour in the close field of steganalysis. Their conclusion is that training on a mix of development processes is not as powerful as identifying the specific development process and conducting training specifically for it. Extension of this idea to image forensics could be an interesting future work.

TABLE 10. Testing accuracy (in %) for GMM- and CNN-based methods. We compare testing accuracy when training set contains original-sized images and when it contains resized images of a mix of resizing factors. Factors are selected randomly from a set of values (see text). Testing set is consistent with training set.

	GF	MF	USM	WGN	JPEG	AVG
GMM original-size	91	86	97	98	89	92
GMM on mix	65	85	89	86	80	81
CNN original-size	79	85	91	86	79	84
CNN on mix	65	74	84	89	74	77

F. SUMMARY

From the above experiments and analysis, we can see that GMM adaptation is effective for the forensics problem of detecting manipulation on patches of pre-resized images. It also seems to be a better approach than training on a mix of resizing factors. After applying our proposed GRAFT method, the detector provides in general satisfying testing accuracy, especially under upscaling, where the accuracy improvement can be as high as +32% for GF. Additional experimental results (*e.g.*, under nearest neighbor interpolation, base scores obtained by training GMMs from scratch on patches of pre-resized images, *etc.*) are included in the Supplementary Material. At last, the different behaviours of GMM, SPAM and CNN under different resizing factors (and also under different manipulations) suggest that an appropriate fusion of the three approaches could lead to better performances.

VI. CONCLUSIONS

This study highlights some concerns on how resizing, a common pre-processing operation, alters performances of detectors based on local statistical model of images, on classical steganalytic features and on convolutional neural network. Yet, these detectors are state-of-the-art approaches for detecting image manipulations. We expose some statistical differences between training and (pre-resized) testing samples, as an explanation to this performance drop of various detectors. We then propose an effective adaptation algorithm named GRAFT for GMM-based detector. GRAFT is able to improve results in the challenging scenario of unsupervised adaptation, without using any ground-truth label in target domain. In addition, according to our experiments, GRAFT also provides better result than training models from scratch on a mix of original-sized and resized images. The good performance of GRAFT is mainly due to the formulation of unsupervised adaptation as a problem of likelihood maximization of GMMs on pseudo-labeled target samples.

This work raises new questions on the weaknesses of existing forensics detectors regarding their generalization capabilities to pre-processing and more generally image development processes, which is a timely and important research problem. It would be interesting to study the effect of pre-resizing on detection of more complex tampering operations like splicing or copy-move. In machine learning, in order to obtain better accuracy and generalization, one usually uses larger and more diverse databases. However, this may not be the best answer here and is even sometimes impossible due to lack of labeled target samples. An alternative and flexible solution is therefore needed as illustrated in this paper. Semi/weakly-supervised approaches are also promising ideas for future work toward this direction. This would make a step forward in moving image forensics toward real-world applications.

ACKNOWLEDGMENT

The authors would like to thank Dr. Patrick Bas for the discussions on cover-source mismatch for steganalysis [4] and Dr. Pedro Rodrigues for inspiring discussions about his work on Riemannian geometry [25].

REFERENCES

[1] A. Piva, "An overview on image forensics," *ISRN Signal Process.*, vol. 2013, Jan. 2013, Art. no. 496701.

[2] M. C. Stamm, M. Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, vol. 1, pp. 167–200, 2013.

[3] J. Kodovsky and J. Fridrich, "Effect of image downsampling on steganographic security," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 5, pp. 752–762, May 2014.

[4] Q. Giboulot, R. Cogranne, and P. Bas, "Steganalysis into the wild: How to define a source?" in *Proc. SPIE Media Watermarking, Secur., Forensics*, vol. 318, 2018, pp. 318:1–318:12.

[5] W. Fan, K. Wang, and F. Cayre, "General-purpose image forensics using patch likelihood under image statistical models," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Nov. 2015, pp. 1–6.

[6] L. Darmet, K. Wang, and F. Cayre, "GRAFT: Adaptation non-supervisée au re-dimensionnement pour la détection de manipulation d'image," presented at the Colloque GRETSI, 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02335837/document

[7] H.-D. Yuan, "Blind forensics of median filtering in digital images," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 4, pp. 1335–1345, Dec. 2011.

[8] X. Kang, M. C. Stamm, A. Peng, and K. J. R. Liu, "Robust median filtering forensics using an autoregressive model," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 9, pp. 1456–1468, Sep. 2013.

[9] Z. Fan and R. L. de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 230–235, Feb. 2003.

[10] W. Luo, J. Huang, and G. Qiu, "JPEG error analysis and its applications to digital image forensics," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 480–491, Sep. 2010.

[11] G. Cao, Y. Zhao, R. Ni, and X. Li, "Contrast enhancement-based forensics in digital images," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 3, pp. 515–525, Mar. 2014.

[12] G. Cao, Y. Zhao, R. Ni, and A. C. Kot, "Unsharp masking sharpening detection via overshoot artifacts analysis," *IEEE Signal Process. Lett.*, vol. 18, no. 10, pp. 603–606, Oct. 2011.

[13] J. Chen, X. Kang, Y. Liu, and Z. J. Wang, "Median filtering forensics based on convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1849–1853, Nov. 2015.

[14] X. Qiu, H. Li, W. Luo, and J. Huang, "A universal image forensic strategy based on steganalytic model," in *Proc. 2nd ACM Workshop Inf. Hiding Multimedia Secur. (IH&MMSec)*, 2014, pp. 165–170.

[15] H. Li, W. Luo, X. Qiu, and J. Huang, "Identification of various image operations using residual-based features," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 1, pp. 31–45, Jan. 2018.

[16] B. Bayar and M. C. Stamm, "Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2691–2706, Nov. 2018.

[17] T. Pevný, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 215–224, Jun. 2010.

[18] J. Fridrich and J. Kodovský, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.

[19] L. Darmet, K. Wang, and F. Cayre, "Weakly supervised adaptation to re-sizing for image manipulation detection on small patches," in *Proc. Int. Workshop Digit.-Forensics Watermarking*, 2019, pp. 1–16. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02335866/document

[20] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva, "ForensicTransfer: Weakly-supervised domain adaptation for forgery detection," 2018, *arXiv:1812.02510*. [Online]. Available: http://arxiv.org/abs/1812.02510

[21] J. Kodovský, V. Sedighi, and J. Fridrich, "Study of cover source mismatch in steganalysis and ways to mitigate its impact," *Proc. SPIE Media Watermarking, Secur., Forensics*, vol. 9028, pp. 90280J:1–90280J:12, Feb. 2014.

[22] A. Ker and T. Pevný, "A mishmash of methods for mitigating the model mismatch mess," *Proc. SPIE Media Watermarking, Secur., Forensics*, vol. 9028, pp. 90280I:1–90280I:15, Feb. 2014.

[23] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[24] P. Zanini, M. Congedo, C. Jutten, S. Said, and Y. Berthoumieu, "Transfer learning: A Riemannian geometry framework with applications to brain–computer interfaces," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 5, pp. 1107–1116, May 2018.

[25] P. L. C. Rodrigues, C. Jutten, and M. Congedo, "Riemannian procrustes analysis: Transfer learning for brain–computer interfaces," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 8, pp. 2390–2401, Aug. 2019.

[26] A. Gretton, K. M. Borgwardt, M. Rasch, B. Scholköpf, and A. J. Smola, "A kernel approach to comparing distributions," in *Proc. Nat. Conf. Artif. Intell.*, 2007, pp. 1637–1641.

[27] A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Scholköpf, and A. J. Smola, "A kernel statistical test of independence," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2008, pp. 585–592.

[28] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 479–486.

[29] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 137–144.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.
[31] P. Knysh and Y. Korkolis, "Blackbox: A procedure for parallel optimization of expensive black-box functions," 2016, *arXiv:1605.00998*. [Online]. Available: http://arxiv.org/abs/1605.00998
[32] T. Gloe and R. Böhme, "The Dresden image database for benchmarking digital image forensics," in *Proc. ACM Symp. Appl. Comput.*, 2010, pp. 1585–1591.
[33] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2016, pp. 785–794.
[34] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1853–1865, Sep. 2017.
[35] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

LUDOVIC DARMET (Student Member, IEEE) received the master's degree from the École Centrale de Lille, Lille, France, in 2017. He is currently pursuing the Ph.D. degree with the GIPSA-Lab, Grenoble, France. His current research interest includes digital image forensics.

KAI WANG received the Ph.D. degree in computer science from the University of Lyon, Lyon, France, in 2009. He was a Postdoctoral Research Fellow at the Inria, Nancy, France, in 2010. In 2011, he joined the French National Centre for Scientific Research (CNRS), as a full-time Researcher at the GIPSA-Lab, University of Grenoble Alpes, Grenoble, France. His current research interests include multimedia security and shape analysis.

FRANÇOIS CAYRE received the Ph.D. degree from Telecom ParisTech, Paris, France, and the Université Catholique de Louvain, Louvain-la-Neuve, Belgium, in 2003. He was a Postdoctoral Fellow with the Institut National de Recherche en Informatique et Automatique, Rennes, France, until 2005. He joined the Grenoble Institute of Technology, Grenoble, France, in 2005, as an Assistant Professor. His current research interests include watermarking security and multimedia security at large.