Published in IET Image Processing
Received on 13th May 2014
Revised on 8th August 2014
Accepted on 9th September 2014
doi: 10.1049/iet-ipr.2014.0316

ISSN 1751-9659

Acquisition source identification through a blind image classification

Irene Amerini[1], *Rudy Becarelli*[1], *Bruno Bertini*[1], *Roberto Caldelli*[1,2]

[1]*Media Integration and Communication Center, University of Florence, Italy*
[2]*CNIT – National Interuniversity Consortium for Telecommunications, Parma, Italy*
E-mail: irene.amerini@unifi.it

Abstract: Image forensics, besides understanding if a digital image has been forged, often aims at determining information about image origin. In particular, it could be worthy to individuate which is the kind of source (digital camera, scanner or computer graphics software) that has generated a certain photo. Such an issue has already been studied in literature, but the problem of doing that in a blind manner has not been faced so far. It is easy to understand that in many application scenarios information at disposal is usually very limited; this is the case when, given a set of L images, the authors want to establish if they belong to K different classes of acquisition sources, without having any previous knowledge about the number of specific types of generation processes. The proposed system is able, in an unsupervised and fast manner, to blindly classify a group of photos without neither any initial information about their membership nor by resorting at a trained classifier. Experimental results have been carried out to verify actual performances of the proposed methodology and a comparative analysis with two SVM-based clustering techniques has been performed too.

1 Introduction

Digital images can be easily manipulated by common users for disparate purposes so that origin and authenticity of the digital content we are looking at is often very difficult to be assessed without uncertainty. Hence technological instruments which allow to give answers to basic questions regarding image source and image originality are needed [1, 2]. However, by focusing on the task of assessing image origin, the aspects to be studied are mainly two: the first one is to understand which kind of source has generated that digital image (e.g. a scanner, a digital camera, a computer graphics software) [3–6] and the second one is to succeed in determining which is the specific sensor that has acquired such a content (i.e. the specific brand and/or model of a camera) [7, 8]. In this paper, the first aspect is focused. The generally adopted approach is to extract from digital images, some robust and identifying features which can permit to distinguish the various classes of devices (e.g scanned images, photos, computer generated). Such features are distinctive because they exploit some characteristic traces left over the digital content during the operation of image creation as described in [9]. Usually, such features are extracted from a training set of images whose provenance is known and used to train a classifier (e.g. SVM). The trained classifier then is able to evaluate a digital asset and establish which category it belongs to. The first approach in this sense was introduced by Lyu and Farid [5]. In this study, the authors use a statistical model of 216-dimensional feature vectors calculated from the first four order statistics of the wavelet decomposition to discriminate between computer generated and natural images. Based on the estimation of the noise pattern of the devices Khanna *et al.* presented in [3] a method for discriminating between scanned, non-scanned and computer generated images. The basic idea in [3] is to analyse noises of the scanner from row to row and column to column and then combining them with the noise of the camera, calculated as difference between the de-noised image and the input one. A different approach was proposed in [4] where a fusion of a set of device colour interpolation coefficients and noise statistics to differentiate between images produced by cameras, cell phone cameras, scanners and computer graphics is introduced.

In this paper, on the other hand, the problem of blindly grouping images belonging to a given set according to the kind of acquisition source is faced (see a sketch of the idea in Fig. 1). The aim of this method is to quickly identify in a generic bunch of photos, taken (created) by different sources, which of them have been acquired by a camera or a scanner or computer-generated by means of a computer graphics software, without any type of previous knowledge. This has already been studied in literature to blindly distinguish among brands and models of a certain device (e.g. a digital camera) [10–12], but never, to the best of our knowledge, to classify images according to their acquisition sources. It is out of the scope of the present work to distinguish among different brand/model relates to each source device.

The paper is organised as follows: Section 2 provides some motivations which are behind the idea of using this kind of blind clustering also in relation with a training-based

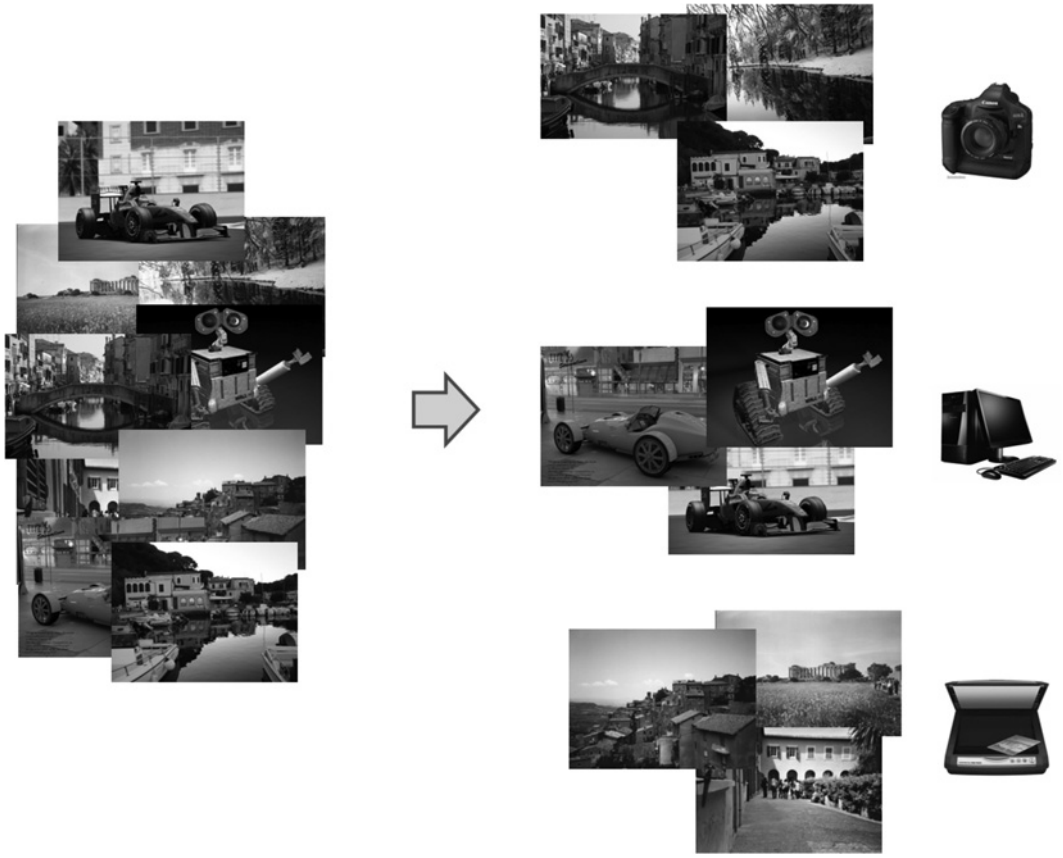

Fig. 1 *Clusters obtained with our method starting from a bunch of images without any initial information about their membership*

classification and Section 3 introduces the proposed method. In Section 4 experimental results are presented and Section 5 concludes the paper.

2 Motivations

In practical forensic applications, very often it is necessary to reliably discriminate between natural and computer generated multimedia data [13, 14], because of the fact that computer softwares are able to create highly realistic image contents that can be, for a human being, impossible to be distinguished from the natural ones. To succeed in discerning between highly realistic CG images and images acquired by digital cameras would be invaluable especially in real forensic scenario such as child pornography [15]. Furthermore separating scanned images [6] from the others, it is also important because the knowledge of an image acquisition source can be helpful in determining the authenticity and the provenance of the image, as well as in tracing back who was responsible for creating it.

Blind image clustering by means of unknown acquisition sources could be also interesting in circumstances similar to the following. Let us consider the case, for instance, in which a group of images are attributed to a person (e.g. found in his hard disk drive) because apparently taken by his personal camera (e.g. names of digital files, EXIF of the photos etc.) and let us imagine that establishing if some of these pictures have been fraudulently inserted within this set is crucial for steering the investigation in a specific direction. A forensic analyst could be willing to carry out an initial analysis to discern those images taken by a camera from those ones artificially prepared through a computer graphic software and/or digitised by means of a scanner, without having any information about the possible composition of such image set (we assume that the EXIF information of such images cannot be trusted). Once images acquired by a device, that is, a digital camera, are individuated through our proposed blind method, other forensic tools such as that one presented in [10], could be launched onto such a subset to determine how many cameras have generated those images: this procedure could also improve performances of the second method.

In addition to this, when operating in a real forensic scenario, often available instruments are limited and knowledge at disposal are a few; hence in such a situation there is the need to perform an analysis restricted to the dataset under examination without any side information. On the other side, it could be remarked that working 'blindly' could be not only an operating requirement but also the chance to work without depending on, for example, a trained classifier that, although it should offer superior performances, it is inevitably influenced by the characteristics of the images used for training according to their format, quality, compression, image content (textured/smooth), acquisition device, processing tools and so on. This is especially true when the distinctive features, adopted for the classification, are strongly dependent from the image content (e.g. denoising features, DWT features etc., see Section 3.1). Furthermore, training-based classifiers have to be completely retrained if just one new distinctive feature is included to improve class characterisation and/or if a different kind of image has to be added (e.g. computer graphic images generated with superimposed a specific pseudo-realistic noise) into the clustering process. In both cases, blind clustering is easier to be applied, but,

doi: 10.1049/iet-ipr.2014.0316

nevertheless, it is not alternative to training-based classification, in fact this last could be used for a more in-depth analysis (e.g. link between obtained cluster and the type of device) and, above all, when stronger operative resources are available.

3 Proposed method

We present a novel approach for blind grouping images taken by different sources based on a spectral partitioning [16]. A schema of the whole system is pictured in Fig. 2. The first step consists in a pre-processing phase, performed on the whole dataset, to extract some features distinctive of each class and then a similarity matrix is computed. In the second step, a clustering procedure is performed allowing to distinguish among different devices.

In particular, for each image I ($I = 1, \ldots, L$) in the dataset, a feature vector $\boldsymbol{v}_i$ is computed, composed by 111 elements (37 features for each RGB component). The considered features will be described in detail in the following Subsection 3.1. However, it is worth to point out that the selected features are fundamentally a mixture of selected features coming from [3, 4]; the main guideline in such a choice was mainly led by the need to find a trade-off between distinctiveness and computational burden. As we will show in Section 4.3 the choice of our features set has been demonstrated a very good selection respect to considering the features [3, 4] separately. In addition to this, a diverse denoising filter [17] was used to better take into account characteristics of the sensor related to sensor pattern noise. In particular, features based on colour-filter-array, proposed in [4], have been discarded: this was done basically because the computational burden was too heavy compared with the actual improvement given in terms of performance in the classification. Such experimental results have been preparatory for the definition of the whole procedure and they are included in Sections 4.3 and 4.4.

After the features have been calculated then the similarity matrix $\boldsymbol{S}$ is computed. In our approach, the similarity matrix is specified by the normalised correlation between the feature vectors of each image. The similarity matrix afterward will be a $L \times L$ symmetric matrix (see the end of Subsection 3.1 for details).

After the similarity matrix $\boldsymbol{S}$ has been computed, the clustering phase is performed. In particular, the spectral partitioning class of algorithms is exploited [16]. At the end of the clustering procedure, the number of clusters K is obtained, which represents the actual number of sources which generated the given L images.

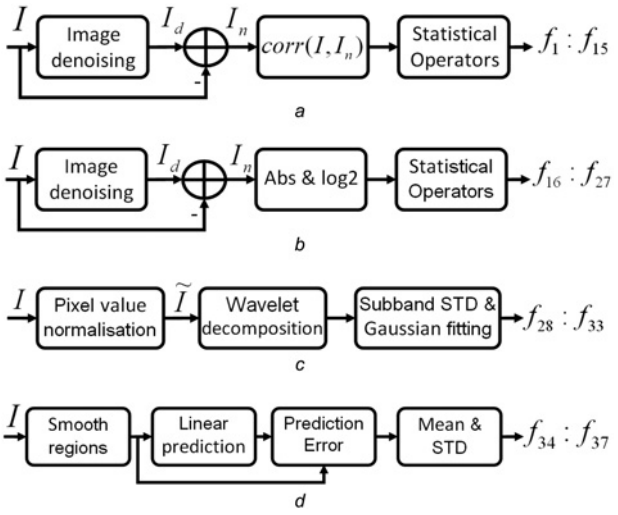

Fig. 3 *Feature extraction for each RGB component of an image*
a and *b* From image denoising
c From DWT analysis
d From linear prediction

3.1 Pre-processing step: features extraction and similarity matrix computation

It is possible to subdivide the features taken into account in three different classes (an overview is reported in Fig. 3). The first class of the considered features is based on the idea that a residual pattern noise exists in images obtained from digital cameras and scanners. On the contrary, the residual noise present in computer generated images does not have structures similar to the pattern noise of cameras and scanners. A number of 15 features is obtained for each image accordingly to the paper in [3] for each RGB component (see Fig. 3*a*).

The mean, standard deviation, skewness and kurtosis of $\rho_{\text{row}}(i)$ and $\rho_{\text{col}}(j)$ are the first eight features $[f_1; f_8]$, that is, the correlation between the noise residual I_{n}, obtained with the Mihcak filter [17] from the original image $I(M \times N)$ and I_{n}^{r}, I_{n}^{c}, respectively, which have been computed by the

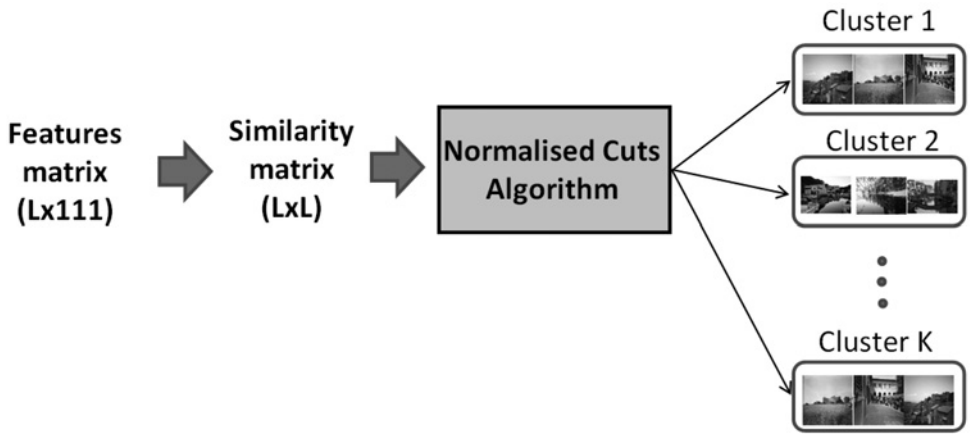

Fig. 2 *Outline of the proposed framework*

following equation

$$I_{\mathrm{n}}^{\mathrm{r}}(1,j)=\frac{1}{M}\sum_{i=1}^{M}I_{\mathrm{n}}(i,j);\ \ 1\le j\le N \tag{1}$$

$$I_{\mathrm{n}}^{\mathrm{c}}(i,1)=\frac{1}{N}\sum_{i=1}^{N}I_{\mathrm{n}}(i,j);\ \ 1\le i\le M \tag{2}$$

The standard deviation, skewness and kurtosis of $I_{\mathrm{n}}^{\mathrm{r}}$ and $I_{\mathrm{n}}^{\mathrm{c}}$ correspond to features $[f_9{:}f_{14}]$. The last feature for every input image is given by the following:

$$f_{15}=\left(1-\frac{\sum_{i=1}^{N}\rho_{\mathrm{col}}(j)}{\sum_{i=1}^{M}\rho_{\mathrm{row}}(i)}\right)*100 \tag{3}$$

Others features, based on image denoising, are obtained by applying different types of denoising algorithms to the input image and by computing the mean and the standard deviation of the $\log_2$ of the estimated noise magnitudes (see Fig. 3*b*). We applied five different denoising techniques accordingly to [4]: linear filtering with an average filter of size 3×3, with a Gaussian filter of size 3×3 and standard deviation 0.5, median filtering 3×3, Wiener adaptive image denoising (kernel sizes 3×3 and 5×5) and, in addition to what happens in [4], a Mihcak filter [17] as sixth one. These six denoising algorithms capture different statistical properties of the sensor noise giving us 12 different features $[f_{16}{:}f_{27}]$.

To achieve the second class of features (see Fig. 3*c*), discrete wavelet transform (DWT) is applied to the image, to measure the statistical properties of sensor noise in the frequency domain [4].

As a first step, the image is normalised as

$$\widetilde{I(i,j)}=\frac{I(i,j)}{\left(\frac{1}{MN}\sum_{k=1}^{M}\sum_{l=1}^{N}I(k,l)^2\right)^{1/2}} \tag{4}$$

Then a two-dimensional one level wavelet decomposition of $\widetilde{I}$, to obtain the LH_1, HL_1, HH_1 sub-bands, is performed. The mean μ and the variance σ^2 of the sub-band coefficients are computed. The variances constitute the first three features (f_{28}, f_{29}, f_{30}) of the second set of features. Then the goodness of fitting a Gaussian distribution $N(\mu_Y,\sigma_Y^2)$ to the distribution of each wavelet sub-band coefficients is quantified to obtain additional features. Let $p(y)$ and $q(y)$ denote the probability density functions of the Gaussian distribution $N(\mu_Y,\sigma_Y^2)$ and the distribution of the sub-band wavelet coefficients, respectively, we quantify the goodness of Gaussian fitting by measuring the distance between $p(y)$ and $q(y)$ as

$$\sum_i\left|p(y_i)-q(y_i)\right|\Delta y \tag{5}$$

where i is the index of the histogram bins, Δy is the width of the bin $\sum_i q(y_i)\Delta y=1$ and $p(y_i)$ and $q(y_i)$ denote the histogram values at the bin centres. This gives three other features (f_{31}, f_{32}, f_{33}).

Finally, the last class of features is obtained from neighbourhood prediction by measuring the error in the prediction of neighbouring pixels in smooth regions of the image [4] (see Fig. 3*d*). This is based on the observation that the image acquisition noise in the smooth regions of the image results in prediction errors which can provide forensic evidence about the origin of the image. Given an image I, we first normalised it to obtain $\widetilde{I}$, as in (4) and identified its smooth regions by comparing the local gradient values with a threshold (in our implementation the Prewitt operator with a threshold equal to 2 has been used). The pixels in the chosen smooth regions are then expressed as a weighted summation of its neighbourhood values to obtain a set of linear equations. More specifically, each pixel value b_i in a given region is predicted using a linear model on its eight surrounding neighbours

$$\widehat{b}_i=\sum_{k=1}^{8}x_k a_{i,k} \tag{6}$$

The absolute prediction errors are then obtained as $\Delta b=\left|\hat{b}-b\right|$. The mean and the variance of Δb computed in smooth regions, subdivided in bright and dark areas according to the median of normalised intensities of all image pixels, constitute our third set of four statistical noise features ($f_{34}, f_{35}, f_{36}, f_{37}$).

Hence the number of the extracted features is totally of 111 (37×3 bands) per image and a vector of feature $\boldsymbol{v}_i$, whose dimension is 111, can be computed to represent every image. Being L the number of images belonging to the dataset, a similarity matrix $\boldsymbol{S}$ can be obtained by calculating the normalised correlation (represented by $\otimes$ in (7)) among the feature vectors of each image. The similarity matrix afterwards will be a $L\times L$ symmetric matrix. The diagonal elements of $\boldsymbol{S}$ will be 1 representing the auto-correlations.

$$s_{i,j}=v_i\otimes v_j;\quad i,j=1,2,\ \ldots,L \tag{7}$$

3.2 Clustering step: the normalised cuts algorithm

The basic idea of this spectral partitioning method is to refer to the to-be-clustered images as points in a feature space that can in turn be regarded as nodes in a weighted undirected graph [16]. The edges, connecting each pair of nodes/images, are weighted by means of a chosen similarity function $w(i,j)$, being i and j two nodes of the graph. In our case the weights $w(i,j)$ are the correlations stored in the matrix $\boldsymbol{S}$, described in the previous Section 3.1. A graph $G=(V,E)$ is partitioned into two disjoint graphs A and B by simply removing edges connecting the two parts. The total weight of the edges removed during partitioning gives a computation of the degree of dissimilarity between these two parts: this is called the cut value and computed as

$$\mathrm{cut}(A,B)=\sum_{u\in A,v\in B}w(u,v) \tag{8}$$

In [16], the authors propose a 'disassociation measure', as a fraction of the total edge connections to all the nodes in the graph, called the normalised cut (Ncut) and defined as in the following

$$\mathrm{Ncut}(A,B)=\frac{\mathrm{cut}(A,B)}{\mathrm{assoc}(A,V)}+\frac{\mathrm{cut}(A,B)}{\mathrm{assoc}(B,V)} \tag{9}$$

where the 'association measure' (assoc(.,.)) represents the total connections from nodes in A to all nodes in the graph

doi: 10.1049/iet-ipr.2014.0316

Table 1 Device/software for each different source

Sources	Device/software	Amount
camera	Nikon D80 (10 Mp)	10
	Nikon Coolpix S220 (10 Mp)	10
	Sony DSC-W130 (8 Mp)	69
	Canon PS A570 (5 Mp)	11
CG	Maya, AutoCAD, Photoshop, Cinema 4D, 3D studio max	miscell.
scanner	Canon CanonScan Lide 60	50
	Canon CanonScan L110	50

(similarly for association of B). The number of groups segmented by this method is controlled directly by the maximum allowed Ncut that represents the threshold τ of the proposed method. See [16] for more details of the implementation and Section 4.1 for considerations about the threshold evaluation.

4 Experiments

To verify the performances of the presented blind clustering procedure, the system has been tested on a dataset populated by 300 pictures taken by three different sources (100 from four diverse digital cameras, 100 computer graphics generated and 100 from two different scanners, as reported in detail in Table 1). In particular, the digital cameras have been set to their maximum resolution which was obviously different one from each other, ranging from 5 M pixels to 10 M pixels and stored in JPEG format; computer graphics images were downloaded from the web (www.realsoft.com, www.3dlinks.com, www.maxon.net and Google Images) in JPEG format. The scanned images have been acquired at two different resolutions of 600 and 800 dpi and saved in TIFF format. For each image a central 1024×1024 block is used for features extraction, then we computed the 111 features, described in Section 3.1, then we create the similarity matrix $\boldsymbol{S}$. Initial tests, although not reported in the paper, have been done to define the dimension of the central block to take and the choice of 1024×1024 has been deemed a good trade-off between the amount of data to analyse and computational complexity. In Table 1 are reported in detail the sources of the different classes of digital images used in our experiments.

Images contain very general contents and represent landscapes, groups of persons, buildings, cars, objects, fruits and so on. Both highly textured and smooth images are present in the dataset (e.g. large zones picturing sky or sea).

Table 2 Composition of the 21 test sets

Cam	CG	Scan	Cam	CG	Scan	Cam	CG	Scan
30	100	90	70	40	100	20	80	80
40	100	80	70	50	90	30	80	70
50	100	70	70	60	80	40	80	60
60	100	60	70	70	70	50	80	50
70	100	50	70	80	60	60	80	40
80	100	40	70	90	50	70	80	30
90	100	30	70	100	40	80	80	20

Permutation of each test set over the three classes (e.g. 30-100-90/100-30-90/30-90-100) generates 63 data-sets.

Table 3 TPR/FPR for different values of τ and distances of the couple (TPR, FPR) from (0,1) in the ROC space

τ	TPR	FPR	Distance
0.82	0.77	0.09	0.244
0.83	0.75	0.11	0.272
0.84	0.79	0.10	0.235
0.85	0.81	0.11	0.220
0.86	0.77	0.14	0.271
0.87	0.76	0.17	0.289
0.88	0.76	0.16	0.287
0.89	0.73	0.17	0.324
0.90	0.70	0.20	0.357
0.91	0.66	0.25	0.422
0.92	0.61	0.25	0.467

To augment the number of experimental tests, such set of 300 images has been used as a repository to generate different test-sets with diverse characteristics. In particular, according to Table 2, 21 distributions of images among the three classes (camera, CG and scanner) have been chosen (all the three classes/sources are represented in this case). To achieve the highest variability, the number of elements within each category is very different, the total number of images over the three groups is not always the same and, above all, photos are randomly selected each time from the initial set of 300. By permuting the distributions in Table 2 onto the three classes, 63 data-sets have been obtained finally for a global amount of around 13 000 images. Such 63 data-sets have constituted the ground truth for the experimental tests whose results are presented in the sequel of Section 4. More specifically, each of the following test will be launched over every dataset and a confusion matrix depicting the classification performance will be achieved.

4.1 Threshold evaluation

First of all, as evidenced in Section 3.2, it is necessary to set the best working threshold τ, relatively to the proposed procedure, to be used afterwards during all the successive comparative tests. To do so we have resorted at the 63 data-sets. For each experiment a confusion matrix is achieved and then the 63 confusion matrixes are averaged obtaining the overall result. Confusion matrix computation (see Table 4 for reference) permits to individuate, with respect to a ground truth, how many elements of a class have been correctly classified (values on the diagonal) and how many have been wrongly assigned to the other classes (values out of the diagonal). In this section, we set up the value of the threshold τ by means of receiver operating characteristic (ROC) curves through the analysis of the performance, in terms of true positive rate (TPR) and false positive rate (FPR), on the 63 data-sets. More in detail: the TPR is a measure stating how many images are allocated in the correct cluster, with respect to the cardinality of the cluster, known by the ground truth; similarly, the FPR

Table 4 Confusion matrix with $\tau = 0.85$ obtained with the proposed method

(%)	Camera	CG	Scanner
camera	86.0	1.9	12.1
CG	7.8	83.8	8.4
scanner	11.0	16.7	72.3

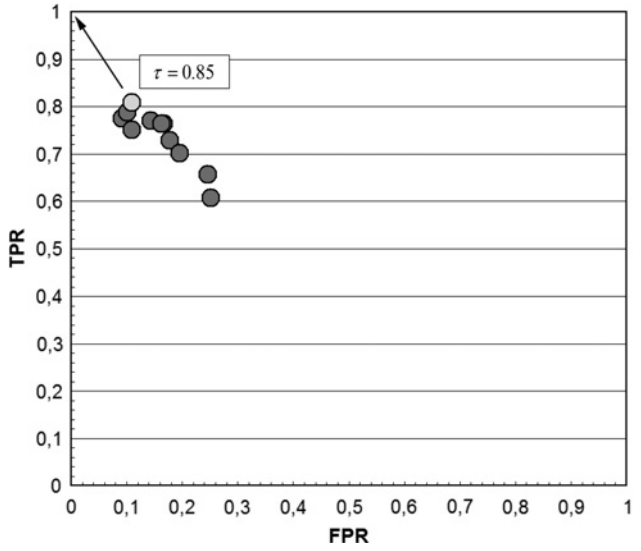

Fig. 4 *ROC space*

states how many images are erroneously assigned to a cluster, with respect to the number of images actually not belonging to it. In formulae, TPR and FPR can be computed as in (10)

$$\begin{aligned} \text{TPR} &= \text{TP}/(\text{TP} + \text{FN}) \\ \text{FPR} &= \text{FP}/(\text{FP} + \text{TN}) \end{aligned} \tag{10}$$

where the used notations have the following meaning:

- TP (true positives): the number of images that are assigned to the correct cluster.
- FN (false negatives): the number of images that are not assigned to the correct cluster.
- FP (false positives): the number of images that are assigned to a wrong cluster.
- TN (true negatives): the number of images correctly not assigned to a cluster.

According to the ROC curve approach, the threshold τ was varied within the range [0.7:0.95] with a step of 0.01 (such a working range has been set up by means of preliminary trials). For each fixed value of τ, the proposed method was launched over the 63 data-sets, as explained previously and a final confusion matrix has been achieved; from such confusion matrix a couple of (TPR, FPR) for each of the found classes can be computed according to (10) and then by averaging over the classes a couple (TPR, FPR) is determined. Such couple represents a point in the ROC space (TPR against FPR) as pictured in Fig. 4.

Hence the best possible threshold τ would yield a point in the upper left corner of the ROC space, corresponding to FPR = 0 (no FP) and TPR = 1 (no FN). By using a criterion of minimum euclidean distance from the ideal point located in the upper left corner, we can select $\tau_{\text{opt}} = 0.85$ as the best value. This can also be appreciated in Table 3 where the distances with respect to the ideal point, TPR and FPR according to different τ values are shown (only a sub-range of τ values is reported). Thus having selected the optimal threshold τ_{opt} that will be used for all the experimental tests, we can also check the performances of the proposed method in such case; the confusion matrix for the selected threshold is reported in Table 4. In such a table and in the following, the names Camera, CG and scanner refer at the knowledge of the ground truth: the method, being completely blind is able to clustering but not to assign the type of source to each obtained cluster. Such result points out a good performance, in fact 86% of the images, acquired by a camera, are correctly assigned, whereas those ones generated by computer graphics achieve a percentage of 84% and 72% is obtained for the scanner case. This last percentage for the scanner is a bit discrepant with respect to the other two, but, however, it is in line, in terms of general trend, with what happens also when other clustering algorithms are used (e.g see Table 6). This behavior seems to be determined by the blind clustering procedure that is not based on a trained classifier, in fact, this does not happen when SVM-based techniques are considered (see Table 9).

Once the threshold τ has been fixed at 0.85, we have launched an experiment over a new dataset, composed by 300 images (100 for each of the three categories belonging to the same devices shown in Table 1) and results are reported in Table 5. It can be appreciated that results are in line with what expected.

Table 5 Confusion matrix with $\tau = 0.85$ obtained with the proposed method on a dataset of 300 images (100 Camera, 100 CG and 100 Scanner)

(%)	Camera	CG	Scanner
camera	85.0	3.0	12.0
CG	3.0	93.0	4.0
scanner	12.0	16.7	72.0

Table 6 Confusion matrix obtained with *k*-means without silhouette coefficient but knowing $K = 3$

(%)	Camera	CG	Scanner
camera	90.2	2.6	7.2
CG	4.1	93.0	2.9
scanner	3.6	11.6	84.8

4.2 Comparison between clustering procedures

After having adequately designed and set-up the proposed method, we have performed a comparison with a well-known clustering procedure: the *k*-means clustering. The *k*-means algorithm seeks to minimise the average squared distance between points in the same cluster [18]. The implementation adopted for *k*-means is that provided by Matlab Statistics Toolbox, setting the input parameters Distance = correlation, Replicates = 10 and other parameters at default. However, to be used in the proposed application scenario, it is necessary to overcome a major drawback of the *k*-means algorithm, namely the assumption that the number of clusters (sources) is given. This assumption is a crucial input for the *k*-means algorithm that affects both performance and accuracy. In fact, when *k*-means, knowing the number of expected clusters ($K = 3$), is itself tested over the 63 data-sets, the obtained results (see Table 6) show an undoubted performance improvement with respect to the proposed spectral clustering (see Table 4) that anyway does not require such initial information as input. However, this is not the case of our blind application scenario, in which no knowledge is given.

doi: 10.1049/iet-ipr.2014.0316

Table 7 Comparison between spectral clustering and *k*-means (input $K=3$) when the test set is composed by 100 images coming from a single source

(%)	Camera	CG	Scanner
proposed method	90.0	4.0	6.0
k-means	60.0	19.0	21.0

As a quick actual proof for that, if we consider a dataset composed by images coming from only one source (e.g. 100 images taken by digital cameras), the results for the *k*-means clustering with $K=3$, whose previous performances were very good, drastically decrease with respect to the proposed method, as shown in Table 7.

To overcome such a problem and adapt *k*-means to be usable in this framework, we employed the 'silhouette coefficient' [11] as a measure for the clustering quality and as an instrument to a-posteriori choose the best clustering situation on the basis of the number of K clusters. The silhouette coefficient of an element s_i is computed as follows: first, we compute the average distance of s_i from the other elements in the same cluster, let a_i denote this distance. Then, for each cluster C that does not contain s_i we compute the average distance of s_i from all the elements in C. Let b_i denote the average distance to these clusters. Then the silhouette coefficient of s_i is defined as

$$\mathrm{SC}_i = (b_i - a_i)/\max(a_i, b_i) \tag{11}$$

the value of the silhouette coefficient varies between -1 and $+1$. In our case, a value near -1 indicates that the image is badly clustered, whereas a value near $+1$ indicates that the image is well-clustered.

We apply this calculation at each loop of the algorithm and at every element in the data we are examining; more precisely: at the iteration q it is calculated a global measure of the silhouette coefficient SC_q by averaging the silhouette coefficients related to each element that belongs to a certain cluster and taking the average value with respect to all the current k clusters. Then, it is found the maximum coefficient over all the iterations obtained and the corresponding index q^* is chosen as the iteration that has to be taken as the best cluster partition.

In Fig. 5 is reported the values of the silhouette coefficient, computed each time onto the 63 data-sets, against different K input clusters in a range from $K=2$ till $K=30$ (K again represents the number of expected clusters).

Generally, the bigger (close to 1) the silhouette coefficient the better the clustering provided by the *k*-means algorithm:

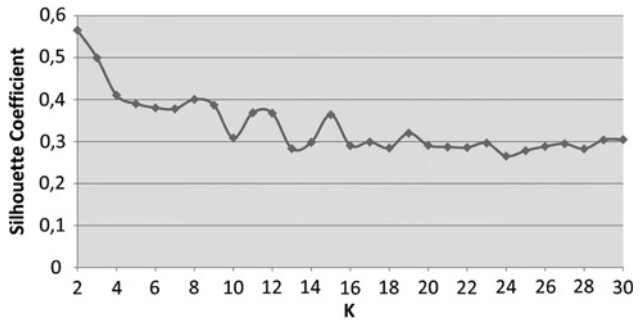

Fig. 5 *Silhouette coefficient values against K number of expected clusters*

Table 8 Confusion matrix obtained with *k*-means and choice of K by means of the silhouette coefficient

(%)	Camera	CG
camera	96.5	3.5
CG	4.2	95.8
scanner	52.5	47.5

hence the choice falls on $K=2$. Consequently, the confusion matrix for the *k*-means clustering evaluated for $K=2$ is reported in Table 8.

It is worth to point out that the *k*-means clustering with the use of silhouette coefficient fails to properly classify the three classes (camera, computer graphics and scanner) obtaining only two groups. In fact in Table 8, the two clusters obtained are assigned to Camera and Computer Graphics classes, respectively. On the other hand, the images of the class scanner are almost equally split in the Camera and Computer Graphics clusters. On the contrary, the proposed procedure, based on spectral clustering, properly detects the number of classes demonstrating the validity of the proposed procedure for a blind classification scenario as already demonstrated.

4.3 Comparison with SVM-based techniques

Finally we make a comparison of our proposed blind method (with $\tau=0.85$) against two well-known techniques for device identification based on SVM training [3, 4]. We used the dataset of 300 images (as before 100 images for each source): 80 images are used for the training set and 20 images for the test set for each kind of source. It is important to note that only 60 images are used to test our method since it does not require a training phase; this issue gives an advantage from the point of view of the execution time, especially regarding the pre-processing phase. Then we perform 50 experiments defined by different permutations of the training and test set whose cardinality always were of 80 and 20 images, respectively, for each source (a ratio of 4/1); this has determined that, at the end, $50\times60=3000$ images have been used for test session, against $50\times240=12\,000$ images used for the training session of the two SVM-based methods. It is straightforward that such circumstance is very challenging for the proposed method and that the SVM-based techniques are expected to perform better by resorting a trained classifier, but the intent was to comprehend how competitive was the proposed methodology hence to avoid the need of a learning phase. The results for each considered method are then averaged over the 50 experiments and reported in Table 9.

Finally, the result of our method is reported in Table 10. The obtained results, although are lower than the method presented in [4], as expected, are however in line with what obtained by the SVM-based method in [3]. In this case,

Table 9 Confusion matrix for the SVM-based technique in [3] and in [4], respectively, (result [3]/result [4]).

(%)	Camera	CG	Scanner
camera	74.4/**97.1**	17.8/3.3	7.8/0.0
CG	4.6/14.2	81.0/85.4	14.4/0.4
scanner	6.0/7.0	20.0/1.0	74.0/92.0

Table 10 Confusion matrix for the proposed technique (O_x states for spurious clusters)

(%)	Camera	CG	Scanner	O1
camera	83.6	1.8	9.4	5.2
CG	3.6	80.1	8.8	7.5
scanner	13.8	8.4	72.0	5.8

being the number of available images for the test phase quite limited (each time only 60), it can be observed that, sometimes, the proposed method tends to wrongly generate a fourth spurious cluster (O1) which contributes to worsen the performances. Anyway, it is necessary to point out again that our method works with no training information so the result could be considered very encouraging.

Finally, to have an idea of the impact of the features on the general performances, we have carried out a further experiment over the previous test set of 3000 images. We have substituted, within the proposed method, the features introduced by the work in [3] and in [4], respectively, to evaluate the obtained results when working in a blind application scenario (see Tables 11 and 12). It can be seen that in the first case (Table 11), images are overspread whereas in the second case (Table 12) images coming from camera and scanner are mixed. This behaviour shows that using a mixture of features improves the performance of our blind method and justifies such a choice. Furthermore our features selection also represents a good trade-off between distinctiveness and computational burden (see Section 4.4).

4.4 Considerations about execution time

First of all, it is necessary to highlight that from the point of view of computational time, our method is preferable with respect to SVM methods because, obviously, the time for training is not required at all. Also from the point of view of features extraction, which is the cumbersome part of the various computation phases, the proposed technique is competitive with the other two. In fact, with reference to the experimental tests presented in Section 4.3 which were obtained with code implementation in Matlab (Version R2010a for Windows 64 bits) running on a Dell Precision T1500 Intel Core i7 (CPU 2.80 GHz, RAM 4 GB), the average time per image for feature extraction was for the proposed method of 53s which is higher with respect to the 2.33s for the method in [3] but it is extremely lower than 6 m 39s for the one in [4]. This last aspect confirms again that the proposed procedure offers a good trade-off between source classification reliability and limited computational complexity, which, together to the chance to work without the need of a trained classifier, provide desirable characteristics for a real-life forensic scenario.

Table 11 Confusion matrix for the proposed technique but considering features proposed in [3] (O_x states for spurious clusters)

(%)	Cam	CG	Scan	O1	O2	O3	O4
cam	86.0	0.0	0.0	0.0	10.0	2.0	2.0
CG	12.0	48.0	6.0	8.0	12.0	4.0	10.0
scan	6.0	8.0	40.0	12.0	6.0	12.0	16.0

Table 12 Confusion matrix for the proposed technique but considering features proposed in [4] (O_x states for spurious clusters)

(%)	Cam	CG	Scan	O1	O2
cam	94.0	4.0	0.0	0.0	2.0
CG	36.0	54.0	4.0	0.0	6.0
scan	95.0	0.0	3.0	2.0	0.0

5 Conclusions

A new clustering procedure to classify a generic bunch of images according to the acquisition sources, without resorting at any side information, has been presented. Experimental results confirm that such proposed method permits to achieve good performances in terms of clustering reliability and is also competitive with SVM-based technique, although not requiring a trained classifier. It also allows good results for non-uniform data-sets demonstrating the adaptability of our method to a real case. Future works will be dedicated to improve the criterion for threshold τ selection and to individuate more distinctive features. Furthermore, the procedure will be evaluated in a more challenging scenario in which a dataset can be composed by images with different quality, compression and post-processing settings.

6 References

1 Birajdar, G.K., Mankar, V.H.: 'Digital image forgery detection using passive techniques: A survey', *Digit. Invest.*, 2013, **10**, (3), pp. 226–245
2 Mahdian, B., Saic, S.: 'A bibliography on blind methods for identifying image forgery', *Signal Process.: Image Commun.*, 2010, **25**, (6), pp. 389–399
3 Khanna, N., Chiu, G.T.-C., Allebach, J.P., Delp, E.J.: 'Forensic techniques for classifying scanner, computer generated and digital camera images', Proc. IEEE ICASSP, Las Vegas, NV, USA, March 2008, pp. 1653–1656
4 McKay, C., Swaminathan, A., Gou, H., Wu, M.: 'Image acquisition forensics: Forensic analysis to identify imaging source'. ICASSP'08, 2008, pp. 1657–1660
5 Lyu, S., Farid, H.: 'How realistic is photorealistic?', *IEEE Trans. Signal Process.*, 2005, **53**, (2), pp. 845–850
6 Caldelli, R., Amerini, I., Picchioni, F.: 'A DFT-based analysis to discern between camera and scanned images', *Int. J. Digit. Crime Forensics (IJDCF), e-Forensics*, 2009, **2**, (1), pp. 21–29
7 Lukás, J., Fridrich, J., Goljan, M.: 'Digital camera identification from sensor pattern noise', *IEEE Trans. Inf. Forensics Sec.*, 2006, **1**, (2), pp. 205–214
8 Bayram, S., Sencar, H.T., Memon, N.: 'Classification of digital camera-models based on demosaicing artifacts', *Digit. Investig.*, 2008, **5**, (1–2), pp. 49–59
9 Khanna, N., Mikkilineni, A.K., Martone, A.F., *et al.*: 'A survey of forensic characterisation methods for physical devices, Digital Investigation 3'. Proc. Sixth Annual Digital Forensic Research Workshop (DFRWS'06), 2006, pp. 17–28
10 Li, C.-T.: 'Unsupervised classification of digital images enhanced sensor pattern noise'. IEEE Int. Symp. on Circuits and Systems (ISCAS'10), 2010
11 Caldelli, R., Amerini, I., Picchioni, F., Innocenti, M.: 'Fast image clustering of unknown source images, in: Information Forensics and Security (WIFS)'. 2010 IEEE Int. Workshop on', 2010, pp. 1–5
12 Liu, B.B., Lee, H.-K., Hu, Y., Choi, C.-H.: 'On classification of source cameras: A graph based approach, in: Information Forensics and Security (WIFS)'. 2010 IEEE Int. Workshop on, 2010, pp. 1–5
13 Pan, F., Huang, J.: 'Discriminating computer graphics images and natural images using hidden markov tree model', in Kim, H.-J., Shi, Y., Barni, M. (Eds.): 'Digital watermarking, Vol. 6526 of Lecture Notes in Computer Science' (Springer, Berlin Heidelberg, 2011), pp. 23–28
14 Zhang, R., Wang, R.-D., Ng, T.-T.: 'Distinguishing photographic images and photorealistic computer graphics using visual vocabulary on local image edges', in Shi, Y., Kim, H.-J., Perez-Gonzalez, F. (Eds.): 'Digital forensics and watermarking, Vol. 7128 of lecture

doi: 10.1049/iet-ipr.2014.0316

notes in computer science' (Springer, Berlin Heidelberg, 2012), pp. 292–305
15 Farid, H.: 'Creating and detecting doctored and virtual images: Implications to the child pornography prevention act', Technical Report, 2004
16 Shi, J., Malik, J.: 'Normalised cuts and image segmentation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2000, **22**, (8), pp. 888–905
17 Mihcak, M., Kozintsev, I., Ramchandran, K.: 'Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising'. ICASSP, IEEE Int. Conf. on Acoustics, Speech and Signal Processing Proc., 1999, vol. 6, pp. 3253–3256
18 Kaufman, L., Rousseeuw, P.: 'Finding groups in data an introduction to cluster analysis' (Wiley Interscience, New York, 1990)