The Watchful Forensic Analyst: Multi-Clue Information Fusion with Background Knowledge

Marco Fontani #1, Enrique Argones-Rúa *2, Carmela Troncoso *3, Mauro Barni #4

Dept. of Information Engineering and Mathematical Sciences
University of Siena, Via Roma 56, 53100, Siena (IT)
[1] marco.fontani@unisi.it
[4] barni@dii.unisi.it

* *Gradiant (Galician R&D Center in Advanced Telecommunications)*
Campus Universitario de Vigo, 36310, Vigo, Pontevedra (ES)
[2] eargones@gradiant.org
[3] ctroncoso@gradiant.org

***Abstract*—Image Forensics (IF) is a challenging research topic, that suffers from strong limitations when facing with real world applications. A possible way to cope with these limitations is to resort to data fusion, whereby the outputs of different forensic tools are used to reach a final decision about the analyzed image. Nevertheless, existing schemes do not take full advantage of all the information available to the analyst, like the knowledge of the dependence of the performance of forensic tools on side conditions. Specifically, in this paper we show how the performance of forensic tools varies according to a number of parameters, most of which are directly observable by the analyst. After showing some practical examples, we propose a method to cast this background information into two multi-clue information fusion frameworks, yielding a significant improvement of the overall performance at virtually no cost.**

I. Introduction

Nowadays the majority of images are created, stored and distributed in a digital format that is fairly easy to edit and tamper with. As a result, digital image forensics has become an important research field, that aims at proving the authenticity and integrity of digital images. Many algorithms have been proposed to tackle this problem [1], especially focusing on JPEG images. In general, however, forensics algorithms are presented as stand alone tools, focusing on the detection of the trace of a particular kind of forgery; for instance, some techniques are applicable only if the image has been compressed twice, while others require that the image has never been compressed.

Since the forensic analyst does not know in advance which kind of processing may have been applied to produce a forgery, a scrupulous analysis requires the use of several tools, in order to detect as many different traces as possible. This fact fostered the development of information fusion systems tailored for image forensics, that allow to merge the results stemming from different analysis tools. Some of these systems fuse the information at the so-called *feature level*, that is, by devising a complex classifier that accounts for multiple footprints [2] [3]. Recently, two approaches have been proposed working at the *score level*, meaning that the scalar output of the tools is considered during fusion [4], [5].

When performing information fusion, quantifying the reliability of each tool is a crucial need, especially in situations where conflicts between the output of the tools arise. It may be the case, for example, that one specific tool is reliable when the analyzed image has been compressed only slightly, while another tool may be more robust to compression, and so on. When the analyst has access to some background information that sheds some light on the reliability of each tool, he can potentially improve the performance by a great amount. This fact has been investigated in the biometric research field: Kryszczuk and Drygajlo proposed to use a so-called "Q-stack" classifier, that takes into account quality measures of the input signal to improve the classification performance [6]; more recently, Argones-Rúa et al. derived a necessary and sufficient condition for reducing error when introducing quality-based score normalisation, and presented a score normalising technique for the speaker identification problem [7]. If we turn to image forensics, and to the best of our knowledge, [4] and [5] are the only works containing a first intuition of this issue, since the analyst can specify a "reliability score" for each algorithm of the pool of tools at his disposal. However, both in [4] and [5], such a score is chosen empirically, based on reported results and experiments, and we lack from an automatic method to infer this score from the information available to the analyst; furthermore, the score is incorporated within the system in a rather ad-hoc, non structured way. Currently, two interesting questions need to be faced in image forensics: *how to understand which background information can help detection*, by establishing its impact on the performance of forensic tools, and *how to fruitfully exploit* such a background information.

This paper deals with both the questions: in Section II we introduce a case study, that will be used as reference in the rest of the paper; then in Section III we propose a methodology to investigate the impact of background parameters on the performance of forensic tools. As a key contribution, in Section IV, we investigate two methods to endow multi-clue analysis with this background information: the former builds on the framework presented in [4]; while the latter, based on SVMs, is inspired by the idea of Q-stack classifiers [6]. Finally, in Section V we show that a significant improvement on performance can be obtained by taking into account background information.

II. A Reference Case Study

Without limiting the generality of our analysis, the topic addressed in this paper is more easily defined and treated by relying on a case study. We choose the splicing detection task because of its vast

WIFS'2013, November 18-21, 2013, Guangzhou, China.
ISBN 978-1-4673-5593-3

interest in the image forensics community: given a suspect region, the goal is to understand whether the region has been manipulated (e.g., it has been copy-pasted from another image) or not. Since a great deal of images in the world are stored in JPEG format, we consider the task of fusing the information coming from tools for splicing detection in JPEG images. As explained in [1], a significant amount of tools have been proposed for this task, based on the analysis of different traces. With the aim of fusing algorithms that: i) search for complementary traces, and ii) work in different domains (DCT- versus pixel- domain), we select the following 5 tools from the state of the art: the algorithm by Farid et al. based on JPEG-ghosts [8] (termed JPGH from now on); the tool by Bianchi et al [9] and the one by Lin et al. for detecting aligned double JPEG artifacts (JPDQ and LPLC, respectively); the tool described in [10] and the one proposed by Luo in [11] for detecting *non* aligned double JPEG artifacts (JPNA and JPBM, respectively).

During the creation of a splicing, different kinds of traces may be left into the image, depending on the processing steps applied by the forger. When dealing with JPEG splicings, the four kinds of forgery procedures described in Table I cover the vast majority of combination of traces. The rightmost column of the table makes it clear that most kinds of tampering are detected only by a subset of the available tools.

TABLE I
PROCEDURE FOR THE CREATION OF DIFFERENT CLASSES OF TAMPERING.

Class	Procedure	Detected by
Class 1	Region is cut from a JPEG image and pasted, breaking the 8x8 grid, into an uncompressed one; the result is saved as JPEG.	JPNA JPBM
Class 2	Region is taken from an uncompressed image and pasted into a JPEG one; the result is saved as JPEG.	JPGH JPDQ JPLC
Class 3	Region is cut from a JPEG image and pasted into an uncompressed one in a position multiple of the 8x8 grid; result is saved as JPEG.	JPGH
Class 4	Region is cut from a JPEG image and pasted (without respecting the original 8x8 grid) into a JPEG image; the result is saved as JPEG.	JPGH JPDQ JPLC JPNA JPBM

In order to generate a sufficiently large dataset, we collected a total of 630 uncompressed images representing a variety of scenes (indoor, outdoor, people, landscapes, etc.), all cropped to size 1536×1536 pixels. We considered as possible values for the size of the tampering: $\{64 \times 64, 128 \times 128, 256 \times 256, 512 \times 512, 1024 \times 1024\}$ pixels. Each tampering is created by pasting, in the center of the image, a region cut from another version (e.g., uncompressed or compressed differently, see Table I) of the same image. This tampering strategy creates forgeries that are virtually undetectable to the eye. For tampered images we let the quality of the first JPEG compression (Q_1) take values in the set $\{40, 45, \ldots, 100\}$, and the quality of the final compression is chosen as $Q_2 = Q_1 + \delta$, where δ is chosen at random from the set $\{5, 10, 15, 20\}$. Untouched images are compressed only once with $Q = \{65, 70, \ldots, 100\}$. By combining the above settings, from each uncompressed image the following files have been created:

- 40 non-tampered JPEG images, by using all possible values for QF_1, and taking all possible sizes for the suspect (although not tampered) region;
- 40 forged images, by using all of the 5 possible sizes of the tampering and two random coupling for Q_1 and Q_2, thus obtaining 10 images forged according to each different procedure.

The dataset therefore consists of a total of 50400 JPEG images, half of them tampered. Each different class of splicing consists of 25200/4 = 6300 sample images. During the creation of the dataset, we annotated both the average value and the standard deviation of pixels in the suspect region (in the case of a color image, the image is converted to the YCbCr space and the Y channel is considered). The resulting dataset can be downloaded[1], together with the output obtained from the 5 considered tools.

III. Background Parameters and their Influence

The idea behind image forensics is that when a digital image is processed some traces are left in it, that depend both on the kind of processing and on the type of image (e.g., JPEG or uncompressed). By searching for appropriate footprints, many different algorithms have been developed to investigate the processing history of images; for example, some algorithms work directly in the pixel domain, while others work in a transformed domain like the DCT. A common feature of all detectors is that when a footprint becomes "less detectable", forensic algorithms relying on that footprint become less *reliable*, meaning that they do not discriminate well between presence and absence of the trace. Giving a formal definition of the detectability of a generic footprint is beyond the scope of this paper. Besides, the detectability of different footprints is affected by different parameters, and a golden rule seems hard to derive.

These considerations suggest that the reliability of a given tool can be better investigated by using a sound experimental approach, that is, by conveniently testing the tool. To this end, we propose a possible procedure that the analyst may use to validate the reliability of the various tools as a function of a set of measurable parameters, so to establish if they actually impact the performance of the tools.

Suppose we have a set $\mathcal{F}$ of forensic tools whose goal is to tell, for a given image, whether it contains a specific trace of forgery (we denote this hypothesis with $\mathcal{H}_1$) or not ($\mathcal{H}_0$). For the case study defined in Section II, we can write:

$$F = \{\text{JPGH}, \text{JPDQ}, \text{JPLC}, \text{JPNA}, \text{JPBM}\}.$$

For simplicity, we assume that each tool $f \in \mathcal{F}$ outputs a score $s_f(x)$ (that may be, for example, a probability for the presence of the tampering trace the tool is looking for), and decides for $\mathcal{H}_0$ for the images such that $s_f(x) \leq \tau$. In this way, the tool partitions the space of possible images $\mathcal{X}$ in two regions: Λ_0, containing images for which $\mathcal{H}_0$ is accepted, and Λ_1, defined similarly for $\mathcal{H}_1$. According to classical detection theory, the probability of detection and false alarm for the specific tool and a given τ are defined, respectively, as:

$$P_D^f = \int_{\Lambda_1(\tau)} p(x|\mathcal{H}_1)\ \mathrm{d}x \quad \text{and} \quad P_{FA}^f = \int_{\Lambda_1(\tau)} p(x|\mathcal{H}_0)\ \mathrm{d}x,$$

where $p(x|\mathcal{H}_0)$ denotes that the image does not contain the trace and $p(x|\mathcal{H}_1)$ denotes the opposite case.

Now, let us assume that the analyst has access to a vector of independent parameters $p \in \mathcal{P}$, where $\mathcal{P} = \mathcal{P}_1 \times \mathcal{P}_2 \times \cdots \times \mathcal{P}_P$. We are interested in relating the performance of each tool to subsets of $\mathcal{P}$; for simplicity, we restrict one parameter at a time to a subrange of its possible values $\mathcal{R} \subset \mathcal{P}_j$. To do that, we define

$$\mathcal{R}_j = \mathcal{P}_1 \times \ldots \mathcal{P}_{j-1} \times \{\mathcal{P}_j \cap \mathcal{R}\} \times \ldots \mathcal{P}_P. \tag{1}$$

[1] http://clem.dii.unisi.it/~vipp/index.php/download/imagerepository

Practically, $\mathcal{R}_j$ denotes the set of images whose j-th parameter takes value in $\mathcal{R}$. Notice that the assumption of independent parameters was made so to simplify the discussion; the framework can be adapted to account for the presence of dependent parameters by choosing a refined definition for the set $\mathcal{P}$.

Using the above notation, we can write the probability of detection and the probability of false alarm of f when the analysis is restricted to a specific set of images (those for which the parameter j belongs to $\mathcal{R}$):

$$P_D^f(\mathcal{R}_j) = \int_{\Lambda_1(\tau)\cap\mathcal{R}_j} p(x|\mathcal{H}_1)\ \mathrm{d}x, \tag{2}$$

$$P_{FA}^f(\mathcal{R}_j) = \int_{\Lambda_1(\tau)\cap\mathcal{R}_j} p(x|\mathcal{H}_0)\ \mathrm{d}x. \tag{3}$$

Equations (2) and (3) give the probabilities for a given threshold τ. By varying τ, a Receiver Operating Characteristic curve is generated, that is commonly used to evaluate the discrimination capability of a detector. By taking the integral of the ROC, the Area Under Curve (AUC) is obtained and, finally, the Gini coefficient [12], denoted with ρ, can be used to summarize the performance of the tool:

$$\rho = 2 \times \mathrm{AUC} - 1. \tag{4}$$

By varying $\mathcal{R}_j$ in (1), the forensic analyst can investigate whether the performance of a tool change significantly when different subsets of $\mathcal{X}$ are considered.

Let us now apply this approach to the case study described in the previous section. We define a product set of four possibly relevant parameters

$$\mathcal{P} = \mathrm{Q} \times \mathrm{Z} \times \mathrm{A} \times \mathrm{S},$$

defined as follows:

- Q - *compression strength:* lossy coding after the manipulation process discards some information, thus concealing the already vanishing footprints left by the processing steps. Stronger compressions are against the analyst, because they erase the footprints more deeply.
- Z - *size of the analysed region:* most forensic tools rely either on a statistical model or on the extraction and classification of some features. In both cases, working with more data results in a more reliable analysis.
- A - *average value of pixels in the analysed region:* many forensic tools suffer from saturated regions (i.e., having very low or very high luminance values). This holds especially for DCT-based algorithms, where the truncation errors due to saturation introduce anomalies in DCT coefficients.
- S - *standard deviation of pixels in the analysed region:* uniform (i.e., having very low standard deviation) content yields an extremely sparse DCT representation, that can hardly lead to a reliable forensic analysis.

We use the dataset introduced in Section II to investigate the dependency of the performance of tools on each of the above parameters. Figure 1 shows the ROC curves obtained by each tool in $\mathcal{F}$ for different ranges of the parameter Q, along with the value of ρ calculated for each curve. We can definitely state that this parameter strongly influences the performance of tools in $\mathcal{F}$ and, noticeably, some tools are more sensitive than others (compare, for example, the variation of the ρ value for JPGH and JPDQ).

Plotting similar figures for each of the parameters is not possible here, so we summarize with Table II the analysis for other parameters in $\mathcal{P}$. We see that all the parameters affect the performance of the tools and, most noticeably, not all the tools are affected in the same way, like, for instance, the size of the analyzed region (parameter Z), that strongly affects the performance of JPGH and JPBM but does not influence much JPNA. When performing a joint analysis, such an information can greatly help the analyst in reaching a correct global decision.

TABLE II
IMPACT OF PARAMETERS Z, A AND S ON THE PERFORMANCE OF FIVE IMAGE FORENSIC TOOLS. INTERVALS ARE CHOSEN SO TO EMPHASIZE EXTREME VALUES FOR EACH PARAMETER.

Tool	$\mathbf{R}_{\mathrm{Z}}^1$: (0,64]	$\mathbf{R}_{\mathrm{Z}}^2$: (64,128]	$\mathbf{R}_{\mathrm{Z}}^3$: (128,256]	$\mathbf{R}_{\mathrm{Z}}^4$: (256,512]	$\mathbf{R}_{\mathrm{Z}}^5$: (512,1024]
JPGH	0.63	0.67	0.71	0.75	0.80
JPDQ	0.37	0.62	0.72	0.75	0.78
JPLC	040	0.39	0.36	0.31	0.21
JPNA	0.74	0.75	0.74	0.73	0.72
JPBM	0	0.08	0.21	0.31	0.40
	$\mathbf{R}_{\mathrm{A}}^1$: (0,30]	$\mathbf{R}_{\mathrm{A}}^2$: (30,60]	$\mathbf{R}_{\mathrm{A}}^3$: (60,150]	$\mathbf{R}_{\mathrm{A}}^4$: (150,230]	$\mathbf{R}_{\mathrm{A}}^5$: (230,255]
JPGH	0.49	0.68	0.73	0.62	0.20
JPDQ	0.50	0.63	0.70	0.54	0.04
JPLC	0.09	0.35	0.38	0.25	0.19
JPNA	0.58	0.78	0.80	0.60	0.36
JPBM	0.15	0.19	0.23	0.14	-0.23
	$\mathbf{R}_{\mathrm{S}}^1$: (0,5]	$\mathbf{R}_{\mathrm{S}}^2$: (5,10]	$\mathbf{R}_{\mathrm{S}}^3$: (10,15]	$\mathbf{R}_{\mathrm{S}}^4$: (20,40]	$\mathbf{R}_{\mathrm{S}}^5$: (40,60]
JPGH	0.51	0.69	0.70	0.73	0.74
JPDQ	0.31	0.60	0.65	0.71	0.73
JPLC	0.28	0.28	0.34	0.38	0.33
JPNA	0.46	0.65	0.76	0.79	0.80
JPBM	0.07	0.13	0.18	0.21	0.30

IV. EXPLOITING THE BACKGROUND INFORMATION

In this Section we describe two methods to endow multi-clue information fusion for image forensics with background knowledge. The first method is an extension of the framework in [4], while the second one is a modification of a Q-stack SVM, as defined in [6]. We start by giving a brief introduction to Dempster-Shafer Theory of Evidence.

A. Introduction to Dempster-Shafer Theory of Evidence

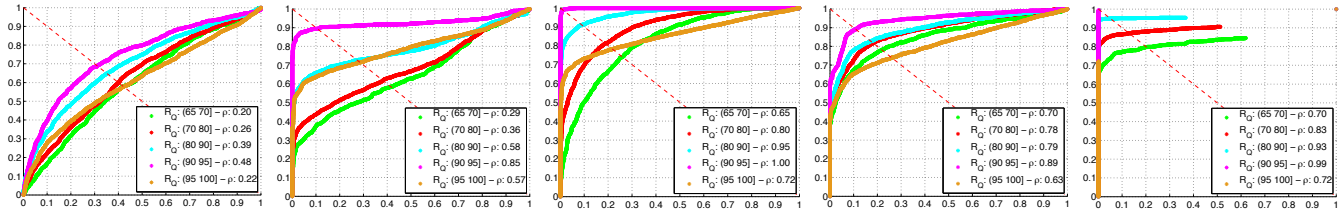
Dempster-Shafer Theory of Evidence (DST) [13] can be seen as a generalization of the Bayesian theory of probability. Let the frame $\Theta_x = \{x_1, x_2, \ldots, x_n\}$ define a finite set of mutually exclusive and exhaustive possible values of a variable x. A Basic Belief Assignment (BBA) is a function $m : 2^\Theta \to [0, 1]$ satisfying:

$$m(\emptyset) = 0; \qquad \sum_{A\subseteq\Theta} m(A) = 1 \tag{5}$$

where 2^Θ is the power set of Θ, that is the set of all possible *propositions* about x. Barely speaking, $m(A)$ denotes the part of belief that supports exactly A but, due to the lack of information, does not support any strict subset of A.

A fundamental element of DST is Dempster's combination rule, that allows to combine several belief functions defined over the same frame, provided that they are obtained from independent sources of evidence.

Definition Let m_1 and m_2 be BBAs over the same frame Θ. Let us also assume that K, defined below, is positive. Then for all non-



(a) JPBM (b) JPLC (c) JPGH (d) JPDQ (e) JPNA

Fig. 1. ROC curves for tools in $\mathcal{F}$ for different ranges of last JPEG compression quality factor: $\mathcal{R}_{\mathrm{Q}}(a,b]$ denotes the set of all images in the dataset whose last compression quality factor falls within $(a,b]$. In each plot, the probability of detection P_D^f is plotted against the probability of false alarm P_{FA}^f.

empty $A \subseteq \Theta$ the function m_{12} defined as:

$$m_{12}(A) = \frac{1}{1-K} \cdot \sum_{\substack{i,j: \\ A_i \cap B_j = A}} m_1(A_i) \cdot m_2(B_j) \tag{6}$$

where $K = \sum_{i,j:A_i \cap B_j = \emptyset} m_1(A_i) m_2(B_j)$, is a BBA function and is called the *orthogonal sum* of m_1 and m_2, denoted by $m_1 \oplus m_2$.

B. Reference Multi-Clue Fusion Framework

A framework that exploits DST for combining the evidence coming from several tools has recently been proposed in [4]; the framework basically follows five steps:

1) outputs from each tool are separately mapped to a BBA about the presence/absence of the searched trace;
2) the analyst provides, if possible, a reliability score for each tool that discounts the certainty of the tool;
3) information from tools searching for the same trace is fused using Dempster's rule;
4) the obtained BBAs about different traces are adapted to a common domain and fused together;
5) compatibility relationships between different traces (modelled through a BBA) are introduced using Dempster's rule.

We are mainly concerned about the first steps of the scheme, where information coming from each tool is converted to a BBA about the presence or absence of a trace. In [4], this mapping is totally delegated to the analyst: in step 1, he uses three mapping functions, that are considered as an input to the fusion framework, to map the scalar output of the tool to a BBA over the frame $\Theta_\alpha = \{t\alpha, n\alpha\}$, where $t\alpha$ is the proposition "trace α is present", $n\alpha$ is the proposition "trace α is not present", and $\{t\alpha \cup n\alpha\}$ is the doubtful proposition "trace α may or may not be present". In step 2, the analyst models the reliability of the tool with a scalar number, obtained from experimental results, and uses it as a weighting parameter to increase the doubt that resulted from the first step. Hence, the task of interpreting background information is also delegated to the analyst. Needless to say, when the number of background parameters increases, such an interpretation becomes very difficult.

C. A DST-based Method to Endow Multi-Clue Analysis with Background Information

We now introduce an alternative way to automatically interpret the output of the tool *and simultaneously account for the background information*, without any need for the analyst to interpret the latter. Since in this preliminary phase of the framework tools are treated separately, in the following we will refer to a single, generic tool.

The problem we are facing with can be formalized as follows: after running the tool on a suspect image searching for a trace α, the analyst obtains a scalar output o. He also extracts the background information, yielding an array $(p_1, \ldots, p_P)$ of parameter values. The analyst wants to map this information to a BBA on the frame Θ_α, that will be fused with those coming from other tools.

In accordance with the hypotheses of [4], let us suppose that the analyst has a training set

$$\mathcal{T} = \{t^i = (o^i, p_1^i, \ldots, p_P^i) : i = 1 \ldots N\} \tag{7}$$

of N training samples, where, for the i-th sample, o^i denotes the output obtained from the tool and p_j^i denotes the value assumed by the j-th background parameter, properly scaled and normalized (see the Appendix of the paper for details). Each training sample belongs to one of the possible classes in $\mathcal{C} = \{C_0, C_1\}$, where C_0 is the class of images containing the searched trace, and C_1 the class of images without the trace.

As opposed to common classification problems, our goal here is not to assign an unseen sample $u = (o^u, p_1^u, \ldots, p_P^u)$ to one class in $\mathcal{C}$. Instead, we want to map it into a basic belief assignment over the frame Θ_α, reflecting the confidence of the tool about the presence of the looked-for trace. The key idea we build upon, that was first introduced in [14], is to model the elements of $\mathcal{T}$ as a source of evidence about u, and use Dempster's rule to pool the evidence. Intuitively, the closer a training sample is to u, the stronger will be the supporting evidence it provides. Formally, let $\mathcal{T}_{u,k} \subset \mathcal{T}$ be the set of k training samples nearest to u according to some distance $d(\cdot,\cdot)$. Then, an element $t_i \in \mathcal{T}_{u,k}$ belonging to class C_0 provides the following BBA over Θ_α:

$$m_i^u(X) = \begin{cases} \beta e^{-\gamma d(u,t_i)} & \text{for } X = \{t\alpha\} \\ 1 - \beta e^{-\gamma d(u,t_i)} & \text{for } X = \{t\alpha \cup n\alpha\} \end{cases}, \tag{8}$$

where $\beta \in (0,1)$ denotes the maximum belief we commit to a single training sample, and γ controls the width of the kernel. On the contrary, an element t_i belonging to class C_1 provides:

$$m_i^u(X) = \begin{cases} \beta e^{-\gamma d(u,t_i)} & \text{for } X = \{n\alpha\} \\ 1 - \beta e^{-\gamma d(u,t_i)} & \text{for } X = \{t\alpha \cup n\alpha\} \end{cases}. \tag{9}$$

As to the distance function, a reasonable choice is

$$d(u, t_i) = ||u - t_i||^2,$$

provided that values are well distributed within a common interval (this issue is addressed in the Appendix of the paper).

Equations (8) and (9) deserve a comment: a sample belonging to class C_0 assigns some evidence to the proposition "u comes from an image containing the searched trace" and the rest of the evidence to the doubtful proposition "the image may or may not contain the trace". The same reasoning applies for samples belonging to class C_1, as in equation (9). Notice that, when the unseen sample u is very far from t_i, this training sample will provide a BBA that is completely doubtful, instead of partitioning the mass between the two

propositions $t\alpha$ and $n\alpha$. On the other hand, such a partitioning may occur after evidence pooling, when some of the k nearest neighbours belong to one class and some to the other, and they are all near to u. This situation means that the unseen sample lays in a "confused" part of the space (i.e., a region where the tool is less reliable): there are training samples near to it, but they belong to different classes.

Once the BBA assigned by each element in $\mathcal{T}_{u,k}$ has been calculated, we can use Demspter's combination rule to pool the evidence, yielding:

$$m^u(X) = \bigoplus_{i=1}^{k} m_i^u(X), \tag{10}$$

where $\oplus$ denotes the application of Dempster's orthogonal sum defined in (6) to all the m_i^u. The pooled BBA in (10) summarizes the belief of the analyst about presence of the trace after observing the output of one tool and the background information. As desired, this certainty is strongly influenced by background parameters: as one or more parameters move towards unfavourable values, samples in the dataset are likely to mix between the two classes, resulting in a less informative pooled BBA. Moreover, the final BBA will be increasingly doubtful as the unseen sample moves in unpopulated parts of the space, where few training samples are available: this perfectly models the fact that the analyst does not know how much the tool can be trusted in such working conditions.

After obtaining $m^u(X)$ in (10) for each of the tools available to the analyst, the framework in [4] can be used to fuse them together and yield a global belief about the authenticity of the image.

D. SVM-based approach

Another solution to endow multi-clue information fusion with reliability parameters is to train classifier that mixes outputs from tools and background information, as proposed by authors of [6] under the name of Q-stack classifier. Given a generic signal that has to be classified and a set of classifiers, the Q-stack framework basically follows two-steps:

1) Each classifier analyzes the signal and produces a *test score*;
2) A set of *quality measures* are extracted from the signal;
3) A second-phase classifier (this motivates the word "stack") is trained that jointly considers test scores and quality measures.

To cast our scenario into this framework, we can consider each forensic tool as a classifier, and let each parameter in $\mathcal{P}$ play the role of a "quality information". Here lays the main conceptual difference between the background information considered in this paper and the one in [6]: parameters in $\mathcal{P}$ do not describe the quality of the analyzed image, they rather describe the capability of the image of carrying the searched trace of processing. This fact further motivates the need for a methodological approach that helps the analyst in selecting the proper background information, like the one we proposed in Section III.

After running each forensic tool and obtaining a set of outputs o_j^i, $j = 1 \dots F$, an *evidence vector* e_i is created, by concatenating tools outputs and background information (symbols are defined as in (7)):

$$e_i = (o_1^i, o_2^i, \dots, o_F^i, p_1^i, p_2^i, \dots, p_P^i),$$

and the training dataset is then defined as $\mathcal{S} = \{s_i : i = 1 \dots N\}$. Also in this case, evidence vectors are normalized to facilitate the classifier (details are in the Appendix). Notice that, differently from $\mathcal{T}$ in (7), each element of $\mathcal{S}$ contains the output from all the tools, since the classifier has to simultaneously perform score fusion and handle the background information. In view of fusing heterogeneous forensic tools, this fact has an important impact on the scalability of the framework; the main problem is not about the "curse of dimensionality" (F and P will usually be small), it is rather about generating a suitable training dataset for such a classifier, allowing it to learn the relationship between traces searched by tools. As it has been shown in [4], the training dataset must contain sample forgeries for each of the possible combination of traces (e.g., the four "classes" listed in Table I), and this number grows exponentially in the number of traces. Not only, each kind of forgery must be represented with a sufficient number of images, so to allow the system to exploit the background information. These facts lead to huge training dataset, that are not easy to generate and manage. A possible strategy to reduce the complexity, that will be addressed in future work, could be to design a hierarchical classifier: in a first stage, the output from each tool is combined with the background information related to it; then, outputs from the first stage are used to train a second classifier, whose goal is to perform fusion. Nevertheless, this second stage classifier would still need to be trained on a dataset spanning all possible combination of forensic traces.

V. Experimental Results

In this section we compare the proposed methods to their equivalent implementations without background information, so to investigate the impact of such information on classification accuracy. To this end, we used the DST-based framework in [4] and a SVM classifier trained only with tool outputs (properly normalized, see the Appendix). Experiments were carried on the dataset described in Section II, and ROC curves and their Gini coefficient (4) were used to compare the detection performance.

In order to turn the output of DST-based frameworks to a binary decision, we used the method suggested in [4]: an image is classified as tampered when the belief for the presence of at least one trace overcomes the belief about the absence of all traces by a factor δ. By varying δ from -1 to 1, a ROC is obtained. We used a grid search to determine the best values for the parameters that tune the proposed BBA mapping method, yielding $k = 4$, $\beta = 0.8$ and $\gamma = 10$. As to the SVM classifiers, we adopted a RBF kernel, whose C and γ parameters have been determined through a grid search in $C \in \{2^{-1}, 2^0, 2^1, \dots 2^{16}\}$, and $\gamma \in \{2^{-4}, 2^{-3}, \dots 2^6\}$, resulting in $\{C = 2^{15}, \gamma = 2^{-4}\}$ for the proposed SVM and $\{C = 2^9, \gamma = 2^1\}$ for its fusion-only counterpart. A model providing posterior probability estimates has been trained [15], allowing us to plot a ROC curve also for this method.

The available dataset was split into two parts, one used for training and the other for testing; this procedure was repeated 10 times to have a statistically significant comparison (uncertainty bars are plotted to account for the variability between different tests). Notice that the dataset we are considering in this paper is even more challenging than the one used in [4], e.g. due to the presence of small differences between the quantization factors used in the first and the second JPEG compression, and to the size of tampered areas, that can be very small.

ROC curves are reported in Figure 2: we can definitely state that the use of background information significantly improve performance, with a +9% gain for the DST-based method and +14% gain for the SVM-based method on their respective counterparts.

As a general comment, as opposed to higher performance the SVM-based method has a weaker scalability, since huge datasets are needed to fuse heterogeneous tools. As to the DST-based method, the BBA mapping scales well because is performed separately for each tool, and fusion rules are given by logical relationships between forensic traces [4]. From the computational complexity point of view,

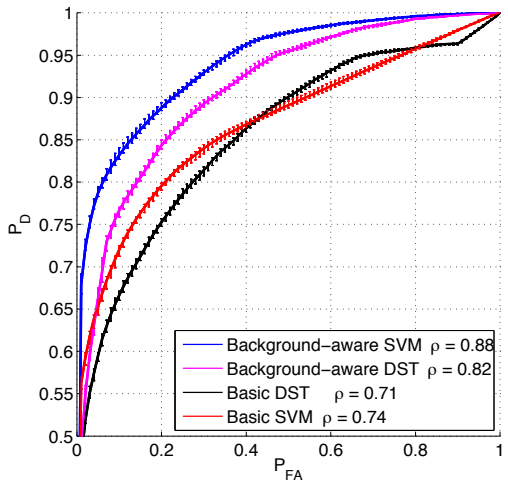


Fig. 2. ROC curves with error bars and averaged Gini coefficients for background information aware methods and their counterparts.

the two methods are quite different: the SVM-based approach needs a heavier training phase to produce the model, then classification is performed at constant time; the DST-based method perform the k-NN search and mass pooling at each query, but this computation can be efficiently implemented to be linear in the size of the training dataset (on a standard desktop computer, the two steps took 0.5 millisecond per image on average). Finally we stress that, in addition to the accuracy gain, both the proposed approaches greatly facilitate the analyst's task compared to those in [4] and [5], since the interpretation of tool outputs and background information is learned automatically, instead of being delegated to the analyst.

VI. Conclusion

In this paper we investigated the use of background information to improve the performance of multi-clue information fusion systems in image forensic scenarios. First, we proposed a way to describe and quantify the influence of background parameters on the performance of forensic tools; then we devised two methods to perform multi-clue analysis while taking these parameters into account. Both background-aware methods obtained a significant and, interestingly, comparable performance gain compared to their "fusion-only" counterpart. Future work will focus on: widening the theoretical perspective of this framework; applying it under different information fusion strategies, and test it when a more heterogeneous set of tools is fused.

Acknowledgment

This work was partially supported by the LIFTGATE Project funded by the FP7-Capacities Programme (grant 285901), by the REWIND Project funded by the FP7-Future and Emerging Technologies (FET) Programme under grant 268478, and by the European Office of Aerospace Research and Development under Grant FA8655-12-1-2138: AMULET - A multi-clue approach to image forensics.

VII. Appendix

Both the DST- and SVM- based approaches take a significant advantage from normalization of feature vectors, so that they are well distributed within a common interval. We choose this interval to be [0,1], and show how we scaled the tool outputs and the considered background parameters. In the following, W denotes the normalized version of $\hat{W}$.

a) Normalization of tool outputs: Among the set of tools described in Section II, those that are based on DCT coefficients analysis tend to produce outputs that, despite being defined in [0,1], are concentrated toward extremes. If we call $\hat{x}_W$ the output of tool W, its normalization x_W is obtained as follows:

- JPLC: $x_L = (\log_{10}(\hat{x}_L)/15) + 1$;
- JPNA: $x_N = \frac{\log_2(\hat{x}_N)}{20\log_2(1.5)} + 1$;
- JPBM: $x_B = \log_{10}(\hat{x}_B)/6 + 1$;

Outputs from tools JPGH and JPDQ are well-distributed in the interval [0,1], so they do not need any processing.

b) Normalization of reliability parameters: Reliability parameters can be very different in nature, so we used different order-preserving functions to normalize them in the interval [0,1].

- Size of the suspect region: denote with X and Y the height and width of the image, then:

$$\mathrm{S} = \frac{\log_2(\sqrt{X * Y}) - 3}{6}.$$

- Compression Quality Factor: $\mathrm{QF} = \hat{\mathrm{QF}}/100$.
- Average pixel value: $\mathrm{AVG} = \hat{\mathrm{AVG}}/255$.
- Standard deviation (STD): for natural images, the standard deviation will unlikely assume values higher than 100. Therefore, the scaled parameter is obtained as: $\mathrm{STD} = \hat{\mathrm{STD}}/100$.

References

[1] A. Piva, "An overview on image forensics," *ISRN Signal Processing*, vol. 2013, 2013.

[2] G. Chetty, J. Goodwin, and M. Singh, "Digital image tamper detection based on multimodal fusion of residue features," in *Advanced Concepts for Intelligent Vision Systems*, 2010, vol. 6475, pp. 79–87.

[3] Y.-F. Hsu and S.-F. Chang, "Statistical fusion of multiple cues for image tampering detection," in *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, oct. 2008, pp. 1386 –1390.

[4] M. Fontani, T. Bianchi, A. De Rosa, A. Piva, and M. Barni, "A framework for decision fusion in image forensics based on Dempster-Shafer Theory of Evidence," *Information Forensics and Security, IEEE Transactions on*, vol. 8, no. 4, pp. 593–607, 2013.

[5] M. Barni and A. Costanzo, "A fuzzy approach to deal with uncertainty in image forensics," *Signal Processing: Image Communication*, vol. 27, no. 9, pp. 998 – 1010, 2012.

[6] K. Kryszczuk and A. Drygajlo, "*Q-stack*: Uni- and multimodal classifier stacking with quality measures," in *Proc. of the 7th International Workshop on Multiple Classifier Systems, MCS*, 2007, pp. 367–376.

[7] E. Argones-Rua, J. Alba-Castro, and C. Garcia-Mateo, "On the use of quality measures in face and speaker identity verification based on video and audio streams," *Signal Processing, IET*, vol. 3, no. 4, pp. 301–309, 2009.

[8] H. Farid, "Exposing digital forgeries from JPEG ghosts," *IEEE T. on Information Forensics and Security*, vol. 4, no. 1, pp. 154–160, 2009.

[9] T. Bianchi, A. De Rosa, and A. Piva, "Improved DCT coefficient analysis for forgery localization in JPEG images," in *ICASSP*. IEEE, 2011, pp. 2444–2447.

[10] T. Bianchi and A. Piva, "Detection of non-aligned double JPEG compression with estimation of primary compression parameters," in *Image Processing (ICIP), 2011*, sept. 2011, pp. 1929 –1932.

[11] W. Luo, Z. Qu, J. Huang, and G. Qiu, "A novel method for detecting cropped and recompressed image block," in *Proc. of ICASSP 2007*, vol. 2, Apr 2007, pp. II–217 –II–220.

[12] L. Ceriani and P. Verme, "The origins of the Gini index: extracts from Variabilità e Mutabilità (1912) by Corrado Gini," *Journal of Economic Inequality*, vol. 10, no. 3, pp. 421–443, September 2012.

[13] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *Annals of Mathematical Statistics*, vol. 38, pp. 325–339, 1967.

[14] T. Denoeux, "A k-nearest neighbor classification rule based on dempster-shafer theory," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 25, no. 5, pp. 804–813, 1995.

[15] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.