A Dempster-Shafer Framework for Decision Fusion in Image Forensics

M. Fontani *#[1], T. Bianchi *†[2], A. De Rosa *[3], A. Piva *[4], M. Barni #[5]

Dept. of Information Engineering, University of Siena
[1] marco.fontani@unisi.it [5] barni@dii.unisi.it

† *National Inter-University Consortium for Telecommunications, Via S. Marta 3, 50139 Firenze, Italy*
[2] tiziano.bianchi@unifi.it

* *Dept. of Electronics and Telecommunications, University of Florence*
[3] alessia.derosa@unifi.it [4] alessandro.piva@unifi.it

***Abstract*—In this work a decision fusion strategy for image forensics is presented, based on Dempster-Shafer's Theory of Evidence. The goal is to automatically summarize the information provided by several image forensics tools, allowing both a binary and a soft interpretation of the global output produced. The proposed strategy is easily extendable to an arbitrary number of tools, it does not require that the output of the various tools be probabilistic and it takes into account available information about tools reliability. Comparison with logical disjunction- and SVM-based fusion shows an improvement in classification accuracy.**

I. Introduction

In the last years many algorithms for detecting photographic tampering have been proposed. In particular, several schemes have been proposed to detect traces left by different kinds of tampering (see, for instance, [1], [2] and [3]). However, in most cases, tampering is obtained by applying a small set of processing tools, hence only a part of the available trace detectors will reveal the presence of tampering. Furthermore, it may happen that the positive answer of one algorithm inherently implies the negative answer of another because they search for mutually excluding traces. Finally, trace detectors often give uncertain if not wrong answers since their performance are far from ideal. For these reasons, taking a final decision about the authenticity of an image relying on the output of a set of forensic tools, is not a trivial task. This problem can be addressed in different ways as illustrated, for the steganalysis problem, in [4]. According to [4], there are basically three kinds of approaches to fusion. The first is to perform fusion at the *feature* level: each tool extracts some features from the data, then a subset of these feature is selected and used to train a global classifier. The second is to consider the output of the tools (usually a scalar) as they are and fuse them (*measurement* level). The last approach consists in fusing the output of the tools after they have been thresholded (*abstract* level).

Most of the existing works are based on the first approach [5] [6] [7]; an hybrid approach has been investigated in [8], but still focusing on feature fusion. A problem with fusion at the feature level is the difficulty of handling cases involving a large number of features (curse of dimensionality) and the difficulty to define a general framework, since ad-hoc solutions are needed for different cases.

In order to get around the above problems, we chose to perform fusion at the measurement level. In fact, this choice delegates the responsibility of selecting features and training classifiers (or other decision methods) to each single tool, thus keeping the fusion framework more general and easy to extend while avoiding to lose important information about tool response confidences, as would happen when fusing at the *abstract* level. Specifically, we present a fusion framework based on the Dempster-Shafer's "Theory of evidence" (DS Theory) [9] that focuses exclusively on fusion at the measurement level. The proposed framework exploits knowledge about tool performances and about compatibility between various tool responses, and can be easily extended when new tools become available. It allows both a "soft" and a binary (tampered/non-tampered) interpretation of the fusion result, and can help in analyzing images for which taking a decision is critical due to conflicting data. Note that a fusion approach involving DS Theory has already been proposed in [10], however such a scheme applies fusion at the feature level hence inheriting the general drawbacks of feature-level fusion, noticeably the lack of scalability and the need to retrain the whole system each time a new tool is added.

This paper is organized as follows: the Dempster-Shafer's framework is briefly introduced in sec. II; the proposed model is presented in sec. III, and its application to three well known tools ([1], [2] and [3]) along with experimental results are presented in sec. IV.

II. Dempster-Shafer framework

Dempster-Shafer's theory of evidence [9] is a framework for reasoning under uncertainty that allows the representation of ignorance and of available information in a more flexible way with respect to Bayes theory. When using classical probability theory for finding the probability of a certain event A, the additivity rule must be satisfied; so by saying that $Pr(A) = p_A$ one implicitly says that $Pr(\bar{A}) = 1 - p_A$, thus committing information about the probability of event

WIFS'2011, November, 16-19, 2011, Foz do Iguaçu, Brazil.
978-1-4577-1019-3/11/$26.00 (c) 2011 IEEE.

A to its complementary $\bar{A}$. Another consequence of the rule of additivity regards the representation of ignorance: complete ignorance about a dichotomic event A can be represented only by setting $Pr(A) = Pr(\bar{A}) = 0.5$ (according to Laplace's principle of insufficient reasoning), but this probability distribution would also be used to model perfect knowledge about probability of each event being 0.5 (as for a fair coin tossing). Furthermore, reasoning in the Bayesian framework often urges to apply insufficient reasoning to assign a-priori probabilities, thus introducing extraneous assumptions. DS theory, instead, abandons the classical probability frame and allows to reason without a-priori probabilities through a new formalism.

A. Shafer's formalism

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. We are interested in quantifying the belief for propositions of the form "the true value of x is in H", where $H \subseteq \Theta_x$ (so the set of all possible propositions is the power set of Θ_x, 2^{Θ_x}). Each proposition is mapped onto a single subset and is assigned a basic belief *mass* through a Basic Belief Assignment.

Definition Let Θ be a frame. A function $m : 2^{\Theta} \to [0, 1]$ is called a Basic Belief Assignment (BBA) if:

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

where the summation is taken over every possible subset A of Θ. Each set S such that $m(S) > 0$ is called a *focal element* for m. Thus $m(A)$ is the part of belief that supports exactly A but, due to the lack of further information, does not support any strict subset of A. Intuitively, if we want to obtain the total belief that a given BBA commits to A, we must add the mass of all proper subsets of A plus the mass of A itself, thus obtaining the *Belief* for the proposition A.

Definition A function $Bel : 2^{\Theta} \to [0, 1]$ is a belief function over Θ if:

$$Bel(A) = \sum_{B \subseteq A} m(B)$$

$Bel(A)$ summarizes all our reasons to believe in A. Relationships and interpretations of $m(A)$, $Bel(A)$ and other functions derived from these are well explored in [11]. Here we just notice that $Bel(A) + Bel(\bar{A}) \leq 1 \ \forall A \subseteq \Theta$ and $1 - (Bel(A) + Bel(\bar{A}))$ is the lack of information about A.

B. Combination Rule

We are interested in using the DS framework to perform data fusion. Dempster defined a *combination rule* that allows to combine several belief functions defined over the same frame.

Definition Let Bel_1 and Bel_2 be belief functions over the same frame Θ with BBAs m_1 and m_2. Let us also assume that K, defined below, is positive. Then for all non-empty $A \subseteq \Theta$ the function m_{12} defined as:

$$m_{12}(A) = \frac{1}{1-K} \cdot \sum_{\substack{i,j: \\ A_i \cap B_j = A}} m_1(A_i) m_2(B_j) \tag{2}$$

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 Bel_1 and Bel_2, denoted by $Bel_1 \oplus Bel_2$.

This rule has many properties [11], in this work we are mainly interested in its associativity and commutativity. Note that K is a measure of the *conflict* between m_1 and m_2: the higher the K, the higher the conflict.

In definition (2) it is assumed that the two BBAs, m_1 and m_2, are defined over the same frame. Whenever we need to combine BBAs that are defined on different domains, we have to redefine them on the same target frame: this can be done using *marginalization* and *vacuous extension*. Let us call domain D the set of variables on which evidence is defined, and let denote a BBA on domain D with m^D.

Definition Let m^{D_1} be a BBA function defined on a domain D_1, then its vacuous extension to $D_1 \cup D_2$, denoted with $m^{D_1 \uparrow (D_1 \cup D_2)}$, is defined as:

$$m^{D_1 \uparrow (D_1 \cup D_2)}(C) = \begin{cases} m^{D_1}(A) & \text{if } C = A \times \Theta_{D_2}, A \subseteq \Theta_{D_1} \\ 0 & \text{otherwise} \end{cases}$$

This allows to extend the frame of a BBA function without introducing extraneous assumptions (no new information is provided about variables that are not in D_1). The inverse operation of vacuous extension is marginalization.

Definition Let m^D be a BBA function defined on a domain D, its marginalization to the domain $D_0 \subseteq D$, denoted with $m^{D \downarrow D_0}$, is defined as

$$m^{D \downarrow D_0}(A) = \sum_{B \downarrow A} m^D(B)$$

where the index of the summation denotes all sets $B \subseteq \Theta_D$ such that the configurations in B reduce to those in $A \subseteq \Theta_{D_0}$ by the elimination of variables in D that are not also in D_0.

III. DST-BASED DATA FUSION IN IMAGE FORENSICS

In this section we present our framework for combining evidence coming from two or more tamper detection algorithms.

A. Assumptions

We consider a case in which we want to investigate the integrity of a known suspect region of an image. We assume that two or more tools are available that, given the suspect region, look for specific tampering traces whose presence reveals tampering. In some cases, we may know that two tools search for mutually-exclusive traces (so that if the first tool reveals a trace, the second one should not find its own); in some other cases, tools search for compatible traces. A single tool can never tell if the image is definitely unmodified: it can only indicate whether the image contains the trace it has looked for or not. We assume that each tool outputs a number

in [0,1], where values near 1 indicate a high confidence about the analyzed region being tampered; we also assume to have some information (possibly image dependent) about tools reliability (for instance such an information could derive from experimental evidence).

B. Formalization for one tool

For sake of clarity, we start by formalizing the proposed framework for one tool only, let us call it *ToolA*, which returns a value $A \in [0,1]$ and has a reliability $R \in [0,1]$. We first consider the information coming from the detection value by introducing a variable T_a, with frame: $\Theta_{T_a} = \{ta, na\}$, where *ta* is the event "image has undergone a tampering detectable using *ToolA*" and *na* is the event "image has not undergone a tampering detectable using *ToolA*". Information provided by *ToolA* can then be summarized with the following BBA over the frame Θ_{T_a}:

$$m_A^{T_a}(X) = \begin{cases} A_T & \text{for } X = \quad \{(ta)\} \\ A_N & \text{for } X = \quad \{(na)\} \\ A_{TN} & \text{for } X = \{(ta) \cup (na)\} \end{cases} \tag{3}$$

We see that this BBA assigns a mass to every element of the power set of Θ_{T_a}; $\{(ta) \cup (na)\}$ is the doubt that *ToolA* has about the presence of the trace, so it refers to the proposition "image has or has not undergone a tampering detectable using *ToolA*". The way A is mapped into A_T, A_N and A_{TN} is an *interpretation* of *ToolA* response and is used to model knowledge about tool behavior (see fig. 1 for an example).

We have assumed that the reliability of *ToolA* is R (R can optionally depend on the specific image the tool is working on). This information can be formalized introducing a new variable R_a, with frame: $\Theta_{R_a} = \{ra, ua\}$ where *ra* is the event "*ToolA* is reliable" and *ua* is the event "*ToolA* in not reliable". In our framework we summarize the reliability information by using a BBA that has only two focal elements:

$$m_A^{R_a}(X) = \begin{cases} A_R & \text{for } X = \quad \{(ra)\} \\ 1 - A_R & \text{for } X = \quad \{(ua)\} \end{cases}$$

This BBA does not assign a mass to doubt: this means that in our framework knowing that a tool is not reliable and ignoring whether it is reliable or not are considered in the same way. Consequently, the most intuitive mapping from R to this BBA assignment is to choose $A_R = R$.

Being defined on different frames, $m_A^{T_a}$ and $m_A^{R_a}$ cannot be combined directly. We need to extend them to a common domain: the simplest one is $T_a \times R_a$. We use vacuous extension to find $m_A^{R_a \uparrow (T_a \times R_a)}$ while, for extending $m_A^{T_a}$ to $m_A^{T_a \times R_a}$, we use a different approach, to give a specific interpretation of what tool reliability should mean: we assume that if the tool is unreliable, its detection should not be considered. This can be easily expressed by putting all elements representing propositions in which the tool is not reliable (i.e. all $(\cdot, ua)$ elements) in every focal element of the combined BBA:

$$m_A^{T_a \times R_a}(X) = \begin{cases} A_T & \text{for } X = \quad \{(ta, ra) \cup (ta, ua) \cup (na, ua)\} \\ A_N & \text{for } X = \quad \{(na, ra) \cup (ta, ua) \cup (na, ua)\} \\ A_{TN} & \text{for } X = \quad \{(ta, ra) \cup (na, ra) \cup (ta, ua) \cup \\ & \qquad \cup (na, ua)\} \end{cases}$$

Now, using (2) we can combine reliability and detection BBAs to yield m_A, which summarizes all the knowledge we have about *ToolA*:

$$m_A(X) = \begin{cases} A_R \cdot A_T & \text{for } X = \quad \{(ta, ra)\} \\ A_R \cdot A_N & \text{for } X = \quad \{(na, ra)\} \\ A_R \cdot A_{TN} & \text{for } X = \{(ta, ra) \cup (na, ra)\} \\ 1 - A_R & \text{for } X = \{(ta, ua) \cup (na, ua)\} \end{cases}$$

C. Introducing new tools

Suppose we want to introduce in our framework a new tool *ToolB*, that satisfies the assumptions in III-A. The same formalism used in III-B will lead us to write m_B, a BBA that summarizes the knowledge for this new tool, defined over the frame $\Theta_{T_b} \times \Theta_{R_b}$. Because we cannot combine m_A and m_B unless they are defined on the same frame, we choose the following strategy: first marginalize both the BBAs eliminating reliability variables; then redefine $m_A^{T_a}$ and $m_B^{T_b}$ on the new domain $T_a \times T_b$ using vacuous extension; finally use Dempster's rule to combine these two BBAs, yielding m_{AB}:

$$m_{AB}(X) = \begin{cases} A_R \cdot A_T \cdot B_R \cdot B_T & \text{for } X = \{(ta, tb)\} \\ A_R \cdot A_T \cdot B_R \cdot B_N & \text{for } X = \{(ta, nb)\} \\ A_R \cdot A_T \cdot C_B & \text{for } X = \{(ta, tb) \cup (ta, nb)\} \\ A_R \cdot A_N \cdot B_R \cdot B_T & \text{for } X = \{(na, tb)\} \\ A_R \cdot A_N \cdot B_R \cdot B_N & \text{for } X = \{(na, nb)\} \\ A_R \cdot A_N \cdot C_B & \text{for } X = \{(na, tb) \cup (na, nb)\} \\ C_A \cdot B_R \cdot B_T & \text{for } X = \{(ta, tb) \cup (na, tb)\} \\ C_A \cdot B_R \cdot B_N & \text{for } X = \{(ta, nb) \cup (na, nb)\} \\ C_A \cdot C_B & \text{for } X = \{(ta, tb) \cup (na, tb) \cup \\ & \qquad \cup (ta, nb) \cup (na, nb)\} \end{cases}$$

where $C_A = (1 - A_R(A_T + A_N))$ and $C_B = (1 - B_R(B_T + B_N))$, *tb* is the proposition "image has undergone a tampering detectable using *ToolB*" and *nb* is the proposition "image has not undergone a tampering detectable using *ToolB*". If another tool *ToolX* becomes available, the associativity of Dempster's rule allows to combine directly its BBA m_X with m_{AB}, so we will always need to extend the domain of only two BBAs: the one coming from the new tool and the one we had for previous tools. This strategy makes this model easily extendable up to an arbitrarily high number of tools.

D. Tool compatibility

By now we have considered tool responses as if they were independent from each other. This allowed us to avoid conflicts between tools, obtaining an easily expandable fusion framework, however, as we noted in III-A, this is not always the case in real applications. Suppose we have three tools (*ToolA*, *ToolB*, *ToolC*) and suppose that ideally only some combinations of their outputs can be expected; for example, it may be that the presence of the trace detectable by *ToolA* implies the absence of the trace detectable by *ToolB* and *ToolC*, so, at least ideally, the three tools should never detect tampering simultaneously. This information can be easily incorporated within the DST model by using a BBA defined on the domain $T_a \times T_b \times T_c$, that has only one focal set, which contains the union of all events that are considered possible,

while all other events have a null mass. For example, if we have 3 tools with compatibilities as in table I, BBA would be:

$$m_\gamma(X) = \begin{cases} 1 \text{ for } X = \{(ta, nb, nc) \cup (na, tb, tc) \cup (na, nb, tc) \cup (ta, tb, tc)\} \\ 0 \text{ for } X = \{(na, tb, nc) \cup (ta, tb, nc) \cup (ta, nb, tc)\} \end{cases}$$

This BBA should be combined, as a last step, with m_{ABC} and, since some events are considered impossible, the presence of conflict should be revealed (represented by K in eq. 2).

E. Final decision

We can now define the final output of the fusion procedure, i.e. we want to know whether a given region of an image has been tampered with or not. To do so we consider the belief of two sets: the first one, T, is the union of all events in which at least one algorithm revealed a tampering, the second one, N, is the single event in which none of the tools detected a tampering (in the previous example it would be $N = (na, nb, nc)$). The output of the fusion process therefore consists of two belief values and a measure of the conflict detected during decision fusion; formally, the output is given by the triplet $\{Bel(T); Bel(N); K\}$ where K is defined in sec. II-B. These outputs summarize the information provided by the available tools, without forcing a final decision. If a binary decision about image authenticity is required, an interpretation of these outputs has to be made; the most intuitive binarization rule is to classify an image as tampered when $Bel(T) > Bel(N)$, but we can also make a simple implementation of the "presumption of innocence" principle by requesting $Bel(T) > Bel(N)+K$. The Receiver Operating Curve can thus be obtained by classifying images according to $Bel(T) > Bel(N) + K + \delta$ and sampling δ in [-1,1]. Notice that we did not need to introduce a-priori probabilities about an image being original or forged: in a Bayesian framework, this would have been harder to obtain.

IV. Experimental results

In order to validate the effectiveness of the proposed approach, we compared it with one of the approach proposed in [8], where image manipulations are detected by taking the logical disjunction (OR) of the outputs of single tools. Logical disjunction is indeed one of the simplest and most widely used methods for decision fusion, and is quite well-suited to the proposed case study[1]. On the other side, several methods have been proposed for decision fusion at feature level in image forensics [5] [6] [7] [10], but they are typically based on feature selection and are therefore not directly comparable to the method proposed in this work. In particular, in [10] DS Theory is employed in a decision fusion framework, but it is used to fuse features instead of tool responses: the actual decision is taken using a SVM, thus requiring an additional training step which hinders one of the main advantages of the proposed method (namely, that each tool can be added without retraining the whole system). Nevertheless, because all cited methods end up using a classifier (usually a SVM) the best we can do for comparing our framework to them without exiting the measurement level is to train a SVM using the output of the single tools as input features, and see how the SVM performs in discriminating between tampered and original images.

[1]Actually, taking the OR of binarized outputs is an "*abstract* level" approach. However, logical disjunction is one of the most used approaches among the post-classification ones [4], so we compare directly to it.

A. Experiment setup

We choose to perform experiments by fusing outputs obtained from three algorithms for tampering detection, namely: the one from Luo et al.[1] (which we will call *ToolA*), the one from Lin et al.[2] (*ToolB*) and the one from Farid [3] (*ToolC*). All of these tools aim to check if a certain region of the image has been substituted with one cropped from another image, before performing a last JPEG re-compression of the resulting image with quality factor QF_2. In particular, *ToolA* checks if the region has been cropped, without preserving JPEG grid alignment, from another JPEG image, that was compressed with quality QF_1; *ToolB* reveals both if the region has been cropped from an uncompressed image or from a JPEG compressed image (quality QF_1) but without preserving grid alignment; *ToolC* checks if the region has been cropped from a JPEG compressed image (quality QF_1) and pasted preserving JPEG grid alignment.

To be compliant with the assumptions in section III-A, each tool has to output a value in [0,1], where values near 1 indicate a high confidence about the analyzed region being tampered. For *ToolA*, this value is obtained using the approach in [12] to get a probabilistic output from the SVM (training is performed on a separated dataset); for *ToolB*, the detection is taken as the median (over the suspected region) of the probability map [2]; for *ToolC*, the value of the KS statistic is directly used [3], exploiting the fact that the DST framework does not require the input values to have a probabilistic meaning.

To build the test dataset, we considered four different tampering procedures that, starting from two images (of whom at least one is in JPEG format) automatically produce a forgery by cut-and-pasting a portion (256x256 pixel) of one image into the other and saving the result in JPEG format. Namely, Class1 forgeries are obtained by recompressing only the pasted region (that is, host image is not JPEG) breaking JPEG grid alignment, according to [1]; images in Class2 are obtained by recompressing only the untouched region (source image is not JPEG) preserving grid alignment according to [2]; Class3 is obtained by recompressing only the pasted region preserving grid alignment, according to [3] and finally images in Class4 are produced by recompressing both the pasted and the untouched regions, breaking JPEG grid alignment only in the former, thus matching requirements in [1] and [2]. The original, non-tampered, images are obtained by applying JPEG compression to uncompressed TIFF images (1024x1024 pixels), choosing randomly the quality factor of the compression from the set {40, 50, ..., 100}. For tampered images, the quality factor of the first compression (QF_1) is chosen in the same way, while the quality of the second compression is set to $QF_2 = QF_1+20$. From the above description and from

experimental evidence gathered by testing each single tools separately we can write the following compatibility table.

TABLE I
DETECTION COMPATIBILITY: EACH ELEMENT OF THE TABLE SPECIFIES WHETHER THE CLASS OF TAMPERING (COLUMN) IS DETECTED (Y) OR NOT (N) BY THE TOOL ON THE LEFT ROW.

Tool	Class 1	Class 2	Class 3	Class 4	Original
ToolA	Y	N	N	Y	N
ToolB	N	Y	N	Y	N
ToolC	N	Y	Y	Y	N

We built a dataset of 1600 images as training set for tuning the model parameters and to obtain a ROC curve for each of the available tools. Among these 1600 images, 800 are kept unmodified and 800 are used to simulate different kinds of tampering (200 images are produced for each of the four defined classes). Tuning model parameters consists in choosing for each tool a reliability value and a mapping from its scalar output to the three values of its BBA, according to eq. (3). Note that there is no need for *cross-tool* training, thus considerably simplifying the tuning process. By relying on our tests and on available knowledge about tools performances, we defined the reliability of the various tools as follows: for *ToolA* we let $R = 0.4 \cdot \mathrm{QF}_2$ (where QF_2 is scaled to [0,1]), for *ToolB* we set $R = 0.4$ and for ToolC $R = 0.85$. The mapping of detection values into BBAs (eq. 3) are reported in the right column of fig. 1. These curves have been chosen by looking at the histogram of each algorithm response both for original and tampered images (left column of figure 1), considering only images that satisfy the working assumptions of the tool (see table I). As shown in figure 1 the only algorithm for which doubt is employed is *ToolB*: this choice has been made considering that *ToolB* tends to return "extreme" detection values (so the central bins in detection histogram are not very populated). It is reasonable to associate some amount of doubt to detection values that falls in a region where few examples were observed. On the contrary, doubt is not employed for *ToolA* and *ToolC* because, although their detection values overlap in the central part of the histogram, they seem to be "equally sure" about those images being tampered or original instead of unsure about both. Experiments confirm that assigning similar value to "tampered" and "untampered" propositions for these "confused" detection values performs better than assigning strong doubt.

Validation of tamper detection tools is usually carried out by relying on Receiver Operating Characteristic (ROC) curves, so we need to train an *aggregate* ROC for the three algorithms, which represents their behavior when combined with the OR operator. The ROC curve for logical disjunction was then obtained as follows: first the ROC of each algorithm is calculated separately, considering only images that satisfy the corresponding working assumptions (see table I); then for each value of false alarm probability (p_{FA}) these ROCs are used to find the threshold that gives that p_{FA} for each algorithm. These threshold triplets are then used to binarize the output of the algorithms on the whole dataset, and the final decision is taken as the logical disjunction of these outputs, thus obtaining a point for the aggregated ROC.

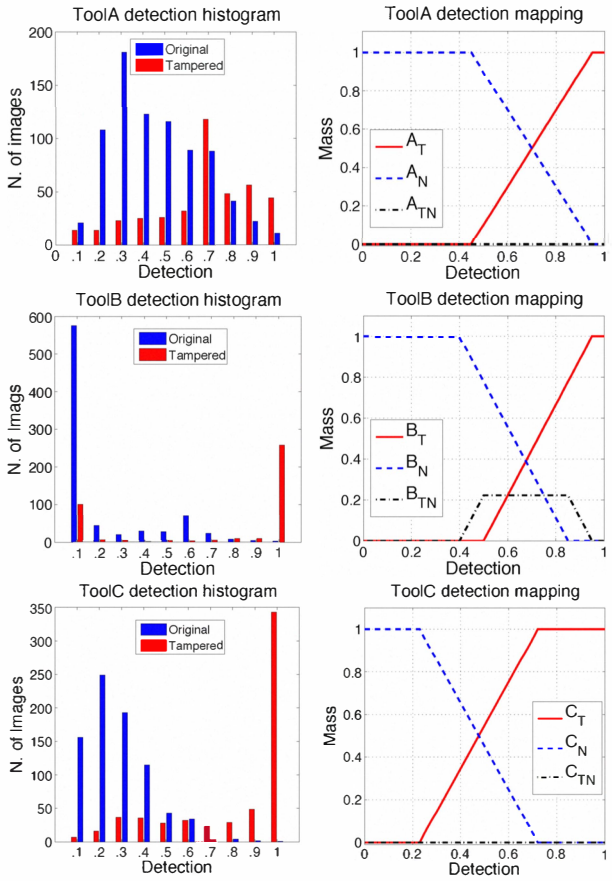

Fig. 1. For each tool, the histogram of detection values on training dataset are shown on the left column. The right column shows how this histogram is interpreted to define a mapping from tool detection value (x-axis) to mass assignment (y-axis). See eq. 3 for an explanation of each line meaning.

SVM-based fusion of tools's outputs is obtained by training the SVM using a RBF kernel with parameters (obtained through a 5-fold c.v.) $\gamma = 2.48$ and $C = 0.1$. To obtain a ROC for this model, we use again the method in [12] to get probabilistic outputs from the SVM (sigmoid parameters are $A = 2.14$, $B = 0.033$), then we threshold this soft output with values sampled from $[0, 1]$.[2] Another approach, which we may consider in future experiments, could be to use the margin distance to get a sort of classification confidence.

B. Results

We performed two different sets of experiments. In the first one, we built a test dataset of 1600 images, generated with exactly the same procedure used to build the training set, but using different images. We have 800 original images (that is, JPEG compressed once with QF in $\{40, 50, \ldots, 100\}$) and 800 tampered images, 200 for each of the four classes. We run the three forensic tools and fuse their outputs on all of these

[2]It should be noted that the SVM is trained to maximize the total accuracy, which maps to a *single* point in a ROC. However, this way of obtaining a ROC from an SVM is extensively used in machine learning literature.

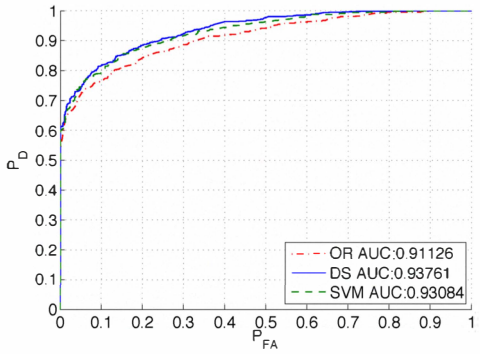

Fig. 2. ROC for logical disjunction (red dashed-dotted line), proposed model (solid blue line) and SVM (dashed green line) on the standard dataset.

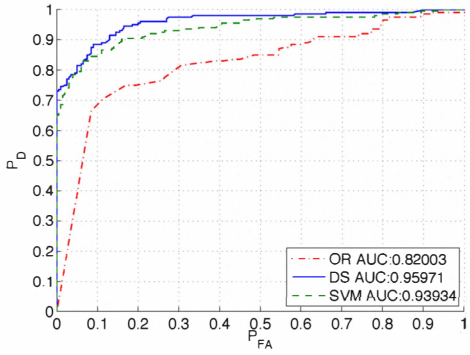

Fig. 3. ROC for logical disjunction (red dashed-dotted line), proposed model (solid blue line) and SVM (dashed green line) on the "critical" dataset.

images, using for each fusion model the parameter set defined during the training phase. In figure 2 we compare the ROC and the Area Under Curve (AUC) for the various methods.

We can see that, although we are binarizing the output of the model, the proposed framework gives better results with respect to both logical disjunction and SVM. The gain is not exciting (approx. +2.6% in AUC on OR, +0.7% on SVM), but it is encouraging, especially because it has been obtained on a dataset that was built in such a way to match the working assumptions of the various tools hence minimizing the presence of uncertain situations for which the DST should provide the greatest advantages.

To better highlight the usefulness of the proposed framework, we performed a second experiment: we generated 200 original and 50×4 forgeries, using only images showing strong textures (e.g. trees or city landscapes) and compressing original images with high quality factors (picked from the set $\{0.85, 0.90, 0.95, 1\}$). We chose these settings because we noticed that *Tool2* produces a large number of false positives on this kind of images, and we want to test how robust the various fusion techniques are in the presence of non-ideal situations. Results obtained on this new dataset are reported in figure 3: in this case DST fusion significantly outperforms logical disjunction (+13.9% in AUC), and increases his advantage also on SVM (+2% in AUC). That is because, when *Tool2* generates a false positive, the output triplet "NYN" is observed, but this triplet, being *not* included in table I, is mapped to the nearest plausible one. This behavior derives from the combination of the compatibility BBA (m_γ, in sec. III-D) with the global mass assignment obtained from tools.

V. Conclusions

In this paper we have addressed a central problem in image forensics, namely the fusion of information stemming from the application of several tamper detection tools. The fusion strategy we have developed is easily extendable to even a large number of tools. Other advantages derive from the adoption of a DST framework, since such a theory permits to cope with situations in which incomplete information is available about the a-priori tampering probabilities. Information about the dependence among the output of the single tools and their reliability can also be easily incorporated within the model. Experimental results are encouraging: the proposed model gives significantly better results than a fusion approach based on logical disjunction, and also outperforms SVM-based fusion (that presents the additional disadvantage of requiring a global training of the final SVM classifier, limiting the scalability of this approach).

Future work will focus on validating the proposed scheme on larger datasets and assess its capacity to handle situations in which the output of more than three tools has to be fused.

Acknowledgment

This work was partially supported by the REWIND and LivingKnowledge Projects, funded by the Future and Emerging Technologies (FET) programme within the 7FP of the EC, respectively under grants 268478 and 231126.

References

[1] 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.

[2] Z. C. Lin, J. F. He, X. Tang, and C. K. Tang, "Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis," *Pattern Recognition*, vol. 42, no. 11, pp. 2492–2501, Nov. 2009.

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

[4] M. Kharrazi, H. T. Sencar, and N. D. Memon, "Improving steganalysis by fusion techniques: A case study with image steganography," *T. Data Hiding and Multimedia Security*, vol. 4300, pp. 123–137, 2006.

[5] Y.-F. Hsu and S.-F. Chang, "Statistical Fusion of Multiple Cues for Image Tampering Detection," in *Asilomar Conference on Signals, Systems, and Computers*, 2008.

[6] G. Chetty and M. Singh, "Nonintrusive image tamper detection based on fuzzy fusion," *IJCSNS*, vol. 10, no. 9, pp. 86–90, Sep 2010.

[7] D. Hu, L. Wang, Y. Zhou, Y. Zhou, X. Jiang, and L. Ma, "D-S Evidence Theory based digital image trustworthiness evaluation model," in *Proc. of MINES 2009*, ser. MINES '09, vol. 1, 2009, pp. 85–89.

[8] S. Bayram, I. Avcibas, B. Sankur, and N. Memon, "Image manipulation detection," *J. Electronic Imaging*, vol. 15, no. 4, 2006.

[9] G. Shafer, *A Mathematical Theory of Evidence*. Princeton: Princeton University Press, 1976.

[10] P. Zhang and X. Kong, "Detecting image tampering using feature fusion," *Proc. of ARES 2009*, vol. 0, pp. 335–340, 2009.

[11] A. Benavoli, L. Chisci, B. Ristic, A. Farina, and A. Graziano, *Reasoning under uncertainty: from Bayesian to Valuation Based Systems*. ISBN: 978-8886658430, 2007.

[12] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in large margin classifiers*. MIT Press, 1999, pp. 61–74.