Signal Processing: *Image Communication* 27 (2012) 998–1010

Contents lists available at SciVerse ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

A fuzzy approach to deal with uncertainty in image forensics

M. Barni [a], A. Costanzo [b,*]

[a] University of Siena, Department of Information Engineering, Via Roma 56, 53100, Siena, Italy
[b] CNIT - Consorzio Nazionale Interuniversitario Telecomunicazioni, Research Unit of Siena, University of Siena - Via Roma 56, 53100, Siena, Italy

ARTICLE INFO

Article history:
Received 20 December 2011
Accepted 20 July 2012
Available online 9 August 2012

Keywords:
Image forensics
Tampering detection
Fuzzy logic
Data fusion

ABSTRACT

Image forensics research has mainly focused on the detection of artifacts introduced by a single processing tool, thus resulting in the development of a large number of specialized algorithms looking for one or more specific footprints under precise settings. As one may guess, the performance of such algorithms are not ideal, so the output they provide may be noisy, inaccurate and only partially true. Moreover, in real scenarios a manipulated image is often the result of the application of several tools made available by the image processing software. As a consequence, reliable tamper detection requires that several tools developed to deal with different scenarios are applied. The above observations raise two new problems: (i) deal with the uncertainty introduced by error-prone tools and (ii) devise a sound strategy to merge the information provided by the different tools into a single output. To overcome these problems we propose a decision fusion framework based on the Fuzzy Theory, which permits to cope with the uncertainty and lack of precise information typical of image forensics, by leveraging on the widely known ability of the Fuzzy Theory to deal with inaccurate and incomplete information. We describe a practical implementation of the proposed framework and validate it in a realistic scenario in which five forensic tools exploit JPEG compression artifacts to detect cut&paste tampering within a specified region of an image. The results are encouraging, and provide a significant advantage with respect to those obtained by simply OR-ing the outputs of the single tools.

1. Introduction

Digital image capture devices, such as digital cameras or mobile phones, have become very common due to their low cost and ease of use. 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 field of research to prove the authenticity and integrity of digital images. A large number of techniques have been developed in the past years to identify the processing that an image has undergone [1,2]. The techniques developed so far have been able to detect a wide variety of manipulations such as single and multiple compression [3,4] and resampling [5,6] or forgeries such as cut&paste [7,8,4] and copy&move [9,10].

Generally each forensic technique deals with the detection of a typical footprint left by a single processing tool under specific settings. Forensic techniques, however, like any other realistic process or system, are never perfect and their measurements are usually affected by uncertainty, ambiguity or impreciseness. A noisy or unreliable response may have many causes, such as: wrong tool settings; particular characteristics of the analyzed images (e.g. color space or type of compression); partial presence (or absence) of the feature(s) the tool is looking for; deviation from the working assumptions of the applied technique.

Another obstacle that one needs to overcome when judging the integrity of a given image is that most of the

* Corresponding author. Tel.: +39 0577 234850x1061; fax: +39 0577 233609.
E-mail addresses: barni@dii.unisi.it (M. Barni), andreacos82@gmail.com (A. Costanzo).

0923-5965/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.image.2012.07.006

times a tampered image is not the result of the application of a single processing tool. On the contrary, even "non-expert users" will resort to several tools provided by any processing software to alter an image. Since rarely the kind of manipulation that the image has undergone is known beforehand, the application of a single footprint detection technique may not be enough, thus requiring the parallel use of more than one technique. A problem with the use of several tools looking for different footprints is that each tool provides an output describing the degree of presence of the specific footprint it is looking for. Even when using more than one tool, we are usually interested in obtaining a single global answer allowing to decide whether the image under analysis is authentic or not. Obtaining such a global answer, however, is not a trivial task: outputs may not only be inaccurate, but also heterogeneous. For example, one tool may provide a binary output, another tool a scalar value to be compared with a threshold, while a third tool may output the probability that the image has undergone a certain processing. Moreover, depending on the input image, forensic tools may have technical limitations, be prone to errors or be in disagreement with each other, thus introducing another form of uncertainty. In these cases, classic techniques such as simple majority vote (an image is tampered if the majority of tools say that the image is tampered) or binary OR (an image is considered to be tampered if at least one tool detects tampering traces) may not provide satisfactory results. Therefore, it is necessary to develop new efficient methods to keep the uncertainty of different outputs under control while merging them into a single final decision.

Following Kharrazi et al. [11], we can consider three main approaches to merge the outputs of several tools: *feature level* fusion, *measurement level* fusion and *abstract level* fusion. The feature level fusion consists in the aggregation of all the features provided by the various tools before actually taking a final decision (e.g. by means of SVM or neural networks). In the measurement level fusion each tool makes a partial decision by relying only on its features and then all the partial detection scores are aggregated into a global score. The abstract level fusion first applies separated thresholds to all partial scores, thus obtaining binary values that are then aggregated into a global value. Fusion methods proposed so far usually focus either on the first or on the second method. We chose to focus on the measurement level because it does not suffer of the high dimensionality typical of many forensic features.

More specifically, we present a solution based on fuzzy logic that permits to handle the uncertainty and impreciseness of forensics tools and merge their outputs when several such tools are available. Fuzzy logic has been used in a very wide range of domains, such as sensor networks, automatic vehicle navigation, industrial and aerospatial applications and databases. The fuzzy-logic approach has demonstrated to be useful in those applications where reasoning needs to be robust against noise, approximate or imprecise inputs [12–14]. For this reason we believe that a system based on fuzzy logic may also help to deal with the incomplete or conflicting outputs provided by different forensic algorithms and to resolve them into a single final value. Moreover, one of the main advantages of fuzzy logic is the capability to address problems whose mathematical or statistical models are hard to define. In this way one may design automated frameworks that resort to the experience and the knowledge of human operators to mimic their behavior. Let us try to imagine how a forensic analyst would face the problems of uncertainty and fusion. First of all, he/she would tweak the tools at his/her disposal by gathering as much information as possible (e.g. which ones are the most trustworthy, on what kind of images they work, how they interact with each other), thus tackling with the uncertainty problem. Then he/she would run all the tools on the image under analysis and exploit the previously gathered knowledge to make a final decision, thus tackling with the fusion problem. It is the goal of this paper to propose a framework based on fuzzy logic that relieves the forensic analyst from these tasks.

To the best of our knowledge, the usage of fuzzy logic to address the problem of uncertainty in image forensics has been very limited in the past. The only technique we are aware of is the one proposed by Chetty et al. in [15]. However, as opposed to our contribution, the system described in [15] relies on fuzzy integrals applied to the features extracted by the forensic algorithms, thus making impossible a direct comparison with our scheme.

We validated the fuzzy framework we have developed by applying it to a scenario in which five forensic tools exploit JPEG compression artifacts to detect cut&paste tampering within a specified region of an image. Results were compared against those obtained through OR-based fusion on three different data sets: a data set composed of synthetic images tailored to match the requirements of the employed tools (thus depicting a limited uncertainty scenario); a data set formed by images thought to trigger the erroneous responses of some tools; a data set consisting of realistic tampered images that have undergone several different processing (thus deviating significantly from the working assumptions of the forensics tools). The experiments confirmed the superiority of the new approach with respect to OR-based fusion approach for all the data sets, with the most significant improvements occurring when dealing with realistically tampered images, where uncertainty and impreciseness are most likely to arise. The proposed framework has several other strengths including: generality since it does not depend on the employed tools; straightforward integration of new tools; automatic generation of fuzzy inference rules.

The rest of this paper is organized as follows: in Section 2 we present an overview of fuzzy logic principles. In Section 3 we propose a formalization of the forensic decision problem and we introduce a general fuzzy framework to address it. In Section 4 we discuss the experimental results. We conclude the paper with Section 5, where we outline some directions for future research.

2. Foundations of fuzzy theory

Fuzzy sets theory was conceived in 1965 by Lotfi Zadeh as an extension of classic set theory [16]. From this initial concept a multi-value fuzzy logic has been derived in subsequent years as an extension of Boolean logic. According

to Zadeh, the main rationale behind fuzzy logic is the observation that despite people do not require precise, numerical information input for their reasoning, they are capable of highly adaptive control. If such capability could somehow be transferred to systems, they would perhaps be more effective and easier to implement. Moreover, he also claimed that "as the complexity of a system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance become almost mutually exclusive characteristics" [13]. Fuzzy logic was designed to deal with imperfect information, which in the real world is more often the norm than the exception. Zadeh defined *computing with words* the methodology of dealing with incomplete, unreliable or partially true information.

In order to understand the way fuzzy logic works, we need to introduce three concepts: fuzzy sets, fuzzy operators and if–then rules. In the following we briefly describe these concepts.

2.1. Fuzzy sets

Let $\mathcal{X}$ be the universe set and $\mathcal{C} \subseteq \mathcal{X}$ be a set contained in $\mathcal{X}$; then $\mathcal{C}$ can be represented by its characteristic function:

$$\mu_{\mathcal{C}}(x) = \begin{cases} 1 & \text{if } x \in \mathcal{C} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

Sets characterized as in (1) are also called crisp sets. Fuzzy set theory extends this concept. A fuzzy set $\mathcal{F} \subseteq \mathcal{X}$ is defined through a generalized characteristic function $\mu_{\mathcal{F}}(x) : \mathcal{X} \rightarrow [0,1]$ (instead of $\{0,1\}$) [16]. The function $\mu_{\mathcal{F}}(x)$ is called *membership function* and associates to each element $x \in \mathcal{X}$ a grade of membership that is a real number in the interval [0,1]. In Zadeh's framework an element x can belong (or not belong) to a given fuzzy set $\mathcal{F}$ with a certain grade of membership. Let, for instance, X be the space of all temperatures. In classical set theory a temperature x is either hot or not hot. In fuzzy theory, x can be at same time hot and non-hot with degrees $\mu_{hot}(x)$ and $\mu_{\overline{hot}}(x)$. A value of $\mu_{hot}(x)$ near 1 indicates a high degree of membership of x in the fuzzy set *hot*, a value near 0 indicates a high degree of membership in $\overline{hot}$.

Similar to classic sets, Zadeh has defined operations like intersection, union and complement to be applied to fuzzy sets [16]. Let A and B be two fuzzy sets and $\mu_A(x)$, $\mu_B(x)$ be their membership functions. The basic set-operators can be generalized as follows:

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)),$$

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)),$$

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x). \tag{2}$$

By relying on the operations defined by Eq. (2), Zadeh also demonstrated the validity of the basic properties of crisp sets operations like commutativity, associativity, distributivity and De Morgan's law.

2.2. Fuzzy operators

If we interpret the values of membership functions as truth values we can extend the concepts of fuzzy sets theory to obtain a multi-valued fuzzy logic [16]. Classic Boolean logic requires that a proposition is either true (1) or false (0). There are no other possible values to assign. Based on real world experience, with fuzzy logic we can affirm that a proposition is not always totally false or totally true but true or false to some grade in the interval [0,1]. Doing so, it is possible to claim that a proposition is true, *more or less* true, *somewhat* true and so on. Since Boolean logic can be seen as a particular case of fuzzy logic where one can only assign values 0 and 1 to membership functions, the extension of logical operators is quite simple and intuitive. In a nutshell, fuzzy AND, OR and NOT can be obtained directly from Eq. (2) by parallelizing these operators respectively to intersection, union and complement as follows: $\mu_{A \wedge B}(x) = \min(\mu_A(x), \mu_B(x))$; $\mu_{A \vee B}(x) = \max(\mu_A(x), \mu_B(x))$; $not(A) = 1 - \mu_A(x)$.

2.3. If–then statements

If–then rules are the basic instructions that permit to define the behavior of a system by means of commands that are easily understandable by a machine. The definition of such rules is a critical step in the process of building a fuzzy control system. It is the aim of if–then rules to establish a linguistic relationship between the description of a situation and an action to be performed. A simple example of this kind of relationship could be the rule *if obstacle is too close then reduce speed to low*. More specifically, if–then rules are based on the fuzzy logic principles outlined above and define how fuzzy sets and logic operators interact with each other by means of membership functions.

Let us now formalize these concepts more rigorously. Let $x_1,..,x_n$ and $y_1,..,y_m$ be fuzzy variables (i.e. variables that can assume as value the label of a fuzzy set) and let $A_1,..,A_n$ and $B_1,..,B_m$ be fuzzy sets. An if–then rule can be defined as follows:

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_1 \text{ AND } x_2 \text{ is } A_2 \text{ AND} \ldots \text{AND } x_n \text{ is } A_n \\ &\text{THEN } y_1 \text{ is } B_1 \text{ AND } y_2 \text{ is } B_2 \text{ AND } \ldots \text{AND } y_m \text{ is } B_m \end{aligned} \tag{3}$$

The first part of the rule (introduced by IF) is called antecedent or premise; the second part (introduced by THEN) is called consequent or conclusion; the rule itself is called implication. While the structure of the antecedent is quite standard, a consequent can be defined in different ways [14,17]. The consequent form used in Eq. (3) is called Mamdani's model [17] and represents the most common methodology in fuzzy applications due to its simplicity. It is also possible to construct compound rules by means of (nested) conditional structures such as "if–then–else". For example, a compound rule could be: *if* x_1 is A_1 *then* (*if* x_2 is A_2 *then* y_1 is B_1 *else* y_2 is B_2). Such structures can always be decomposed in a set of basic if–then rules as in Eq. (3) [18]. The number of expressions composing a rule is arbitrary. However, expressions belonging to the antecedent and to the consequent are combined separately. In this paper we will work with compound rules that can be reduced to basic Mamdani's

rules with an arbitrary number of terms for the antecedent and a single term for the consequent (i.e. $m=1$).

Regardless of the model, in most cases the adoption of one rule only is not effective: there is need of a set of two or more rules that can play off one another, so that a system can react correctly to a large number of situations.

2.4. Fuzzy inference systems

We are now ready to see how the basic concepts described so far can be used to build a fuzzy inference system. Simply put, a fuzzy system is nothing else but a set of fuzzy rules that converts inputs to outputs. More specifically, a fuzzy system receives input variables that are crisp numbers (e.g. a measure of temperature) that need to be turned into something fuzzy. This task is performed by means of fuzzy sets. Once input values are transformed into fuzzy entities, they are combined accordingly to if–then rules. Result is something fuzzy and usually needs to be turned again into something crisp. With these ideas in mind, the interpretation of a set of if–then rules as in Eq. (3) may be carried out in four steps:

(1) *Fuzzification of input:* This is the process in which the crisp quantities are converted to fuzzy sets. A degree of membership is assigned to each input according to the membership functions of fuzzy sets.
(2) *Application of fuzzy operators to multiple antecedents:* This is the core of the whole process (the actual *reasoning* part): all the degrees of membership obtained from fuzzification are combined with the rules of behavior. Specifically, if the antecedent consists of more than one term, the fuzzy logic operators defined in Eq. (2) are applied to resolve the antecedent into a single value called degree of support of the rule.
(3) *Application of implication method:* The degree of support is used to shape the output fuzzy set. The consequent of a rule, in fact, assigns to the output an entire fuzzy set that is truncated according to the degree of support of the rule. Usually a fuzzy system features several rules, each of which contributes with its own truncated output set. However, to make a decision one needs to look at a single output fuzzy set, thus requiring some kind of aggregation procedure. The most common method of aggregation consists of the `max` criterion.
(4) *Defuzzification of the output:* The result of the previous step is a fuzzy set that a system typically cannot directly use to make a final decision. Therefore, a process of conversion from fuzzy quantities to a crisp global value is required. This process is called defuzzification and can be performed in several different ways [19], the most common of which is the `centroid` method (also referred as center of gravity or center or area).

3. A fuzzy inference system for image forensics

In this section we first provide a formalization of the ideas expressed in Section 1 by the light of the fuzzy logic concepts describe in Section 2 and then describe a practical system implementing them.

3.1. Image forensic tools

Let $\mathcal{T}$ be a set of K image forensic tools for detecting whether a certain region within an image I is tampered or not. Each tool $t_i \in \mathcal{T}$ analyzes a feature set in the specified region looking for tampering traces and generates an output saying whether the trace is present or not. At the end of this process we have K outputs. If we want to answer the question "*has the selected region been tampered with?*" we need a method to deal with the uncertainty affecting the K outputs and merge them into a single value. Based on this value, we will then take a final decision on the authenticity of the region. Amongst the simplest methods to achieve the goal outlined above, we mention majority decision and logical disjunction. In the first case a region is considered to be tampered with if more than half of the tools tells that a tampering has occurred; in the second case, a region is considered to be tampered with if at least one tool says that a tampering has occurred. There is an additional step to be made, since the output of the tools is not binary, before applying an OR-fusion or a majority vote it is necessary that the output is thresholded. As the number of adopted tools increases several problems may arise, thus making these classical decision methods ineffective. Let us consider some examples. Two or more tools could be mutually exclusive: if one finds traces of tampering then the other(s) will not find anything. In this case a decision method based on majority may not work as intended. Moreover, tools are usually not perfect. Practical implementation of a forensic algorithm can be a delicate process: from tuning of parameters to choice of training data set, many factors can affect the final performance. This may result in a tool that is prone to errors. Let us imagine a case where we have $K-1$ tools that work perfectly and one that is really bad. It may happen that most of the times this tool claims that the image region has been tampered with, thus inducing a simple logic disjunction operator to error. For these reasons we need to devise an alternative reliable method to cope with multiple noisy inputs. In the sequel we describe our proposal to deal with multiple, uncertain forensics measurements by relying on fuzzy logic.

3.1.1. Tool outputs

In order to apply the concepts introduced in Section 2 we need to so-to-say *standardize* the output of the forensic tools. In the sequel we assume that all the tools share the same output format, consisting of a pair of values (D,R). $D \in [0,1]$ is the degree of *detection*, that is a measure of the presence of the tampering trace the forensic tool is looking for within the analyzed image region. Values near 1 indicate a strong presence of the trace. $R \in [0,1]$ is the *reliability* of D, i.e. a measure of the confidence the tool has on the detection value. Values near 1 indicate a high confidence. D does not necessarily need to be a probability and generally changes from image to image. R can either be a constant value depending only on the overall performance of the tool or depend on the characteristics of the analyzed region (i.e. size, color, visual content). In general, in order to define the reliability of a forensic tool, we need some information about its performance, drawn either from theory or experimental analysis.

3.1.2. Definition of ideal tools tables

Now that we have established a common output for each tool, we need to inform the system about their expected behavior, including their mutual relationship. Let us suppose that a region of an image I has undergone a tampering. The question we ask is: "*If all the tools at our disposal work as intended, what kind of output do we expect from them*?". Depending on the nature of the manipulation, a tool may or may not be able to identify a region as tampered. Let us indicate the capability of detecting the tampering with Y and the incapability with N. Therefore, if we have K tools, each kind of manipulation (or absence of manipulation) is identified by one or more K-dimensional sequences of Y and N, each specifying the expected behavior of the tools under *ideal* conditions.

In the following we will use the symbols T_{true} and T_{false} to indicate the tables whose columns correspond to the expected (*standard*) answer of the tools in the presence and absence of tampering, respectively; we will use the symbol T_{doubt} to refer to the table of unexpected (*non-standard*) K-uples of tools' outputs belonging neither to T_{true} nor to T_{false}. Since the definition of these tables depends on the tools and is based on the knowledge of their performances, in the following we assume that they are always available.

To exemplify the above concepts, let us consider a simple case in which only two tools are available. Let us assume that these tools, t_1 and t_2, can detect traces of aligned and misaligned double JPEG compression respectively. Let us assume that t_1 (t_2) considers a region with aligned (misaligned) double compression as tampered. The ideal tables corresponding to this simple scenario are shown in Table 1, whose columns are explained in the following. If we run the tools on a region to which an aligned double compression was applied, we expect a (Y,N) answer (first column); if the region has undergone misaligned double compression we expect a (N,Y) answer (second column); if the region has not been tampered with we expect a (N,N) answer (third column); if we obtain a (Y,Y) answer (fourth column) we are in the presence of a noisy, unreliable, maybe partially true answer.

3.2. Fuzzy fusion of tools' outputs

The pairs (D,R) provided by the forensic tools are the input fuzzy variables of the proposed inference system. Several ideas contributed to the definition of the shape of the proposed system, the most important of which are the following:

1. *Noisy input:* Often in a real scenario a tool is not perfectly secure about the presence (absence) of a manipulation. Therefore it may output a value of D that is high (res. low) but not necessarily near 1 (res. 0).
2. *Unreliable input:* A tool may not be confident in its analysis thus providing a low value of R. We may be tempted to discard this unreliable answer. However, this may result in a loss of information that could be still used somehow.
3. *Ease of generalization:* Usually a fuzzy system benefits of any additional information one can provide: the more knowledge we put into the system the better. In our case this additional information corresponds to new tools, whose integration within the system should be as simple as possible.
4. *Absence of mathematical model:* It is not always possible to fully describe a scenario by means of a mathematical model [17]. This is the case of our forensic application, in which fuzzy logic can successfully translate heuristic rules stated by a human operator into a valid fusion strategy.

Table 1
Tables of ideal interactions between the two tools of the example introduced in Section 3.1.2. The first (resp. second) column of T_{true} corresponds to the tampering obtained by pasting a region whose JPEG grid is aligned (resp. misaligned) with respect to the one of the target image.

Tool	T_{true}		T_{false}	T_{doubt}
t_1	Y	N	N	Y
t_2	N	Y	N	Y

As we said, the system we propose attempts to meet these requirements by reasoning on K input pairs (D,R) produced by the forensic tools. The construction of the inference system starts by building a set of fuzzy if–then rules based on T_{true} and T_{false} tables. After that, sequences in T_{doubt} are mapped into standard cases and another set of if–then rules is built accordingly. Rules obtained in this way are then applied to the outputs produced by the forensics tools producing a number that needs to be compared with a threshold to obtain a final answer on image region authenticity. In the sequel we will give a detailed description of each of these steps. Meanwhile, Fig. 1 provides a high-level scheme of the proposed approach.

3.2.1. Definition of fuzzy sets

For the sake of clarity in this section we give an intuitive general description of the chosen fuzzy sets and the construction of the fuzzy inference rules. There are various methods to define the appropriate membership functions, either based on human understanding of the underlying problems or on specific algorithms or logical procedures. Given our knowledge of the employed tools, in this work we chose to define membership functions by means of the first method. Since the shape of membership functions depends on the application at hand and is the key for making fuzzy set theory practically useful, we refer to Section 4 for a more in-depth technical explanation.[1] Intuitively, detection and reliability values can either be considered `low` or `high`, where by low and high we mean fuzzy sets characterized by a membership function. Similarly, the presence of tampering derived from pairs (D,R) of all tools can have different degrees of intensity. In our implementation we have chosen five fuzzy sets for the presence of tampering: `very weak`, `weak`, `neither weak nor strong`, `strong` and `very strong`. In the following we describe how these sets are used to automatically generate the if–then rules.

[1] The soundness of the proposed approach does not depend on the particular shape of the membership functions.

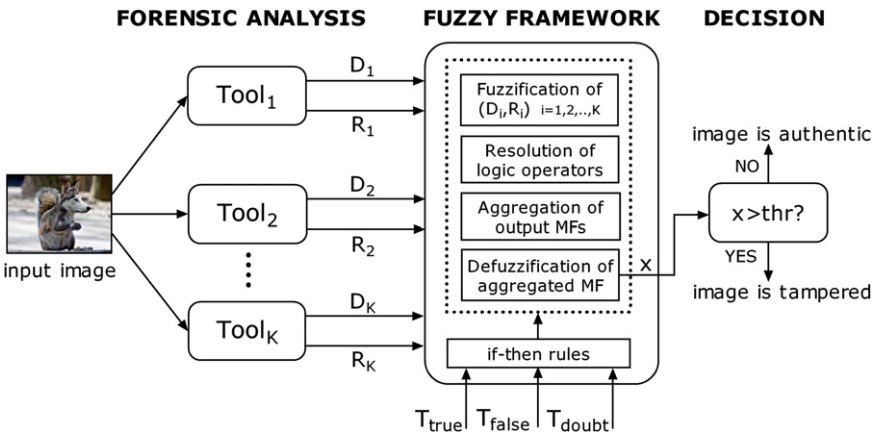

Fig. 1. Workflow of the proposed approach. K forensic tools analyze an input image providing K (D,R) pairs. Two sets of if–then rules are built according to T_{true}, T_{false} and T_{doubt}. By relying on steps 1–4 defined in Section 2.1, a crisp value x of tampering presence is computed and compared against a threshold *thr* (e.g. 0.5) to finally decide on image authenticity.

3.2.2. Automatic construction of standard rules

As dictated by the Fuzzy Theory paradigm, we begin by describing with perceptual and linguistic terms the goal we would like to achieve. Then we try to translate the linguistic description into a set of rules emulating our understanding of the problem. If we were to decide on tampering presence, we would behave as follows: *if the input values of detection correspond to the tampering case we are analyzing and the tools are reliable enough, then we fully trust the joint indications of the tools. On the contrary, if the tools are not reliable enough, then we are still willing to trust them, but only to a lesser extent.*

Let us begin with the standard cases of tampering described by the columns of T_{true} and T_{false}. In the following we define the relationship between detection D and the expected presence (Y) or absence (N) of tampering. Generally no tool is either wholly capable or incapable of detecting a certain tampering. More often, a tool is capable or incapable of detecting a tampering to some degree, depending on the characteristics of the analyzed image. Specifically, we consider a tool capable (incapable) of detecting a manipulation if it provides a high (low) value of detection as follows:

$$\begin{aligned} &\mathrm{Y} = \text{detection is } \textit{high} \\ &\mathrm{N} = \text{detection is } \textit{low}. \end{aligned} \tag{4}$$

In fuzzy terms, D will be a fuzzy variable and `high` and `low` fuzzy sets. As for reliability R, we follow a similar reasoning: a trustworthy tool will have a strong membership in the `high` fuzzy set, a less trustworthy tool a strong membership in the `low` fuzzy set.

The extension of this concept to K tools is immediate. In Section 3.1 we saw that columns of T_{true} and T_{false} are K-dimensional arrays whose elements are either Y or N. If $\boldsymbol{s}$ is one of these arrays, we just need to substitute each element of $\boldsymbol{s}$ with the corresponding expression in Eq. (4) and put all the resulting expressions in AND relationship. Given the values $D_1,..,D_K$ and a standard column **s**, the final expression describes to which degree the inputs belong to the tampering case **s**. For example, in a four-tool scenario, a case **s**=(Y,Y,N,N) will become: D_1 `high` $\wedge$ D_2 `high` $\wedge$ D_3 `low` $\wedge$ D_4 `low`.

The trustworthiness of a tool (hence R) impacts the nature of the consequent. If we fully trust a tool, then we assign to the output the most intense fuzzy set (`very strong` or `very weak` if $\boldsymbol{s}$ belongs to T_{true} or to T_{false} respectively). If we do not trust enough the tool, then we opt for the less intense fuzzy set (`strong` or `weak`).

For the sake of clarity we now detail the automatic construction of the rules with the help of the example introduced in Section 2. Once the procedure is clear we can easily generalize it to an arbitrary number of tools. Let us then consider two tools and the case $(\mathrm{Y,N}) \in T_{true}$ describing the expected interaction in the presence of a cut&paste tampering with aligned JPEG grids. The resulting fuzzy rule is formalized as follows:

$$\begin{aligned} &\text{IF} \quad (D_1 \ \texttt{high} \wedge D_2 \ \texttt{low}) \\ &\text{THEN} \quad [\text{IF } (R_1 \ \texttt{high} \wedge R_2 \ \texttt{high}) \text{ THEN tampering is} \\ &\qquad \texttt{very strong} \\ &\qquad \text{ELSE tampering is } \texttt{strong}] \end{aligned} \tag{5}$$

Although correct, the rule in Eq. (5) is not expressed in the form presented in Section 2.3, however we can easily reduce it to a standard form by starting from the expression inside square brackets, which can be decomposed in two contributes [18] as follows:

$$\begin{aligned} &\text{IF } (D_1 \ \texttt{high} \wedge D_2 \ \texttt{low}) \\ &\quad \text{THEN } [\text{IF } (R_1 \ \texttt{high} \wedge R_2 \ \texttt{high}) \text{ THEN tampering is} \\ &\quad \texttt{very strong}] \end{aligned} \tag{6a}$$

$$\begin{aligned} &\text{IF } (D_1 \ \texttt{high} \wedge D_2 \ \texttt{low}) \\ &\quad \text{THEN } [\text{IF } (\overline{R_1 \ \texttt{high} \wedge R_2 \ \texttt{high}}) \text{ THEN tampering} \\ &\quad \texttt{is strong}] \end{aligned} \tag{6b}$$

These two new compound rules can be further written [18] as:

$$\begin{aligned} &\text{IF } (D_1 \ \texttt{high} \wedge D_2 \ \texttt{low}) \wedge (R_1 \texttt{high} \wedge R_2 \ \texttt{high}) \\ &\quad \text{THEN tampering is } \texttt{very strong} \end{aligned} \tag{7a}$$

$$\begin{aligned} &\text{IF } (D_1 \ \texttt{high} \wedge D_2 \ \texttt{low}) \wedge (\overline{R_1 \ \texttt{high} \wedge R_2 \ \texttt{high}}) \\ &\quad \text{THEN tampering is } \texttt{strong} \end{aligned} \tag{7b}$$

Now both rules are in the form used in Eq. (3). The first rule tells us that, given an image and a standard case (that is (Y,N) in our example), if D_1 and D_2 have a high grade of membership and both the tools are reliable, then we assign to the consequent the most intense level of tampering. The second rule tells us that if one of the tools (or both) is not reliable (recall that according to De Morgan identity: $\overline{R_1\ \texttt{high} \wedge R_2\ \texttt{high}} = \overline{R_1\ \texttt{high}} \vee \overline{R_2\ \texttt{high}}$), then we assign to the consequent a less intense level of tampering. In this way we can generate the set of if–then rules without any intervention from the analyst since the system will automatically translate the interaction tables into $\{Y,N\}$ sequences.

The extension of the above reasoning to a general case of K tools is quite easy. The K tools will produce rules characterized by the same compound structure of Eq. (5) that can be reduced to a set of standard rules by following the same steps of Eqs. (6) and (7). The main difference between our example and its generalization consists of how the fuzzy sets for the consequent are chosen. In the general case, in fact, we employed a majority criterion to link reliabilities to the intensity of output tampering. Therefore, if more than half of the tools are reliable enough, then we assign the most intense fuzzy set to the consequent, else we assign the less intense set. Note that there could be other ways to achieve the same goal, such as to consider the subset of the most reliable tools. However, as the number of employed tools grows, two problems may arise: (i) to define trustworthy subsets that are valid in general for all the images of a data set would not be a trivial task; (ii) to trust only a subset of tools would go against the idea of fusion of all contributions including those affected by uncertainty. Consequently, for the current version of the system we have adopted the simplest solution, that is the majority voting.

3.2.3. Automatic construction of non standard rules

The approach to the construction of if–then rules for *non-standard* cases belonging to T_{doubt} is similar to that of standard cases. However, when a non-standard case occurs we do not have a support from theory or experiments, therefore we need some further reasoning. This is then our new goal: *first, reduce the non-standard case back to something we know* (*i.e. a standard case*); *then adopt a construction similar to that used for the standard cases*.

The first task is carried out by means of a mapping strategy that takes into account the reliability of the various tools. Let $\boldsymbol{ns}$ be a non-standard sequence belonging to T_{doubt} and $\boldsymbol{s}$ a standard sequence belonging either to T_{true} or T_{false}. Let us create a binary sequence by assigning values 0 and 1 to N and Y respectively. We evaluate the distance between $\boldsymbol{ns}$ and $\boldsymbol{s}$ by means of the following weighted Hamming distance:

$$d(\boldsymbol{ns},\boldsymbol{s}) = \sum_{i=1}^{K} R_i \cdot \mathrm{XOR}[\boldsymbol{ns}(i),\boldsymbol{s}(i)] \tag{8}$$

where: K is the number of tools, R_i is the reliability of the i-th tool, XOR is the bitwise exclusive-OR, $\boldsymbol{ns}(i)$ and $\boldsymbol{s}(i)$ are the i-th bits of $\boldsymbol{ns}$ and $\boldsymbol{s}$ respectively. With Eq. (8) we compute the distance of $\boldsymbol{ns}$ from all the M standard sequences and select the closest one as follows:

$$\boldsymbol{s}_{min} = \arg \min_{n=1,2\ldots M}[d(\boldsymbol{ns},\boldsymbol{s}_n)]. \tag{9}$$

Note that Eq. (9) does not define a fuzzy inference rule, since the mapping is performed before actually building the fuzzy inference system.

Since the mapping is an approximation based on experimental parameters, it is not wise to lean towards the presence or absence of tampering as much as we did in Eq. (5). For this reason we choose to employ only the less intense fuzzy set available for the consequent (`strong` or `weak`) regardless of reliability.

Once again we illustrate the behavior of the fuzzy inference system, by relying on the simplified example introduced in Section 3.1.

This time we want to evaluate the case $\boldsymbol{ns}$ = (Y,Y) where both t_1 and t_2 claim that the analyzed region has been tampered with. Since we picked up these tools so that they should be mutually exclusive, this is a doubtful case. Let us suppose that the standard case at distance d_{min} is $\boldsymbol{s}$ = (Y,N), that is a case indicating the presence of tampering. The resulting if–then rule will be:

IF (D_1 `high` $\wedge$ D_2 `high`)

THEN [*regardless of reliabilities* tampering is `strong`] (10)

It is worth noting that in Eq. 10 we chose not to take again into account the role of reliability, since it has already been exploited during the mapping procedure. In fact, the mapping already defined the consequent, which has now just one level of intensity. Therefore, we just need to use the detection values to determine which of the unknown cases we are addressing.

Although rare, it is possible that two or more tools are equally reliable, thus generating more than just one $\boldsymbol{s}$ at distance d_{min} from the non-standard case $\boldsymbol{ns}$. In this situation we act as follows: if all sequences $\boldsymbol{s}$ at distance d_{min} belong to T_{true} or T_{false} then we choose the first $\boldsymbol{s}$ of the set and we proceed exactly as described in Eq. (10); if $\boldsymbol{ns}$ is equally close to at least one $\boldsymbol{s}$ belonging to T_{true} and one to T_{false} we apply to the consequent only the fuzzy set `neither weak nor strong` regardless of reliabilities.

3.2.4. Some remarks on the number of if–then rules

Before we move on to the experimental analysis, we pause to consider the number of if–then rules the inference system consists of. The number of rules we need to consider increases exponentially with the number of variables (that is the dimension of the input space). When *many* tools are employed this may become a problem that should not be underestimated. As a matter of fact, the exponential explosion of the number of rules in fuzzy inference systems, often referred as "curse of dimensionality", is a well known weakness of all fuzzy systems. An excessive number of rules can be the cause of several serious drawbacks, such as: the difficulty of giving a meaningful linguistic description of the scenario; the loss of generality, transparency and effectiveness; the increase of computational burden required to control the system. Although the scientific literature has carefully analyzed this problem proposing a number of

methods to tackle with it (e.g. by means of hierarchical approaches) [20,21], it is not the aim of this work to deal with this situation. When needed, our framework could be enriched by introducing hierarchical rules clustering or by adopting an approach like the one we used to deal with non-standard rules (8) to reduce the number of rules composing the system.

4. Experimental analysis

In this section we describe a practical implementation of the general approach described so far. Our goal is to validate our ideas on a realistic image forensic scenario. We first briefly introduce the tools that we employed, then we describe how we collected the data sets of images used to evaluate the accuracy of the proposed system. Finally, we compare the experimental results with those obtained by a method that simply takes the OR of thresholded tools' outputs. In this work we chose not to compare our method with other soft decision methods, e.g. based on Bayesian theory or on learning (SVM, NN). The first one, in fact, requires information that are not easily acquired (i.e. the a priory probabilities) and lacks of generality, while the latter one requires a very fine parameter tuning and a computationally heavy training.

4.1. Description of forensic tools

In our experiments we investigated the presence of particular artifacts introduced by multiple JPEG compressions within a specified region of an image. Such artifacts, in fact, can be exploited to find out whether the image has been tampered with or not. The basic idea is the following. A cut&paste tampering is commonly carried out by taking a region R from a source image S and pasting it into a target image T, thus creating a fake image F. Let us suppose that either the source or the target images (or both) are JPEG compressed and that the fake image F is saved in JPEG format after the manipulation.[2] The superimposition of multiple JPEG compression steps, either with aligned or misaligned 8×8 grids, characterized by different quality factors, typically brings into the fake image a number of inconsistencies that can be taken as an evidence of tampering.

To detect these artifacts introduced by JPEG compression we used a set of $K=5$ forensics tools. These tools rely on the methods proposed by Luo et al. [7], Lin et al. [8], Farid [4] and Bianchi et al. [22,23] respectively. All of them can be used to detect cut&paste manipulations. In the following we will refer to them as t_A, t_B, t_C, t_D and t_E. In a nutshell they work as follows:

(A) t_A determines whether a region has been cropped from a JPEG image with quality QF_1 and pasted without preserving grid alignment on a second image that afterwards is JPEG compressed with quality factor $QF_2 > QF_1$. Detection relies on a statistical analysis of image blockiness. When used to detect cut&paste tampering, regions where JPEG grids are not aligned are considered as tampered [7].

(B) t_B determines whether a region has been cropped from a JPEG image or from an uncompressed image and pasted on a JPEG target image without preserving grid alignment. This result is achieved by assuming that as a consequence of the cut&paste operation the destination image is compressed twice, such a double compression is detected with a study of the so called Double Quantization (DQ) effect. Since such an effect cannot be revealed in areas where the two compression grids were not aligned, regions wherein DQ effect is not revealed are considered as tampered with [8].

(C) t_C determines whether a region has been cropped from a JPEG image and pasted while preserving compression grid alignment. This result is achieved by analyzing the so called JPEG ghosts, i.e. coefficients previously compressed with a higher quantization step. Regions where JPEG ghosts are revealed are considered tampered [4].

(D) t_D detects the presence of non-aligned double JPEG compression by relying on a single feature which depends on the integer periodicity of the DCT coefficients. Intuitively, the method evaluates how a subset of the DCT coefficients (the DC coefficients, on which the quantization effects are more evident) cluster around a given lattice for any possible JPEG grid shift. This measure is compared with a threshold to decide whether grids are aligned or not [22].

(E) t_E is a direct improvement of t_B and discriminates between original and forged regions in JPEG images, under the hypothesis that the former are compressed twice while the latter just once. Such a task is performed by relying on two specific probability models for the DCT coefficients of regions that are JPEG compressed once and twice. This method provides better discriminating performance in comparison with t_B, especially when $QF_2 < QF_1$ [23].

For a more in-depth description of these techniques the reader is referred to the respective papers. In Section 3.1.1 we stated that each tool has to provide a detection value in [0,1]. For t_A such value is obtained by applying a probabilistic SVM classifier as described in [24]; for t_B and t_E the detection values correspond to the median of the probability map of the analyzed region [8,23]; for t_C the detection value is equal to the KS statistics used in [4]; for t_D the detection value is a simple normalization in [0,1] of the un-thresholded statistic proposed in [22].

4.2. Interactions between forensic tools

The next step consists of the construction of T_{true} and T_{false} tables. According to the principles underlying the employed tools and according to a preliminary experimental analysis we identified four classes of tampered images for which the tools ideally provide different 5-uples of answers. By relying on such an analysis we built

[2] Given the wide use of JPEG format to store images, this seems to be a reasonable assumption.

Table 2
Expected interactions between the 5 tools. Depending on the type of tampering a tool may detect (Y) or not detect (N) the manipulation that has occurred. First 4 columns correspond to T_{true}, fifth column to T_{false}. Combinations not listed here belong to T_{doubt}.

Tool	Class 1	Class 2	Class 3	Class 4	Class 5
t_A	Y	Y	N	N	N
t_B	N	Y	N	Y	N
t_C	N	Y	Y	Y	N
t_D	Y	Y	N	N	N
t_E	N	Y	N	Y	N

Table 2, from which the T_{true} and T_{false} tables can be immediately derived. For a detailed description of the four classes of tampering appearing in the table we refer to the next section and in particular to Table 3. For the sake of brevity we omit the 5-uples belonging to T_{doubt}, which can be easily derived from Table 2.

Note that the two pairs of tools (t_A,t_D) and (t_B,t_E) work in the same operative conditions (in particular, t_E has been developed to overcome some limitations of t_B). For this reason we expect their answers always to be in agreement with each other.

4.3. Construction of image data sets

To validate our approach we have built three different data sets consisting of original and tampered images: (i) a large data set of 1600 images for the general evaluation of the proposed method; (ii) a smaller data set of 400 textured natural images to better highlight the benefits brought by our method in a particular case where one of the tools shows an erroneous behavior; (iii) a data set of 60 images simulating real-world tampering.

The first two synthetic data sets share the same generation procedure. Starting from a set of images, we created four classes of images that simulate a cut&paste tampering. Each class has been designed so that at least one tool (or a pair of tools) is able to detect the presence of the manipulation.[3] We performed each tampering by means of slight changes to the cut&paste procedure as shown in Table 3. More specifically, all manipulations have been conducted by substituting the 256×256 central block of the image. In order to bypass most of the technical limitations of the algorithms we decided to apply the following criteria: $QF_1 \in \{55,60,65,70\}$ and $QF_2 - QF_1 = 20$. Tests have been conducted on the 256×256 central area of each image.

For the first data set we have collected 100 uncompressed TIFF images with different visual content (landscapes, people, macros). Each original image has been used to create two tampered images, thus leading to 200 images for each class and a total of 800 tampered images. Finally we added 800 non-tampered images that have been simply compressed once.

[3] For example, the tampering belonging to "Class 1" in Table 3 consists of double JPEG compression with misaligned grids, therefore only t_A and t_D are supposed to detect it.

The second data set derives from the observation of a peculiar behavior of t_B. This tool, in fact, tends to claim as tampered a specific type of natural images, those containing textures and regular geometric edges (e.g. buildings, walls, squares), compressed once with a very high quality factor. The other tools do not show any particular behavior on this specific category of images. The erroneous behavior of t_B on this class of images generates doubtful cases that cause an OR-based method to provide wrong results, therefore we expect our fuzzy system to perform better. It is important to point out that these images are very common in real-world scenarios, thus making plausible our experiment. To build this second data set, we have gathered 50 natural images whose central regions contain textures and regular edges, compressed once with native camera quality factor $QF_1 = 100$. We created 200 tampered images and 200 original images with the same procedure of Table 3. Again, both tampering and testing have been conducted on the 256×256 central area. A few examples of such images are shown in Fig. 2(a)–(b).

The third data set originated from a simple consideration: rarely in real-world scenarios a tampering is obtained by playing around only with JPEG compression on well defined square regions. A "typical image user" will usually resort to several tools provided by some image editing software in the attempt to cut&paste regions of irregular shape and variable size. After that he/she will likely spend some time to correct inconsistencies of color, size and region edges. Finally, most of the times the partial/final result will be saved in JPEG format.

We then created a set of images of convincing visual quality by using several popular processing tools. As for the subjects of the manipulations, we have chosen images containing frontal faces. It is very common, in fact, to stumble on manipulated faces on the Web due to the meaningful message they are able to convey (e.g. satirical or political). To this aim we gathered 30 original images and we created 30 fakes by substituting the original faces. Tampering was performed by means of Adobe Photoshop®. A variety of processing tools have been used including: geometrical manipulations (scaling, rotation, horizontal flip); color manipulations (brightness and contrast correction); enhancement of pasted region's borders (by means of tools such as *lasso*, *magic wand* and *healing brushes*). In case of multiple JPEG compressions, all these processing steps tend to attenuate or eliminate the JPEG artifacts of the oldest compression step. To avoid the complete loss of such traces, we paid attention to quality factors before and after the processing. Since Photoshop® typically defines JPEG quality with linguistic terms rather than with numerical values, we chose to use *medium quality* for QF_1 and *maximum quality* for QF_2 (recall that all tools perform better when $QF_1 < QF_2$). The experiments have been conducted on the bounding boxes containing the faces. Fig. 2(c) provides an example of an original image and 2(d) its tampered counterpart.

4.4. Experimental settings

For the sake of reproducibility, in this section we describe the parameters of the fuzzy inference system and the values we assigned to them.

Table 3
Tampering classes. Each class has been created by varying the number of compression steps with aligned or non-aligned grids. The fifth class corresponds to non-tampered images.

Class	Tampering procedure
Class 1	Outer region is compressed once. Inner region is compressed twice with misaligned grids
Class 2	Outer region is compressed twice with aligned grids. Inner region is compressed twice with misaligned grids
Class 3	Outer region is compressed once. Inner region is compressed twice with aligned grids
Class 4	Outer region is compressed twice with aligned grids. Inner region is compressed once
Class 5	Non-tampered images. The image is compressed once with a random but fairly high quality factor: $QF \in \{70,75,80,90\}$

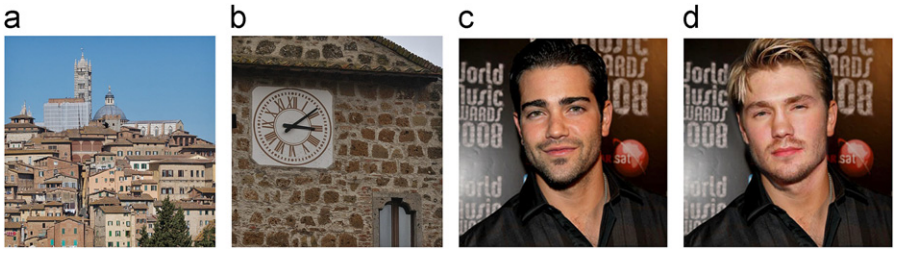

Fig. 2. Examples of the images used for the experiments. (a)–(b): second data set; (c)–(d): third data set. In particular, (c) is and original image while (d) is a tampered image obtained by pasting a new face into (c).

4.4.1. Fuzzy parameters, detection and reliability

One of the advantages of our system is that the set of fuzzy parameters is rather small. We chose the Mamdani's model for the if–then rules. We implemented the AND operator with the `min` function while combining the terms of antecedents. We aggregated if–then rules by means of `max` function. Finally we performed defuzzification by means of the `centroid` method. The system features 10 inputs ($D_{A,B,C,D,E}$ and $R_{A,B,C,D,E}$) and one output (*tampering*).

In order to calculate D, instead of considering only the detection value of each tool in the specified region, we performed two separate analysis: one on the region itself (D_{inner}) and one on the rest of the image (D_{outer}). We then used $D=|D_{outer}-D_{inner}|$ as an input of the fuzzy fusion system. We think that this approach is, in fact, more robust to false positives. Given a tool, if no tampering has occurred we expect similar values of detection for the inner and outer regions. Therefore the difference between these two values should be small (ideally 0). On the other hand, if the region has been tampered with we expect very dissimilar values. The difference between the two values, then, should be large enough (ideally 1) to make the system lean towards a correct revelation of tampering. It goes without saying that, due to the uncertainty usually affecting forensics tools, often we will work with differences D that are quite distant from their ideal values.

While defining R, we noticed that t_A, t_D and t_E are more reliable when the second JPEG quality factor QF_2 is high. For this reason we decided to linearly increase such reliabilities depending on QF_2. The coefficients of the linear transformation have been derived from the curves of accuracy as a function of QF_2 published by the authors of the tools [7,22,23]. R_A ranges from a minimum of 0.73 when $QF_2=60$ to a maximum of 0.96 when $QF_2=100$; R_D ranges from 0.65 to 1.0; R_E ranges from 0.659 to 0.91. The rest of reliabilities corresponding to other values of QF_2 are obtained by means of interpolation. Reliabilities of t_B and t_C do not seem to be affected by QF_2, therefore, following a previous experimental testing on separated data sets, we assigned to them constant values: $R_B=0.40$ and $R_C=0.85$. In Section 4.7 we discuss the robustness of the fuzzy system to variations of the values assigned to reliabilities.

Note that, in this particular case - in which all tools exploit JPEG artifacts - we could have used directly the pairs (D,QF_2) as input to the fuzzy fusion framework rather than the pairs (D,R) by defining custom membership functions of QF_2. However this approach, although correct, would not have been general and would have ceased to work, for example, if we added a tool exploiting features that are not related to JPEG. Therefore, in order to preserve the generality of the framework we chose to delegate to each tool the definition of a suitable reliability.

4.4.2. Membership functions and if–then rules

We used two different families of membership functions for both inputs and outputs: piecewise (trapezoidal) and smooth (combination of gaussians). Both types are fairly easy to implement and can be built automatically. Fig. 3(a) shows that each input can belong to two fuzzy sets: `low` and `high`. The point where the two functions cross is where we measure maximum fuzziness (namely the *cross-over point*), since an input value is characterized by the same degree of membership to both classes ($\mu_{low}=\mu_{high}=0.5$). Values to the left of this point have a higher degree of membership in the fuzzy set `low` ($\mu_{low}>\mu_{high}$); values to the right of this point have a higher grade of membership in the fuzzy set `high` ($\mu_{low}<\mu_{high}$). For an explanation of how we chose the points of maximum fuzziness we refer to Section 4.6. Fig. 3(b) shows membership functions for the output variable representing intensity of tampering. We have defined five possible fuzzy sets accordingly to Section 3.2.1. From left to right `very weak`, `weak`, `neither weak nor strong`, `strong` and `very strong`.

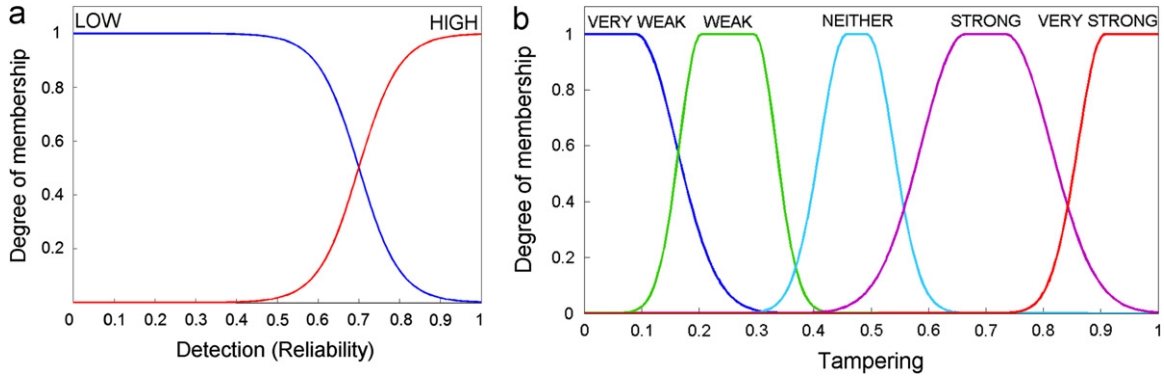

Fig. 3. Smooth membership functions for system variables: (a) input detection depending on variable point p of max fuzziness (e.g. $p=0.7$). Input reliability uses membership functions with the same shape but with fixed $p=0.5$; (b) output.

4.5. Performance evaluation

Our experiments started with a separate analysis of the forensic tools at our disposal. To this aim we created a dedicated data set of 1600 images with the same procedure of Table 3 but starting from a different set of original images. The reason we used a new data set rather than the one described in Section 4.3 is the following. At this point we are still evaluating the performance of each tool separately. This step is meant to provide us with the parameters necessary to set up our aggregated evaluation method. If we tune such parameters on a data set that we use again to evaluate the overall results, we obviously benefit of an unfair advantage.[4]

We calculated the Receiver Operating Curve (ROC) of each tool only on those subsets of the data set that satisfy the assumptions the tool relies on: t_A and t_D on classes 1 and 2; t_B and t_E on classes 2 and 4; t_C on classes 2,3 and 4.

At this point we have to evaluate the ROC curves obtainable by using the fuzzy and OR-based fusion methods. In order to do so, we proceeded as follows:

(1) *Sampling the ROC of each tool:* We first sampled the probability of false alarm P_{fa} of each curve with a step of 10^{-3}. Then, for each value of P_{fa} we obtained its corresponding probability of correct detection P_d. We used the pair (P_{fa},P_d) to calculate the threshold $\in [0,1]$ that needs to be applied to the detection values provided by the various tools in order to obtain those probabilities. We repeated the previous procedure for all curves, thus obtaining five thresholds for each value of P_{fa}. We organized these thresholds in 5-uples.

(2) *Aggregated ROC of the OR method:* The values of each 5-uple have been simply used as binary thresholds. For example, the first value of the 5-uple was used to threshold D_1 (the output of tool t_A), the second to threshold D_2 and so on. We decided on the authenticity of the analyzed region by OR-ing the five binary values.

(3) *ROC of fuzzy fusion:* The values of each 5-uple have been used to set the point of maximum fuzziness for the `high` and `low` membership functions of D. For example, the first value of the 5-uples has been used for the membership functions of D_1, the second for those of D_2 and so on. Once membership functions have been defined, the process of fuzzification, reasoning and defuzzification has been carried out. In order to make a final decision on image authenticity, the defuzzified value of tampering presence has been compared with the binary threshold $thr=0.5$. Note that only membership functions of detection are set according to the values of the 5-uples, while the others are fixed.

4.6. Results and discussion

The accuracy of OR-based fusion and our fuzzy fusion system has been evaluated in terms of Area Under Curve (AUC). Fig. 4(a) shows the results we obtained on the data set of 1600 images. Since the performance of smooth and piecewise fuzzy methods are extremely similar, for the sake of readability we omit the latter's results. We clearly see that the results of the two methods are very close to each other: with the fuzzy approach that slightly outperforms the OR approach (+2.7% AUC). This can be explained by noting that the tampering classes have been designed so that at least one tool is ideally able to correctly detect the tampering. We did not introduce any unknown tampering that could alter the analyzed features. In addition, the number of tools we considered is quite limited. This is a case that is likely to be managed quite satisfactorily even by a simple OR operator, nevertheless, the fuzzy method already performs better.

In order to better highlight the potentiality of the fuzzy framework we have performed the same test on the second data set. Fig. 4(b) confirms that, as expected, the benefits brought by our system are now more significant (+6.9% AUC).

Let us now consider the data set of tampered faces. The previous data set has been used to test the robustness of the proposed fuzzy approach to incorrect information. It is the aim of this third and last data set to test the robustness to common yet powerful image processing that have not been considered in the previous experiments. This data set is of particular interest since it represents the closest we can

[4] From a machine learning perspective this would mean performing both training and testing phases on the same data.

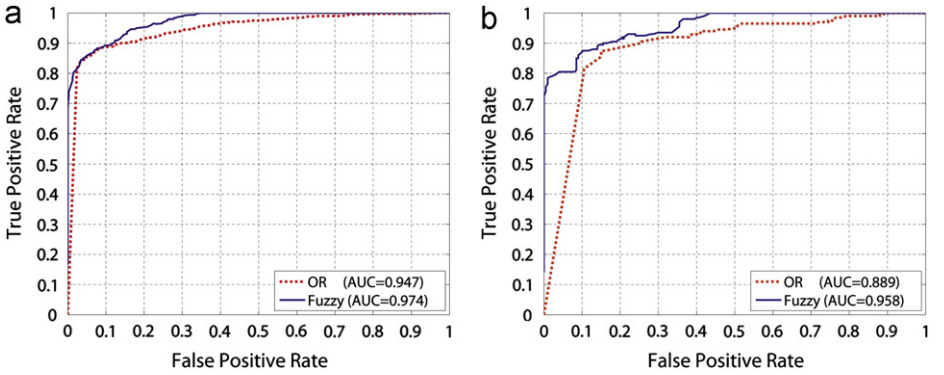

Fig. 4. ROCs for the two synthetic data sets: OR (dotted red line) and Smooth Fuzzy (solid blue line). (a) Synthetic data set of 1600 images and (b) synthetic data set of 400 *textured* images.

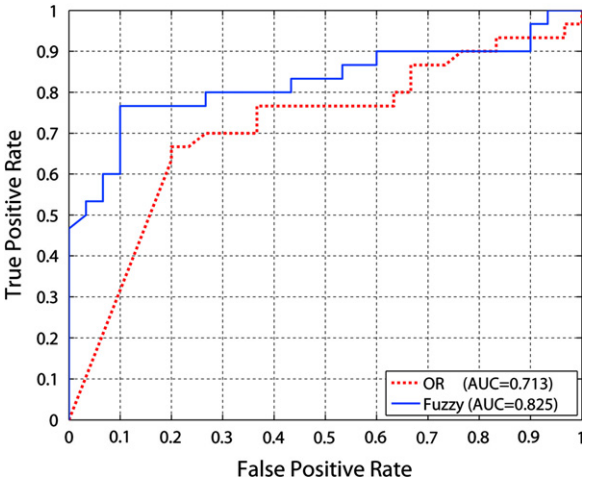

Fig. 5. ROCs for data set of tampered faces: OR (dotted red line) and Smooth Fuzzy (solid blue line). The proposed fuzzy approach outperforms the logical OR method by 11% in terms of AUC.

get to real cases of tampering. As we can clearly see in Fig. 5, in this case the benefits brought by our framework are evident (+ 1.2% AUC). Moreover, it is important to point out that a large portion of such gain is located in the leftmost part of the curve that corresponds to low P_{fa} ($P_{fa} < 0.15$), which represent the most common working conditions that are likely to be used in practice. In conclusion, the results we obtained represent an encouraging step towards the correct understanding of what may happen in real-world scenarios, where unknown processing is likely to introduce doubtful cases that the fuzzy approach can handle more efficiently.

4.7. Robustness against variations of parameters

According to Section 4.4, the reliabilities R_A, R_C and R_E have been derived from the respective papers while R_B and R_C were defined experimentally. Although this is the typical domain of system designers is exploited, such assignment of constant values may appear as an arbitrary choice depending on experimental data. In this section we demonstrate the robustness of the proposed approach with respect to relatively small variations of reliability values. To this aim we iterated the experimental procedure of Section 4.5 by varying R_B in [0.3, 0.5] and R_C in [0.7, 0.9] with step 0.05. The results are shown in Fig. 6: for each data set we show the ROC curves corresponding to the best (solid lines) and worst (dotted lines) performance of both fusion methods. We can observe that in terms of AUC the difference between the best and worst cases is small on all data sets (1.7%, 5% and 3% respectively). This experiment confirms the robustness of the fuzzy approach when one assigns suboptimal values to reliability. Finally, it is important to note that the fuzzy method continues to outperform the OR method also in the worst cases.

4.8. Computational complexity

In this section we discuss the computational complexity of our fuzzy approach. In a general case in which K forensic tools are employed we have 2^K possible K-uples belonging either to T_{true}, T_{false} or T_{doubt}. Each interaction corresponds to an if–then rule in the form of Eq. (5) or (10), thus resulting in 2^K compound rules. However, our Matlab® implementation cannot deal directly with rules in such a form, thus requiring further processing to convert them into an acceptable format. Such processing consists of Eqs. (6) and (7) and on a particular indexing for fuzzy sets, that is necessary in order to apply the majority criterion to reliabilities. The final number of basic rules amounts to 2^{2K}. In our specific case of $K=5$ tools the system consists of 32 cases generating a set of 1024 if–then rules. On a common desktop configuration (3 GHz dual-core processor, 4GB RAM, 32bit OS), the optimized version of our code resulted in the following execution times: about 1 second to build T_{true}, T_{false} and T_{doubt} (this operation is performed only once for each data set); 0.2 seconds to build the fuzzy inference system and 0.5 seconds to resolve all rules (these operations are performed once for each image).

5. Concluding remarks

In this paper we focused on the problem of dealing with uncertainty introduced by the parallel use of several unreliable image forensics tools. Usually each forensic technique deals with a single type of manipulation, while

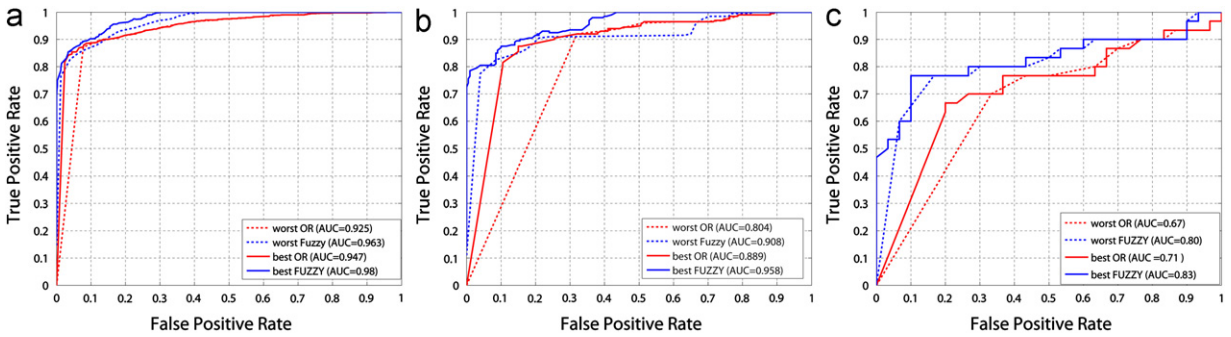

Fig. 6. Robustness with respect to variations of R_B and R_C. Solid (resp. dotted) lines indicate the performance of the best (resp. worst) case. Sensitivity to variations of reliability in the neighborhood of the assigned values is rather small. (a) synthetic data set of 1600 images, (b) synthetic data set of 400 *textured* images and (c) realistic data set of 60 natural images.

a real tampering is often the result of several processing. It is therefore necessary to employ more tools cooperating with each other. Several problems may arise when we need to make a single decision from outputs that are heterogenous, discording or incomplete. To cope with these problems, we proposed a fusion framework based on fuzzy logic. The results we obtained are promising, nevertheless several issues still need to be explored, including: implementation of a wider set of forensic tools working on different manipulations; implementation of a strategy to tackle the exponential growth of if–then rules occurring every time that a new tool is added to the system (e.g. by exploiting the weighted Hamming distance currently employed to map non standard cases or by means of hierarchical clustering); extension of the theoretical framework to the most complex case where the suspicious tampered region is not known a priori; test of accuracy on a large real-world data set of tampered images (i.e. images gathered from the Web); comparison against other soft decision approaches (e.g. Bayesian).

Acknowledgements

This work was partially supported by the REWIND Project, funded by the Future and Emerging Technologies (FET) programme within the 7FP of the EC, under grant 268478.

References

[1] E. Delp, N. Memon, M. Wu, Special issue on digital forensics, IEEE Signal Processing Magazine 26 (2).
[2] J. Redi, W. Taktak, J. Dugelay, Digital image forensics: a booklet for beginners, Multimedia Tools and Applications (2011) 1–30.
[3] J. Lukáš, J. Fridrich, Estimation of primary quantization matrix in double compressed JPEG images, in: Proceedings of Digital Forensic Research Workshop, Cleveland, Ohio, USA, August, 2003, pp. 5–8.
[4] H. Farid, Exposing digital forgeries from JPEG ghosts, IEEE Transactions on Information Forensics and Security 4 (2009) 154–160.
[5] A. Popescu, H. Farid, Exposing digital forgeries by detecting traces of resampling, IEEE Transactions on Signal Processing 53 (2) (2005) 758–767.
[6] B. Mahdian, S. Saic, Blind authentication using periodic properties of interpolation, IEEE Transactions on Information Forensics and Security 3 (3) (2008) 529–538.
[7] W. Luo, Z. Qu, J. Huang, G. Qiu, A novel method for detecting cropped and recompressed image block, in: Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP), Honolulu, Hawaii, USA, April, 2007, pp. II-217–II-220.
[8] Z.C. Lin, J.F. He, X. Tang, C.K. Tang, Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis, Pattern Recognition 42 (2009) 2492–2501.
[9] S. Bayram, H. Sencar, N. Memon, A survey of copy-move forgery detection techniques, in: IEEE Western New York Image Processing Workshop, Rochester, NY, USA, November, 2008.
[10] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, G. Serra, A SIFT-based forensic method for copy-move attack detection and transformation recovery, IEEE Transactions on Information Forensics and Security, (3) 1099–1110. in press, http://dx.doi.org/10.1109/TIFS.2011.2129512.
[11] M. Kharrazi, H.T. Sencar, N. Memon, Improving steganalysis by fusion techniques: a case study with image steganography, in: Transactions on Data Hiding and Multimedia Security, 2006, pp. 123–137.
[12] T. Terano, K. Asai, M. Sugeno, Fuzzy Systems Theory and its Applications, Academic Press, Boston, 1992.
[13] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision, IEEE Transactions on Systems, Man, and Cybernetics SMC-3 (1973) 28–44.
[14] M. Sugeno, Industrial Applications of Fuzzy Control, Elsevier Science Inc., New York, NY, USA, 1985.
[15] G. Chetty, M. Singh, Nonintrusive image tamper detection based on fuzzy fusion, International Journal of Computer Science and Network Security 10 (2010) 86–90.
[16] L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.
[17] E. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, International Journal of Man-Machine Studies 7 (1) (1975) 1–13.
[18] S. Sivanandam, S. Sumathi, S. Deepa, Introduction to Fuzzy Logic using MATLAB, Springer Verlag, 2007.
[19] W. Leekwijck, E. Kerre, Defuzzification: criteria and classification, Fuzzy Sets and Systems 108 (2) (1999) 159–178.
[20] M. Delgado, A. Gómez-Skarmeta, F. Martin, A fuzzy clustering-based rapid prototyping for fuzzy rule-based modeling, IEEE Transactions on Fuzzy Systems 5 (2) (1997) 223–233.
[21] X. Zeng, J. Keane, Learning for hierarchical fuzzy systems based on the gradient-descent method, in: Proceedings of IEEE International Conference on Fuzzy Systems, Vancouver, Canada, July 2006, pp. 92–99.
[22] T. Bianchi, A. Piva, Detection of non-aligned double JPEG compression with estimation of primary compression parameters, in: Proceedings of IEEE International Conference on Image Processing (ICIP), Brussels, Belgium, September 2011, pp. 1969–1972.
[23] T. Bianchi, A.D. Rosa, A. Piva, Improved DCT coefficient analysis for forgery localization in JPEG images, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, May 2011, pp. 2444–2447.
[24] J.C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: Advances in Large Margin Classifiers, 1999, pp. 61–74.