DEALING WITH UNCERTAINTY IN IMAGE FORENSICS: A FUZZY APPROACH

M. Barni and A. Costanzo

University of Siena, Dept. of Information Engineering, Via Roma 56, Siena, Italy
barni@dii.unisi.it, andreacos82@gmail.com

ABSTRACT

Image forensics research has mainly focused on the detection of artifacts introduced by a single processing tool. In tamper detection applications, however, the kind of artifacts the forensic analyst should look for is not known beforehand, hence making it necessary that several tools developed for different scenarios are applied. The problem, then, is twofold: i) devise a sound strategy to elaborate the information provided by the different tools into a single output, and ii) deal with the uncertainty introduced by error-prone tools. In this paper, we introduce a framework based on Fuzzy Theory to overcome these problems. We describe a practical implementation of the proposed framework putting the theoretical principles in practice. To validate the proposed approach, we carried out some experiments addressing a simple realistic scenario in which three forensic tools exploit artifacts introduced by JPEG compression to detect cut&paste tampering within a specified region of an image. The results are encouraging, especially when compared with those obtained by simply XOR-ing the output of the the single detection tools.

1. INTRODUCTION

Nowadays the majority of images are created, stored and distributed in a digital format that is fairly easy to edit and tamper with. As a result, digital image forensics has become an important 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, 3, 4, 5]. Usually each forensic technique deals with a single type of manipulation, however when an analyst is asked to decide about the integrity of a given image, the kind of manipulation the image has undergone is not know beforehand. Therefore, if we are interested in deciding whether an image has been tampered with or not, the application of a single detection method may not be enough. Several problems may arise in this scenario. A forensic tool usually is not perfect: it may have technical limitations, be prone to errors or provide unreliable information. As a consequence, the output of such tool is likely to be affected by some degree of uncertainty and is not fully trustable. While the reliability of forensics analysis may be improved by combining the output of several tools, the way such outputs should be combined is not obvious, since different tools may provide heterogeneous and imperfect indications, referring to completely different analysis. In this paper we present a solution based on fuzzy logic. Fuzzy logic has been used in a very wide range of domains, such as sensor networks, automatic vehicle navigation, industrial and aerospatial applications, databases and domotics. The fuzzy-logic approach has demonstrated to be useful in those applications where reasoning needs to be robust against noise, approximate or imprecise inputs [6]. For this reason we believe that fuzzy logic may also help to deal with the incomplete or conflicting outputs provided by different forensic algorithms. 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 [7]. However, as opposed to our contribution, the system described in [7] relies on fuzzy integrals applied to the features extracted by the forensic algorithms, thus making impossible a direct comparison with our scheme. 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 problem and we introduce a general fuzzy framework to address it. In section 4, we discuss our experimental results. We compare the performance of our framework with those of a binary OR method. Finally, in section 5, where we outline some directions for future research.

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.

2. FOUNDATIONS OF FUZZY LOGIC

Fuzzy sets theory was conceived in 1965 by Zadeh as an extension of classic set theory [8]. From this initial concept a multi-value fuzzy logic has been derived in subsequent years as an extension of Boolean logic. Zadeh affirmed that despite people do not require precise information as input for their reasoning, they are capable of highly adaptive control. If such capability could be transferred to systems, they would perhaps be more effective and easier to implement. Fuzzy logic was designed to deal with imperfect information, which in the real world is more often the norm rather than the exception. Zadeh defined this methodology *computing with words*.

2.1. Fuzzy sets, operators and if-then statements

In order to understand the way fuzzy logic works, we need to introduce three concepts: fuzzy sets, fuzzy operators and if-then rules.

Let X be a space of objects. A fuzzy set A in X is a class of objects of X characterized by a membership function $\mu_A(x)$, that is a curve defining how each point $x \in X$ is mapped into a membership value in the interval $[0, 1]$. The value $\mu_A(x)$ represents the grade of membership of x in A [8].

Let us now apply this concept to logic. Classic Boolean logic requires that a proposition is either true (1) or false (0). Based on real world experience, fuzzy logic affirms that a proposition is not always totally false or totally true but true or false to some grade in the interval $[0, 1]$. In this way, 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 not too complicate. Let x and y be two fuzzy variables and μ a membership function. Standard fuzzy logical operators can be redefined as follows: AND(x,y) = min($\mu(x), \mu(y)$);

OR(x,y) = max($\mu(x), \mu(y)$); NOT(x) = $1 - \mu(x)$. The next step is the definition of fuzzy *if-then* rules. Let A and B be fuzzy sets. A fuzzy if-then rule is commonly represented as follows:

$$\text{IF } x \text{ is } A \text{ THEN } y \text{ is } B \tag{1}$$

The first part of the rule (x is A) is called antecedent, the second part (y is B) consequent. An antecedent can also consist of an arbitrary number of expressions. Most of the times one rule alone is not effective: there is the need of two or more rules that can play off one another. A set of if-then rules can be used to describe the behavior of a system.

More specifically, the interpretation of a set of if-then rules as in equation (1) consists of the following four steps [6]: assigning to each crisp input value a degree of membership according to the membership function of the respective fuzzy set; resolving multiple antecedents into a single value; truncating the output fuzzy sets and aggregating all rules; resolving the aggregation into a crisp output (e.g. by means of centroid or mean operators).

3. PROBLEM FORMALIZATION

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 set of features in the specified region looking for tampering traces and generates an output that tells 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 reduce the noise affecting the K outputs while merging them into a single value. Based on this value, we will then take a final decision on the authenticity of the region. As the number of available tools increases several problems may arise, thus making classic methods ineffective. For example, two mutually exclusive tools can invalidate a majority method while a tool very prone to errors can invalidate an OR method. For these reasons we need to devise an alternative reliable method to cope with multiple noisy inputs.

In order to apply the concepts introduced in section 2 we need to so-to-say *standardize* the output of the forensic tools. In particular, we require that all the tools share the same output format, consisting of a pair of values (D, R), where: $D \in [0, 1]$ is the degree of *detection*, that is a measure of the presence of the tampering trace within the analyzed image region. Values near 1 indicate a high presence of the tampering trace; $R \in [0, 1]$ is the *reliability* of D, that is a measure of the confidence of the tool on the detection value. Values near 1 indicate a high confidence.

With a common ground for each tool, we need to describe the behavior we expect from them. Let us suppose that a region of an image *I* has undergone a tampering operation. The question we want to answer is: "*If everything goes smoothly, what kind of output are we expecting from the tools at our disposal?*". Depending on the nature of the manipulation, a tool may or may not be able to detect a region as tampered. Let us indicate the capability of detecting a tampering trace with Y and the incapability with N. If we have K tools, each 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 in *ideal* conditions. Note that there may be some sequences of outputs that are not specified a priori: some sequences are not possible under an ideal behavior or correspond to unknown situations. In the following we will use the symbols T_{true} and T_{false} to indicate the tables whose columns give the expected 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 non-expected (*non-standard*) K-uples of tools' outputs. Since the definition of these tables depends on the available tools and the knowledge of their expected behavior, in the following will we assume that they are always available.

To exemplify, let us consider a case in which two tools (t_1 and t_2) are available. Let us assume that t_1 (t_2) considers tampered a region of an image if there are traces of aligned (misaligned) double compression. If we apply these tools to a region that has undergone an aligned double compression, we expect a (Y,N) answer from the 2 tools; on the other side, if the region has undergone misaligned double compression we expect a (N,Y). Moreover, if region is not tampered we expect a (N,N). Finally, if we obtain a (Y,Y) we are dealing with a doubtful, maybe partially true answer.

3.1. The proposed Fuzzy framework

In our framework the pairs (D, R) provided by the forensic tools are the input fuzzy variables of the system. In a real scenario a tool is not perfectly secure about the presence (absence) of a manipulation therefore it may output a noisy value of D that is high (low) but not necessarily near 1 (0). Moreover, 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, thus risking a loss of information that could be still used somehow. The system we propose deals with each of these problems by reasoning on K input pairs (D, R) produced by the forensic tools. The algorithm construction starts by building a set of if-then rules based on the tables T_{true} and T_{false}. 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, thus producing a number that needs to be compared with a threshold to obtain a final answer on region authenticity.

3.2. Framework implementation

For sake of clarity in this section we give an intuitive description of the chosen fuzzy sets and the construction of the fuzzy inference rules. Detection and reliability values can either be considered low or high, where with `low` and `high` we mean fuzzy sets characterized by a membership function. Similarly, 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 to represent the presence of tampering: `very weak`, `weak`, `neither weak nor strong`, textttstrong and textttvery strong. In the following we describe how we use these sets to generate if-then rules.

3.2.1. Automatic construction of standard rules

We create the so called *standard* if-then rules by looking exclusively at the columns of T_{true} and T_{false}. These tables describe the behavior we expect from the tools in the presence of a certain tampering. Intuitively, we try to assign them a linguistic meaning. Generally no tool is either wholly capable or incapable of detecting a certain tampering, rather a tool can be *more* or *less* capable or incapable. Let us focus on the capability of detection (Y): if a tool provides a high value of detection with a high reliability we consider it *more* capable of detection. We consider the same tool *less* capable (but still able of correct discrimination) if it provides a high value of detection with a low reliability. Similarly for the incapability of detection (N). These

concepts can be formalized as follows:

$$\begin{array}{lcl} \text{Y} = & (\text{detection is } \texttt{high} & \text{AND reliability is } \texttt{high}) \text{ OR} \\ & (\text{detection is } \texttt{high} & \text{AND reliability is } \texttt{low}) \\ \text{N} = & (\text{detection is } \texttt{low} & \text{AND reliability is } \texttt{high}) \text{ OR} \\ & (\text{detection is } \texttt{low} & \text{AND reliability is } \texttt{low}) \end{array} \tag{2}$$

This step allows us to express the capability (incapability) of detection of each tool in terms of fuzzy variables (D, R) and fuzzy sets (`low` and `high`). In fact, in section 3 we saw that columns of T_{true} and T_{false} are K-dimensional arrays whose elements are either Y or N. Let $\boldsymbol{s}$ be one of these arrays: the full antecedent of an if-then rule is built by substituting to each element of $\boldsymbol{s}$ the corresponding expression of equation 2. Finally, the choice of the consequent of the rule depends on whether $\boldsymbol{s}$ belongs to T_{false} or T_{true}:

$$\begin{array}{l} \text{if } \boldsymbol{s} \in T_{true} \text{ consequent is: THEN tampering is } \texttt{very strong} \\ \text{if } \boldsymbol{s} \in T_{false} \text{ consequent is: THEN tampering is } \texttt{very weak} \end{array} \tag{3}$$

For sake of clarity, let us consider again the example of section 3. Let us consider the case (Y,N)$\in T_{true}$. We can express this case in the form of an if-then rule as follows: IF (t_1 = Y) AND (t_2 = N) THEN region is tampered. Accordingly to equations (2) and (3) the new if-then rule becomes:

IF (D_1 `high` AND R_1 `high` OR D_1 `high` AND R_1 `low`)
AND (D_2 `low` AND R_2 `high` OR D_2 `low` AND R_2 `low`)
THEN tampering is `very strong`

3.2.2. Automatic construction of non standard rules

Construction of if-then rules for *non-standard* cases belonging to T_{doubt} is similar to that of standard cases. Again, we build antecedents as described in equation (2). However, to define consequents we need some further reasoning. When a non-standard case occurs we do not have a support from theory or experiments. Therefore we map this case into something that we know, according to the reliability of the various tools. The more a tool is reliable, the more we are willing to trust it. 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:

$$d(\boldsymbol{ns}, \boldsymbol{s}) = \sum_{i=1}^{K} R_i \cdot \text{XOR}\big(\boldsymbol{ns}(i), \boldsymbol{s}(i)\big) \tag{4}$$

where: K is the number of tools; R_i is the tool reliability; 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 equation (4) we compute the distance of $\boldsymbol{ns}$ from all the M standard sequences and select the closest one as follows:

$$\boldsymbol{s}_{min} = \arg\min_{n} \big[d(\boldsymbol{ns}, \boldsymbol{s}_n)\big], n = 1, 2, .., M \tag{5}$$

Since this process is an approximation based on experimental parameters, it is not wise to lean too much towards presence or absence of tampering. Therefore we choose to mitigate the consequent:

$$\begin{array}{l} \text{if } \boldsymbol{s}_{min} \in T_{true} \text{ consequent is: THEN tampering is } \texttt{strong} \\ \text{if } \boldsymbol{s}_{min} \in T_{false} \text{ consequent is: THEN tampering is } \texttt{weak} \end{array} \tag{6}$$

4. EXPERIMENTAL ANALYSIS

Our set of forensics tools consists of 3 algorithms working on JPEG compression characteristics. These tools rely on methods proposed by Luo et al. [5], Lin et al. [4] and Farid [2] respectively and can be used to detect cut&paste manipulations. In the following we will refer to them as t_A, t_B and t_C. In a nutshell they work as follows: 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 JPEG image with quality $QF_2 > QF_1$; t_B determines whether a region has been cropped from a JPEG image or from an uncompressed image and pasted without preserving grid alignment; t_C determines whether a region has has been cropped from a JPEG image and pasted preserving grid alignment. For a more in-depth explanation of these techniques we refer to the respective papers. Each tool provides a detection value in $[0, 1]$ as follows: t_A by means of a probabilistic SVM method; t_B by means of the median of the probability map in the analyzed region [4]; t_C by means of KS statistics as in [2].

The next step consists in the construction of T_{true} and T_{false} tables. According to the principles underlying t_A, t_B and t_C, and according to a preliminary experimental analysis, we identified four classes of tampered images for which the tools ideally provide different output triplets. By relying on such an analysis we built table 1, from which the T_{true} and T_{false} tables can be derived. The triplets that are not represented in table 1 will form the T_{doubt} table.

Tool	Class 1	Class 2	Class 3	Class 4	Original
t_A	Y	N	N	Y	N
t_B	N	Y	N	Y	N
t_C	N	Y	Y	Y	N

Table 1. Expected interactions between the 3 tools.

4.1. Image datasets

Starting from a set of 100 TIFF images we created 4 classes of images that simulate a cut&paste tampering by varying the quality factors and the alignment of JPEG compressions. Each class has been designed so that only a single tool (or a pair of tools) is able to detect the presence of the manipulation and is composed by 200 images. We have finally added 800 images that have been compressed only once (un-tampered images) In order to better highlight the advantages of the proposed framework we also built a second dataset. We observed that t_B tends to claim as tampered a specific type of natural images, those whose central region contains textures and regular geometric edges (e.g. buildings, walls, squares), compressed once with a very high quality factor. We expect our fuzzy system to perform better than a simple scheme based on the OR of the tools' outputs. To validate the above arguments, we have gathered a set of 50 natural images whose central regions contain textures and regular edges, compressed once with native camera quality factor $QF_1 = 100$. With these images we have built another dataset of 400 images with the same procedure of the first dataset. Both tampering and testing have been conducted on the 256×256 central area.

4.2. Experimental settings

The system features 6 inputs ($D_{A,B,C}$, $R_{A,B,C}$) and one output (tampering). We noticed that t_A is more reliable when the second JPEG quality factor QF_2 (normalized in $[0, 1]$) is high. Reliability of t_B and t_C does not seem to be affected by QF_2. Therefore we have used the following reliability values: $R_A = 0.4 \cdot QF_2$, $R_B = 0.4$

and $R_C = 0.85$. We used two different families of membership functions (MF) for both inputs and outputs: piecewise (trapezoidal) and smooth (combination of gaussians). Figure 1 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, since an input value is characterized by the same grade of membership for both classes. For an explanation of how we chose the points of maximum fuzziness we refer to section 4.3. Figure 2 shows the

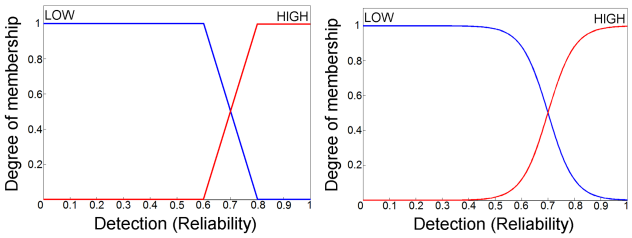


Fig. 1. MFs for input variables: piecewise (left) and smooth (right) depending on point of max fuzziness (e.g. $p = 0.7$).

membership functions for the output variable representing intensity of tampering evidence. We have defined five fuzzy sets accordingly to section 3.2. From left to right `very weak`, `weak`, `neither weak nor strong`, `strong`, `very strong`. In some of our

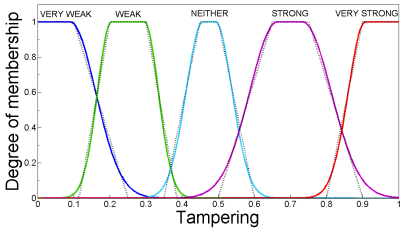
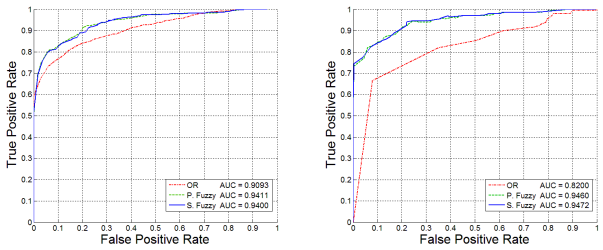


Fig. 2. MFs for tampering: smooth (solid) and piecewise (dotted).

experiments we have also slightly changed the shapes and the points of max fuzziness of each set of curves. Although for sake of brevity we are not describing such tests, the results we obtained were not noticeably different from those we are presenting.

4.3. Results and discussion

In our experiments we first calculated the Receiver Operating Curve (ROC) of each algorithm on dedicated datasets (i.e. only on images constructed to satisfy the assumptions the tools rely on). We then aggregated these 3 curves by sampling the probability of false alarm (P_{fa}) with a step of 0.01. This allowed us to obtain at each step the three thresholds giving a specific P_{fa} for all the algorithms. We organized these thresholds in triplets that we finally used as binary thresholds to build the ROC of logical OR and as points of maximum fuzziness to build the ROC of fuzzy methods. Figure 3(a) shows the results we obtained on the dataset of 1600 images. We can observe that the performance of piecewise and smooth fuzzy methods are basically the same. When compared to the logic OR, results are promising, although not dramatically better (+3% AUC). This can be explained by noting that the classes of tampering 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



(a) 1600 images (b) 400 images

Fig. 3. ROCs for the two datasets: OR (red dash-dotted line), Piecewise Fuzzy (dashed green line) and Smooth Fuzzy (solid blue line).

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 dataset (figure 3(b)). As expected, the benefits brought by our system are now more significant (+12.7% AUC). Such dataset, in fact, simulates what is likely to happen in real-world scenarios, where unknown processing is likely to introduce doubtful cases that the fuzzy approach can handle more efficiently.

5. CONCLUSIONS

In this paper we focused on the problem of dealing with uncertainty introduced by the parallel use of several unreliable image forensics tools. The results we obtained are promising, nevertheless several issues still need to be explored, including: integration of a wider set of forensic tools; test of the accuracy on a real-world dataset of tampered images; extension to handle situations where the suspicious tampered region is not known a priori; comparison with other soft decision approaches.

6. REFERENCES

[1] E. Delp, N. Memon, and M. Wu, "Special issue on digital forensics," *IEEE Signal Processing Magazine*, vol. 26, 2009.

[2] Hany Farid, "Exposing digital forgeries from JPEG ghosts," *IEEE T. on Information Forensics and Security*, vol. 4, pp. 154–160, 2009.

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

[4] 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, pp. 2492–2501, 2009.

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

[6] T. Terano, K. Asai, and M. Sugeno, *Fuzzy Systems Theory and its Applications*, Academic Press Boston, 1992.

[7] G. Chetty and M. Singh, "Nonintrusive image tamper detection based on fuzzy fusion," *International Journal of Computer Science and Network Security*, vol. 10, pp. 86–90, 2010.

[8] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.