Paper submitted to International Journal of Digital Crime and Forensics

A Forensic Tool for Investigating Image Forgeries

Marco Fontani, University of Siena, Via Roma 56, 53100, Siena, Italy

Tiziano Bianchi, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129, Torino, Italy

Alessia De Rosa, National Inter-University Consortium for Telecommunications, Via di S.Marta 3, 50139, Firenze, Italy

Alessandro Piva, University of Florence, Via di S.Marta 3, 50139, Firenze, Italy –* alessandro.piva@unifi.it

Mauro Barni, University of Siena, Via Roma 56, 53100, Siena, Italy

ABSTRACT

Images have always been considered a reliable source of evidence in the past. Today, the wide availability of photo editing software urges us to investigate the origin and the integrity of a digital image before trusting it. Although several algorithms have been developed for image integrity verification, a comprehensive tool that allows the analyst to synergically exploit these algorithms, and to reach a final decision based on their output, is still lacking. In this work we propose an image forensic tool trying to fill this gap. The proposed tool exploits state of the art algorithms for splicing detection, with forgery localization capabilities, and make them available to the analyst through a graphical interface. In order to help the analyst in reaching a final assessment, a decision fusion engine is employed to intelligently merge the output of different algorithms, boosting detection performance. The tool has a modular architecture, that makes it easily scalable.

Keywords: Image Forensics, Forgery Localization, Decision Fusion, Image Forensic Analyst, Dempster-Shafer Theory of Evidence, Integrity Verification, Authenticity Assessment.

INTRODUCTION

The advent of image processing technologies easily enables modification and manipulation of digital visual data, so that we are no longer confident that what we are seeing in a photo is a true representation of what really happened: the value of photography as a record of events must be carefully evaluated. Such a need comes from different fields of application: one of the most important is the forensic scenario, in which the trustworthiness of images must be assured before using them as potential evidences. Image Forensics (IF) (under the umbrella of the more general Digital Forensics) is the science addressing the validation, identification, analysis, interpretation of digital images as potential evidences. One of the most interesting tasks in IF is *splicing detection*, that aims at understanding if a given photo is a composition of different shots. Several approaches for splicing detection have been proposed recently (Piva, 2013), sharing the same basic idea: creating a forgery usually requires some processing steps, and these leave some statistical footprints into the signal.

In this context, the Image Forensic Analyst (IFA from now on) is the professional that applies technological means for extracting information on image history and for assuring its credibility, after the chain of custody (COC) procedures have been applied for acquiring, transferring and storing the visual data (see Figure 1). Usually, the IFA has not any previous knowledge about the history of the images that he is considering (i.e., what device acquired them, whether a processing software has been used to edit

them or not, and so on), and must produce a report about the credibility of the analysed contents. To reach this goal, the IFA today could use algorithms developed in the IF literature, but in practice several problems rise: algorithms are stand-alone, in that they focus on a specific footprint and ignore the others; they assume some prior knowledge about the kind of processing that could have been carried on the media; and, finally, they do not expose a user interface helping the IFA in setting up the analysis and interpreting the results.

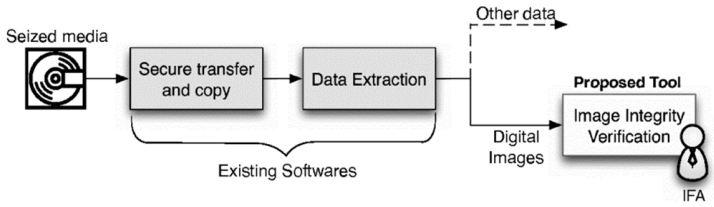


Figure 1: A simplified version of the chain of custody, where the positioning of the proposed tool is highlighted.

As a result, several issues are still open when we consider to apply the results coming from academic research to practical cases, where the IFA needs technological instruments that facilitate him in reaching a conclusion:

- there are no tools that help the IFA to exploit the different capabilities of existing algorithms. We should consider that, in the end, the IFA is mainly concerned about image integrity (i.e. algorithm output), and only indirectly concerned about footprint detection (i.e. algorithm functioning);
- usually the presence/absence of forensic fingerprints can be verified on the image as a whole, or on a selected suspected region; only few examples of tools that provide a fine-grained localization of forgery within a digital image have been proposed;
- each tool usually considers to reveal one specific trace of tampering, but the IFA cannot know in advance which traces should be searched for. Therefore, there is need for a tool that helps in interpreting and putting together the outputs from different algorithms.

For these reasons, we believe that providing a comprehensive system for image splicing detection is an important contribution for the diffusion of image forensic technologies.

In this work we present a tool for evaluating the integrity of a digital image, by revealing whether the image is a malicious composition of different contents or an original shot of an event. This tool combines different image forensic algorithms for splicing detection through a specifically tailored decision fusion framework, improving the detection performance with respect to single tools, and provides an intuitive and functional interface that allows the IFA to easily access this multi-clue analysis. The system we propose contributes to solve each of the previously listed problems by:

- integrating some of the most recent and effective splicing detection algorithms into a single graphical tool to be easily used by the IFA;
- providing the IFA with probability maps telling which parts of the image are more likely to be tampered, so as to help in the selection of suspect regions to be analysed;
- fusing algorithms outputs using a decision fusion method, in order to provide a single output on the credibility of the evaluated region, with improved accuracy with respect to single splicing detection algorithms.

TOOLS FOR DIGITAL IMAGE FORENSICS

In the literature, several approaches have been proposed in order to verify the integrity of an image (Piva, 2013). Most of the existing tools look for the presence of some specific traces left by the acquisition process, coding, or subsequent processing steps.

As to the acquisition process, forensic tools may use sensor imperfections (Chen, 2008), color filter array interpolation (Swaminathan, 2008; Ferrara, 2012), lens characteristics (Choi,2006; Dirik, 2008), or scanner features (Gou 2009). As to coding, most of existing forensic tools deal with artifacts left by JPEG compression, particularly in the presence of multiple JPEG compressions (Farid, 2009; Bianchi, 2012a; Lin, 2009; Popescu, 2004; Li, 2008; Bianchi, 2011, Luo, 2007; Barni, 2010). Common processing steps like image resizing, rotation, or de-mosaicking, also leave useful traces exploited by a number of tools (Mahdian, 2008; Popescu, 2005a; Popescu, 2005b; Gallagher 2008). Other approaches consider traces left by image enhancement (Kirchner 2010) or by particular kinds of attacks, like copy-move forgeries (Amerini, 2011; Fridrich, 2003). An effective way to detect image manipulation is to verify whether light color, position and intensity are consistent throughout the scene (Kee, 2010).

JPEG forensic tools

The JPEG format is adopted in most of the digital cameras and image processing tools. Since many forensic methods have been studied to detect the presence of tampering in JPEG images, in our tool we will consider the integration of algorithms operating on this class of images.

In general, the manipulation is detected by analyzing proper artifacts introduced by JPEG recompression occurring when the forged image is created; in particular, such artifacts can be categorized into two classes, according to whether the second JPEG compression adopts a discrete cosine transform (DCT) grid aligned with the one used by the first compression or not. Approaches belonging to the first category include (Farid, 2009; Lin, 2009; Popescu, 2004, Li, 2008; Bianchi, 2011), whereas the presence of non-aligned double JPEG compression has been investigated in (Bianchi, 2012a; Luo, 2007; Barni, 2010). A promising approach is the one introduced by Popescu & Farid (2004): here, it is proposed to detect the presence of double aligned JPEG compression by observing the effect of consecutive quantizations on the same DCT coefficient, modeled as

$$y = \left(round\left(\frac{x}{Q_1} \right) \frac{Q_1}{Q_2} \right). \quad (1)$$

where x is the original DCT coefficient, y the same coefficient after two quantizations, the first one with a factor Q_1 and the second with a factor Q_2.

It is observed that the above processing introduces periodic artifacts into the histogram of DCT coefficients, which can be revealed by means of various approaches (Lin, 2009; Bianchi, 2011). An interesting technique is that proposed in (Farid, 2009), where differently compressed versions of the image are compared: when the same quality factor of the tampered area is adopted, a spatial local minima, the so-called JPEG ghost, will appear in correspondence of the forgery. This is consistent with the fact that quantization of y by Q_1 will result in a nearly idempotent operator.

Concerning the methods for the detection of non-aligned double JPEG compression, a well-known approach is the one presented in (Luo, 2007): an 8x8 blocking artifact characteristics matrix (BACM) is computed in the pixel domain to measure the symmetrical property of the blocking artifacts in a JPEG image; an asymmetric BACM will reveal the presence of misaligned JPEG compressions; 14 features are extracted from a BACM and fed to a classifier in order to distinguish the BACM of doubly compressed images.

In (Bianchi, 2012a), the authors propose a method based on a single feature whose experimental results are superior to the previous works. The method is based on the computation of an integer periodicity map (IPM) of size 8x8 indicating whether the histogram of the DCT coefficients obtained by applying each of the 64 possible grid shifts exhibit a periodic behavior. If the IPM has an higher value at position (r,c), this indicates a double JPEG compression with shift (r,c). Conversely, a mostly uniform IPM indicates a singly compressed image. The above effect is measured by computing the entropy of the IPM, which can range from 0 (high evidence of double compression) to 6 (high evidence of single compression).

From tampering detection to forgery localization

Most of the above approaches rely on the hypothesis to have some knowledge about the location of a possibly manipulated area, for example by applying a segmentation of the image under test before the forensic analysis as done in (Barni, 2010), or they are just designed to analyse the whole image, so that the correct localization of the forgery in a tampered image is still an open issue.

For many algorithms, a coarse forgery localization can be achieved by resorting to block processing: however, only few forensic algorithms have been specifically designed to localize in an automatic way the tampered regions with fine resolution.

For what concerns the IFA, the first step toward forgery localization can be thought of as an algorithm that, without any prior information about the location of the manipulated area, outputs a map giving the probability for each pixel of being tampered. Most of the existing approaches exploit JPEG artifacts, which can be analysed at a fine-grained scale of 8x8 blocks of pixels; promising results have been obtained in the case of aligned double JPEG compression artifacts (Lin, 2009; Bianchi, 2011) and, more recently, in the case of non-aligned double JPEG compression artifacts (Bianchi, 2012b). In (Lin, 2009), double JPEG compression is detected by computing a tampering probability map of the image according to a proper statistical model of DCT coefficients. In (Bianchi, 2011), a significant improvement of the accuracy of the probability map estimation is obtained by modeling DCT coefficients as a mixture of doubly and singly compressed coefficients. A similar mixture model approach has also been applied for the localization of non-aligned double JPEG artifacts in (Bianchi, 2012b).

DECISION FUSION: TURNING CLUES INTO BELIEF

Very often, the creation of a realistic forgery involves the application of more processing steps in order to make the final result realistic and credible. Therefore, a number of different footprints may be left that can be used to detect the presence of tampering, and this suggests to analyse the authenticity of a digital image by using different tamper detection tools. Furthermore, it may happen that the presence of one footprint inherently implies the absence of another, since some footprints are mutually exclusive by definition, so simple decision fusion approaches like majority voting are ruled out. Finally, information about the reliability of forensic algorithms is usually available, since their performance often depend on observable characteristics of the image under analysis (noticeable examples are, in the case of JPEG images, the quality of the last compression, or the size of the analysed region).

For these reasons, we think it is essential for the IFA to perform a multi-clue analysis, employing a set of forensic algorithms. To this end, we integrate in the proposed tool an effective decision fusion engine, based on Dempster-Shafer Theory of Evidence (DST) and tailored to the image forensics scenario, that was first discussed in (Fontani, 2011).

The basic idea underlying this model is to treat each forensic algorithm as an "expert" providing its knowledge to the system about presence of a specific footprint. This information is then fused, taking into account the reliability of each algorithm, and the knowledge about plausible/impossible combination of footprints. In this Section this framework is introduced: first we give the formalization for a single algorithm, then we show how new algorithms can be added, and finally we introduce knowledge about footprint relationships into the framework.

DST formalization of the problem: single tool

DST was introduced by Arthur Dempster (1967), and today is a widely used theory in inference reasoning. With respect to the more classical Bayesian approach, the use of DST avoids the necessity of assigning prior probabilities (that, in the image forensics field, would be extremely difficult to estimate) and also provides more intuitive tools for managing the uncertain knowledge resulting from the forensic algorithms. Throughout the following, we will make use of several instruments of DST, that are defined and explained in (Dempster, 1967), to define the decision fusion framework embedded in the tool.

For sake of clarity, we start by formalizing the proposed framework for one tool only, let us call it *ToolA*. We assume that the algorithm outputs a value $A \in [0, 1]$ and has a reliability $A_R \in [0,1]$. We first consider the information coming from the detection value by introducing a variable with frame: $\Theta_A = \{ta, na\}$, where *(ta)* is the event "image has undergone a tampering detectable using *ToolA*" and *(na)* is the event "image has not undergone a tampering detectable using *ToolA*". Information provided by *ToolA* can then be summarized with the following Basic Belief Assignment (BBA) (Dempster, 1967) over the frame Θ_A:

$$m_A^{\Theta_A}(X) = \begin{cases} A_T \text{ for } X = & \{(ta)\} \\ A_N \text{ for } X = & \{(na)\} \\ A_{TN} \text{ for } X = & \{(ta) \cup (na)\} \end{cases} \quad (2)$$

where A_T, A_N and A_{TN} are functions that convert the output of *ToolA*, respectively, in a belief assignment for proposition *(ta)*, *(na)* and *(ta)*$\cup$*(na)*; this last proposition models the *doubt* that *ToolA* has about the presence of its footprint (Dempster, 1967). Choosing these functions is a rather intuitive task, since they basically tell how the tool output must be interpreted in terms of presence/absence of the trace; some examples will be shown later.

We now turn to introduce the available knowledge about reliability of the tool, carried by A_R. We adopt the convention that $A_R = 0$ means that *ToolA* is totally unreliable, and $A_R = 1$ indicates that *ToolA* is an oracle; equation (2) can thus be rewritten, according to the "belief discounting" method, in the following intuitive way (see (Fontani, 2013) for details):

$$m^{\Theta_A}(X) = \begin{cases} A_R \cdot A_T \text{ for } X = & \{(ta)\} \\ A_R \cdot A_N \text{ for } X = & \{(na)\} \\ C_A \quad \text{for } X = & \{(ta) \cup (na)\} \end{cases} \quad (3)$$

where $C_A = (1 - A_R(A_T + A_N))$. Looking at equation (3) we see clearly that the reliability parameter acts as a discount with respect to beliefs on the informative propositions *(ta)* and *(na)*.

Introducing new tools

Suppose we want to introduce in our framework a new tool *ToolB*, that outputs a value $B \in [0,1]$ and has a reliability B_R . The same formalism used previously will lead us to write m^{Θ_B}, a BBA that summarizes the knowledge for this new tool, defined over the frame Θ_B. Since m^{Θ_A} and m^{Θ_B} are defined on different frames, they cannot be fused with Dempster's rule directly. However, we can first marginalize both the BBAs eliminating reliability variables; then redefine m^{Θ_A} and m^{Θ_B} on the new frame $\Theta_A \times \Theta_B$ using vacuous extension (Dempster, 1967); finally use Dempster's rule to combine these two BBAs, yielding $m^{\Theta_A \times \Theta_B}$:

$$m^{\Theta_A \times \Theta_B}(X) = \begin{cases} A_R \cdot A_T \cdot B_R \cdot B_T \text{ for } X = \{(ta,tb)\} \\ A_R \cdot A_T \cdot B_R \cdot B_N \text{ for } X = \{(ta,nb)\} \\ A_R \cdot A_T \cdot C_B \text{ for } X = \{(ta,tb) \cup (ta,nb)\} \\ A_R \cdot A_N \cdot B_R \cdot B_T \text{ for } X = \{(na,tb)\} \\ A_R \cdot A_N \cdot B_R \cdot B_N \text{ for } X = \{(na,nb)\} \\ A_R \cdot A_N \cdot C_B \text{ for } X = \{(na,tb) \cup (na,nb)\} \\ C_A \cdot B_R \cdot B_T \text{ for } X = \{(ta,tb) \cup (na,tb)\} \\ C_A \cdot B_R \cdot B_N \text{ for } X = \{(ta,nb) \cup (na,nb)\} \\ C_A \cdot C_B \text{ for } X = \begin{Bmatrix} (ta,tb) \cup (na,tb) \cup \\ \cup (ta,nb) \cup (na,nb) \end{Bmatrix} \end{cases} \quad (4)$$

Where $C_A = (1 - A_R(A_T + A_N))$ and $C_B = (1 - B_R(B_T + B_N))$, *(tb)* is the proposition "image has undergone a tampering detectable using *ToolB*" and *(nb)* is the proposition "image has not undergone a tampering detectable using *ToolB*". If another tool *ToolX* becomes available, the associativity of Dempster's rule allows to combine directly its BBA with the one currently present in the framework, so we will always need to extend the domain of only two BBAs.

Introducing traces relationships

Up to now, we did not consider whether footprint searched by the tools were compatible or not. Actually, this is an extremely important information: suppose, for example, that presence of the trace searched by *ToolA* implies absence of trace searched by *ToolB*, then these two algorithms should never detect tampering simultaneously, and their being in disagreement would be a positive fact.

We can easily introduce this information in our framework by using a new belief assignment: starting from the previous formalization, we define a BBA on the domain $\Theta_A \times \Theta_B$, that assigns all the mass to the set containing the union of all propositions (i.e, combination of traces) that are considered possible, while all others have a null mass. For example the following BBA:

$$m_{comp}^{\Theta_A \times \Theta_B}(X) = \begin{cases} 1 \text{ for } X = \{(ta,nb) \cup (na,tb) \cup (na,nb)\} \\ 0 \text{ for } X = \{(ta,tb)\} \end{cases} \quad (5)$$

models the fact that traces searched by *ToolA* and *ToolB* can not be present at the same time, because proposition (*ta*, *tb*) has a null mass.

This latter BBA can be combined with the one coming from fusion of all the tools, and, thanks to Combination Rule's associativity and commutativity, we are free to make this combination only as a last step. This helps the modularity of the framework, since adding new tools will not require to revisit the whole system but only the last part. The scheme in Figure 2 gives an overall idea of the proposed decision fusion method.

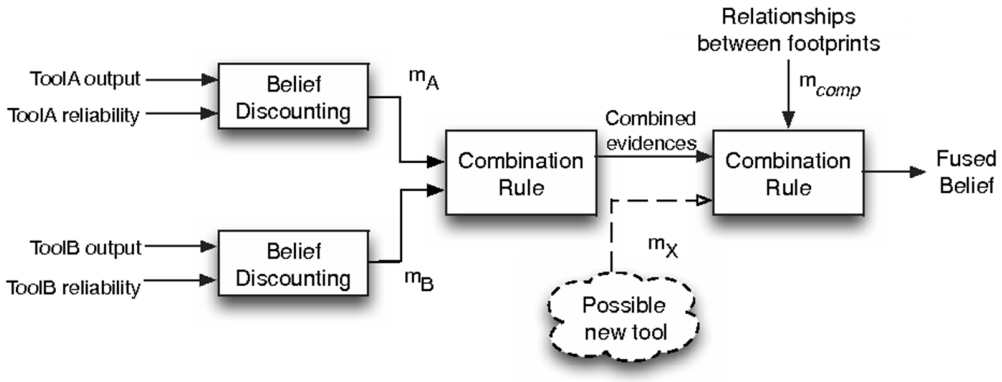


Figure 2: Overview of the employed decision fusion scheme. Notice that, being the Combination Rule commutative, traces relationships can be taken into account only at the end, thus facilitating scalability. In fact, if traces relationships were introduced in the first step, then adding a new tool would require to consider its impact on all the tools that were already present.

Belief for the presence of tampering

Once information coming from all tools and from their relationships have been fused, we can extract the final belief for the analysed region being forged or not. This is equivalent to asking the question: "does the analysed region expose some traces of forgery?". In our formulation, answering this question requires to sum the final mass assigned to propositions where at least one footprint has been detected. Since the only proposition supporting authenticity of the region is the one where none of the footprints is present (e.g., (*na*, *nb*) in previous formulas), answering the above question simply requires to add the mass of all propositions that do not include that proposition. If we consider, for example, equation (4), we should add the masses from lines 1, 2, 3, 4 and 7. The obtained quantity, that takes values in [0,1] is what DST calls *Belief* for our proposition "The region is forged" (Dempster, 1967), and coincides with the output of the decision fusion engine.

ARCHITECTURE OF THE PROPOSED TOOL

As a consequence of IF being a blooming discipline, novel techniques are constantly emerging, providing more reliable and informative analysis. Since it is our goal to design a tool that provides the forensic analyst with state of the art techniques, it is mandatory to keep its architecture scalable.

Furthermore, both due to user needs and tool capabilities, different modalities of analysis are to be foreseen: if the IFA has some suspects about a specific region of the image, he would prefer a focused analysis for that region of interest (ROI); on the other hand, when the analyst has not any kind of prior information, he would need a fully-automatic analysis. The use case diagram in Figure 3 outlines the actions that are made available to the IFA by the proposed tool, while Figure 4 shows a block diagram of the implemented functions.

If we focus on forensic algorithms capabilities, dual considerations are in order: algorithms that provide forgery localization could also be asked to provide the probability of a given region of being forged rather than to actually localize tampered regions; conversely, it may happen that algorithms that do not allow localization are to be adapted to perform this kind of task. In this case, the simplest approach is

to perform several analysis in a block-wise fashion, considering a small block as the suspect region at each execution: by putting the results together, a kind of localization is obtained.

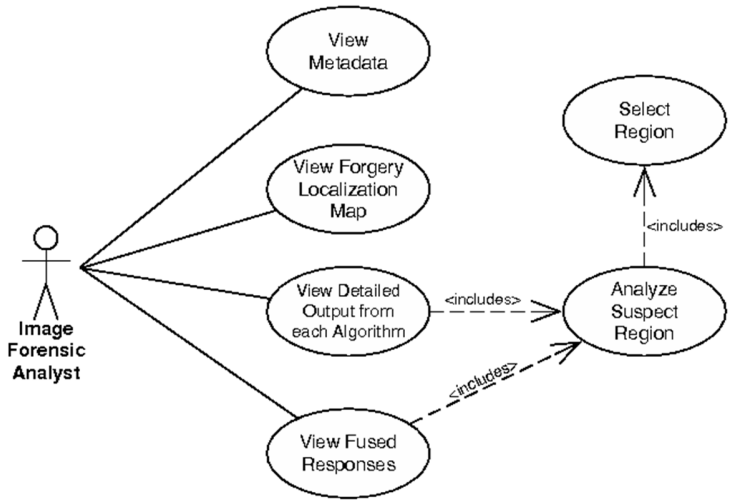


Figure 3. Use case diagram of the proposed tool. Minor functionalities, such saving/loading the current selection of ROIs, are not shown.

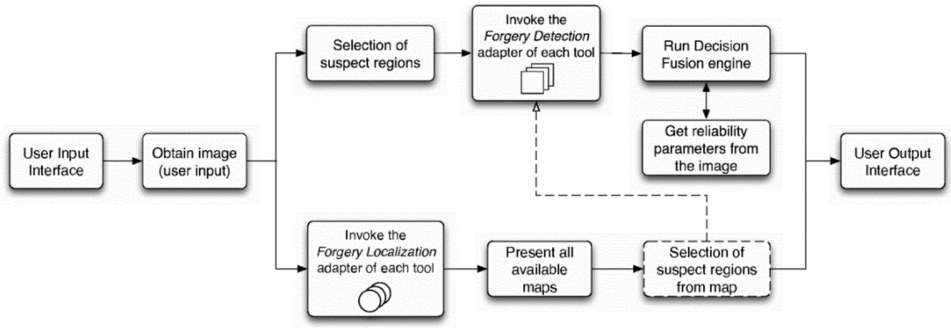


Figure 4. Block diagram for the proposed tool. Notice that two main streams of analysis are possible: the upper-stream allows performing forgery detection, while the bottom-stream allows doing forgery localization. Notice also that (optionally, as denoted by dashed lines) the analyst may first use the forgery localization functionality and then, by selecting suspect regions directly on a probability map, invoke the forgery detection as a second phase.

To cast all these considerations in, we must rely on an fully modular architecture, and this requirement reflects mainly on two aspects: software architecture and decision fusion module.

Software modularity

From the software point of view, it is essential to structure the tool in such a way that a single algorithm can expose its functionalities (i.e., forgery detection, forgery localization, or both) to the tool, abstracting from the underlying principle of analysis, and that new tools can be introduced with minimal effort.

Since a tool can be employed in different ways, we find it appropriate to separate between its core algorithm, which actually performs the analysis, and its interface, that invokes the core and performs the interpretation of the results. In such a way, the IFA simply selects the modality he prefers (forgery detection or localization), and the tool exploits accordingly the forensic algorithms.

From a software engineering point of view, we refer to the *adapter* pattern to model this scenario; this pattern allows to expose different (software) interfaces of the same algorithm for different needs (in our case, forgery localization or ROI integrity verification) while invoking the same core routine for image analysis. Specifically, the tasks in charge of the tool adapter are:

- to turn requests from client into appropriate invocations of underlying methods, running the core analysis several times if needed;
- to interpret the output of the tool, in order to make it compliant with the implemented interface (map is expected, scalar value is expected, ...)

Besides, since adapters share the same interface (in terms of input and output parameters), forensic algorithms are kept separated from the container tool: algorithms are simply integrated into the tool by adding a reference to their adapter into a list, that is maintained by the tool. This allows introducing new algorithms by updating the list.

Modularity of the decision fusion method

After analyzing the data, the system uses the previously described decision fusion engine to fuse algorithm outputs. Notice that, since the engine exploits knowledge about relationships between footprints (that is a kind of cross-algorithm information), the overall modularity of the system could be potentially undermined by this phase of the analysis. This motivates our choice of the described decision fusion framework, that has been thought to be scalable: it is not based on machine learning, so introducing new tools does not require to re-train the system; relationships between tools are written as a Basic Belief Assignment that, due to the commutative and associative properties of Dempster's combination rule, can be introduced only in the last step of the fusion. As a result the impact of adding a new algorithm is limited to updating the Basic Belief Assignment that models the relationships between tools and fusing the old information with the new one, without needing to recompute everything from scratch. It is important to notice that this operation is done off-line and *only one time*, producing a formula for the final belief that is simply made of products and sums.

Output representation

The last step of the analysis is the presentation of the output to the user. The user is allowed to choose among different levels of detail: in the case of known suspect region, he can simply consider a binary classification of each ROI as tampered vs. original, or he can take a deeper look at the output generated by each single tool for each analysed ROI, and access information about tool reliability. Although not having been implemented yet, similar considerations hold for the forgery localization case: the user should be able to view either the map returned by each algorithm separately or the one resulting from the application of map-level decision fusion.

Complexity of the Tool

From an algorithmic point of view, the complexity of the proposed tool is determined by the complexity of underlying forensic algorithms. Generally, the complexity of these algorithms grows

linearly with the number of pixels of the image when the analysis is performed in the spatial domain, while the complexity is at least $O(n \log n)$ when the DCT domain is considered.

The complexity induced by the decision fusion engine is negligible at execution time, since it reduces to evaluating a linear formula (like equation (4)) involving outputs from algorithms. The heavy part of the decision fusion engine, that is deriving that formula, can be executed off-line, and needs to be re-computed only when new algorithms are added to the tool. Table 1 shows the time that is needed for both the off-line and on-line phases for an increasing number of algorithms: notice that the on-line phase is very fast, and it is practically constant along the rows.

Number of tools	Off-line Time (seconds)	On-line Time (seconds)
2	0.05	0.033
3	0.15	0.033
4	0.41	0.034
5	1.05	0.036

Table 1: execution time for the off-line phase (preparation of the decision fusion formula) and for the on-line phase (evaluation of the formula during tool execution). The results have been obtained with a Matlab implementation of the tool.

PRACTICAL APPLICATIONS AND CASE STUDIES

In this section we describe a practical implementation of the proposed tool, which makes use of state of the art algorithms for splicing detection in JPEG images, and provide experimental results along with a case-study.

Chosen set of tools

Since a great deal of digital images are stored in JPEG format, we focused the first implementation of the proposed tool on JPEG-based splicing detection. To this end, we selected five state of the art forgery detection algorithms among those described in the Section on Tools for Image Forensics:

1. the algorithm by Farid (2009) based on JPEG-ghost (that will be termed JPGH for brevity from now on);
2. the tool by Bianchi & Piva (2012) for detecting aligned double JPEG artifacts (termed JPDQ);
3. the tool by Lin (2009), still searching for aligned double quantization, termed JPLC;
4. the tool described in (Bianchi, 2012a) for detecting non-aligned double JPEG compression, termed JPNA ;
5. the algorithm proposed by Luo (2007), still based on non-aligned double JPEG artifacts, termed JPBM.

Among these algorithms, only JPBM leverages on machine learning techniques. We trained that algorithm following indications suggested by its authors in (Luo 2007), with the only difference that we employed a SVM providing probability estimates.

Considering the traces they look for, the selected algorithms are in some sense complementary, and their synergistic use is essential to achieve good performance in forgery detection. Notice that the only algorithms providing forgery localization among those selected are JPLC and JPDQ, with the latter being more recent and reliable. These two algorithms can also be used to perform forgery detection on suspect ROIs. Due to how the decision fusion framework is defined, the output provided by forensic algorithms must be interpreted and turned into BBAs for each one of the possible propositions about presence or absence of the trace in the suspect region. This is done by interpreting, according to published experimental results, the output value of each tool in terms of presence/absence of the searched trace. Details for each tool follow:

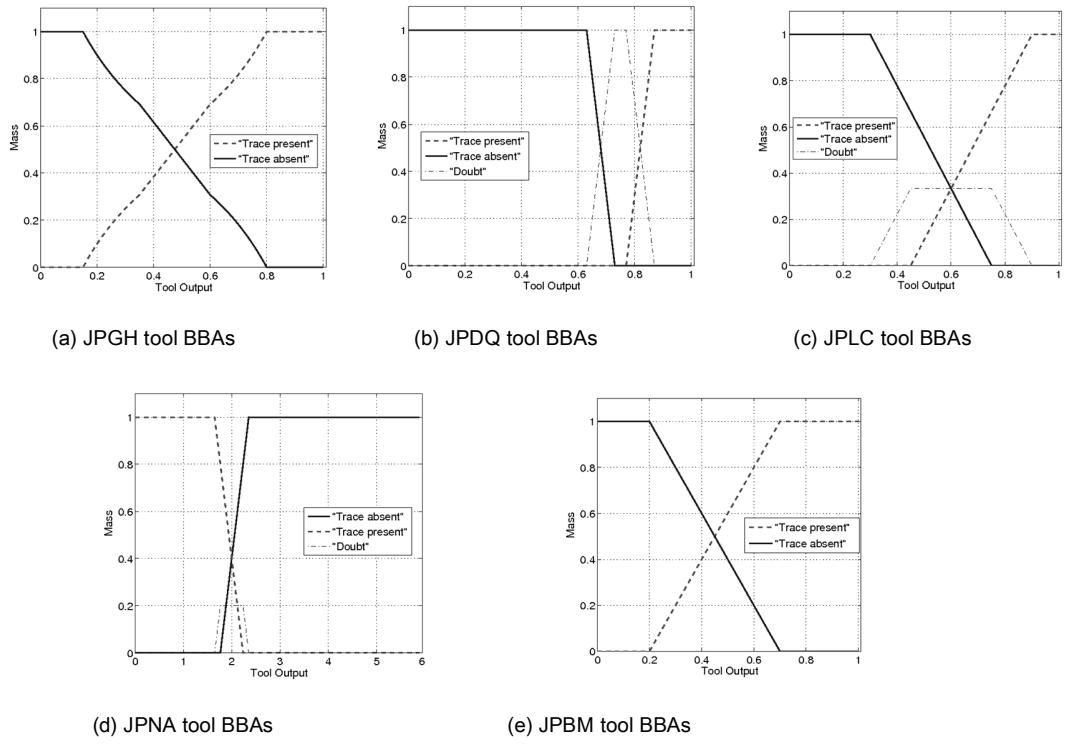
- JPGH: the value of Kolmogorov-Smirnov statistic, that is already in the interval [0,1], is turned into a basic belief assignment (BBA) with the mapping shown in Figure 5 (a);
- JPDQ: the median value of the probability map calculated over the selected ROI is converted into BBAs according to Figure 5 (b);
- JPLC: also here the median value of the map is considered and converted according to Fig. 5 (c);
- JPNA: the extracted metric takes values in [0,6], where lower values means a higher confidence for the image being tampered. The mapping is therefore performed as in Figure 5 (d);
- JPBM: output of the soft-margin SVM is converted into BBAs according to curves in Fig. 5 (e).

Curves in Figure 5 can be generated in different ways and, since they depend on the specific algorithm, a general formula cannot be given; they should be rather considered as an input to the system, telling how outputs have to be interpreted in terms of belief assignments. As a matter of fact, a plausible way to obtain these curves is to:

1. run each algorithm on a set of tampered and original images tailored for it (e.g., generated according to experiments described in the scientific work presenting the algorithm);
2. take the histogram (e.g. with bins in 0, 0.05, … 1) of outputs, separately for positive and negative samples;
3. fit trapezoidal curves to the hull of the two histograms, or use another kind of fitting.

This method is also suggested in (Fontani, 2011); the choice of trapezoidal curves imposes a kind of smoothness constraint that removes noisy measurements. Methods to automatically learn the mapping from algorithm outputs to BBAs will be the object of future work.

Notice that the same algorithm-specific datasets mentioned above can be used to analyse the reliability of algorithms (e.g., evaluating the overall accuracy on the dataset). As shown in Table 2, we followed this approach for tool JPLC, while we directly used results presented in scientific papers for other algorithms, since they were coherent with experimental data.



(a) JPGH tool BBAs (b) JPDQ tool BBAs (c) JPLC tool BBAs

(d) JPNA tool BBAs (e) JPBM tool BBAs

Figure 5. Mappings from algorithm output to Basic Belief Assignments. Each line maps the output of the algorithm (on the x-axis) to a mass assignment (y-axis). In each figure, the solid line is the mass assigned to proposition ``trace is not present'', the dashed line is the mass assigned to proposition ``trace is present'' and the gray dashed line, when present, gives the mass assigned to doubt.

Tool	**Reliability**
JPGH	R = 0.85, according to values in (Farid, 2009)
JPDQ	R = f(Q_2), *f* according to tables in (Bianchi, 2012b)
JPLC	R = 0.4 (estimated experimentally)
JPNA	R = f(Q_2), *f* according to tables in (Bianchi, 2012a)
JPBM	R = f(Q_2), *f* according to tables in (Luo 2007)

Table 2. Reliability for various algorithms; Q_2 denotes the quality factor of the JPEG image, that can be easily estimated from the JPEG quantization table present in the header file.

Implementation details

The proposed system has been implemented using Matlab. We chose Matlab because of its versatility, and because it provides optimized implementation of many common image processing operations that are intensively used by forensic algorithms. As a matter of fact, the simple layout of the proposed architecture does not suffer from using such a language. We also made use of the GUIDE toolkit to develop the Graphical User Interface.

Concerning single algorithms, in the current implementation of the system we do not provide JPGH and JPNA with localization capabilities, but we plan to do so in a future work. The DST based decision fusion engine is implemented using features of the Matlab Symbolic Toolbox, and is provided with functions helping the developer in updating it when new tools can be added. The proposed implementation makes use of the Matlab JPEG Toolbox to access JPEG coefficients and quantization tables when analysing images.; note that this toolbox makes, in turn, use of the Independent JPEG Group (IJG) JPEG code library.

As to time and memory usage, the computing time and the allocated memory depend on the size of images, being dominated by the resource consumption of forensic algorithms. Table 3 reports memory usage and execution times obtained on a standard desktop computer (2GHz CPU, 4GB RAM) using the Matlab implementation of the tool. Times are evaluated on three sets of images of different size, each set containing 10 images. We see that time and memory usage are acceptable for common-sized images, and that the overhead due to decision fusion is negligible with respect to the time needed by algorithms to complete the analysis.

	Localization		**Detection**		**Decision Fusion**	
Image size	Time	Memory	Time	Memory	Time	Memory
660x440	0.54	~20 MB	2.76 s	~87 MB	0.03 s	< 1 MB
1024x768	1.34	~70 MB	6.09 s	~170 MB	0.03 s	< 1 MB
1920x1200	1.69	~160 MB	18.24 s	~420 MB	0.03 s	< 1 MB

Table 3. Execution times and memory usage for running forgery localization (JPDQ algorithm), forgery detection (5 algorithms) and decision fusion using the Matlab implementation of the tool.

Case study and Experimental results

We now turn to evaluate tool simplicity and usefulness. For the first aspect, we analyse a case-study image to illustrate the analysis process for the IFA; for the latter aspect, we show that using the tool is better than using single algorithms. This is done by studying the detection performance obtained with and without using the decision fusion module on a dataset of both original images and hand-made realistic forgeries.

A sample case study

Let us play the role of the IFA, and suppose that the image in Figure 6 must be analysed. Considering the content of the image, we may reasonably think that, if a splicing is present, it may involve one or both the faces. Therefore, we want to check the integrity of those regions. Clicking on the "Add ROI" button, we can freely draw a polygon around, thus defining the desired ROIs; otherwise if we have no other suspects about the image, we may want to use the localization capabilities of the JPDQ algorithm. This is done by clicking on "Compute Probability Map" button, yielding the result in Figure 7: we see that our suspects about the face of the boy are confirmed, while the girl's face seems untouched.

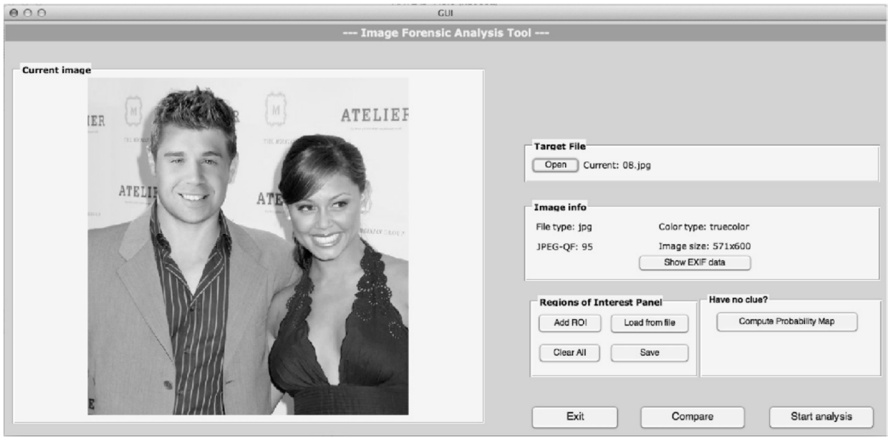


Figure 6. Case-study image.

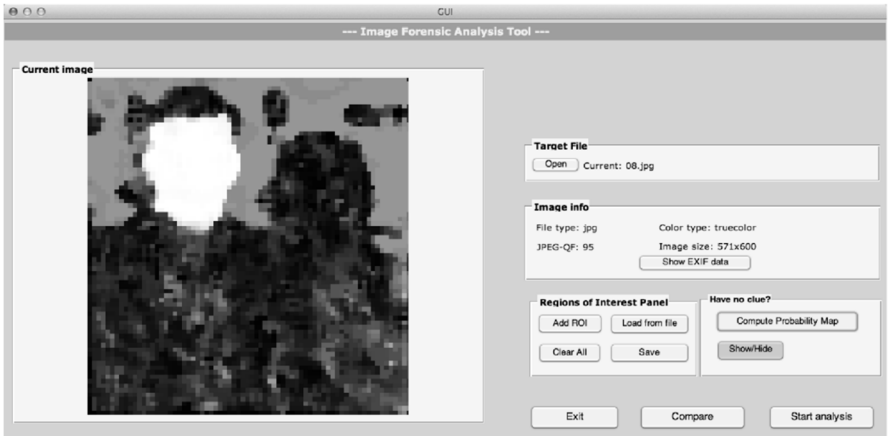


Figure 7. Viewing probability maps obtained with JPDQ localization algorithm (brighter colors denote higher probability).

However, we may choose to analyse both faces, selected manually with "Add ROI", to examine the joint output of all the available forensic algorithms. Clicking on "Start Analysis" we allow the tool to run them, fuse their answers and open the output interface (Figure 8).

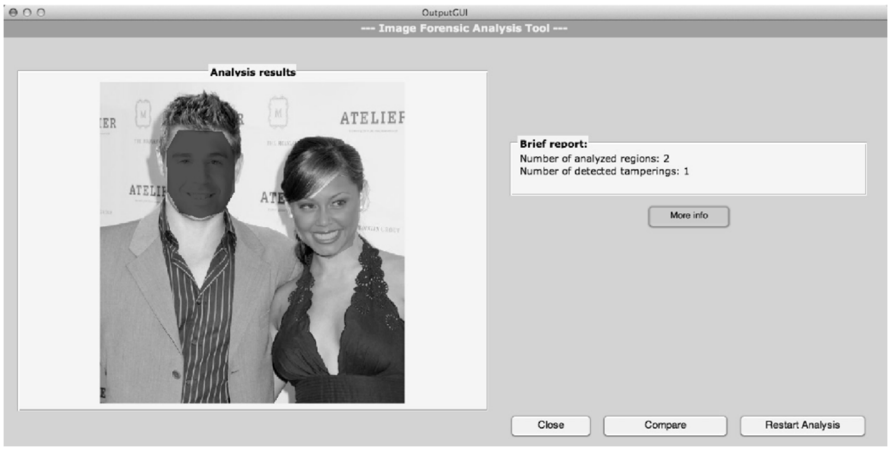


Figure 8. Coarse-details output.

Figure 9. Detailed results for boy's face. Percentual values for tampering in the table refer to the tool outputs scaled to be in [0,1].

Figure 10. Detailed results for girl's face.

We see immediately that only the boy's face is classified as tampered. Clicking on "More details", and selecting that face by clicking into the ROI, we see the table with results coming from all algorithms (Figure 9 for boy's face, Figure 10 for girl's face). Notice that for boy's face one of the algorithms does not detect any trace of tampering on that region: this is perfectly reasonable because, as said, a splicing will not necessarily leave all the possible traces. Since this fact is known to the decision fusion framework, the final belief for the ROI being tampered remains very high. Notice that in some cases the output of some tools can be shown as 0.0%: this is due to the fact that some forensic tools may output values very close to 0 (even in the order of 10^{-16}), that are truncated for presentation clarity.

Detection performance

In this paragraph we show that using the proposed tool performances are significantly increased with respect to running separately each algorithm. To do so, we analyse a set of images, and compare performance of decision fusion output with respect to those of single algorithms. We use a dataset of 83 realistic splicings of various kind and 83 original images (the dataset is publicly available at this website http://clem.dii.unisi.it/~vipp/files/datasets/HANDMADE_FORGERIES.zip); tampered images have been created by students using common photo editing software, respecting only some constraints on the JPEG quality factor used for saving the final forgery (quality less than 7/10 was forbidden). Of course, ground truth is available for each sample. Each test consists in: selecting the tampered region (or, for original images, some object); running the analysis on each region; considering the output of each algorithm separately, and the output of the decision fusion module.

We iteratively threshold these scalar output values, obtaining the Receiver Operating Characteristic (ROC) curves for the five algorithms and for the decision fusion method (Figure 11). A ROC gives the probability of correctly classifying a tampered image (detection probability) as a function of the probability of classifying an original image as tampered (false alarm probability). Therefore, an ideal detector has a "square" ROC curve, that lays on the y-axis and then on the line y=1.

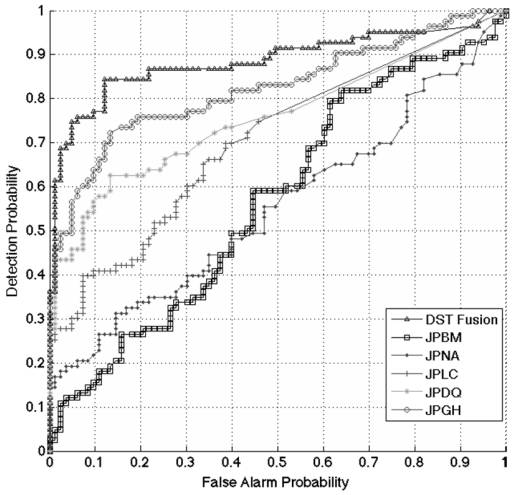


Figure 11. This figure shows, for the described dataset, the performance of single algorithms and that obtained using the decision fusion method.

Each line associated to an algorithm in Figure 11 is obtained running that algorithm on all the images of the dataset. The reader will probably be surprised by the poor performance obtained by separated algorithms, but they are perfectly reasonable: as a matter of fact, when a forensic algorithm is developed and evaluated in literature, tests are run on images that are automatically tampered *in a way that is detectable by the algorithm*. Although being useful to evaluate the discriminative power of a specific footprint, this approach ignores that a real analyst does not know in advance which kind of tampering could have been performed with the image under analysis. Furthermore, analyzing hand-made realistic forgeries is much harder than analyzing artificially generated splicings.

Notice that, as shown in (Fontani, 2011; Fontani, 2013), using the decision fusion module, a significant improvement is obtained in terms of performance, so the proposed tool proves to be not only useful in making forensics algorithms accessible to the IFA, but also in increasing his confidence in the analysis.

It is of interest to investigate how the proposed tool works when only a subset of the forensic algorithms are employed. Table 4 shows the behavior of the fusion system for various configurations, by plotting the integral of the ROC curve (also called Area Under Curve, AUC). An AUC equal to 1 characterizes a perfect classifier, while a random classifier would obtain 0.5. Noticeably, despite the contribution of some algorithms (e.g., JPBM and JPNA) being quite limited on the considered dataset, the use of 5 algorithms yields the best performance. This experiment shows that the proposed tool is able to "pick the best" from available forensic algorithms, even when their contribution is limited.

Enabled Algorithms					AUC
JPBM	**JPLC**	**JPGH**	**JPNA**	**JPDQ**	
					0.500
●			●		0.583
	●			●	0.719
			●	●	0.732
	●	●			0.848
	●	●		●	0.884
●	●	●		●	0.871
	●	●	●	●	0.893
●	●	●	●	●	0.894

Table 4. Performance of the decision fusion system for different subsets of the algorithms, obtained on the same dataset used in previous experiments. Each row indicates which algorithms were enabled (black dot) and which were not (empty cell). The obtained AUC is reported in the rightmost column.

Finally we selected some samples from the dataset to show the "path" followed by the proposed tool, that goes from the output of single algorithms to a final belief for presence of tampering. To do that we refer to Table 5, and consider 4 different images in the dataset (two forged and two untouched). The first row shows an example where all algorithms agree about absence of their respective footprints, and the final belief for presence of tampering is very small. The second row shows an interesting sample where one of the algorithms, namely JPBM, is confused about the presence of the footprint: thanks to the knowledge about tool reliability and compatibility of algorithms, the final belief is kept small. Similar comments apply to the last two rows, referring to forged images: notice that in the third row we have 3 algorithms out of 5 that are more convinced about the absence of the searched trace. Notwithstanding that, the tool leverages on the information provided by JPGH and JPDQ to reach a high belief for presence of the forgery.

Ground Truth	**JPBM**			**JPLC**			**JPGH**			**JPNA**			**JPDQ**			**Fused Belief of Tampering**
	Out	m_T	m_N	Out	m_T	m_N	Out	m_T	m_N	Out	m_T	m_N	Out	m_T	m_N	
Untouched	0.26	0.13	0.87	10^{-9}	0	1	0.11	0	1	0.01	0	1	0.00	0	1	0.01
Untouched	0.53	0.65	0.35	10^{-5}	0	1	0.17	0.04	0.96	0.00	0	1	0.03	0	1	0.13
Tampered	0.35	0.31	0.69	10^{-6}	0	1	0.55	0.61	0.39	0.14	0	1	0.44	1	0	0.86
Tampered	0.74	1	0	0.99	1	0	0.35	0.31	0.69	0.78	1	0	0.98	1	0	0.99

Table 5:each row of the table shows, for every algorithm, the output and the corresponding mapping to belief for presence (m_T) and absence (m_N) of the footprint. The rightmost column shows the final (fused) belief for the analysed region being tampered.

CONCLUSIONS

Image integrity assessment is a key task in nowadays communications. In this paper, we presented a tool for detecting splicing in digital images. The proposed system allows the IFA to exploit the capabilities of existing forensic algorithms, and moreover provides him with a decision fusion framework that automatically interprets their outputs, increasing detection performance. Furthermore, the tool can exploit localization capabilities of forensic algorithms to show a probability map to the IFA, helping him to identify suspect regions.

We proposed a modular architecture that makes this tool easily extendible with new forensic algorithms. Also, the employed decision fusion framework is easy to extend (Fontani, 2011; Fontani, 2013) and does not require any form of training apart from that needed for each forensic algorithm alone, since it exploits theoretical knowledge about relationships between traces searched by tools.

As future work, we plan to develop decision fusion strategies for forgery localization, so to allow the IFA to merge probability maps obtained with different tools. We also plan to add novel functionalities to the tool, like automatic image (and probability map) segmentation, and to transcode the current Matlab implementation to a more portable and fast programming language.

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, and by the European Office of Aerospace Research and Development under Grant FA8655-12-1-2138: AMULET - A multi-clue approach to image forensics.

REFERENCES

Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., & Serra, G. (2011). A SIFT-based forensic method for copy-move attack detection and transformation recovery. *IEEE Transactions on Information Forensics and Security*, 6 (3), 1099-1110.

Barni, M., Costanzo, A., & Sabatini, L. (2010). Identification of cut & paste tampering by means of double-JPEG detection and image segmentation. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS2010)* (pp. 1687-1690). IEEE.

Bianchi, T., De Rosa, A, & Piva, A. (2011). Improved DCT Coefficient Analysis For Forgery Localization In JPEG Images. In *Proceedings of the 2011 International Conference on Acoustics, Speech, and Signal Processing (ICASSP2011)* (pp. 2444-2447). IEEE.

Bianchi, T., & Piva, A. (2012a). Detection of nonaligned double JPEG compression based on integer periodicity maps. *IEEE Transactions on Information Forensics and Security*, 7 (2), 842-848.

Bianchi, T., & Piva, A. (2012b). Image forgery localization via block-grained analysis of JPEG artifacts. *IEEE Transactions on Information Forensics and Security*, 7 (3), 1003-1017.

Chen, M., Fridrich, J., Goljan, M., & Lukáš, J. (2008). Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security*, 3 (1), 74-90.

Choi, K. S., Lam, E. Y., & Wong, K. K. Y. (2006). Automatic source camera identification using the intrinsic lens radial distortion. *Optics Express*, 14 (24), 11551-11565.

Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38, 325-339.

Dirik, A. E., Sencar, H. T., & Memon, N. D. (2008). Digital single lens reflex camera identification from traces of sensor dust. *IEEE Transactions on Information Forensics and Security*, 3 (3), 539-552.

Farid, H. (2009). Exposing digital forgeries from JPEG ghosts. *IEEE Transactions on Information Forensics and Security*, 4 (1), 154-160.

Ferrara, P., Bianchi, T., De Rosa, & A., Piva, A. (2012). Image Forgery Localization via Fine-Grained Analysis of CFA Artifacts. IEEE Transactions on Information Forensics and Security, 7 (5), 1566-1577.

Fontani, M., Bianchi, T., De Rosa, A., Piva, A., & Barni, M. (2011). A Dempster-Shafer framework for decision fusion in image forensics. In *Proceedings of 2011 IEEE International Workshop on Information Forensics and Security (WIFS2011)* (pp. 1-6) . IEEE.

Fontani, M., Bianchi, T., De Rosa, A., Piva, A., & Barni, M. (2013). A Framework for Decision Fusion in Image Forensics Based on Dempster–Shafer Theory of Evidence. *IEEE Transactions on Information Forensics and Security*, 8 (4), 593-607.

Fridrich, J., Soukal, B. D., & Lukáš, J. (2003). Detection of copy-move forgery in digital images. In *Proceedings of Digital Forensic Research Workshop*. DFRWS.

Gallagher, A., & Chen, T. (2008). Image authentication by detecting traces of demosaicing. In *Proceedings of Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08.*, (pp. 1-8). IEEE.

Gou, H., Swaminathan, A., & Wu, M. (2009). Intrinsic sensor noise features for forensic analysis on scanners and scanned images," *IEEE Transactions on Information Forensics and Security*, 4 (3), 476-491.

Kee, E., & Farid, H. (2010). Exposing digital forgeries from 3-D lighting environments. In *Proceedings of 2010 IEEE International Workshop on Information Forensics and Security (WIFS2010)* (pp. 1-6) . IEEE.

Kirchner, M., & Fridrich, J. J. (2010). On detection of median filtering in digital images. In Memon N. D., Dittmann, J., Alattar, A. M., & Delp, E. J. (Eds.), Media Forensics and Security II, SPIE Proceedings Vol. 7541 (754110). SPIE.

Li, B., Shi, Y. Q., & Huang, J. (2008). Detecting doubly compressed JPEG images by using mode based first digit features. In *Proc. of the 2008 International Workshop on Multimedia Signal Processing (MMSP 2008)* (pp. 730-735). IEEE.

Lin, Z., He, J., Tang, X., & Tang, C.-K. (2009). Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis. *Pattern Recognition*, 42, (11), 2492-2501.

Luo, W., Qu, Z., Huang, J., & Qiu, G. (2007). A novel method for detecting cropped and recompressed image block. In *Proceedings of the 2007 International Conference on Acoustics, Speech, and Signal Processing (ICASSP2007), Vol 2* (pp. 217-220). IEEE.

Mahdian, B., & Saic, S. (2008). Blind authentication using periodic properties of Interpolation. *IEEE Transactions on Information Forensics and Security*, 3 (3), 529-538.

Popescu, A., & Farid, H. (2005a). Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing*, 53 (2), 758-767.

Piva, A. (2013). An Overview on Image Forensics. *ISRN Signal Processing*, vol. 2013, Article ID 496701, 22 pages.

Popescu, A. C., & Farid, H. (2004). Statistical tools for digital forensics. In J. J. Fridrich (Ed.), *Information Hiding, Lecture Notes in Computer Science* Vol. 3200 (128-147). Springer.

Popescu, A., & Farid, H. (2005b). Exposing digital forgeries in color filter array interpolated Images. *IEEE Transactions on Signal Processing*, 53 (10), 3948-3959.

Swaminathan, A., Wu, M., & Liu, K. J. R. (2008). Digital image forensics via intrinsic fingerprints. *IEEE Transactions on Information Forensics and Security*, 3 (1), 101-117.

Tang, Z., Wang, S., Zhang, X., Wei, W., & Su, S. (2008). Robust image hashing for tamper detection using non-negative matrix factorization. *Journal of Ubiquitous Convergence Technology*, 2 (1), 1-18.