2014 International Conference on Network-Based Information Systems

A Possible Pitfall in the Experimental Analysis of Tampering Detection Algorithms

Giuseppe Cattaneo, Gianluca Roscigno
Dipartimento di Informatica
Università degli Studi di Salerno
Via Giovanni Paolo II, 132, I-84084 Fisciano (SA), Italy
Email: {cattaneo,giroscigno}@unisa.it

***Abstract*—This paper aims to give a contribute to the experimental evaluation of tampered image detection algorithms (i.e. Image Integrity algorithms), by describing a possible way to improve these experimentations with respect to the traditional approaches followed in this area. In particular, the paper focuses on the problem of choosing a proper test dataset allowing to keep low the bias on the experimental performance of these kind of algorithms. The paper first describes a JPEG image integrity algorithm, the Lin *et al.* algorithm [1], that has been used as benchmark during our experiments. Then, the experimental performance of this algorithm are presented and discussed. These performance have been measured by running it on the `CASIA TIDE` public dataset, which represents the *de facto* standard for the experimental evaluation of image integrity algorithms. The considered algorithm apparently performs very well on this dataset. However, a closer analysis reveals the existence of some statistical artifacts in the dataset that improve the performance of the algorithm. In order to confirm this observation, we assembled an alternative dataset. This new dataset has been conceived to not exhibit the statistical artifacts existing in the images of the `CASIA TIDE` dataset, while producing an uniform distribution of some physical image features such as the quality factor. Then, we repeated the same experiments conducted on the `CASIA TIDE` dataset, using this new dataset. As expected, we observed a performance degradation of the Lin *et al.* algorithm, thus confirming our hypotheses about the `CASIA TIDE` dataset being, in some way, flawed.**

***Keywords*-Digital Image Forensics; JPEG Image Integrity; Evaluation Datasets; Experimental Analysis; Double Quantization Effect;**

I. Introduction

Nowadays, photos and videos accompany us in our daily and personal life. For instance, it has been estimated that about 2.5 billion mobile phones with embedded camera were in worldwide use [2] at the end of the 2009. From the man in the street viewpoint, photos and videos have become part of a sort of new communication language, that mixes together the spoken language with multimedia digital contents. This is witnessed by the enormous amount of digital images exchange through online social networks and photo sharing websites. For example, in September 2013, the total number of photo uploaded to Facebook was about 250 billion, while the average number of photos uploaded per user was about 217 images [3]. In such a scenario, it becomes often important to ensure that a digital image is authentic and has not been subject to any form of manipulation, especially in some application fields such as journalism, criminal investigations and legal matters. This risk is today higher than in the past, thanks to the flourishing of applications and online services for editing and tampering digital images.

In the recent years, a new discipline is born, called *Digital Image Forensics*. It is responsible for the acquisition and the analysis of images found on digital devices or on the Web for investigation purposes. This research field is very active, as witnessed by the many contributions proposed in this area (e.g., [4], [5], [6], [7], [8], [9], [10]).

In this field, a well-studied problem, often referred to as *Image Integrity* or *Tampered Image* detection problem, is the automatic detection of any manipulation carried on a digital photo after its acquisition. In other words, given an input image, is it possible to state if *it has been tampered by any kind of operation or it is exactly equivalent to the camera output?* An image is *tampered* if it is the result of the application of any technique able to alter its content, such as painting new objects in a portrayed scene or copying the region of an image over another image (*splicing* operation). Image integrity detection algorithms can be divided into two main categories: *active* and *passive*. Active algorithms are based on the embedding of a watermarking (either visible or not) in the image or on the calculation of a signature of the image. If the original image (called also genuine or authentic) is modified, the embedded watermark will result to be broken or the original image signature will not match anymore the same signature calculated on the modified image. Passive methods, instead, try to check the integrity of an image, without requiring any a priori information. In this work, we focus our interest on passive algorithms for checking the tampering of digital images.

A. Tampered JPEG Image Detection

Most of the contributions developed in this field are focused on the JPEG digital images data format, as it is the *de facto* standard for acquiring and exchanging digital images. The success of the JPEG format is mainly due to the possibility of achieving an optimal trade-off between the compression rate of an image and its resulting quality (according to a quality factor chosen by the user). The

978-1-4799-4224-4/14 $31.00 © 2014 IEEE
DOI 10.1109/NBiS.2014.82

quality factor (*QF*) of a JPEG image (see [11]), i.e. inverse of the compression rate, can vary in the range $[1; 100]$, where smaller values result in a lower quality of the compressed image and a higher compression degree. On the one hand, detecting whether a JPEG image is tampered or not can be harder than for other formats because the compression employed by this encoding may delete the forgery traces left in a photo. On the other hand, it is possible to design an algorithm that detects compression artifacts, and use them to track possible forgeries. The *JPEG blocking artifacts* introduced by JPEG compression can be considered as an inherent "watermark" for compressed images. When a JPEG image is tampered, these artifacts should be always altered by the forgery operations. Therefore, an image obtained by splicing a sub image read from a JPEG file with a quality factor over another image with a different quality factor is easily detectable as forged.

Many tampered image detection algorithms work using traces of artifacts produced from JPEG compression (see, e.g., [1], [6], [7]). Generally, these algorithms use some of the statistical properties of the JPEG *Discrete Cosine Transform* (DCT) coefficients to detect inconsistencies in the blocking artifacts of an image. One of the first contributions in this area is described in [12]. The authors estimate the primary quantization table from a doubly compressed JPEG image using the histograms of the individual DCT coefficients. A similar approach has been presented in [6] by Ye *et al.*. Here, the histogram of the DCT coefficients is used to estimate the quantization step size and, then, the algorithm measures the inconsistency of quantization errors between different image regions for estimating local compression blocking artifacts measure. Unfortunately, it requires a preliminary human intervention to select a suspicious region of the image to analyze. An alternative algorithm is proposed by Farid [7]. It detects *JPEG ghosts*, to establish whether a region of an image was originally compressed at quality factor different than others regions of the same image. The major disadvantage of this technique is that it only works when the tampered region has a lower quality than the surrounding image. Lin *et al.* presented in [1] a method for detecting and locating doubly compressed JPEG 8×8 blocks in an image. They examine the *double quantization* (DQ) effect contained in the JPEG DCT coefficients, computing the Block Posterior Probability Map (BPPM) using a Bayesian approach and, then, they use a SVM classifier trained through features extracted from BPPM.

B. Evaluating the Performance of Tampered Image Detection Algorithms

It is a common practice to validate tampered image detection algorithms, by using them to check the integrity of a reference set of digital images. This set is typically comprised of both authentic and tampered images. Thus, the aim of the experiment is to measure the number of images correctly identified as authentic or tampered, against the number of authentic images classified as tampered (false positives) and the number of tampered images classified as authentic (false negatives). Indeed, the usage of a standard reference dataset simplifies the comparison of different algorithms while fostering the development of more efficient solutions. Moreover, it becomes possible to evaluate the performance of an algorithm in a neutral way. It is important as well to assemble such a dataset using images that would not advantage a particular algorithmic approach but, instead, would be in some way representative of the typical images that are exchanged on the Internet, both in their authentic and tampered versions. Despite of this, many of the scientific contributions in this field proposed so far have been tested using each a custom, often self-made, dataset, typically made of a small number of pictures.

C. Our Contribution

The experimentation of tampered image detection algorithms has improved in the recent years because of the introduction of same reference image datasets that allows for a simpler and more effective comparison among different algorithmic approaches.

In a previous work [9], concerning the experimentation of the algorithm by Lin *et al.* [1] for the detection of tampered images, it has been noticed that the images belonging to a popular reference dataset and used for evaluating the algorithm exhibited some statistical artifacts that were able to simplify and to improve the detection process. The dataset used in that work is the `CASIA TIDE` public dataset [13], probably the most used dataset for these kind of experimentations. Such a behavior, if confirmed, would open to the possibility that the results of all the experimentations conducted so far using the `CASIA TIDE` are, in some way, inaccurate.

In this paper, we further analyze the performance of the Lin *et al.* algorithm when run on the `CASIA TIDE` dataset, by reviewing the way in which the statistical artifacts existing in the images of the dataset may influence the behavior of the algorithm. Then, we introduce an alternative dataset (called `UniSa TIDE`) which does not exhibit this kind of artifacts. Finally, we present the outcomings of an experiment conducted on the Lin *et al.* algorithm using our dataset and compare them against the same outcomings measured on the `CASIA TIDE` datasets. The results show that when using our dataset, the performance of the Lin *et al.* algorithm clearly are deteriorated, thus confirming the significant influence of the statistical artifacts existing in the images of the `CASIA TIDE` dataset on the detection process.

D. Organization of the paper

In Section II, we briefly discuss the algorithm by Lin *et al.*, to be used as a benchmark in our experiments. The

Section III describes the experiments we have conducted on the `CASIA TIDE` dataset and their results. In Section IV, we present an experimental dataset, i.e., the `UniSa TIDE` dataset, then we describe the outcomes of a round of experiments conducted on this new dataset and we compare them with the results showed in Section III. Finally, in Section V we draw some concluding remarks for our work.

II. Our Benchmark: Lin *et al.* Algorithm

During our experiments on the datasets used for testing image integrity algorithms, we have chosen as a reference algorithm the algorithm by Lin *et al.* [1]. It detects tampered JPEG images by examining the *double quantization* (DQ) effect contained in the JPEG *Discrete Cosine Transform* (DCT) coefficients. This effect occurs when the DCT coefficients histogram of a JPEG image has periodically missing values or some periodic pattern of peaks and valleys. Lin *et al.* states that the 8×8 JPEG image blocks that do not exhibit the DQ effect are probably tampered, in fact, in a non-genuine (tampered) image, unmodified blocks will exhibit the DQ effect, while tampered blocks (also called *doctored* blocks) will not.

In a few words, the Lin *et al.* algorithm works as follows:

1) As a preliminary step, if the input image is not a JPEG image, it is converted to this format using the highest quality factor ($QF = 100$).
2) The algorithm extracts from the input image the DCT coefficients and the quantization tables for each of the three YCbCr color channels.
3) The algorithm builds one DCT coefficients histogram for each of the three YCbCr color channels, and for each of the 64 frequencies of the Discrete Cosine Transform.
4) These histograms are used for determining a probability value which indicates if a particular 8×8 block of the input channel image is tampered, by checking the DQ effect.
5) Tampered blocks probabilities are combined together to produce the *Block Posterior Probability Map* (BPPM).
6) The BPPM is thresholded to distinguish between (probably) tampered blocks (C_0) and genuine (authentic) blocks (C_1).
7) A four-dimensional vector is extracted for each of the three YCbCr color channels, using the following features:
   - The sum of the variances of the probabilities in C_0 and C_1.
   - The squared difference between the mean probabilities of C_0 and C_1.
   - A threshold T_{opt} that maximizes the ratio between the second feature and the first feature. When using T_{opt}, we expect that the blocks in the class C_0 (i.e. those that have lower probability of T_{opt}) correspond to the tampered blocks.
   - A measure K_0 of the connectivity of C_0 blocks (more is connected C_0, then smaller is K_0).
8) Finally, a trained Support Vector Machine (SVM) dichotomous classifier is run to decide, starting from the previously extracted features, whether the image is doctored or not. In fact, we need training classifier before deciding if an input image is tampered or not.

A. Our Implementation

We developed a Java-based implementation of Lin *et al.* algorithm called `DQD`[1] that includes two core modules:

- The **Feature Extractor** module. It is in charge of extracting the features used for the classification from a batch of input images. We use only the first 32 lower frequencies for building DCT coefficients histograms.
- The **SVM Manager** module. It is in charge of managing the SVM classifier to be used for detecting tampered images or not. The SVM implementation is available with the Java Machine Learning library [14], adopting the *LIBSVM* module. To get more accurate decisions, our implementation uses the *Cross Validation* and the *Grid Search* techniques.

For pursuing our goal, it is important to train the SVM classifier using an image set (called *training image set*) and validate it adopting another image set (called *testing image set*). The validation returns the *recognition rate* (*RR*), i.e. the percentage of testing images correctly recognized as tampered or not.

III. The First Experimental Stage

In a first experimental stage, we analyzed the performance of the Lin *et al.* algorithm when run on the `CASIA TIDE` dataset. Our goal was to repeat the experiments presented in [9] while, at the same time, gathering additional information about the behavior of the considered algorithm, useful to explain the possible pitfalls existing in the considered dataset.

A. The `CASIA TIDE` Dataset

The many contributions proposed so far about the detection of tampered images motivated the development of public reference image dataset to be used for experimenting these contributions. If we restrict our interest to JPEG-based datasets, there is one dataset that has become the *de facto* standard for these experimentations: the `CASIA TIDE` dataset [13]. This dataset is available in two versions. The first version is a low resolution image small set, while the second version (`CASIA TIDE` v2.0[2]) is a large-scale

[1] A copy of the source code of our implementation of Lin *et al.* algorithm is available upon request.

[2] Credits for the use of the CASIA Tampered Image Detection Evaluation Database (`CASIA TIDE`) v2.0 are given to the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science, Corel Image Database and the photographers. http://forensics.idealtest.org

dataset containing $7,491$ authentic and $5,123$ tampered color images. The photos have different resolutions, varying from 240×160 to 900×600 pixels, and different file formats (i.e., JPEG, BMP, TIFF). In this version, tampered images are the result of splicing operations that have been optionally followed by a *blur filter* over the image.

In the rest of this paper, we will refer to the second version of this dataset, as it contains a larger number of images, tampered using more sophisticated techniques. As we anticipated before, this dataset has been widely used by many contributions in this field, such as [15], [16], [17], [18].

For our tests, we extracted a subset of images contained in the CASIA TIDE dataset, called $CASIA_{SC\_ALL}$, containing $2,000$ random training photos divided into $1,000$ authentic (990 JPEG and 10 BMP) and $1,000$ tampered (367 JPEG and 633 TIFF). The remaining $6,491$ authentic ($6,447$ JPEG and 44 BMP) and $3,875$ tampered images ($1,449$ JPEG and $2,426$ TIFF) were used for SVM testing. We have included also non-JPEG images because we treat them as JPEG image with maximum quality factor. Moreover we have excluded some tampered images because they originated by non-JPEG images and, thus, they could not exhibit the DQ effect on which relies the Lin *et al.* algorithm. In addition, we have used a second dataset, called $CASIA_{SC\_JPEG}$, obtained by filtering only the JPEG images of $CASIA_{SC\_ALL}$. A third dataset CASIA TIDE-derived, called $CASIA_{DC\_ALL}$, was obtained including all images of $CASIA_{SC\_ALL}$ and performing a JPEG recompression on authentic images using a quality factor randomly chosen in the set {100, 99, 95, 90, 85, 80, 75, 70}. Finally, $CASIA_{DC\_JPEG}$ dataset was obtained by filtering only the JPEG images existing in $CASIA_{DC\_ALL}$.

B. CASIA TIDE *Preliminary Experimentation*

In a preliminary experiment, we measured the recognition rate of the Lin *et al.* algorithm when run on the CASIA TIDE-derived datasets. Here, the results show a good performance of the algorithm, as it has been able to correctly recognize as original or tampered $\approx 70\%$ of the images belonging to the $CASIA_{SC\_ALL}$ and $CASIA_{DC\_ALL}$ datasets. This percentage increases by a $\approx 15\%$ when considering the variants of these datasets including only JPEG images. Table I shows the percentage of testing images correctly recognized as authentic or tampered (*RR*) when we ran DQD on the datasets defined in Section III-A. These results are in line with those presented in [9] and they confirm the ability of the algorithm to discriminate between tampered images or not, by looking at the DQ effect, also when run on authentic images compressed only once and differently from the experiments performed by Lin *et al.* in [1] (we recall that the algorithm has been designed for dealing with doubly compressed JPEG images). Moreover, the performance of the algorithm were still good even when run on non-JPEG images. JPEG datasets are more able to retain some statistical artifacts useful for the classifier to detect tampered images. Here there is an improvement of the performance of the algorithm on doubly compressed images while the inclusion of non-JPEG images ($CASIA_{DC\_ALL}$) leads to a small degradation of the overall performance of the algorithm.

Starting from these good results, we analyzed the internal structure of the images of the CASIA TIDE v2.0 dataset to determine if there were some properties that could, in some way, influence the detection process. One important property we noticed is that JPEG tampered images belonging to the $CASIA_{SC\_ALL}$ dataset are always saved using few fixed quality factors, whereas their non-tampered counterparts are saved using different quality factors. We conducted this analysis on compression rates adopting an estimation of luminance quality factor starting by the luminance quantization table embedded in each JPEG file. In particular, we applied an estimate of the inverse formula of [11] that, given a JPEG image quantization table and the standard quantization table, returns an estimation of the quality factor used to determine the image quantization table. Fig. 1 and Fig. 2 show the distributions of luminance-estimated image quality factors over authentic and tampered testing subset in $CASIA_{SC\_ALL}$ image set (for clarity $QF \in [84; 100]$, in fact, there are few images out of this scale). Here, in the authentic testing and tampered testing datasets, the images $QF_{estimated} \approx 100$ are non-JPEG images.

This difference between authentic and tampered images of the CASIA TIDE may provide a detection algorithm with an important hint about which image is tampered and which not, especially because the quality factor of an image may have an important role in the detection process.

Our thesis is indirectly supported by [19]. Here, the authors observe that CASIA TIDE dataset exhibits even more *statistical artifacts*, other than the ones introduced by the tampering and the one related to the images quality factors. Firstly, the JPEG compression applied to authentic images is one-time less than that applied to tampered images. Secondly the size of the chrominance components of $7,140$ authentic JPEG images is only one quarter of that of $2,061$ tampered JPEG images.

C. *Trying to Exploit Statistical Artifacts in* CASIA TIDE

In order to further prove the significance of the anomalies of the CASIA TIDE v2.0 and/or improving the recognition rate, we modified the SVM classifier used by the Lin *et al.* algorithm to decide about tampered images, by introducing three new groups of features.

The first feature is an estimation of the image luminance quality factor, as described above. This feature has been chosen mainly because we noticed that some of the Lin *et al.* algorithm features are influenced by the quality factors

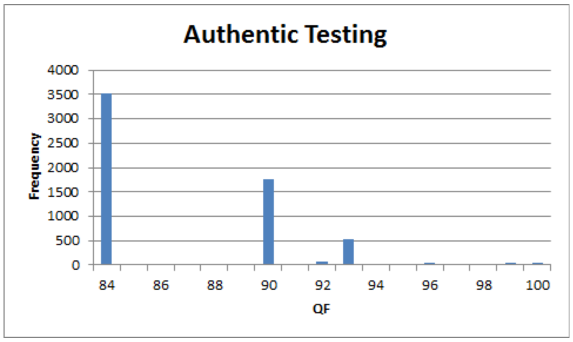

Figure 1. Frequencies of the luminance QF values of the authentic images of the $CASIA_{SC\_ALL}$ dataset in the range $[84; 100]$.

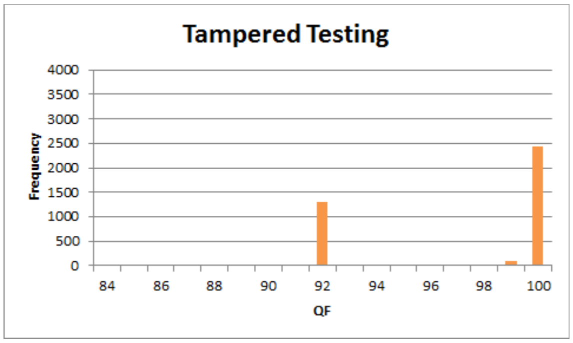

Figure 2. Frequencies of the luminance QF values of the tampered images of the $CASIA_{SC\_ALL}$ dataset in the range $[84; 100]$.

of the input image. This allows us to exploit the anomaly we have found in the CASIA TIDE dataset.

The second feature is the relative frequency of tampered blocks existing in authentic and tampered images on each color channel. In our preliminary experimentation, we noticed that both tampered and authentic images contain blocks that have been considered doctored by the Lin *et al.* method.

The third feature is the spatial resolution of the input image (i.e., width and height features). We introduce these features because we noticed that some of the original features used by the Lin *et al.* algorithm depend on the input image resolution.

This reformulation of the Lin *et al.* algorithm has been tested again on CASIA TIDE-derived datasets. In particular, we have trained and validated the SVM classifier using the Lin *et al.* original features plus different combinations of the new ones. In Table I we show some recognition rates, in percentage, of the Lin *et al.* algorithm on our CASIA TIDE-derived image sets adopting Lin *et al.* original features plus new features. Here, the R capital letter indicates the inclusion of the image width feature and image height feature, the F capital letter marks the inclusion of the relative frequencies of tampered blocks for each color channel feature and the Q capital letter indicates the inclusion of the luminance quality factor estimation feature.

TABLE I. Recognition rates, in percentage, of the Lin *et al.* algorithm on different variants of the CASIA TIDE v2.0 dataset using different features.

	CASIA TIDE Datasets			
Algorithm	**SC_ALL**	**DC_ALL**	**SC_JPEG**	**DC_JPEG**
DQD	71.00	69.55	84.32	85.08
DQD_R	71.92	71.29	84.79	85.51
DQD_F	71.74	70.85	85.76	86.41
DQD_Q	86.01	74.41	85.37	85.79
DQD_RFQ	89.15	80.17	89.25	90.10

At first glance, the introduction of the feature related to the image resolution (DQD_R) brings only a slight advantage on the *RR* of the DQD algorithm on CASIA TIDE-derived datasets. An improvement of *RR* is also achieved when considering the relative frequencies of tampered blocks features (DQD_F), especially for the case of JPEG datasets. A noteworthy behavior is provided when we add the image luminance quality factor estimation as another feature in Lin *et al.* algorithm. Here the *RR* on the $CASIA_{SC\_ALL}$ dataset led a consistent *RR* improvement over DQD ($\approx 15\%$). This effect on the $CASIA_{DC\_ALL}$ image set was consistent yet, but smaller than on the images of $CASIA_{SC\_ALL}$. On the $CASIA_{SC\_JPEG}$ and $CASIA_{DC\_JPEG}$ image sets, we measured only a slight performance improvement. These statistical artifacts provide to SVM classifier a clear distinction between tampered and non-tampered images, thus justifying the performance boost of DQD_Q on $CASIA_{SC\_ALL}$. If we leave out non-JPEG images ($CASIA_{SC\_JPEG}$ image set), the improvement is still present but is smaller. Here the gap between the quality factors of authentic images ($\approx$ 87, in the average) and tampered images ($\approx$ 90, in the average) is smaller, thus preventing a clear separation between these two classes.

Introducing doubly compressed authentic images may led to two opposite consequences: on a side, it would help the algorithm to exploit the DQ effect also on authentic images. On the other side, it could weaken statistical artifacts of CASIA TIDE dataset. In $CASIA_{DC\_ALL}$ we have considered authentic images that have been doubly compressed using uniformly at random quality factors. Here the anomaly of the CASIA TIDE dataset is less evident (the performance boost of DQD_Q is much smaller than in the $CASIA_{SC\_ALL}$ set). In $CASIA_{DC\_JPEG}$, there is a little benefit to DQD_Q compared to DQD. Here the performance loss of the algorithm (due to the absence of a clear distinction between the quality factors of the two classes) is balanced by the ability of the algorithm to perform better when dealing with the DQ effect of doubly compressed images. Finally, when we use together all the new features, we measure a further performance improvement, as witnessed by the recognition rate of DQD_RFQ (up to $\approx 90\%$).

These results confirm that by leveraging the statistical artifacts existing in CASIA TIDE, it is possible to develop an ad-hoc classifier able to achieve very good performance. This, on a side, arises the question about the soundness of the experimental results presented so far, in the scientific literature, starting from the CASIA TIDE dataset. On the other side, this calls for the need of improving this dataset.

IV. The Second Experimental Stage

Following the results presented in previous section, we designed a set of experiments to highlight how the performance of the Lin *et al.* algorithm are influenced by the distribution of input images physical properties. We have previously observed that the features extracted for detecting tampered images are influenced by several image properties such as resolution, compression rate (QF) or, even, the number of tampered pixels. Therefore, an ideal testing image dataset should exhibit a uniform distribution of these properties. In this way, it should be possible to avoid the anomalies existing in the CASIA TIDE dataset.

This is the approach we followed in the preparation of a dataset of tampered and authentic images alternative to CASIA TIDE. Our dataset includes only images with two distinct resolutions and produced by two different models of camera. Moreover, and differently from the CASIA TIDE dataset, quality factors and tampered regions sizes are uniformly distributed. This would make easier to identify the influence in the identification process by each property of the input images (avoiding mixed side effects among different properties).

A. *UniSa TIDE Dataset*

For assembling of our dataset, called UniSa TIDE[3], we used 200 original (not tampered) raw images taken using two digital cameras (Nikon D5100 and Nikon D600) at the maximum resolution (respectively, 4928×3264 and 6016×4016). Each image has been first converted in the TIFF format and, then, compressed as a JPEG file employing the following quality factors: $\{100, 98, 95, 90, 85, 80, 75, 70, 65, 60\}$. Therefore, from each input image we obtained 10 JPEG images. This would serve us to validate the robustness of the algorithm in presence of different artifacts related to quality factors. Once obtained these JPEG images, we have tampered them by hand, using splicing operations (see, e.g., Figure 3). In particular, we have selected 200 authentic JPEG images to tamper, with uniformly distributed quality factors and portraying different scenes. In the splicing operation, a region of one of the genuine images of the dataset (or other authentic datasets) is copied into another authentic image, thus giving rise to a tampered image. Sometimes, we have used the same image for duplicating areas, even using image regions with different quality factors. Then, each tampered images was compressed in JPEG format using the following quality factors: $\{100, 98, 95, 90, 85, 80, 75, 70, 65, 60\}$. Therefore, we have obtained a dataset with $2,000$ authentic JPEG images and $2,000$ tampered JPEG images, equally distributed among training and testing datasets, i.e. $1,000$ authentic training images, $1,000$ tampered training images, $1,000$ authentic testing images, $1,000$ tampered testing images. This dataset is called $UniSa_{SC}$, where “SC” stands for single compression authentic images (i.e., authentic images compressed only once). Then, we created two variants of this dataset, including a recompressed version of all the authentic images in $UniSa_{SC}$, plus all the tampered images in $UniSa_{SC}$, with no further modifications: $UniSa_{DC\_A}$ and $UniSa_{DC\_B}$. The two variants differ in the set of quality factors used for decompressing images: respectively, $QF_{DC\_A} = \{100, 99, 95, 90, 85, 80, 75, 70\}$ and $QF_{DC\_B} = \{100, 98, 95, 90, 85, 80, 75, 70, 65, 60\}$.

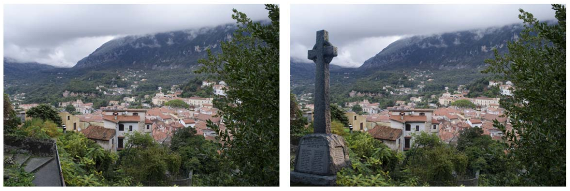

Figure 3. An authentic image and its tampered counterpart. Both are JPEG files with $QF = 100$.

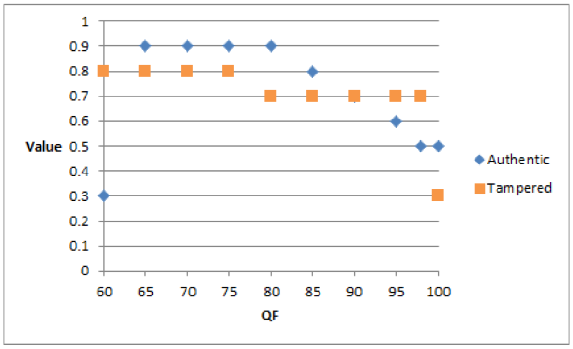

Figure 4. The values associated to the T_{opt} feature on the luminance channel for the images shown in Figure 3., determined for different quality factors.

This second set is the same used when the authentic images are saved from TIFF to JPEG file or the tampered images are saved. The first set is compatible with one used in $CASIA_{DC\_ALL}$ and $CASIA_{DC\_JPEG}$. We have used QF_{DC_A} and QF_{DC_B} because it is worthy to investigate the behavior of SVM classifier when authentic and tampered image are compressed using the same quality factors against the case where they are compressed using different ones. Adopting our methodology, the average quality factor in $UniSa_{SC}$ dataset is ≈ 83. Double compressed UniSa TIDE dataset is generated adopting a similar criteria used by Lin *et al.*.

[3] Università degli Studi di Salerno - Tampered Image Detection Evaluation image set.

B. Experimenting with UniSa TIDE

Figure 4 shows the values associated to T_{opt} feature on the luminance channel with different quality factors for the images shown in Figure 3. Here, for the authentic image, the threshold is very low for $QF = 60$, then it becomes very high between $QF = 65$ and $QF = 80$, while dropping again when using higher quality factors. When turning to the tampered image, the threshold keeps almost constant from $QF = 60$ to $QF = 75$ (below the corresponding authentic images), then it decrements and keeps stable for higher quality factor values. When using very high quality factors, it gets near 0. The cross point between the classes is fixed with $QF = 90$. These observations show that the T_{opt} feature on the luminance is also QF-dependent. In general, the image physical properties dependence is present in other Lin *et al.* features. In fact, in UniSa TIDE, despite of the same image is saved adopting different QFs, Lin *et al.* features are different.

Table II reports the recognition rates, in percentage, obtained running the same algorithm by Lin *et al.* on the UniSa TIDE datasets. The first thing to notice is that the average recognition rate achieved by the Lin *et al.* algorithm on these datasets is significantly smaller than the one reported when processing images of the CASIA TIDE dataset. Moreover, the different variants of the original algorithm, that we developed for taking advantages of the statistical artifacts existing in the CASIA TIDE dataset images, not perform better than their vanilla counterpart on these new datasets. Differently, we recall that these algorithms were subject to a substantial improvement with respect to DQD on the CASIA TIDE dataset. This reinforces our thesis about the results of these experiments being influenced by the quality factors distribution of the image dataset.

Using $UniSa_{DC\_A}$ image set, DQD_Q variant still produces an improvement with respect to DQD. This can be explained because authentic images are doubly compressed using different compression rates respect to tampered images. When $UniSa_{DC\_B}$ has been used, no relevant improvement has been measured. Moreover using $UniSa_{SC}$, a slight improvement has been obtained adding F features (DQD_F), while adding R features (DQD_R) decreased the algorithm performance.

Running DQD and DQD_Q using the $UniSa_{SC}$ and $UniSa_{DC\_B}$ datasets produced quite similar results confirming that there is no need to apply double compression on authentic images.

Finally, analyzing the results obtained with the $UniSa_{DC\_A}$ dataset, even if the *RR* of DQD was worse than the one obtained using $UniSa_{SC}$, we notice that a separation criteria between authentic and tampered images has been introduced and this increased the *RR* of DQD_Q. These results show, on a side, that all the new features that have been introduced during the experimental phase slightly increased the accuracy of the algorithm by Lin *et al.* (except in $UniSa_{DC\_B}$). On the other side, we remark that the recognition rates that we observed on these experiments are always much lower than the corresponding values obtained using CASIA TIDE images. As matter of fact, if the two datasets would have the same statical distribution the expected results should be similar. On the contrary, Lin *et al.* algorithm and all its variants have exhibited completely different behaviors during the two experiments, confirming that the experiments presented in Section III have been influenced by the non uniform distribution of the physical properties, such as QFs, of the images of the CASIA TIDE dataset.

TABLE II. Recognition rates, in percentage, of the Lin *et al.* algorithm on different variants of the UniSa TIDE dataset using different features.

	UniSa TIDE Datasets		
Algorithm	**SC**	**DC_A**	**DC_B**
DQD	55.20	54.85	55.50
DQD_R	54.70	54.95	55.95
DQD_F	56.50	55.40	53.60
DQD_Q	55.45	59.50	55.60
DQD_RFQ	59.15	59.20	53.75

Comparing the results on CASIA TIDE and UniSa TIDE datasets, the first experimental stage shows an high recognition rate of tampering or not using Lin *et al.* algorithm, while the second experimental stage depicts a detection algorithm that cannot be effective in real practice. Our experimentations describe that recognition rates are the result of a bad training dataset that could have statistical artifacts, different from tampering, that can help the separation between authentic and tampered classes.

V. Conclusions, Remarks and Future Works

In this paper, we discussed the problem of the experimental evaluation of tampered image detection algorithms. Namely, we focused on the evaluation of a public dataset of images, the CASIA TIDE dataset, that is the *de facto* standard for the experimental evaluation of these algorithms. Starting from the results of previous studies conducted on literature, we analyzed the properties of the images available in this dataset in order to verify if they could, in some way, influence the detection process. This has been done by profiling the performance of a reference algorithm, the algorithm by Lin *et al.*, on this dataset.

The obtained results confirm the existence of some statistical artifacts in the way the dataset has been built (i.e., many of the tampered images have been saved using almost-fixed QF compared to authentic images QFs). These observations could facilitate the detection activities of tampered images respect to authentic images also only using these statistical artifacts. This problem has been confirmed by some further experiments. Namely, we added to the original Lin *et al.* algorithm the ability to detect whether an image is tampered or not by looking also at these anomalies. In fact, the revised versions of the algorithm exhibited a significant performance boost on a image set featuring almost all the images of the CASIA TIDE dataset. Therefore, we have defined an alternative image dataset, not affected by the problems found on the CASIA TIDE. When experimenting with this new dataset, we noticed a recognition rate significantly smaller than the one measured with the CASIA TIDE dataset. These results would suggest the opportunity to review the experimental studies conducted so far using CASIA TIDE image set, as their outcomes may have been influenced by the aforementioned statistics anomalies.

REFERENCES

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

[2] T. Ahonen, "Celebrating 30 Years of Mobile Phones, Thank You NTT of Japan," http://communities-dominate.blogs.com/brands/2009/11/celebrating-30-years-of-mobile-phones-thank-you-ntt-of-japan.html, April 2014.

[3] C. Smith, "By the Numbers: 105 Amazing Facebook User Statistics," http://expandedramblings.com/index.php/by-the-numbers-17-amazing-facebook-stats, April 2014.

[4] J. Lukáš, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, pp. 205–214, November 2006.

[5] N. Khanna, A. K. Mikkilineni, G. T. Chiu, J. P. Allebach, and E. J. Delp, "Scanner identification using sensor pattern noise," *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, no. 1, pp. 1–11, 2007.

[6] S. Ye, Q. Sun, and E.-C. Chang, "Detecting digital image forgeries by measuring inconsistencies of blocking artifact," *IEEE International Conference on Multimedia and Expo 2007*, pp. 12–15, 2007.

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

[8] G. Cattaneo, P. Faruolo, and U. Petrillo, "Experiments on improving sensor pattern noise extraction for source camera identification," in *Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, July 2012, pp. 609–616.

[9] G. Cattaneo, G. Roscigno, and U. F. Petrillo, "Experimental evaluation of an algorithm for the detection of tampered JPEG images," in *Information and Communication Technology-EurAsia Conference (ICT-EurAsia) 2014*. Springer, 2014, pp. 643–652.

[10] G. Cattaneo, G. Roscigno, and U. Petrillo, "A scalable approach to source camera identification over Hadoop," in *28th IEEE International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2014, pp. 366–373.

[11] "Independent JPEG Group code library," http://www.ijg.org/, March 2014.

[12] J. Lukáš and J. Fridrich, "Estimation of primary quantization matrix in double compressed JPEG image," *Proc. Digital Forensic Research Workshop*, pp. 5–8, 2003.

[13] Institute of Automation, Chinese Academy of Sciences (CASIA), "CASIA Tampered Image Detection Evaluation Database (CASIA TIDE) v2.0," http://forensics.idealtest.org/, 2013.

[14] T. Abeel, Y. V. de Peer, and Y. Saeys, "Java-ML: A Machine Learning Library," *Journal of Machine Learning Research*, vol. 10, pp. 931–934, 2009.

[15] W. Wang, J. Dong, and T. Tan, "Image tampering detection based on stationary distribution of Markov chain," in *17th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2010, pp. 2101–2104.

[16] P. Sutthiwan, Y. Q. Shi, W. Su, and T.-T. Ng, "Rake transform and edge statistics for image forgery detection," in *IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2010, pp. 1463–1468.

[17] W. Wang, J. Dong, and T. Tan, "Tampered region localization of digital color images based on JPEG compression noise," in *Digital Watermarking*. Springer, 2011, pp. 120–133.

[18] M. Jaberi, G. Bebis, M. Hussain, and G. Muhammad, "Improving the detection and localization of duplicated regions in copy-move image forgery," in *18th International Conference on Digital Signal Processing (DSP)*. IEEE, 2013, pp. 1–6.

[19] P. Sutthiwan, Y. Q. Shi, H. Zhao, T.-T. Ng, and W. Su, "Markovian rake transform for digital image tampering detection," in *Transactions on data hiding and multimedia security VI*. Springer, 2011, pp. 1–17.