Experimental Evaluation of an Algorithm for the Detection of Tampered JPEG Images

Giuseppe Cattaneo[1], Gianluca Roscigno[1], and Umberto Ferraro Petrillo[2,⋆]

[1] Dipartimento di Informatica
Università degli Studi di Salerno, I-84084, Fisciano, SA, Italy
{cattaneo,giroscigno}@unisa.it
[2] Dipartimento di Scienze Statistiche
Università di Roma *"La Sapienza"*, I-00185, Roma, Italy
umberto.ferraro@uniroma1.it

Abstract. This paper aims to experimentally evaluate the performance of one popular algorithm for the detection of tampered JPEG images: the algorithm by Lin *et al.* [1]. We developed a reference implementation for this algorithm and performed a deep experimental analysis, by measuring its performance when applied to the images of the CASIA TIDE public dataset, the *de facto* standard for the experimental analysis of this family of algorithms. Our first results were very positive, thus confirming the good performance of this algorithm. However, a closer inspection revealed the existence of an unexpected anomaly in a consistent part of the images of the CASIA TIDE dataset that may have influenced our results as well as the results of previous studies conducted using this dataset. By taking advantage of this anomaly, we were able to develop a variant of the original algorithm which exhibited better performance on the same dataset.

Keywords: Digital Image Forensics, JPEG Image Integrity, Double Quantization Effect, Evaluation Datasets.

1 Introduction

Nowadays, the manipulation of digital images is simpler than ever thanks to solutions like sophisticated photo-editing software or photo-sharing social networks. Indeed, one of the key characteristics of digital images is their pliability to manipulation. As a consequence, we can no longer take the authenticity of digital photos for granted. This can be a serious problem in situations where the reliability of the images plays a crucial role, such as when conducting criminal investigations. This has led, in the recent years, to the development of the Digital Image Forensics, a discipline that is responsible for the acquisition and the analysis of images found on digital devices for investigation purposes (see, e.g., [2,3,4,5,6,7]).

⋆ Corresponding author.

Linawati et al. (Eds.): ICT-EurAsia 2014, LNCS 8407, pp. 643–652, 2014.

One of the problems faced by this discipline is to verify if a digital image is authentic or has been manipulated after the acquisition (*image tampering*). Several algorithms have been proposed in the scientific literature for solving this problem, such as [1], [4], [5]. It is a common practice to accompany these contributions with an experimental analysis devoted to prove the viability of the proposed approach in the real practice. A popular dataset that is widely used for these experimentations is the CASIA TIDE dataset [8]. In this paper, we focus our attention on a particular algorithm for the detection of tampered JPEG digital images: the algorithm by Lin *et al.* [1]. Despite its popularity, this algorithm has never been tested on the CASIA TIDE dataset. We decided to close this gap, by developing an implementation of this algorithm and testing it over several different variants of this dataset. The results are contrasting. On a side, they confirm the good performance of the Lin *et al.* algorithm. On the other side, we experimented an anomaly in a consistent number of images forming the CASIA TIDE dataset which is likely to partially influence the results of the experiments conducted using these images. By taking advantage of this anomaly, we were able to further improve the performance of the original algorithm.

Organization of the Paper. In Section 2 we provide some details about the JPEG standard. In Section 3, we review some of the main contributions in the field of the JPEG image integrity. The aim of this work is illustrated in Section 4. Then, in Section 5, we focus on the Lin *et al.* algorithm. Section 6 describes the experiments we have conducted and their results. Finally, in Section 7 we draw some concluding remarks for our work.

2 The JPEG Standard

The JPEG [9] is a very popular standard for the encoding of digital images. It uses several techniques to guarantee very high compression rates at the expense of a small degradation in the image quality. The core of this standard is a lossy-compression technique based on the discrete cosine transform (*DCT*). In a few words, this technique assumes that an input image is encoded in the YCbCr format, where the Y channel holds the luminance component of the image, and the Cb and Cr chrominance channels hold, respectively, the blue minus luminance component of the image and the red minus luminance component of the image. The DCT operation works by logically splitting each image channel into blocks of 8×8 pixels and by converting them from the spatial domain into the frequency domain, producing blocks of 8×8 *DCT coefficients*, for a total of 64 frequencies. Once in the frequency domain, these coefficients are compressed by drastically reducing the amount of information provided by the high frequencies (i.e., quantization). This is done by dividing for a fixed constant (*quantization step*) each component of the 8×8 block of the DCT coefficient matrix and, then, by rounding the resulting values. The quantization steps can be chosen according to the amount of information to leave out from the frequencies, thus influencing the quality and the size of the compressed image (*quality factor*,

QF). For different frequencies and channels, the quantization steps are saved in quantization matrices which can be extracted from the JPEG image. In general, there is a luminance quantization matrix for the luminance channel and a chroma quantization matrix for the two chrominance channels. At the end of this step, quantized coefficients are sequenced and losslessly compressed. The quality factor of a JPEG image (see [10]) can vary in the range $[1, 100]$, where smaller values result in a lower quality of the compressed image and a higher compression degree.

3 JPEG Image Integrity

The digital image integrity research field concerns with the problem of assessing if a digital image is the result of some *forging* operation. By this term, we mean all the techniques used to alter the contents of an authentic image, such as painting new objects in a portrayed scene or copying the region of an image over another image (*splicing* operation). Detecting the forgery of a JPEG image can be harder than for other formats because the compression steps employed by this encoding may delete the forgery traces left in a tampered image. However, an algorithm could also try to discover new traces caused by recompression of a tampered image and use these traces to detect the forgeries. In fact, the artifacts introduced by JPEG compression (so said *JPEG blocking artifacts*) can be seen as an inherent "watermark" for compressed images. These artifacts result to be modified when a JPEG image is altered by means of forging operations.

Many image integrity algorithms follow this approach (see, e.g., [1], [4], [5]). These algorithms use some of the statistical properties of the DCT coefficients to detect inconsistencies in the blocking artifacts of a target JPEG image. One of the first detection technique is described in [11]: it is a method that estimates the primary quantization matrix from a doubly compressed JPEG image using the histograms of the individual DCT coefficients. A similar approach has been proposed by Ye *et al.* in [4]. First, they used the histogram of the DCT coefficients to estimate the quantization step size and, then, they measured the inconsistency of quantization errors between different image regions for estimating local compression blocking artifacts measure. A major drawback of this algorithm is that it requires a preliminary human intervention to select a suspicious region of the image to analyze. Farid [5] proposed a technique, based on the detection of *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 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 blocks in a JPEG image. This is done by examining the *double quantization* (DQ) effect contained in the DCT coefficients, and computing the Block Posterior Probability Map (BPPM) using a Bayesian approach.

4 Aim of This Work

The aim of this work is to experimentally assess the effectiveness of the Lin *et al.* algorithm for detecting the tampering of JPEG images. The authors of this algorithm already conducted an experimental evaluation of its performance. However, this evaluation was conducted using a proprietary dataset, thus limiting the possibility to compare the performance of their algorithm with other alternative approaches. As recognized by the authors of the algorithm, such a choice was motivated by the difficulty in assembling a large set of images that have not been (or that have been) tampered for sure. Nowadays, this problem seems to be partially solved. The arising interest toward this topic has led to the development of public datasets of images, to be used for evaluating digital tampering detection algorithms. This fact marks an important opportunity for the scientific community working in this field, as it allows to compare the different algorithms proposed so far according to a common benchmark. Moreover, it becomes possible to evaluate the performance of an algorithm in a neutral way.

5 The Algorithm by Lin *et al.*

The algorithm by Lin *et al.* [1] detects tampered images by examining the *double quantization* (DQ) effect contained in the DCT coefficients. This effect occurs when the DCT coefficients histogram of an image has periodically missing values or some periodic pattern of peaks and valleys. According to Lin *et al.*, this effect can be used for image authentication. To this end, they show that the image regions (i.e. 8×8 blocks) that do not exhibit the DQ effect are probably tampered. Namely, in a tampered image, untampered blocks will exhibit the DQ effect, while tampered blocks (also called *doctored* blocks) will not.

The algorithm works as follows. As a preliminary step, if the input image I is not a JPEG image, it is converted to this format at highest quality. The first step of the algorithm is the extraction from I of the DCT coefficients and of the quantization tables for each of the three YCbCr channels. As second step, the algorithm builds one DCT coefficients histogram for each of the three YCbCr channels and for each of the 64 frequencies. The computed histograms are used for determining a probability value which indicates if a particular 8×8 block of the input channel image is doctored, by checking the DQ effect. In turns, in the third step, these probabilities are combined together to produce the Block Posterior Probability Map (BPPM). The resulting BPPM is thresholded to distinguish between (probably) doctored parts and undoctored parts. In details, for each image and for each channel, the values of the corresponding BPPM are classified, according to a threshold T, into two classes: tampered blocks (C_0) and untampered blocks (C_1). The fourth step is the extraction of a four-dimensional feature vector for each of the three YCbCr channels. The first feature is the sum of the variances of the probabilities in C_0 and C_1. The second feature is the squared difference between the mean probabilities of C_0 and C_1. The third feature, T_{opt}, is a threshold 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 doctored blocks. The last feature, K_0, is a measure of the connectivity of C_0 blocks: more is connected C_0, then smaller is K_0. In the last step of the algorithm, 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.

6 Experimental Analysis

In this section we detail the results of the experimental analysis we conducted on the Lin *et al.* algorithm. To begin, we developed a Java-based implementation of this algorithm[1]. This implementation, here named `DQD`, includes two core modules:

- The **Feature Extractor** module is in charge of extracting the features used for the classification by the Lin *et al.* algorithm from a batch of input images. The extraction is performed with the help of the `libjpeg` library [12]. If the input image is not in the JPEG format, we apply the JPEG compression algorithm at the maximum quality factor.
- The **SVM Manager** module is in charge of managing the SVM classifier to be used for detecting tampered images or not. It uses the SVM implementation available with the Java Machine Learning library [13], using the LIBSVM module. To get more accurate decisions, our classifier uses the *Cross Validation* (*CV*) and the *Grid Search* (*GS*) techniques. The *CV* technique is a method of sampling used to divide the original sample into two subsets: in the first subset we estimate the parameters of a model, while in the second subset we measure the predictive ability of the estimate. The *GS* technique is a search technique used to identify the parameters that optimize the performance of the classifier exploiting the cross validation.

The next step has been to assess the performance of the developed implementation by analyzing its experimental behavior on several datasets. In the following, we introduce the datasets used in our experimentations, then we describe the experimental setting and discuss the outcoming results.

6.1 Dataset

In the recent years, several public datasets have been released for evaluating tampered image detection algorithms. If we restrict our interest to JPEG-based datasets, there is only one choice that has become the *de facto* standard for these experimentations: the CASIA TIDE dataset[2] [8]. It is available in two

[1] A copy of the source code of our implementation 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`

versions: v1.0 and v2.0. The v1.0 version is a small-scale dataset of low resolution images. The v2.0 version is a large-scale dataset containing 7,491 authentic and 5,123 tampered color images. The images have different resolutions, varying from 240×160 to 900×600 pixels, and different formats (i.e., BMP, TIFF, JPEG). Tampered images included in this dataset are the result of a splicing that has been dissimulated, sometimes, by using a *blur* operation over the tampered image.

We have chosen for our experiments the v2.0 of the dataset because it contains a large number of high resolution images, tampered using more sophisticated techniques. In addition, this dataset has been widely used in the scientific literature for evaluating tampered images detection algorithms. Thus, its adoption would simplify the comparison of the performance of the Lin *et al.* algorithm with other detection algorithms.

Starting from the original CASIA TIDE v2.0 dataset, we extracted a subset of images, called *SC_ALL*, containing 2,000 random training photos divided into 1,000 authentic (990 JPEG and 10 BMP) and 1,000 tampered (633 TIFF and 367 JPEG). The remaining 6,491 authentic (44 BMP and 6,447 JPEG) and 3,875 tampered images (2,426 TIFF and 1,449 JPEG) were used for SVM testing. Notice that some tampered images were left out of our experiments because they originated from non-JPEG images and, thus, they could not exhibit the DQ effect on which relies the Lin *et al.* algorithm. A second dataset, called *SC_JPEG*, has been obtained by filtering only the JPEG images of *SC_ALL*. A third dataset, called *DC_ALL*, was obtained by considering all images of *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, we introduced a fourth dataset, *DC_JPEG*, obtained by filtering only the JPEG images existing in *DC_ALL*.

6.2 Preliminary Experimentations

In this section, we present the results of a first round of experimentations of the Lin *et al.* algorithm. To begin, we conducted a preliminary test to evaluate the optimal number of frequencies to be used for building the DCT coefficients histograms required by the algorithm. In theory, for each color channel, these coefficients are related to an overall number of 64 frequencies. However, since the high frequencies DCT coefficients are often quantized to zero, we will use only the lower frequencies histograms for each channel. To this end, we ran the algorithm on our datasets using, respectively, the 32, the 48 and the 64 lower frequencies. The outcoming results show that the best setting, in terms of *recognition rate* (*RR*, i.e. the percentage of testing images correctly recognized as authentic or tampered), is reached when using 32 frequencies. Therefore, in subsequent experiments, we will always use this value.

Table 1 shows the percentage of testing images correctly recognized as authentic or tampered (*RR*) when we ran `DQD` on the datasets defined in Section 6.1 using the 32 lower frequencies. These numbers confirm the good performance of the Lin *et al.* algorithm. We observe that the algorithm was able to correctly

Table 1. Recognition rates, in percentage, of the Lin *et al.* algorithm on different variants of the CASIA TIDE v2.0 dataset using different features

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_FQ	88.78	78.72	87.46	89.30
DQD_QR	86.13	75.10	85.45	85.93
DQD_FR	73.06	72.25	86.36	87.30
DQD_FQR	89.15	80.17	89.25	90.10

recognize as original or tampered about the 70% of the images belonging to the *SC_ALL* and *DC_ALL* datasets. This percentage increases by, approximately, a 15% when considering the variants of these datasets including only JPEG images. This seems to indicate that, differently from TIFF tampered images, JPEG tampered images are able to retain some statistical artifacts useful for the classifier to detect their forgery. This is confirmed by the improvement of the performance of the algorithm on doubly compressed images (recall that the algorithm has been designed for dealing with doubly compressed JPEG images) while the inclusion of TIFF images (*DC_ALL*) leads to a small degradation of the overall performance of the algorithm.

A deeper analysis of the images of the CASIA TIDE dataset revealed a sort of anomaly. The luminance quality factor of the majority of the JPEG tampered images is always the same (i.e., either set to about 90 or to about 100), while authentic images have more variable quality factors, ranging mostly in the interval $[70, 100]$. This fact may compromise the quality of this dataset since it introduces a separation criteria between authentic and tampered images that is relatively easy to verify.

6.3 Experimenting with New Features

Starting from the observations made in Section 6.2, we explored the possibility to improve the performance of the Lin *et al.* algorithm, by enriching the SVM classifier with three additional groups of features.

The first feature is an estimation of the image luminance quality factor. This feature has been chosen for two reasons. First, we noticed that some of the Lin *et al.* algorithm features are influenced by the applied JPEG compression rate, i.e. quality factor. Second, we are interested in exploiting the anomaly found in the CASIA TIDE dataset, about the luminance quality factors being always the same for tampered images, to see how much it could influence the performance of the classification. In our experiments, we extracted this feature starting by the

luminance quantization table embedded in each JPEG image. Namely, we apply an estimate of the inverse formula of [10] 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 matrix.

The second feature is the relative frequency of doctored blocks existing in authentic and tampered images on each channel. We noticed that both tampered and authentic images contain blocks that have been considered doctored by the Lin *et al.* method. However, if we consider only the chrominance channels, the average number of doctored blocks for tampered images clearly differs from the average number of doctored blocks for authentic images. Therefore, this information can improve the separation of these two classes.

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

After introducing these new features, we have trained and tested the SVM classifier using the original features of the Lin *et al.* algorithm plus different combinations of the new ones. The results are presented in Table 1. Here, the `R` capital letter marks the inclusion of the image resolution feature (i.e, width feature and 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 marks the inclusion of the luminance quality factor estimation feature.

According to these results, it seems that the introduction of the feature related to the image resolution (`DQD_R`) brings only a slight advantage on the performance of the `DQD` algorithm on all considered datasets. An improvement is also achieved when considering the features related to the relative frequencies of tampered blocks (`DQD_F`), especially for the case of JPEG images. This is a further confirmation of the ability of the algorithm to detect the DQ effect on doubly compressed JPEG tampered images.

A completely different behavior is the one we observed with the introduction of the feature related to the image luminance quality factor. On the *SC_ALL* dataset, the inclusion of this feature led to a consistent performance improvement over `DQD`. On the *DC_ALL* dataset the improvement was consistent ye,t but smaller than on the *SC_ALL* dataset. On the *SC_JPEG* and *DC_JPEG* datasets, we measured only a slight performance improvement. To explain this we recall the observation made in Section 6.2 about the JPEG tampered images belonging to the CASIA TIDE datasets. These images are always saved using fixed quality factors, whereas their non-tampered counterparts are saved using variable quality factors. In addition, TIFF tampered images are automatically converted by our algorithm into JPEG images with a quality factor fixed to 100. These two facts provide the classifier with a clear distinction between tampered and non-tampered images, thus justifying the performance boost of `DQD_Q` on *SC_ALL*. If we leave out from the comparison the TIFF images, like in the *SC_JPEG* case, the improvement is still present but is smaller. Here the gap between the quality factor of authentic images (87, in the average) and

tampered images (90, in the average) is smaller, thus preventing a clear separation between these two sets. The anomaly of the CASIA TIDE dataset is weakened when considering authentic images that have been recompressed using random quality factors, such as in *DC_ALL*. Here, the performance boost of `DQD_Q` is much smaller than in the *SC_ALL* case. Instead, the recompression of the input images brings a little benefit to `DQD_Q` when dealing with only JPEG images. In this case, the performance loss due to the absence of a clear distinction between the quality factor of tampered and authentic images is balanced by the ability of the algorithm to perform better when dealing with the DQ effect of doubly compressed images. According to our experiments, even the mixing of these new features improves the overall quality of the detection. This is especially the case of `DQD_FQR` algorithm, which is the variant of the Lin *et al.* algorithm exhibiting the better performance, by considering all the features of the original algorithm plus all the features we introduced in our experiments.

7 Conclusions and Future Work

In this paper, we analyzed the experimental performance of the algorithm by Lin *et al.* [1], a popular technique for the detection of tampered JPEG images. The experiments have been conducted on several variants of the v.2.0 CASIA TIDE dataset, a collection of images developed for the experimental evaluation of this family of algorithms. The final aim of this experimentation was to facilitate the comparison of the Lin *et al.* algorithm with other alternative approaches by measuring its performance on a widely-used testbed.

On a side, the results we obtained confirmed the good performance of this algorithm. On the other side, we were able to detect a sort of anomaly in the way the CASIA TIDE dataset has been built (i.e., many of the tampered images have been saved using almost-fixed quality factors) that could facilitate the detection activity. The relevance of this anomaly has been confirmed by a further experimentation, where we added to the original Lin *et al.* algorithm the ability to detect whether an image is tampered or not by looking also at this anomaly. As a matter of fact, the revised version of the algorithm exhibited a significant performance boost on a dataset featuring almost all the images of the CASIA TIDE dataset.

These observations pose the question about the full statistical soundness of the CASIA TIDE dataset while suggesting the opportunity to reconsider the results of all the experimental studies that have been conducted so far using this dataset. As a future direction, it would be useful to fix the anomaly we found in the CASIA TIDE dataset or, as an alternative, to introduce of an improved testbed for the experimentation of image integrity detection algorithm able to solve the problems existing in the CASIA TIDE dataset. Concerning the Lin *et al.* algorithm, a future direction for this work would be to extend our experimental analysis to other relevant contributions existing in the field of algorithms for assessing the integrity of JPEG images.

References

1. Lin, Z., He, J., Tang, X., Tang, C.-K.: Fast, Automatic and Fine-grained Tampered JPEG Image Detection via DCT Coefficient Analysis. Pattern Recognition 42(11), 2492–2501 (2009)
2. Lukáš, J., Fridrich, J., Goljan, M.: Digital Camera Identification from Sensor Pattern Noise. IEEE Transactions on Information Forensics and Security 1, 205–214 (2006)
3. Khanna, N., Mikkilineni, A.K., Chiu, G.T.C., Allebach, J.P., Delp, E.J.: Scanner Identification using Sensor Pattern Noise. In: Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents IX, vol. 6505(1), pp. 1–11 (2007)
4. Ye, S., Sun, Q., Chang, E.-C.: Detecting Digital Image Forgeries by Measuring Inconsistencies of Blocking Artifact. In: IEEE International Conference on Multimedia and Expo 2007, pp. 12–15 (2007)
5. Farid, H.: Exposing Digital Forgeries from JPEG Ghosts. IEEE Transactions on Information Forensics and Security 4(1), 154–160 (2009)
6. Cattaneo, G., Faruolo, P., Petrillo, U.F.: Experiments on improving sensor pattern noise extraction for source camera identification. In: 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pp. 609–616 (July 2012)
7. Castiglione, A., Cattaneo, G., Cembalo, M., Petrillo, U.F.: Experimentations with Source Camera Identification and Online Social Networks. Journal of Ambient Intelligence and Humanized Computing 4(2), 265–274 (2013)
8. Institute of Automation, Chinese Academy of Sciences (CASIA). CASIA Tampered Image Detection Evaluation Database (CASIA TIDE) v2.0 (2013), `http://forensics.idealtest.org/`
9. Wallace, G.K.: The JPEG Still Picture Compression Standard. Communications of the ACM, 30–44 (1991)
10. Independent JPEG Group code library (December 2013), `http://www.ijg.org/`
11. Lukáš, J., Fridrich, J.: Estimation of Primary Quantization Matrix in Double Compressed JPEG Images. In: Proc. Digital Forensic Research Workshop, pp. 5–8 (2003)
12. Tom Lane and the Independent JPEG Group (IJG). libjpeg (2013), `http://libjpeg.sourceforge.net/`
13. Abeel, T., de Peer, Y.V., Saeys, Y.: Java-ML: A Machine Learning Library. Journal of Machine Learning Research 10, 931–934 (2009)