Multiple Compression Detection for Video Sequences

S. Milani, P. Bestagini, M. Tagliasacchi, S. Tubaro

Dipartimento di Elettronica ed Informazione, Politecnico di Milano
piazza Leonardo da Vinci 32
20133 Milano, Italy
milani/bestagini/tagliasa/tubaro@elet.polimi.it

***Abstract*—Nowadays, thanks to the increasingly availability of powerful processors and user friendly applications, the editing of video sequences is becoming more and more frequent. Moreover, after each editing step, any video object is almost always encoded in order to store it using a less amount of memory. For this reason, inferring the number of compression steps that have been applied to such a multimedia object is an important clue in order to assess its authenticity.**

In this paper we propose a method to recover the number of compression steps applied to a video sequence. In order to accomplish this goal, we make use of a classifier based on multiple Support Vector Machines (SVM) exploiting the Benford's law. Indeed, the feature vectors used to train and test the SVM are based on the statistics of the most significant digit of quantized transform coefficients.

The proposed method is tested with a generic hybrid video encoder combining motion-compensation and block coding. Results show that this method is able to discriminate up to three compression stages with high accuracy.

I. Introduction

The recent development of multimedia devices and editing tools, together with the proliferation of video sharing web sites, has made the acquisition, alteration, and diffusion of video contents relatively-easy tasks. As a consequence, we find more and more video sequences available on the Internet, but each of them is potentially tampered with by anyone [1].

A typical operation involved in video tampering is the compression step. Indeed, video signals usually consist in an extensive set of data that make their handling and transmission prohibitive. For this reason, a first compression stage may take place on the acquisition device in order to allow an effective storage on the on-board memory. A second compression (or even more) is operated by the same owner after editing the acquired content (e.g., adjusting the video sequence to enhance its quality, or including titles and references). An additional compression can then be performed when uploading the content on a video-sharing platform or a website. As a result, video sequences are usually compressed multiple times.

However, encoding a video object involves non-invertible operations such as quantization, and this leaves peculiar footprints on the sequence itself. Thus, by studying these footprints, a forensic analyst may infer the number of compression steps that have been applied to a sequence. As a matter of fact, the detection of the number of compression steps is a significant indicator of the number of elaborations that were applied. This has significant implications on media authenticity and validation, particularly when the analysts have to provide legal evidence in a trial, or some copyright violations need to be verified.

So far, multimedia forensic analysts have mainly focused on the detection of double image and video compression, i.e., they aim at detecting whether a video has been compressed once or twice. Additional works have also been focusing on disguising compression footprints (antiforensic strategies) in order to make the resulting image as if it has been compressed once [2]. In [3], [4] Farid *et al.* analyze coefficient statistics to detect double compression and find out the adopted quantization steps, while statistics on bitrates for coded frame allow the identification of cuts and changes in the GOP size. Antiforensics strategies targeting frame insertion/deletion have been studied as well in order to test the robustness of these approaches whenever a malicious user attempts to trick them [5]. It is also possible to exploit compression artifacts in order to reveal previous compressions as Luo *et al.* show in [6] or distortion introduced by interlacing and de-interlacing [7]. Another work by Liao *et al.* [8] targets the problem of double compression in H.264/AVC standard. In [9], Bestagini *et al.* employ the idempotency principle to detect the parameters of the first coding stage provided that compression has been operated twice. In this case, the number of compression that have been operated is assumed to be known. In the approach by Chen and Shi [10] the first digit feature, derived from double JPEG compression works, is used in order to discriminate between single or double compression.

However, as stated before, multimedia signals are very likely to be compressed more than twice. Thus, identifying the number of compression stages is crucial information in the reconstruction of the past processing history of the video and in the evaluation of the reliability of the displayed content (i.e., how faithfully it reproduces reality).

In this paper, we focus on multiple video compressions (up to three) operated by a generic hybrid video coder combining transform coding and motion compensation. The detector is based on the combination of multiple binary Support Vector

MMSP'12, Sep. 17-19, 2012, Banff, Canada.

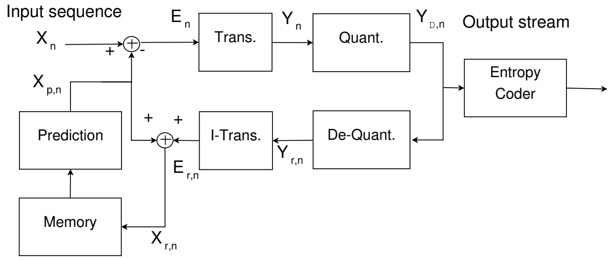


Fig. 1. Block diagram for generic video compression.

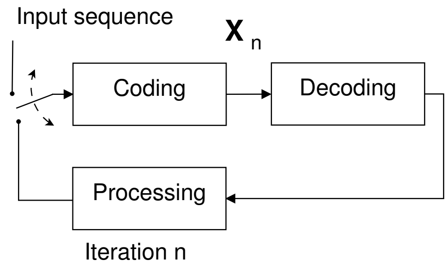


Fig. 2. Block diagram for multiple compression.

Machines (SVM). The used feature vectors are built collecting information about the statistics of the most significant digit of quantized transformed coefficients. The theoretical aspect of this work is justified by Benford's law. The work is a natural extension of [11], where multiple image compression was considered. However, in this case, estimation is made more complex because of motion estimation that alters the alignment between blocks mixing coefficient statistics.

In the following, Section II describes the behavior of DCT coefficient statistics, as they go through multiple quantization stages, analyzing the theoretical aspects of the problem. Section III presents the proposed classification method based on multiple SVM. Experimental results are reported in Section IV, and final conclusions are drawn in Section V.

II. Multiple compressions and coefficients statistics

Most of the recent video compression standards combines spatial and/or temporal prediction with block-based transform coding. At the n-th compression stage, the input frame is divided into K regular blocks $\mathbf{X}_n^k,\ k = 1, ..., K$, which are spatially or temporally predicted by a predictor block. From now on we omit the apex k for compactness without loss of generality, focusing on a single block at a given compression step. The predictor $\mathbf{X}_{p,n}$ is chosen among the pixel blocks of the previously reconstructed frames (see Figure 1). The residual block $\mathbf{E}_n = \mathbf{X}_n - \mathbf{X}_{p,n}$ is then transformed and the resulting block $\mathbf{Y}_n$ of coefficients is quantized into the block $\mathbf{Y}_{\Delta,n}$. Most of the time quantization is performed independently on each coefficient using a uniformly-distributed output levels with a dead-zone around the zero in order to maximize the percentage of null reconstructed coefficients. In the adopted notation, the quantization levels are

$$\mathbf{Y}_{\Delta,n}(i,j) = \text{sign}(\mathbf{Y}_n(i,j))\text{round}\left(\frac{|\mathbf{Y}_n(i,j)|}{\Delta_n(i,j)}\right), \quad (1)$$

where the indexes (i, j) denote the position of the elements in the block. The adopted quantization step at the n-th compression stage is $\Delta_n(i, j)$, which may change according to the spatial frequencies of the coefficient and the type of coding for the current block. The coded block $\mathbf{Y}_{r,n}$ can be reconstructed by multiplying the quantized values $\mathbf{Y}_{\Delta,n}(i, j)$ with the corresponding quantization step $\Delta_n(i, j)$. The displayed block $\mathbf{X}_{r,n}$ can be obtained by inversely transform the block $\mathbf{Y}_{r,n}$ and adding the corresponding predictor. As a result, it is possible to model the coding distortion with the distortion block $\mathbf{D}_{r,n}$ so that $\mathbf{X}_{r,n} = \mathbf{X}_n + \mathbf{D}_{r,n}$. Note that, in case of medium-high distortions, the distortion $\mathbf{D}_{r,n}$ is correlated with the quantized signal.

These operations can be iterated multiple times (see Figure 2) by re-encoding the output sequence after some eventual processing steps (e.g., cropping, rescaling, etc...). As a result, the coefficient statistics is altered according to the number of compression stages that were applied to the original data.

Let us consider the double compression case. After the first coding stage, the output signal is $\mathbf{X}_{r,1} = \mathbf{X}_1 + \mathbf{D}_{r,1}$ where $\mathbf{D}_{r,1}$ is correlated with the error $\mathbf{E}_1 = \mathbf{X}_1 - \mathbf{X}_{p,1}$. During the second coding stage, $\mathbf{X}_2 = \mathbf{X}_{r,1}$ is predicted by $\mathbf{X}_{p,2} = \mathbf{X}_{p,1} + \mathbf{D}_{r,2}^p$, where $\mathbf{D}_{r,2}^p$ is the distortion introduced on the predictor by the second compression. In this case, we assume that the estimated motion vectors (MVs) are approximately the same because of either the smoothness of motion field or the regularity among MVs introduced by rate-distortion optimization (i.e., smooth MV fields permit a more effective compression of MV data) and MV prediction.

Therefore, the second compression stage has to quantize $\mathbf{E}_2 = \mathbf{E}_1 + \mathbf{D}_{r,1} - \mathbf{D}_{r,2}^p$, where $\mathbf{D}_{r,2}^p$ is uncorrelated with $\mathbf{E}_1 + \mathbf{D}_{r,1}$ since it is referred to previous blocks. It is possible to notice that, assuming that a uniform quantizer is chosen, the statistics of quantized $\mathbf{E}_2$ for a given realization of $\mathbf{D}_{r,2}^p$ equals a shifted version of the statistics of quantized $\mathbf{E}_1 + \mathbf{D}_{r,1}$. Averaging the probability mass function (pmf) over the different realizations of $\mathbf{D}_{r,2}^p$ we have that the statistics of quantized $\mathbf{E}_2$ equals the statistics of a double quantization on $\mathbf{E}_1$ (statistics of $\mathbf{E}_1 + \mathbf{D}_{r,1} + \mathbf{D}_{r,2}$). From these assumptions, it is possible to relate the statistics of double video compression to the analysis of double quantization for random variables.

In the case of transform coefficients from video coding (which can be modeled with exponential variables), it is possible to detect the number of quantizations by analyzing the violations of Benford's law (also known as first digit law or significant digit law) for the quantized transform coefficients [12]. Let m denote the first digit (FD) m of coefficient $\mathbf{Y}_{\Delta,n}(i, j)$, i.e.

$$m = \text{FD}(\mathbf{Y}_{\Delta,n}(i,j)) = \left\lfloor \frac{|\mathbf{Y}_{\Delta,n}(i,j)|}{10^{\lfloor \log_{10} |\mathbf{Y}_{\Delta,n}(i,j)| \rfloor}} \right\rfloor. \quad (2)$$

It has been observed that the empirical pmf $\hat{p}(m)$ of m follows

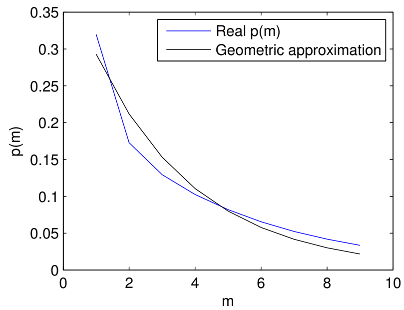


Fig. 3. Central term of first digit distribution for an exponential variable after single and double quantization.

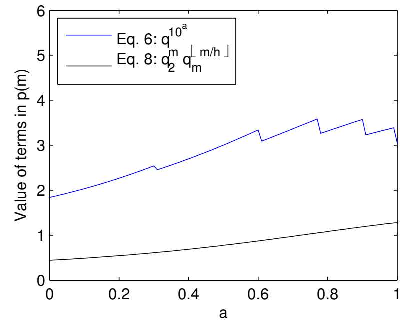


Fig. 4. Central term of first digit distribution for an exponential variable after single and double quantization.

the generalized equation

$$p(m) = K \log_{10}\left(1 + \frac{1}{\alpha + m^\beta}\right), \text{ with } m = 1, \ldots, 9. \quad (3)$$

According to the pmf deviation with respect to the above equation, it is possible to detect whether m has been generated from $\mathbf{Y}_{\Delta,n}(i,j)$ or $\mathbf{Y}_{\Delta,m}(i,j)$, with $n \neq m$. In fact, whenever $n = 1$, the pmf $\hat{p}(m)$ computed from $\mathbf{Y}_{\Delta,n}(i,j)$ satisfies the Benford's equation quite accurately; in case $n > 1$, the distribution $\hat{p}(m)$ deviates from $p(m)$ depending on the number of coding stages n.

It is possible to provide an analytical explanation for this by modeling the absolute values of the quantized coefficients $y_1 = |\mathbf{Y}_{\Delta_1,1}(i,j)|$ via a geometric variable with parameter q (we omit the indexes (i,j) for the sake of conciseness).

As a matter of fact, the probability of the FD $m = \text{FD}(y_1)$ is

$$\begin{aligned} \hat{p}_1(m) &= P[\text{FD}(y_1) = m] \\ &= \sum_{k=0}^{+\infty} P\left[\text{FD}(y_1) = m, 10^k \leq y_1 < 10^{k+1}\right] \quad (4) \\ &= \sum_{k=0}^{+\infty} q^{10^k\ m}\left(1 - q^{10^k}\right). \end{aligned}$$

Assuming that $q^{10^k} \to 0$ rapidly for small values of k and that $(1 - q^{10^k}) \to 1$, it is possible to write

$$\hat{p}_1(m) \simeq (1-q)\ q^m + (1-q^{10})\ q^{10m}. \quad (5)$$

In this case, eq. (5) can be approximated by a geometric variable (Figure 3) with parameter $q_m < q$ (usually 15% lower).

It is possible to demonstrate that standard Benford's law (i.e., $\alpha = 0$ and $\beta = 1$) holds for any geometric variable. More precisely, Benford's law proves to be valid for variables m whose logarithm $\log_{10}(m)$ present a uniform distribution. As for a geometric variable m with parameter q_m,

$$P[\log_{10} m \leq a] = \begin{cases} 0 & \text{if } a < 0 \\ 1 - q_m\ q_m^{10^a}/(q_m - q_m^{10}) & \text{if } 0 \leq a < 1 \\ 1 & \text{otherwise.} \end{cases} \quad (6)$$

which proves to be linear since for small values of q_m the curve $q_m^{10^a}$ is approximately linear. As a consequence, $P[\log_{10} m \leq a]$ proves to be the distribution of a uniform random variable.

Let us suppose that a second quantization is operated on the reconstructed coefficients $\mathbf{Y}_{\Delta_1,1}(i,j) \cdot \Delta_1$ with quantization step Δ_2. In this analysis we avoid the trivial cases $\Delta_2 = s\Delta_1$ or $\Delta_1 = s\Delta_2$ for $s \in \mathbb{N}$. In case $s = 1$, the second quantization has no effect, while in case $s > 1$ the distribution of $y_2 = |\mathbf{Y}_{\Delta_2,2}(i,j)|$ is geometric as well with parameter $q_2 = q_m^s$.

However, in case $\Delta_2 = s\Delta_1 + r$, $r \neq 0$ where $\Delta_1 = h \cdot r$, $h \in \mathbb{N}, h > 1$, the resulting pmf of the quantized levels y_2 is

$$\hat{p}_2(y) = q_2^y\ (1 - q_2) q_m^{\lfloor y/h \rfloor}. \quad (7)$$

Note that for the log-FD distribution ($\log(m_2) = \log(\text{FD}(y_2))$) we have the distribution

$$\hat{p}_2(m) \simeq K\ q_2^m (1 - q_2) q_m^{\lfloor m/h \rfloor} \quad (8)$$

where K is a normalizing factor. The related equation of first digits shows that $p_2(m)$ can not be approximated by a linear function since the oscillating elements evidenced in eq. (8) makes the log-FD distribution non linear, as Figure 4 shows. The figure reports the main term of the probability distribution function for the FDs of $\mathbf{Y}_{\Delta_1,1}$ and $\mathbf{Y}_{\Delta_2,2}$. It is possible to notice that the curve presents oscillations depending on the redistribution of coefficients among the different quantization bins.

Similarly, in case $\Delta_1 = s\Delta_2 + r$ with $\Delta_2 = h \cdot r$, $h \in \mathbb{N}, h > 1$, we also obtain an oscillating probability distribution

$$\hat{p}_2(y) = \sum_{t=0}^{+\infty} q^t\ (1 - q)\ \mathcal{I}(y == t\ s + \lfloor t/h \rfloor)). \quad (9)$$

which can not be linear since in eq. (9) probability is scaled with factor q every s values of y. An intuitive demonstration is provided in Figure 5, where the integer values of the first quantization are mapped into integer values of the second quantization. The resulting distribution is no longer a geometric random variable since the remapping is non uniform.

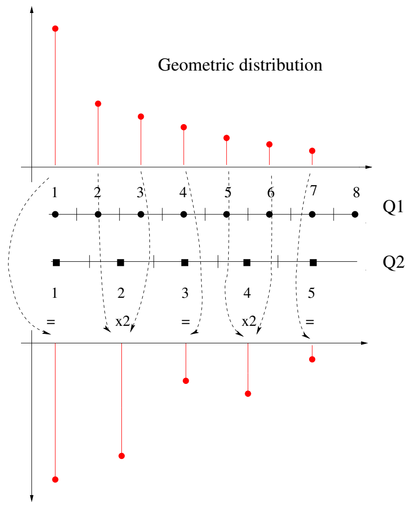


Fig. 5. Example of double quantization for a geometric variable.

III. The Proposed Detection Algorithm

The estimation of the number of coding stages N requires selecting a set of robust features that present a strong correlation with the traces left by quantization. In this first analysis we limited the number of compression stages to 3, but this limitation can be extended later with a more complex classifier. Previous works, focused on JPEG compression, relied on the analysis of the pmf of the FD m for a subset of spatial frequencies. In [12], 20 spatial frequencies were considered, leading to feature vectors of 180 elements. Other approaches considered the relative difference between the actual pmf and the Benford's equation, i.e. $\chi(m) = (p(m) - \hat{p}(m))/\hat{p}(m)$ [13].

In our approach, we work with the coefficients of a 4×4 DCT blockwise transform applied to residual blocks $\mathbf{E}_n$. The estimate of the pmf of the first digit m consists in the 9 bins histogram of the FD for each coefficient. This fact potentially gives us a 144 dimensions feature vector. However our aim is to reduce the size of the feature vector by properly selecting those features that are extremely sensible to the number of compression stages. For this reason our approach considers only the first 7 spatial frequencies (by reading the 4×4 matrix in zig-zag mode) and computes, for each of them, the histogram of the FD. This gives us a 63 dimensions feature vector, which is actually almost a third of that considered in [12].

The choice of using only a subset of the given coefficients is justified by the fact that by increasing the spatial frequency, we obtain more and more values quantized to zero. These coefficients obviously bring less information, and for this reason they can be discarded before the classification process.

In the design of the classifier a crucial role has been played by the optimization of the SVM classifier. Usually, multi-class SVM classifiers are decomposed into binary SVM classifier (one versus the others) which are then recombined into a single multi-class. In our approach we adopted the same strategy, but the optimization process was carried on taking into account the complete multi-class problem. More precisely, we generated an extensive set of sequences coded a different number of times (up to N). The different feature arrays $\mathbf{v}_i$ of the training set are divided into N different classes $\mathcal{C}_n$ such that $\mathcal{C}_n$ includes all the feature arrays for sequences coded n times. For each sequence, the array of features were computed and the optimization of the parameter of SVM kernels was done minimizing the worst precision/performance computed within the single set $\mathcal{C}_n$.

Given the quality factor of the last compression stage, which can be extracted from the available bitstream, it is possible to build a set of K binary SVM classifiers, $\mathcal{S}_k$ ($k = 1, \ldots, K$). In designing $\mathcal{S}_k$, we adopted the radial basis exponential kernel

$$K(\mathbf{v}_i, \mathbf{v}_j) = \exp\left(-\gamma_k \|\mathbf{v}_i - \mathbf{v}_j\|^2\right), \tag{10}$$

where the parameter γ_k is found in the training phase. The training phase for the classifier $\mathcal{S}_k$ consisted in finding the value for γ_k that maximizes the performance of the classifier. The training set of arrays $\mathbf{v}_i$ is randomly partitioned in two sets: an computation set (where the vectors and parameters are optimized given an assigned value for γ_k) and the verification set (where the performance of the classifier is tested). Note that these two sets differ from the final test set from which the results reported in Section IV were obtained. This procedure permits a sort of cross-validation of the classifiers. The performance of the classifier $\mathcal{S}_k$ is parameterized computing its recall (correctly classified / elements in the set) for each set $\mathcal{C}_n$ and selecting the minimum value. In this way, despite the classifier is binary, it is optimized considering the original multiclass problem.

Each classifier $\mathcal{S}_k$ outputs a confidence value ξ_k that reports the distance from the secant hyperplane (related to slack variables). By using this distance it is possible to evaluate the likelihood of a sequence satisfying a given condition.

A set of possible hypothesis has been considered designing an SVM classifier for each of them. In the following, we will refer to each classifier with the name of the output variable (which indicates which class the analyzed sequence belongs to). Each classifier $w_{i,j}$ tests whether the sequence has been encoded i or j times.

In order to state if a given sequence has been coded i-times we combine the $w_{i,j}$ values for several conditions that have to be tested in different phases.

The output values $w_{i,j}$ are then recombined into three parameters (associated to a supposed number of coding steps)

$$\begin{aligned} D_1 &= w_{1,2} + w_{1,3} - w_{2,1} - w_{2,3} \\ D_2 &= w_{2,1} + w_{2,3} - w_{1,2} - w_{3,2} \\ D_3 &= w_{3,1} + w_{3,2} - w_{1,3} - w_{2,3}, \end{aligned} \tag{11}$$

where the different classifiers are combined together with sign $+$ or $-$ according to the coherence of the descriptor with respect to the number of coding steps associated to D_i. The three parameters lie on a plane (since they are recombination of the same outputs from SVM classifiers), and therefore, one of them can be omitted. The values of D_1 and D_2 span over a

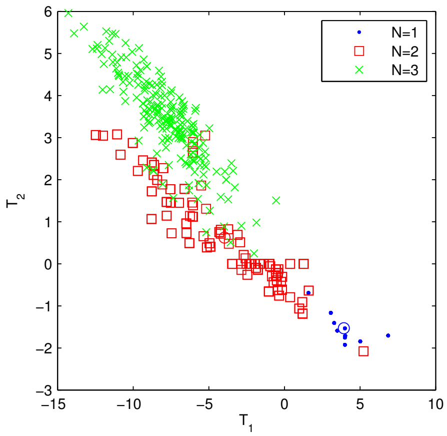


Fig. 6. Parameters T_1 and T_2 used to discriminate $N = 2$ from $N = 3$ for different data sets.

bidimensional interval where ideal data should lie on different corners according to the number of coding steps. Therefore, classification can be performed quite easily by mean of a simple clustering algorithm. In our case we adopted k-means.

However, this classification permits distinguishing very well the video sequence compressed once, but video sequences compressed two or three times proves to be extremely intermingled and a too simplistic clustering algorithm seems to fail.

As a matter of fact, after having classified sequences encoded once, additional discriminative parameters need to be added in order to refine the distinction between $N = 2$ and $N = 3$. More precisely, we considered:

$$\begin{aligned} T_1 &= w_{2,3} - w_{3,2} \\ T_2 &= w_{3,1} - w_{1,3} \end{aligned} \quad (12)$$

where the single classifiers $w_{2,3}$ and $w_{3,2}$ are enforced by combination and the additional evaluation performed by $w_{1,3}$ and $w_{3,1}$. Fig. 6 reports the values of the two different parameters T_1 and T_2 for the different classes.

The proposed parameters permit discriminating the correct number of compression in many cases, even if feature arrays for $N = 2$ are very similar to those related to $N = 3$. Actually the additional classifiers $w_{3,1}$ and $w_{1,3}$ permit making the global classifier more robust since the arrays of SVM support are also derived from class $N = 1$ whose features present a lower similarity with respect to $N = 3$. As a matter of fact, a single SVM classifier could sometimes be unable to discriminate the correct number of compression. However, by combining variables together it is possible to enforce an inaccurate classifier with the results from the other tests.

In the following, the performance of the algorithm will be evaluated.

IV. Experimental results

In order to evaluate the accuracy of the proposed classifier, we selected a training set of 12 video sequences that were coded using a generic hybrid transform-based motion-compensated video codec. In this approach, the adopted transform is a 4×4 integer DCT defined in the standard H.264/AVC. The quantizers have been inherited from the same standard as well. As for the adopted SVM implementation, we adopted the SVMlight software [14] designed by T. Joachims.

TABLE I
Training and test sequences

Training		
foreman	news	mobile
crew	city	salesman
table	paris	flower
irene	bridgeclose	waterfall
Test		
soccer	tempete	

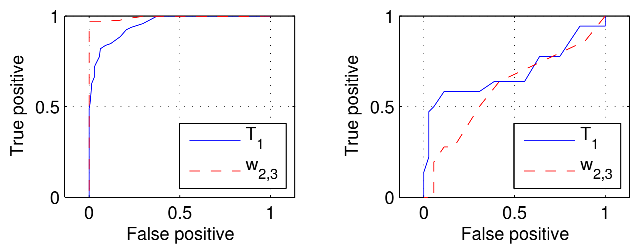
At compression stage n, the quantization parameter QP_n was sampled from a random variable uniformly distributed in the interval $[QP_r - 10, QP_r + 10]$, where QP_r is the average quantization parameter for the sequence.

This assumption is reasonable, since strong variations in QP across the different compression stages would lead to severe quality degradation that would make the resulting video sequence useless. Moreover, we also imposed that the QPs between two consecutive compression stages must differ by at least 2 units since small variations in quantization could be perfectly transparent to the final resulting signal. In the training phase we adopted 14 different realizations of QP chains for the compression of each sequence.

Table II reports the confusion matrix obtained with the proposed method. It is possible to notice that the case of a single compression stage is always correctly identified. Thus, in the case a sequence was compressed once, the proposed method performs very well. As a matter of fact, the proposed approach works well in detecting double compression since it is very accurate in determining whether a video has been coded once or more.

The effectiveness of the approach changes whenever the number of compression is $N > 1$. In this case, with respect to the still images one, discriminating multiple compression is much harder due to motion estimation which scrambles the statistics of coefficients. As a matter of fact, sequences coded twice are not easily distinguishable from those coded three times, and therefore, identifying N proves to be more difficult. However, the proposed approach permits obtaining an accuracy higher than 73%, which compares well with respect to other double compression approaches (e.g., for the approach [10], accuracy is around 70%). Similar conclusions can be drawn considering other double compression detection methods which reaches an accuracy around 92 %.

It is also possible to evaluate the robustness of combining different classifiers together. In Fig. 7, we report the ROC curves that compares the performance of the parameter T_1 with respect to the single parameter $w_{2,3}$ (which follows the classification operated in [10] or in [15]). It is possible to notice that although the performance of $w_{2,3}$ is optimal on the



(a) (b)

Fig. 7. ROC curves comparing T_1 vs. $w_{2,3}$. a) Results on training set b) Results on test set.

TABLE II
CONFUSION MATRIX FOR $QP_N = 25$.

N , N*	**1**	**2**	**3**
1	100 %	0.00 %	0.00 %
2	0.00 %	73.89 %	26.11 %
3	0.00 %	22.22 %	77.78 %

training set (where it has been optimized by the SVM learning routine), its performances dramatically fall when applied to test data. The parameter T_1 limits the accuracy decrement and permit improving both accuracy and recall.

Therefore, it is possible to conclude that the proposed solution works very well as a double compression detector and, moreover, permits revealing the eventuality that some additional compression has been operated previously.

V. CONCLUSIONS

The paper describes a classification strategy that permits detecting the number of video compression stages performed on a single sequence. The approach relies on a set of SVM classifiers applied to features based on the statistics of the first digits of quantized DCT coefficients. The proposed solution performs well with respect to previous double compression detectors and permits revealing whether there have been additional compression stages (more than two). Future work will be devoted to improve the robustness of the approach (making it more accurate as the number of compression stages increases) and to analyze its possible employment in a tampering detection approach. A second set of tests will be devoted to verify the robustness of the approach whenever some editing is performed between two consecutive compressions (frame insertion/removal, interpolation). Moreover, we will analyze the performance of the algorithm in presence of antiforensic attacks aimed at hiding some of the compression stages operated on the video sequence.

ACKNOWLEDGEMENT

This work was partially supported by the project REWIND. The project REWIND acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number:268478.

REFERENCES

[1] Shiguo Lian, Nikolaos Nikolaidis, and Husrev T. Sencar, "Content-based video copy detection - a survey," in *Intelligent Multimedia Analysis for Security Applications*, Husrev T. Sencar, Sergio A. Velastin, Nikolaos Nikolaidis, and Shiguo Lian, Eds., vol. 282 of *Studies in Computational Intelligence*, pp. 253–273. Springer, 2010.

[2] M. C. Stamm and K. J. R. Liu, "Anti-forensics of digital image compression," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1050 – 1065, Sept. 2011.

[3] Weihong Wang and Hany Farid, "Exposing digital forgeries in video by detecting double quantization," in *Proceedings of the 11th ACM workshop on Multimedia and security*, New York, NY, USA, 2009, MM&Sec '09, pp. 39–48, ACM.

[4] Weihong Wang and Hany Farid, "Exposing digital forgeries in video by detecting double MPEG compression," in *MM&Sec*, Sviatoslav Voloshynovskiy, Jana Dittmann, and Jessica J. Fridrich, Eds. 2006, pp. 37–47, ACM.

[5] M.C. Stamm and K.J.R. Liu, "Anti-forensics for frame deletion/addition in mpeg video," in *Proc. of ICASSP 2011*, Prague, Czech Republic, may 2011, pp. 1876 –1879.

[6] W. Luo, M. Wu, and J. Huang, "MPEG recompression detection based on block artifacts," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Mar. 2008, vol. 6819 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*.

[7] Weihong Wang and Hany Farid, "Exposing digital forgeries in interlaced and deinterlaced video," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3-1, pp. 438–449, 2007.

[8] Dandan Liao, Rui Yang, Hongmei Liu, Jian Li, and Jiwu Huang, "Double H.264/AVC compression detection using quantized nonzero AC coefficients," in *SPIE Conference on Media Watermarking, Security, and Forensics*, 2011.

[9] P. Bestagini, A. Allam, S. Milani, M. Tagliasacchi, and S. Tubaro, "Video codec identification," in *Proc. of ICASSP 2012*, Kyoto, Japan, Mar. 25 – 30, 2012, Accepted; available at http://www.rewindproject.eu/.

[10] W. Chen and Y. Q. Shi, "Detection of double mpeg compression based on first digit statistics," in *Proc of IWDW 2008, LNCS 5450*, 2009, vol. 5450, pp. 16–30.

[11] S. Milani, M. Tagliasacchi, and M. Tubaro, "Discriminating multiple jpeg compression using first digit features," in *Proc. of ICASSP 2012*, Kyoto, Japan, Mar. 25 – 30, 2012.

[12] Bin Li, Yun Q. Shi, and Jiwu Huang, "Detecting doubly compressed JPEG images by using mode based first digit features," in *Proc. of MMSP 2008*, Oct. 2008, pp. 730–735.

[13] DongDong Fu, Yun Q. Shi, and Wei Su, "A generalized benfords law for jpeg coefficients and its applications in image forensics," in *Proceedings of SPIE, Volume 6505, Security, Steganography and Watermarking of Multimedia Contents IX*, Jan. 28 – Feb. 1, 2009, vol. 6505, pp. 39–48.

[14] T. Joachims, *Making large-Scale SVM Learning Practical*, MIT-Press, 1999.

[15] Jan Lukás and Jessica Fridrich, "Estimation of primary quantization matrix in double compressed jpeg images," in *Proc. of DFRWS*, 2003.