Joint detection of full-frame linear filtering and JPEG compression in digital images

V. Conotter $^{\#1}$, P. Comesaña $^{\#2}$, F. Pérez-González $^{\#*3}$

# Signal Theory and Communications Department, University of Vigo, Spain
** Gradiant (Galician Research and Development Center in Advanced Telecommunications), Vigo, Spain*
[1] conotter@gts.uvigo.es
[2] pcomesan@gts.uvigo.es
[3] fperez@gts.uvigo.es

***Abstract*—Image forensics aims at detecting clues regarding the history of a digital image. However, many forensic techniques present some limitations in real life scenarios, where the application of post-processing operators may alter the characteristic footprints exploited by forensic tools. In this paper we study the combination of JPEG compression and full-frame linear filtering, analyzing their impact on the statistical distribution of the Discrete Cosine Transform (DCT) coefficients of the image. We extract characteristic features from the DCT distributions and build an effective classifier able to jointly disclose the applied compression quality factor and the filter kernel. The proposed technique is computationally efficient and effective, as demonstrated by the extensive experimental analysis.**

I. Introduction

Increasingly sophisticated and cheap digital technologies, coupled with the wide spread of Internet, have made it possible to easily acquire, copy, share and manipulate digital images with very little effort. This has led to problematic issues concerning multimedia authenticity and reliability. Digital image forensics has emerged as a new discipline to help regaining some trust in digital photographs, by detecting clues about the history of a content [1]. In the absence of any form of digital watermarks or signatures, this field works on the assumption that most forms of tampering will disturb some properties of the image. To the extent that these perturbations can be quantified and detected, they can be used to authenticate a photo. Techniques in digital forensics can be categorized as: (1) Pixel-based, detecting statistical patterns at the pixel level [2]; (2) Format-based, detecting statistical patterns specific to an image compression format (e.g., JPEG or GIF) [3]; (3) Camera-based, exploiting artifacts introduced by the camera lens, sensor or on-chip post-processing [4]; (4) Physically-based, modeling and measuring the interactions between physical objects, light and the camera [5]; and (5) Geometry-based, exploiting the principles of image formation as governed by projective geometry [6].

A vast amount of forensic techniques have been proposed in the literature so far, mainly aimed at detecting the specific processing operators an image went through [7], but little attention has been paid to collecting information about processing chains of operators. Indeed, the application of multiple post-processing may seriously affect the forensic analysis of a content, altering or even erasing the characteristic inherent properties left by previous processing and exploited by existing forensic detectors [8].

Since the JPEG standard describes the image compression scheme that is currently most widely used, this compression represents a forensically interesting operation to be studied. Indeed, several forensic works in the literature exploit the characteristic footprint left in the DCT coefficients distribution of an image during compression, aiming at discovering traces of previous JPEG compression and estimating the employed quantization step [3][9][10]. Recent results show that even multiple instances of JPEG compression can be detected [11][12][13]. Unfortunately, the above mentioned techniques have some limitations in real life scenarios, where chains of operators may have been applied to the content [14]. In our previous work [15], we demonstrated that linear image processing, such as filtering, often applied to the entire image (full-frame) as post processing for image enhancement, but possibly also for forensic footprints removal, may alter the characteristic artifacts introduced by the JPEG compression scheme. In [15] an accurate mathematical model was proposed to theoretically characterize the probability distribution of the DCT coefficients of JPEG full-frame linearly filtered images. Assuming the quantization step to be known, such knowledge can be exploited in order to retrieve the applied filter kernel by measuring the difference between the derived models (each model depends on the applied filter kernel) and the actual distribution of a to-be-tested image.

In this work, we relax the quite strong assumption we made in [15] about the knowledge of the compression quantization step and propose a simple yet very effective forensic tool which is now able to jointly detect the filter kernel and the quality factor of the JPEG compression that have been applied to an image, so to retrieve the entire processing history of the content. We extract a set of significant features of the DCT distributions of the compressed and filtered image and build a linear classifier able to effectively discriminate different combinations of filtering and compressions. To the best of our knowledge, the presented work represents a first approach to

WIFS'2013, November 18-21, 2013, Guangzhou, China.
ISBN 978-1-4673-5593-3

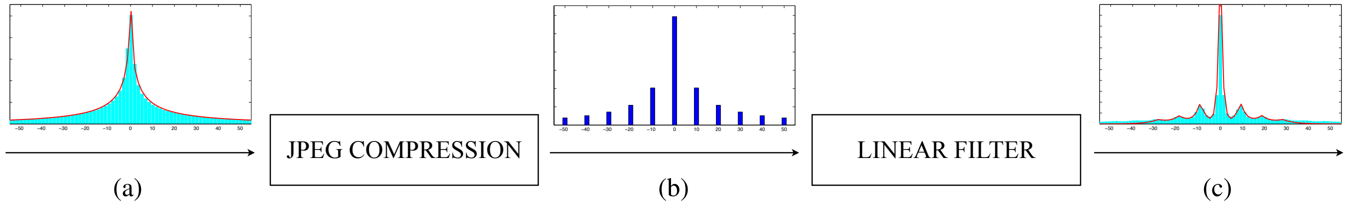


Fig. 1: Shown is the block scheme of the considered application scenario. Panel (a) shows the original DCT histogram for frequency $(0,1)$ for uncompressed images and its curve fitting. In panel (b), the same distribution after JPEG quantization with step $q(0,1) = 10$ is presented. Panel (c) shows the given distribution when a linear average filter of size 3×3 has been further applied to the image, together with the derived model for such distribution (red line).

jointly disclose traces of chain processing operators such as JPEG compression and full-frame linear filtering.

The structure of the paper is the following: in Section II we describe the considered processing operators and their effect on the probability distributions of the DCT coefficients of an image, while in Section III we present our forensic approach to jointly disclose traces of compression and linear filtering; Section IV deals with the experimental results, and finally in Section V some conclusions are drawn.

II. Application scenario

In this work, we study the case of JPEG compressed images that have been further post-processed with a full-frame linear filter, as shown in Fig.1. The main goal is to disclose the quality factor used for compression, together with the filter kernel employed to post-process the JPEG image.

Nowadays, JPEG standard is the most widely used compression scheme for digital images. In its most commonly used format, it is well known to be a lossy scheme, i.e., some information is lost during the process, mainly due to a quantization operation. JPEG compression is based on a 8×8 non-overlapping block-by-block DCT frequency decomposition. In order to quantize each DCT coefficient, a 8×8 quantization table is used, consisting of 64 integer-valued quantization steps $q(i,j)$, where $(i,j) \in \{0,\dots,7\}$ indicates to the (i,j)-th frequency within each block. The JPEG standard provides standard quantization tables, which correspond to specific compression Quality Factors QF and have to be properly selected in order to achieve a good trade-off between visual quality and compression rate. The quantized DCT coefficients are finally entropy-encoded (Huffman coding) and stored in the JPEG file format. Decompression can be performed by applying all the steps in reverse order.

The JPEG compression scheme forces the DCT coefficients to be mapped to integer multiples of the quantization step, resulting in specific artifacts in the frequency domain. In Fig. 1, panel (a) shows the histogram of the DCT coefficients at frequency $(0,1)$ collected from a set of 1338 of un-compressed images [16], while panel (b) depicts the distribution of the same data after quantization, with $q(0,1) = 10$. It becomes clear that the structure of such histogram is related to the employed quantization step. For the sake of presentation we disregarded in the picture the round-off and truncation errors, introduced in the pixel domain by the compression scheme.

However, certain post-processing is very likely to be applied to the compressed image (e.g., to remove blocking artifacts, or to remove the JPEG quantization footprints) and this may perturb or even delete the characteristic artifacts left by JPEG compression. In such a scenario, existing forensic tools which exploit JPEG footprints to disclose the compression history of a content may become ineffective. In this work we study the case where a full-frame linear filtering is applied to a JPEG compressed image. Linear filtering is a very common and useful tool applied for image enhancement, such as edge sharpening, noise removal, illumination correction and deblurring. The convolution between an image and a linear filter kernel produces a filtered image whose pixels are the weighted sum of a certain number of neighboring pixels. Panel (c) in Fig. 1 shows the histogram of the DCT frequency coefficients of panel (b) after filtering the image in the pixel domain with a linear Average 3×3 filter. The characteristics of the histogram of the quantized coefficients are clearly perturbed, but new patterns appear, depending both on the employed compression quality factor and the filter kernel.

In our previous work [15], we studied such artifacts in order to estimate the filter operator an image has gone through, assuming that the compression quality factor (specifically, the quantization table) used at the encoder is known by the estimator. We analyzed the DCT statistical properties of a JPEG compressed and linearly-filtered image and mathematically established the relationship between the DCT coefficients before and after filtering, as follows:

$$Y^{i_8,j_8}(i,j) \quad = \quad \gamma_{i,j} \cdot X^{i_8,j_8}(i,j) + N^{i_8,j_8}(i,j). \qquad (1)$$

where X and Y are the $L_1 \times L_2$ size JPEG image and its filtered version, respectively, in the 8×8 DCT domain. So $\cdot^{i_8,j_8}(i,j)$ stands for the (i,j) DCT coefficient at the (i_8,j_8) block, where $i,j \in \{0,\dots,7\}$, $i_8 \in \{0,\dots,(L_1/8)-1\}$, and $j_8 \in \{0,\dots,(L_2/8)-1\}$.[1] Moreover, $\gamma_{i,j}$ and $N^{i_8,j_8}(i,j) \in \mathbb{R}$ are a frequency dependent scaling factor and noise term, respectively. By exploiting the statistical properties of the distribution of DCT coefficients in JPEG images $X^{i_8,j_8}(i,j)$,

[1] For the sake of simplicity, we will assume L_1 and L_2 to be integer multiples of 8.

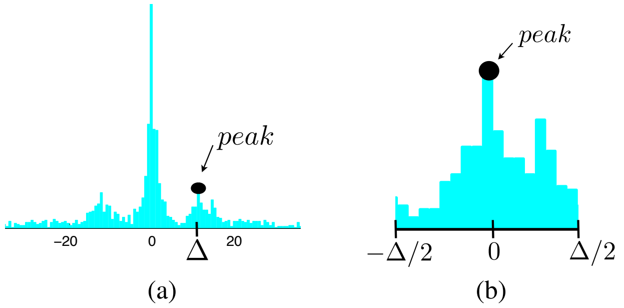


Fig. 2: In panel (a) is the distribution of DCT coefficient at frequency $(0,1)$ of an image that has been compressed with $q = 10$ and linearly filtered with a linear average filter of size 3×3. The black dot marks the first positive peak in the distribution, located at Δ. Panel (b) shows the modulo-Δ reduced version of the histogram.

we derived a model to theoretically characterize the probability distribution of the DCT coefficients of a JPEG image filtered with a given filter kernel. Note that the effect of the filtering is not purely multiplicative on the transform domain, mainly due to two reasons: 1) we are using the block-based DCT, not the Fourier Transform, 2) due to the block-based nature of the considered transform, the full-frame filtering introduces interference from neighboring blocks, which tends to scatter the original comb-shaped histogram. Indeed, we have shown in [15] that the characteristic peaks in the DCT distribution of JPEG compressed and filtered images are located at integer multiples of scaled versions of the employed quantization steps, namely, $\gamma_{i,j} \cdot q(i,j)$ plus the mean of the noise term in (1). Finally, in order to identify the applied linear filter, a distinguishability measure (the chi-square measure) was employed to quantify the difference between the derived models (each model depends on the applied filter kernel) and the actual histogram of the tested image. The minimum value of this distinguishability measure was taken as evidence for the identification of the applied filter. We refer to [15] for further details.

In this work we remove the assumption on the quantization step to be fixed and known, proposing a novel forensic technique able to detect clues about the history of the content, i.e., in this context, the compression quality factor and the filter kernel that have been applied to the image. This technique is based on a set of novel metrics, which, to the best of authors' knowledge, are used for the first time in forensic applications.

III. Proposed approach

As described in Section II, we have at hand a well-established model for the probability distribution of the DCT coefficients of a JPEG compressed and linearly filtered image. The characteristic peaks of the DCT distribution, as shown in Fig. 1(c) and formalized in (1), are located at integer multiples of a scaled version of the corresponding quantization step. It becomes clear that both the location and the shape of these peaks strongly depend on the applied compression factor and the kernel used to filter the image.

In this work, we exploit such knowledge for defining a set of significant features that characterize the compressed and filtered images. Given a to-be-tested JPEG-compressed and filtered image, let us define $\mathbf{Y}(i,j)$ as the vector containing the 8×8 DCT frequency (i,j) coefficients and whose k-th element is $Y^{\lfloor 8k/L_2 \rfloor, k-L_2\lfloor 8k/L_2 \rfloor}(i,j)$; furthermore, $H(\mathbf{w},\delta)$ denotes the histogram of a vector $\mathbf{w}$ with bin width δ, $H(\mathbf{w},\delta,k)$ its value at bin k, and $\Delta(i,j) > 0$ the location of the characteristic peaks in $H(|\mathbf{Y}(i,j)|,\delta(i,j))$. Note that the peak location depends on the analyzed (i,j)-th DCT frequency coefficient. For the sake of notational simplicity, in the following we will avoid to explicitly express the frequency dependency of both the histogram and peak location, and will drop the indices (i,j). Fig. 2(a) serves as an example to show the detected peak (black dot) in the distribution of the $(0,1)$ DCT coefficients of an image that has been quantized with step $q(0,1) = 10$ and post-processed with a linear average filter of size 3×3. Different peak detection algorithms may be employed. In the current work we seek those points whose neighbors on both sides are smaller by, at least, a given threshold T; then, based on the typical monotonically decreasing nature of the DCT coefficients distribution about the origin, from the resulting set we get the point with the largest histogram value. Mathematically,

$$\begin{aligned}\mathcal{S} &= \{k \in \mathbb{N}^+ : H(|\mathbf{Y}|,\delta,k) \geq H(|\mathbf{Y}|,\delta,k+1) + T, \\ &\qquad H(|\mathbf{Y}|,\delta,k) \geq H(|\mathbf{Y}|,\delta,k-1) + T\}, \\ \Delta &= \delta \arg\max_{k \in \mathcal{S}} H(|\mathbf{Y}|,\delta,k).\end{aligned}$$

This algorithm has been proven to be computationally very efficient and accurate.

Once we have determined the peak location Δ, we are interested in computing a peakiness measure of the histogram of $\mathbf{Y}$ around the integer multiples of Δ. In order to do that, we define the modulo-Δ reduced version of $\mathbf{Y}$, specifically

$$\tilde{\mathbf{Y}} \triangleq \mathbf{Y} \mathrm{mod} \Delta,$$

which returns values in $(-\Delta/2, \Delta/2]^{L_1 L_2/64}$, and compute the empirical variance of $\tilde{\mathbf{Y}}$, i.e.,

$$\sigma^2_{\tilde{Y}} = \frac{64}{L_1 L_2} \sum_{k_1=0}^{L_1/8-1} \sum_{k_2=0}^{L_2/8-1} \left(\tilde{Y}^{k_1,k_2}\right)^2;$$

please note that we have assumed $\tilde{\mathbf{Y}}$ to be zero-mean, based on the symmetry of X with respect to the origin. Figure 2(b) shows the result of modulo-Δ reduction of the histogram from panel (a). Given $\sigma^2_{\tilde{Y}}$, we measure the peakiness of the histogram of $\mathbf{Y}$ as

$$\beta = \frac{\sigma^2_{\tilde{Y}}}{\Delta^2}. \tag{2}$$

It is clear that if the values in $\mathbf{Y}$ are clustered around the integer multiples of Δ, then the values in $\tilde{\mathbf{Y}}$, which take values in $(-\Delta/2, \Delta/2]$, will be clustered around the origin; therefore the empirical variance of the latter will be small in comparison

feature	description
Δ	peak location
β	measure for histogram peakiness (2)
$\lvert\mathcal{Z}\rvert$	number of zero DCT coefficients
σ_Y^2	empirical variance of $\mathbf{Y}$

TABLE I: Features used to train the classifier, extracted from the histogram of the DCT coefficients for each frequency.

with Δ^2, and consequently a low value of β will be a clue of a peaky histogram of $\mathbf{Y}$.

Typically, in the quantization tables provided by the JPEG standard, larger quantization steps are employed for high-frequency DCT coefficients, since they have little visual significance. As a consequence, in some images the coefficient values of certain high-frequency DCT subbands might be all quantized to zero. According to [17], the number of zero quantized DCT coefficients mainly contributes to the compression performance. Correspondingly, no peaks would be detected in this case. However, it could also be the case that the image just does not contain high frequencies, because it was, for example, low-pass filtered. Let us define $\mathcal{Z}_{i,j} = \{i_8, j_8 : Y^{i_8,j_8}(i,j) = 0\}$ as the set of indices of zero-quantized coefficients for each DCT frequency subband (i,j); in the sequel we will avoid the (i,j) indices for the sake of notational simplicity. We compute the cardinality $|\mathcal{Z}|$ of such a set and take it as another significative feature to characterize the behavior of DCT coefficients at high frequencies.

Similarly, we calculate also the empirical variance of $\mathbf{Y}$ as

$$\sigma_Y^2 = \frac{64}{L_1 L_2} \sum_{k_1=0}^{L_1/8-1} \sum_{k_2=0}^{L_2/8-1} \left(Y^{k_1,k_2}\right)^2 ,$$

in order to take into account the general behavior of each frequency. This value will depend on the nature of the image itself, on the applied compression and on the employed filter.

These features, which are extracted for each AC 8×8-DCT frequency and are summarized in Table I, take into account the location of the peaks, the peakiness of the histogram, and the variance of the DCT coefficients of the given frequency. Therefore, they are expected to summarize the properties of the considered histogram. Consequently, a linear kernel Support Vector Machine (SVM) [18], fed with these sets of features, i.e., with a vector of dimensionality $4 \times 63 = 252$, is used to classify the considered images in terms of their compression factor and undergone linear filter.

IV. Experimental results

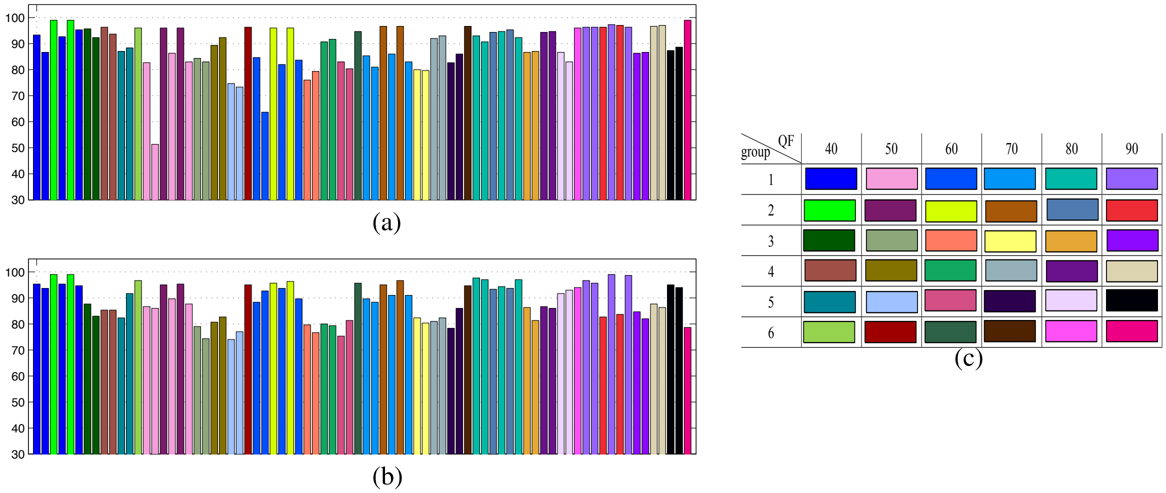
The main goal of the proposed approach is to jointly disclose the compression quality factor and the filter kernel, if any, applied to an image. We carry out an extensive experimental analysis in order to verify the effectiveness of the proposed approach.

We start considering 600 uncompressed images present in the UCID dataset [16], and compress them using compression quality factors QF $\in \{40, 50, 60, 70, 80, 90\}$. We then convolve the compressed images with a filter kernel chosen among a fixed set of linear filters. In these experiments, we selected both low-pass (LP) and high-pass (HP) filters (e.g., Moving Average, Gaussian, Laplacian), with different settings for the window size, the variance σ_F^2 or the scale parameter α, as reported in Table II. Note that also the case when the identity filter (i.e., no filter at all) has been applied is taken into account. As in [15], we notice that some of the selected filters present a high similarity in their frequency response, thus being difficult to be distinguished. This issue is not specific of the presented framework, but a general limitation. In light of this, we group the filters based on the similarity of their frequency responses for the analyzed DCT coefficients and allow them to be classified within the same class. Table II shows groups based on the similarity of the linear filters selected to be in the dictionary.

group 1	1. LP Average [3×3]
	2. LP Average [5×5]
	4. LP Gaussian [3×3], σ_F^2 = 1
	6. LP Gaussian [5×5], σ_F^2 = 1
group 2	3. LP Gaussian [3×3], σ_F^2 = 0.5
	5. LP Gaussian [3×3], σ_F^2 = 0.5
group 3	7. HP Laplacian, α = 0.2
	8. HP Laplacian, α = 0.7
group 5	9. LP Laplacian, α = 0.2
	10. LP Laplacian, α = 0.7
group 4	11. HP Average [3×3]
	12. HP Average [5×5]
group 6	13. Identity filter

TABLE II: Filters selected to be in the filter dictionary, grouped according to the similarity of their frequency response.

The image dataset is then split into two halves: one for training and one for testing. In particular, in our experiments we will have 300 images that will be applying each possible combination of JPEG compression (6 quality factors QF) and linear filtering (13 filters). As a result, we have a total of $300 \times 13 \times 6 = 23400$ images, both for training and testing. For each image, the set of features in Table I is extracted and the classifier is trained with them. Given the 6 groups used for representing the filters and the 6 considered compression factors QF, a total of 36 classes will have to be discriminated by the SVM classifier. The performance of the proposed algorithm is verified in terms of the percentage of correct classification over the image database, jointly for JPEG compression and image filtering. The accuracy of correct classification is reported in Fig. 3(a). Results are reported as bar graphs, where each bar represents one of the combinations of the 13 filters and the 6 chosen quality factors. For example, the first 13 bars, starting from the left, correspond to the classification of images that have been filtered with each of the filters in the dictionary and compressed with quality factor QF= 40. The next 13 bars refer to images filtered and compressed with QF= 50, and so on until QF= 90. Colors



(a)

(b)

(c)

Fig. 3: Panel (a) shows the classification accuracy when each image in the database has been filtered with each of the 13 selected filters and compressed with different QFs. Each color identifies the class considered during classification (6 groups of filters with 6 selected quality factors, for a total of 36 classes). Similarly, in Panel (b) is the classification accuracy, but with a further post-processing JPEG compression with QF$=90$. Panel (c) shows the legend.

have been chosen in order to identify the 36 classes considered by the SVM classifier. The matrix in Fig. 3(c) explains this color choice. The overall accuracy is 89.2%, indicating that the proposed method is quite effective in jointly disclosing the applied JPEG quality factor and linear filter kernel.

If we discarded the frequency response similarity among the filters, so to have a total of 78 classes (13 filters and 6 different compression factors) we would be able to reach an average accuracy of just 74.5%, thus confirming the confusion in the classification introduced by the filters similarity.

The proposed forensic approach results to be very versatile and could be employed for slightly different purposes. As an example, it could be also used for detecting only the applied compression factor, by discarding the detection of the linear filtering. In such a scenario, the number of classes narrows down to 6, corresponding to the considered quality factors QF $\in \{40, 50, 60, 70, 80, 90\}$. Given the same model computed in the above mentioned study case when 36 classes were considered (i.e., exactly the same SVM is used), the system is now able to reach an average accuracy of 89.4%. Please note that the obtained value is almost identical to the accuracy achieved for the joint estimation of JPEG QF and linear filter kernel. Therefore, no advantage is obtained by marginalizing the classification with respect to the filter kernel estimation; this seems to indicate that the used SVM is indeed exploiting with a similar significance the impact of both the QF and the filter kernel, i.e., the consideration of both parameters is indeed required even if only one of them wants to be estimated.

As a particularly interesting case, we can also check the performance of the proposed scheme when the input images have been only compressed, i.e., when the identity filter (or no filter) has been applied. In this scenario, we compress each of the selected 600 images with different quality factors, QF $\in \{30, 40, 50, 60, 70, 80, 90, 100\}$. Then, the features in Table I are extracted, and half of them are selected to train a new classifier, with 8 classes. The reached overall classification accuracy is 99.1%. This test demonstrates that the proposed method not only is the first successful approach to disclose both compression and filtering together, but it could be effectively applied to detect only compression when no linear filtering has been applied to the image. Indeed, the obtained accuracy is comparable with the performance given by state-of-the-art forensic techniques [3][9][10], yet being very simple and computationally efficient. It is worth noticing that techniques in the literature designed to detect the applied compression factor in JPEG images dramatically decrease their performance when a linear filtering is applied as post-processing.

As a last experiment, we want to test the robustness of the proposed approach with respect to double compression. The study case here considered will be the same as described in Fig. 1, but with the addition of a further final operator after filtering, i.e., another JPEG compression. From a practical point of view, this problem makes sense in a number of application scenarios. Just as an example, we can mention the case where an image seller tries to sell a low-quality (small JPEG QF) image as a high-quality one (with large JPEG QF). Since schemes are already available in the literature for detecting double JPEG compression [11][12][13], in order to succeed he/she needs to remove the low-quality JPEG footprints; using a linear filter is a plausible choice. Then, the image is re-compressed with the new larger QF. To the best of our knowledge, this scenario has not been considered previously in the literature. As we have done in the previous tests, we compress each image in the dataset with each of the selected QF $\in \{40, 50, 60, 70, 80, 90\}$ and filter them

Detect	# filters	QF range	# classes SVM	Accuracy
F_{groups} + QF	13	$\{40, ..., 90\}$	36	89.2%
F + QF	13	$\{40, ..., 90\}$	78	74.5 %
QF	13	$\{40, ..., 90\}$	6	89.4%
QF	1	$\{30, ..., 100\}$	8	99.1%
F_{groups} + QF *	13	$\{40, ..., 90\}$	36	88.6%

* double compressed with second $\text{QF}_2 = 90$

TABLE III: Experimental results in terms of classification accuracy obtained in different application scenarios.

with each one of the selected filters in Table II. Next, each image is further compressed using a compression quality factor $\text{QF}_2 = 90$. The same features as in Table I are extracted and given as input to the classifier. The accuracy of correct classification is reported in Fig. 3(b). We remind that each bar in the graph represents one of the combination of the 13 filters and the 6 chosen quality factors. For example, the first 13 bars, starting from the left, correspond to the classification of images that have been filtered with each of the filters in the dictionary and compressed with quality factors $\text{QF}_1 = 40$ and $\text{QF}_2 = 90$. Colors identify the 36 classes here considered by the SVM classifier. The average accuracy that the system reaches is 88.6%, proving that the extracted features are significant for classification even in presence of a second compression.

The experimental analysis we conducted and the obtained results are summarized in Table III. The first column contains the goal of the experiment: to jointly detect the applied filter kernel and the compression factor, either taking into account the filters groups in Table II (F_{groups} + QF) or not (F + QF), or to detect only the quality factor QF. The second column contains the number of tested and applied linear filters, while the third column contains the range of the applied compression factors (with step 10). The number of the classes considered during classification via SVM is reported in the fourth column, while the average total accuracy is shown in the last column.

The presented experimental analysis demonstrates the effectiveness and the robustness of the proposed forensic tool, but also its versatility to be applied for different goals, as the detection of compression both in JPEG images and in JPEG linearly filtered images, or to jointly disclose the applied chain of operators (here, JPEG compression and full-frame linear filtering).

V. Conclusions

We have presented a novel and computationally efficient forensic technique able to jointly detect the applied compression factor and the undergone linear filter kernel in digital images. Compared to [15], we do not make any assumptions about the knowledge of the compression quantization step, which is here blindly disclosed together with the filter kernel. Four characteristic features of the DCT histogram are proposed; they are used for performing the compression and filter detection by using a SVM classifier. Experimental results demonstrated the effectiveness and the versatility of the proposed approach. This framework may be regarded as a first approach to analyze and successfully classify JPEG images that have been further post-processed with a full-frame linear filtering, opening the door to investigate the effects of other chain operators.

Acknowledgment

Research supported by the European Union under project REWIND (Grant Agreement Number 268478), the European Regional Development Fund (ERDF) and the Spanish Government under projects DYNACS (TEC2010-21245-C02-02/TCM) and COMONSENS (CONSOLIDER-INGENIO 2010 CSD2008-00010), and the Galician Regional Government under projects ”Consolidation of Research Units” 2009/62, 2010/85 and SCALLOPS (10PXIB322231PR).

References

[1] E. Delp and M. Wu, “Digital forensics,” *IEEE Signal Processing Magazine*, vol. 2, no. 26, pp. 14–15, 2009.

[2] J. Rhedi, W. Taktak, and J.-L. Dugelay, “Digital image forensics: a booklet for beginners.” *Multimedia Application Tools*, vol. 51, pp. 133–162, 2011.

[3] R. Neelamani, R. de Queiroz, Z. Fan, S. Dash, and R. Baraniuk, “JPEG compression history estimation for color images,” *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1365–1378, 2006.

[4] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, “Determining image origin and integrity using sensor noise,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, 2008.

[5] J. O’Brien and H. Farid, “Exposing photo manipulation with inconsistent reflections,” *ACM Transactions on Graphics*, vol. 1, no. 31, pp. 1–11, 2012.

[6] V. Conotter, J. O’Brien, and H. Farid, “Exposing digital forgeries in ballistic motion,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 283–296, 2012.

[7] A. Piva, “An overview on image forensics,” *ISRN Signal Processing*, vol. 2013, p. 22 Article ID 496701, 2013.

[8] M. Kirchner and R. Böhme, “Hiding traces of resampling in digital images,” *15th International Conference on Multimedia*, vol. 3, no. 4, pp. 582–592, 2008.

[9] W. Luo, J. Huang, and G. Qiu;, “JPEG error analysis and its application to digital image forensics,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 480–491, 2010.

[10] D. Fu, Y. Q. Shi, and W. Su, “A generalized Benford’s law for JPEG coefficients and its applications in image forensics,” *SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents*, vol. 6505, 2007.

[11] T. Bianchi and A. Piva, “Detection of nonaligned double JPEG compression based on integer periodicity maps,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 842–848, 2012.

[12] T. Penvy and J. Friedrich, “Detection of double-compression in JPEG images for applications in steganography,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 247–258, 2008.

[13] S. Milani, M. Tagliasacchi, and S. Tubaro, “Discriminating multiple JPEG compression using first digit features,” *International Conference on Acoustic, Speech ans Signal processing (ICASSP)*, pp. 2253–2256, 2012.

[14] T. Bianchi and A. Piva, “Reverse engineering of double JPEG compression in the presence of image resizing,” *IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 127–132, 2012.

[15] V. Conotter, P. Comesaña, and F. Pérez-González, “Forensic analysis of full-frame linearly filtered JPEG images,” *International Conference on Image Processing (ICIP)*, 2013.

[16] G. Schaefer and M. Stich, “UCID - an uncompressed colour image database,” *SPIE Conference on Storage and Retrieval Methods and Applications for Multimedia*, pp. 472–480, 2004.

[17] J.-Y. Lee and H. W. Park, “A rate control algorithm for DCT-based video coding using simple rate estimation and linear source model,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, pp. 1077–1085, 2005.

[18] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.