Forensic detection of processing operator chains: recovering the history of filtered JPEG images

Valentina Conotter, Pedro Comesaña, and Fernando Pérez-González

***Abstract*—Powerful image editing software is nowadays capable of creating sophisticated and visually compelling fake photographs, thus posing serious issues to the trustworthiness of digital contents as a true representation of reality. Digital image forensics has emerged to help regain some trust in digital images by providing valuable aids in learning the history of an image. Unfortunately, in real scenarios its application is limited, since multiple processing operators are likely to be applied, which alters the characteristic footprints exploited by current forensic tools. In this work, we develop a novel forensic technique that is able to detect chains of operators applied to an image. In particular, we study the combination of JPEG compression and full-frame linear filtering, and derive an accurate mathematical framework to fully characterize the probabilistic distributions of the Discrete Cosine Transform (DCT) coefficients of the quantized and filtered image. We then exploit such knowledge to define a set of features from the DCT distribution and build an effective classifier able to jointly disclose the quality factor of the applied compression and the filter kernel. Extensive experimental analysis illustrates the efficiency and versatility of the proposed approach, which effectively overcomes the state-of-the-art.**

***Index Terms*—Full-frame linear filtering, image forensics, JPEG compression.**

I. Introduction

THE rapid growth and spread of Internet, along with the popularity of advanced digital technologies, grants easy access, manipulation, and distribution of digital media. Nowadays visually compelling and sophisticated photographic fakes pervade nearly every aspect of our society, including media, politics, and advertisement, posing serious issues regarding the authenticity and reliability of digital images as a true representation of reality. In turn, this situation creates an urgent need for further technologies that are able to ensure the trustworthiness of digital contents.

Although digital watermarking [1] is a valuable approach for content protection and integrity verification problems, it has the significant drawback of requiring the watermark to be embedded before any (possibly illegal) processing, thus limiting its application. For example, if the watermark were embedded at the time of recording, especially equipped cameras should be used. On the other hand, digital image forensics works in absence of any watermark or special hardware, and, therefore, has emerged as a new discipline that helps to solve authentication issues, and, consequently, regain trust in photographs [2]. Digital forensics relies on the fact that, although most forms of tampering may not leave any obvious visual clues in the image, they may disturb some specific properties. Over the last decade, plenty of forensic techniques have been developed to detect such perturbations. These techniques provide valuable solutions to the problem of image authentication and verification, and important evidence about the history of a content. They use, for example, statistical patterns at the pixel level [3], specific to a compression format (e.g., Joint Photographic Experts Group (JPEG) or Graphic Interchange Format (GIF)) [4], [5], or introduced by the camera lens or sensor [6], [7]; models of the interactions between physical objects, light and the camera [8], or the projective geometry principles of image formation [9].

One of the most extensively studied cases in digital image forensics is JPEG compression (which in turn is an extremely popular image format). By leveraging the characteristic footprints left in the distribution of Discrete Cosine Transform (DCT) coefficients during compression, those schemes detect, for example, a previous JPEG compression jointly with the used quantization table [4], [5], [10], or even disclose multiple instances of JPEG compression [11], [12], [13].

However, a significant drawback of the many forensic techniques proposed so far (including those dealing with JPEG) is that they are designed to detect a single operation (e.g., region duplication, single JPEG compression, resampling, or chromatic aberrations; see [3] for a general survey). Unfortunately, little attention has been paid up to now to the forensic analysis of manipulation chains, with the exception of double JPEG compression, where the same operator is applied twice. Indeed, the application of multiple heterogenous processing operators, often inevitable when creating a fake photograph, may seriously affect the performance of existing forensic algorithms, weakening or even erasing the specific footprints left by previous processing and exploited by forensic tools [14], [15]. Recalling the JPEG case, it is very likely that an image, after being stored in JPEG format, will be altered by a further post-processing, such as linear filtering, either to enhance its quality (e.g., edge sharpening) or to reduce the JPEG-compression blocking artifacts. Such post-processing will perturb the characteristic patterns present in the DCT distribution of JPEG images, and therefore reduce their reliability for forensic purposes.

In this work, we study the real-practical scenario in which a JPEG-compressed image is linearly filtered and saved in a lossless format (e.g., TIFF). This particular operator chain (i.e., the combination of JPEG compression and full-frame linear post-processing) is frequently used for quality enhancement,

V. Conotter is with the Department of Engineering and Computer Science, University of Trento, Trento, Italy e-mail: conotter@disi.unitn.it.

P. Comesaña and F. Pérez-González are with the Department of Signal Theory and Communications, University of Vigo, Spain. e-mail: pcomesan@gts.uvigo.es, fperez@gts.uvigo.es

Manuscript received 2014; revised xxxx.

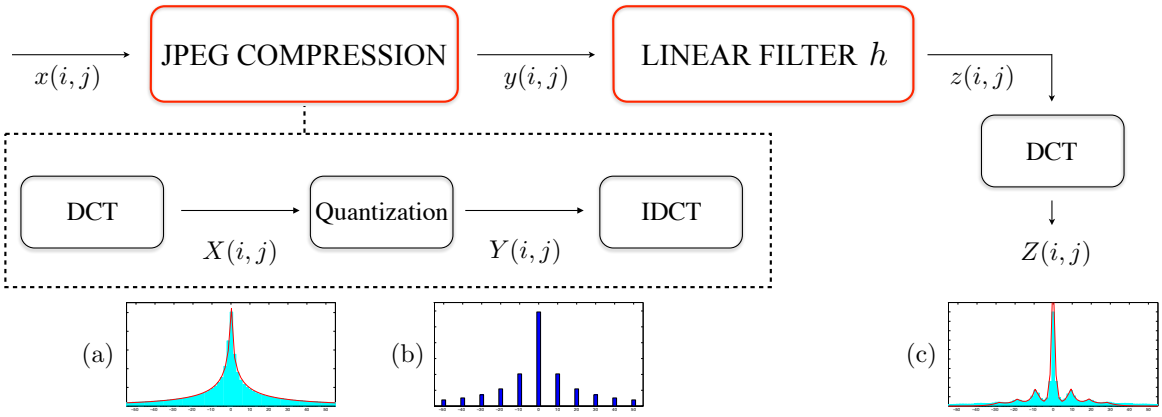

Fig. 1: Block diagram of the considered framework. Panel (a) shows the original DCT histogram for frequency $(0,1)$ for uncompressed images and its curve fitting. Panel (b) depicts the corresponding histogram after JPEG quantization with stepsize $q_{0,1} = 10$. Panel (c) shows the histogram when a 3×3 average filter is further applied to the image, and the model derived for such distribution (red line).

noise removal, illumination correction, deblurring, or blocking artifact reduction of JPEG images. Our analysis leverages on modeling the effect of the complete chain on the statistical distribution of DCT coefficients. Fig. 1 shows the block diagram of the analyzed framework, in which an uncompressed image (x) is JPEG compressed (yielding y), and subsequently undergoes a linear filtering (whose output is denoted by z). Fig. 1 also shows the histogram of the DCT coefficients of an image before and after each processing step: panel (a) shows the histogram before compression, panel (b) after compression, and panel (c) after linear filtering. It becomes clear that the quantization artifacts introduced by JPEG compression in the DCT distribution are clearly perturbed by filtering; however, new patterns appear in the DCT coefficients of the filtered image. In this work, we illustrate the dependence of such patterns on both the used JPEG compression quality factor (JPEG-QF) and the filter kernel, and exploit them to jointly disclose the filter kernel and the JPEG-QF that have been applied to an image.

In order to achieve this goal, we firstly derive a statistical model of the DCT coefficients of a JPEG image filtered with a given kernel; this model was introduced in our preliminary work [16]. As a by-product of this analysis, we show that the prevalent assumption in which the image "Alternating Current" (AC) DCT coefficients for different frequencies are independent, and i.i.d. for any fixed frequency [1], [17], does not hold, as neither the inter nor the intra-block redundancy of the quantized DCT coefficients can be neglected. The proposed statistical model will be exploited to derive a set of significant features from the DCT histogram of the input image; these features are fed to a linear classifier that effectively discriminates among different combinations of linear filtering and compression. The proposed forensic technique, preliminary presented in [18], is computationally efficient and effective, and able to jointly detect the filter kernel and the JPEG-QF that have been applied to an image. Indeed, the proposed forensic tool is shown to be valuable not only for disclosing the targeted chain of operators, but as valuable byproducts is also able to detect single operators (e.g., the applied compression factor only), and it is robust to double compression; we will empirically show that our detector outperforms state-of-the-art schemes.

The remaining of this paper is organized as follows: Sect. II presents our framework and recalls basic JPEG and linear-filtering concepts. Then, Sect. III presents the statistical model of the DCT coefficients of filtered JPEG images; from this model, a set of classification features is proposed. Those features are fed to a Support Vector Machine (SVM) classification scheme, whose performance is reported in Sect. IV. Finally, conclusions are drawn in Sect. V.

A. Notation

Throughout this paper, lower case letters (e.g., x) denote images of size $L_1 \times L_2$ in the spatial domain;[1] $x(i,j)$ represents the pixel of image x at position (i,j), with $i \in \{0, \cdots, L_1 - 1\}$ and $j \in \{0, \cdots, L_2 - 1\}$. Images in the 8×8-DCT domain are denoted by uppercase letters (e.g., X), so $X^{i_8,j_8}(i',j')$ stands for the (i',j')th DCT coefficient at the (i_8,j_8)th block, where $i',j' \in \{0, \cdots, 7\}$, $i_8 \in \{0, \cdots, (L_1/8) - 1\}$, and $j_8 \in \{0, \cdots, (L_2/8) - 1\}$; similarly, $x^{i_8,j_8}(i',j')$ denotes the (i',j')th pixel at the (i_8,j_8)th block. For the sake of notational simplicity, and due to the similarity with the pixel domain notation, we will also use $X(i,j)$, where $i = i' + 8 \cdot i_8$ and $j = j' + 8 \cdot j_8$, to denote $X^{i_8,j_8}(i',j')$. Consequently, prime variables will denote modulo 8 reduced variables, e.g., $i' = i \text{mod} 8$.

Following this notation, Fig. 1 depicts the main variables used in this work: an uncompressed image x is JPEG-compressed to generate y; X and Y are their 8×8-DCT versions, respectively. The application of a linear filter h in the spatial domain yields the filtered JPEG image z, whose 8×8-DCT coefficients are denoted by Z.

II. Framework description

As it was already mentioned, the aim of this work is to study the footprints left in the DCT coefficients by JPEG

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

compression followed by full-frame linear filtering. Consequently, we will firstly focus on JPEG compression, a standard which is based on an 8×8 non-overlapping block-by-block DCT transform. We will restrict our analysis to the luminance channel; further improvements may be possible by resorting to a color representation, but they are not pursued here. In particular, an image x of size $L_1 \times L_2$, is firstly partitioned into $L_1/8 \times L_2/8$ non-overlapping blocks of size 8×8. Each block $x^{i_8,j_8}(i,j)$ is then independently transformed from the spatial to the frequency domain, using the DCT, i.e.,

$$X^{i_8,j_8}(i',j') = \sum_{n_1=0}^{7}\sum_{n_2=0}^{7} \frac{c(i')}{2}\frac{c(j')}{2} x^{i_8,j_8}(n_1,n_2) \cdot \\ \cdot \cos\left(\frac{2n_1+1}{16}\pi i'\right)\cos\left(\frac{2n_2+1}{16}\pi j'\right), \quad (1)$$

where $c(k) = 1/\sqrt{2}$ if $k = 0$, and $c(k) = 1$ otherwise.

In its most widely used version, JPEG is a lossy scheme, implying that some information is lost during the compression process due to the quantization of the DCT coefficients, i.e.,

$$\hat{Y}^{i_8,j_8}(i',j') = \text{round}\left(\frac{X^{i_8,j_8}(i',j')}{q_{i',j'}}\right),$$

where $\hat{Y}$ are the indices of the quantized DCT coefficients, q is the so-called quantization table (an 8×8 matrix that contains the 64 integer-valued quantization steps), and $q_{i',j'}$ is its element at the (i',j')th location. The choice of the quantization table is user-dependent, and it is critical, as it must enable a good trade-off between visual quality and compression rate. The quantized DCT coefficients are finally entropy-encoded (by using Huffman coding) and stored in the JPEG file format.

Decompression is performed by applying the reverse processing. Specifically, the (i',j')th DCT coefficient of the (i_8,j_8)th block of the reconstructed image Y is

$$Y^{i_8,j_8}(i',j') = q_{i',j'} \cdot \text{round}\left(\frac{X^{i_8,j_8}(i',j')}{q_{i',j'}}\right),$$

and it is transformed from the frequency to the spatial domain by applying the Inverse DCT (IDCT) on each 8×8 block, yielding

$$y^{i_8,j_8}(i',j') = \sum_{k_1=0}^{7}\sum_{k_2=0}^{7} \frac{c(k_1)}{2}\frac{c(k_2)}{2} Y^{i_8,j_8}(k_1,k_2) \cdot \\ \cdot \cos\left(\frac{2i'+1}{16}\pi k_1\right)\cos\left(\frac{2j'+1}{16}\pi k_2\right). \quad (2)$$

Since JPEG compression forces the DCT coefficients to be integer multiples of the quantization steps, specific artifacts are introduced 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 uncompressed images [19], 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 used quantization step. Notice that the round-off and truncation errors introduced in the pixel domain were disregarded in that plot.

This DCT domain structure produces a characteristic blocking effect in the pixel domain. Due to the perceptual impact of such distortion, some post-processing is frequently applied to the compressed image. This processing can alter the characteristic artifacts left by JPEG compression, and consequently forensic JPEG detectors might become ineffective (cf. Sect. IV); indeed, it could be also the case that such processing is not aimed at enhancing image quality, but removing the JPEG artifacts, that is, with a counterforensic purpose. As it was previously said, this work deals with the case where full-frame linear filtering is applied to a JPEG compressed image. The convolution between an image y and a $[(2N+1)\times(2N+1)]$-sized linear filter kernel h is computed as

$$z(i,j) = \sum_{s_1=-N}^{N}\sum_{s_2=-N}^{N} h(s_1,s_2)y(i+s_1,j+s_2). \quad (3)$$

Panel (c) in Fig. 1 illustrates the histogram of the DCT coefficients of panel (b) after filtering the image in the pixel domain with a 3×3 average filter. The footprints in the quantized coefficient histogram are clearly perturbed. However, new patterns emerge; in the next section we show that these patterns depend on both the employed JPEG-QF and the filter kernel.

III. Mathematical Model

In order to statistically model the DCT coefficients of JPEG-compressed and linear filtered images, the following procedure is carried out:

1) Derive the deterministic relationship between Z and Y (see Fig. 1).
2) Propose a statistical model for Y; jointly with the previous step, this will give a statistical model for Z.
3) Propose a set of features, based on the previous mathematical model, that summarize the compression and filtering artifacts.

A. Deterministic relationship between Z and Y

The formulas establishing the deterministic relationship between Z and Y can be found in Appendix A. From (12), (13) and (14), it is clear that if a filter kernel of size smaller than or equal to 17×17 were used, then the coefficients of Y contributing to the calculation of $Z(i,j)$ would be those from the same block of $Z(i,j)$, jointly with those from the 8 immediate surrounding blocks; this results in a 24×24 coefficient neighborhood.

Finally, (15) gives the contribution of DCT coefficient $Y(i,j)$ to $Z(i,j)$, summarized next,

$$Z(i,j) = \gamma_{i',j'}Y(i,j) + R(i,j), \quad (4)$$

where $\gamma_{i',j'}, R(i,j) \in \mathbb{R}$ are a frequency dependent scaling factor and a noise term, respectively, accounting for the contribution of all the neighboring coefficients of $Y(i,j)$. Note that, differently from the effect of the circular convolution on DFT coefficients, the resulting relationship between $Y(i,j)$ and $Z(i,j)$ is not purely multiplicative.

B. Probability distribution

Given the derived deterministic expression of $Z(i,j)$ in (12), we can exploit the knowledge about the distribution of the quantized coefficients $Y(i,j)$ to study the distribution of

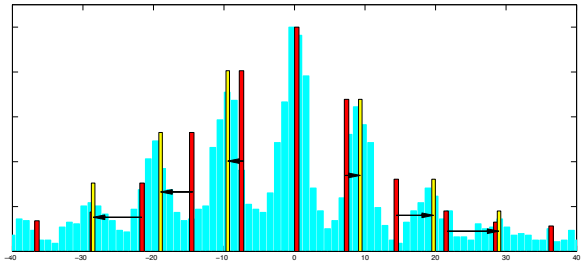

Fig. 2: Histogram of DCT coefficients at frequency $(0,1)$, for $q_{0,1} = 10$ and 3×3 average filter. The red bars are located at the integer multiples of $\gamma_{0,1} q_{0,1}$, while the yellow ones correspond to $\gamma'_{0,1} q_{0,1}$.

the DCT coefficients of the filtered image z. Usually, the probability distribution of 8×8 block-DCT coefficients in natural images for a particular coefficient (i,j) is modeled as a zero-mean Generalized Gaussian Distribution (GGD) [4], i.e.,

$$f_{X(i,j)}(t) = \frac{\beta_{i',j'}}{2\alpha_{i',j'}\Gamma(1/\beta_{i',j'})} \exp(-|t|/\alpha_{i',j'})^{\beta_{i',j'}}, \qquad (5)$$

where we have explicitly considered that all the coefficients corresponding to a particular frequency (i.e., those (i,j) such that $(i \mod 8, j \mod 8) = (i',j')$) are identically distributed, Γ denotes the gamma function, and $\alpha_{i',j'}$ and $\beta_{i',j'}$ are the scale and the shape parameters, respectively.

Due to quantization, the probability mass function (pmf) of $Y(i,j)$, given that the quantization step $q_{i',j'}$ is used, is [4]

$$f_{Y(i,j)|q_{i',j'}}(\tau) = \sum_k \delta(\tau - k q_{i',j'}) L_{i',j'}(k q_{i',j'}), \qquad (6)$$

where

$$L_{i',j'}(k q_{i',j'}) \triangleq \int_{(k-\frac{1}{2})q_{i',j'}}^{(k+\frac{1}{2})q_{i',j'}} f_{X(i,j)}(\tau) d\tau, \qquad (7)$$

with $k \in \mathbb{Z}$. Therefore, it becomes evident that the pmf of each frequency coefficient of a JPEG image presents specific artifacts, whose structure is related to the quantization step. In particular, the DCT coefficients corresponding to the (i',j')th frequency will be located at multiples of the applied quantization step $q_{i',j'}$, as illustrated in Fig. 1(b).

Returning to (4), it is clear that the distribution of $Z(i,j)$ will depend on the distribution of $R(i,j)$. The prevalent statistical models for DCT coefficients [1], [17], [20], regard different frequency components as mutually independent, and coefficients at a given frequency as i.i.d; consequently, one would expect $R(i,j)$ to follow a symmetric zero-mean distribution independent of $Y(i,j)$. Nevertheless, if one plots the histogram for a particular frequency of the DCT of the compressed and filtered image, it is evident that its peaks are not located at $\gamma_{i',j'} Y(i,j)$ (i.e., $\gamma_{i',j'} k q_{i',j'}$); see Fig. 2 for an example. We can conclude that common models for DCT coefficients are not appropriate enough for our particular problem. In other words, for our analysis we cannot assume the different DCT coefficients to be mutually independent. In turn, we will model the noise component $R(i,j)$ as a GGD random variable, whose parameters (mean, variance, and shaping) depend on the value of $Y(i,j)$.

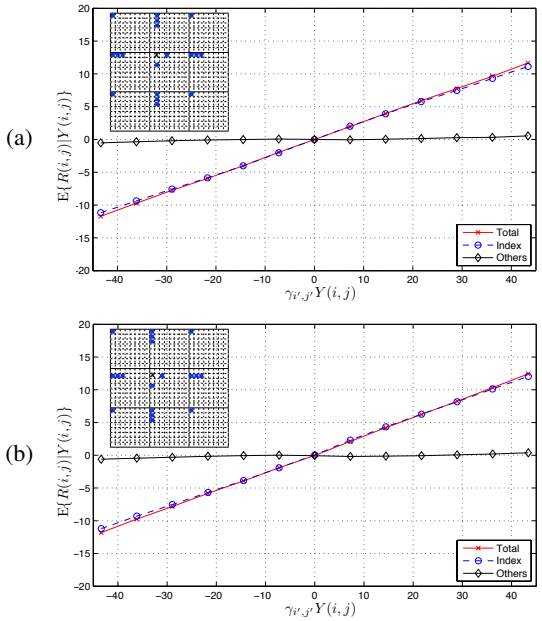

Fig. 3: Inter- and intra-block spatial redundancy affecting $X(i,j)$, in panel (a) $(i,j) = (0,1)$ and in panel (b) $(i,j) = (1,0)$) over the entire image dataset. The total mean of the noise component $\mathrm{E}\{R(i,j)|Y(i,j)\}$ (red curve) is plotted as a function of $\gamma_{i',j'}Y(i,j)$. The blue curve represents the contribution of a set of coefficients (depicted in blue in the grid) and the black curve corresponds to the contribution of the complementary subset coefficients.

To elaborate, in Fig. 3 we plot in red $\mathrm{E}\{R(i,j)|Y(i,j)\}$ as a function of $\gamma_{i',j'}Y(i,j)$ when a 3×3 average filter is applied, showing an approximately linear relationship; a similar behavior has been verified for different filter kernels and frequencies. Subsequently, we have identified the subset of DCT coefficients from the aforementioned 24×24 neighborhood with the largest contribution to $\mathrm{E}\{R(i,j)|Y(i,j)\}$; for each DCT frequency this subset of coefficients was determined by comparing their contribution to $\mathrm{E}\{R(i,j)|Y(i,j)\}$, in absolute value, with an empirically determined threshold. The grid in the upper left part of Figs. 3 (a)-(b) shows the resulting coefficient subsets for the AC coefficients $(0,1)$ and $(1,0)$, where the blue dots correspond to the relevant coefficients. The blue curve in Figs. 3 (a)-(b) represents the contribution to the mean of the coefficients in that subset, while the black curve represents the contribution of the complementary subset. From these empirical observations it is reasonable to modify the model in (4) so as to consider the linear relationship between $\mathrm{E}\{R(i,j)|Y(i,j)\}$ and $Y(i,j)$. The model then becomes

$$Z(i,j) = \gamma'_{i',j'} Y(i,j) + R'(i,j), \qquad (8)$$

where $\gamma'_{i',j'}$ is the slope of the linear regression of the points $(k q_{i',j'}, \tilde{\mathrm{E}}\{Z(i,j)|Y(i,j) = k q_{i',j'}\})$, $k \in \mathbb{Z}$, that is, we exploit the linear dependence between $k q_{i',j'}$ and the sample mean (denoted by $\tilde{\mathrm{E}}\{\cdot\}$) of $Z(i,j)$ for those (i',j')-frequency coefficients such that the corresponding DCT coefficient of the

JPEG compressed image is equal to $kq_{i',j'}$. Note that

$$\begin{aligned} &\mathrm{E}\{Z(i,j)|Y(i,j)=kq_{i',j'}\} \\ &= \gamma_{i',j'}kq_{i',j'} + \mathrm{E}\{R(i,j)|Y(i,j)=kq_{i',j'}\} \\ = \quad &\gamma'_{i',j'}kq_{i',j'} + \mathrm{E}\{R'(i,j)|Y(i,j)=kq_{i',j'}\} = \gamma'_{i',j'}kq_{i',j'}, \end{aligned}$$

where the first equality follows from (4), the second equality is a consequence of (8), and for the third we have considered that $\mathrm{E}\{R'(i,j)|Y(i,j)=kq_{i',j'}\}=0$, as $R'(i,j)$ is the linear estimation residual error, and consequently it is uncorrelated with $Y(i,j)$. Consequently, Fig. 3 illustrates that $\gamma'_{i',j'} \neq \gamma_{i',j'}$, as the slope of $\mathrm{E}\{R'(i,j)|Y(i,j)=kq_{i',j'}\}$ with respect to $Y(i,j)$ is non-null. Additionally, $\gamma'_{i',j'}$ may be larger than 1 and/or negative, meaning that the considered frequency is amplified and/or phase-inverted, respectively.

Similarly, Fig. 2 shows that the peaks of the histogram of DCT coefficients from JPEG-compressed and filtered images corresponding to a particular frequency are indeed centered at the integer multiples of $\gamma'_{i',j'}q_{i',j'}$. This reflects the fact that both the inter- and intra- block redundancy of the quantized DCT coefficients must be taken into account when constructing an accurate statistical model for the distribution of the DCT coefficients of filtered JPEG images.

We remark that γ' strongly depends on the applied filter kernel and the correlation among DCT coefficients, which in turn is modified by the applied compression factor. In order to illustrate this, Fig. 4 graphically reports the obtained $\gamma'_{i,j}$ matrices for the cases where QF$=40$ and filter no. 1 are used (denoted by $(\gamma')^a$); QF$=90$ and filter no. 1 (denoted by $(\gamma')^b$), and QF$=40$ and filter no. 11 (denoted by $(\gamma')^c$).[2] These examples clearly show that in general $\gamma' \neq 1$ and that γ' depends on the applied JPEG-QF, and, more significantly, on the considered filter.

In order to quantify the goodness of the proposed model, we have computed the *sample Pearson correlation coefficient*, defined for input points (X_i, Y_i), $1 \leq i \leq n$, as

$$\rho \doteq \frac{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)\left(Y_i-\bar{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2}\sqrt{\sum_{i=1}^{n}\left(Y_i-\bar{Y}\right)^2}},$$

where $\bar{X}$ and $\bar{Y}$ are the sample means. Note that ρ^2 is the proportion of the variance in $\mathbf{Y}$ that is explained by a linear function of $\mathbf{X}$. Consequently, the closer $|\rho|$ is to 1, the better the proposed linear model will be. Noticing that for those frequencies with small values of $|\gamma'|$ the variance of $R'(i,j)$ will be large in comparison with $(\gamma'_{i',j'}Y(i,j))^2$, and consequently the proportion of the variance of $Z(i,j)$ explained by the latter is expected to be small, we computed the average values of $|\rho(i',j')|$ for those frequencies with $|\gamma'| \geq 0.05$ in the three considered examples, giving $\bar{\rho}^a = 0.9936$, $\bar{\rho}^b = 0.9913$, and $\bar{\rho}^c = 0.9974$; these results confirm again the excellent fit of the proposed model.

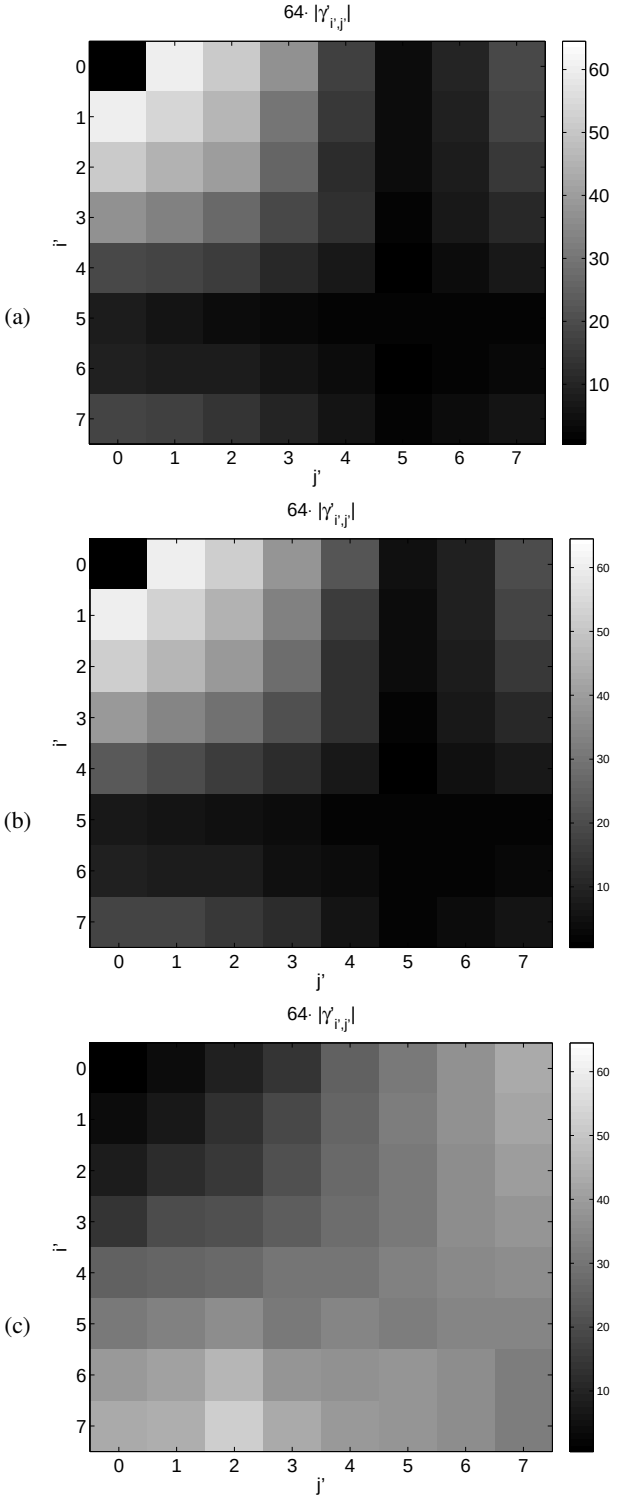

Fig. 4: Graphical representation of: (a) $(\gamma')^a$ (QF$=40$ and filter no. 1); (b) $(\gamma')^b$ (QF$=90$ and filter no. 1); (c) $(\gamma')^c$ (QF$=40$ and filter no. 11);

C. Statistical features for classification

The model derived in the previous section is now exploited to propose a set of features that characterize the JPEG

[2]In order to replicate the real scenario, the filtered images where converted to 8-bit unsigned integer values, i.e. they are in $\{0,1,\ldots,255\}$; therefore, the reported results consider both the clipping and rounding effects of such conversion.

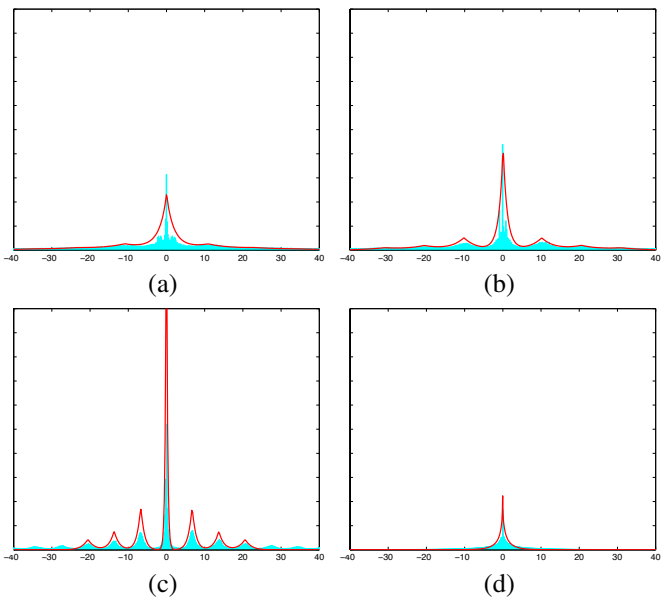

Fig. 5: DCT coefficient histograms at frequency $(0,1)$ collected from 100 UCID images: (a) compressed with QF $= 40$ and linearly filtered with a 5×5 average filter; (b) compressed with QF $= 50$ and linearly filtered with a 3×3 average filter; (c) compressed with QF $= 70$ and linearly filtered with a 3×3 Gaussian filter with $\sigma_F^2 = 0.5$ (d) compressed with QF $= 80$ and linearly filtered with a 3×3 high-pass filter. The corresponding derived theoretical models are overlaid onto the histograms (red line).

compression and full-frame linear filtering. In particular, we consider the distribution of $Z(i,j)$ which, as we have just shown, has characteristic peaks located at integer multiples of a scaled version of the corresponding quantization step. Fig. 5 shows four examples of histograms of DCT coefficients for the frequency $(0,1)$ collected from 100 UCID images that have been processed with different combinations of JPEG compression and filtering. In particular, panel (a) depicts the DCT coefficient histogram of images compressed with a JPEG-QF $= 40$ and processed with a 5×5 average filter; panel (b) shows the histogram for images compressed with JPEG-QF $= 50$ and processed with a 3×3 average filter; panel (c) corresponds to images compressed with a JPEG-QF $= 70$ and linearly filtered with a 3×3 Gaussian filter with $\sigma_F^2 = 0.5$; finally, panel (d) contains the histogram for images compressed with JPEG-QF $= 80$ and processed with a 3×3 high-pass filter. On each plot, the probability distribution derived in the previous section is overlaid to the corresponding histogram, showing a good accuracy.

Fig. 5 graphically illustrates a fact that can be inferred from (8): both the location and the shape of the peaks present in the DCT distribution of filtered JPEG images strongly depend on the applied compression factor and the kernel used to filter the image; such dependence will be exploited by our forensic detector. Specifically, it is obvious the dependence of the peak location with $\gamma'_{i',j'}$ in (8); additionally, for a given $\gamma'_{i',j'}$, the larger $\mathrm{Var}\{R'(i,j)\}$, the smoother the resulting histogram. Consequently, our set of features will try to summarize the information on the peak location and shape, as well as the variance of the DCT coefficients. In order to do

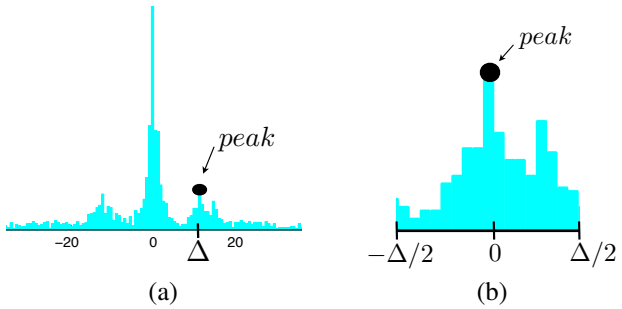

Fig. 6: (a) Histogram of DCT coefficients at frequency $(0,1)$ of an image that has been compressed with $q_{0,1} = 10$ and linearly filtered with a 3×3 average filter; the black dot indicates the first peak in the histogram, located at $\Delta_{0,1}$. (b) Modulo-$\Delta_{0,1}$ reduced version of the histogram in (a).

so, we will find it useful to denote by $\mathbf{Y}(i',j')$ the vector containing the coefficients for the (i',j')th frequency and whose kth element is $Y^{\lfloor 8k/L_2\rfloor,k-L_2\lfloor 8k/L_2\rfloor}(i',j')$; $H(\mathbf{w},\delta)$ denotes the histogram of an arbitrary vector $\mathbf{w}$ with bin width δ; $H(\mathbf{w},\delta,k)$ represents its value for the kth bin, and, finally, $\Delta_{i',j'} > 0$ denotes the location of the characteristic peaks in $H(|\mathbf{Y}(i',j')|,\delta_{i',j'})$. Note that the peak location depends on the (i',j')th DCT coefficient (besides the JPEG-QF and the kernel filter). For the sake of notational simplicity, whenever it is clear from the context, we will avoid to make explicit reference to the frequency index of both the histogram and peak location (i.e., indices (i',j') will be dropped).

1) Peak location feature: A variety of peak detection algorithms have been proposed in the literature, many of which may be used to construct this feature. Here we propose to look for those points with neighbors, both on the left and the right, that are smaller than the point of interest by, at least, a given threshold T (in particular, we set $T = 0.05$). Those points form a candidate set. Noticing the typical monotonically decreasing nature of the distribution of the DCT coefficients away from the origin, we select that point from the candidate set with the largest histogram value (excluding the origin). Formally,

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

where $|\mathbf{Y}|$ denotes the component-wise absolute value operator. This algorithm has been shown to be computationally efficient and reliable. For the sake of illustration, Fig. 6(a) shows the detected peak (black dot) at location Δ, in the histogram of the $(0,1)$ DCT coefficients of an image that has been quantized with $q_{0,1} = 10$ and post-processed with a 3×3 average filter.

2) Peak shape feature: Given the detected peak location Δ, we now aim at computing a measure of the peakiness of the histogram of $\mathbf{Y}$ around the integer multiples of Δ. To this end, we formalize the modulo-Δ reduced version of $\mathbf{Y}$; specifically, we define

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

which returns values in $(-\Delta/2,\Delta/2]^{L_1L_2/64}$; Fig. 6(b) illustrates the modulo-Δ reduced version of the histogram in Fig.

feature	description
Δ	peak location
β	peak shape
$\lvert\mathcal{Z}\rvert$	zero-quantized DCT coefficients
$\sigma^2_{\tilde{Y}}$	DCT coefficient variance

TABLE I: Classification features, extracted from the histogram of the DCT coefficients for each DCT frequency (i', j').

6(a). Then, we compute the empirical variance of $\tilde{\mathbf{Y}}$ as

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

Recall that $\tilde{Y}^{k_1,k_2}$ actually represents $\tilde{Y}^{k_1,k_2}(i',j') = Y^{k_1,k_2}(i',j') \text{mod} \Delta_{i',j'}$. Furthermore, we have assumed $\tilde{\mathbf{Y}}$ to be zero-mean, according to the symmetry of X with respect to the origin.

Given $\sigma^2_{\tilde{Y}}$, the peakiness of the histogram of $\mathbf{Y}$ is quantified as

$$\beta = \frac{\sigma^2_{\tilde{Y}}}{\Delta^2}. \tag{9}$$

Clearly, if the values in $\mathbf{Y}$ are clustered around integer multiples of Δ, then the values in $\tilde{\mathbf{Y}}$, with range $(-\Delta/2, \Delta/2]$, will be clustered around the origin; therefore, the empirical variance of the latter will be small in comparison with Δ^2, and, consequently, a low value of β will be a clue of a peaky histogram of $\mathbf{Y}$. On the other hand, if the values in $\tilde{\mathbf{Y}}$ are uniformly distributed in $(-\Delta/2, \Delta/2]$ (i.e., if there is no peak at all), then β will be close to $1/12$, hinting at a smooth histogram of $\mathbf{Y}$.

3) Zero-quantized DCT coefficients feature: Typically, in JPEG compression, large quantization steps are used for high-frequency DCT coefficients, since they have little visual significance. Moreover, it is well known that the larger the number of zero-quantized DCT coefficients, the larger the compression rate [21]. As a consequence, in some images all the high-frequency DCT coefficient values will be quantized to zero, so no peaks would be detected in this case. It could also be the case that the image just does not contain high frequencies, because it was, for example, low-pass filtered. In order to quantify these properties, 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 the (i',j')th DCT frequency. We compute the cardinality $|\mathcal{Z}_{i',j'}|$ of such a set and take it as another significative feature to characterize the behavior of DCT coefficients at high frequencies.

4) DCT coefficient variance feature: Similarly, we compute the empirical variance of $\mathbf{Y}$ to take into account the general behavior of each (i',j') frequency, and in particular to deal with those frequencies where most of DCT coefficients are zero-quantized. Specifically,

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

This value will depend on the nature of the image itself, the applied compression, and the post-processing filter.

Group 1	**Group 4**
1. LP Average $[3 \times 3]$	9. LP Laplacian, $\alpha = 0.2$
2. LP Average $[5 \times 5]$	
4. LP Gaussian $[3 \times 3]$, $\sigma^2_F = 1$	10. LP Laplacian, $\alpha = 0.7$
6. LP Gaussian $[5 \times 5]$, $\sigma^2_F = 1$	
Group 2	**Group 5**
3. LP Gaussian $[3 \times 3]$, $\sigma^2_F = 0.5$	11. HP Average $[3 \times 3]$
5. LP Gaussian $[5 \times 5]$, $\sigma^2_F = 0.5$	12. HP Average $[5 \times 5]$
Group 3	**Group 6**
7. HP Laplacian, $\alpha = 0.2$	13. Identity filter
8. HP Laplacian, $\alpha = 0.7$	

Tab. II: Filters in the filter dictionary, grouped according to the similarity of their frequency responses.

These features (which are summarized in Table I) were proposed for the first time in [18]. They are extracted for each AC 8×8-DCT frequency and are expected to provide a complete and proper summary of the properties of the considered histogram.

IV. Experimental analysis

In a previous work [16], we have shown that, when the JPEG-QF is assumed to be known, the model in Sect. III can be exploited when finding out the applied filter kernel. To this aim, we have employed a distinguishability metric (i.e., the χ^2 histogram distance) to quantify the difference between the statistical model (as parameterized by the filter kernel) and the corresponding histogram of the image under test. Through experimental analysis, the derived scheme was shown to be effective in characterizing the distribution of the DCT coefficients of a filtered JPEG image.

Here, we relax such strong assumption regarding the knowledge of the JPEG-QF. Specifically, the forensic detector used in the current work (and preliminarily introduced in [18]) is able to disclose both the applied JPEG-QF and the applied linear filter kernel. The basic idea behind this forensic scheme is to feed a linear classifier, precisely a linear kernel SVM [22], with the features described in the previous section corresponding to all the AC DCT coefficients; this produces an SVM input vector of dimensionality $4 \times 63 = 252$. In this section, we show that this technique results to be a simple yet very effective forensic tool which is able to jointly detect the filter kernel and the JPEG-QF that have been applied to an image, so as to uncover the processing history of the content.

In order to verify the effectiveness and versatility of the proposed forensic technique, an extensive experimental analysis is conducted here. The performance of the algorithm is evaluated in terms of the classification accuracy achieved by the proposed classifier.

A. Dataset definition

Unless it were otherwise specified, we consider a subset of 600 uncompressed images, randomly selected from the UCID dataset [19]. Each image was compressed using different JPEG-QFs $= 40, 50, 60, 70, 80, 90$, subsequently convolved

with a filter kernel chosen from a dictionary of linear filters and finally saved in a lossless format (e.g., TIFF).

Note that we only consider a small subset of possible JPEG-QFs as representatives of the full range of QFs. Interestingly, our experiments, not reported here, show that images compressed with intermediate QFs are classified with a very large probability as compressed with either of the two closest representatives. For instance, when the true QF is 46, images are classified with a very large probability as compressed with either QF= 40 (17.1%) or QF= 50 (82.2%). Consequently, our classifier is actually performing a coarse estimation of the JPEG-QF, which is sufficient for most practical purposes.

The filter set contains both low-pass (LP) and high-pass (HP) filters (e.g., average, Gaussian, Laplacian), each of them parameterized by different settings (e.g., window size, variance σ_F^2 for the Gaussian filters, or shape parameter α for the Laplacian ones). Table II enumerates the filters included in our dictionary; for example, filter no. 1 is an average filter with window size 3×3, while filter no. 3 is a low-pass Gaussian filter with $\sigma_F^2 = 0.5$ and the same size, and filter no. 7 is a high-pass Laplacian filter with $\alpha = 0.2$. Notice that filter no. 13 is the identity, so as to include in the analysis also the case in which no filter is used (that is, only JPEG compression is applied).

Obviously, the filter dictionary could contain a larger number of filters, but we consider that the chosen ones are sufficiently representative of the possible postprocessing a compressed image can go through. Furthermore, as it was already reported in [16], some of the filters in the dictionary may share a similar frequency response, thus being it difficult to distinguish between them; this issue is not specific of the present technique, but an inherent limitation. Therefore, the filters are grouped according to the similarity of their frequency responses; Table II shows those groups. In any case, from the point of view of the forensic application, in most instances it will suffice to decide which group the filter belongs to.

Since we consider 6 JPEG-QFs and 13 linear filters, the constructed image dataset contains $600 \times 13 \times 6 = 46800$ images. Then, this dataset is split in halves, in order to generate the SVM training and test sets. It is worth mentioning that all the 78 versions of a given original image are always placed in the same set, so as to avoid biases in the training or the test stages. Next, the set of forensic features in Table I is extracted for each image, and the SVM is trained with those images in the training test.

The performance of the proposed algorithm is quantified in terms of the classification accuracy averaged over the test set.

B. Joint JPEG-QF and filter classification

As a first experiment, we considered the simplest case, where 78 classes must be discriminated by the SVM classifier, corresponding to each possible pair of the 6 JPEG-QFs and the 13 filters in the dictionary. The overall accuracy is 74.5% (Application Scenario 1 in Table III); this is a rather low accuracy, which is likely due to the similarity in the frequency response of the filters that misleads the classifier. Panel (a) in Fig. 7 shows the classification accuracy achieved for each possible pair of JPEG-QF and filter.

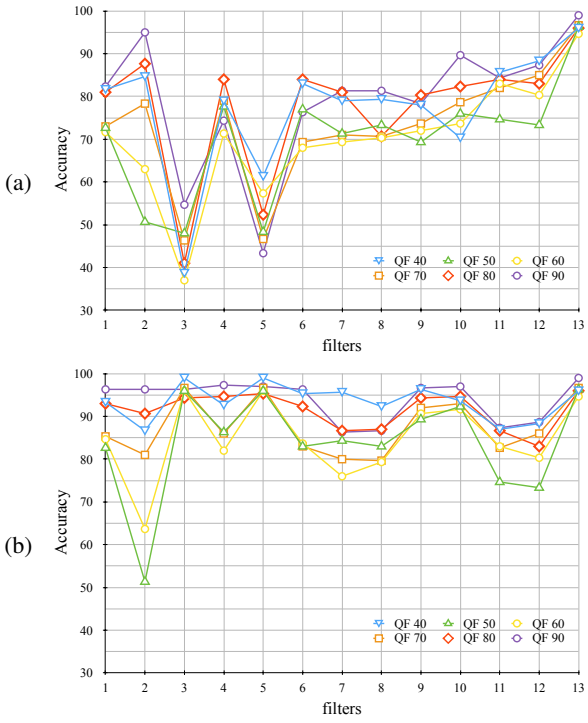

Fig. 7: Panel (a) shows the classification accuracy for each combination of selected filter and JPEG compression, while panel (b) reports results for the classification accuracy when 6 groups are considered for representing the filters.

To verify the influence of the similarity between filters on the classification, we considered the 6 filter groups described above, so the number of classes is reduced to 36 (6 groups of filters and 6 different compression factors). The average accuracy obtained is now 89.2%, which indicates that the proposed method is quite effective in jointly disclosing the applied JPEG-QF and linear filter group (Application Scenario 2 in Table III). Panel (b) in Fig. 7 shows the classification accuracy achieved for each possible pair of JPEG-QF and the 13 filters in the dictionary, when the 6 filter groups are considered for classification (i.e., 36 different classes are taken into account, but the reported results are disaggregated depending on which of the 13 filters was actually applied). It becomes clear that in this case the accuracy averaged over all the combinations of JPEG-QF and filter is increased with respect to the classification without grouping the filters, shown in panel (a). For instance, the achieved accuracy for classifying images compressed with JPEG-QF= 90 and processed with filter no. 3 is 54.7% without considering the filter groups; however, it significantly improves to 96.3% when filter group classification is taken into account (see Table II), thus exploiting the filter similarity.

In order to verify the effectiveness of the proposed method in different practical scenarios, we performed an additional test by including in the dataset some images coming from different sources. Indeed, in practice prior knowledge of the

App. Scenario	Detect	group filters	# filters	QF range	# classes SVM	Accuracy
1.	F + QF	no	13	$\{40, ..., 90\}$	78	74.5%
2.	F + QF	yes	13	$\{40, ..., 90\}$	36	89.2%
3.	only QF	no	13	$\{40, ..., 90\}$	6	89.4%
4.	QF (no filtering)	no	1 (= Identity filter)	$\{30, ..., 100\}$	8	99.1%
5.	F + QF (double compressed with $QF_2 = 90$)	yes	13	$\{40, ..., 90\}$	36	88.6%
6.	single/double compression	no	1(= Identity filter)	$\{40, ..., 90\}$	2	99.6%
7.	single/double compression	yes	13	$\{40, ..., 90\}$	2	99.9%

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

image source will not be available. Therefore, a more realistic strategy for forensic analysis would be to train the detector by using a pool of different source images. In particular, we included an additional set of 45 raw images coming from two mobile phones. We partitioned such set into two subsets: 24 images for training, and 21 for testing. Each image was compressed using JPEG-QFs $= 40, 50, 60, 70, 80, 90$, and subsequently convolved with a filter kernel chosen from the dictionary, for a total of $45 \times 13 \times 6 = 3510$ new images, which were integrated in the existing dataset, yielding a total of 25272 images for training and 25038 for testing. This dataset was used for training a SVM for Application Scenario 1, where the JPEG-QF and the filter kernel are jointly disclosed. Experiments confirm the good performance of the proposed algorithm in this practical scenario, where images from different sources are taken into account. Specifically, we obtained an accuracy equal to 73.25% when only UCID images were considered for testing, and 84.67% when the test only used images from mobile phones; the overall accuracy was 74%, which is just slightly smaller than the result reported for the original dataset (i.e., 74.5%).

We must remark that since there are no previous works in the literature addressing the problem of joint JPEG-QF and filter classification, we cannot compare the performance of our schemes with other approaches.

C. JPEG-QF classification

The proposed forensic scheme can also be used to estimate only the applied JPEG-QF, i.e., in this application scenario we will not take into account the filter classification. In this case, the number of classes to be discriminated by the classifier is 6, corresponding to the considered JPEG-QFs (those mentioned in the previous section, i.e., JPEG-QF $\in \{40, 50, 60, 70, 80, 90\}$). A newly trained SVM with only 6 classes yielding an average accuracy of 89.4% (Application Scenario 3 in Table III) was used here. It is worth noticing that the obtained accuracy is very close to the one achieved in Application Scenario 2. Indeed, the marginalization of the classification with respect to the filter group provides only a minor accuracy increase, as the SVM trained for Application Scenario 2 seems to be already exploiting adequately the footprints left by both the JPEG-QF and the filter kernel.

Although no forensic techniques exist in the current literature dealing with the detection of JPEG filtered images, there are a number of algorithms that try to estimate the JPEG-QF applied to an image, in absence of any post-processing [4]. Therefore, in order to provide a fair comparison, we will now focus on the case where the input images were only compressed, i.e., where the identity filter (i.e., no filter) was used. To this end, the 600 original images in the dataset are compressed with different JPEG-QFs $= 30, 40, 50, 60, 70, 80, 90, 100$, and the features in Table I are extracted; therefore, an 8-class SVM is considered. The average accuracy obtained by our proposal is 99.1% (Application Scenario 4 in Table III); this result well illustrates the versatility of our scheme.

Interestingly, the obtained accuracy is comparable to the performance achieved by the state-of-the-art forensic technique in [4], although our proposal is much simpler and more computationally efficient. In [4] the authors introduce a statistical model to characterize the JPEG-induced near-periodic structures in the DCT distribution of compressed images; this model is exploited to retrieve the applied quantization table (information that could be used, for example, for recompression purposes). Table IV (case *a*) reports the comparison of our algorithm with [4] in terms of classification accuracy on detecting the JPEG-QF when no filter is applied; we consider that there is a classification error in [4] whenever the quantization step for any DCT frequency is not correctly estimated. According to this criterion, in Application Scenario 4 the method proposed in [4] yields an average accuracy of 83%; this fact highlights how our proposal improves the state-of-the-art. It is worth mentioning that the lower accuracy achieved by [4] is due to errors in some particular positions of the extracted quantization table; indeed, as stated in [4], those errors have a negligible impact on the targeted application, which is recompression. Furthermore, it is fair to mention that the estimate in [4] works componentwise, while the current approach exploits the dependences among the quantization steps of different DCT-coefficients, as it deals with the JPEG-QF estimate; consequently, the current approach is more robust against errors in particular positions of the estimated quantization table (at the price of assuming that the quantization table belongs to a QF-indexed set).

Moreover, we checked how the performance of the JPEG-QF estimation schemes in the literature degrades when the JPEG compressed image is filtered. In particular, the technique proposed in [4] dramatically decreases its classification accuracy to only 9.3% when the JPEG images are further processed with a linear filter (case *b* in Table IV); recall that

the accuracy of our method for Application Scenario 3 is 89.4%. Furthermore, note that in the tests run with the method introduced in [4] we excluded the high-pass filters (numbers 7,8,11,12 in Table II), since the implementation provided by the authors reported some errors.[3]

D. JPEG + filtering + JPEG: joint JPEG-QF and filter classification under double compression

As a further experiment, we analyzed the effectiveness of the proposed approach when the filtered JPEG images goes through a second JPEG compression, with quality factor JPEG-QF$_2$, that is, we tested the robustness of our technique under double compression (Application Scenario 5). Such a processing chain has a great practical interest; in fact, a plethora of forensic methods already exist for detecting double JPEG compression [11], [12], [13]. In such scenario, it makes perfect sense to use a linear filter as a counterforensic measure in order to conceal the footprints left by the first (and generally, lower quality) JPEG compression. To the best of our knowledge, this scenario in which a linear filter is placed between the two compressions has not been considered previously in the literature.

In order to test our algorithm in this application scenario, we took the previously created dataset (600 images compressed with JPEG-QFs $= 40, 50, 60, 70, 80, 90$, and 13 filters) and further compressed them by using a JPEG-QF$_2 = 90$. Again, the features in Table I were extracted and fed to the SVM. The average accuracy of the system is 88.6% (see Table III, Application Scenario 5), showing the effectiveness of the proposed forensic technique also under recompression.

As a representative of the state-of-the-art schemes for this framework, we selected the method described in [15], which deals with the detection of the primary JPEG-QF in case of double-compression by relying on the integer periodicity of the blockwise DCT coefficients. For this comparison we considered 300 images (i.e., the number of original images in our test set) that were compressed with JPEG-QFs $= 40, 50, 60, 70, 80$, and then recompressed with JPEG-QF$_2 = 90$ (case *c* in Table IV). In order not to penalize the results provided by the method proposed in [15], no filtering was applied; in such framework, the latter method yields an outstanding 100% average accuracy. On the other hand, for our algorithm, we used the SVM trained for Application Scenario 5, where only the identity filter and JPEG-QFs in $\{40, 50, 60, 70, 80\}$ were considered, thus producing 5 classes. The average accuracy for our scheme is 97%, which is close to that given by the method in [15] (see Table IV), which, unlike ours, is specifically tailored to the problem at hand. In fact, when some linear filtering is applied between the two compressions, the performance of [15] dramatically decreases to 14.2%, while our algorithm still reports an accuracy of 90.2% (case *d* in Table IV).

E. Single vs Double JPEG compression classification

As a last experiment, we adapted our method to detect whether an input image was singly or doubly JPEG compressed. These results are particularly helpful in several practical applications where a customer wishes to learn if a compressed image that he/she is interested in buying was not previously compressed with a lower quality. In this case, the training set was generated from 300 original images; the single compression class was produced by compressing each of those images with JPEG-QF $= 90$, while for the double compression class the same set of images were compressed with JPEG-QF in $\{40, 50, 60, 70, 80, 90\}$, and subsequently a second quantization with JPEG-QF$_2 = 90$ was applied. A similar procedure, but using 300 different original images, was used for creating the test set. No filtering was applied at this stage. Please note that in those experiments the sample sizes of the two classes are extremely disparate. This is a common issue in classification problems, where the number of available training data in each class might be unbalanced. According to [22], this problem can be overcome by using different penalty parameters for each class in the SVM setup. Nevertheless, experimental results obtained by using weights inversely proportional to the number of elements of each class in the training set show no significant performance differences (specifically, around $\pm 0.001\%$ accuracy difference).

Case Study	Detect	Filters	Accuracy		
			ours	[4]	[15]
a	QF	no	99.1%	83%	-
b	QF	yes	89.4%	9.3%	-
c	QF (with QF$_2 = 90$)	no	97.0%	-	100%
d	QF (with QF$_2 = 90$)	yes	90.2%	-	14.2%

Tab. IV: Comparison with the state-of-the-art in terms of classification accuracy in different application scenarios.

The average accuracy of our classifier was as large as 99.6% (Application Scenario 6 in Table III). Next, we considered the same problem, i.e., single vs double compression, but where a filter is now placed between both quantizers (in case there is double compression). As we mentioned above, the filter removes some of the footprints, thus making the classification more difficult. A new SVM that considers all the 13 filters in Table II and 300 original images is trained in this setup. The obtained average accuracy (averaged over a dataset generated from the remaining 300 images and the 13 filters) is now 99.9% (Application Scenario 7 in Table III). Furthermore, since in this case we are dealing with a binary classification problem, it is also useful to provide the Area Under Curve (AUC), which in the application scenarios described above is 0.982 and 0.998, respectively.

The latter SVM (trained by considering the 13 filters in the dictionary) is also used for testing the performance of our scheme for each of the 13 filters. In order to do so, 13 test datasets were created: for the double compression class 300 images were compressed with JPEG-QFs $= 40, 50, 60, 70, 80, 90$, then filtered with one of the 13 filters in the filter dictionary, and recompressed with JPEG-QF$_2 = 90$, while for the single compression class they were compressed only once with JPEG-QF $= 90$. The orange bars in Fig. 8 report the AUC for each tested dataset. Its average value over

[3]Code is available at: http://dsp.rice.edu/software/jpeg-chest.

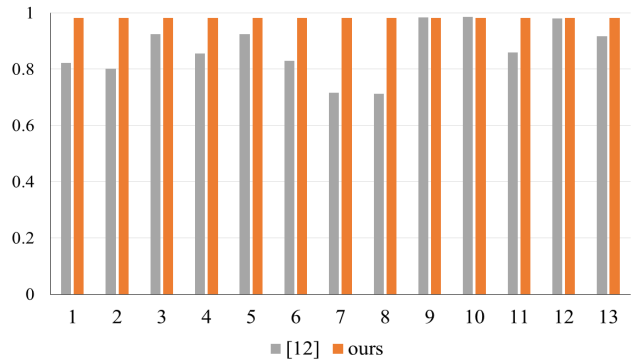

Fig. 8: Performance comparison in terms of AUC for binary classification of single vs double compression.

all filters is AUC $= 0.98$.

In order to compare with state-of-the-art, we focused on the algorithm described in [12], which detects double compression in JPEG images, and especially aims at steganographic applications. Therein, a SVM classifier is fed with a vector of features composed by the histograms of low-frequency DCT coefficients. This algorithm was implemented in its native design, i.e., by two classes; the training set for the first one is composed of 300 images compressed with JPEG-QF $= 90$, while the training set for the second one contains the same 300 images compressed with JPEG-QFs $= 40, 50, 60, 70, 80, 90$ and recompressed with JPEG-QF$_2 = 90$. The test set was built by following the same procedure, but considering 300 different original images. A grid search for the optimal parameters of the Gaussian kernel SVM was performed, which gives an AUC $= 0.92$, with $c = 3$ and $\gamma = -5$.

Finally, we analyzed the performance of the algorithm in [12] when the double compression case includes a filter between the two quantizers; to this end, we used the SVM that is trained as described in the previous paragraph. 13 different test datasets, each of them corresponding to a filter in the dictionary, were built from the same 300 images used for testing in the previous paragraph. Figure 8 shows the AUC obtained for each dataset by our proposal (orange bars), and by the algorithm introduced in [12] (gray bars); it is clear that our method outperforms that in [12]. This is to be expected for filtered images, as [12] does not consider the possibility of using any kind of JPEG-artifact removal between the two JPEG compressions; in fact, when filtering is introduced, the performance of [12] significantly decreases (average AUC $= 0.87$), while our method still provides a very good performance (average AUC $= 0.98$). But notice that even when the applied filter is the identity (filter no. 13), i.e., the scenario for which the method in [12] was originally designed, the results of our scheme (which considers filtered images in the training set, that in turn could mislead the decisions in this case) are better than those given by the algorithm in [12].

Interestingly, for some filters, e.g., filter no. 9, the AUC achieved by the method described in [12] is larger than that obtained by the same method for the identity filter. In order to learn the rationale for this behavior, we trained the 2-classes SVM used by [12] by including in the training set 300 single-compressed images (QF $= 90$) for the first class, while for the second class we compressed those 300 images with JPEG-QFs $= 40, 50, 60, 70, 80, 90$, and then recompressed them with JPEG-QF$_2 = 90$, where filter no. 9 was used after the first compression. The test set was built by using the same procedure, but considering 300 different original images. The obtained AUC is 0.997 (without filtering AUC $= 0.98$ was obtained). The same SVM yields an AUC $= 0.916$ when no filter is applied (i.e., filter no. 13), which is only slightly smaller than the AUC $= 0.917$ obtained by the SVM built without considering any filters. Consequently, it seems that the effect of some filters, such as no. 9, does not significantly perturb, and even improves, the classification features used in [12].

V. Conclusions

A large number of works in the literature have addressed the footprints left by JPEG compression, and studied how those footprints can be exploited to get insight into the image history (quantization table estimation, estimation of the primary quantization table in double compression, single vs double JPEG-compression, etc.). In sharp contrast, very few works (to the best of our knowledge just [11], [15]) consider the scenario where some post-processing is in place (in the cited cases, shifting that produces misalignment, and resizing, respectively) which complicates the operation of any forensic tool, often quite dramatically. We analyzed the impact of a commonly used postprocessing for JPEG images, namely, full-frame linear filtering, on the JPEG compression footprints. Our main target was to exploit the altered footprints for recovering the processing history of an image. To this end, we have modeled the probability distribution of the DCT coefficients of filtered JPEG images, proving the dependence of the corresponding distribution with respect to both the JPEG-QF and the applied filter. Based on this analysis, a set of features, that capture such dependence, is proposed. These features are then input to a linear SVM that uncovers the desired information about the compression quality factor and the filter kernel.

Extensive experimental results show the effectiveness and robustness of the proposed forensic scheme. Interestingly, our method extends very well to other originally unforseen practical cases. We have tested it in a number of different application scenarios, and compared with state-of-the-art schemes natively conceived for those scenarios. In all cases the proposed scheme has shown excellent results, both in the absence and presence of filtering postprocessing.

The previous discussion should not lead to conclude that our detector will be robust against any kind of attack [23]. Indeed, *ad-hoc* counterforensic attacks can be designed in order to fool the current detector; for example, one might use the optimal counterforensic method against histogram-based detectors with non-convex detection regions (as those used in the current work), which was recently introduced in [24].

Future work will be devoted to enlarging the used filter dictionary, and investigating the application of other linear and non-linear filters.

$$Z^{i_8,j_8}(i',j') = \sum_{k_1,k_2} \frac{c(i')}{2}\frac{c(j')}{2}\cos\left(\frac{2k_1+1}{16}\pi i'\right)\cos\left(\frac{2k_2+1}{16}\pi j'\right) z(8i_8+k_1, 8j_8+k_2) \quad (10)$$

$$= \sum_{k_1,k_2}\sum_{l_1,l_2} \frac{c(i')}{2}\frac{c(j')}{2}\cos\left(\frac{2k_1+1}{16}\pi i'\right)\cos\left(\frac{2k_2+1}{16}\pi j'\right) h(l_1,l_2)\cdot y(8i_8+k_1-l_1, 8j_8+k_2-l_2) \quad (11)$$

$$= \sum_{k_1,k_2}\sum_{l_1,l_2}\sum_{m_1,m_2} \phi(k_1,k_2,l_1,l_2,m_1,m_2,i',j')h(l_1,l_2)Y^{b_1,b_2}(m_1,m_2), \quad (12)$$

where

$$\phi(k_1,k_2,l_1,l_2,m_1,m_2,i',j') = \frac{c(i')}{2}\frac{c(j')}{2}\frac{c(m_1)}{2}\frac{c(m_2)}{2}\cos\left(\frac{2k_1+1}{16}\pi i'\right)\cos\left(\frac{2k_2+1}{16}\pi j'\right)\cos\left(\frac{2d_1+1}{16}\pi m_1\right)\cos\left(\frac{2d_2+1}{16}\pi m_2\right),$$

$$b_1 = \left\lfloor \frac{8i_8+k_1-l_1}{8} \right\rfloor, \quad (13)$$

$$b_2 = \left\lfloor \frac{8j_8+k_2-l_2}{8} \right\rfloor, \quad (14)$$

$$d_i = (k_i - l_i) \mod 8.$$

$$Z^{i_8,j_8}(i',j') = \left[\sum_{l_1,l_2}\sum_{k_1\in\Omega_1}\sum_{k_2\in\Omega_2} \phi(k_1,k_2,l_1,l_2,i',j',i',j')h(l_1,l_2)\right] Y^{i_8,j_8}(i',j') \quad + \quad R^{i_8,j_8}(i',j'), \quad (15)$$

APPENDIX A
RELATIONSHIP BETWEEN Z AND Y

The target of this appendix is to establish the relationship between Z and Y. Specifically, (10) computes Z as the 8×8 block-DCT of z, where $\sum_{k_1,k_2}$ indicates the double summation over k_1 and k_2, with $(k_1,k_2)\in\{0,\ldots,7\}$. In turn, z is the convolution of the quantized signal y and the filter kernel h, a fact which is reflected in (11), where $\sum_{l_1,l_2}$ is the double summation over l_1 and l_2, with $(l_1,l_2)\in\{-N,\ldots,N\}$. On the other hand, taking into account that y is the 8×8 block-IDCT of Y, then (12) is obtained, where $\sum_{m_1,m_2}$ is the double summation over m_1 and m_2, with $(m_1,m_2)\in\{0,\ldots,7\}$. Finally, we quantify the contribution of $Y^{i_8,j_8}(i',j')$ on $Z^{i_8,j_8}(i',j')$. In order to do so, we separately consider in (12) those sum indices such that $b_1 = i_8$, $b_2 = j_8$, $m_1 = i'$, and $m_2 = j'$, yielding (15), where $\Omega_1 = \{0,\ldots,7\}\cap\{l_1,\ldots,l_1+7\}$ and $\Omega_1 =\in \{0,\ldots,7\}\cap\{l_2,\ldots,l_2+7\}$. Moreover, $R^{i_8,j_8}(i',j')$ denotes the contribution from other coefficients of Y, and the term within the brackets is denoted by $\gamma_{i',j'}$.

Please note that in this analysis we have neglected the effect of clipping and quantization noise that appears when the pixels are quantized by using a fixed number of bits (typically 8).

ACKNOWLEDGMENT

Research supported by the Illegal use of Internet (INT) call within the Prevention of and Fight against Crime (ISEC) programme of the Home Affairs Department of the European Commission under project NIFTy (Project Number HOME/2012/ISEC/AG/INT/4000003892), the European Union under project REWIND (Grant Agreement Number 268478), the European Regional Development Fund (ERDF) and the Galician Regional Government under agreement for funding the Atlantic Research Center for Information and Communication Technologies (AtlantTIC), and the Galician Regional Government under projects "Consolidation of Research Units" CN/2012/260 and GRC2013/009.

REFERENCES

[1] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*. Morgan Kaufmann Publishers, 2007.

[2] E. Delp, N. Memon, and M. Wu, "Digital forensics," *IEEE Signal Processing Magazine*, vol. 2, no. 26, pp. 14–15, March 2009.

[3] A. Piva, "An overview on image forensics," *ISRN Signal Processing*, vol. 2013, 2013.

[4] 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, June 2006.

[5] 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, September 2010.

[6] 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, March 2008.

[7] V. Conotter and G. Boato, "Analysis of sensor fingerprint for source camera identification," *Electronic Letters*, vol. 47, no. 25, pp. 1366–1367, December 2011.

[8] J. O'Brien and H. Farid, "Exposing photo manipulation with inconsistent reflections," *ACM Transactions on Graphics*, vol. 1, no. 31, pp. 1–11, January 2012.

[9] V. Conotter, G. Boato, and H. Farid, "Detecting photo manipulation on signs and billboards," in *IEEE International Conference on Image Processing*, Hong Kong, China, September 2010, pp. 1741–1744.

[10] D. Fu, Y. Shi, and W. Su, "A generalized Benford's law for JPEG coefficients and its applications in image forensics," in *SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents*, vol. 6505, San Jose (CA), January 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, April 2012.

[12] T. Pevny and J. Fridrich, "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, June 2008.
[13] S. Milani, M. Tagliasacchi, and S. Tubaro, "Discriminating multiple JPEG compression using first digit features," in *IEEE International Conference on Acoustic, Speech and Signal processing*, Kyoto, Japan, March 2012, pp. 2253–2256.
[14] M. Kirchner and R. Böhme, "Hiding traces of resampling in digital images," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 582–592, December 2008.
[15] T. Bianchi and A. Piva, "Reverse engineering of double JPEG compression in the presence of image resizing," in *IEEE International Workshop on Information Forensics and Security*, Tenerife, Spain, December 2012, pp. 127–132.
[16] V. Conotter, P. Comesaña, and F. Pérez-González, "Forensic analysis of full-frame linearly filtered JPEG images," in *IEEE International Conference on Image Processing*, Melbourne, Australia, September 2013, pp. 4517–4521.
[17] M. Barni and F. Bartolini, *Watermarking systems engineering - Enabling digital assets security and other applications*. CRC Press, 2004.
[18] V. Conotter, P. Comesaña, and F. Pérez-González, "Joint detection of full-frame linear filtering and JPEG compression in digital images," in *IEEE International Workshop on Information Forensics and Security*, Guangzhou, China, November 2013, pp. 156–161.
[19] G. Schaefer and M. Stich, "UCID - an uncompressed colour image database," in *SPIE Conference on Storage and Retrieval Methods and Applications for Multimedia*, San Jose (CA), January 2004, pp. 472–480.
[20] P. Moulin and M. K. Mihçak, "A framework for evaluating the data-hiding capacity of image sources," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 1029–1042, September 2002.
[21] 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, no. 9, pp. 1077–1085, September 2005.
[22] 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, April 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
[23] W. Fan, K. Wang, F. Cayre, and Z. Xiong, "JPEG anti-forensics with improved tradeoff between forensic undetectability and image quality," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 8, pp. 1211–1226, August 2014.
[24] P. Comesaña and F. Pérez-González, "The optimal attack to histogram-based forensic detectors is simple(x)," in *IEEE Workshop on Information Forensics and Security*, Atlanta, GA, USA, December 2014, pp. 1730–1735.

Valentina Conotter Valentina Conotter received her M.S. degree (cum laude) in Telecommunication Engineering from the University of Trento in 2007, and her Ph.D. in Telecommunication Engineering from the ICT International Doctorate School, within the Dept. of Engineering and Computer Science (DISI) at the University of Trento in 2011. Between 2009-2010, she was a visiting student in the Image Science Group at Dartmouth College. In 2011-2013 she was employed as a postDoc at the Signal Theory & Communications Department at the University of Vigo, Spain. In 2013-2015 she was a postdoc researcher at the University of Trento, working within the Multimedia Signal Processing and Understanding Lab. Her research was mainly focused on active and passive multimedia forensics, including digital watermarking and digital image and video forensics. Currently she is Project Manager within the Research and Innovation Area at Social IT, Trento, Italy, focusing her attention on social and health care services.

Pedro Comesaña (M'08-SM'15) received the Telecommunication Engineer degree from the University of Vigo, Vigo, Spain in 2002 and the Ph. D. degree in telecommunications engineering from the same institution in 2006.

During his Ph. D. studies Dr. Comesaña stayed at the Technische Universiteit Eindhoven (The Netherlands, 2004); then, he hold post-doc positions at the University College Dublin (Ireland, 2006), Università degli Studi di Siena (Italy, 2007-2008), and University of New Mexico (Albuquerque, USA, 2010-2011). In 2008 Dr. Comesaña joined the faculty of the School of Telecommunication Engineering, University of Vigo, as an Assistant Professor, where he is currently an Associate Professor. His research interests lie in the areas of multimedia security (including watermarking and forensics), and digital communications. He has coauthored several international patents related to watermarking for video surveillance, and fingerprinting of audio signals.

Dr. Comesaña has co-authored over 50 papers in leading international journals and peer-reviewed conferences; he was recipient of IEEE-WIFS'2014 Best Paper Award. He has participated in the European projects ECRYPT, REWIND, and NIFTY. Currently, Dr. Comesaña serves as an Associate Editor of IET Information Security (from 2012), and is a member of both the IEEE SPS Information Forensics and Security-Technical Committee and the IEEE SPS Student Services Committee; furthermore, he is also Technical co-Chair of ACM IH&MMSec 2015, and Area Chair of IEEE ICIP 2015.

Fernando Pérez-González (M'90-SM'09) received the Telecommunication Engineer degree from the University of Santiago, Santiago, Spain in 1990 and the Ph.D. degree in telecommunications engineering from the University of Vigo, Vigo, Spain, in 1993.

He joined the faculty of the School of Telecommunication Engineering, University of Vigo, as an assistant professor in 1990, where he is currently a Professor. From 2009 to 2011 he was the Prince of Asturias Endowed Chair of Information Science and Technology with the University of New Mexico, Albuquerque, NM, USA, where he is currently a Research Professor. His research interests lie in the areas of digital communications, adaptive algorithms, privacy enhancing technologies, and information forensics and security. He has coauthored several international patents related to watermarking for video surveillance, integrity protection of printed documents, fingerprinting of audio signals, and digital terrestrial broadcasting systems.

Prof. Prez-Gonzlez has co-authored over 50 papers in leading international journals and more than 160 peer-reviewed conference papers. He has been the principal investigator of the University of Vigo group which participated in several European projects, including CERTIMARK, ECRYPT, REWIND, NIFTY and WITDOM. From 2007 to 2010 he was Program Manager of the Spanish National R&D Plan on Electronic and Communication Technologies, Ministry of Science and Innovation. From 2007 to 2014 he was Founding Executive Director of the Galician Research and Development Center in Advanced Telecommunications (GRADIANT). He served as an Associate Editor of IEEE SIGNAL PROCESSING LETTERS (2005-2009) and the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (2006-2010). Currently, he is an Associate Editor of the LNCS TRANSACTIONS ON DATA HIDING AND MULTIMEDIA SECURITY, and the EURASIP INTERNATIONAL JOURNAL ON INFORMATION FORENSICS AND SECURITY.