Chapter 3
Multimedia Forensics Before the Deep Learning Era

Davide Cozzolino and Luisa Verdoliva

Abstract Image manipulation is as old as photography itself, and powerful media editing tools have been around for a long time. Using such conventional signal processing methods, it is possible to modify images and videos obtaining very realistic results. This chapter is devoted to describe the most effective strategies to detect the widespread manipulations that rely on traditional approaches and do not require a deep learning strategy. In particular, we will focus on manipulations like adding, replicating, or removing objects and present the major lines of research in multimedia forensics before the deep learning era and the rise of deepfakes. The most popular approaches look for artifacts related to the in-camera processing chain (camera-based clues) or the out-camera processing history (editing-based clues). We will focus on methods that rely on the extraction of a camera fingerprint and need some prior information on pristine data, for example, through a collection of images taken from the camera of interest. Then we will shift to blind methods that do not require any prior knowledge and reveal inconsistencies with respect to some well-defined hypotheses. We will also briefly review the most interesting features of machine learning- based methods and finally present the major challenges in this area.

3.1 Introduction

Digital image manipulation has a long history, and nowadays several powerful editing tools exist that allow creating realistic results that can easily fool visual scrutiny. Very common operations are adding, replicating, or removing objects, as in the examples

D. Cozzolino · L. Verdoliva (✉)
University Federico II of Naples, via Claudio 21, Naples, Italy
e-mail: verdoliv@unina.it

D. Cozzolino
e-mail: davide.cozzolino@unina.it

C. Rathgeb et al. (eds.), *Handbook of Digital Face Manipulation and Detection*,
Advances in Computer Vision and Pattern Recognition,
https://doi.org/10.1007/978-3-030-87664-7_3

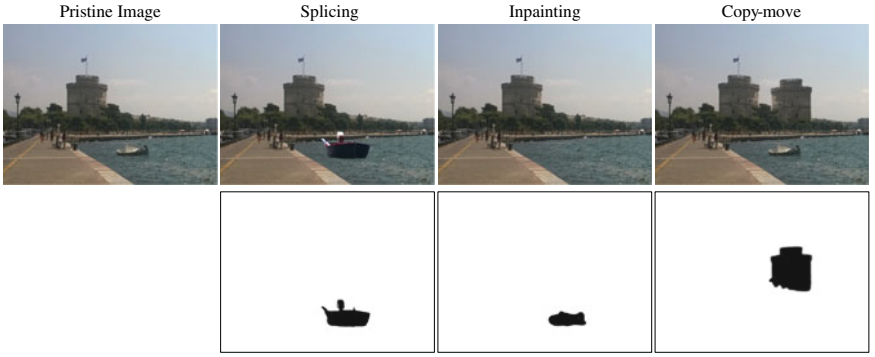

Fig. 3.1 Examples of image manipulations carried out using conventional media editing tools. First row: adding an object (splicing), removing an object (inpainting), and duplicating an object (copy-move). Second row: corresponding binary ground truths that indicate the pixels that have been modified in the image

of Fig. 3.1. A new object can be inserted by copying it from a different image (splicing), or from the same image (copy-move). Instead, an existing object can be deleted by extending the background to cover it (inpainting). Some suitable post-processing, like resizing, rotation, and color adjustment, can also be applied to better fit the object to the scene, both to improve the visual appearance and to guarantee coherent perspective and scale.

In the last few years, there has been intense research toward the design of methods for reliable image integrity verification [63]. Some tools discover physical inconsistencies [39, 41], regarding, for example, shadows or illumination or perspective, which may also be noticed by an attentive observer. In most cases, however, well-crafted forgeries leave no visible traces and appear semantically correct. Nonetheless, digital manipulations typically modify the underlying statistics of the original source, leaving a trail of traces which, although invisible to the eye, can be exploited by pixel-level analysis tools. In fact, each image is characterized by a number of features which depend on the different phases of its history, from the very same acquisition process to the internal camera processing (e.g., demosaicing and compression), to all external processing and editing operations (see Fig. 3.2). Therefore, by studying possible deviations of such features from their expected behavior, one can establish with good confidence whether image integrity has been violated.

Based on this general principle, a certain number of approaches have been proposed. For example, the acquisition process leaves on each image a "camera fingerprint", the photo-response non-uniformity noise (PRNU), unique for each specific device. Armed with this fingerprint, one can reliably discover and localize various types of attacks. It is also possible to use model-specific rather than device-specific features, related to manufacturing choices (like the color filter array) and in-camera processing (like the demosaicing algorithm) peculiar of each brand and model. As for external processing, the lion's share is taken by methods exploiting the proper-

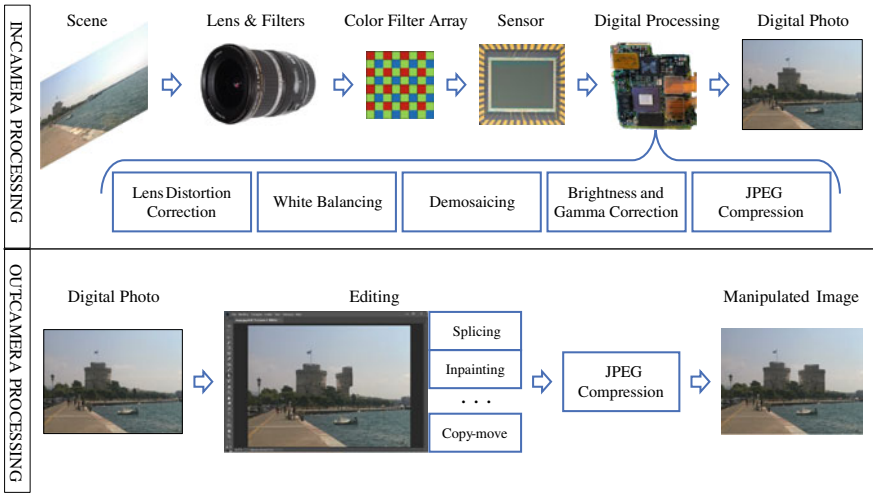

Fig. 3.2 An image is captured using an acquisition system whose basic components are represented in this figure (in-camera processing); the image can then be edited in several ways (out-camera processing)

ties of JPEG compression. Indeed, after a forgery is performed, the image is very often saved again in a JPEG compressed format. Therefore, by studying anomalies in DCT coefficients due, for example, to double quantization, or JPEG grid misalignments, integrity violation can be detected and localized. Finally, a very common form of forgery involves copy-moving image regions to duplicate or hide objects. The presence of identical regions in the image represents by itself a distinctive feature indicating manipulation, which may be discovered efficiently by several approaches, even in the presence of rotation, resizing, and other geometric distortions. Turning to videos, very simple manipulations consist in deleting or replicating entire frames. Of course, also in this case it is possible to insert or hide objects using more sophisticated editing tools [52].

This chapter will present an overview of some of the most effective tools for image forgery detection and localization that have been proposed before the rise of deep learning. In particular, we will focus on passive methods that look at the image content and disregard the associated metadata information. The most popular approaches look for artifacts related to the in-camera processing chain (camera-based clues) or the out-camera processing history (editing-based clues). These approaches often follow a model-based paradigm typically relying on statistical analyses or are based on handcrafted features and apply more classical machine learning tools. Each method relies on its own set of hypotheses, which may or may not hold for a specific manipulation, thereby limiting its applicability to a subset of cases. For example, the camera PRNU can be reliably estimated only if the camera itself is available or a large number of images taken from it. Likewise, methods thought for copy-move discovery are obviously ineffective in the presence of a splicing. Some of them are much more general, since they are based on detecting anomalies in the noise residuals.

A defining property of the approaches proposed so far is the prior knowledge they rely upon, which impacts their suitability for real-world applications. First, we will describe PRNU-based methods that require a collection of images taken from the camera of interest. Then we will present blind methods, where no prior knowledge is required. Finally, we will give a short review of machine learning-based methods which rely on a suitable training set comprising both pristine and manipulated data.

3.2 PRNU-Based Approach

Manufacturing imperfections in the silicon wafer used for the imaging sensor generate a unique sensor pattern, called photo- response non-uniformity (PRNU) noise. It is specific to each individual camera, stable in time, and independent of the scene. All images acquired by a given camera bear traces of its PRNU pattern, hence it can be considered as a sort of camera fingerprint and used for source attribution tasks, as well as for image forgery detection. If a region of the image is tampered with, the corresponding PRNU pattern is removed, which allows one to detect the manipulation.

PRNU-based forgery detection was first proposed in [49], and it is based on two main steps: *(i)* the PRNU pattern is estimated off-line from a large number of images taken from the camera, and *(ii)* the target image PRNU is estimated at test time, by means of a denoising filter, and compared with the reference (see Fig. 3.3). This approach relies on some important prior knowledge, since it assumes the availability of a certain number of images taken from the device itself. On the other hand, it is an extremely powerful approach, since it can detect every type of attack: whenever an anomaly arises due to the absence of the camera fingerprint, manipulation can be detected.

Beyond this standard methodology, there are several alternatives proposed in the literature. It is possible to model the strong spatial dependencies present in an image through a Markov Random Field so as to make joint rather than isolated decisions [16], or to rely on discriminative random fields [12] and multi-scale analysis [43]. It is worth noting that the PRNU-based approach can be also extended to blind scenarios, where no prior information about the camera is known provided a suitable clustering procedure identifies the images which share the same PRNU [20, 21]. It is even possible to recover some information about PRNU by estimating it from a single image or a group of frames in a video [51, 53, 60].

In the following, we will describe the basic approach proposed in [14]. Let y be a digital image, defined on a rectangular lattice Ω, with y_i the value at site $i \in \Omega$, observed at the camera output, either as a single color band or the composition of multiple color bands. Let us assume in a simplified model [37] that y can be written as[1]

$$y = (1 + k)x + \theta = xk + x + \theta \tag{3.1}$$

[1] All the operations are intended pixel-wise.

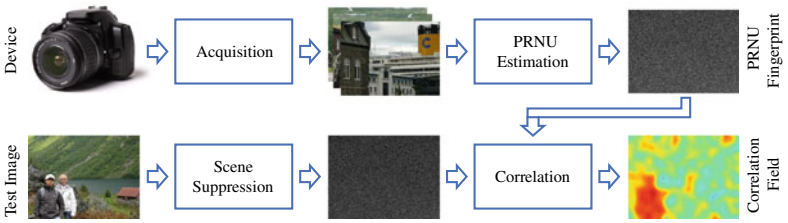

Fig. 3.3 PRNU-based forgery localization procedure. Top: the device PRNU pattern is estimated by averaging a large number of noise residuals. Bottom: the image PRNU pattern is estimated by denoising, and compared with the reference pattern: the low values in the correlation field suggest a possible manipulation

where x is the ideal noise-free image, k the camera PRNU, and θ an additive noise term which accounts for all types of disturbances. The PRNU k is the signal of interest, very weak w.r.t. both additive noise θ and the ideal image x. In this context also, the image x plays the role of unwanted disturbance, since our goal is to decide whether or not the image PRNU comes from the camera under test so as to detect possible forgeries. To increase the signal-to-noise ratio, we can subtract from y an estimate of the ideal image $\widehat{x} = f(y)$ obtained through denoising, in order to compute the so-called noise residual

$$r = y - \widehat{x} = yk + (x - y)k + (x - \widehat{x}) + \theta = yk + n \tag{3.2}$$

where, for convenience, k multiplies the observed image y rather than the unknown original x, and the small difference term $(x - y)k$ has been included, together with the denoising error $(x - \widehat{x})$ and other disturbances in a single noise term n.

In the following, we describe in more detail the image integrity verification procedure proposed in [14] which comprises the following basic steps:

- estimation of the camera PRNU (off-line);
- computation of image noise residual and of derived statistics;
- sliding-window pixel-wise forgery detection test.

3.2.1 PRNU Estimation

As a preliminary step, the true camera PRNU pattern should be reliably estimated. This requires that either the target camera, or a large number of photos taken by it, is available. Note that the PRNU is a deterministic signal, as opposed to the other image components, and it can be easily estimated starting from the noise residuals. In addition, one can take care of using mostly uniform images (e.g., off-focus pictures of a cloudy sky) to further improve accuracy or to use fewer images to obtain the same performance. In these conditions, the maximum likelihood estimate of the PRNU

from M given images is computed in [14] as

$$\widehat{k} = \sum_{m=1}^{M} y_m r_m \Big/ \sum_{m=1}^{M} y_m^2 \tag{3.3}$$

where the weights y_m account for the fact that dark areas of the image present an attenuated PRNU and hence should contribute less to the overall estimate. Of course, this is only an estimate, however, for the sake of simplicity, we will neglect the estimation error and will assume to know the camera PRNU perfectly, that is $\widehat{k} = k$.

3.2.2 Noise Residual Computation

In the second step of the algorithm, we compute the noise residual r and suppress most of the scene content by subtracting a denoised version of the image itself:

$$r = y - f(y) = y - \hat{x} \tag{3.4}$$

where f denotes a denoising algorithm. Even in the best case, with perfect denoising, $\hat{x} = x$, the remaining noise term is likely to dominate r which, therefore, will be only weakly correlated with the camera PRNU. In the presence of textured areas, however, denoising is typically less accurate and some signal components leak into the residual contributing to reducing the operative SNR, to the point of making detection virtually impossible. Especially in these areas, the effectiveness of the denoising algorithm becomes crucial for the overall performance.

3.2.3 Forgery Detection Test

Assuming $z = yk$, the detection problem can be formulated as a binary hypothesis test between hypothesis H_0 and H_1. Under hypothesis H_0 the camera PRNU is absent, hence the pixel has been tampered, while under hypothesis H_1, PRNU is present, hence the pixel is genuine:

$$\begin{cases} H_0 : r_i = n_i \\ H_1 : r_i = z_i + n_i \end{cases} \tag{3.5}$$

Notice that, since we focus on the detection of forgeries, denoted by the *absence* of the PRNU, the role of two hypotheses is inverted w.r.t. what is usual. The true and estimated pixel classes will be denoted by u_i and $\widehat{u}_i$, both defined in $\{0, 1\}$, while the detection test is based on the normalized correlation index between r_{W_i} and z_{W_i}, the restrictions of r and z, respectively, to a window W_i centered on the target pixel:

$$\rho_i = \mathrm{corr}(r_{W_i}, z_{W_i}) = \frac{(r_{W_i} - \overline{r}_{W_i}) \odot (z_{W_i} - \overline{z}_{W_i})}{\|r_{W_i} - \overline{r}_{W_i}\| \cdot \|z_{W_i} - \overline{z}_{W_i}\|} \tag{3.6}$$

where $\odot$ denotes inner product, and the usual definitions hold for mean, norm, and inner product

$$\overline{x} = \frac{1}{K}\sum_{i=1}^{K} x_i, \qquad \|x\|^2 = \sum_{i=1}^{K} x_i^2, \qquad x \odot y = \sum_{i=1}^{K} x_i y_i \tag{3.7}$$

Pixel labeling is obtained by comparing the decision statistic with a threshold γ_1

$$\widehat{u}_i = \begin{cases} 0 & \rho_i < \gamma_1 \\ 1 & \text{otherwise} \end{cases} \tag{3.8}$$

To ensure the desired false acceptance rate (FAR), which is a small probability that a tampered pixel is identified as genuine, the threshold is set using the Neyman-Pearson approach. The pdf of ρ under hypothesis H_0 is estimated by computing the correlation between the camera PRNU and a large amount of noise residuals coming from other cameras, and using standard density fitting techniques. To obtain reliable estimates, rather large square blocks should be used; a dimension of $128\times$ 128 pixels represents a good compromise [14].

Once the desired FAR is fixed, the objective is to minimize the false rejection rate (FRR), which is the probability that a genuine pixel is declared tampered. This is not an easy task, since under hypothesis H_1, the decision statistic is influenced by the image content. In fact, even in the absence of forgery, the correlation might happen to be very low when the image is dark (since y multiplies the PRNU), saturated (because of intensity clipping), or in very textured areas where denoising typically does not perform well and some image content leaks into the noise residual. One possible solution to this problem is to include a "predictor" [14], which based on local images features, such as texture, flatness, and intensity, computes the expected value $\widehat{\rho_i}$ of the correlation index under hypothesis H_1. When $\widehat{\rho_i}$ is too low, indicating that, even for a genuine pixel, one could not expect a correlation index much larger than 0, the pixel is labeled as genuine, the less risky decision, irrespective of the value of ρ_i. Therefore, the test becomes

$$\widehat{u}_i = \begin{cases} 0 & \rho_i < \gamma_1 \ \text{ AND } \ \widehat{\rho_i} > \gamma_2 \\ 1 & \text{otherwise} \end{cases} \tag{3.9}$$

The second threshold γ_2 is chosen heuristically by the user and separates, in practice, reliable regions from problematic ones. It is worth underlining that the refined decision test (3.9) can only reduce the false rejection rate but does not increase (actually it might reduce) the probability of detecting an actual forgery. In addition, the choice of the threshold itself is not obvious and can significantly impact the performance. Note also that the final binary map needs some post-processing operations to remove

random errors and better define the shape of the forgery. This is typically done by means of morphological filtering.

3.2.4 Estimation Through Guided Filtering

As already highlighted in the previous section, a major issue with PRNU-based analysis is the impossibility to perfectly denoise the image. As a consequence, the noise residual contains traces of the image content that increase the false acceptance rates. To address this problem, it is possible to improve the denoising algorithm as done in [15], where wavelet-based denoising has been replaced by a nonlocal approach. Another possibility is to rely on the use of guided filtering [17], a strategy that turns out to be especially helpful when small forgeries are present.

In order to better understand this approach, we will elaborate some more on Eq. (3.6) and introduce some simplifications. First of all, we neglect the means (which are typically negligible) and, considering that the terms at the denominator serve only to normalize the correlation, focus on the scalar product on the numerator. Remember that $z = yk$ is the camera PRNU multiplied point-wise by the input image and, likewise, $r = hy + n$ is the noise residual, with h the observed PRNU which might or might not coincide with k. Therefore, if we divide all terms point-wise by y, we obtain the quantity

$$\tau_i = \frac{1}{|W_i|} \sum_{j \in W_i} \frac{r_j}{y_j} \frac{z_j}{y_j} = \frac{1}{|W_i|} \sum_{j \in W_i} \left(h_j + \frac{n_j}{y_j} \right) k_j \tag{3.10}$$

By defining a new noise field $\eta = nk/y$, and introducing generic weights ω_{ij}, Eq. (3.10) becomes

$$\tau_i = \sum_{j \in W_i} \omega_{ij} (h_j k_j + \eta_j) \tag{3.11}$$

This can be interpreted as the linear filtering of the image hk affected by the additive noise η. In Eq. (3.10), the weights are all equal to one $1/|W_i|$, hence, a simple boxcar filtering is carried out.

Assuming that the whole analysis window is homogeneous, either genuine ($h = k$) or forged ($h \neq k$) and, for the sake of simplicity, that y is constant over the window, so that $E[\eta_i] = \sigma_\eta^2$, we can characterize the random variable τ as

$$E[\tau] = \begin{cases} \langle k^2 \rangle_i & h = k \\ 0 & h \neq k \end{cases} \tag{3.12}$$

$$\mathrm{VAR}[\tau] = \sigma_\eta^2 \sum_j \omega_{ij}^2 \tag{3.13}$$

where $\langle k^2 \rangle$ is the power of the camera PRNU estimated over W_i. In this condition, using uniform weights $\omega_{ij} = 1/|W_i|$ is indeed optimal, as it minimizes the variance of the estimate, and maximizes the probability of deciding correctly. However, if some of the predictor pixels are not homogeneous with the target, that is, forged instead of genuine or vice versa, the estimate will suffer a systematic bias, namely the means will not be 0 or $\langle k^2 \rangle$ anymore, but some intermediate values, heavily affecting the decision performance. In this case, the uniform weights are no more optimal, in general, and one should instead reduce the influence of heterogeneous pixels by associating a small or even null weight with them. This is exactly the problem of small-size forgeries. By using a large analysis window with fixed weights, we happen to include pixels of different nature, and the decision variable becomes strongly biased and basically useless, even in favorable (bright, smooth, and unsaturated) areas of the image. If we could find and include in the estimation only predictors homogeneous with the target, all biases would disappear, at the cost of an increased estimation variance.

The bias/variance trade-off is indeed well-known in the denoising literature. This problem has received a great deal of attention, recently, in the context of nonlocal filtering, where predictor pixels are weighted based on their expected similarity with the target. The similarity, in its turn, is typically computed by comparing patches of pixels centered on the target and the predictor pixels, respectively. This approach cannot work with our noise-like input image, rz, as it lacks the structures necessary to compute a meaningful similarity measure. However, we can take advantage of the original observed image y, using it as a "pilot" to compute similarities, and applying the resulting weights in the actual filtering of the rz field. This basic idea is implemented in [17] by means of guided filtering, a recently proposed technique which implements nonlocal filtering concepts by leveraging heavily on the use of a pilot image associated with the target image [34].

In Fig. 3.4, we show the detection performance, measured in terms of probability of detection P_D versus probability of false alarm (P_{FA}), obtained when a square forgery is placed at the center of the image. The performance obtained with the plain boxcar filter (left) and guided filtering (right) is almost the same when large forgeries are considered (128×128 pixels). However, guided filtering becomes more and more preferable as the forgeries become smaller, up to the limiting case of 48×48 pixels. This is also clear from the examples shown in Fig. 3.5, where the correlation field shows the ability of guided filtering to detect even very small forgeries, which are completely lost using boxcar filtering.

3.3 Blind Methods

Blind approaches do rely exclusively on the media asset under analysis and reveal inconsistencies with respect to some well-defined hypotheses. In particular, they look for a number of specific artifacts originated by in-camera or out-camera processing (Fig. 3.2). For example, the demosaicing algorithm is typically different for different camera models. Therefore, when a manipulation involves the composition of parts

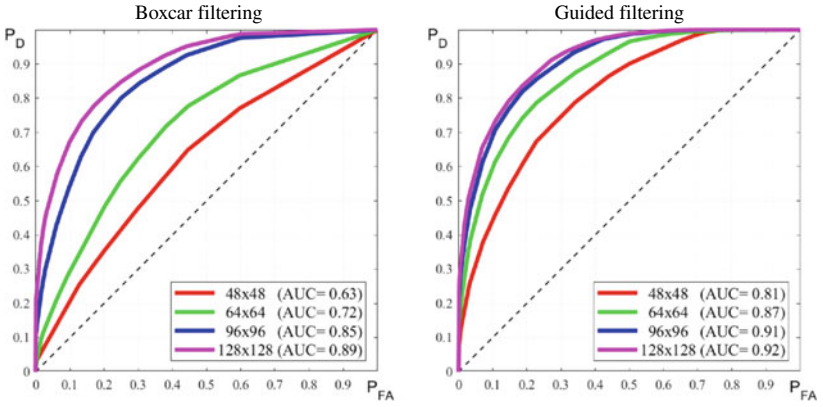

Fig. 3.4 ROCs obtained with boxcar filtering (left) and guided one (right) by varying the forgery size. Each ROC is the upper envelope of pixel-level (P_D, P_{FA}) points obtained as the algorithm parameters vary. We used a test set of 200 uncompressed 768×1024-pixel images with a square forgery at the center, drawn at random from a different image

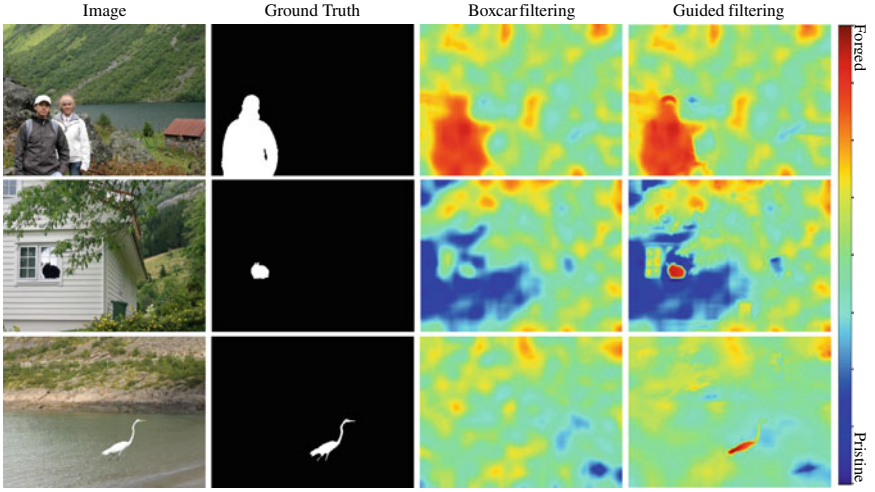

Fig. 3.5 Comparison between boxcar and guided filtering. From left to right: forged image, ground truth, and the correlation field computed using boxcar and guided filtering

of images acquired from different models, demosaicing-related spatial anomalies arise. Likewise, the out-camera editing process may introduce a specific correlation or disrupt fingerprint-like camera-specific patterns. Of course, most of these traces are very subtle and cannot be perceived at a visual inspection. However, once properly emphasized, they represent a precious source of information to establish digital integrity.

For example, most digital cameras use a color filter array (CFA), with a periodic pattern, so that each individual sensor element records light only in a certain range of wavelengths (i.e., red, green, and blue). The missing color information is then interpolated from surrounding pixels, an operation known as demosaicing. This process introduces a subtle periodic correlation pattern in all acquired images. Whenever a manipulation occurs, this periodic pattern is perturbed. In addition, since CFA configuration and interpolation algorithms are specific to each camera model [8, 11], when a region is spliced in a photo taken by another camera model, its periodic pattern will appear anomalous. One of the first methods to exploit these artifacts was proposed by Popescu and Farid [57] back in 2005, based on a simple linear model to capture periodic correlations. Of course, periodic signals produce strong peaks in the Fourier domain. The problem can be also recast in a Bayesian framework, as proposed in [29], obtaining a probability map in output which allows for fine-grained localization of image tampering.

In the following, we will describe blind approaches that rely on noise patterns, compression, and editing artifacts.

3.3.1 Noise Patterns

Instead of focusing on a specific camera artifact, a more general approach is to highlight noise artifacts introduced by the whole acquisition process, irrespective of their specific origin. The analysis of *local* noise level may help reveal splicings, as shown in [50, 56], because different cameras are characterized by different intrinsic noise.

To define expressive features that are able to capture traces left locally by in-camera processing, in [23] the high-pass noise residual of the image is used and then co-occurrence-based features are extracted to capture local correlations. These features, known as rich models, are inspired by the work done in steganalysis [30], which pursue a very similar goal, i.e., detecting hidden artifacts in the signal. These features have been used successfully in a supervised learning setting for the detection task of the first IEEE IFS-TC Image Forensics Challenge [19, 20]. To form the noise residual image, r, only a linear high-pass filter of the third order has been considered of all the models proposed in [30]. In formulas

$$r_{ij} = x_{i,j-1} - 3\,x_{i,j} + 3\,x_{i,j+1} - x_{i,j+2} \tag{3.14}$$

where x and r are the original image and the noise residual, respectively, and i, j indicate spatial coordinates. The next step is to compute residual co-occurrences along the vertical and horizontal directions. First of all, residuals are quantized, using a very small number of bins to obtain a limited feature length and then truncated as

$$\widehat{r}_{ij} = \mathrm{trunc}_T(\mathrm{round}(r_{ij}/q)) \tag{3.15}$$

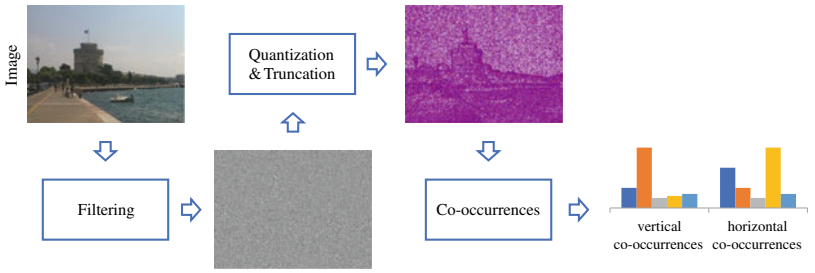

Fig. 3.6 Block diagram for computing residual co-occurrences: high-pass filtering, quantization-truncation operation, and the computation of the co-occurrence histogram

with q the quantization step and T the truncation value. Co-occurrences are computed on four pixels in a row, that is,

$$C(k_0, k_1, k_2, k_3) = \sum_{i,j} I(\widehat{r}_{i,j} = k_0, \widehat{r}_{i+1,j} = k_1, \widehat{r}_{i+2,j} = k_2, \widehat{r}_{i+3,j} = k_3)$$

where $I(A)$ is the indicator function of event A, equal to 1 if A holds and 0 otherwise. The homologous column-wise co-occurrences are pooled with the above based on symmetry considerations. A block diagram is presented in Fig. 3.6.

Different from [30], the normalized histograms are passed through a square-root non-linearity, to obtain a final feature with unitary L2 norm. In fact, in various contexts, such as texture classification and image categorization, histogram comparison is performed by measures such as χ^2 or Hellinger that are found to work better than the Euclidean distance. After square rooting, the Euclidean distance between features is equivalent to the Hellinger distance between the original histograms. We consider two different scenarios for image forgery localization, supervised and unsupervised. In both cases, we will follow an anomaly detection rule, building a model for the host-camera features based on a fraction of the image under analysis.

- *Supervised scenario.* In this case, the user is required to select a bounding box, which will be subject to the analysis, while the rest of the image is used as a training set. In Fig. 3.7, we show some examples where some specific areas of the images are selected and then analyzed. The analysis is carried out in sliding-window modality, using blocks of size $W \times W$, from which the normalized histogram of co-occurrences, $\mathbf{h}$, is extracted. The N blocks taken from the training area are used to estimate in advance mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ of the feature vector:

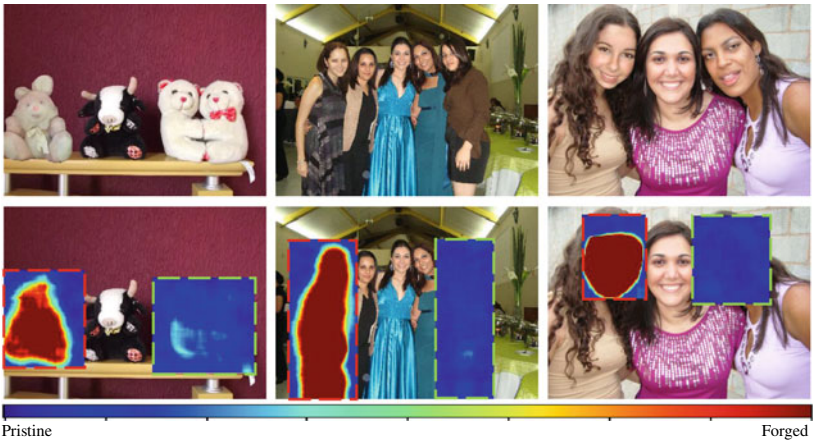

Fig. 3.7 Detecting noise artifacts in supervised modality. If a suspicion region is present, the analysis can be restricted to the region of interest (RoI), and the rest of the image is used as a reference for the pristine data

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{h_n} \tag{3.16}$$

$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{h_n}-\boldsymbol{\mu})(\mathbf{h_n}-\boldsymbol{\mu})^{\mathbf{T}} \tag{3.17}$$

Then, for each block of the test area, the associated feature $\mathbf{h}'$ is extracted, and its Mahalanobis distance w.r.t. the reference feature $\boldsymbol{\mu}$ is computed

$$D(\mathbf{h}', \boldsymbol{\mu}; \boldsymbol{\Sigma}) = (\mathbf{h}'-\boldsymbol{\mu})^{\mathbf{T}}\boldsymbol{\Sigma}^{-\mathbf{1}}(\mathbf{h}'-\boldsymbol{\mu}) \tag{3.18}$$

Large distances indicate blocks that deviate significantly from the model. In the output map provided to the user, each block is given a color associated with the computed distance. Note that the user may repeat the process several times with different bounding boxes, implying that a meaningful analysis can be conducted even in the absence of any initial guess on the presence and location of a forgery.

- *Unsupervised scenario.* In this case, after the feature extraction phase, carried out on the whole image with unit stride, we rely on an automatic algorithm to jointly compute the model parameters and the two-class image segmentation and resort to a simple expectation-maximization (EM) clustering.
  As input, we need the mixture model of the data, namely the number of classes, their probabilities, $\pi_0, \pi_1, \ldots$, and the probability model of each class. For us, the number of classes is always fixed to two, corresponding to the genuine area of the image (hypothesis H_0) and the tampered area (hypothesis H_1). We will consider two cases for the class models:

1. both classes are modeled as multivariate Gaussian

$$p(\mathbf{h}) = \pi_0 \mathcal{N}(\mathbf{h}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) + \pi_1 \mathcal{N}(\mathbf{h}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$

2. class H_0 is modeled as Gaussian, while class H_1 is modeled as Uniform over the feature domain Ω,

$$p(\mathbf{h}) = \pi_0 \mathcal{N}(\mathbf{h}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) + \pi_1 \alpha_1 \mathbf{I}(\Omega)$$

We note explicitly that the Gaussian model is only a handy simplification, lacking more precise information on the feature distribution. The first model is conceived for the case when the forged area is relatively large w.r.t. the whole image. Therefore, the two classes have the same dignity, and can be expected to emerge easily through the EM clustering. The block-wise decision statistic is the ratio between the two Mahalanobis distances.
When the forged region is very small, instead, the intra-class variability, mostly due to image content (e.g., flat vs. textured areas) may become dominant w.r.t. inter-class differences, leading to wrong results. Therefore, we consider the Gaussian-Uniform model, which can be expected to deal better with these situations, and in fact has been often considered to account for the presence of outliers, e.g., [58]. Note that, in this case, the decision test reduces to comparing the Mahalanobis distance from the Gaussian model with a threshold λ as already done in [64]. Typically, forgeries are quite small with respect to the dimension of the image and often the latter model gives more satisfying results (some examples are shown in Fig. 3.8). This idea has been extended to videos in [54] where the noise residuals of consecutive frames are analyzed and suitable features are extracted to discover traces of both intra-frame and inter-frame manipulations.

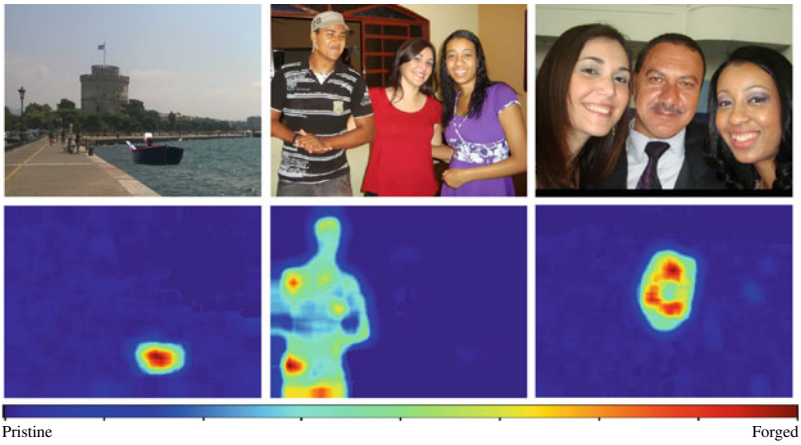

Fig. 3.8 Detecting noise artifacts in unsupervised modality (splicebuster). A clustering algorithm is used to distinguish pristine data from forged ones

3.3.2 Compression Artifacts

Exploiting compression artifacts is a very powerful tool in image forensics. Most images are compressed using JPEG coding standard and whenever an image is edited, it will be subjected to a new compression step. An early popular approach is to exploit the so-called block artifact grid (BAG). In fact, because of the block-wise JPEG processing, discontinuities appear along the block boundaries of compressed images, giving rise to a distinctive and easily detected grid-like pattern [26]. In the presence of splicing or copy-move manipulations, the BAGs of inserted object and host image typically mismatch, enabling detection [45, 47].

Another common and very effective approach relies on double compression traces. In fact, when a JPEG-compressed image undergoes a local manipulation and is compressed again, double compression artifacts appear all over the image except in the forged region [48]. These artifacts change depending on whether the two compressions are spatially aligned or not [10, 13]. Other methods [32, 44, 55] look for anomalies in the statistical distribution of the original DCT coefficients assumed to comply with the Benford law. More specifically, this empirical law states that the probability distribution of the first digits of DCT coefficients is logarithmic:

$$p(d) = \log_{10}\left(1 + \frac{1}{d}\right) \tag{3.19}$$

If the image is modified, for example, double compressed, it will not follow anymore such distribution. In Fig. 3.9, we show an example of DCT coefficient histogram for a single compressed image and a double compressed one, together with the distribution of the first 14 AC coefficients of the DCT block.

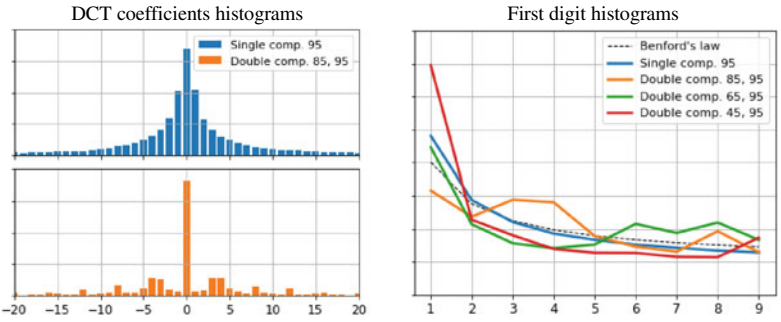

Fig. 3.9 Histograms relative to the first 14 AC coefficients in the DCT block. On the left, the histograms for single and double compression. The single compression image satisfies the Laplacian distribution; this does not happen for the double compressed image. On the right, the histograms of the first digits for single and double compressed images. In the first case, the distribution follows Benford's law, while double compressed images deviate from such distribution

Another approach relies on the so-called JPEG ghosts [27] that arise in the manipulated area when two JPEG compressions use the same quality factor (QF). To highlight ghosts, the target image is compressed at all QFs and analyzed. This approach is also at the basis of the so-called Error Level Analysis (ELA), widely used by practitioners for its simplicity. A further direction is to exploit the model-specific implementations of the JPEG standard, including customized quantization tables and post-processing steps [40]. For example, in [1] model-specific JPEG features have been defined, called JPEG dimples. These artifacts are caused by the specific procedure used when converting real to integer values, e.g., ceil, floor, and rounding operator, and represent a very discriminant clue for images saved in JPEG format.

Exploiting compression artifacts for detecting video manipulation is also possible, but it is much more difficult because of the complexity of the video coding algorithm. Traces of MPEG double compression were first highlighted in the seminal paper by Wang and Farid for detecting frame removal [65]. In fact, the de-synchronization caused by removing a group of frames introduces spikes in the Fourier transform of the motion vectors. A successive work by [62] tried to improve the double compression estimation especially in the more challenging scenario when the strength of the second compression increases and proposed a distinctive footprint, based on the variation of the macroblock prediction types in the reencoded P-frames.

3.3.3 Editing Artifacts

When an image is manipulated, for example, by adding an object, it typically needs several post-processing steps to fit the new context well. These include geometric transformations, like rotation and scaling, contrast adjustment, and blurring, to smooth the object-background boundaries. Therefore, many papers focus on detecting these basic operations as a proxy for possible forgeries. Some methods [42, 56] try to detect traces of resampling, always necessary in the presence of rotation or resizing by exploiting periodic artifacts. Other approaches focus on anomalies on the boundaries of objects when a composition is performed [25] or on blurring-related inconsistencies [3].

A very common manipulation consists in copy-moving image regions to duplicate or hide objects. Of course, the presence of identical regions is a strong hint of forgery, but clones are often modified to disguise traces, and near-identical natural objects also exist, which complicate the forensic analysis. Studies on copy-move detection date back to 2003, with the seminal work of Fridrich et al. [31]. Since then, a large amount of the literature has grown on this topic. Effective and efficient solutions are now available which allow for copy-move detection even in the presence of rotation, resizing, and other geometric distortions [18]. The common pipeline for copy-moves methods is based on three main steps (see Fig. 3.10):

- *feature extraction*: a suitable feature is computed for each pixel of interest, expressing the image behavior in its neighborhood;

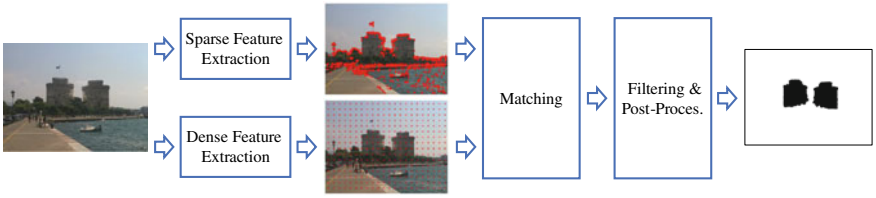

Fig. 3.10 Block diagram relative to copy-move forgery detection methods. The top stream is relative to key-point-based methods, while bottom stream is relative to dense-based methods. Both methodologies have three steps: a feature extraction, a matching search, and a filtering and post-processing step

- *matching*: the best matching of each pixel is computed, based on the associated feature;
- *post-processing*: the offset field, linking pixels with their nearest neighbors, is filtered and processed in order to reduce false alarms.

Some methods [2, 61] extract image key-points and characterize them by means of suitable local descriptors, such as Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Feature (SURF), Local Binary Pattern (LBP), and other variants of these local features. They are very efficient, but work only for additive forgeries, and not on occlusive ones that typically involve smooth regions. This performance gap is shown in the extensive evaluation carried out in [18] and motivates the importance to work on a block-based approach that analyzes the whole image. Of course, in this case the major problem is complexity, since all pixels undergo the three phases of feature extraction, matching, and post-processing. First of all, it is important to use features that are robust to some common forms of distortion in order to deal for example with rotated and/or rescaled duplications. Circular harmonic transforms, such as Zernike moments and polar sine and cosine transforms, are well-suited to provide rotation invariance [22, 59]. As for scale-invariance, research has mostly focused on variations of the Fourier-Mellin transform, based on a log-polar sampling.

Besides feature selection, the literature has devoted much attention to the matching step. In fact, an exhaustive search of the best matching (nearest neighbor) feature is prohibitive due to its huge complexity. A significant speed-up can be obtained by adopting some approximate nearest-neighbor search strategy, like kd-trees or locality-sensitive hashing. Nonetheless, computing the nearest-neighbor field (NNF) is too slow for the large images generated by today's cameras. A much better result can be obtained, however, by exploiting the strong regularity exhibited by the NNFs of natural images, where similar offsets are often associated with neighboring pixels, as done in PatchMatch [5], a fast randomized algorithm which finds dense approximate nearest neighbor matches between image patches. The basic algorithm described above finds only a single nearest-neighbor, and does not deal with scale changes and rotations, hence in [22] it has been proposed to add first-order predictors to the zero-order predictors used in PatchMatch, so as to deal effectively also with linear object deformations. In Fig. 3.11, we show some results of this approach that can effectively

Fig. 3.11 Examples of inpainting manipulated images with binary masks obtained using the dense-based copy-move detection algorithm proposed in [23]

deal both with additive manipulations and occlusive ones, typically carried out using inpainting methods.

Extensions to videos have been also proposed both for detection and localization [9, 24], the main issue being complexity, handled in [24] through a multi-scale processing and parallel implementation of a 3D version of the modified version of PatchMatch [22].

3.4 Learning-Based Methods with Handcrafted Features

These methods are based on machine learning and need large datasets of pristine and manipulated images. An important step is the definition of suitable features that help to discriminate between pristine and manipulated images, then a classifier is trained on a large number of examples of both types. The choice of the features depends on which type of traces one wants to discover. For example, some features have been devised to detect specific artifacts, especially those generated by double JPEG compression [14, 35, 38].

However, more precious are the *universal* features, based on suitable image statistics, which allow detecting many types of manipulations. Major efforts have been devoted to finding good statistical models for natural images in order to select the features that guarantee the highest discriminative power. In order to single out statistical fluctuations caused by manipulation operations, it is important to first remove the semantic image content, to be regarded as noise [7]. The pioneering work of Farid and Lyu [28], back in 2003, proved the potential of statistics-based features extracted from the high-pass bands of the Wavelet domain. These features capture subtle variations in the image micro-textures and prove effective in many application fields beyond image forensics. Other approaches work on residuals in the DCT

domain [36] or in the spatial domain [46, 66]. Particularly effective, again, are the features extracted from the high-pass filtered version of the image and that are on the co-occurrence of selected neighbors [30] (see Fig. 3.6).

As an alternative to the two-class problem, it is also possible to learn only from pristine images and then look for possible anomalies. Since cameras of the same model share proprietary design choices for both hardware and software, they will leave similar marks on the acquired images. Therefore, in [64] it was proposed to extract local descriptors from same-model noise residuals to build a reference statistical model. Then, at test time, the same descriptors are extracted in sliding-window modality from the target noise residual and compared with the reference. Strong deviations from the reference statistics suggest the presence of a manipulation.

3.5 Conclusions

Multimedia forensics has been an active research area for a long time and many approaches have been proposed to detect classic manipulations. PRNU-based methods represent very powerful tools, however, they need a certain amount of data coming from the camera in order to reliably estimate the sensor fingerprint. In addition, it is important to note that the internal pipeline of new cameras is changing, with more sophisticated software and hardware. For example, the introduction of new coding schemes and new shooting modes makes the classic sensor noise estimation less reliable [4] and calls for new ways of detecting the camera traces.

A major appeal of blind methods is that they do not require further data besides those under test. However, methods based on very specific details depend heavily on their statistical model, and mostly fail when the hypotheses do not hold. This happens, for example, when these images are posted on social networks and undergo a global resizing and compression. The final effect is to disrupt some specific clues and impairing sharply the performance of most methods, as shown in [63]. Copy-move detectors, instead, are more reliable, even in the presence of post-processing, but can only detect cloning and some types of inpainting. On the contrary, methods based on noise patterns are quite general, and robust to post-processing, as they often do not depend on explicit statistical models but look for anomalies in the noise residual. Interestingly, many recent deep learning-based methods rely on these basic concepts [63]. For example, some of them include a constrained first layer that performs high-pass filtering of the image, in order to suppress the scene content and allow to work on residuals.

As for machine learning-based methods, they can achieve very good detection results: in the 2013 challenge the accuracy was around 94% [19]. However, performance depends heavily on the alignment between training set and test data. It is very high when training and test sets share the same cameras, same types of manipulation, same processing pipeline, like when a single dataset is split in training and test or cross-validation is used. As soon as unrelated datasets are used, the performance

drops sometimes to random guesses. Lack of robustness limits the applicability of learning-based approaches to very specific scenarios.

Moreover, a skilled attacker, aware of the principles on which forensic tools work, may enact some counter-forensic measure on purpose to evade detectors [6, 33]. Therefore, the integration of multiple tools, all designed to detect the same type of attack but under different approaches, may be expected to improve performance, and especially robustness with respect to both casual and malicious disturbances. In support of this hypothesis, it is worth mentioning that the winners of the First IEEE Image Forensics Challenge resorted to the fusion of multiple tools both for the detection and the localization tasks [19, 20] and similar approaches are routinely used also for deep learning-based solutions. More in general, most of the key concepts and problems encountered in the context of AI-based forensics were already present and investigated in classical multimedia forensics, which therefore represents a necessary starting point for new advances.

Acknowledgements This material is based on research sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under agreement number FA8750-20-2-1004. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and AFRL or the U.S. Government. This work is also supported by the PREMIER project, funded by the Italian Ministry of Education, University, and Research within the PRIN 2017 program and by a Google gift.

References

1. Agarwal S, Farid H (2017) Photo forensics from JPEG dimples. In: IEEE international workshop on information forensics and security (WIFS), pp 1–6
2. Amerini I, Ballan L, Caldelli R, Bimbo AD, Serra G (2011) A SIFT-based forensic method for copy-move attack detection and transformation recovery. IEEE Trans Inform Forensics Secur 6(3):1099–1110
3. Bahrami K, Kot A, Li L, Li H (2015) Blurred image splicing localization by exposing blur type inconsistency. IEEE Trans Inform Forensics Secur 10(5):999–1009
4. Baracchi D, Iuliani M, Nencini A, Piva A (2020) Facing image source attribution on iPhone X. In: International workshop on digital forensics and watermarking (IWDW), pp 196–207
5. Barnes C, Shechtman E, Finkelstein A, Goldman DB (2009) PatchMatch: a randomized correspondence algorithm for structural image editing. ACM Trans Graph 28(3)
6. Barni M, Stamm M, Tondi B (2018) Adversarial multimedia forensics: overview and challenges ahead. In: European signal processing conference (Eusipco), pp 962–966
7. Bayram S, Avcibaş I, Sankur B, Memon N (2006) Image manipulation detection. J Electron Imaging 15(4):1–17
8. Bayram S, Sencar H, Memon N, Avcibas I (2005) Source camera identification based on CFA interpolation. In: IEEE international conference on image processing (ICIP), pp III–69
9. Bestagini P, Milani S, Tagliasacchi M, Tubaro S (2013) Local tampering detection in video sequences. In: IEEE international workshop on multimedia signal processing (MMSP), pp 488–493

10. Bianchi T, Piva A (2012) Image forgery localization via block-grained analysis of JPEG artifacts. IEEE Trans Inform Forensics Secur 7(3):1003–1017
11. Cao H, Kot A (2009) Accurate detection of demosaicing regularity for digital image forensics. IEEE Trans Inform Forensics Secur 4(5):899–910
12. Chakraborty S, Kirchner M (2017) PRNU-based forgery detection with discriminative random fields. In: International symposium on electronic imaging: media watermarking, security, and forensics
13. Chen YL, Hsu CT (2011) Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection. IEEE Trans Inform Forensics Secur 6(2):396–406
14. Chen M, Fridrich J, Goljan M, Lukàš J (2008) Determining image origin and integrity using sensor noise. IEEE Trans Inform Forensics Secur 3(4):74–90
15. Chierchia G, Parrilli S, Poggi G, Sansone C, Verdoliva L (2010) On the influence of denoising in PRNU based forgery detection. In: ACM workshop on multimedia in forensics, security and intelligence, pp 117–122
16. Chierchia G, Poggi G, Sansone C, Verdoliva L (2014) A Bayesian-MRF approach for PRNU-based image forgery detection. IEEE Trans Inform Forensics Secur 9(4):554–567
17. Chierchia G, Cozzolino D, Poggi G, Sansone C, Verdoliva L (2014) Guided filtering for PRNU-based localization of small-size image forgeries. In: IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 6231–6235
18. Christlein V, Riess C, Jordan J, Angelopoulou E (2012) An evaluation of popular copy-move forgery detection approaches. IEEE Transactions on Information Forensics and Security 7(6):1841–1854
19. Cozzolino D, Gragnaniello D, Verdoliva L (2014a) Image forgery detection through residual-based local descriptors and block-matching. In: IEEE international conference on image processing (ICIP), pp 5297–5301
20. Cozzolino D, Gragnaniello D, Verdoliva L (2014b) Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques. In: IEEE international conference on image processing (ICIP), pp 5302–5306
21. Cozzolino D, Marra F, Poggi G, Sansone C, Verdoliva L (2017) PRNU-based forgery localization in a blind scenario. In: International conference on image analysis and processing (ICIAP), pp 569–579
22. Cozzolino D, Poggi G, Verdoliva L (2015) Efficient dense-field copy-move forgery detection. IEEE Trans Inform Forensics Secur 10(11):2284–2297
23. Cozzolino D, Poggi G, Verdoliva L (2015) Splicebuster: a new blind image splicing detector. In: IEEE international workshop on information forensics and security (WIFS), pp 1–6 (2015)
24. D'Amiano L, Cozzolino D, Poggi G, Verdoliva L (2019) A PatchMatch-based dense-field algorithm for video copy-move detection and localization. IEEE Trans Circ Syst Video Technol 29(3):669–682
25. Dong J, Wang W, Tan T, Shi Y (2006) Run-length and edge statistics based approach for image splicing detection. In: International workshop on digital watermarking, pp 177–187
26. Fan Z, de Queiroz R (2003) Identification of bitmap compression history: JPEG detection and quantizer estimation. IEEE Trans Image Process 12(2):230–235
27. Farid H (2009) Exposing digital forgeries from JPEG Ghosts. IEEE Trans Inform Forensics Secur 4(1):154–160
28. Farid H, Lyu S (2003) Higher-order wavelet statistics and their application to digital forensics. In: IEEE workshop on statistical analysis in computer vision, pp 1–8
29. Ferrara P, Bianchi T, De Rosa A, Piva A (2012) Image forgery localization via fine-grained analysis of CFA artifacts. IEEE Trans Inform Forensics Secur 7(5):1566–1577
30. Fridrich J, Kodovsky J (2012) Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security 7:868–882
31. Fridrich J, Soukal D, Lukáš J (2003) Detection of copy-move forgery in digital images. In: Proceedings of the 3rd digital forensic research workshop

32. Fu D, Shi Y, Su W (2007) A generalized Benford's law for JPEG coefficients and its applications in image forensics. In: Proceedings of SPIE, security, steganography, and watermarking of multimedia contents IX, 65051L
33. Gloe T, Kirchner M, Winkler A, Böhme R (2007) Can we trust digital image forensics? In: ACM international conference on multimedia, pp 78–86
34. He K, Sun J, Tang X (2013) Guided image filtering. IEEE Trans Pattern Anal Mach Intell 35(6):1387–1409
35. He J, Lin Z, Wang L, Tang X (2006) Detecting doctored JPEG images via DCT coefficient analysis. In: European conference on computer vision (ECCV), pp 425–435
36. He Z, Lu W, Sun W, Huang J (2012) Digital image splicing detection based on Markov features in DCT and DWT domain. Pattern Recogn 45:4292–4299
37. Healey G, Kondepudy R (1994) Radiometric CCD camera calibration and noise estimation. IEEE Trans Pattern Anal Mach Intell 16(3):267–276
38. Jiang X, He P, Sun T, Xie F, Wang S (2018) Detection of double compression with the same coding parameters based on quality degradation mechanism analysis. IEEE Trans Inform Forensics Secur 13(1):170–185
39. Johnson M, Farid H (2007) Exposing digital forgeries in complex lighting environments. IEEE Trans Inform Forensics Secur 2(3):450–461
40. Kee E, Johnson M, Farid H (2011) Digital image authentication from JPEG headers. IEEE Trans Inform Forensics Secur 6(3):1066–1075
41. Kee E, O'Brien J, Farid H (2013) Exposing photo manipulation with inconsistent shadows. ACM Trans Graph 32(3):28–58
42. Kirchner M (2008) Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue. In: 10th ACM workshop on multimedia and security, pp 11–20
43. Korus P, Huang J (2017) Multi-scale analysis strategies in PRNU-based tampering localization. IEEE Trans Inf Forensics Secur 12(4):809–824
44. Li B, Shi Y, Huang J (2008) Detecting doubly compressed JPEG images by using mode based first digit features. In: IEEE workshop on multimedia signal processing (MMSP), pp 730–735
45. Li W, Yuan Y, Yu N (2009) Passive detection of doctored JPEG image via block artifact grid extraction. Signal Processing 89(9):1821–1829
46. Li H, Luo W, Qiu X, Huang J (2018) Identification of various image operations using residual-based features. IEEE Trans Circ Syst Video Technol 28(1):31–45
47. Lin Z, He J, Tang X, Tang CK (2009) Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis. Pattern Recognition 42(11):2492–2501
48. Lukáš J, Fridrich J (2003) Estimation of primary quantization matrix in double compressed JPEG images. In: Proceedings of the 3rd digital forensic research workshop
49. Lukàš J, Fridrich J, Goljan M (2006) Detecting digital image forgeries using sensor pattern noise. In: Proceedings of SPIE, pp 362–372
50. Lyu S, Pan X, Zhang X (2014) Exposing region splicing forgeries with blind local noise estimation. Int J Comput Vis 110(2):202–221
51. Mandelli S, Bestagini P, Tubaro S, Cozzolino D, Verdoliva L (2017) Blind detection and localization of video temporal splicing exploiting sensor-based footprints. In: European signal processing conference (EUSIPCO), pp 1362–1366
52. Milani S, Fontani M, Bestagini P, Barni M, Piva A, Tagliasacchi M, Tubaro S (2012) An overview on video forensics. APSIPA Trans Signal Inform Process 1
53. Mondaini N, Caldelli R, Piva A, Barni M, Cappellini V (2017) Detection of malevolent changes in digital video for forensic applications. In: SPIE Security, steganography, and watermarking of multimedia contents IX, pp 300–311
54. Mullan P, Cozzolino D, Verdoliva L, Riess C (2017) Residual-based forensic comparison of video sequences. In: IEEE international conference on image processing (ICIP), pp 1507–1511
55. Pasquini C, Boato G, Pèrez-Gonzàlez F (2017) Statistical detection of JPEG traces in digital images in uncompressed formats. IEEE Trans Inform Forensics Secur 12(12):2890–2905
56. Popescu A, Farid H (2004) Statistical tools for digital forensics. In: International workshop on information hiding, pp 128–147

57. Popescu A, Farid H (2005) Exposing digital forgeries in color filter array interpolated images. IEEE Trans Signal Process 53(10):3948–3959
58. Popescu A, Farid H (2005) Exposing digital forgeries by detecting traces of resampling. IEEE Trans Signal Process 53(2):758–767
59. Ryu SJ, Kirchner M, Lee MJ, Lee HK (2013) Rotation invariant localization of duplicated image regions based on Zernike moments. IEEE Trans Inform Forensics Secur 8(8):1355–1370
60. Scherhag U, Debiasi L, Rathgeb C, Busch C, Uhl A (2019) Detection of face morphing attacks based on PRNU analysis. IEEE Trans Biometr Behav Identity Sci 1(4):302–317
61. Silva E, Carvalho T, Ferreira A, Rocha A (2015) Going deeper into copy-move forgery detection: Exploring image telltales via multi-scale analysis and voting processes. J Visual Commun Image Represent 29:16–32
62. Vázquez-Padín D, Fontani M, Bianchi T, Comesana P, Piva A, Barni M (2012) Detection of video double encoding with GOP size estimation. In: IEEE international workshop on information forensics and security (WIFS), pp 151–156
63. Verdoliva L (2020) Media forensics and deepfakes: an overview. IEEE J Sel Top Signal Process 14(5):910–932
64. Verdoliva L, Cozzolino D, Poggi G (2014) A feature-based approach for image tampering detection and localization. In: IEEE international workshop on information forensics and security (WIFS), pp 149–154
65. Wang W, Farid H (2006) Exposing digital forgeries in video by detecting double MPEG compression. In: Workshop on multimedia and security, pp 37–47
66. Zhao X, Wang S, Li S, Li J, Yuan Q (2013) Image splicing detection based on noncausal Markov model. In: IEEE international conference on image processing (ICIP), pp 4462–4466