Determining Image Origin and Integrity Using Sensor Noise

Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukáš

Abstract—**In this paper, we provide a unified framework for identifying the source digital camera from its images and for revealing digitally altered images using photo-response non-uniformity noise (PRNU), which is a unique stochastic fingerprint of imaging sensors. The PRNU is obtained using a Maximum Likelihood estimator derived from a simplified model of the sensor output. Both digital forensics tasks are then achieved by detecting the presence of sensor PRNU in specific regions of the image under investigation. The detection is formulated as a hypothesis testing problem. The statistical distribution of the optimal test statistics is obtained using a predictor of the test statistics on small image blocks. The predictor enables more accurate and meaningful estimation of probabilities of false rejection of a correct camera and missed detection of a tampered region. We also include a benchmark implementation of this framework and detailed experimental validation. The robustness of the proposed forensic methods is tested on common image processing, such as JPEG compression, gamma correction, resizing, and denoising.**

Index Terms—**Photo-response non-uniformity, imaging sensor, digital forensic, digital forgery, camera identification, authentication, integrity verification.**

I. INTRODUCTION

While digital representation of reality brings unquestionable advantages, digital images can be easily modified using powerful image editing software, which creates a serious problem of how much their content can be trusted when presented as silent witness in a courtroom. Another problem brought by digitization is verification of origin. Is it possible to prove that a certain image was taken by a specific camera? Reliable methods for establishing the integrity and origin of digital images are urgently needed in situations when a digital image or video forms a key piece of evidence, such as in child pornography and movie piracy cases, insurance claims, and cases involving scientific fraud [1,2].

The tasks of digital forensics can be broadly divided into the following six categories:

1. Source Classification whose objective is to classify images according to their origin, such as scans vs. digital camera images, or Canon vs. Kodak, etc.

2. Device Identification aims at proving that a given image was obtained by a specific device (prove that a certain camera took a given image or video).

3. Device Linking groups objects according to their common source. For example, given a set of images, we would like to find out which images were obtained using the same camera.

4. Processing History Recovery whose objective is to recover the processing chain applied to the image. Here, we are interested in non-malicious processing, e.g., lossy compression, filtering, resizing, contrast/brightness adjustment, etc.

5. Integrity Verification or forgery detection is a procedure aimed at discovering malicious processing, examples of which are object removal or adding.

6. Anomaly Investigation deals with explaining anomalies found in images that may be a consequence of digital processing or other phenomena specific to digital cameras.

In this paper, we focus on Device Identification and Integrity Verification. The problem of image origin (Tasks 1–3) has been approached in the past by detecting camera processing artifacts [3–5] and using classification of image-derived features [6]. While these methods are useful for Source Classification, they cannot be used for Device Identification as this problem calls for an equivalent of a unique "biometrics for cameras." The first sensor biometrics were defective pixels (hot and dead pixels) [7,8]. Recently [9], the pixel photo-response non-uniformity (PRNU) was proposed as sensor biometrics and used for reliable Device Identification even from processed images. Approaches combining sensor noise with machine-learning classification were described in [9–14].

The second forensic task investigated in this paper, is Integrity Verification commonly known as forgery detection. Recently, numerous methods for detecting digital forgeries were proposed [15–27,38]. Most methods are based on detecting local inconsistencies, such as in resampling artifacts [19], color filter array (CFA) interpolation artifacts [20], illumination [21], or optical defects [23]. Some approaches detect identical regions in a copy-move forgery [25,26]. A different class of methods use classification of image features [16–18,24,38]. Each of these methods only works when specific assumptions are satisfied and will fail if the assumptions are not met. Obviously, digital forgery detection is a complex problem with no universally applicable solution. What is needed is a large set of tools based on different principles that can all be applied to the image at hand. Indeed, while it may be easy to create a forgery undetectable using each individual detection method, it may be hard to make sure that none of them applies. The decision about the content authenticity is then reached by interpreting the results obtained from different approaches. This accumulative evidence may provide a convincing enough argument that each individual method cannot.

In this paper, we describe a unified framework for both Device Identification and Integrity Verification using pixel PRNU as proposed in [9]. Both tasks start with estimation of the PRNU, which is achieved using a Maximum Likelihood estimator derived from a simplified model of sensor output. This improved estimator makes better use of available data in the sense that the number of images needed to estimate the PRNU can be significantly smaller than what was reported in [9]. By establishing the presence of PRNU in an image or in an image block, we can determine the image origin or verify the block integrity. A good analogy is to think of the PRNU signal as a unique authentication watermark involuntarily inserted by the imaging sensor. Establishing its presence in an image amounts to identifying the sensor, while checking its integrity reveals tampered (forged) regions.

The paper is organized as follows. In Section II, we describe a simplified sensor output model that will be used to derive the PRNU estimator and detector. In Section III, we derive a ML estimator for the PRNU and point out the need to preprocess the estimated signal to remove certain systematic patterns that might increase false alarms in Device Identification and missed detections in Integrity Verification. The task of detecting the PRNU is formulated as Neyman-Pearson hypothesis testing problem in Section IV. The correlation predictor used to obtain the distribution of the test statistics is detailed in Section V. Benchmark implementation of the proposed framework for both forensic tasks and the performance evaluation appear in Section VI. The experimental results are accompanied with discussion of limitations and ideas for future research. The paper is summarized in Section VII.

Everywhere in this paper, boldface font will denote vectors of length specified in the text, e.g., **X** and **Y** are vectors of length n and **X**[i] denotes the i-th component of **X**. Unless mentioned otherwise, all operations among vectors, such as product, ratio, raising to a power, etc., are *elementwise*. The dot product of vectors is denoted as $\mathbf{X} \odot \mathbf{Y} = \sum_{i=1}^{n} \mathbf{X}[i]\mathbf{Y}[i]$ and $\|\mathbf{X}\| = \sqrt{\mathbf{X} \odot \mathbf{X}}$ is the norm of **X**. Denoting the mean values with a bar, the normalized correlation is

$$corr(\mathbf{X},\mathbf{Y}) = \frac{(\mathbf{X}-\bar{\mathbf{X}}) \odot (\mathbf{Y}-\bar{\mathbf{Y}})}{\|\mathbf{X}-\bar{\mathbf{X}}\| \cdot \|\mathbf{Y}-\bar{\mathbf{Y}}\|}.$$

II. IMAGING SENSOR OUTPUT MODEL

The signal processing chain in digital cameras is quite complex and varies greatly with different camera types and manufacturers. It typically includes signal quantization, white balance, demosaicking (color interpolation), color correction, gamma correction, filtering, and, optionally, JPEG compression. Because details about the processing are not always easily available (they are hard-wired or proprietary), we decided to use a simplified model [28] that captures various elements of typical in-camera processing that are most relevant to our task. This enables us to develop low-complexity algorithms applicable to a wider spectrum of cameras. A more accurate model tailored to a specific camera would likely produce more reliable camera identification and forgery detection results at the cost of increased complexity.

We denote by **I**[i] the signal in a selected color channel at pixel i, $i = 1, \ldots, n$, generated by the sensor before demosaicking is applied. Let **Y**[i] be the incident light intensity at pixel i. Here, we assume that the pixels are indexed, for example, in a row-wise manner. Dropping the pixel indices for better readability, we use the following vector form of the sensor output model

$$\mathbf{I} = g^{\gamma} \cdot \left[(\mathbf{1}+\mathbf{K})\mathbf{Y} + \mathbf{\Lambda}\right]^{\gamma} + \mathbf{\Theta}_q . \quad (1)$$

We remind that all operations in (1) are element-wise. In (1), g is the color channel gain and γ is the gamma correction factor (typically, $\gamma \approx 0.45$). The gain factor g adjusts the pixel intensity level according to the sensitivity of the pixel in the red, green, and blue spectral bands to obtain the correct white balance. The multiplicative factor **K** is a zero-mean noise-like signal responsible for PRNU (the sensor fingerprint). Finally, **Λ** is a combination of the other noise sources including the dark current, shot noise, and read-out noise [29,30], and $\mathbf{\Theta}_q$ is the quantization noise. The model (1) was selected over the more common additive model typically used in denoising because we intend to estimate the PRNU factor **K**.

Because in natural images the dominant term in the square bracket in (1) is the scene light intensity **Y**, we factor it out and only keep the first two terms in the Taylor expansion of $(1 + x)^{\gamma} = 1 + \gamma x + O(x^2)$ at $x = 0$:

$$\begin{aligned}\mathbf{I} &= (g\mathbf{Y})^{\gamma} \cdot \left[\mathbf{1}+\mathbf{K}+\mathbf{\Lambda}/\mathbf{Y}\right]^{\gamma} + \mathbf{\Theta}_q \doteq \\ &(g\mathbf{Y})^{\gamma} \cdot (\mathbf{1}+\gamma\mathbf{K}+\gamma\mathbf{\Lambda}/\mathbf{Y}) + \mathbf{\Theta}_q = \mathbf{I}^{(0)} + \mathbf{I}^{(0)}\mathbf{K} + \mathbf{\Theta}.\end{aligned} \quad (2)$$

To simplify the notation and avoid introducing too many symbols, in the last expression of (2) we absorbed the gamma correction factor into the PRNU factor **K** and wrote **K** instead of γ**K**. The signal $\mathbf{I}^{(0)} = (g\mathbf{Y})^{\gamma}$ is the sensor output in the absence of noise, $\mathbf{I}^{(0)}\mathbf{K}$ is the PRNU term, and $\mathbf{\Theta} = \gamma \mathbf{I}^{(0)}\mathbf{\Lambda}/\mathbf{Y} + \mathbf{\Theta}_q$ is a complex of independent random noise components.

III. PRNU ESTIMATION AND PREPROCESSING

In this section, we describe the first step in the proposed framework for Camera Identification and Integrity Verification, which is the estimation of the PRNU factor **K**.

A. PRNU estimation

Assuming that either the camera that took the image is available to the analyst or at least some other (non-tampered) images taken by the camera are available, we estimate **K** from a set of N images taken by the camera. To improve the SNR between the signal of interest $\mathbf{I}^{(0)}\mathbf{K}$ and observed data **I**, we perform host signal rejection and suppress the noiseless image $\mathbf{I}^{(0)}$ by subtracting from both sides of (2) the denoised version of **I**, $\hat{\mathbf{I}}^{(0)} = F(\mathbf{I})$, obtained using a denoising filter[1] F:

$$\mathbf{W} = \mathbf{I} - \hat{\mathbf{I}}^{(0)} = \mathbf{IK} + \mathbf{I}^{(0)} - \hat{\mathbf{I}}^{(0)} + (\mathbf{I}^{(0)} - \mathbf{I})\mathbf{K} + \mathbf{\Theta}$$
$$= \mathbf{IK} + \mathbf{\Xi}\,. \qquad (3)$$

The noise $\mathbf{\Xi}$ is a sum of $\mathbf{\Theta}$ and two additional terms introduced by the denoising filter. Since the image content is significantly suppressed in the noise residual **W**, we can better estimate (and detect) the PRNU from **W** than from **I**. On the other hand, the denoising filter further shapes the signal we are trying to estimate or detect and it makes the noise $\mathbf{\Xi}$ non-stationary (e.g., var($\mathbf{\Xi}$) is larger in textured areas).

We now derive the estimator of the PRNU factor **K** from N images $\mathbf{I}_1$, …, $\mathbf{I}_N$ obtained by the camera. We assume here that the images are relatively smooth. If the camera is available, we can for example take images of blue sky. For such images, we can accept a simpler model of the noise term and model the sequence $\mathbf{\Xi}_1[i]$, …, $\mathbf{\Xi}_N[i]$ for each pixel i as white Gaussian noise (WGN) with variance σ^2. We would like to emphasize here that the energy of the PRNU signal **IK** is small compared to the noise term $\mathbf{\Theta}$ and thus it is reasonable to assume that $\mathbf{\Xi}$ is independent of **IK**. From (3), we have for each $k = 1, \ldots, N$

$$\frac{\mathbf{W}_k}{\mathbf{I}_k} = \mathbf{K} + \frac{\mathbf{\Xi}_k}{\mathbf{I}_k}, \quad \mathbf{W}_k = \mathbf{I}_k - \hat{\mathbf{I}}_k^{(0)},\ \hat{\mathbf{I}}_k^{(0)} = F(\mathbf{I}_k)\,. \qquad (4)$$

The log-likelihood of observing $\mathbf{W}_k / \mathbf{I}_k$ given **K** is

$$L(\mathbf{K}) = -\frac{N}{2}\sum_{k=1}^{N}\log(2\pi\sigma^2/(\mathbf{I}_k)^2) - \sum_{k=1}^{N}\frac{(\mathbf{W}_k/\mathbf{I}_k - \mathbf{K})^2}{2\sigma^2/(\mathbf{I}_k)^2}\,. \qquad (5)$$

The ML estimate $\hat{\mathbf{K}}$ for the PRNU factor is obtained by taking partial derivatives of (5) with respect to individual elements of **K** and solving for **K**

$$\frac{\partial L(\mathbf{K})}{\partial \mathbf{K}} = \sum_{k=1}^{N}\frac{\mathbf{W}_k/\mathbf{I}_k - \mathbf{K}}{\sigma^2/(\mathbf{I}_k)^2} = 0 \;\Rightarrow\; \hat{\mathbf{K}} = \frac{\sum_{k=1}^{N}\mathbf{W}_k\mathbf{I}_k}{\sum_{k=1}^{N}(\mathbf{I}_k)^2}\,. \qquad (6)$$

By computing the second partial derivative, we obtain the Cramer-Rao Lower Bound (CRLB) on the variance of $\hat{\mathbf{K}}$

$$\frac{\partial^2 L(\mathbf{K})}{\partial \mathbf{K}^2} = -\frac{\sum_{k=1}^{N}(\mathbf{I}_k)^2}{\sigma^2} \Rightarrow var(\hat{\mathbf{K}}) \geq \frac{1}{-E\left[\dfrac{\partial^2 L(\mathbf{K})}{\partial \mathbf{K}^2}\right]} = \frac{\sigma^2}{\sum_{k=1}^{N}(\mathbf{I}_k)^2}. \qquad (7)$$

Because our model (3) is linear, the ML estimator is minimum variance unbiased (MVU) and the CRLB determines its variance var($\hat{\mathbf{K}}$) ~ $1/N$. Equation (7) also informs us what images are best for estimation of **K**. The luminance $\mathbf{I}_k$ should be as high as possible but not saturated because saturated pixels ($\mathbf{I}_k[i] \approx 255$ for an 8-bit grayscale image) carry no information about the PRNU factor. Also notice that var($\hat{\mathbf{K}}$) ~ σ^2, the variance of the sum of image acquisition noise sources $\mathbf{\Theta}$ and the terms $\mathbf{I}_0 - \hat{\mathbf{I}}_0$ introduced by denoising. Therefore, better estimates are obtained using smooth test images. The best images are images of bright (but not saturated) uniformly white scenes. In practice, we recommend taking out-of-focus images of cloudy sky.

We use estimator (6) also when $\mathbf{I}_k$ are images of natural scenes. The estimator still performs well, but more images are needed to obtain the same quality PRNU estimate (about twice as many). In terms of the derived CRLB (7), this is because the average light intensity in natural images is lower and the noise residual $\mathbf{\Xi}$ has larger variance due to edges present in natural images.

B. PRNU preprocessing

The estimated factor $\hat{\mathbf{K}}$ contains all components that are systematically present in every image, most importantly some weak artifacts of color interpolation, on-sensor signal transfer [32], and sensor design [36]. Such artifacts are not unique to the sensor and are shared among cameras of the same brand or cameras sharing the same imaging sensor design[2]. The PRNU factors estimated from two different cameras may thus be slightly correlated, which would increase the false identification rate and decrease the reliability of camera identification. Therefore, we suppress these artifacts from the estimated factor $\hat{\mathbf{K}}$ before using it in our proposed framework. We evaluate how successfully the unwanted artifacts were removed using the correlation between PRNU factors, aiming for the correlation to be as close to zero as possible.

We have experimentally identified three primary processes responsible for the artifacts.

a) Color interpolation. Cameras equipped with a CFA can only capture one color at each pixel. Due to varying sensitivity of silicone to light of different wavelengths, the sensor output is first adjusted for gain before color interpolation. The remaining colors are interpolated from neighboring pixels. Thus, interpolated colors are obtained through a different mechanism than colors that were truly registered at the pixel. As a result, slightly offset gains may introduce small but measurable biases in the interpolated colors. Since all CFAs form a periodic structure, these biases will introduce a periodic pattern in column and row averages of the estimated factor $\hat{\mathbf{K}}$.

b) Row-wise and column-wise operation of sensors and processing circuits. The row-wise and column-wise character of operations of digital imaging sensors and/or image processing circuits [32] also introduces a bias into

[1] We use the filter described in [9,31].

[2] This observation was already made in [9] but was not further investigated.

each column and row.

c) Other artifacts. Besides the row and column artifacts, there may exist other artifacts in the estimated PRNU factor caused by the sensor design, such as the polygonal patterns reported in [36]. Since these artifacts manifest themselves as structures in the Fourier transform of $\hat{\mathbf{K}}$, they can be removed by high-pass filtering in the Fourier domain.

We note that strong JPEG compression also creates blockiness artifacts that can propagate into the estimated PRNU factor. This is especially true for video [33] where the PRNU is estimated from thousands of frames; it is less of an issue for still images. Such artifacts are best removed using frequency selective filters.

Since the first two artifacts are specific to the sensor design, the CFA, and color interpolation, they may be potentially useful for identification of the camera brand or model.

We propose the following procedure to suppress the unwanted artifacts in the estimated PRNU factor $\hat{\mathbf{K}}$. To remove artifacts a) and b), we subtract the column average from each pixel in the column (for each color channel separately) and then subtract the row average from every pixel in the row. The processed PRNU factor will thus have zero mean in every row and column. We denote this operation as $ZM(\hat{\mathbf{K}})$ and call the difference $LP(\hat{\mathbf{K}}) = \hat{\mathbf{K}} - ZM(\hat{\mathbf{K}})$ the *linear pattern*, which is a useful forensic entity by itself. The linear pattern is weak compared to $\hat{\mathbf{K}}$ with SNR below –10dB for compact or single lens reflex (SLR) cameras. It can be stronger for cheap cameras, such as those built in cell-phones. As an example, we show in Fig. 1 a magnified portion of the linear pattern for the Canon S40 camera. As this camera has the Bayer CFA with period 2 along the rows and columns, the linear pattern exhibits the same periodicity.

The final step in preprocessing the estimated PRNU factor is applied only when there exist artifacts in the form of visually identifiable patterns in $ZM(\hat{\mathbf{K}})$ [36]. We transform $ZM(\hat{\mathbf{K}})$ into the Fourier domain, filter it using the Wiener filter, and only keep the noise component

$$WF\left(ZM(\hat{\mathbf{K}})\right) = \mathcal{F}^{-1}\left\{\mathcal{F}\left(ZM(\hat{\mathbf{K}})\right) - W\left(\mathcal{F}\left(ZM(\hat{\mathbf{K}})\right)\right)\right\},$$

where W is the 3×3 Wiener filter with variance obtained as the sample variance of the magnitude of the Fourier transform $\left|\mathcal{F}\left(ZM(\hat{\mathbf{K}})\right)\right|$. The resulting PRNU has a flatter frequency spectrum than $ZM(\hat{\mathbf{K}})$ (see Fig. 2).

To demonstrate how effective the preprocessing is in removing the artifacts, we show in Table I the correlations between differently processed PRNU factors estimated from Canon G2 and Canon S40. We note that both cameras come from the same manufacturer and share the same CCD sensor. The PRNU estimate was generated from 20 high quality JPEGs (for G2) and 20 raw images (for S40).

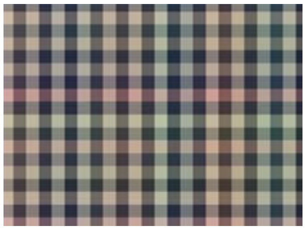


Fig. 1: Detail of the linear pattern for Canon S40.

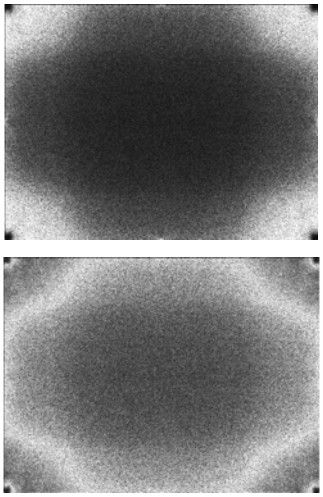


Fig. 2: Fourier transform of the estimated PRNU before and after Wiener filtering.

TABLE I. CORRELATION BETWEEN $\hat{\mathbf{K}}_{\mathrm{G2}}$ FOR CANON G2 AND $\hat{\mathbf{K}}_{\mathrm{S40}}$ FOR CANON S40 BEFORE AND AFTER PRE-PROCESSING.

Correlation	Red	Green	Blue
$\hat{\mathbf{K}}_{\mathrm{G2}}$ vs. $\hat{\mathbf{K}}_{\mathrm{S40}}$	0.0251	0.0134	0.0197
$ZM(\hat{\mathbf{K}}_{\mathrm{G2}})$ vs. $ZM(\hat{\mathbf{K}}_{\mathrm{S40}})$	0.0122	0.0113	0.0077
$WF(ZM(\hat{\mathbf{K}}_{\mathrm{G2}}))$ vs. $WF(ZM(\hat{\mathbf{K}}_{\mathrm{S40}}))$	0.0013	0.0007	0.0005

TABLE II. CORRELATION BETWEEN $\hat{\mathbf{K}}_{\mathrm{LG}}$ FOR LG VX8100 AND $\hat{\mathbf{K}}_{\mathrm{SA}}$ SAMSUNG A900 BEFORE AND AFTER PRE-PROCESSING.

Correlation	Red	Green	Blue
$\hat{\mathbf{K}}_{\mathrm{LG}}$ vs. $\hat{\mathbf{K}}_{\mathrm{SA}}$	0.0164	–0.0058	0.0344
$ZM(\hat{\mathbf{K}}_{\mathrm{LG}})$ vs. $ZM(\hat{\mathbf{K}}_{\mathrm{SA}})$	–0.0011	0.0014	–0.0020
$WF(ZM(\hat{\mathbf{K}}_{\mathrm{LG}}))$ vs. $WF(ZM(\hat{\mathbf{K}}_{\mathrm{SA}}))$	0.0002	0.0010	–0.0012

In Table II, we show another example for two 1.3 megapixel cell phone cameras LG VX8100 and Samsung A900. Only 10 images of sky and gray wall were used for $\hat{\mathbf{K}}_{\mathrm{LG}}$ and 125 similar images for $\hat{\mathbf{K}}_{\mathrm{SA}}$. The zero-meaning significantly reduced the correlation between the estimated PRNU factors. Fourier filtering further decreased the correlation.

To avoid introducing too many symbols, in the remainder of this paper, the symbol $\hat{\mathbf{K}}$ will denote the PRNU factor estimated using (6) *and* subsequently processed as explained in Section III.B to suppress any non-PRNU components.

IV. PRNU DETECTION

As described in the introduction, we approach both Camera Identification and Integrity Verification by establishing the presence of sensor PRNU in the entire image or its region, respectively. In this section, we formulate the detection task as a hypothesis testing problem and explain the detection methodology.

The model (3) with the assumption that $\boldsymbol{\Xi}$ is WGN is an approximately valid representation of reality for images with smooth content. It was already explained in Section III.A that while such a model was reasonable for PRNU estimation (we can make sure test images have the required properties) for most real-life images a more complex model is needed. First, the noise $\boldsymbol{\Xi}$ becomes colored as it is strongly influenced by texture. Second, the PRNU term is modulated by an attenuation factor because we may be subtracting a part of PRNU with the denoised image $\hat{\mathbf{I}}_0$ in (3). Thus, the model we accept for PRNU detection is

$$\mathbf{W} = \mathbf{TI}\hat{\mathbf{K}} + \boldsymbol{\Xi}, \qquad (8)$$

where $\mathbf{T}[i]$ is a pixel-wise multiplicative attenuation factor and $\boldsymbol{\Xi}[i]$ is a sequence of independent Gaussian variables with unequal variances $\boldsymbol{\sigma}_{\boldsymbol{\Xi}}^2[i]$ (colored Gaussian noise).

We now formulate the problem of detection of PRNU in the noise residual $\mathbf{W} = \mathbf{I} - \hat{\mathbf{I}}_0$ of a given image region as a binary hypothesis testing

$$\begin{aligned} &\mathrm{H}_0\text{: } \mathbf{W} = \boldsymbol{\Xi}, \\ &\mathrm{H}_1\text{: } \mathbf{W} = \mathbf{TX} + \boldsymbol{\Xi}, \end{aligned} \qquad (9)$$

where $\mathbf{X} = \mathbf{I}\hat{\mathbf{K}}$ is the non-attenuated PRNU term. The noise-only hypothesis H$_0$ should have been written as $\mathbf{W} = \mathbf{TI}\hat{\mathbf{K}}' + \boldsymbol{\Xi}$, where $\hat{\mathbf{K}}'$ is a PRNU factor from some other camera. However, because the combined noise term $\boldsymbol{\Xi}$ dominates the contribution from the PRNU, we consider the PRNU term as a weak signal and include it in the noise term.

In the next two sections, we derive corresponding optimal test statistics separately for Camera Identification and for Integrity Verification.

A. Camera identification

In order to estimate the shaping factor $\mathbf{T}$ and the variance $\boldsymbol{\sigma}_{\boldsymbol{\Xi}}^2$, we can either accept a parametric model and estimate the parameters or estimate $\mathbf{T}$ and $\boldsymbol{\sigma}_{\boldsymbol{\Xi}}^2$ locally from the image. However, it is not an easy task to find a good model because both quantities depend on a complex interplay between the denoising filter, local texture, and image content. Similarly, it is not possible to accurately estimate these two non-stationary quantities at every pixel due to insufficient data. Thus, we opted for the following simplified approach.

We start by dividing the image into M disjoint blocks and assume that, within each block $b \in \{1, \ldots, M\}$, $\mathbf{T}[i]$ and $\boldsymbol{\sigma}_{\boldsymbol{\Xi}}^2[i]$ are constant equal to T_b and σ_b^2, respectively[3]. Both quantities will be replaced with their estimated values $\hat{T}_b$, $\hat{\sigma}_b^2$, obtained using a predictor (see Section V). Allowing these estimates to be accurate up to an unknown multiplicative factor *common to all blocks* (this factor may be due to processing uniformly applied to the whole image, such as lossy compression or filtering), we arrive at the following hypothesis testing problem

$$\begin{aligned} &\mathrm{H}_0\text{: } \mathbf{W} = c\boldsymbol{\Xi}, \\ &\mathrm{H}_1\text{: } \mathbf{W} = a\mathbf{TX} + c\boldsymbol{\Xi}, \end{aligned} \qquad (10)$$

where now $\boldsymbol{\Xi}[i]$, $i \in B_b$, is WGN with zero mean and known variance $\hat{\sigma}_b^2$ and $\mathbf{T}[i]$, $i \in B_b$, is a known constant $\hat{T}_b$. Both a and c are unknown multiplicative factors that are the same for all blocks. The optimal detector for (10) is the normalized Generalized Matched Filter (see Chapter 4.4 in [34])

$$\rho = \frac{\sum_{b=1}^{M} \frac{\hat{T}_b}{\hat{\sigma}_b^2} (\mathbf{X}_b \odot \mathbf{W}_b)}{\sqrt{\sum_{b=1}^{M} \frac{\hat{T}_b^2}{\hat{\sigma}_b^2} \|\mathbf{X}_b\|^2} \sqrt{\sum_{b=1}^{M} \frac{1}{\hat{\sigma}_b^2} \|\mathbf{W}_b\|^2}} = \sum_{b=1}^{M} \beta_b \rho_b\,, \qquad (11)$$

where

$$\beta_b = \frac{\frac{\hat{T}_b}{\hat{\sigma}_b^2} \|\mathbf{X}_b\| \cdot \|\mathbf{W}_b\|}{\sqrt{\sum_{i=1}^{M} \frac{\hat{T}_i^2}{\hat{\sigma}_i^2} \|\mathbf{X}_i\|^2} \sqrt{\sum_{i=1}^{M} \frac{1}{\hat{\sigma}_i^2} \|\mathbf{W}_i\|^2}}$$

and $\rho_b = \mathbf{X}_b \odot \mathbf{W}_b / \|\mathbf{X}_b\| \|\mathbf{W}_b\| = corr(\mathbf{X}_b, \mathbf{W}_b)$ is the normalized correlation between the PRNU term $\mathbf{X}_b = \mathbf{I}_b \hat{\mathbf{K}}_b$ and the noise residual $\mathbf{W}_b$ ($\hat{\mathbf{K}}_b$ is the PRNU factor in the b-th block estimated using (6)).

We now explain how the shaping factor T_b and variance σ_b^2 are estimated. Under H$_1$, $\mathbf{W}$ comes from the tested camera and we have $\rho_b = corr(\mathbf{X}_b, T_b \mathbf{X}_b + ca^{-1}\boldsymbol{\Xi}_b)$ or

$$\rho_b = \frac{T_b \|\mathbf{X}_b\|^2 + ca^{-1}\mathbf{X}_b \odot \boldsymbol{\Xi}_b}{\|\mathbf{X}_b\| \sqrt{T_b^2 \|\mathbf{X}_b\|^2 + 2T_b ca^{-1}\mathbf{X}_b \odot \boldsymbol{\Xi}_b + \|ca^{-1}\boldsymbol{\Xi}_b\|^2}}. \qquad (12)$$

Because $\boldsymbol{\Xi}_b$ is zero mean and independent of $\mathbf{X}_b$, the mixed term $\mathbf{X}_b \odot \boldsymbol{\Xi}_b$ is small compared to the other terms in the denominator of (12). Thus,

$$\rho_b \approx \frac{T_b \|\mathbf{X}_b\|}{\sqrt{T_b^2 \|\mathbf{X}_b\|^2 + \|ca^{-1}\boldsymbol{\Xi}_b\|^2}} = \frac{1}{\sqrt{1 + \frac{\|ca^{-1}\boldsymbol{\Xi}_b\|^2}{T_b^2 \|\mathbf{X}_b\|^2}}}, \qquad (13)$$

where $\|\boldsymbol{\Xi}_b\|^2 = \sigma_b^2 |B_b|$ is the energy of the noise in the b-th block and $|B_b|$ is the cardinality of B_b. From (13), we obtain the SNR between the signal of interest and noise

[3] The blocks will be described by their index sets $B_b \subset \{1, \ldots, n\}$. Signals constrained to the b-th block will be denoted with subscript b, e.g., $\mathbf{X}_b$, $\mathbf{W}_b$, etc.

$$\frac{c^2 a^{-2} \| \boldsymbol{\Xi}_b \|^2}{T_b^2 \| \mathbf{X}_b \|^2} = \frac{1}{\rho_b^2} - 1 . \tag{14}$$

Because from (10) $\| \mathbf{W}_b \|^2 = a^2 T_b^2 \| \mathbf{X}_b \|^2 + c^2 \| \boldsymbol{\Xi}_b \|^2$, using (14) we obtain $\hat{T}_b$ and $\hat{\sigma}_b^2$ as functions of the known signals $\mathbf{X}_b$, $\mathbf{W}_b$, and ρ_b :

$$\hat{\sigma}_b^2 = \frac{1-\rho_b^2}{c^2 |B_b|} \| \mathbf{W}_b \|^2 , \ \hat{T}_b = \frac{\rho_b}{a \| \mathbf{X}_b \|} \| \mathbf{W}_b \| . \tag{15}$$

In the expressions for $\hat{\sigma}_b^2$ and $\hat{T}_b$, we can skip the multiplicative factors $(c^2|B_b|)^{-1}$ and a^{-1}, respectively, because they are the same for all blocks (if all blocks are of the same size) and thus do not influence the value of ρ in (11).

The estimates (15), however, depend on ρ_b, which we only know under hypothesis H_1 but not under H_0. We address this problem by constructing a *predictor* of the normalized correlation ρ_b on small blocks under H_1 based on our knowledge of $\hat{\mathbf{K}}$ and a few block-derived features that most influence ρ_b (see Section V for details). We use the *predicted values* $\hat{\rho}_b$ in (15) to obtain $\hat{T}_b$ and $\hat{\sigma}_b^2$. The correlation value can thus be expressed as

$$\rho_b = \hat{\rho}_b + \nu_b , \tag{16}$$

where $\hat{\rho}_b$ is the predicted value and ν_b is a random variable representing the modeling error. The data from which the predictor was constructed can also be used to experimentally determine the pdf of the prediction error ν_b, which is needed to obtain the pdf of the test statistics (11) under hypothesis H_1.

To decide whether or not a given image **I** was taken with a specific camera whose PRNU **K** has been estimated in Section III, we use Neymann-Pearson hypothesis approach and set the decision threshold to obtain a required false alarm rate (FAR). False acceptance occurs when hypothesis H_0 is true but we decide H_1, while false rejection occurs when we accept H_0 when H_1 is true. To estimate the pdf of the test statistics (11) under H_0, $p(x|H_0)$, we evaluate (11) for a large amount of images coming from other cameras against the PRNU **K** and model $p(x|H_0)$ using density estimation techniques [35].

The false rejection rate (FRR) depends on the pdf $p(x|H_1)$ of the test statistics for images coming from the same camera. However, as shown in Section VI, it is not possible to find a simple statistical model for $p(x|H_1)$ because ρ strongly depends on image content, which makes the distribution too dependent on the available database of images. Moreover, trying to fit one pdf for all images would lead to overly conservative error estimates for "good" images with high ρ and overly optimistic error estimates for highly textured images that typically have a much lower value of ρ. We should be evaluating the FRR against images of approximately the same content or approximately the same value of ρ. The correlation predictor enables us to achieve this goal, because it provides us with the pdf of the prediction error ν_b across blocks with a similar $\hat{\rho}_b$ (see Section V). All necessary details concerning the density modeling and NP testing are explained in Section VI.

B. Integrity verification

Having obtained an estimate of the PRNU $\hat{\mathbf{K}}$, some types of tampering operations can be identified as those lacking the PRNU. We work under the assumption that tampered regions will not contain the PRNU from the camera, which is certainly true if the region was pasted from another image from a different camera. It is also true if the region is from an image coming from the same camera as long as its spatial alignment in the image is different than in the forged image. We note that some malicious changes in the image may preserve the PRNU, such as changing the color of a stain to blood stain [38]. Such manipulations will not be detected using our method.

If the forged image was modified using some known geometrical transformation, such as resizing or cropping, the PRNU must be preprocessed in the same manner before applying our forgery detection algorithm. If the geometrical operation is not known, the forgery detection algorithm might still apply after the geometrical transformation is estimated and the image is properly aligned. For this purpose, we might use the PRNU itself as a registration pattern or apply other forensic techniques, such as estimation of resampling parameters [19].

The forgery detection at each pixel is formulated as hypothesis testing (9) applied to a block surrounding the pixel. We test for the presence of PRNU in each sliding block separately and finally fuse all local decisions. As in Camera Identification, we assume that, *over each block*, the unknown multiplicative factor $\mathbf{T}[i]$ is constant and the noise term is WGN with an unknown variance $\boldsymbol{\sigma}_{\Xi}^2[i]$. In contrast to Camera Identification, it is not necessary to estimate these unknown factors, because we test each block separately. Thus, we readily obtain the optimal detector as the normalized correlation

$$\rho_b = corr\left(\mathbf{X}_b, \mathbf{W}_b\right) . \tag{17}$$

The distribution of this test statistics under hypothesis H_0, $p(x|H_0)$, is obtained by correlating the known signal $\mathbf{X}_b$ with noise residuals from other cameras. The distribution of ρ_b under H_1, $p(x|H_1)$, can be again obtained using the same *predictor* as already mentioned in Section IV.A. In simple words, the predictor tells us what the value of the test statistics ρ_b and its distribution were if the block b was not tampered and indeed came from the camera.

Specific implementation details of this forgery detection algorithm and all experiments appear in Section VI.

V. CORRELATION PREDICTOR

In this section, we construct a predictor of the correlation $\rho_b = corr(\mathbf{X}_b, \mathbf{W}_b)$ on small blocks for images coming from the same camera as the PRNU. From experiments, we determined that the three factors that influence ρ_b the most are image intensity, texture, and signal flattening.

The predictor is constructed as a mapping from some feature space to a real number in the interval [0,1]—the predicted value of ρ_b. The block size should not be too large because then the assumption of stationarity of $\mathbf{T}$ and $\boldsymbol{\sigma}_{\Xi}^2$ is less likely to hold, neither can it be too small to avoid the lack of statistically significant data. For typical sizes of digital camera images with 1 million pixels or more, square blocks with $|B_b| = 128\times128$ pixels performed overall the best.

Image intensity. Because the PRNU term $\mathbf{I}\hat{\mathbf{K}}$ is multiplicative, the correlation is higher in areas of high intensity. However, due to the finite dynamic range, the PRNU term is not present in saturated regions ($\mathbf{I}[i] = 255$) and is attenuated for $I_{crit} \le \mathbf{I}[i] < 255$ because the chances that the intensity falls into the saturation range is higher. The critical value of intensity I_{crit} depends on the camera. We define the intensity feature as the average image intensity attenuated close to the maximum dynamic range

$$f_{\mathbf{I}} = \frac{1}{|B_b|}\sum_{i\in B_b} att(\mathbf{I}[i]), \qquad (18)$$

where $att(x)$ is the attenuation function

$$att(\mathbf{I}[i]) = \begin{cases} e^{-(\mathbf{I}[i]-I_{crit})^2/\tau}, & \mathbf{I}[i] > I_{crit}, \\ \mathbf{I}[i]/I_{crit}, & \mathbf{I}[i] \le I_{crit}, \end{cases} \qquad (19)$$

and τ is a constant. For example, for our tested Canon G2 camera, we experimentally determined I_{crit} = 250, τ = 6. The algorithm was a simple brute-force search in the intervals $I_{crit}\in[230, \ldots, 255]$, $\tau\in[3, \ldots, 8]$ that produced the best predictor (in the MSE sense) over a mixture of high-intensity cloudy sky images and natural scenes.

Texture. The correlation tends to be lower in textured areas because the variance $\boldsymbol{\sigma}_{\Xi}^2$ is larger and the attenuation factor $\mathbf{T} < 1$ since the denoising filter removes part of the signal of interest. We calculate the *texture feature* $f_{\mathbf{T}}$ from the high-frequency component of the image. Since the denoising filter performs wavelet transform, we conveniently use this intermediate data and generate a high-pass filtered image $\mathbf{F}$ as the inverse wavelet transform of the LH, HH, and HL in the two outmost bands (6 subbands altogether). The texture feature is then computed as

$$f_{\mathbf{T}} = \frac{1}{|B_b|}\sum_{i\in B_b} \frac{1}{1+\mathrm{var}_5(\mathbf{F}[i])}, \qquad (20)$$

where $\mathrm{var}_5(\mathbf{F}[i])$ is the variance of $\mathbf{F}$ in the 5×5 neighborhood of pixel i. The reciprocal normalizes $f_{\mathbf{T}}$ to the interval [0, 1].

Signal flattening. Image processing that is of low-pass filtering nature, such as JPEG compression, further attenuates the PRNU noise and thus decreases the correlation. In a relatively flat and high intensity unsaturated area, the predictor would overestimate the correlation. These “flattened” areas will typically have a low value of the local variance. Thus, we added the third feature $f_{\mathbf{S}}$ defined as the ratio of pixels in the block with average local variance below a certain threshold

$$f_{\mathbf{S}} = \frac{1}{|B_b|}\left|\{i \in B_b \mid \sigma_{\mathbf{I}}[i] < c\mathbf{I}[i]\}\right|, \qquad (21)$$

where c is an appropriately chosen constant that depends on the variance of the PRNU $\mathbf{K}$ (e.g., c = 0.03 for Canon G2) and $\sigma_{\mathbf{I}}^2[i]$ is the local variance of image intensity $\mathbf{I}[i]$ at pixel i estimated from a local 5×5 neighborhood.

From experiments, the correlation coefficient strongly depends on the collective influence of texture and intensity. Sometimes, highly textured regions are also high-intensity regions. Thus, we included the following combined **texture-intensity** feature

$$f_{\mathbf{TI}} = \frac{1}{|B_b|}\sum_{i\in B_b} \frac{att(\mathbf{I}[i])}{1+\mathrm{var}_5(\mathbf{F}[i])}. \qquad (22)$$

Having specified the features, machine learning can now be used to learn the relationship between the features and the correlation. In this paper, we opted for a simple polynomial multivariate least square fitting. Let $\boldsymbol{\rho}$ be a column vector of K normalized correlations (12) calculated for K image blocks and $\mathbf{f_I}$, $\mathbf{f_T}$, $\mathbf{f_S}$, and $\mathbf{f_{TI}}$ be the corresponding K-dimensional feature vectors. We model $\boldsymbol{\rho}$ as a linear combination of the features and their second-order terms

$$\begin{aligned}\boldsymbol{\rho}[k] &= \theta_0 + \theta_1\mathbf{f_I}[k] + \theta_2\mathbf{f_T}[k] + \theta_3\mathbf{f_S}[k] + \theta_4\mathbf{f_{TI}}[k] + \\ &\theta_4\mathbf{f_I}[k]\mathbf{f_I}[k] + \theta_5\mathbf{f_I}[k]\mathbf{f_T}[k] + \ldots + \boldsymbol{\psi}[k],\end{aligned} \qquad (23)$$

where $\boldsymbol{\psi}[k]$ is the modeling noise and $\boldsymbol{\theta}$ is the vector of coefficients to be determined. In (23), there are total of 1+4+10=15 terms. Rewriting (23) in a matrix form, we have

$$\boldsymbol{\rho} = \mathbf{H}\boldsymbol{\theta} + \boldsymbol{\psi}, \qquad (24)$$

where $\mathbf{H}$ is a K×15 matrix of features and their multiplications and $\boldsymbol{\theta} = (\theta_0, \theta_1, \ldots, \theta_{14})$ is the unknown vector parameter. Applying the least square estimator (LSE) to estimate $\boldsymbol{\theta}$, we obtain

$$\hat{\boldsymbol{\theta}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\boldsymbol{\rho} \qquad (25)$$

and the predictor is

$$\hat{\rho}_b = \left[f_{\mathbf{I}}, f_{\mathbf{T}}, f_{\mathbf{S}}, f_{\mathbf{TI}}, f_{\mathbf{I}}f_{\mathbf{I}}, f_{\mathbf{I}}f_{\mathbf{T}}, \ldots\right]\hat{\boldsymbol{\theta}}. \qquad (26)$$

We note that the images used to build the predictor should be as diverse as possible so that the features extracted from the blocks cover a large portion of the feature space. This is in contrast to the requirements for calculating the PRNU factor where we preferred smooth uniform images. By overlapping the blocks, one can extract several hundreds of blocks from one image, depending on the image size. In practice, good predictors can be obtained from as few as 10 images.

As an example, we show in Fig. 3 (top) the true value of the correlation vs. the predicted value for one of the cameras tested in Section VI. This experimental data can be used to determine the pdf of the prediction error ν_b and, in turn, the pdf of the test statistics $p(x|\mathrm{H}_1)$ for both Camera Identification and Integrity Verification. Fig. 3 (bottom) shows the scatter plot of the test statistics ρ obtained using (11) vs. its predicted value obtained using the predictor $\hat{\rho} = \sum_{b=1}^{M} \beta_b \hat{\rho}_b$.

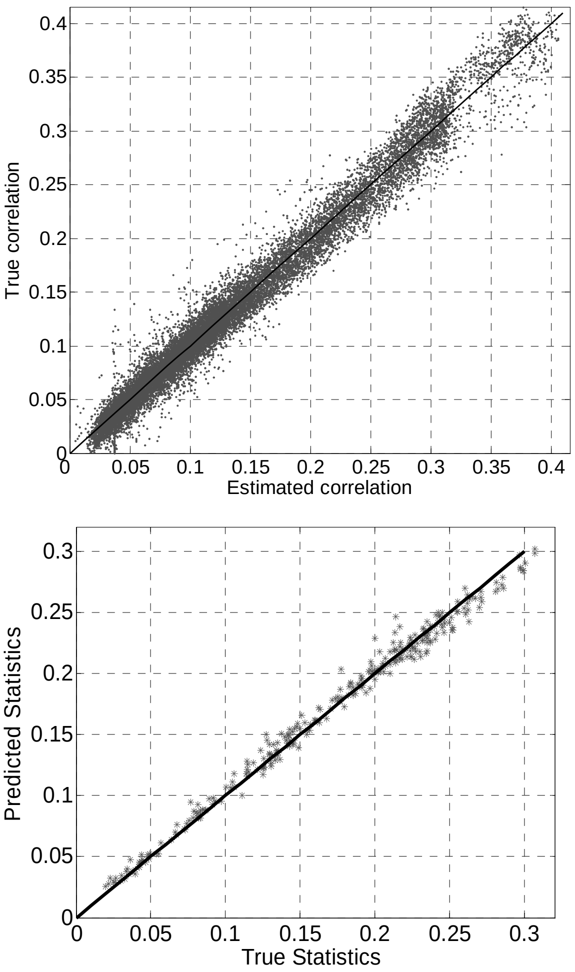


Fig. 3: Top: scatter plot of ρ_b vs. $\hat{\rho}_b$ for K=30,000 128×128 blocks from 300 TIF images for Canon G2. Bottom: true test statistics ρ (11) vs. $\hat{\rho} = \sum_{b=1}^{M} \beta_b \hat{\rho}_b$ for 285 Canon G2 images not used for calculating the PRNU or the predictor. The PRNU was obtained from 30 blue sky images (see Section VI)**.**

We note that if the image under investigation **I** is a JPEG image, a slightly better predictor is achieved if the predictor is trained on JPEG images of approximately the same quality factor. If the image has undergone an unknown processing that influences the values of correlations, the predictions may be off by a multiplicative factor because the PRNU will be attenuated in such images. This does not invalidate our approach, however, because in our model (10) we allow unknown global multiplicative factors a and γ.

We remark that the features can be defined in other ways and evaluated in different domains. We tested predictors based on features all calculated in the spatial domain and obtained a very similar performance. Likewise, other machine learning tools that we tested (neural networks) provided similar results. It is very likely that a more detailed study of the influence of image properties on the test statistics combined with better machine-learning tools will further improve the predictor and thus lead to more accurate Camera Identification and Integrity Verification methods.

Note that all features (18)–(22) can be evaluated using FFT-based convolution, which greatly reduces the complexity and computation time of the resulting camera identification and forgery detection algorithms.

VI. EXPERIMENTS

This section describes implementation details and contains extensive experimental tests of both Device Identification and Integrity Verification. Both tasks can be formulated as binary hypothesis testing and are achieved by detecting the presence of PRNU (see Table III).

TABLE III HYPOTHESIS TESTING FOR BOTH FORENSIC TASKS.

	Camera Identification	Integrity Verification
H_0	Image not taken by camera	Region tampered
	PRNU not present	
H_1	Image taken by the camera	Region non-tampered
	PRNU present	

A. Experimental Setup

We use a wavelet-based denoising filter F, which has been described in detail in our previous work [9] and in the original publication [31]. The filter accepts a parameter, which is the variance of the white additive Gaussian noise that it removes. We set this variance to 3 assuming the dynamic range of 8 bits for **I**. The PRNU term for each camera was calculated from 30 blue sky images or uniformly lit test images obtained using a light box.

The estimated PRNU factor $\hat{\mathbf{K}}$ is further preprocessed by zero-meaning the rows and columns of $\hat{\mathbf{K}}$ and filtering in the Fourier domain as described in Section III.B. The purpose of this step is to remove artifacts that are not unique to the camera sensor and would thus slightly increase false identification for camera ID or produce false integrity verification.

TABLE IV CAMERAS USED IN EXPERIMENTS, PROPERTIES OF THEIR SENSORS, AND IMAGE FORMAT USED FOR TEST IMAGES.

Camera brand	Sensor	Native Resolution	Image Format
Canon PowerShot G2	1/1.8” CCD	2272×1704	TIFF
Canon PowerShot S40	1/1.8” CCD	2272×1704	TIFF
Olympus Camedia C765-1	1/2.5” CCD	2288×1712	TIFF
Olympus Camedia C765-2	1/2.5” CCD	2288×1712	TIFF
Olympus Camedia C3030	1/1.8” CCD	2048×1536	JPEG
Sigma SD9	20.7×13.8mm CMOS-Foveon X3	2268×1512	TIFF

Our set of test cameras includes 5 CCD digital still cameras and one single lens reflex (SLR) camera with the Foveon X3 CMOS sensor. Relevant technical specifications of the cameras are shown in Table IV. Note that the set contains two cameras of exactly the same brand (Olympus C765).

B. Camera Identification

In this section, we first describe the Neyman-Pearson test for the Camera Identification algorithm and then report experimental results.

The identification method starts with dividing the image under investigation into M 128×128 blocks where we assume that the attenuation factor is constant and the noise is white. We calculate in each block b the detection statistic ρ.

To estimate the probabilities of erroneous decisions for Camera Identification, we need the distribution of ρ on images obtained using other cameras, $p(x|\mathrm{H}_0)$, and the distribution of ρ on images obtained using the same camera, $p(x|\mathrm{H}_1)$. The distribution $p(x|\mathrm{H}_0)$ determines the false acceptance ratio (FAR), while $p(x|\mathrm{H}_1)$ determines the false rejection ratio (FRR).

To obtain $p(x|\mathrm{H}_0)$ for each tested camera, we calculated ρ for 2500 images from over 1000 different cameras in their native resolution downloaded from www.flickr.com, including the images from the other five tested cameras. We modeled $p(x|\mathrm{H}_0)$ as Generalized Gaussian distribution[4] $GG(\mu_0, \sigma_0, \alpha_0)$, where the parameters μ_0, σ_0, and α_0 were estimated using the method of moments [37]. In Fig. 4, we show an example of the distribution of ρ under H_0 for Canon G2, while Fig. 5 shows the goodness of the GG fit using the log-tail plot.

Estimating $p(x|\mathrm{H}_1)$ is more difficult because the test statistics strongly depends on the image content. Thus, determining the distribution from available images would be influenced too much by the database of test images unless the database is very large and diverse. Moreover, this way we would obtain overly conservative error estimates for "good" images and too optimistic error estimates for highly textured images. We really should be evaluating the FRR against images of *approximately the same content*. The correlation predictor will enable us to achieve this goal in the following manner. Note that the variance of the prediction error ν_b does not depend on the predicted correlation value (see Fig. 3 top). Thus, data used to train the correlation predictor can also be used to estimate the pdf of the prediction error ν_b.

We model ν_b as $GG(0, \sigma_\nu, \alpha_\nu)$ by fitting the GG model through $\rho_b - \hat{\rho}_b$ for all blocks available for training the predictor. The GG fit is shown using the log-tail plot in Fig. 6. Note that it is the left tail that plays the role in determining the FRR. The fit is thus conservative as the left tail of the data falls off faster than the model. In all our experiments, we constructed the predictor from several thousands of 128×128 blocks obtained from 20 images.

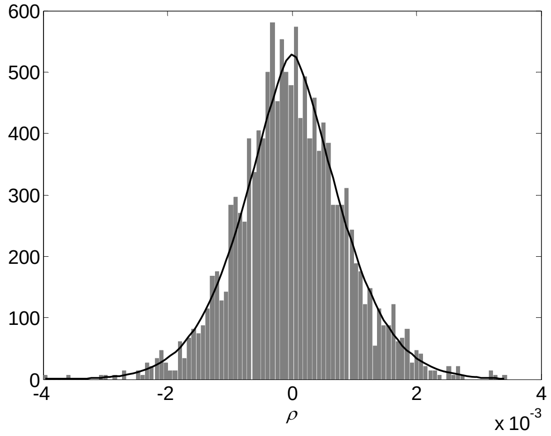


Fig. 4: Test statistics ρ for PRNU factor from Canon G2 and noise residual from 2500 images from other cameras. The curve is the GG fit.

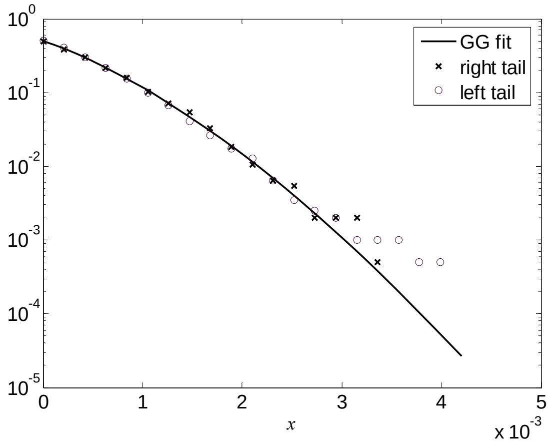


Fig. 5: The log-tail plot of data shown in Fig. 2 showing the probability of observing a value larger than x (for the right tail) and less than $-x$ (for left tail) determined from the model (thick curve) and from measured data as a function of $|x|$.

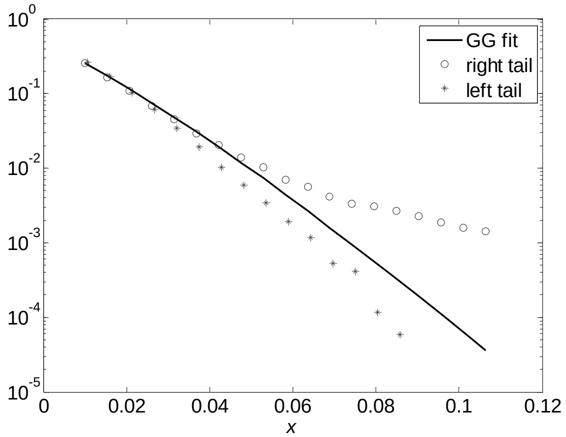


Fig. 6: The log-tail plot for the prediction error showing the probability of observing a value of the error larger than x (for the right tail) and less than $-x$ (for left tail) determined from the model (thick curve) and from measured data as a function of $|x|$.

Since the test statistics (11) is a linear combination of random variables (12), we now have a means to estimate its

[4] $GG(\mu,\sigma,\alpha) = \alpha/(2\sigma\Gamma(1/\alpha))e^{-(|x-\mu|/\sigma)^{\alpha}}$ is the pdf of a GG model with variance $\sigma^2\Gamma(3/\alpha)/\Gamma(1/\alpha)$, mean μ, and shape parameter α.

distribution. The errors from different blocks, however, are *not independent* because it is likely that neighboring blocks will have similar content and thus the prediction errors will likely be similar. To avoid modeling these dependencies and to obtain a conservative estimate of the FRR, we made the assumption that the prediction errors are *completely correlated*, in which case ρ is $GG(\mu_1, \sigma_1, \alpha_1)$ with

$$\mu_1 = \sum_{b=1}^{M} \beta_b \hat{\rho}_b, \;\; \sigma_1^2 = \left(\sum_{b=1}^{M} \beta_b\right)^2 \sigma_\upsilon^2, \;\; \alpha_1 = \alpha_\nu . \quad (27)$$

We decide that **I** was taken by the camera when $\rho > t$. As we use the Neyman-Pearson criterion, we set the decision threshold t for the test statistics ρ to obtain

$$\text{FAR} = \int_t^{\infty} GG(0, \sigma_0, \alpha_0) dx = 10^{-5}$$

and calculate the probability of false rejection FRR as

$$\text{FRR} = \int_{-\infty}^{t} GG(\mu_1, \sigma_1, \alpha_1) dx . \quad (28)$$

We remind that the FRR is the probability that an image (taken with the camera) with the same value of predicted correlation would be mistakenly identified as not coming from the camera.

C. Robustness to processing

All tests were performed on images transformed to grayscale. For example, the noise residual was extracted from each color channel and then combined into a grayscale image using the linear combination as in RGB to grayscale conversion. The estimated PRNU factors for each color channel were also combined in this manner. The predictor features were always extracted from the image converted to grayscale. We tested all available test images from each camera (approximately 300 per camera) that were not used for calculation of the PRNU or the predictor in their native format as shown in Table IV. Then, we repeated the tests after the images were processed using some common image processing operations that included JPEG compression with quality factors 90 and 75, denoising using the Wiener filter with a 3×3 window[5], gamma correction with $\gamma = 0.5$, and scaling by a factor of 0.6. The last three operations were followed by JPEG compression with quality factor 90.

In Tables V–X, we display the FRR (at FAR=10^{-5}) for eight selected images. For each camera, we selected the worst image in the set (No. 1) and the best image (No. 8). By best and worse, we understand images with the lowest and highest value of the test statistics. The other six images were chosen uniformly between the extremes. In Fig. 7, we show the eight images used in tests with Canon G2. We note that the worst images were always heavily textured, such as images taken in the woods combined with very bright (saturated) and dark areas. The PRNU in such images is largely attenuated and difficult to detect.

In agreement with our expectations, the combination of Wiener denoising and JPEG compression increases the FRR but not beyond making the identification process unreliable (FRR of a few percent for FAR = 10^{-5}).

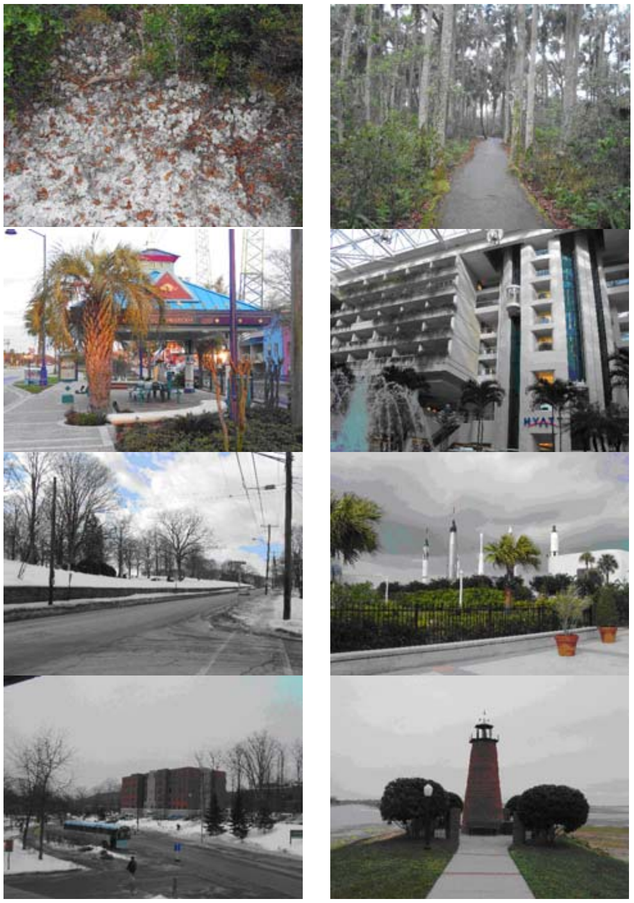


Fig. 7: Eight Canon G2 images used in experiments reported in Table III.

The identification is markedly worse for Sigma SD9. Its PRNU is weaker because its 20.7×13.8 mm CMOS sensor has larger pixels, which translates to better pixel uniformity in sensor manufacturing. We believe that the noticeable difference between FRR for RAW and 90% quality JPEGs (Table X) is due to the fact that this camera is equipped with the Foveon sensor and thus does not have color interpolation in its processing chain. Consequently, the PRNU energy is more in high spatial frequencies than for the other cameras because color interpolation acts as a low-pass filter shifting the energy of the PRNU to medium frequencies, which increases resistance to JPEG compression.

TABLES V–X FRR FOR FAR=10^{-5} FOR ALL 6 TESTED CAMERAS.

Canon S40	RAW	JPG 90	JPG 75	Wiener JPG 90	$\gamma = 0.5$ JPG 90	$s = 0.6$ JPG 90
1	3.5e–2	3.1e–2	6.0e–2	9.6e–2	4.2e–2	1.7e–3
2	9.5e–3	8.4e–3	2.5e–2	5.6e–2	1.8e–2	6.2e–3
3	3.5e–4	4.3e–4	8.1e–3	1.9e–2	4.1e–3	3.7e–3
4	2.4e–5	2.5e–5	1.2e–3	4.2e–3	7.5e–4	7.9e–4
5	3.9e–7	2.6e–7	1.5e–4	2.4e–4	1.7e–5	1.6e–4
6	6.9e–11	1.8e–10	3.2e–4	3.6e–5	1.3e–4	4.2e–6
7	1.3e–12	1.4e–13	2.3e–6	7.6e–7	4.8e–6	6.5e–9
8	8.4e–15	1.3e–13	4.1e–5	2.5e–6	2.9e–5	1.0e–7

[5] The noise variance for the Wiener filter was chosen as the average local variance over the whole image (default setting in Matlab).

TABLE VI

Canon G2	RAW	JPG 90	JPG 75	Wiener JPG 90	$\gamma = 0.5$ JPG 90	$s = 0.6$ JPG 90
1	2.9e–2	2.7e–2	4.7e–2	1.0e–1	2.7e–2	6.4e–3
2	4.8e–3	6.9e–3	1.8e–2	3.8e–2	5.5e–3	2.3e–3
3	6.2e–4	1.9e–3	9.4e–3	2.0e–2	3.0e–3	1.2e–3
4	3.5e–5	1.6e–4	2.1e–3	6.0e–3	6.6e–4	1.9e–6
5	5.0e–7	2.7e–6	1.9e–4	3.9e–4	7.3e–5	5.9e–7
6	1.8e–8	1.1e–7	1.7e–5	1.8e–5	3.9e–6	3.7e–9
7	5.8e–9	4.9e–8	9.4e–6	9.9e–6	1.8e–6	1.8e–9
8	8.6e–11	2.3e–9	4.3e–6	3.6e–7	1.5e–7	3.0e–12

TABLE VII

Olympus C765-1	RAW	JPG 90	JPG 75	Wiener JPG 90	$\gamma = 0.5$ JPG 90	$s = 0.6$ JPG 90
1	5.2e–2	6.7e–2	1.1e–1	1.7e–1	5.6e–2	2.4e–2
2	1.1e–2	1.8e–2	3.3e–2	3.3e–2	7.6e–3	1.5e–2
3	6.0e–3	1.0e–2	2.1e–2	2.9e–2	6.2e–3	7.2e–3
4	1.1e–3	2.7e–3	9.6e–3	9.5e–3	1.7e–3	4.5e–3
5	3.5e–4	1.3e–3	6.7e–3	4.3e–3	2.1e–3	8.5e–4
6	3.7e–5	9.4e–5	5.6e–4	8.4e–4	2.5e–4	1.0e–4
7	5.2e–6	1.4e–5	2.7e–4	4.5e–4	1.3e–4	2.2e–5
8	1.9e–8	7.0e–8	1.5e–5	1.7e–5	3.5e–5	2.3e–8

TABLE VIII

Olympus C765-2	RAW	JPG 90	JPG 75	Wiener JPG 90	$\gamma = 0.5$ JPG 90	$s = 0.6$ JPG 90
1	6.5e–2	8.3e–2	1.4e–1	2.2e–1	8.2e–2	3.9e–2
2	2.8e–2	4.1e–2	7.8e–2	1.2e–1	3.2e–2	2.9e–2
3	9.2e–3	1.9e–2	4.9e–2	7.4e–2	2.0e–2	2.2e–2
4	6.1e–4	1.9e–3	1.1e–2	1.2e–2	2.5e–3	3.5e–3
5	1.6e–4	6.7e–4	5.5e–3	5.3e–3	1.4e–3	1.8e–3
6	4.2e–6	1.2e–5	2.1e–4	3.8e–4	1.3e–4	3.3e–5
7	1.8e–6	1.9e–5	5.6e–4	3.6e–4	1.7e–4	3.0e–5
8	1.4e–9	7.0e–9	2.0e–6	2.2e–6	3.6e–7	2.5e–7

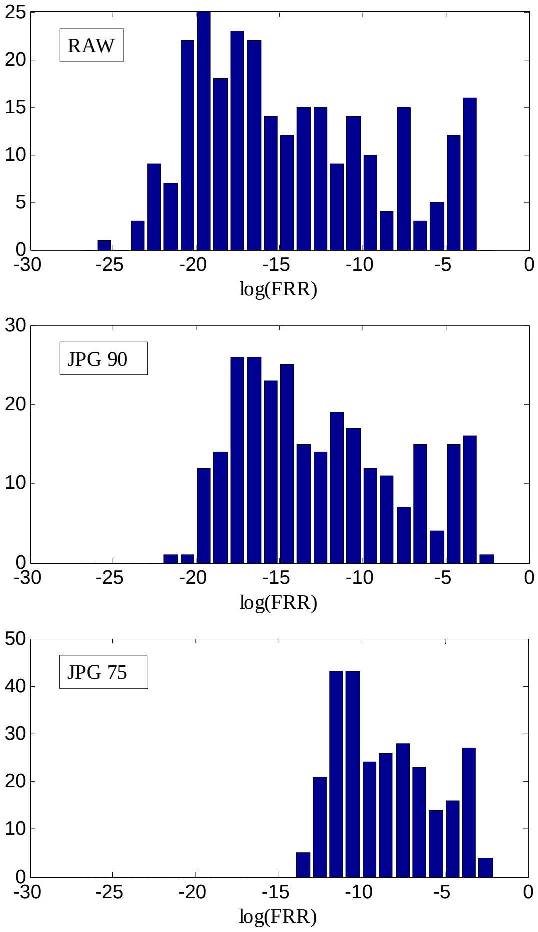
TABLE IX

Olympus C3030	RAW	JPG 90	JPG 75	Wiener JPG 90	$\gamma = 0.5$ JPG 90	$s = 0.6$ JPG 90
1	4.0e–2	4.4e–4	3.4e–3	9.4e–2	1.1e–2	7.3e–5
2	1.5e–4	4.7e–5	3.5e–4	4.1e–3	2.9e–3	5.9e–5
3	1.4e–5	1.3e–5	1.9e–4	2.0e–3	3.3e–3	7..8e–6
4	9.9e–7	7.7e–7	2.6e–5	2.2e–4	6.3e–4	1.2e–6
5	3.0e–7	2.5e–7	8.9e–6	1.4e–4	8.2e–4	1.9e–7
6	3.9e–9	1.2e–9	1.4e–7	3.2e–6	2.0e–4	1.1e–10
7	2.1e–10	1.0e–10	3.5e–8	3.0e–8	4.8e–5	1.5e–11
8	5.8e–12	7.1e–12	1.5e–9	5.5e–9	8.1e–5	1.3e–13

TABLE X

Sigma SD9	RAW	JPG 90	JPG 75	Wiener JPG 90	$\gamma = 0.5$ JPG 90	$s = 0.6$ JPG 90
1	7.7e–2	1.5e–1	2.5e–1	3.0e–1	1.6e–1	1.5e–1
2	3.1e–2	9.4e–2	1.6e–1	2.3e–1	9.0e–2	1.1e–1
3	7.6e–3	4.5e–2	1.0e–1	1.1e–1	6.2e–2	6.3e–2
4	2.3e–3	1.9e–2	5.4e–2	7.2e–2	3.7e–2	2.8e–2
5	3.1e–4	1.1e–2	4.3e–2	4.5e–2	3.9e–2	2.1e–2
6	9.9e–5	7.4e–3	4.0e–2	2.3e–2	4.1e–2	1.8e–2
7	1.1e–5	3.8e–3	2.6e–2	1.2e–2	1.6e–2	2.0e–2
8	2.2e–6	1.1e–3	1.2e–2	4.6e–3	8.8e–3	3.5e–3

To obtain a better insight of the performance of the proposed camera identification method on a larger scale, in Fig. 8 we show the histograms of $\log_{10}$ FRR obtained for all 274 tested images for Canon G2 (and for FAR = 10^{-5} as before). Each histogram corresponds to one type of processing.



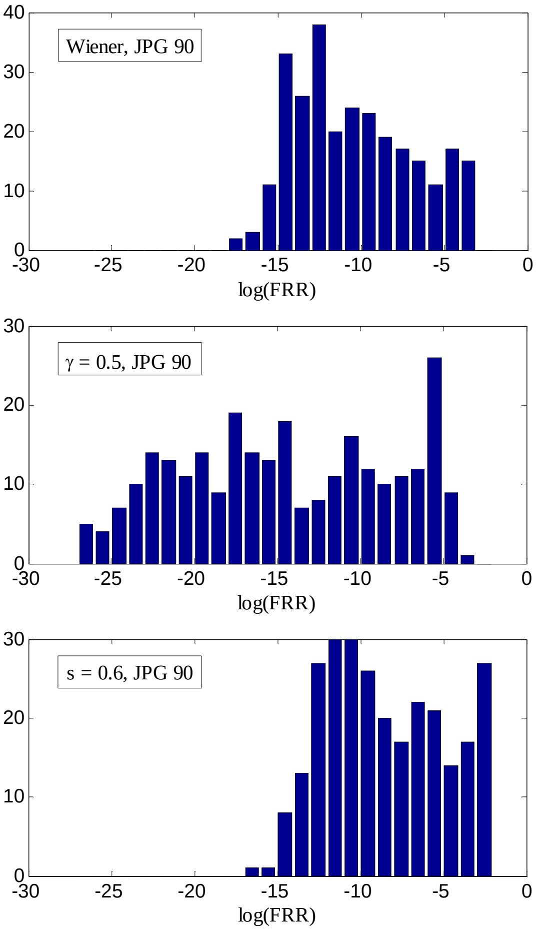


Fig. 8: Histograms of $\log_{10}$ FRR for all 274 tested Canon G2 images after processing.

D. Integrity Verification

In this section, we report the implementation details and experimental results for the Integrity Verification algorithm as described in Section IV.B.

The algorithm proceeds by sliding a 128×128 block across the image and evaluates the test statistics $\rho_b = corr(\mathbf{X}_b, \mathbf{W}_b)$ for each block b. As in Camera Identification, the pdf $p(x|\mathrm{H}_0)$ is estimated from more than 30,000 correlations between 128×128 blocks $\mathbf{X}_b$ and the noise residual $\mathbf{W}_b$ from blocks coming from 200 images obtained using *other* cameras. A GG model was then fit through the correlations. The pdf $p(x|\mathrm{H}_1)$ is obtained for each block from the predictor in the same way as in the previous section and modeled as $GG(\hat{\rho}_b, \sigma_v, \alpha_v)$.

We set the decision threshold t for the test statistics ρ_b to twice the standard deviation of $p(x|\mathrm{H}_0)$. Accepting the Neyman-Pearson criterion, we decide that block b has been potentially tampered if $\rho_b < t$ but only attribute this decision to the central pixel i of the block. As a result, for an $N_1 \times N_2$ image, we obtain an $(N_1 – 127) \times (N_2 – 127)$ binary array $\mathbf{Z}[i] = \rho_b < t$ indicating the potentially tampered pixels with $\mathbf{Z}[i] = 1$. Our choice of threshold gives about 1% probability of misidentifying a tampered block as non-tampered.

Since the NP criterion is oblivious to the distribution $p(x|\mathrm{H}_1)$, it decides 'tampered' whenever $\rho_b < t$ even though ρ_b may be "more compatible" with $p(x|\mathrm{H}_1)$. This is more likely to occur when ρ_b is small, such as for highly textured blocks. To reach a conservative decision and avoid labeling non-tampered pixels as tampered, we select a threshold[6] $\beta > 0$ and label pixels i for which

$$p[i] = \int_{-\infty}^{t} p(x \mid \mathrm{H}_1)dx > \beta \,, \qquad (29)$$

as non-tampered (we set $\mathbf{Z}[i] = 0$). Here, $p[i]$ is the probability of falsely labeling a pixel as tampered when it was not. The resulting binary map $\mathbf{Z}$ identifies the forged regions in their raw form.

At first sight, it appears that Bayesian hypothesis testing would be perhaps more appropriate, however, our experiments indicate that this NP approach gives overall slightly better results over the Bayesian approach.

The block dimensions impose a lower bound on the size of tampered regions that our algorithm can identify. Thus, we remove all simply connected tampered regions from $\mathbf{Z}$ that contain fewer than 64×64 pixels (one quarter of the number of pixels in the block). Finally, we dilate the resulting binary map $\mathbf{Z}$ with a square 20×20 kernel. The purpose of this final step is to compensate for the fact that we attribute the decision about the whole block only to its central pixel and thus potentially miss portions of the tampered boundary region. The resulting processed binary array $\mathbf{Z}$ is what we take as the final output from the Integrity Verification algorithm.

1) Tests on forged images

We manipulated two images from each tested camera. The forgeries varied from a simple copy-move within one image to object removing and pasting objects from other images. The manipulated images were stored in three versions: TIFF and JPEG with quality factors 90 and 75. To further test the robustness of the proposed algorithm, we processed the forged TIFF images in three additional ways to simulate typical processing or enhancement that might be done in real life. The TIFF forgery was denoised using the Wiener 3×3 filter with the default noise variance in Matlab, gamma corrected with γ = 0.5, and scaled to 60% of its original size. All three processed images were then additionally JPEG compressed with quality factor 90.

We give full results of this experiment for one camera and briefly comment on the remaining cases. Figs. 9 and 10 show the output of the Integrity Verification algorithm for two forged images coming from the Olympus C765-1 camera, including the original image, the forged image, the potentially tampered pixels as determined by the NP criterion only, and the final detected region highlighted in the forged image. The algorithm overall performs very well and is able to identify the tampered region very accurately even from processed images. In Fig. 10, we included the

[6] The threshold was set to $\beta = 0.01$.

worst result of this test, which shows a woodchuck on grass. The forged image has the same woodchuck duplicated. Both the woodchuck's fur and the grass are highly textured regions with low values of the test statistics ρ_b. While the algorithm does not produce any falsely tampered regions, the pasted woodchuck is less accurately identified, especially after combined Wiener filtering and JPEG compression.

In Fig. 11, we show more examples from the other five tested cameras for the forged images stored as 90% quality JPEG. In all cases, the tampered regions were accurately estimated.

As an implementation detail, we note that the sliding-window calculation of image features for the predictors can be computed efficiently using convolution implemented using FFT. A Matlab implementation of the proposed algorithm was able to verify the integrity of a 4 megapixel image in 4–5 minutes on a computer equipped with a 3.4 GHz Pentium processor.

2) Tests on non-forged images

To estimate the probability of falsely identifying a non-forged region as tampered, we applied the proposed algorithm to more than 400 non-forged images from Canon G2, Olympus C765-1 and Olympus C3030 stored in the JPEG format with quality factor 90. The false alarms have occurred in specific regions that can be easily characterized (two examples are shown in Fig. 12). All misidentified regions contain saturated background with dark regions, often combined with a complex texture, such as saturated sky showing through a mesh of black tree branches. Such regions naturally lack the PRNU and thus cannot be authenticated using our algorithm. Although not incorporated in this paper, these singular cases could probably be removed by post-processing or expanding the predictor by adding a suitable feature.

3) Large scale test on forged images

In order to assess the accuracy of the proposed algorithm, we subjected it to a large scale test. We took 345 Canon G2 images and pasted into them regions from images from other cameras. The regions were rectangular with varying shapes and sizes with the smallest and largest sides of 228 and 512 pixels, respectively. All forged images were saved with JPEG quality factor 90 and 75. We then ran our Integrity Verification algorithm on all forged images registering the percentage of correctly detected forged pixels and the percentage of falsely identified pixels (non-tampered pixels marked as tampered). Both percentages were calculated with respect to the size of the forged area.

The histograms of results for all 345 tested images are shown in Figs. 13 and 15. We summarize the test by saying that for the JPEG quality factor 90, in 85% of forgeries at least 2/3 of the forged region was correctly identified, while for only 23% of forgeries, the number of falsely identified pixels was larger than 20% of the forged region. For JPEG quality factor 75, in 73% of forgeries at least 2/3 of the forged region was correctly identified, while in 21% of cases we saw more than 20% of falsely identified pixels. The falsely identified areas were generally located around the boundary of the real forged area due to the 128×128 block size of the sliding window and the dilation post processing. By visually inspecting the outliers in this experiment, we concluded that the few cases when a large portion of the tampered region was missed corresponded to the situation when the pasted region contained a large dark region in which the PRNU is naturally suppressed. The few large false positives all were of the type already mentioned above and shown in Fig. 12. In all cases, the algorithm was able to correctly outline the shape of the tampered region.

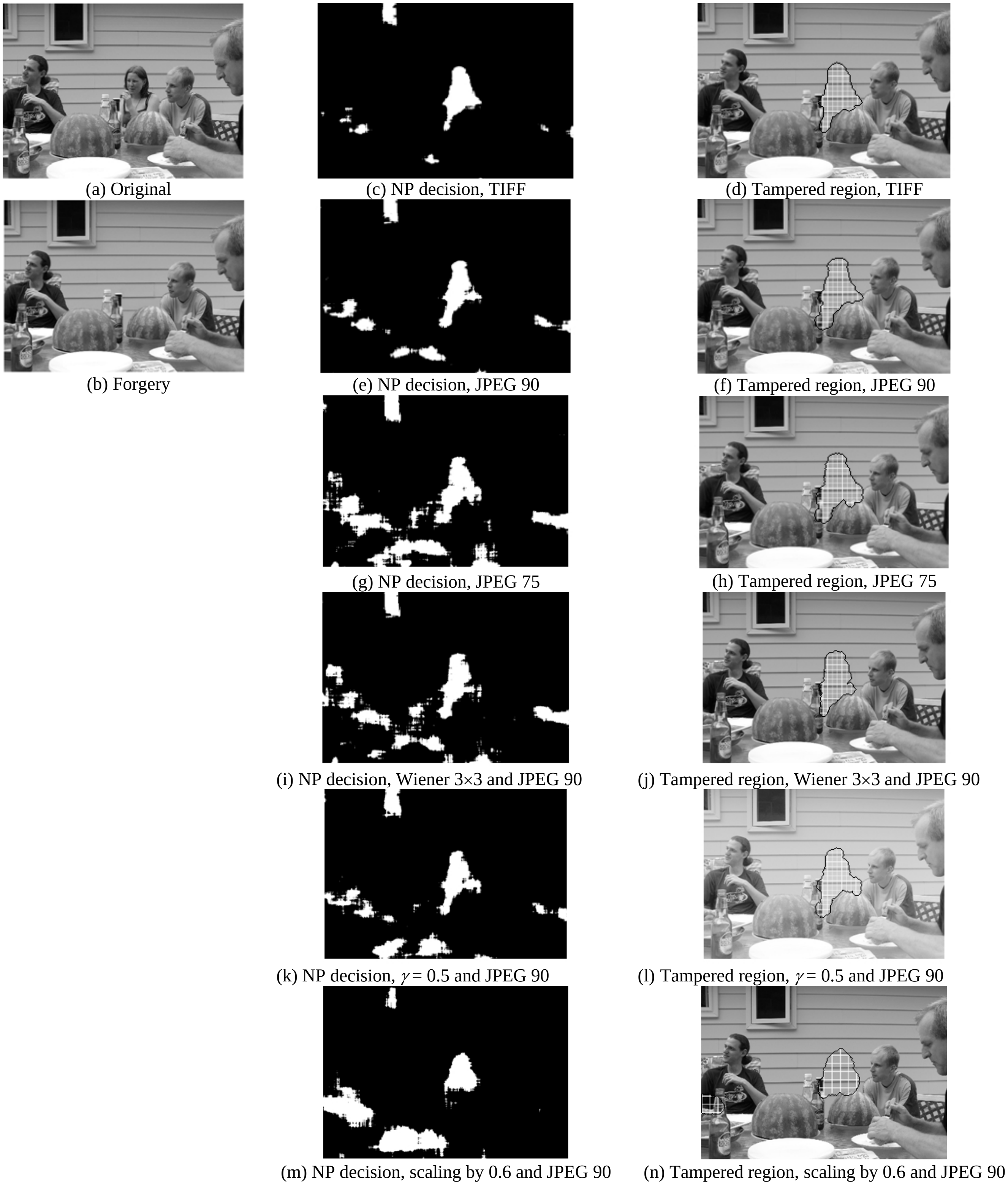
VII. CONCLUSIONS

In this paper, we present a unified framework for Camera Identification and image Integrity Verification. Both digital forensic tasks are approached through detection of pixels' photo-response non-uniformity (PRNU) in the image. The PRNU is first estimated using Maximum Likelihood principle from a simplified model of sensor output. The same model is then used to formulate the task of detecting the PRNU as a hypothesis testing problem using Neyman-Pearson criterion. Appropriate optimal detection statistics are derived for both Camera Identification and Integrity Verification. The distribution of the test statistics under both hypotheses is obtained experimentally from a large number of images from other cameras and using a correlation predictor. The Camera Identification algorithm is tested on six digital cameras. It can reliably identify the source camera from its images even after combined processing and JPEG compression. Compared to our previous work on this subject, the new approach makes better use of available data (significantly fewer images are needed to obtain the PRNU factor) and allows more accurate error estimates.

In our future effort, we plan to investigate other methods for PRNU estimation, such as the image separation using Bayesian source separation scheme [39]. Also, there is potential to use the linear pattern for forensic purposes for camera brand identification (Device Classification). We are currently extending the camera identification technology from images that have been simultaneously cropped and scaled.

ACKNOWLEDGMENT

The work on this paper was supported by the AFOSR grant number FA9550-06-1-0046. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Air Force Research Laboratory, or the U.S. Government.



(a) Original

(c) NP decision, TIFF

(d) Tampered region, TIFF

(b) Forgery

(e) NP decision, JPEG 90

(f) Tampered region, JPEG 90

(g) NP decision, JPEG 75

(h) Tampered region, JPEG 75

(i) NP decision, Wiener 3×3 and JPEG 90

(j) Tampered region, Wiener 3×3 and JPEG 90

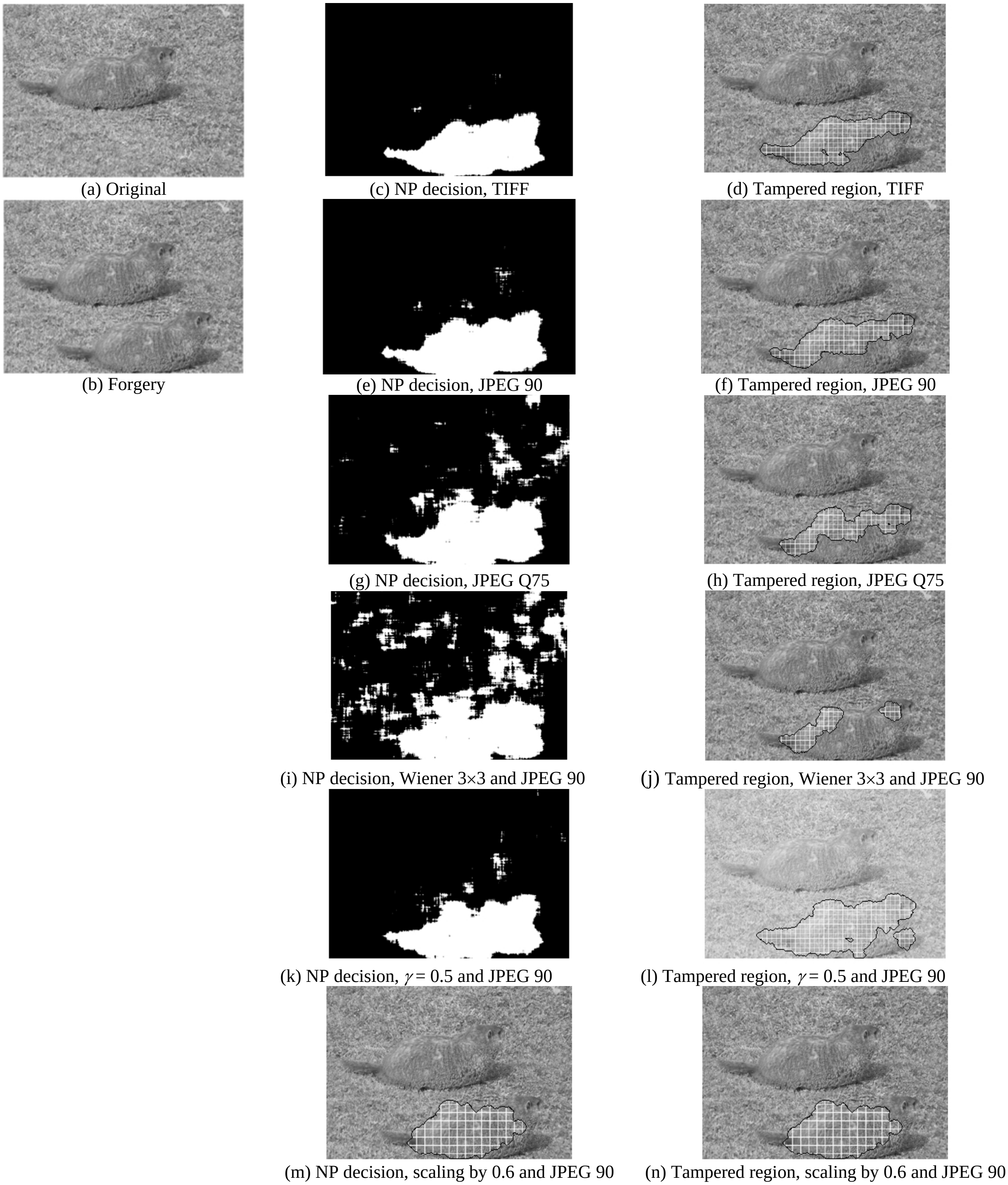
(k) NP decision, $\gamma = 0.5$ and JPEG 90

(l) Tampered region, $\gamma = 0.5$ and JPEG 90

(m) NP decision, scaling by 0.6 and JPEG 90

(n) Tampered region, scaling by 0.6 and JPEG 90

Fig. 9: An Olympus C765-1 forgery and its detection. (a) original image, (b) forged image, (c), (e), (g), (i), (k), (m) are the potentially tampered pixels for TIFF, JPEG 90, JPEG 75, Wiener filtered and JPEG 90, gamma corrected and JPEG 90, and scaled to and JPEG 90, while (d), (f), (h), (j), (l), (n) display the corresponding final forgery detection results.



(a) Original

(c) NP decision, TIFF

(d) Tampered region, TIFF

(b) Forgery

(e) NP decision, JPEG 90

(f) Tampered region, JPEG 90

(g) NP decision, JPEG Q75

(h) Tampered region, JPEG Q75

(i) NP decision, Wiener 3×3 and JPEG 90

(j) Tampered region, Wiener 3×3 and JPEG 90

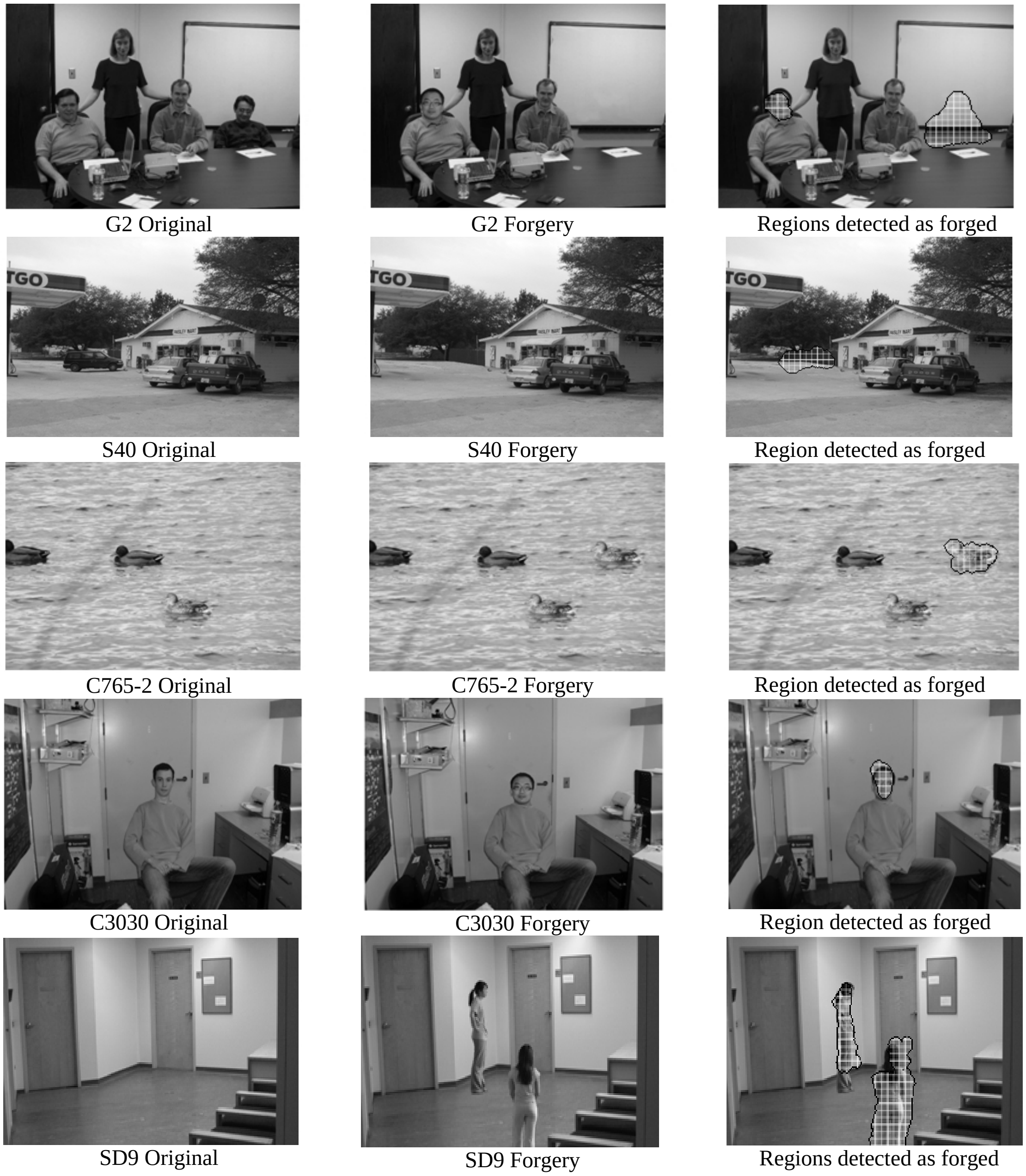
(k) NP decision, $\gamma = 0.5$ and JPEG 90

(l) Tampered region, $\gamma = 0.5$ and JPEG 90

(m) NP decision, scaling by 0.6 and JPEG 90

(n) Tampered region, scaling by 0.6 and JPEG 90

Fig. 10: An Olympus C765-1 forgery and its detection. (a) original image, (b) forged image, (c), (e), (g), (i), (k), (m) are the potentially tampered pixels for TIFF, JPEG 90, JPEG 75, Wiener filtered and JPEG 90, gamma corrected and JPEG 90, and scaled to and JPEG 90, while (d), (f), (h), (j), (l), (n) display the corresponding final forgery detection results.


G2 Original

G2 Forgery

Regions detected as forged

S40 Original

S40 Forgery

Region detected as forged

C765-2 Original

C765-2 Forgery

Region detected as forged

C3030 Original

C3030 Forgery

Region detected as forged

SD9 Original

SD9 Forgery

Regions detected as forged

Fig. 11: Integrity verification results for tampered images from the other five tested cameras Canon G2, Canon S40, Olympus C765-2, Olympus C3030, and Sigma SD9. All tampered images were saved as JPEG with quality factor 90.

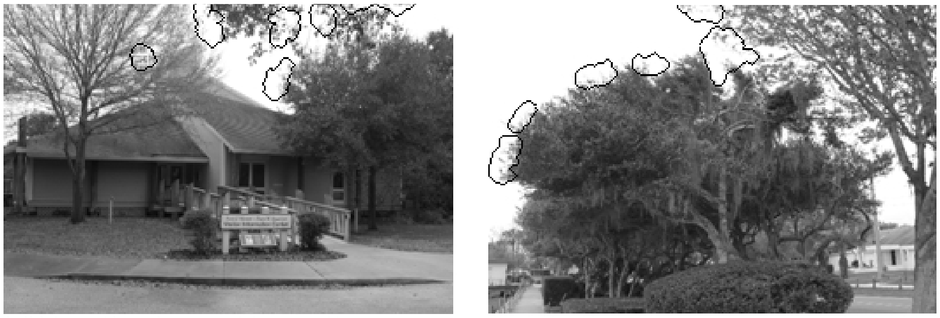


Fig. 12: Example of regions misidentified as tampered.

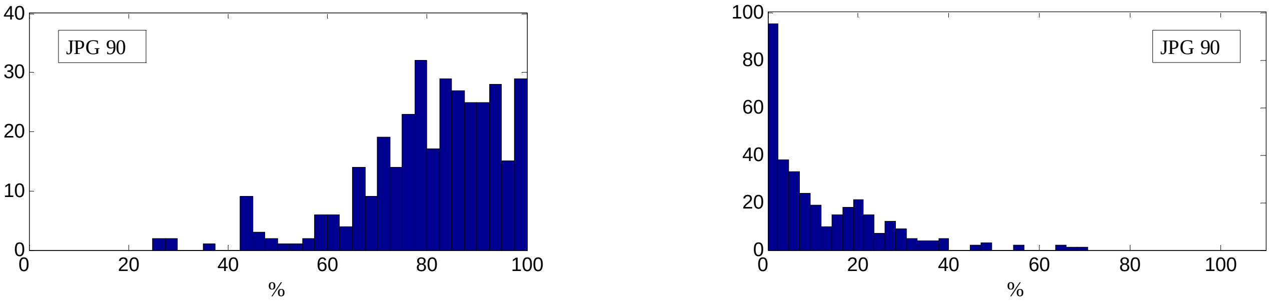


Fig. 13: Histogram of the percentage of correctly identified tampered pixels (left) and falsely identified pixels (right) over 345 tampered images from Canon G2 compressed using JPEG with quality factor 90.

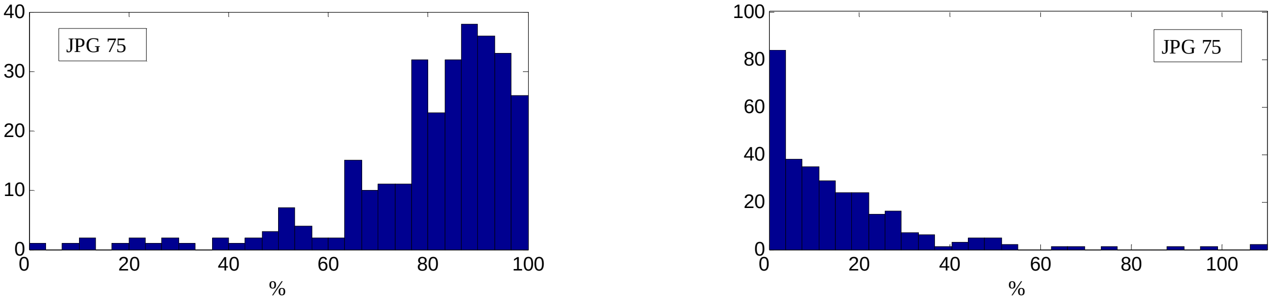


Fig. 14: Histogram of the percentage of correctly identified tampered pixels (left) and falsely identified pixels (right) over 345 tampered images from Canon G2 compressed using JPEG with quality factor 75.

REFERENCES

[1] Kennedy, D.: “Editorial Retraction.” Science, vol. **211**(5759), p. 335, 2006.

[2] Pearson, H.: “Image Manipulation: CSI: Cell biology.” Nature, vol. **434**, pp. 952–953, 2005.

[3] Bayram, S., Sencar, H.T., Memon, N.: “Source Camera Identification Based on CFA Interpolation.” *Proc. ICIP’05*, Genoa, Italy, September 2005.

[4] Swaminathan, A., Wu, M., and Liu, K.J.R.: “Non-intrusive Forensic Analysis of Visual Sensors Using Output Images.” *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing* (ICASSP'06), May 2006.

[5] Swaminathan, A., Wu, M., and Liu, K.J.R.: “Image Authentication via Intrinsic Fingerprints.” *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, January 29–February 1, San Jose, CA, pp. 1J–1K, 2007.

[6] Kharrazi, M., Sencar, H.T., and Memon, N.: “Blind Source Camera Identification,” *Proc. ICIP’04*, Singapore, October 24–27, 2004.

[7] Kurosawa, K., Kuroki, K., and Saitoh, N.: “CCD Fingerprint Method – Identification of a Video Camera from Videotaped Images.” *Proc. ICIP’99*, Kobe, Japan, pp. 537–540, October 1999.

[8] Geradts, Z., Bijhold, J., Kieft, M., Kurosawa, K., Kuroki, K., and Saitoh, N.: “Methods for Identification of Images Acquired with Digital Cameras.” *Proc. of SPIE*, *Enabling Technologies for Law Enforcement and Security*, vol. 4232, pp. 505–512, February 2001.

[9] Lukáš, J., Fridrich, J., and Goljan, M.: “Digital Camera Identification from Sensor Pattern Noise.” *IEEE Transactions on Information Security and Forensics*, vol. 1(2), pp. 205–214, June 2006.

[10] Gou, H., Swaminathan, A., and Wu, M.: “Robust Scanner Identification Based on Noise Features.” *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, January 29–February 1, San Jose, CA, pp. 0S–0T, 2007.

[11] Khanna, N., Mikkilineni, A.K., Chiu, G.T.C., Allebach, J.P., and Delp, E.J. III: “Forensic Classification of Imaging Sensor Types.” *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, January 29–February 1, San Jose, CA, pp. 0U–0V, 2007.

[12] Sankur, B., Celiktutan, O., and Avcibas, I.: “Blind Identification of Cell Phone Cameras.” *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, January 29–February 1, San Jose, CA, pp. 1H–1I, 2007.

[13] Gloe, T., Franz, E., and Winkler, A.: “Forensics for Flatbed Scanners.” *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, January 29–February 1, San Jose, CA, pp. 1I–1J, 2007.

[14] Khanna, N., Mikkilineni, A.K., Chiu, G.T.C., Allebach, J.P., and Delp, E.J. III: “Scanner Identification Using Sensor Pattern Noise.” *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, January 29–February 1, San Jose, CA, pp. 1K–1L, 2007.

[15] Popescu, A.C. and Farid H.: “Statistical Tools for Digital Forensic.” in J. Fridrich (ed.): *6th International Workshop on Information Hiding*, LNCS vol. 3200, Springer-Verlag, Berlin-Heidelberg, New York, pp. 128–147, 2004.

[16] Ng, T.-T., Chang, S.-F., and Sun, Q.: “Blind Detection of Photomontage Using Higher Order Statistics.” *Proc. IEEE International Symposium on Circuits and Systems,* vol. **5**, Vancouver, Canada, pp. v-688–v-691, 2004.

[17] Avcibas, I., Bayram, S., Memon, N., Ramkumar, M., and Sankur, B.: “A Classifier Design for Detecting Image Manipulations.” *Proc. ICIP’04*, vol. **4**, pp. 2645–2648, 2004.

[18] Lin, Z., Wang, R., Tang, X., and Shum, H.-Y.: “Detecting Doctored Images Using Camera Response Normality and Consistency.” *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. **1**, pp. 1087–1092, 2005.

[19] Popescu, A.C. and Farid, H.: “Exposing Digital Forgeries by Detecting Traces of Resampling.” *IEEE Transactions on Signal Processing*, vol. **53**(2), pp. 758–767, 2005.

[20] Popescu, A.C. and Farid, H.: “Exposing Digital Forgeries in Color Filter Array Interpolated Images.” *IEEE Transactions on Signal Processing*, vol. **53**(10), pp. 3948–3959, 2005.

[21] Johnson, M.K. and Farid, H.: “Exposing Digital Forgeries by Detecting Inconsistencies in Lighting.” *Proc. ACM Multimedia and Security Workshop*. New York, pp. 1–9, 2005.

[22] Farid, H.: “Exposing Digital Forgeries in Scientific Images.” *Proc. ACM Multimedia & Security Workshop*. Geneva, Switzerland, pp. 29–36, 2006.

[23] Johnson, M.K. and Farid, H.: "Exposing Digital Forgeries Through Chromatic Aberration." *Proc. ACM Multimedia and Security Workshop*. Geneva, Switzerland, pp. 48–55, 2006.

[24] Chen, W. and Shi, Y.: "Image Splicing Detection Using 2D Phase Congruency and Statistical Moments of Characteristic Function." *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, January 29–February 1, San Jose, CA, pp. 0R–0S, 2007.

[25] Fridrich, J., Soukal, D., and Lukáš, J.: "Detection of Copy-Move Forgery in Digital Images." *Proc. Digital Forensic Research Workshop*. Cleveland, August 2003.

[26] Popescu, A.C. and Farid, H.: "Exposing Digital Forgeries by Detecting Duplicated Image Regions." *Technical Report,* TR2004-515. Dartmouth College, Computer Science, 2004.

[27] Lukáš, J., Fridrich, J., and Goljan, M.: "Detecting Digital Image Forgeries Using Sensor Pattern Noise." *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, vol. 6072. San Jose, California, pp. 0Y1–0Y11, 2006.

[28] Healey, G. and Kondepudy, R.: "Radiometric CCD Camera Calibration and Noise Estimation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16(3), pp. 267–276, March, 1994.

[29] Holst, G. C.: *CCD Arrays, Cameras, and Displays*, 2nd edition, JCD Publishing & SPIE Pres, USA, 1998.

[30] Janesick, J.R.: *Scientific Charge-Coupled Devices*, SPIE PRESS Monograph vol. PM83, SPIE–The International Society for Optical Engineering, January, 2001.

[31] Mihcak, M.K., Kozintsev, I., and Ramchandran, K.: "Spatially Adaptive Statistical Modeling of Wavelet Image Coefficients and its Application to Denoising." *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Phoenix, AZ, vol. 6, pp. 3253–3256, March 1999.

[32] El Gamal, A., Fowler, B. Min, H., and Liu, X.: "Modeling and Estimation of FPN Components in CMOS Image Sensors." *Proc. SPIE*, *Solid State Sensor Arrays: Development and Applications II*, vol. 3301-20, San Jose, CA, pp. 168–177, January 1998.

[33] Chen, M., Fridrich, J., and Goljan, M.: "Source Digital Camcorder Identification Using CCD Photo Response Non-uniformity." *Proc. SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, San Jose, California, January 28 – February 1, pp. 1G–1H, 2007.

[34] Kay, S.M.: *Fundamentals of Statistical Signal Processing*, Volume II, Detection theory, Prentice Hall, 1998.

[35] Webb, A.R.: *Statistical Pattern Recognition* (Second Edition), John Wiley & Sons, LTD, 2002.

[36] Chen, M., Fridrich, J., Goljan, M., and Lukáš, J.: "Digital Imaging Sensor Identification (Further Study)," *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, San Jose, CA, January 28–February 2, pp. 0P–0Q, 2007.

[37] Meignen, S. and Meignen, H.: "On the Modeling of DCT and Subband Image Data for Compression." *IEEE Trans. on Image Processing*, vol. 4, pp. 186–193, 1995.

[38] Bayram, S., Avcibas, I., Sankur, B. and Memon, N.: "Image Manipulation Detection," *Journal of Electronic Imaging*, vol. 15(4), 041102, 2006.

[39] Costagli, M. and Kuruoglu, E.: "Image Separation Using Particle Filter," Digital Signal Processing, 2007.