Apple's Synthetic Defocus Noise Pattern: Characterization and Forensic Applications

David Vázquez-Padín, *Member, IEEE,* Fernando Pérez-González, *Fellow, IEEE,* and Pablo Pérez-Miguélez

***Abstract*—iPhone portrait-mode images contain a distinctive pattern in out-of-focus regions simulating the bokeh effect, which we term Apple's *Synthetic Defocus Noise Pattern* (SDNP). If overlooked, this pattern can interfere with blind forensic analyses, especially PRNU-based camera source verification, as noted in earlier works. Since Apple's SDNP remains underexplored, we provide a detailed characterization, proposing a method for its precise estimation, modeling its dependence on scene brightness, ISO settings, and other factors. Leveraging this characterization, we explore forensic applications of the SDNP, including traceability of portrait-mode images across iPhone models and iOS versions in open-set scenarios, assessing its robustness under post-processing. Furthermore, we show that masking SDNP-affected regions in PRNU-based camera source verification significantly reduces false positives, overcoming a critical limitation in camera attribution, and improving state-of-the-art techniques.**

***Index Terms*—Image forensics, source camera verification, PRNU, portrait mode, depth, computational imaging, Apple.**

I. Introduction

THE widespread adoption of computational photography in modern smartphones, especially with the integration of Artificial Intelligence (AI) algorithms, poses considerable challenges for classic methods in multimedia forensics, particularly in the area of image forensics. Examples of computational imaging techniques include image stitching to create panoramic photos, High Dynamic Range (HDR) imaging to extend the camera's native luminosity range in a single shot, "night mode" to enhance low-light imaging, and "portrait mode" for simulating a shallow depth of field and mimic a realistic bokeh. Digital images produced through any of these computational imaging modes undergo complex processing chains, some involving black-box systems. These chains deviate significantly from well-established image acquisition models that enable source camera verification (also known as attribution) using Photo Response Non-Uniformity (PRNU), detect image manipulations by analyzing demosaicing patterns and/or JPEG compression traces, and more.

Although computational imaging modes are not inherently malicious, Iuliani *et al.* [1] showed that Apple's portrait mode can mislead forensic analysis, causing false positives in camera source attribution. Baracchi *et al.* [2] previously found traditional PRNU-based methods [3] ineffective for iPhone X portrait images and proposed using depth maps to remove the so-called Non-Unique Artifacts (NUAs). Similar issues were later observed in Samsung and Huawei devices in [4], with partial mitigation via a SPAM classifier [5]. More recently, misattributions in other devices were further studied in [6].

Both [4] and [5] recommend isolating images generated with computational imaging modes, which can be done using McCloskey's method [7] to detect focus manipulations like portrait mode. However, forensic practitioners may still face cases where only portrait images are available. As explored in this paper, forensic identification and characterization of computational imaging patterns can help not only recognize processed images, but also isolate affected regions within an image to mitigate errors in camera source attribution and detect inconsistencies where known patterns have been embedded.

Inspired by the call to action in [1], we have chosen to thoroughly investigate Apple's portrait mode. This mirrors the approach taken by Butora and Bas in [8], who examined the pattern introduced by Adobe in the development of raw or 16-bit images, which resulted in PRNU collisions. By modeling the so-called Adobe pattern, they significantly reduced collisions and later proposed a method to locally detect the presence of this pattern (at a resolution of 128×128 pixels) across an entire image in [9]. Our focus on Apple's portrait mode is motivated by Apple's position as a leading smartphone manufacturer, consistently ranking among the top sellers since 2022 [10] and leading smartphone sales with the iPhone 15 (released in September 2023) since Q4 2023 [11].

Our work provides an in-depth characterization of the pattern embedded by Apple in portrait-mode images, first exposed in [2], which we term the Synthetic Defocus Noise Pattern (SDNP), offering tools and insights to effectively handle portrait images in forensic applications. Specifically, our contributions include:

- Methods for SDNP extraction using two lighting modes.
- Characterization of SDNP's dependence on scene luminance and ISO settings.
- Tracking of SDNP variations across different image resolutions, iPhone models, and iOS versions.
- Leveraging extracted SDNPs to reduce the PRNU collisions noted in [1] and improve state-of-the-art tools [2].
- Assessing SDNP detection robustness under complex post-processing conditions, such as image sharing via WhatsApp.

The paper is organized as follows: Sect. I-A introduces the notation, followed by a review of PRNU-based source verification and Apple's portrait mode in Sect. II. SDNP extraction and characterization are covered in Sects. III and IV, with pattern variations discussed in Sect. V. Sect. VI presents forensic applications, and experimental results are reported in Sect. VII. Conclusions are drawn in Sect. VIII.

A. Notation

Matrices are denoted by bold uppercase letters, while regular (non-bold) letters are used for scalars. Unless otherwise stated, matrices are real-valued with dimensions $H \times W$. The (i,j)th element of a matrix $\mathbf{A}$ is written as $A_{i,j}$, where $0 \leq i \leq H-1$ and $0 \leq j \leq W-1$. The total number of elements of $\mathbf{A}$ is $N \triangleq H \cdot W$. The Frobenius inner product of matrices $\mathbf{A}$ and $\mathbf{B}$ is $\langle \mathbf{A}, \mathbf{B} \rangle_{\text{F}} \triangleq \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} A_{i,j} B_{i,j}$, and the Frobenius norm of $\mathbf{A}$ is $\|\mathbf{A}\|_{\text{F}} \triangleq \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle_{\text{F}}}$. The Hadamard product (i.e., element-wise product) of $\mathbf{A}$ and $\mathbf{B}$, denoted $\mathbf{A} \circ \mathbf{B}$, results in a matrix of the same dimension as the operands, with elements $(\mathbf{A} \circ \mathbf{B})_{i,j} = A_{i,j} \cdot B_{i,j}$. The Hadamard inverse of $\mathbf{A}$, denoted $\mathbf{A}^{\circ -1}$, is defined element-wise as $\left(\mathbf{A}^{\circ -1}\right)_{i,j} = A_{i,j}^{-1}$. Scalar-valued functions are denoted by lowercase letters, e.g., $f(\cdot)$, while uppercase letters, e.g., $F(\cdot)$, are used for matrix-valued functions. The sample mean of $\mathbf{A}$ is defined as $\mu(\mathbf{A}) \triangleq \langle \mathbf{A}, \mathbf{1} \rangle_{\text{F}} / N$, where $\mathbf{1}$ is an $H \times W$ matrix of ones. The sample standard deviation of $\mathbf{A}$ is defined as $\sigma(\mathbf{A}) \triangleq \|\mathbf{A} - \mu(\mathbf{A})\|_{\text{F}} / \sqrt{N-1}$. Along the paper, we use the Normalized Cross-Correlation (NCC) between two matrices $\mathbf{A}$ and $\mathbf{B}$, defined as:

$$\rho(\mathbf{A}, \mathbf{B}) \triangleq \frac{\langle \mathbf{A} - \mu(\mathbf{A}), \mathbf{B} - \mu(\mathbf{B}) \rangle_{\text{F}}}{\|\mathbf{A} - \mu(\mathbf{A})\|_{\text{F}} \|\mathbf{B} - \mu(\mathbf{B})\|_{\text{F}}}. \tag{1}$$

II. Preliminaries

This section outlines the key operations involved in PRNU-based camera source verification and explains Apple's portrait shooting mode, highlighting its impact on the PRNU.

A. PRNU-based Camera Source Verification

Lukáš *et al.* [12] pioneered the use of camera sensor noise patterns, particularly the PRNU, to identify the specific camera that captured an image. To understand how the PRNU, which arises from tiny imperfections in the camera sensor, is used for camera source verification, we first consider the assumed sensor output model. For a single-channel image, denoted by a matrix $\mathbf{Y}$, the sensor output model can be approximated by the first two terms of its Taylor series [13], as

$$\mathbf{Y} = (\mathbf{1} + \mathbf{K}) \circ \mathbf{X} + \boldsymbol{\Theta}, \tag{2}$$

where $\mathbf{K}$ is the PRNU signal, $\mathbf{X}$ is the incident light intensity and $\boldsymbol{\Theta}$ represents other noise sources.

1) Baseline PRNU fingerprint extraction: Based on the model in (2), which links the sensor output to the PRNU, a standard procedure has been established to estimate the PRNU of a given camera. This involves capturing a set of L native-resolution images, i.e., $\{\mathbf{Y}_l\}_{l=1}^{L}$. Since the incident light intensity $\mathbf{X}$ in (2) is unknown, it is approximated via a denoising operation $F(\cdot)$ (here, we use the filter from [14]), yielding $\hat{\mathbf{X}} = F(\mathbf{Y})$. The resulting residue $\mathbf{W}_l = \mathbf{Y}_l - F(\mathbf{Y}_l)$ for each image $\mathbf{Y}_l$ is used to estimate the PRNU through the Maximum Likelihood Estimator (MLE) from [13]:

$$\hat{\mathbf{K}} = \left(\sum_{l=1}^{L} \mathbf{W}_l \circ \mathbf{Y}_l \right) \circ \left(\sum_{l=1}^{L} \mathbf{Y}_l \circ \mathbf{Y}_l \right)^{\circ -1}. \tag{3}$$

This procedure assumes all images $\mathbf{Y}_l$ are spatially aligned, with no geometric transformations and with their underlying PRNU patterns consistent pixel by pixel. Use of flat-field images is recommended to minimize content leakage [15].

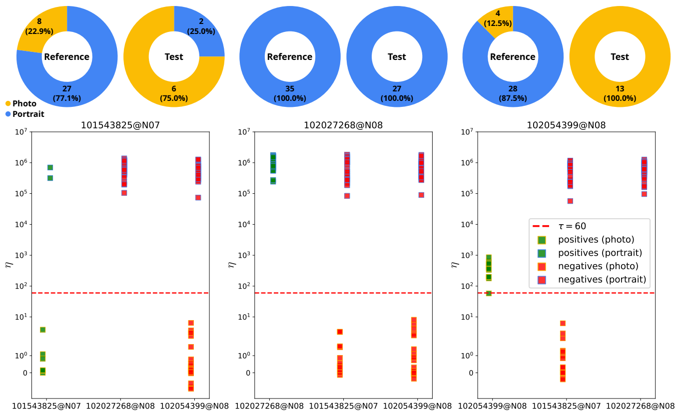


Fig. 1. Values of η for the 3 iPhone 11 Pro users with PRNU collisions in [1]. From left to right, each user's PRNU fingerprint is tested against the other two. The upper pie charts show the distribution of Reference and Test images per user, indicating the proportion of `Portrait` and `Photo` images.

2) Baseline PRNU detection: Given a test image $\mathbf{Y}_t$ with residue $\mathbf{W}_t = \mathbf{Y}_t - F(\mathbf{Y}_t)$ and a PRNU estimate $\hat{\mathbf{K}}$, the following hypothesis testing problem is formulated:

$$\mathcal{H}_0 : \mathbf{W}_t \text{ does not contain the PRNU } \mathbf{K},$$
$$\mathcal{H}_1 : \mathbf{W}_t \text{ contains the PRNU } \mathbf{K}.$$

If $\mathcal{H}_1$ holds, the test image $\mathbf{Y}_t$ likely originates from the camera with PRNU $\mathbf{K}$. In image forensics, this decision is typically made using the Peak-to-Correlation Energy (PCE) statistic, which was validated in [16] on a dataset of over a million images from 6,896 cameras (150 models). A threshold of 60 yielded a false alarm rate of 10^{-5} and a detection rate of 97.62%. The PCE is calculated over multiple shifts to account for potential sensor misalignment. However, assuming spatial alignment and no cropping, we adopt a less computationally intensive alternative: the similarity measure η inspired by the one proposed in [17], defined in terms of the NCC in (1) as:

$$\eta \triangleq N \cdot \text{ssq}\left(\rho(\mathbf{W}_t, \hat{\mathbf{K}} \circ \mathbf{Y}_t) \right), \tag{4}$$

where the signed-squared function $\text{ssq}(\cdot)$ is defined as $\text{ssq}(x) \triangleq \text{sgn}(x) \cdot x^2$, and $\text{sgn}(\cdot)$ is a sign function that returns -1, $+1$, or 0 depending on whether the input is negative, positive, or zero, respectively. Using this similarity measure, the hypothesis test can be performed by evaluating:

$$\eta \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \tau,$$

where τ is a fixed threshold determined by the desired false positive probability. While η differs from the PCE, its values are generally comparable, with the added advantage of being computationally more efficient.

Our baseline PRNU matching approach, while using a slightly different PCE metric than Iuliani *et al.* in [1], successfully reproduces their results for the 3 iPhone 11 Pro users that yield PRNU collisions, as presented in Fig. 1.[1] Iuliani *et al.* suggest that images captured in Apple's portrait mode

[1] To display both positive and negative values on a logarithmic scale, we use a symmetric logarithmic representation (concretely, `symlog` from matplotlib).

might contribute to user/device mismatches, consistent with findings from prior studies [2] and [4]. To further investigate this hypothesis, we analyze the η values obtained for images taken in `Portrait` and `Photo` shooting modes, which are depicted in Fig. 1 using blue and yellow colors, respectively.

Recall first that in the dataset from [1], each user/device has "Reference" and "Test" image sets, with PRNU fingerprints extracted solely from the Reference set and positive samples taken from the corresponding Test set. For cross-user tests, all images from both sets are used. Notably, user `102027268@N08` has only portrait images, and false positives for this user occur exclusively in that mode (see middle plot of Fig. 1). This suggests that Apple's portrait mode embeds a distinct pattern (here referred to as SDNP) that correlates more strongly than the PRNU, consistent with prior findings [2]. As all Reference sets contain portrait images, PRNU fingerprints are influenced by the SDNP, leading to false positives when testing portrait images. Conversely, correct rejections occur only in standard `Photo` mode, where the SDNP is absent. A detailed analysis of the other two users is provided in the technical report [18, Sect. 1]. These results demonstrate that adjusting the detection threshold alone cannot resolve PRNU-based verification issues introduced by computational imaging techniques like the portrait mode.

In Sect. VII-B, these baseline results will be revisited after fully characterizing Apple's SDNP, beginning with the next section, which clarifies its origin.

B. Apple Portrait Mode Description

Standard smartphone cameras, due to their short focal lengths and small apertures, naturally produce images with a large depth of field, keeping most elements in focus. This limits their ability to achieve the aesthetically pleasing background blur, or bokeh, seen in professional photography. However, advances in computational photography have enabled software-based solutions that simulate a shallow depth of field. HTC pioneered this approach in 2014 with their "Portrait Mode," and Apple popularized it two years later in the iPhone 7 Plus. Google also contributed by introducing innovative software-based depth sensing techniques rather than dual-camera hardware [19]. Today, most smartphone manufacturers offer a portrait mode, which introduces new forensic challenges for camera source attribution, as discussed in [4].

Apple's portrait mode implementation remains proprietary, but previous research by Baracchi *et al.* [2] has shed light on its behavior. Since then, the technology has evolved, prompting us to update the current understanding based on the latest iOS 17.5.1, focusing on iPhone models we tested directly (iPhone 15 and iPhone 12 mini).[2] Initially introduced with the iPhone 7 Plus, portrait mode is now available on models from the iPhone 8 Plus and iPhone SE (2nd generation) onward, as well as the iPhone X and later. With the iPhone 15, Apple introduced the ability to capture a regular photo and later convert it into a portrait in the `Photos` app if a person, dog, or cat is detected. Some iPhone models offer multiple zoom options

[2]This study excludes iPad devices, though recent models support front-camera portrait mode. Future work will explore this, expecting similar results.

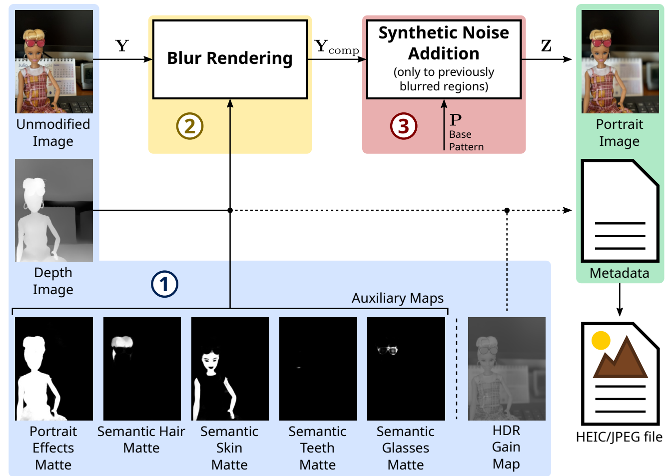


Fig. 2. Apple's portrait mode block diagram highlighting the 3 main stages.

(e.g., 1x and 2x), while older models like the iPhone XR and iPhone SE (2nd generation) require face detection to enable portrait mode with the rear camera. On the iPhone 15, users can also adjust zoom by pinching the screen. The standard resolution for portrait mode photos is 12MP (4032×3024) for rear cameras and 7MP (3088×2316) for front cameras,[3] while the iPhone 15 introduces a new maximum resolution of 24MP (5712×4284). Since iOS 11 (2017), Apple has supported saving images in the High Efficiency Image File (HEIF) format [20], with JPEG remaining an option for broader compatibility.

To gain a general understanding of how software-based portrait modes function, the reader is referred to [19]. Based on our analysis of Apple's implementation, we identified three main stages, as depicted in the block diagram in Fig. 2:

1) Original Capture and Depth Map Generation: Apple's portrait mode first captures an unmodified image, denoted as $\mathbf{Y}$, which corresponds to the model in (2). This original image can be retrieved from `iCloud Photos` or by disabling the portrait effect in the Photos app. Simultaneously, a depth map is generated using one of three methods [21]: (1) analyze disparities between multiple cameras (e.g., wide and telephoto); (2) apply machine learning to a single camera, particularly for human faces on select iPhone models; or (3) leverage specialized depth sensors like TrueDepth or LiDAR. However, these depth maps are lower resolution than the original image (see [18, Tab. 1]), limiting the precision of depth-based effects.

To enhance depth accuracy, iOS 12 introduced the depth-guided Portrait Effects Matte [22], a segmentation mask specifically designed for human subjects using a proprietary neural network. This matte refines subject-background separation, improving depth effects. In iOS 13, Apple extended semantic mattes to include details like hair, skin, teeth, and glasses (as shown in Fig. 2).[4] These mattes, along with the depth map, are embedded in the image file and can be accessed using the open-source `libheif` library [23]. To the best of our knowledge, Apple employs a proprietary algorithm to

[3]Some older models have a slightly different resolution: 3088×2320.

[4]While Fig. 2 includes an HDR gain map for reference, we found no direct connection to portrait mode and do not discuss it further in this paper.

integrate the depth map with auxiliary mattes, determining which areas will be blurred in the subsequent stage.

2) Blur Rendering: Apple patents, such as [24], describe potential methods for generating images with varying background blur levels, as documented in [2]. Using the depth data from the previous stage, the original image $\mathbf{Y}$ is re-rendered with different degrees of blur applied based on pixel depth, the simulated aperture, and the relative position to the focal plane. The focal plane, which determines which pixels remain sharp, can be selected in the viewfinder, while the simulated aperture is adjustable via the `Depth Control` button, ranging from f/1.4 to f/16. The resulting composite image, denoted as $\mathbf{Y}_{\text{comp}}$, is expressed as:

$$\mathbf{Y}_{\text{comp}} = \mathbf{M}'_{(\text{blur})} \circ \mathbf{Y} + \mathbf{M}_{(\text{blur})} \circ \mathbf{Y}', \tag{5}$$

where $\mathbf{M}_{(\text{blur})}$ is a binary mask with 1's in pixel positions where the blur has been applied and 0 elsewhere, $\mathbf{M}'_{(\text{blur})}$ is its logical negation, and $\mathbf{Y}' \triangleq F_{\text{blur}}^{\text{f-stop}}(\mathbf{Y})$ represents the blurred version of $\mathbf{Y}$ based on the simulated aperture defined by the f-stop number. While the specifics of the blur rendering process are beyond the scope of this paper, different smartphone manufacturers define the shape of defocused areas at this stage (with some, like Samsung, allowing post-capture shape adjustments). Apple appears to use a circular bokeh effect.

At this stage, the composite image $\mathbf{Y}_{\text{comp}}$ may exhibit inconsistencies in noise levels, as applying algorithmic blur reduces natural noise in defocused regions compared to sharp areas. To achieve a more realistic depth-of-field effect, synthetic noise must be added, as discussed in the next section.

3) Synthetic Noise Addition: The final step in generating the portrait image involves adding synthetic noise to the blurred regions of $\mathbf{Y}_{\text{comp}}$ (i.e., $\mathbf{Y}'$) to create a realistic optical blur and minimize artifacts at the transitions between sharp and blurred areas (cf. [19, Sect. 5.3]). For iPhone devices, we model the final portrait image $\mathbf{Z}$ as:

$$\begin{aligned}\mathbf{Z} &= \mathbf{Y}_{\text{comp}} + \mathbf{M}_{(\text{blur})} \circ \left(\gamma_{\text{ISO}} \cdot G\left(\mathbf{Y}'\right) \circ \mathbf{P} + \mathbf{\Phi}\right) \\ &= \mathbf{M}'_{(\text{blur})} \circ \mathbf{Y} + \mathbf{M}_{(\text{blur})} \circ \left(\mathbf{Y}' + \gamma_{\text{ISO}} \cdot G\left(\mathbf{Y}'\right) \circ \mathbf{P} + \mathbf{\Phi}\right),\end{aligned} \tag{6}$$

where (5) is applied in the second equality. This model implies that unblurred regions of $\mathbf{Z}$ retain the sensor output characteristics defined in (2), preserving the PRNU, while blurred regions do not reliably exhibit PRNU traces due to low-pass filtering and the addition of Apple's SDNP. The term $\mathbf{\Phi}$ is used to model various sources of noise resulting from full-frame processes in blurred areas such as compression, clipping, and other operations (including remaining traces of the original PRNU).

The added SDNP, modeled as $\mathbf{N} \triangleq \gamma_{\text{ISO}} \cdot G\left(\mathbf{Y}'\right) \circ \mathbf{P}$, is a content-dependent pattern derived from a *Base Pattern* (BP) $\mathbf{P}$. Although this BP is fixed for a given iPhone and iOS version (see Sect. V for details), we assume it can be modeled as a realization of a wide-sense stationary random process with mean $\mu_{\mathbf{P}} = 0$ (as empirically confirmed in Sect. III-B) and standard deviation (std) $\sigma_{\mathbf{P}} = 1$. This unit-std assumption is made for convenience as any scaling of the std could be absorbed into either γ_{ISO} or $G\left(\mathbf{Y}'\right)$.

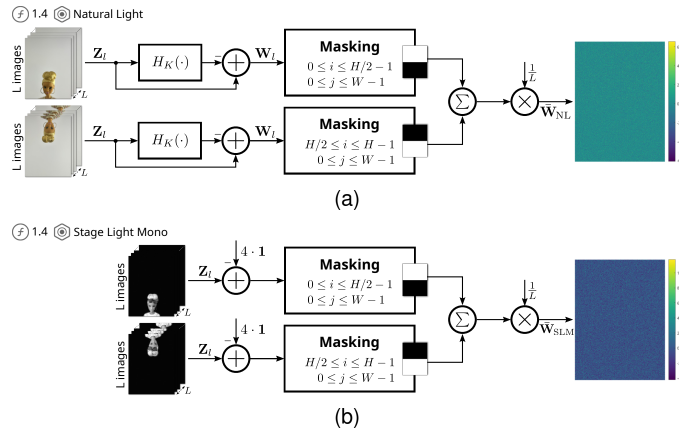


Fig. 3. Block diagram of the extraction process for the averaged residual matrix in each portrait lighting mode under study: $\bar{\mathbf{W}}_{\text{NL}}$ (a) and $\bar{\mathbf{W}}_{\text{SLM}}$ (b).

This normalized BP is then adapted to the content of the captured scene by two scaling operations: 1) multiplication by an ISO-dependent scaling factor, γ_{ISO} (further characterized in Sect. IV-A); and 2) element-wise multiplication by a brightness-dependent matrix $G\left(\mathbf{Y}'\right)$ with the same dimensions as $\mathbf{Y}'$. We model the (i,j)th element of this matrix as $\left[G\left(\mathbf{Y}'\right)\right]_{i,j} = g\left(D_{i,j}\left(\mathbf{Y}'\right)\right)$, where $g : \mathbb{R} \to \mathbb{R}$ is a function that will be determined in Sect. IV-B, and we assume that operator $D_{i,j}(\mathbf{Y}')$ computes a local (possibly weighted) average of pixels of $\mathbf{Y}'$ in a local neighborhood around (i,j).

The base pattern $\mathbf{P}$, extracted as detailed in Sect. III, is the core component of the SDNP, acting as a fingerprint of Apple portrait images. Its presence not only explains the PRNU collisions observed in Apple portrait images (see Fig. 1), but also provides a basis for developing new forensic applications, as further detailed in Sect. VI. The next two sections validate this model and describe procedures for estimating $\mathbf{P}$.

III. BP Extraction from Apple Portrait Images

Following a PRNU-like approach, we outline a procedure to extract the BP fingerprint $\mathbf{P}$ from Apple portrait images. Unlike PRNU estimation, flat-field images are unsuitable here, since a foreground subject is always required. Instead, we use "flat-background" images with a uniform background to ensure a consistent BP application. Because a foreground subject is required, extracting the full-resolution BP involves capturing two scenes with alternating foreground and background regions: in the first scene, the subject is positioned in the lower third of the frame, ensuring a uniform background in the upper half; in the second scene, the subject is placed in the upper third, ensuring a uniform background in the lower half. Fig. 3 illustrates this process for the two extraction methods discussed below.

A. BP extraction with Portrait Lighting: Natural Light

For a given iPhone under investigation, this extraction process involves selecting the default `Natural Light` (NL) portrait effect and capturing $2L$ portrait images across the previously described scenes. This includes L "top" images with a uniform background in the upper half and L "bottom" images with a uniform background in the lower half. Each set is processed separately, and the results are aggregated to form an averaged residual matrix, $\bar{\mathbf{W}}_{\mathrm{NL}}$, which serves as the basis for estimating the BP.

We first describe how to obtain the "top" half of $\bar{\mathbf{W}}_{\mathrm{NL}}$, i.e., $\left(\bar{\mathbf{W}}_{\mathrm{NL}}\right)_{i,j}$ for $0 \leq i \leq H/2-1$ and $0 \leq j \leq W-1$. For each of the L "top" images, denoted as $\mathbf{Z}_l$ with $l = 1, \ldots, L$, a denoising filter is applied to the luminance component[5] to extract the noise residuals:

$$\mathbf{W}_l = \mathbf{Z}_l - H_K\left(\mathbf{Z}_l\right), \tag{7}$$

where $H_K(\cdot)$ represents the denoising operation. Unlike the denoising method used for PRNU extraction (i.e., [14]), we employ a simple linear filter that convolves $\mathbf{Z}_l$ with a normalized $K \times K$ box kernel, due to the uniform background in the region of interest of the captured images. The choice of K for BP extraction from these "flat-background" images is discussed in the technical report [18, Sect. 2.1]. The top half of $\bar{\mathbf{W}}_{\mathrm{NL}}$ is then obtained by averaging the corresponding portions of the residuals as follows:

$$\bar{\mathbf{W}}_{\mathrm{NL}} \triangleq \frac{1}{L}\sum_{l=1}^{L}\mathbf{W}_l. \tag{8}$$

Similarly, after denoising the L "bottom" images as in (7), the "bottom" half of $\bar{\mathbf{W}}_{\mathrm{NL}}$, specifically the elements $\left(\bar{\mathbf{W}}_{\mathrm{NL}}\right)_{i,j}$ for $H/2 \leq i \leq H-1$ and $0 \leq j \leq W-1$, is obtained by averaging the corresponding bottom portions of the residuals, using the same averaging process as in (8).

The preceding procedure for obtaining $\bar{\mathbf{W}}_{\mathrm{NL}}$ (illustrated in Fig. 3a) does not directly estimate $\mathbf{P}$ in (6) due to the scaling operators γ_{ISO} and $G\left(\mathbf{Y}'\right)$. However, the uniform nature of the blurred "flat-background" regions enables two key approximations, allowing us to estimate $\mathbf{P}$ up to a constant scaling factor. First, assuming both a zero-mean BP (discussed further below) and a zero-mean noise component $\mathbf{\Phi}$ in (6), we can approximate $H_K\left(\mathbf{Z}_l\right) \approx \mathbf{Y}'_l$ in these smooth regions. Second, the scaling function can be approximated as $G\left(\mathbf{Y}'_l\right) \approx G\left(\mu\left(\mathbf{Y}'_l\right)\cdot\mathbf{1}\right)$, i.e., $\left[G\left(\mathbf{Y}'_l\right)\right]_{i,j} = g\left(D_{i,j}\left(\mathbf{Y}'\right)\right) \approx g\left(\mu\left(\mathbf{Y}'_l\right)\right), \forall i,j$. This approximation is a direct consequence of the assumed nature of operator $D_{i,j}(\cdot)$ and the fact that $\mathbf{Y}'_l$ is approximately constant. Notice that for clarity and brevity, we will adopt the shorthand notation $y' \triangleq \mu\left(\mathbf{Y}'\right)$ (or similarly, $y'_l \triangleq \mu\left(\mathbf{Y}'_l\right)$), especially when $\mu(\cdot)$ is an argument of the function $g(\cdot)$. Consequently, for this scenario, (7) simplifies to:

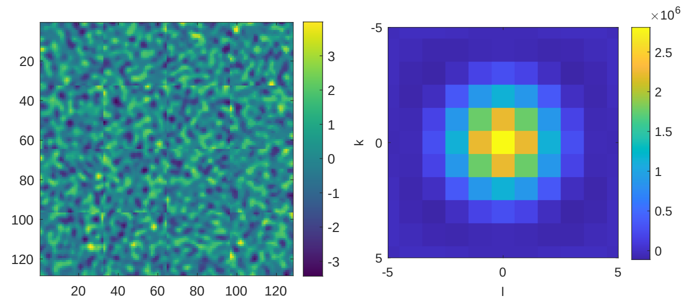
$$\mathbf{W}_l = \gamma_{\mathrm{ISO}}\cdot g(y'_l)\cdot\mathbf{P} + \mathbf{\Psi}_l, \tag{9}$$

where $\mathbf{\Psi}_l$ comprises $\mathbf{\Phi}_l$ and errors from the denoising process. Provided that the same ISO value is used for all $2L$ captured images (i.e., γ_{ISO} is constant), it follows from (8) and (9) that:

$$\bar{\mathbf{W}}_{\mathrm{NL}} = \gamma_{\mathrm{ISO}}\cdot\bar{\lambda}_{\mathrm{NL}}\cdot\mathbf{P} + \bar{\mathbf{\Psi}}_{\mathrm{NL}}, \tag{10}$$

where $\bar{\lambda}_{\mathrm{NL}} \triangleq \frac{1}{L}\sum_{l=1}^{L} g(y'_l)$ represents the average value of the scaling function across the set of images and $\bar{\mathbf{\Psi}}_{\mathrm{NL}} \triangleq \frac{1}{L}\sum_{l=1}^{L}\mathbf{\Psi}_l$ is the estimation noise. To simplify the analysis, we will assume that the noise components $\mathbf{\Psi}_l$ correspond to independent realizations of a wide-sense stationary and ergodic process. One consequence of this assumption is that $\lim_{L\to\infty}\|\bar{\mathbf{\Psi}}_{\mathrm{NL}}\|_{\mathrm{F}} = 0$. Notice that in (9) and (10) we should write $\left(\mathbf{P} - \mu\left(\mathbf{P}\right)\right)$ instead of $\mathbf{P}$, because the denoising process we use (i.e., $H_K(\cdot)$) removes the mean of $\mathbf{P}$. However, as we will confirm in Sect. III-B, $\mu\left(\mathbf{P}\right) \approx 0$, so it is reasonable and notationally simpler to keep (9) and (10) unchanged. From (10) and the assumption that $\mathbf{P}$ is a realization of a stochastic process with zero mean and unit std, it follows that the normalized averaged residual $\bar{\mathbf{W}}_{\mathrm{NL}}$ yields an estimate of $\mathbf{P}$, that is, $\hat{\mathbf{P}}_{\mathrm{NL}} = \bar{\mathbf{W}}_{\mathrm{NL}}/\sigma\left(\bar{\mathbf{W}}_{\mathrm{NL}}\right)$. Since the quality of $\hat{\mathbf{P}}_{\mathrm{NL}}$ depends on a well-curated $\bar{\mathbf{W}}_{\mathrm{NL}}$, we offer guidelines on background selection and capture settings in [18, Sect. 2.2].

[5]The rationale for using the luminance component is detailed in Sect. IV.



(a) ISO 125, $L = 110$ (b)

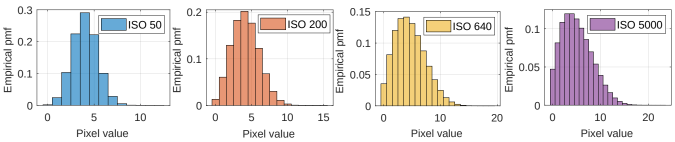
Fig. 4. Upper-left 128×128 patch of $\hat{\mathbf{P}}_{\mathrm{NL}}$ estimated from 12MP HEIF images on the iPhone 15 (a). Autocorrelation of $\hat{\mathbf{P}}_{\mathrm{NL}}$ limited to $k,l \in [-5,5]$ (b).

As an illustrative example, the upper-left 128×128 patch of $\hat{\mathbf{P}}_{\mathrm{NL}}$ is shown in Fig. 4a. Analyzing its properties further, the autocorrelation of $\hat{\mathbf{P}}_{\mathrm{NL}}$ (plotted in Fig. 4b for lags $k,l \in [-5,5]$) indicates a colored, rather than white, noise process. This observation points towards underlying low-pass filtering, consistent with the analysis in [18, Sect. 3.2]. For additional details on BP characteristics, see [18, Sect. 4.2]. Finally, note that although a Minimum Mean Square Error (MMSE) estimate of $\mathbf{P}$ is conceivable, it would require knowledge of the second-order statistics of $\bar{\mathbf{\Psi}}_{\mathrm{NL}}$, while offering only a performance gain that vanishes as L increases. Its computation and analysis are left for future work.

B. BP extraction with Portrait Lighting: Stage Light Mono

The `Stage Light Mono` (SLM) portrait effect offers a convenient alternative for estimating Apple's BP from an iPhone under study. This effect simulates black-and-white stage lighting by isolating the subject in focus against a uniformly black background (see Fig. 3b). Its key advantage is that it removes the need for a physically uniform background, as depth information ensures a constant background luminance of 4 (i.e., $\mathbf{Y}' = 4\cdot\mathbf{1}$ in (6)) before adding the SDNP (an observation empirically validated in the technical report [18, Sect. 3.1]). However, this low luminance level causes values below zero being clipped, leading to saturation and thus information loss in the estimation of $\mathbf{P}$.

Using a methodology analogous to the NL-based BP extraction in Sect. III-A, we capture $2L$ SLM portrait images, comprising L "top" images with a black upper background and L "bottom" images with a black lower background. Unlike



(a) ISO 50 (b) ISO 200 (c) ISO 640 (d) ISO 5000

Fig. 5. Pixel value distribution in $\mathbf{Z}_{\text{SLM}}$ for different ISO values.

the NL case, no filtering is needed to compute residuals, as $\mathbf{Y}'$ is already known. The "top" half of the basis $\bar{\mathbf{W}}_{\text{SLM}}$ to estimate the BP, i.e., $\left(\bar{\mathbf{W}}_{\text{SLM}}\right)_{i,j}$ for $0 \leq i \leq H/2-1$ and $0 \leq j \leq W-1$, is obtained by subtracting the constant luminance matrix $4 \cdot \mathbf{1}$ from each of the L "top" images, $\mathbf{Z}_l$, and averaging the resulting matrices:

$$\bar{\mathbf{W}}_{\text{SLM}} \triangleq \frac{1}{L}\sum_{l=1}^{L}\left(\mathbf{Z}_l - 4\cdot\mathbf{1}\right). \tag{11}$$

The "bottom" half of $\bar{\mathbf{W}}_{\text{SLM}}$, corresponding to elements $\left(\bar{\mathbf{W}}_{\text{SLM}}\right)_{i,j}$ for $H/2 \leq i \leq H-1$ and $0 \leq j \leq W-1$, is computed similarly by averaging the corresponding portions of the L "bottom" images, using the same approach as in (11).

In this scenario, since $\mathbf{Y}'$ is constant, the scaling function $G\left(\mathbf{Y}'\right)$ in (6) also yields a constant matrix. Thus, assuming a fixed ISO value is used across all $2L$ captured portrait images, our basis for estimating $\mathbf{P}$ is given, according to (11), by:

$$\bar{\mathbf{W}}_{\text{SLM}} = \gamma_{\text{ISO}} \cdot \lambda_{\text{SLM}} \cdot \mathbf{P} + \bar{\mathbf{\Psi}}_{\text{SLM}}, \tag{12}$$

where $\lambda_{\text{SLM}} \triangleq g(4)$ is a constant factor and $\bar{\mathbf{\Psi}}_{\text{SLM}} \triangleq \frac{1}{L}\sum_{l=1}^{L}\mathbf{\Phi}_l$ represents the average noise component, including clipping effects from pixels saturated below 0. Unlike in the NL case, the noise components $\mathbf{\Phi}_l$ can no longer be assumed independent, as they originate from a constant scene in $\mathbf{Y}'$ and the same ISO setting. Consequently, in the SLM case, $\|\bar{\mathbf{\Psi}}_{\text{SLM}}\|_{\text{F}}$ does not tend to zero as $L \to \infty$.

While, in principle, one could consider estimating $\mathbf{P}$ by normalizing $\bar{\mathbf{W}}_{\text{SLM}}$ as was done in the NL case (i.e., by computing $\bar{\mathbf{W}}_{\text{SLM}}/\sigma(\bar{\mathbf{W}}_{\text{SLM}})$), the fact that $\bar{\mathbf{\Psi}}_{\text{SLM}}$ does not vanish (and is significant relative to the term containing $\mathbf{P}$) would result in a considerably biased estimate. Instead, in the SLM case, it is preferable to estimate $\mathbf{P}$ by simply dividing $\bar{\mathbf{W}}_{\text{SLM}}$ in (12) by $\gamma_{\text{ISO}} \cdot \lambda_{\text{SLM}}$, where $\lambda_{\text{SLM}} = g(4)$. The next section discusses the estimation of γ_{ISO} and $g(\cdot)$.

IV. Estimation of BP's Scaling Factors

We now focus on the scaling operators γ_{ISO} and $G(\cdot)$ within the Apple portrait image model in (6), which modulate the base pattern $\mathbf{P}$ according to scene characteristics (as detailed in Sect. II-B3). Since $G(\cdot)$ yields a constant matrix under the SLM lighting effect, we use SLM portrait images to determine γ_{ISO} (Sect. IV-A). On the other hand, to determine $G(\cdot)$, which requires capturing a full range of luminance values, we rely on NL portrait images (Sect. IV-B).

A. Estimation of the ISO-Dependent Scaling Factor

As noted in Sect. III-B, the SLM-based BP extraction ensures a constant brightness-dependent scaling factor, i.e., $G\left(\mathbf{Y}'\right) = g(4)\cdot\mathbf{1}$. Since any gain can be absorbed into γ_{ISO}, we arbitrarily set $g(4) = 1$, making $\lambda_{\text{SLM}} = 1$ in (12). The effect of γ_{ISO} can be noticed in Fig. 5, which illustrates the empirical probability mass function (pmf) of $\mathbf{Z}_{\text{SLM}}$ (a submatrix of the background region of an SLM portrait image containing the SDNP) for ISO values of 50, 200, 640, and 5000. As ISO increases, so does the scaling factor γ_{ISO}, resulting in a higher occurrence of saturated pixels.

The empirical pmfs in Fig. 5 show that the right tail of the distribution, less affected by saturation, closely follows a Gaussian distribution. This approximation is confirmed by a Kolmogorov-Smirnov test (see [18, Sect. 3.2]). To exploit this observation, we conduct an exhaustive search to match the observed data distribution (i.e., the empirical pmf of $\mathbf{Z}_{\text{SLM}}$ for a given ISO) with synthetic Gaussian distributions. These distributions are generated using candidate mean and standard deviation values, simulating the effects of $\mu(\mathbf{P})$ and γ_{ISO}, respectively. Specifically, we compare the observed pmf of $\mathbf{Z}_{\text{SLM}}$ for a given ISO value with the distribution (parameterized by μ_Z, σ_Z) of the following random variable:

$$Z = \max\left\{0, \text{round}\left(\mu_Z + \sigma_Z P\right)\right\} \tag{13}$$

where P is a unit-variance Gaussian random variable, and $\text{round}(\cdot)$ represents rounding to the nearest integer, mimicking the 8-bit depth of SLM portrait images. We do not explicitly model the compression that occurs in practice to avoid increasing model complexity and complicating the exhaustive search, which is already performed over two parameters.

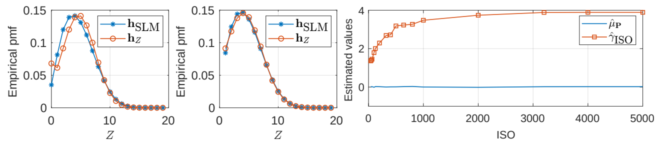
To find estimates for parameters μ_Z, σ_Z we minimize the Kullback-Leibler Divergence (KLD)[6] between the empirical pmf $\mathbf{h}_{\text{SLM}}$ of the observations $\mathbf{Z}_{\text{SLM}}$ (for a given ISO value), and the empirical pmf $\mathbf{h}_Z$ of i.i.d. samples of Z. In our experiments, the estimation process involves exploring a predefined grid of candidate pairs (μ_Z, σ_Z) that minimize the KLD, i.e.,

$$(\hat{\mu}_Z, \hat{\sigma}_Z) \triangleq \arg\min_{(\mu_Z,\sigma_Z)\in\mathcal{M}\times\mathcal{S}} \text{KLD}\left(\mathbf{h}_{\text{SLM}}, \mathbf{h}_Z\right),$$

where each candidate pair $(\mu_Z, \sigma_Z) \in \mathcal{M}\times\mathcal{S}$ is obtained by sampling the intervals $\mathcal{M} \triangleq [\mu(\mathbf{Z}_{\text{SLM}}) - 1, \mu(\mathbf{Z}_{\text{SLM}}) + 1]$ and $\mathcal{S} \triangleq [\sigma(\mathbf{Z}_{\text{SLM}}) - 1, \sigma(\mathbf{Z}_{\text{SLM}}) + 1]$ uniformly, with a step size of 0.1. Following (6) for the particular case $\mathbf{Y}' = 4\cdot\mathbf{1}$ and the assumption $g(4) = 1$, we can model the blurred samples of the SLM image prior to quantization and clipping as $4\cdot\mathbf{1} + \gamma_{\text{ISO}}\cdot\mathbf{P}$. On the other hand, we have just shown that these samples are well modeled by a Gaussian distribution with mean $\hat{\mu}_Z$ and std $\hat{\sigma}_Z$. Therefore, by simple identification, we can infer that $\hat{\gamma}_{\text{ISO}} = \hat{\sigma}_Z/\sigma_{\mathbf{P}} = \hat{\sigma}_Z$ (given our assumption that $\sigma_{\mathbf{P}} = 1$) and $\hat{\mu}_{\mathbf{P}} = (\hat{\mu}_Z - 4)/\hat{\gamma}_{\text{ISO}}$.

Lacking a suitable model for the compression operation, our initial estimate was biased by the peak at $Z = 0$ due to clipping, as shown in Fig. 6a for the particular case of ISO 500. Compression would typically spread these values across neighboring bins. To reduce this bias, we excluded zero-valued pixels from both $\mathbf{Z}_{\text{SLM}}$ and the synthetic variable Z when computing the KLD. This adjustment, illustrated in Fig. 6b, significantly improved the alignment of empirical

[6]We also tested the chi-square distance and obtained similar results.



(a) (b) (c)

Fig. 6. First two panels compare the empirical pmfs $\mathbf{h}_{\text{SLM}}$ and $\mathbf{h}_Z$ at ISO 500, for $Z \geq 0$ (a) and $Z > 0$ (b). The right panel (c) illustrates the estimated values of $\hat{\mu}_{\mathbf{P}}$ and $\hat{\gamma}_{\text{ISO}}$ across different ISO settings.

pmfs and yielded more reliable estimates. Additional details on the exhaustive search using KLD and Chi-square distance are in [18, Sect. 3.3].

Processing images at different ISO values yielded the results in Fig. 6c, showing that the mean of $\mathbf{P}$ remains close to zero regardless of ISO. On the other hand, the scaling factor γ_{ISO} increases with ISO but remains asymptotically bounded. A table listing the estimated $\hat{\gamma}_{\text{ISO}}$ values for various ISO settings is available in [18, Tab. 3]. With the knowledge of $\hat{\gamma}_{\text{ISO}}$, obtained under the assumption that $\lambda_{\text{SLM}} = 1$, we can now estimate the BP from SLM portrait images as described at the end of Sect. III-B, resulting in $\hat{\mathbf{P}}_{\text{SLM}} = \frac{1}{\hat{\gamma}_{\text{ISO}}}\bar{\mathbf{W}}_{\text{SLM}}$.

B. Estimation of the Brightness-Dependent Scaling Function

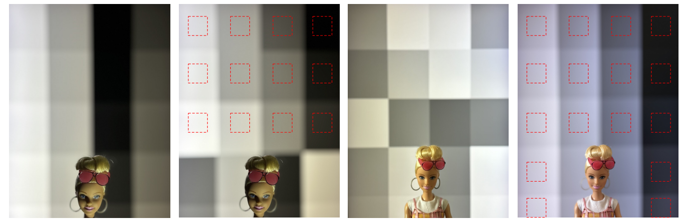
With the ISO-dependent scaling factor γ_{ISO} characterized, we now turn to the brightness-dependent scaling function $G(\mathbf{Y}')$. As mentioned previously, to capture the full luminance range (0 to 255 for 8-bit depth), we use NL portrait images with uniform patches. This allows us to approximate the scaling function as $G(\mathbf{Y}') \approx G(\mu(\mathbf{Y}') \cdot \mathbf{1})$, where $[G(\mathbf{Y}')]_{i,j} \approx g(y'), \forall i,j$ (recall that $y' \triangleq \mu(\mathbf{Y}')$). Thus, we characterize $G(\mathbf{Y}')$ through its element-wise form $g(y')$.

To estimate $g(\cdot)$, we collect M flat-background patches of size $B \times B$ containing the SDNP from each of T images captured at the same ISO, yielding a total of $M \cdot T$ patches. These patches are non-overlapping within each image, co-located across different images (i.e., they share the same coordinates across the T images), and span different background luminance levels (illustrated in Fig. 7). Let $\mathbf{Z}_{m,t}$ represent the mth patch from the tth image, with $\mathbf{Y}'_{m,t}$ and $\mathbf{W}_{m,t}$ similarly defined. $\mathbf{P}_m$ denotes the mth patch of the BP, which is co-located with $\mathbf{Z}_{m,t}$ and of the same size. Since $z_{m,t} \triangleq \mu(\mathbf{Z}_{m,t})$ serves as an estimate of $y'_{m,t} \triangleq \mu(\mathbf{Y}'_{m,t})$, our goal is to obtain pairs $\left(z_{m,t}, \hat{g}(y'_{m,t})\right)$ as estimates of $\left(y'_{m,t}, g(y'_{m,t})\right)$. By carefully selecting the background levels $z_{m,t}$, we ensure that the entire range $[0, 255]$ is well represented.

Given $\mathbf{Z}_{m,t}$ we select a co-located block $\mathbf{Z}_{m,u}$ from a different image, i.e., $u \neq t$, and work with the corresponding residuals $\mathbf{W}_{m,t}$, $\mathbf{W}_{m,u}$. First, we characterize their theoretical cross-correlation:

$$\begin{aligned} \mathrm{E}\left[\langle \mathbf{W}_{m,t}, \mathbf{W}_{m,u}, \rangle_{\mathrm{F}}\right]/B^2 &= \gamma_{\text{ISO}}^2\, g\left(y'_{m,t}\right) g\left(y'_{m,u}\right) \|\mathbf{P}_m\|_{\mathrm{F}}^2/B^2 \\ &\approx \gamma_{\text{ISO}}^2\, g\left(y'_{m,t}\right) g\left(y'_{m,u}\right), \end{aligned} \tag{14}$$

where we apply our assumption that the noise components $\mathbf{\Psi}_t$ and $\mathbf{\Psi}_u$ are mutually uncorrelated and uncorrelated with $\mathbf{P}$. The approximation in (14), which is asymptotically tight as the block size goes to infinity, comes from the assumption that



(a) ISO 400 (b) ISO 125

Fig. 7. Examples of portrait images of the ColorChecker scene captured at full resolution (24MP). The left panel (a) displays two samples from a set of $T = 86$ images taken at ISO 400, while the right panel (b) shows two samples from a set of $T = 64$ images taken at ISO 125. The processed $B \times B$ blocks ($B = 512$) are highlighted in red in the rightmost image of each case.

$\mathbf{P}$ can be modeled by a zero-mean, unit-variance stationary process, and the law of large numbers, which imply that $\lim_{B\to\infty} \|\mathbf{P}_m\|_{\mathrm{F}}^2/B^2 = 1$. Next, we compute the square root of the theoretical second moment of $\mathbf{W}_{m,u}$:

$$\begin{aligned} \left(\mathrm{E}\left[\|\mathbf{W}_{m,u}\|_{\mathrm{F}}^2/B^2\right]\right)^{1/2} &= \Big(\mathrm{E}\Big[\gamma_{\text{ISO}}^2\, g^2\left(y'_{m,u}\right) \|\mathbf{P}_m\|_{\mathrm{F}}^2/B^2 \\ &\qquad + \|\mathbf{\Psi}_{m,u}\|_{\mathrm{F}}^2/B^2\Big]\Big)^{1/2} \\ &\approx \gamma_{\text{ISO}}\, g\left(y'_{m,u}\right), \end{aligned} \tag{15}$$

where the approximation is based on $\lim_{B\to\infty} \|\mathbf{P}_m\|_{\mathrm{F}}^2/B^2 = 1$, as discussed above, and the assumption that the term due to $\mathbf{\Psi}_{m,u}$ is relatively small with respect to the SDNP. Combining (14) and (15), we obtain:

$$g\left(y'_{m,t}\right) \approx \frac{\mathrm{E}\left[\langle \mathbf{W}_{m,t}, \mathbf{W}_{m,u}\rangle_{\mathrm{F}}\right]}{\gamma_{\text{ISO}}\; B\left(\mathrm{E}\left[\|\mathbf{W}_{m,u}\|_{\mathrm{F}}^2\right]\right)^{1/2}}. \tag{16}$$

Now we can leverage (16) to obtain an estimate $\hat{g}(\cdot)$ of $g(\cdot)$: we replace the expectations by their sample estimates, and substitute γ_{ISO} by its estimate $\hat{\gamma}_{\text{ISO}}$ obtained as in Sect. IV-A for the ISO value used to capture the images. Recalling that $z_{m,t}$ is an estimate of $y'_{m,t}$, and repeating the process for all $m \in \{1, \ldots, M\}$ and $t, u \in \{1, \ldots, T\}$, $t \neq u$, we obtain $M \cdot T \cdot (T-1)$ pairs $(z_{m,t}, \hat{g}(y'_{m,t}))$.

We implemented this procedure by capturing NL portrait images of a ColorChecker-like board displayed on a screen behind the subject, creating uniform gray patches with the SDNP present in the background (see Fig. 7). A total of $T = 86$ images (captured at ISO 400) were taken using the iPhone 15 at 24MP resolution. From each image, we processed $M = 12$ co-located $B \times B$ patches, with $B = 512$ pixels (highlighted in red in the rightmost image of Fig. 7a), resulting in $M \cdot T = 1032$ distinct patches in total. The resulting empirical pairs $(z_{m,t}, \hat{g}(y'_{m,t}))$ are plotted separately for the three RGB channels and the luminance component of the YUV color space in Fig. 8. For better visualization, we display the range of values as a shaded region, and plot for each $z_{m,t}$ the average value of the $T-1$ estimates $\hat{g}(y'_{m,t})$ obtained for all $u = 1, \ldots, T$, $u \neq t$, using (16). This smoothed average curve (using a 5-point moving average) serves as an estimate of $g(\cdot)$.

The estimated scaling function behaves similarly across all four components, with the red and blue channels showing the most variability and luminance the least. Since RGB channels follow comparable trends, the BP is likely embedded primarily in the luminance component. Supporting this, the

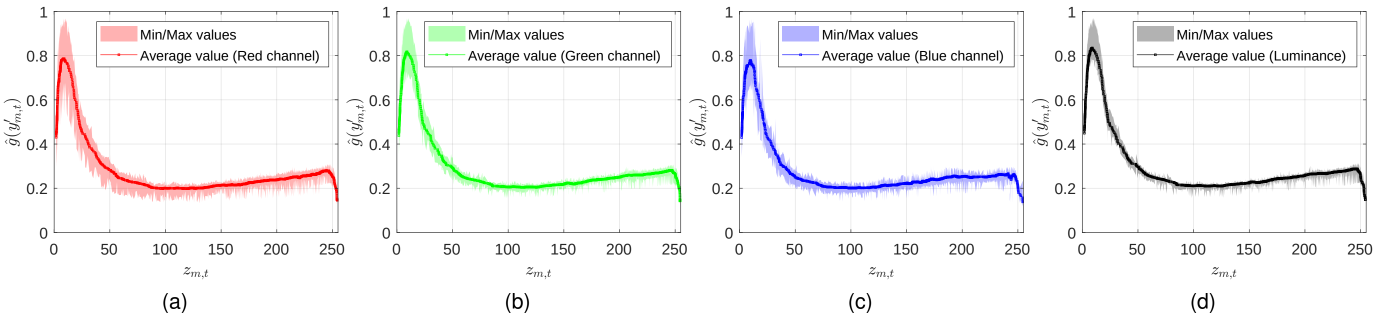


Fig. 8. Evolution of the estimated brightness-dependent scaling function $\hat{g}(y'_{m,t})$ as a function of $z_{m,t}$, based on co-located uniform patches captured at ISO 400. Results are presented for the R (a), G (b), and B (c) channels of the RGB colorspace, and the luminance component (d) of the YUV colorspace. Shaded regions indicate the range of estimated values, while the solid curve represents the smoothed average (5-point moving average) for each $z_{m,t}$.

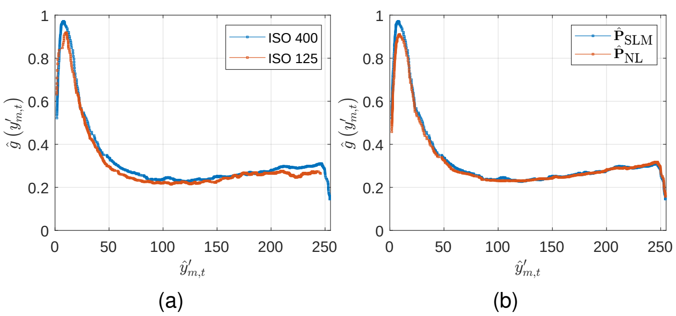


Fig. 9. Estimated pairs $(\hat{y}'_{m,t}, \hat{g}(y'_{m,t}))$ using the LS approach. Panel (a) compares results for $\hat{\mathbf{P}}_{\text{SLM}}$ at ISO 400 and 125. Panel (b) contrasts $\hat{\mathbf{P}}_{\text{SLM}}$ and $\hat{\mathbf{P}}_{\text{NL}}$ at ISO 400. All curves are smoothed using a 5-point moving average.

NCC between $\hat{\mathbf{P}}_{\text{NL}}$ (extracted from luminance) and each color channel was nearly 1: 0.990 (R), 0.995 (G), and 0.991 (B). While not conclusive, this justifies assuming that the BP is embedded in the luminance. Thus, we heretofore focus solely on the luminance component. If BP estimates are available, such as $\hat{\mathbf{P}}_{\text{NL}}$ or $\hat{\mathbf{P}}_{\text{SLM}}$, an alternative estimate of $g(\cdot)$ can be derived using a Least Squares (LS) approach. As detailed in Appendix A, the resulting estimate is:

$$\hat{g}\left(y'_{m,t}\right) = \frac{1}{\hat{\gamma}_{\text{ISO}}} \frac{\left\langle \mathbf{Z}_{m,t}, \hat{\mathbf{P}}_m \right\rangle_{\text{F}} - B^2 z_{m,t} \cdot \mu(\hat{\mathbf{P}}_m)}{\|\hat{\mathbf{P}}_m\|_{\text{F}}^2 - B^2 \left(\mu(\hat{\mathbf{P}}_m)\right)^2}, \quad (17)$$

where $\hat{\mathbf{P}}_m$ represents the mth $B \times B$ block of any available estimate of $\mathbf{P}$ (spatially aligned with $\mathbf{Z}_{m,t}$), and $\hat{\gamma}_{\text{ISO}}$ is the estimated scaling factor corresponding to the ISO value used to capture the observed blocks. Note that while we assume $\mu_{\mathbf{P}} = 0$, this does not hold for certain BP estimates, such as $\hat{\mathbf{P}}_{\text{SLM}}$, where $\mu(\hat{\mathbf{P}}_{\text{SLM}}) \neq 0$ due to saturation effects. In Appendix A we also derive the following estimate $\hat{y}'_{m,t} = z_{m,t} - \hat{\gamma}_{\text{ISO}} \cdot \hat{g}(y'_{m,t}) \cdot \mu(\hat{\mathbf{P}}_m)$, which becomes $z_{m,t}$ when $\mu(\hat{\mathbf{P}}_m) = 0$.

We plot the pairs $\left(\hat{y}'_{m,t}, \hat{g}\left(y'_{m,t}\right)\right)$ for $m = 1, \ldots, M$ and $t = 1, \ldots, T$ smoothed with a 5-point moving average as the final LS estimate. Fig. 9a shows the resulting curves obtained using $\hat{\mathbf{P}}_{\text{SLM}}$ as estimator of $\mathbf{P}$ for two different ISO values. For ISO 400, we use the aforementioned 1032 distinct blocks from $M = 12$ co-located blocks across $T = 86$ ColorChecker portrait images. Similarly, for ISO 125, we use 1072 blocks captured from $T = 67$ images and $M = 16$ co-located blocks (highlighted in red in the rightmost image of Fig. 7b). Although the obtained curves are not identical, their alignment after applying the corresponding $\hat{\gamma}_{\text{ISO}}$ values for ISO 125 and ISO 400 supports the consistency of the model assumed in (6). To compare the estimated scaling functions $\hat{g}(\cdot)$ derived from different BP estimates, we compute (17) using $\hat{\mathbf{P}}_{\text{SLM}}$ and $\hat{\mathbf{P}}_{\text{NL}}$ for the ISO 400 case. Fig. 9b shows that the resulting curves match almost perfectly. Finally, the near-identical curves in Figs. 8 and 9 (from various estimation methods and ISO values) demonstrate the convergence of different approaches and data to very similar brightness-dependent scaling function estimates, providing strong evidence for the model's validity.

Although not explored here, the estimated function $\hat{g}(\cdot)$ could be used to emphasize the extracted BP, as proposed in [25] for the PRNU, but this is left for future work. Meanwhile, after characterizing Apple's SDNP by estimating its scaling parameters and underlying BP, we analyzed key factors affecting BP estimate quality, including the number of images L, scene brightness, portrait lighting modes, and ISO settings. Details of this analysis are provided in [18, Sect. 4].

V. Analysis of Apple's BP Variations

Apple's BP in portrait images varies with resolution, iPhone model, and iOS version. While a detailed analysis of different resolutions, aspect ratios, and other BP characteristics is provided in [18, Sect. 4.1], here we focus on how the BP changes over time for specific iPhone/iOS combinations. As an example, although the BPs from the iPhones 15 and 12 mini (running the same iOS) share statistical properties, their patterns differ. The NCC map between their SLM-based BPs (computed as described below in Sect. VI-B, treating one BP as a residue) shows poor correlation in the right half (Fig.10a), except for a small patch in the upper-right corner. The presence of arc patterns suggests an algorithmic BP generation process with model-specific variations. Interestingly, like the iPhone 15 (cf. [18, Sect. 4.1]), the iPhone 12 mini exhibits no correlation between its 12MP and 7MP BPs. However, the NCC map between the BPs $\hat{\mathbf{P}}_{\text{SLM}}$ of the iPhone 15 and iPhone 12 mini at 7MP resolution (see Fig. 10b) closely resembles that obtained at 12MP (Fig. 10a). This observation indicates that Apple may use a distinct BP algorithm for front (7MP) and rear (24/12MP) cameras, likely in a seed/model-dependent manner.

The evolving BP generation process is also evident in iPhone 11 Pro (iOS 13) portrait images from [1]. The extracted

TABLE I
BP COMPATIBILITY ACROSS DIFFERENT IPHONE MODELS AT 12MP RESOLUTION.

iPhone **iPhone** (Release)\(iOS)	**7 Plus**† (10.3.2)	**8 Plus**† (11.3.1)	**X**† (1.0.1)*	**XR**† (12.3.1)	**11**† (13.2)	**11 Pro**† (13.3.1)	**SE (2)** (16.1.2)	**X** (16.6)	**11** (17.1.1)	**11 Pro** (17.6.1)	**11 ProMax** (16.7.2)	**12** (17.1.2)	**12 mini** (16.6.1)	**12 Pro** (17.1.1)	**13** (17.1.1)	**13 Pro** (16.6)	**13 ProMax** (16.5)	**14** (17.1.2)	**14 ProMax** (17.0.5)	**15** (17.5.1)
7 Plus† (2016)		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
8 Plus† (2017)	✗		${\color{green}\checkmark}$	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
X† (2017)	✗	${\color{green}\checkmark}$		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
XR† (2018)	✗	✗	✗		${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{purple}\checkmark}$	${\color{purple}\checkmark}$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$
11† (2019)	✗	✗	✗	${\color{green}\checkmark}$		${\color{green}\checkmark}$	${\color{purple}\checkmark}$	${\color{purple}\checkmark}$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$
11 Pro† (2019)	✗	✗	✗	${\color{green}\checkmark}$	${\color{green}\checkmark}$		${\color{purple}\checkmark}$	${\color{purple}\checkmark}$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$
SE (2) (2020)	✗	✗	✗	${\color{purple}\checkmark}$	${\color{purple}\checkmark}$	${\color{purple}\checkmark}$		${\color{green}\checkmark}$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$
X (2017)	✗	✗	✗	${\color{purple}\checkmark}$	${\color{purple}\checkmark}$	${\color{purple}\checkmark}$	${\color{green}\checkmark}$		${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$
11 (2019)	✗	✗	✗	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$		${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$
11 Pro (2019)	✗	✗	✗	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}$		${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$
11 Pro Max (2019)	✗	✗	✗	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}$	${\color{green}\checkmark}$		${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$
12 (2020)	✗	✗	✗	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$		${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$
12 mini (2020)	✗	✗	✗	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$		${\color{green}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$
12 Pro (2020)	✗	✗	✗	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$		${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$
13 (2021)	✗	✗	✗	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$		${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$
13 Pro (2021)	✗	✗	✗	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{green}\checkmark}$		${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$
13 Pro Max (2021)	✗	✗	✗	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$		${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$
14 (2022)	✗	✗	✗	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$		${\color{green}\checkmark}$	${\color{green}\checkmark}$
14 Pro Max (2022)	✗	✗	✗	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$		${\color{green}\checkmark}$
15 (2023)	✗	✗	✗	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{purple}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{green}\checkmark}(\leftrightarrow)$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{blue}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	${\color{green}\checkmark}$	

†Images collected from the dataset used in [1]. *Version of the Photos app.

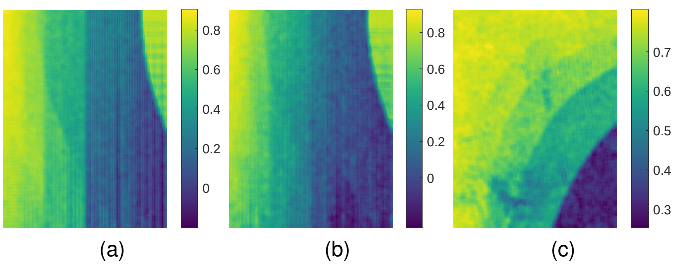


Fig. 10. NCC maps between different BPs: iPhone 15 ($\hat{\mathbf{P}}_{\mathrm{SLM}}$) x iPhone 12 mini ($\hat{\mathbf{P}}_{\mathrm{SLM}}$) for 12MP (a); iPhone 15 ($\hat{\mathbf{P}}_{\mathrm{SLM}}$) x iPhone 12 mini ($\hat{\mathbf{P}}_{\mathrm{SLM}}$) for 7MP (b); and iPhone 15 ($\hat{\mathbf{P}}_{\mathrm{SLM}}$) x iPhone 11 Pro ($\hat{\mathbf{P}}_{\mathrm{NL}}^{(\leftrightarrow)}$) (c).

BP, $\hat{\mathbf{P}}_{\mathrm{NL}}$, does not match that of the iPhone 15 (iOS 17) but aligns partially when flipped horizontally (i.e., $\hat{\mathbf{P}}_{\mathrm{NL}}^{(\leftrightarrow)}$), as seen in Fig. 10c. Correlation degradation and arc patterns suggest again an algorithmic BP generation. Intrigued by these findings, we tracked software-driven BP changes by analyzing portrait images from various iPhone models, leveraging the database from [1] and supplementing it with images from a contest organized locally [26] and through Flickr.[7]

Tab. I summarizes BP compatibility at 12MP resolution across the iPhone models we examined. In this table, the symbol ✗ indicates no match between extracted BPs for a given pair of models. Positive matches are categorized into three classes: ${\color{green}\checkmark}$ denotes a perfect match; ${\color{blue}\checkmark}$ indicates a partial match, consistent with the NCC map in Fig. 10a; and ${\color{purple}\checkmark}$ represents another partial match compatible with Fig. 10c. The symbol $(\leftrightarrow)$ denotes that one of the two BPs must be horizontally flipped to match. Note that images from the first six iPhone models come from the database in [1], which includes devices running iOS up to version 13.3.1, while the remaining devices have iOS 16.1.2 or higher. This explains why different iOS versions for the same models (e.g., iPhone X, iPhone 11, and iPhone 11 Pro) result in different BP outcomes. While early models (e.g., iPhones 7 Plus and 8 Plus) have incompatible BPs, consistency improves from the iPhone XR onward, though some versions require flipping. Starting with the iPhone 11, BP variations emerge, aligning with the iPhone 12 mini. From the iPhone 13 onward, BP remains stable, matching the iPhone 15. As BP appears to be iOS-dependent, these patterns may evolve over time.

[7]We identified Apple portrait images by checking the EXIF tag `CustomRendered` for the values `Portrait` and `Portrait HDR`.

VI. BP-BASED FORENSIC APPLICATIONS

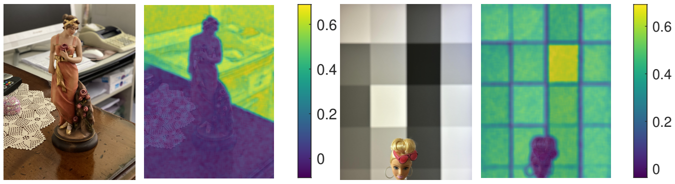
In this section, we propose BP-enabled forensic applications, including BP detection, identification, and localization methods, and a BP-aware approach for PRNU-based camera source verification. A proof-of-concept for forgery localization using BP-aware PRNU analysis is presented in [18, Sect. 5.1].

A. Apple's BP Detection and Identification

Detecting Apple's BP in an image is valuable for labeling or isolating images generated by computational imaging (e.g., portrait mode, as proposed in [4], [7]) and for curating machine learning datasets. Identifying the specific BP also enables tracing the iPhone model and iOS version, aiding forensic investigations.

A simple detection method computes the noise residue $\mathbf{W}$ from the image under analysis via (7) with $K = 5$ (higher K only benefits uniform regions), followed by the NCC in (1) between $\mathbf{W}$ and a given BP estimate $\hat{\mathbf{P}}$. If image orientation is unknown, all $90°$ rotations must be tested. Images sharing the same (spatially aligned) BP should yield a high NCC; thus, those with $\rho(\mathbf{W}, \hat{\mathbf{P}}) > \beta$, where β is a predefined threshold, are flagged as Apple portrait images.

Given multiple BP versions (see Sect. V) and potentially missing EXIF metadata, the NCC must be computed across all BP estimates and $90°$ rotation variants. The BP yielding the highest NCC above β is selected; if none exceed β, no BP match is detected.



(a) ISO 400 (b) ISO 125

Fig. 11. Portrait images and their relative BP-driven NCC maps: iPhone 11 Pro Max (iOS 13.4) [4, C22] (a) and iPhone 15 (iOS 17.5.1) (b). In both cases, $\hat{\mathbf{P}}_{\mathrm{SLM}}$ from the iPhone 15 is used, with a horizontal flip in (a).

B. BP-aware PRNU-based Camera Source Verification

To address the bias that Apple's SDNP induces in PRNU-based source verification (cf. Fig. 1 in Sect. II-A), we exploit the fact that regions containing the SDNP can be localized using a matching BP estimate. In particular, a BP-driven NCC map is computed between a given image residue $\mathbf{W}$ (obtained through (7)) and a BP estimate $\hat{\mathbf{P}}$, with local NCC values evaluated over 16×16 blocks using (1). For blocks near image boundaries, missing samples are padded by mirror reflection. The resulting map is then smoothed with a box filter ($K = 5$) and resized via nearest-neighbor interpolation to match the dimensions of $\mathbf{W}$, yielding the final BP-driven NCC map $\mathbf{R}$. For illustration, we used the BP estimate from 12MP SLM portrait images of the iPhone 15, $\hat{\mathbf{P}}_{\mathrm{SLM}}$, to generate the NCC map $\mathbf{R}$ for two portrait images: one from device C22 in the dataset from [4] (where $\hat{\mathbf{P}}^{(\leftrightarrow)}_{\mathrm{SLM}}$ is used) and another from the same iPhone 15 device. The overlaid maps (Figs. 11a and 11b) reveal regions where the SDNP is present, as indicated by higher NCC values.

Leveraging this map, we propose filtering out SDNP-affected regions during PRNU extraction and detection. When a matching BP estimate is available, it is used directly; otherwise, the method in Sect. VI-A is used to identify the best-matching BP (even if the highest NCC value falls below the threshold β). From this, we compute the BP-driven NCC map $\mathbf{R}$ between the image residue $\mathbf{W}$ and the BP estimate $\hat{\mathbf{P}}$, applying a predefined threshold α to create a binary mask $\mathbf{M}^{(\mathrm{PRNU})}$. This mask excludes regions where the BP is likely present by setting $M^{(\mathrm{PRNU})}_{i,j} = 1$ where $R_{i,j} \leq \alpha$, and $M^{(\mathrm{PRNU})}_{i,j} = 0$ otherwise. Applying this mask in (3) yields a BP-aware PRNU estimate:

$$\hat{\mathbf{K}}' = \left(\sum_{l=1}^{L} \mathbf{W}'_l \circ \mathbf{Z}'_l \right) \circ \left(\sum_{l=1}^{L} \mathbf{Z}'_l \circ \mathbf{Z}'_l \right)^{\circ -1}, \tag{18}$$

where $\mathbf{W}'_l = \mathbf{M}^{(\mathrm{PRNU})}_l \circ \mathbf{W}_l$ and $\mathbf{Z}'_l = \mathbf{M}^{(\mathrm{PRNU})}_l \circ \mathbf{Z}_l$ represent the lth masked residue and masked image, respectively, excluding regions where the BP under analysis is detected. Note the slight abuse of notation here: in this context, $\mathbf{Z}_l$ can represent either a portrait or a non-portrait image. If non-portrait images are processed, we expect $\mathbf{M}^{(\mathrm{PRNU})}_l$ to be a matrix entirely composed of ones. Given the BP-aware PRNU estimate $\hat{\mathbf{K}}'$, we modify the similarity measure defined in (4) to incorporate the SDNP masking. The test for checking the PRNU presence in a test image $\mathbf{Z}_t$ becomes:

$$\eta' \triangleq N \cdot \mathrm{ssq}\left(\rho \left(\mathbf{W}'_t, \hat{\mathbf{K}}' \circ \mathbf{Z}'_t \right) \right) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \tau', \tag{19}$$

where $\mathbf{W}'_t = \mathbf{W}_t \circ \mathbf{M}^{(\mathrm{PRNU})}_t$ and $\mathbf{Z}'_t = \mathbf{Z}_t \circ \mathbf{M}^{(\mathrm{PRNU})}_t$. Similarly to BP alignment, when the image orientation is unknown, we evaluate all possible 90° rotations of $\hat{\mathbf{K}}'$ and select the highest resulting η' value to ensure proper alignment with the PRNU.

VII. Experimental Results

We conducted several experiments to validate our BP-enabled forensic applications, covering BP detection and identification, PRNU collision mitigation, comparison with Baracchi *et al.* [2], and robustness against post-processing.[8]

A. Performance of Apple's BP Detection and Identification

To validate our method for Apple's BP detection and identification (Sect. VI-A), we first tested it in a controlled scenario with a labeled dataset of 10,079 images (12MP). The negative set comprises 9,677 non-Apple portrait images from [1], including images from various iPhone models and Samsung devices. The positive set contains 402 Apple portrait images: 151 and 109 captured by us with the iPhone 15 and iPhone 12 mini (iOS 16/17), and 142 sourced from [1], including 119 from the iPhone 11 Pro, 14 from the iPhone 11, and 9 from the iPhone XR (iOS 12/13). For this experiment, we used $\hat{\mathbf{P}}_{\mathrm{SLM}}$, extracted from the iPhone 15 and 12 mini (both at ISO 125), and $\hat{\mathbf{P}}_{\mathrm{NL}}$ (no SLM images available), derived from 40 images of devices C19–C22 in [4]. These devices run a similar iOS version to iPhones 11 Pro, 11, and XR from [1], sharing the same BP. Note that we excluded portrait images from iPhones 7 Plus, 8 Plus, and X in [1] because their BPs, which differ from the 3 considered here (see Tab. I), were already estimated using these same images in [1]. Still, these BP estimates will be used in the next experiment.

Using the threshold $\beta = 0.01$ for our detector, all non-Apple portrait images were correctly classified as negatives. Among the positive samples, only one iPhone 11 portrait image was misclassified as a negative due to the small region within the image where the SDNP was present. The identification of the specific BP used was perfectly accurate for all correctly detected Apple portrait images.

Our approach was also tested on the FFHQ (in-the-wild-images) dataset [27] to evaluate its performance in an uncontrolled, open-set scenario. Results showed that approximately 18% of 12MP images contained Apple's BP, with only one false positive from a DSLR camera. Some misdetections occurred due to lower-quality BP estimates for older iPhone models and potential post-processing artifacts. Despite this, BP attribution remained highly accurate, with only a few misclassifications, highlighting the method's robustness in real-world conditions. A comprehensive analysis of the obtained results is conducted in the technical report [18, Sect. 5.2].

B. Tackling PRNU Collisions with BP-aware PRNU Matching

Using the BP-aware PRNU-based camera source verification approach from Sect. VI-B (with $K = 5$ and $\alpha = 0.07$), we repeat the experiments from Sect. II-A, excluding 5 misaligned

[8] Code and estimated BPs will be released upon acceptance; meanwhile, they are available on request from the corresponding authors.

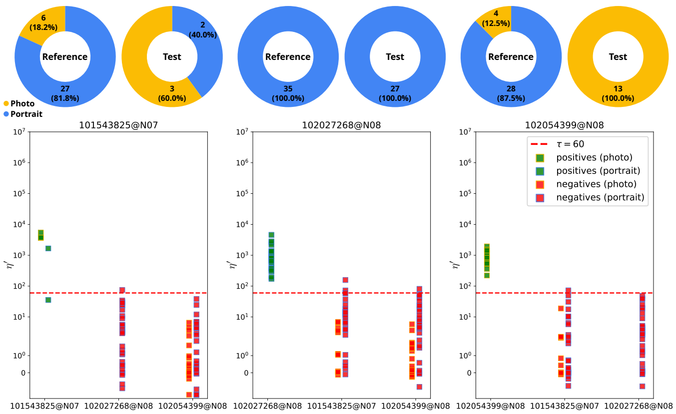


Fig. 12. Results achieved using our BP-aware PRNU matching approach on the dataset employed in [1].

or different-sensor images from user `101543825@N07` (see [18, Sect. 1]). The updated results in Fig. 12 show a significant reduction in the similarity measure η' for portrait-mode images compared to the original η (Fig. 1). Using the original threshold (i.e., $\tau' = 60$) yields few classification errors, while raising it to $\tau' = 160$ eliminates most errors, leaving only one misdetection. These findings confirm that properly accounting for Apple's SDNP enables reliable PRNU-based source camera verification, regardless of portrait mode.

It is important to note that further adjusting the threshold α, which defines the regions where the SDNP is present, could improve both these and subsequent results. We believe that developing a strategy to determine the optimal threshold α for each image would significantly enhance the overall performance of our approach, but we leave the development of this strategy for future work.

C. Comparative Results of BP-aware PRNU Matching

Baracchi *et al.* in [2] were the first to address NUAs in Apple portrait images in a PRNU-based camera source verification context, showing that such NUAs correlated more strongly than the PRNU. After our analysis, however, we understand that the underlying BP of Apple's SDNP is not strictly an artifact but rather a synthetic pattern intentionally embedded by Apple. Baracchi *et al.* proposed two distinct techniques, the "weighted" and "binary" methods, which use depth map information to either weigh or exclude regions in the PRNU estimate $\hat{\mathbf{K}}$ (computed using the baseline PRNU method from Sect. II-A), resulting in $\hat{\mathbf{K}}_{\mathbf{w}}$ and $\hat{\mathbf{K}}_{\mathbf{b}}$, respectively. For both methods, we employ the same test statistic as in (4), substituting $\hat{\mathbf{K}}$ by $\hat{\mathbf{K}}_{\mathbf{w}}$ and $\hat{\mathbf{K}}_{\mathbf{b}}$ accordingly. In our approach (detailed in Sect. VI-B), the image residue $\mathbf{W}$ and the binary mask $\mathbf{M}^{(\mathrm{PRNU})}$ are computed using $K = 5$ and $\alpha = 0.07$, respectively, applying the test statistic from (19). In the following, we compare Baracchi *et al.*'s methods and the baseline PRNU approach with our BP-aware solution by conducting three experiments, each evaluating different BPs.

1) Comparative Analysis on the Dataset from [1]: The tests are conducted on images from the 3 iPhone 11 Pro users identified with PRNU collisions in [1]. PRNU extraction uses the 35 images of the Reference set from user `102027268@N08`. The positive class consists of 27 test images from the same user, while the negative class includes 86 images from the other two users: `101543825@N07` (41 images, excluding 2 taken with digital zoom) and `102054399@N08` (45 images). Our approach employs the BP estimate $\hat{\mathbf{P}}_{\mathrm{NL}}$ extracted from 40 portrait images taken by devices C19, C20, C21, and C22 in [4], whose BP matches that of the iPhone 11 Pro in [1].

The ROC curves obtained for each detector are shown in Fig. 13a. Note that the binary method from [2] is absent, because the exclusion of regions from the images captured by user `102027268@N08` results in a PRNU estimate $\hat{\mathbf{K}}_{\mathbf{b}}$ consisting entirely of zeros. Interestingly enough, the weighted method performs worse than the baseline approach (as also observed during our attempt to replicate the original results in [18, Sect. 5.1]), suggesting that its effectiveness is highly dependent on the specific scene captured and the corresponding depth map. In contrast, our approach, which focuses on localizing the presence of the BP, proves to be very effective, achieving perfect detection, as previously highlighted in Fig. 12. We believe that the primary reason for the lower performance of Baracchi *et al.*'s method is its exclusive reliance on depth map information. As discussed in Sect. II-B2, the exact algorithm Apple uses to determine which regions will have the SDNP added remains unknown and may have evolved over time. In fact, all images in our comparative analysis were captured with newer iOS versions (i.e., iOS 13.3.1 or later) than those analyzed by Baracchi *et al.* (i.e., iOS 12.1.4), which may explain why we are unable to replicate their results in [2].

Observations from the portrait images used in this experiment suggest that Apple's algorithm incorporates more than just the depth map, likely leveraging additional information (such as the auxiliary mattes mentioned in Sect. II-B) to decide which region in the image is finally blurred. This is evident in the example (taken from the reference set) shown in Fig. 14a, where its depth map (Fig. 14b) does not capture details of the girl's eyes, yet Apple's algorithm keeps only the eye region unblurred. The binary mask produced by Baracchi *et al.*'s method, obtained via Otsu thresholding (Fig. 14c), fails to exclude many SDNP-containing regions during PRNU extraction, leading to performance degradation. In contrast, our approach produces a binary mask $\mathbf{M}^{(\mathrm{PRNU})}$ (Fig. 14d) that effectively minimizes the BP leakage into the PRNU estimate $\hat{\mathbf{K}}$. The NCC values computed between the residues and the BP after applying the respective masks are 0.3156 for the weighted approach and 0.0201 for ours.

Another example of depth map-related issues comes from the negative class of the test set (Fig. 14e) and its corresponding depth map (Fig. 14f). Otsu thresholding produces a binary mask (Fig. 14g) that fails to segment the in-focus region, instead isolating an area composed almost entirely of the SDNP. This results in a false positive, with a test statistic of $\eta = 1.71 \times 10^6$. In contrast, our BP-driven mask (Fig. 14h) yields $\eta' = 4.62$, which aligns with the expected value for a negative sample. These findings highlight that minimizing BP leakage during both PRNU extraction and subsequent detection steps is crucial for improving source camera verification performance when dealing with Apple

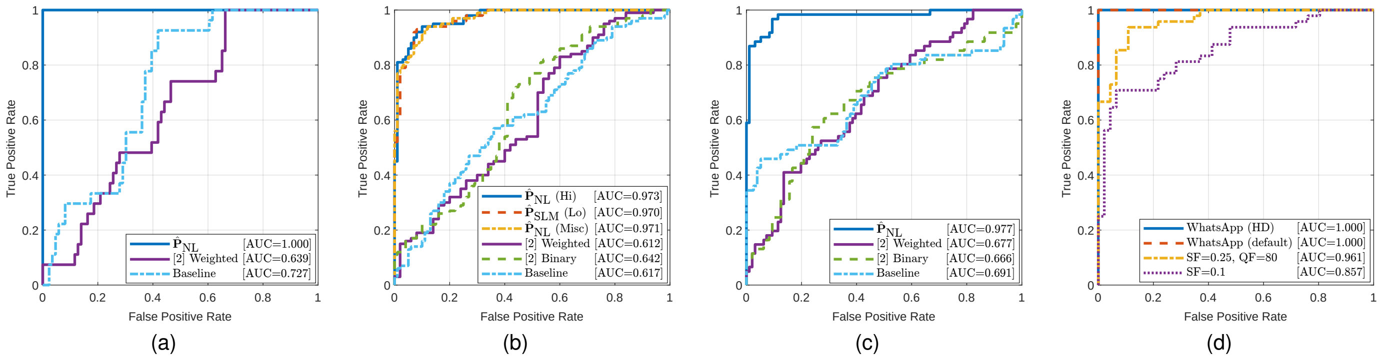


Fig. 13. ROC curves comparing PRNU-based source camera verification performance across different methods: baseline PRNU, the two approaches from [2], and our BP-aware solution. The plots correspond to Sect. VII-C1 (a), Sect. VII-C2 (b), Sect. VII-C3 (c). Panel (d) presents ROC curves from Sect. VII-D, illustrating the impact of post-processing on Apple portrait image detection.

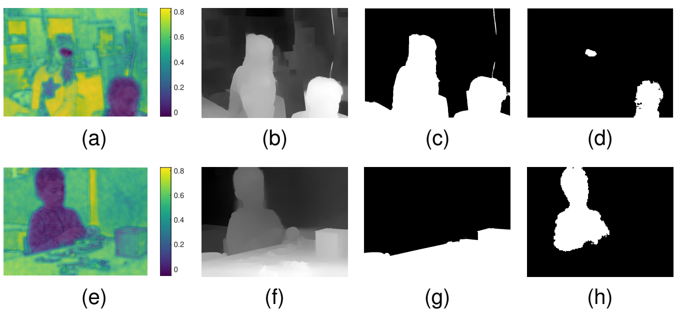


Fig. 14. Example portrait images with overlaid BP-driven NCC maps (a, e), relative depth maps (b, f) and Otsu thresholding masks (c, g) from [2]. Binary masks $\mathbf{M}^{(\mathrm{PRNU})}$ (d, h) are obtained after thresholding NCC maps in (a, e).

portrait images.

2) Impact of BP Quality on Detection Performance: We evaluate the impact of BP estimation quality using three estimates from the iPhone 15: $\hat{\mathbf{P}}_{\mathrm{NL}}$ (Hi) with $L = 110$, $\hat{\mathbf{P}}_{\mathrm{SLM}}$ (Lo) with $L = 10$, and $\hat{\mathbf{P}}_{\mathrm{NL}}$ (Misc) with $L = 158$, following the capture conditions in [18, Sect. 4]. The PRNU is estimated using 10 reference portrait images from the same device. The test set consists of 100 portrait images from the iPhone 15 (positive class) and 100 from an iPhone 13 (negative class), sourced from Flickr. Both devices share the same BP (Tab. I).

In this experiment, we ensured that the images used for PRNU extraction produced a PRNU estimate $\hat{\mathbf{K}}_{\mathbf{b}}$ with non-null support, making it suitable for use with the binary method from [2]. Under these conditions, the binary method outperforms both the weighted and baseline approaches in AUC, whereas our approach achieves the best performance, with minimal variation across the three BP estimates. Among them, $\hat{\mathbf{P}}_{\mathrm{NL}}$ (Hi) achieves the highest AUC, followed by $\hat{\mathbf{P}}_{\mathrm{NL}}$ (Misc) and $\hat{\mathbf{P}}_{\mathrm{SLM}}$ (Lo). Notably, $\hat{\mathbf{P}}_{\mathrm{NL}}$ (Misc) is preferable in low False Positive Rate (FPR) scenarios, maintaining the highest partial AUC for FPR ≤ 0.05 (0.0373), closely followed by $\hat{\mathbf{P}}_{\mathrm{NL}}$ (Hi) (0.0372). While $\hat{\mathbf{P}}_{\mathrm{SLM}}$ (Lo) consistently underperforms, its ease of extraction makes it a practical alternative.

Based on [18, Fig. 8] and [18, Sect. 4], larger differences in detection performance across BP estimates might be expected. However, while those results were obtained using uniform image blocks in controlled conditions, this experiment processes real-world scenes. Here, applying a fixed threshold of $\alpha = 0.07$ across all BP estimates produced highly similar binary masks $\mathbf{M}^{(\mathrm{PRNU})}$, resulting in nearly identical PRNU estimates $\mathbf{K}'$ and comparable residue masking in (19), ultimately yielding similar detection performance. Future work could explore adaptive thresholds tailored to each BP estimate to better capture variations in BP quality.

3) Comparative Analysis with iPhone 12 mini and Multiple Users: In this experiment, we use the BP estimate $\hat{\mathbf{P}}_{\mathrm{NL}}$ ($L = 50$) from the iPhone 12 mini and extract the PRNU from 50 reference portrait images of the same device. The positive class consists of 60 portrait images from the same iPhone 12 mini, while the negative class includes 60 images from Flickr: 37 and 5 from two iPhone 12, and 10 and 8 from two other iPhone 12 mini. These models share the same BP (Tab. I).

The results in Fig. 13c follow the same trend as those in Fig. 13b for the iPhone 15. However, unlike the first experiment with iPhone 11 Pro images from [1] (Fig. 13a), perfect detection is not achieved by our approach for newer devices (iPhone 12 mini and iPhone 15). A key factor is the iOS version: the iPhone 11 Pro images were captured with iOS 13, whereas the newer devices use iOS 17. This suggests that Apple has refined SDNP embedding over time, making it more selective (particularly by omitting certain areas like edges). This is evident in the NCC maps in Fig. 11, where the iPhone 11 Pro Max image (iOS 13, Fig. 11a) shows less correlation loss at the edges compared to the iPhone 15 image (iOS 17, Fig. 11b). In the latter case, correlation loss is pronounced in the brightness transitions on the ColorChecker, affecting scene segmentation for PRNU matching. These areas, still blurred, likely retain fewer PRNU traces, reducing detection accuracy.

D. Robustness of BP Detection under Post-Processing

To assess the robustness of our BP detection approach under post-processing, we tested image sharing via WhatsApp. We sent 48 portrait images from an iPhone 12 mini using two modes: the default mode, which reduces resolution (12MP to 2MP) and applies compression, and the HD mode, which preserves resolution with moderate compression. These formed the positive classes, while the negative classes included 46 non-portrait images shared under the same conditions. As shown in Fig. 13d, both modes had no impact on BP detection, achieving perfect results comparable to cases without post-processing. To illustrate a more challenging case where detection performance starts to drop, we also included ROC curves for images downscaled to a Scaling Factor (SF) of 0.25 and JPEG compressed with a Quality Factor (QF) of 80, as well as for extreme image downscaling by SF $= 0.1$ without additional compression. Further details on the effects of scaling and compression can be found in [18, Sect. 5.3].

VIII. Conclusions

The distinctive Apple's SDNP and its underlying BP have been comprehensively analyzed in this paper, characterizing their role in iPhone portrait images and their forensic implications. We first proposed BP extraction methods under different lighting conditions and then examined its dependencies on luminance, ISO settings, and software variations. Our findings show that SDNP knowledge enhances forensic tasks, including portrait-mode image traceability and improved PRNU-based source verification in realistic forensic scenarios.

Future work will extend this analysis to other Apple devices not covered here (e.g., iPads and other iPhones) and refine forensic applications, such as forgery localization using BP-aware PRNU analysis. We aim to improve BP extraction and detection using $\hat{g}(\cdot)$, and develop adaptive thresholding for SDNP localization, optimizing performance across diverse images and BP quality variations. While this work focuses on Apple devices, an important open question is whether similar methods can generalize to other smartphone brands. We hope this work inspires further forensic research in this area.

Appendix A
Derivation of the LS Estimate of $\hat{g}(\cdot)$

In this appendix we derive the estimate of the brightness-dependent scaling function $\hat{g}(\cdot)$ following an LS approach. Let T and tr denote respectively the transpose and trace matrix operators. Given two matrices $\mathbf{A}, \mathbf{B}$ and a scalar c, let $\mathbf{E} \triangleq (\mathbf{A} - c\mathbf{B})^T (\mathbf{A} - c\mathbf{B})$. Notice that $\|\mathbf{A} - c\mathbf{B}\|_{\mathrm{F}}^2 = \mathrm{tr}(\mathbf{E})$. Taking the derivative of $\mathbf{E}$ w.r.t. c, we have $\frac{d\mathbf{E}}{dc} = 2c\mathbf{B}^T\mathbf{B} - \mathbf{B}^T\mathbf{A} - \mathbf{A}^T\mathbf{B}$. Noticing that $\mathrm{tr}(\mathbf{A}^T\mathbf{B}) = \langle \mathbf{A}, \mathbf{B} \rangle_{\mathrm{F}}$ and that the trace operator commutes with the derivative, we have that $\frac{d\ \mathrm{tr}(\mathbf{E})}{dc} = 2c\|\mathbf{B}\|_{\mathrm{F}}^2 - 2\langle \mathbf{A}, \mathbf{B} \rangle_{\mathrm{F}}$.

Assuming that block $\mathbf{Z}_{m,t}$ contains the base pattern $\mathbf{P}_m$ everywhere, i.e. $\mathbf{M}_{(\text{blur})} = \mathbf{1}$ and $\mathbf{M}'_{(\text{blur})} = \mathbf{0}$, and that it has a flat background, Eq. (6) becomes $\mathbf{Z}_{m,t} = \mathbf{Y}'_{m,t} + \gamma_{\text{ISO}} \cdot g(y'_{m,t})\mathbf{P}_m + \mathbf{\Phi}_{m,t}$. Here we make no assumptions on the mean and std of $\mathbf{P}$, as some estimates of $\mathbf{P}$ (e.g., $\hat{\mathbf{P}}_{\text{SLM}}$) may not share its second order statistics. We are interested in finding $y'_{m,t} \triangleq \mu(\mathbf{Y}'_{m,t})$ and $g(y'_{m,t})$ minimizing

$$\|\mathbf{Z}_{m,t} - \mathbf{Y}'_{m,t} - \gamma_{\text{ISO}} \cdot g(y'_{m,t})\mathbf{P}_m\|_{\mathrm{F}}^2. \quad (20)$$

First, notice that this norm is minimized when $\mu\left(\mathbf{Z}_{m,t} - \mathbf{Y}'_{m,t} - \gamma_{\text{ISO}} \cdot g(y'_{m,t})\mathbf{P}_m\right) = 0$, leading to

$$\hat{y}'_{m,t} = z_{m,t} - \gamma_{\text{ISO}} \cdot g(y'_{m,t}) \cdot \mu(\mathbf{P}_m), \quad (21)$$

where $z_{m,t} \triangleq \mu(\mathbf{Z}_{m,t})$. Now we use the algebraic result at the beginning of this Appendix to compute the derivative of (20) with respect to $g(y'_{m,t})$. Equating to zero, we obtain

$$\langle \mathbf{Z}_{m,t}, \mathbf{P}_m \rangle_{\mathrm{F}} - \langle \mathbf{Y}'_{m,t}, \mathbf{P}_m \rangle_{\mathrm{F}} - \gamma_{\text{ISO}} \cdot g(y'_{m,t}) \cdot \|\mathbf{P}_m\|_{\mathrm{F}}^2 = 0 \quad (22)$$

Since the block is assumed to have flat background, we can approximate $\langle \mathbf{Y}'_{m,t}, \mathbf{P}_m \rangle_{\mathrm{F}} \approx y'_{m,t} \cdot \mu(\mathbf{P}_m)B^2 \approx z_{m,t} \cdot \mu(\mathbf{P}_m)B^2 - \gamma_{\text{ISO}} \cdot g(y'_{m,t}) \cdot (\mu(\mathbf{P}_m))^2 B^2$, where the second approximation follows from substituting $y'_{m,t}$ in the first by its estimate in (21). Replacing $\langle \mathbf{Y}'_{m,t}, \mathbf{P}_m \rangle_{\mathrm{F}}$ in (22) by this approximation, substituting $\mathbf{P}_m$ and γ_{ISO} by their respective estimates, and solving for $g(y'_{m,t})$, we obtain (17).

References

[1] M. Iuliani, M. Fontani, and A. Piva, "A Leak in PRNU Based Source Identification. Questioning Fingerprint Uniqueness," *IEEE Access*, vol. 9, pp. 52 455–52 463, 2021.

[2] D. Baracchi, M. Iuliani, A. G. Nencini, and A. Piva, "Facing Image Source Attribution on iPhone X," in *19th IWDW*, 2020, pp. 196–207.

[3] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, "Source digital camcorder identification using sensor photo response non-uniformity," in *Sec., Steg., and Wat. of Multimedia Contents IX*, vol. 6505. SPIE, 2007, p. 65051G.

[4] C. Albisani, M. Iuliani, and A. Piva, "Checking PRNU Usability on Modern Devices," in *IEEE ICASSP*, 2021, pp. 2535–2539.

[5] N. Nisar Bhat and T. Bianchi, "Investigating Inconsistencies in PRNU-Based Camera Identification," in *IEEE ICIP*, 2022, pp. 851–855.

[6] A. Montibeller, R. A. Asiku, F. Pérez González, and G. Boato, "Shedding Light on some Leaks in PRNU-based Source Attribution," in *ACM IH&MMSec*, 2024, p. 137–142.

[7] S. McCloskey, *Computational Imaging*. Springer, 2022, pp. 41–62.

[8] J. Butora and P. Bas, "The Adobe Hidden Feature and its Impact on Sensor Attribution," in *ACM IH&MMSec*, 2024, p. 143–148.

[9] ——, "Detection of the Adobe Pattern," in *32nd European Signal Processing Conference*, Lyon, France, Aug. 2024.

[10] [Online]. Available: https://www.counterpointresearch.com/insights/global-smartphone-share/

[11] [Online]. Available: https://www.counterpointresearch.com/insights/global-smartphone-sales-top10-best-sellers/

[12] J. Lukáš, J. Fridrich, and M. Goljan, "Determining digital image origin using sensor imperfections," in *Image and Video Communication and Processing 2005*, vol. 5685. SPIE, 2005, pp. 249 – 260.

[13] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, "Determining Image Origin and Integrity Using Sensor Noise," *IEEE TIFS*, vol. 3, no. 1, pp. 74–90, 2008.

[14] M. Kivanc Mihcak, I. Kozintsev, and K. Ramchandran, "Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising," in *IEEE ICASSP*, vol. 6, 1999, pp. 3253–3256 vol.6.

[15] S. Fernández-Menduiña and F. Pérez-González', "On the information leakage quantification of camera fingerprint estimates," in *EURASIP Journal on Information Security*, vol. 6, 2021.

[16] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," in *Media Forensics and Security*, vol. 7254, Int. Society for Optics and Photonics. SPIE, 2009, p. 72540I.

[17] F. Pérez-González, M. Masciopinto, I. González-Iglesias, and P. Comesaña, "Fast Sequential Forensic Detection of Camera Fingerprint," in *IEEE ICIP*, 2016, pp. 3902–3906.

[18] D. Vázquez-Padín, F. Pérez-González, and P. Pérez-Miguélez, "Technical Report: Expanded Analysis and Experimental Evaluation of Apple's Synthetic Defocus Noise Pattern," 2025. [Online]. Available: https://doi.org/10.13140/RG.2.2.28152.15363

[19] N. Wadhwa *et al.*, "Synthetic Depth-of-Field with a Single-Camera Mobile Phone," *ACM Trans. Graph.*, vol. 37, no. 4, jul 2018.

[20] I. 23008-12:2022, "Information technology - High efficiency coding and media delivery in heterogeneous environments - Part 12," 2022.

[21] [Online]. Available: https://developer.apple.com/documentation/avfoundation/additional_data_capture/capturing_photos_with_depth

[22] [Online]. Available: https://developer.apple.com/documentation/avfoundation/avportraiteffectsmatte

[23] [Online]. Available: https://github.com/strukturag/libheif

[24] T. E. Bishop, A. Lindskog, C. Molgaard, and F. Doepke, "Photo-realistic shallow depth-of-field rendering from focal stacks," US Patent US10284835B2, May 2019.

[25] S. Fernández-Menduiña, F. Pérez-González, and M. Masciopinto, "Source camera attribution via PRNU emphasis: Towards a generalized multiplicative model," *Sig. Proc.: Im. Com.*, vol. 114, p. 116944, 2023.

[26] [Online]. Available: https://forms.gle/PJW8UbASFX5oBsUu9

[27] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4396–4405.