Politecnico di Torino

Politecnico di Torino

Communications and Computer Networks Engineering (CCNE)
2021/2022
December

Investigating Inconsistencies in PRNU-based Camera Identification

Master's Thesis

Supervisor:
Prof. Tiziano Bianchi

Candidate:
Nabeel Nisar Bhat
(s271913)

Abstract

PRNU (Photo-Response Non-Uniformity) is widely considered a unique and reliable fingerprint for Camera Identification. The PRNU patterns of two different sensors are always uncorrelated. Such a fingerprint is used as evidence in court for source identification, manipulation detection, *etc.* Recently, there have been enormous advancements in smartphone cameras. The introduction of software features like Portrait, AI effect, HDR10+, Scene Recognition, EIS (Electronic Image Stabilization), *etc.*, which come under the umbrella term *computational photography*, have significantly improved the camera performance. However, these features have a considerable impact on the PRNU fingerprint. The complex in-camera processing associated with these features introduces Non-unique Artifacts (NUA's) in the PRNU patterns of such smartphones. Therefore, PRNU patterns corresponding to different instances of the same smartphone exhibit unexpectedly high (cross) correlation scores, leaving a question mark on the uniqueness of the fingerprint. In other words, the fingerprints of different devices collide with each other. In such a scenario, camera identification can not be performed reliably.

The aim is to address the inconsistencies in PRNU-based source identification. The target consists of images belonging to the recent smartphones from Samsung, Huawei, and Xiaomi. A total of 4643 images belonging to 7 different Smartphone Models are collected from Flickr. The analysis part consists of verifying the abnormal behavior of the PRNU for these smartphones. Two tools are proposed to verify the abnormal behavior: Pair-wise Noise Residual and Pair-wise Fingerprint Comparison. Normalized Correlation and PCE metrics are estimated corresponding to every fingerprint or residual couple. Given the uniqueness of the fingerprint, these metrics should be significantly lower than the threshold value. On the contrary, the (cross) correlation metrics surpass the threshold, leading to many false positives. An in-depth analysis of meta-data content does not reveal any evident link between the Exposure Triangle (Shutter speed, ISO, and Aperture) and the unexpected behavior. Also, some manufacturers

choosing not to embed tags related to computational photography (Portrait Tag, HDR Tag, *etc.*) does not help either.

This work proposes two algorithms to tackle the problem: SPAM Classifier and Meta-Data SceneType Tag Classifier. The goal of the algorithms is to identify the images responsible for the fingerprint collision. The SPAM Classifier is valid for all the models and achieves high accuracy of 99.8% in classifying the images correctly. On the other hand, Meta-Data-based Classifier applies to Samsung Models only. Nevertheless, the latter is quicker and achieves decent accuracy of $\simeq$ 86.5%. The images responsible for the abnormal behavior are discarded as far as fingerprint extraction is concerned. Instead, the fingerprint is estimated from the images, not producing any false alarm. The fingerprint obtained in this way is unique and reliable.

Acknowledgements

First of all, I am eternally grateful to my supervisor, Prof. Tiziano Bianchi for guiding me through this project with great enthusiasm and affection. I also want to sincerely thank Rector, ensuring online teaching and support because of covid-19. Besides, I take the opportunity to thank those who took care of the remote server, virtual classroom, and lockdown browser to guarantee seamless online learning. My colleagues, especially Mr. Paolo Carniello and Mr. Jalees Nehvi, also deserve my appreciation for helping me with the various problems I faced during this project. Finally, I am highly thankful to my parents for keeping me motivated throughout the project.

Contents

List of Figures

List of Tables

LIST OF TABLES

Chapter 1

Introduction

1.1 Background

Source Identification is one of the crucial applications of Digital Image Forensics. It aims at verifying the origin and genuineness of an image. A typical source identification problem involves associating a picture to one of the reference devices. In other words, the purpose is to identify the device which produced the image. Many techniques have been proposed to identify an image source; however, PRNU-based camera identification prevails over the others.

Over the past decade, PRNU (Photo-Response Non-Uniformity) trace has been widely used to identify an image source. The PRNU pattern is unique to a sensor. The patterns of the two different sensors, even if coming from the same silicon wafer, differ significantly. Moreover, the pattern is relatively constant over time and survives most camera processing. Due to its uniqueness and reliability, the PRNU has been used as evidence in courts and by various law enforcement agencies to check the authenticity of an image, detect manipulations in images, *etc.* The reliability of such a trace is vital as it decides whether an accused is criminal or not. Researches have thoroughly investigated the effectiveness of such a fingerprint by performing tests on many datasets, *e.g.*, the Dresden dataset, the VISION dataset, *etc.* The results have been quite convincing; high detection accuracies and low percentages of false alarms have been associated with the PRNU fingerprint. A considerable number of refinements have also been proposed to improve the robustness of PRNU.

1.2 Motivation

From the past few years, the task of camera identification has gained much interest worldwide. According to a 2014 report, 1.8 billion pictures were shared every day on the internet, with a significant stake belonging to smartphones. Recently, with the advent of computational photography, smartphone cameras are competing with professional ones. The introduction of features like Portrait, Night Mode, AI effect, HDR10+, *etc.*, which require a lot of in-camera processing, has turned professional photographers' minds towards smartphones, given the ease and convenience smartphones offer. As a result, the share of smartphone cameras in photography has taken to the skies. As per the 2017 report, 85% of the worlds' pictures were captured by smartphones. These pictures and tons of source devices represent a challenge as far as camera identification is concerned. This poses several questions: Can we use PRNU to identify the origin of such a high number of images? Does the PRNU trace survive complex in-camera processing? Is PRNU for modern devices unique?

Recently, Iuliani *et al.* (2021) highlighted the problems associated with PRNU-based camera identification in their work, where they tested the effectiveness of PRNU for modern devices. The team documented quite surprising results as far as False Alarm Ratio (FAR) is concerned. They reported very high FAR's for most modern smartphones. In such a scenario, PRNU will lead to erroneous camera identification.

This problem is under-researched, and no solution to counter the problem has been proposed so far.

1.3 Focus

The purpose of this thesis is to check the robustness of PRNU in modern smartphones. We focus on verifying the problem on a subset of the dataset of Iuliani *et al.* (2021). To do this, we propose two verification tools: Pair-wise Fingerprint and Pair-wise Noise Residual Comparison. These tools expose the problem of fingerprint collision for modern smartphones. We suppose that this problem occurs due to the introduction of non-unique artifacts in the images. The complex in-camera processing behind the computational photography is likely to be similar for the devices of the same smartphone. As a result, the PRNU patterns and the Noise Residuals exhibit higher cross-correlations than expected. We also analyze the meta-data of the images to find whether some specific settings cause

this problem. In particular, we focus on the Exposure Triangle in the meta-data settings.

The novelty of this research is that we propose two algorithms to tackle this anomalous problem. One of the methods is based on SPAM features, and the other is based on Meta-Data Screening. Our algorithms aim to identify the images that produce anomalous behavior, *i.e.*, the images that lead to a large number of false alarms from those that do not. As a result, we can discard the former images as far as fingerprint extraction is concerned. Both of our methods have low computational costs. In particular, the method based on SPAM features achieves very high accuracy in correctly identifying these images.

1.4 Outline

We begin with the prerequisites in Chapter 2, where we describe the various concepts that form the ingredients of the main idea. Next, we introduce the problem in Chapter 3 and discuss the related work. A detailed description of our dataset follows this in Chapter 4. The dataset chapter is followed by methodology and methods in Chapter 5, in which we discuss the model and algorithms to tackle our problem. We also present two problem verification tools in this chapter. Next, we discuss the experiments concerning the tools and the algorithms in Chapter 6. We also present the accuracy measures of our algorithms in this chapter. Finally, we present conclusions, limitations, and future work in Chapter 7.

1.4 Outline

Chapter 2

Prerequisites

In this chapter, we present the main concepts relevant to our project. The chapter is divided into three main sections: Photo-Response Non-uniformity (PRNU), SPAM features, and Principal Component Analysis (PCA).

2.1 Photo-Response Non-Uniformity (PRNU)

The camera sensor is made of millions of photo-sites or pixels. The function of these photo-sites is to convert the incoming photons into voltages. Ideally, the photo-sites, also known as sensor elements, should output equal voltages. However, because of the manufacturing imperfections, these photo-sites have different sensitivity towards the incoming light. As a result, each sensor element outputs a slightly different voltage from others. Every sensor element provides a distinct gain to the incoming photons. This random fluctuation in the output voltages of the sensor elements is called Photo-Response Non-Uniformity or PRNU. This pattern was first identified by Fridrich *et al.* (2006). PRNU, otherwise known as pattern noise, is quite specific to the sensor. Therefore, the PRNU patterns of two different sensors are strongly uncorrelated Fridrich *et al.* (2006). PRNU is multiplicative, thus survives most in-camera processing. Moreover, it does not depend on environmental conditions like temperature, humidity, *etc.* Therefore, this pattern is relatively stable and constant over time. Hence, the PRNU noise is considered a unique and reliable fingerprint in Image Forensics.

2.1.1 PRNU Model: How is the fingerprint computed?

The mathematical model presented by Chen *et al.* (2008), describing the physics of the camera acquisition process is as follows :

$$S = G^C[(1 + P_N)I_V + N']^C + Q_N \tag{2.1}$$

S represents the output image of the Sensor,
G represents the Gain,
P_N represents the multiplicative PRNU Noise term,
C represents Gamma Correction or Contrast Correction,
I_V represents the intensity of the incident light,
N' represents the Additive Noise (dark current, electronic noise, *etc.*),
Q_N represent the Quantization Noise.
Simplifying the equation and from first-order Taylor approximation $(1 + x)^n = 1 + nx$, the above equation can be re-written as :

$$S = (GI_V)^C[1 + P_N + N'/I_V]^C + Q_N \tag{2.2}$$
$$S = (GI_V)^C[1 + CP_N + CN'/I_V] + Q_N \tag{2.3}$$
$$S = S_0 + S_0P'_N + N \tag{2.4}$$

where $S_0 = (GI_V)^C$ represents a Noise-free image,
$P'_N = CP_N$ represents the PRNU term,
and N represents the equivalent Noise term.
Given the model, the aim is to estimate P'_N observing the image S. Figure 2.1 shows the pipeline of the PRNU estimation. We can estimate P'_N in the following way:

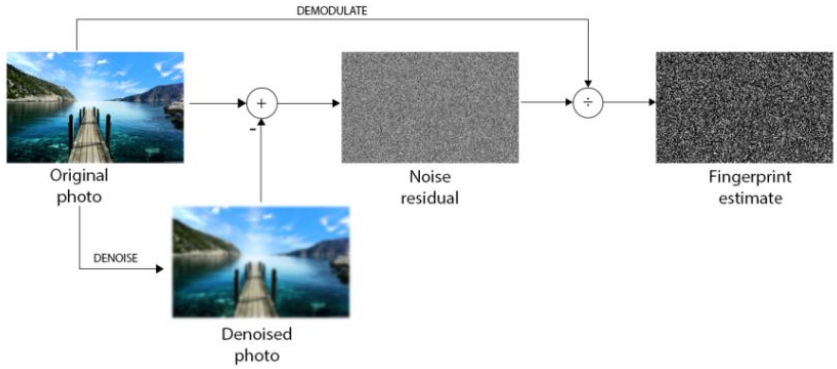


Figure 2.1: Fingerprint Estimation

- Estimate the noise-free image from the original image,
- Subtract the noise-free image from the original image,
- Demodulate the Noise Residual.

Mathematically,

$$\widehat{S_0} = D(S) \tag{2.5}$$
$$W = S - \widehat{S_0} \tag{2.6}$$
$$W = S_0 + S_0 P'_N + N - S_0 \tag{2.7}$$
$$W = S_0 P'_N + N \tag{2.8}$$

where D represents the Denoising Filter,
$\widehat{S_0}$ represents the estimated Denoised Image,
and W represents the Noise Residual.
P'_N is then estimated from k Noise Residuals.

$$\frac{W_k}{S_k} = P'_N + \frac{N_k}{S_k} \tag{2.9}$$

Denoising Filter
The first and the foremost step is the estimation of the noise-free image or denoised image. Since we do not observe the noise-free image, precise estimation of S_0 is of utmost importance. The simplest denoising filter is the Low Pass Filter (LPF). However, the shortcoming of the filter is that the Noise Residual obtained contains traces of image content (the edges) which the LPF cannot suppress. Instead, wavelet-based filters seem to give the best performance. The rationale behind using filters in the wavelet domain is that wavelet transformation sparsifies the signal. Natural images are smooth with few discontinuities, wavelength representation of images result in few coefficients with large magnitude (details of the image, *i.e.,* edges) and large coefficients close to zero, instead of in the pixel domain, all pixels hold equal importance, and there is no way to separate noisy pixels from the noise-free ones. This sparsifying property of the wavelet transform helps us better differentiate between the image's noise component and noise-free component.
Wavelength transform of the noise-free signal is very sparse. Most of the coefficients are zero; on the other hand, the wavelength transform of noise gives small coefficients at all scales. Then thresholding is done to separate noisy coefficients

from the noise-free ones (the details of the image); coefficients above the threshold represent the details of the image. In contrast, coefficients below the threshold represent noise and are therefore disregarded. In this way, a better-denoised image is obtained to LPF. We use the Michak Filter (Wiener Filter applied in the wavelet domain) as the denoising filter. The equation of the Wiener filter in the pixel domain is :

$$\tilde{f[n]} = A(S[n] - \mu_f[n]) + \mu_f[n] \tag{2.10}$$

where $A = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_N^2}$ represents the coefficient of the filter,
μ_f represents the mean of the noise-free image,
σ_f^2 and σ_N^2 represent the variances of the noise-free image and the noise, respectively,
S represents the observed image. Since we do not observe the noise-free image, it is impossible to calculate σ_f^2, so in practice, a slight variation of the Wiener filter, called Lee's Filter, is used. If we consider noise as zero mean and independent from the signal, then the following holds:

$$\mu_S[n] = \mu_f[n] \tag{2.11}$$

$$\sigma_S^2 = \sigma_f^2 + \sigma_N^2 \tag{2.12}$$

Therefore, the equation of the filter can be re-written as:

$$\tilde{f[n]} = max\left(\frac{0, \sigma_S^2 - \sigma_N^2}{\sigma_S^2}\right)(S[n] - \mu_S[n]) + \mu_S[n] \tag{2.13}$$

The rationale behind the above equation is that if the noise variance is low, the filter's coefficient is one and our estimated denoised signal is the observed image itself. On the other hand, if the noise variance is high, the filter's coefficient becomes zero, and the denoised signal is simply the mean of the observed image.

PRNU Computation

Once we get the precise estimation of the denoised image, the PRNU is then estimated maximizing the log-likelihood, and the equation becomes :

$$\widehat{P_N} = \frac{\sum_{k=1}^{K} W_k S_k}{\sum_{k=1}^{K} S_k} \tag{2.14}$$

If single image is considered,

$$\widehat{P_N} = \frac{W}{S} \tag{2.15}$$

We use a single image in computing the fingerprint rather than several images for two reasons. One, we do not have flat-field images. Second, we want to understand if the underlying problem is image-specific or not.

2.1.2 PRNU Detector

The fingerprint detection problem involves binary hypothesis testing. The tests are performed to link an image to one of the reference cameras. We consider a set of reference fingerprints (P_i') corresponding to the different cameras and compute the fingerprint of the image under test (P_N'). For each camera, the outcome of the trial depends on the deciding between the two hypotheses H_0 and H_1 :

$$H_0 : SP_N' + N, P_i' \neq P_N' \tag{2.16}$$

$$H_1 : SP_i' + N, P_i' = P_N' \tag{2.17}$$

if H_1 holds, the test image is taken by the reference camera
if H_0 holds, then the test image is not taken by the reference camera. The outcomes of the above tests are governed by the metrics: Normalized Correlation (ρ) Fridrich *et al.* (2006), and Peak Correlation Energy (PCE) presented by Costa *et al.* (2012). These metrics give the similarity scores of the two images. To compute these metrics, we perform two strategies: Pair-wise Noise Residual and Pair-wise Fingerprint Comparison, where we correlate two Noise Residuals and two PRNU terms, respectively, corresponding to two images.

Normalized Correlation

Normalized Correlation between two images is computed in the following way :

$$\rho = \frac{< W_1, W_2 >}{\|W_1\|\|W_2\|} \tag{2.18}$$

$$< W_1, W_2 > = \|W_1\|\|W_2\| \cos\theta \tag{2.19}$$

where W_1 and W_2 are the Noise Residuals of the two images, with one of them serving as the reference. Thus, ρ is $\cos\theta$ or the cosine distance. If the two residuals are similar, $\theta = 0$ and $\cos\theta = 1$, therefore $\rho = 1$, on the other hand if the two residuals are orthogonal , $\rho = 0$. A similar expression holds when we compare the two PRNU terms *i.e.,* P_{N1}' and P_{N2}'.

$$\rho = \frac{< P_{\mathrm{N1}}', P_{\mathrm{N2}}' >}{\|P_{\mathrm{N1}}'\|\|P_{\mathrm{N2}}'\|} \tag{2.20}$$

$$< P_{\mathrm{N1}}', P_{\mathrm{N2}}' > = \|P_{\mathrm{N1}}'\|\|P_{\mathrm{N2}}'\| \cos\theta \tag{2.21}$$

ρ values are then compared to the threshold (T) , if $\rho > T$, the two images are likely taken by the same camera, otherwise not. The threshold is chosen in order to minimize the probability of the false alarm. Threshold is calculated as :

$$T = \sqrt{\frac{2}{L}} \text{erfc}^{-1} \left(2 \cdot FPR\right) \tag{2.22}$$

where FPR refers to the False Positive Rate : no of times detector predicts H_1 as true, instead of H_0,
L refers to the size of the sensor.
If we assume noise to be i.i.d, then for the H_0 and H_1 hypothesis, the ρ values behave as Gaussian distribution, with zero mean and the other with some positive mean, respectively. The variance of ρ, when the image belongs to a different camera, does not depend on the content of the image; the same does not hold for H_1 when the images come from the same camera.

$$H_0 : \sigma_\rho^2 = 1/L \tag{2.23}$$

$$H_1 : \sigma_\rho^2 = \frac{\sigma_N^2}{\|SP_N\|^2 + L\sigma_N^2} \tag{2.24}$$

Therefore, under H_1, the mean and the variance depend on the content of the image.

Peak Correlation Energy
Peak Correlation Energy or PCE is defined as the ratio of peak energy of the two Noise Residuals to the average energies. It involves computing cross-correlation between the two residuals or PRNU terms for every shift. At a certain shift, when the two residuals or the two PRNU terms align, we get a substantial peak, while for other shifts, we get random behavior (low values). We take the energy of the value. The resulting value is, therefore, significantly large.

On the other hand, if the two residuals do not align for any shift, we have random values for all shifts; the PCE is 1. PCE is more robust than ρ and does not depend on the content of the image. Mathematically,

$$\text{PCE} = \frac{max_s \rho\left(s\right)^2}{\frac{1}{N-|W|} \sum_{s \notin W} \rho\left(s\right)^2} \tag{2.25}$$

where W is the region around which maximum is expected.
Unfortunately, there is not an analytical model as far as PCE is concerned.

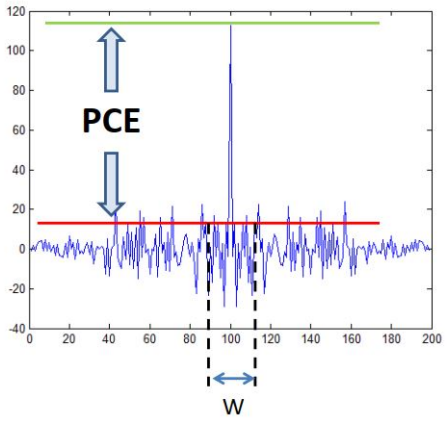


Figure 2.2: PCE Metric

Thresholds have been computed empirically, and the most commonly used threshold for PCE is 60. So if PCE > 60, the two residuals and hence the images belong to the same camera.

2.2 SPAM Features

Steganography is the process of hiding information such as text *etc.*, in an image. Steganography is done by adding an independent noise signal, stego noise, to the Cover Image (Original Image). The final image, which contains confidential information, is called Stego Image. Due to the addition of noise signal, there are no significant changes in the image's content. The human eye can not differentiate between the stego image and the cover image. The embedding of information is done at the pixel level. LSB (Least Significant Bit) steganography is one of the techniques to hide information, where the least significant bit of every pixel is modified. As a result of this modification, pixel intensity only changes by one, *e.g.,* 0, which represents black becomes 1, less dark. This does not have any visual impact on the content and color of the image. Figure 2.3 shows the simple process of LSB steganography. A simple text message, *Fingerprint Collision*, is added to the cover image. The stego image is similar to the cover image. This confidential information is not limited to text only; images, *etc.,* can also be hidden within the image. To identify the hidden content, Steganalysis is done.

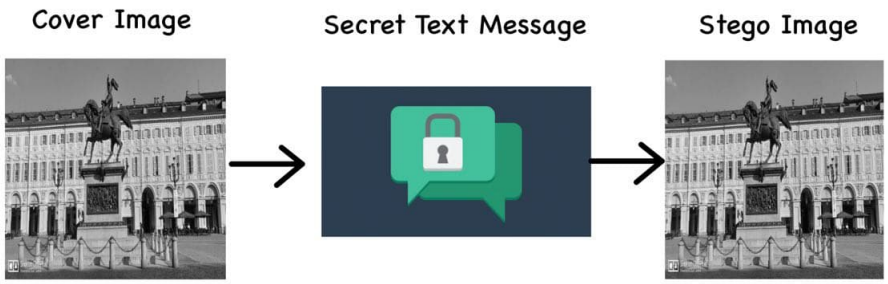


Figure 2.3: Steganography

In this project, the confidential information is synonymous with NUA's (Non-unique Artifacts) introduced by features associated to computational photography. We aim at detecting the presence of such artifacts with SPAM Analysis. Figure 2.4 shows the pipeline of Steganalysis with SPAM.

SPAM or Subtractive Pixel Adjacency Matrix, Pevny *et al.* (2010) , as the name suggests involves subtracting the adjacent pixels in the image. The differences are calculated along eight directions *viz,* horizontal directions $(\leftarrow, \rightarrow)$, verticals $(\uparrow, \downarrow)$, and the diagonals $(\nearrow, \swarrow, \searrow, \nwarrow)$.

$$D^{\rightarrow}(m,n) = S^{\rightarrow}(m,n) - S^{\rightarrow}(m,n+1) \tag{2.26}$$

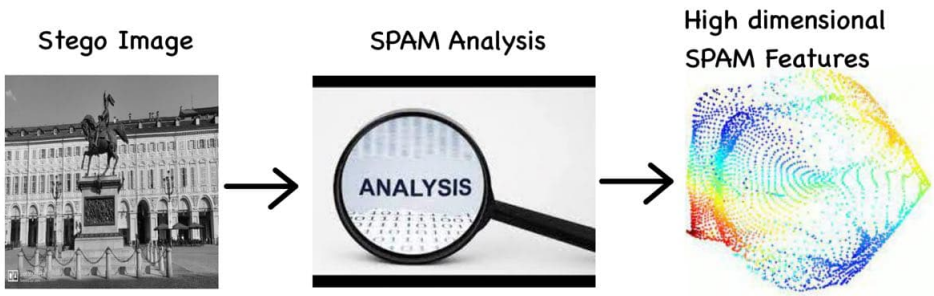


Figure 2.4: Steganalysis by SPAM

where $D^{\rightarrow}$ represents the difference matrix in horizontal ($\rightarrow$) direction. We can repeat the same procedure for every direction. As a result, the original image matrix (S) is replaced by the difference matrix D. The rationale behind the pixel differences is that it suppresses the image content and exposes the stego noise. The difference kernel $[-1+1]$ is the simplest edge detector, and the difference matrix essentially represents the high pass version of the image. Since natural images are smooth, the distribution of differences approximates Gaussian behavior, *i.e.,* differences peak around zero, representing the mean value, and quickly fall on either side. Therefore, it is quite realistic to consider the differences in the smaller window $[-W+W]$.
The pixels in the difference matrix are then modeled as 2^{nd} order Markov Chain (M) and transition probability matrices are calculated along each direction.

$$M^{\rightarrow}_{\mathrm{xyz}} = P\left(D^{\rightarrow}_{\mathrm{m,n+2}} = x | D^{\rightarrow}_{m,n+1} = y | D^{\rightarrow}_{m} m, n = z\right) \tag{2.27}$$

where $M^{\rightarrow}$ refers to the Markov Chain in the horizontal direction,
$x, y, z \in [-W, +W]$.
Finally,the probability matrices are averaged to obtain the SPAM features (S_F).

$$S_{\mathrm{F1}} = \frac{1}{4}\left[M^{\rightarrow} + M^{\leftarrow} + M^{\uparrow} + M^{\downarrow}\right] \tag{2.28}$$

$$S_{\mathrm{F2}} = \frac{1}{4}\left[M^{\nearrow} + M^{\nwarrow} + M^{\swarrow} + M^{\searrow}\right] \tag{2.29}$$

where S_{F1} refers to the SPAM features in horizontal and vertical directions,
S_{F2} refers to the SPAM features along diagonals, the final SPAM feature set S_F is $[S_{F1}; S_{F2}]$. For an image, 686 2^{nd} order SPAM features are obtained.

The SPAM features obtained are microscopic, i.e., they capture minute dependencies between the pixels of an image. The relations or dependencies between the pixels in the stego image are modified due to the embedding of hidden information. SPAM features capture this information, and therefore the features are distinct for stego and cover images. Once we extract SPAM features, a classifier is trained to learn the features corresponding to the cover and stego images, respectively. However, because of the high dimensionality of the SPAM features, we resort to PCA before training the classifier.

2.3 Principal Component Analysis (PCA)

Principal Component Analysis or PCA is one of the data transformation techniques in which data is re-represented or redefined in a new coordinate system. The axes of the new coordinate system are called Principal Components

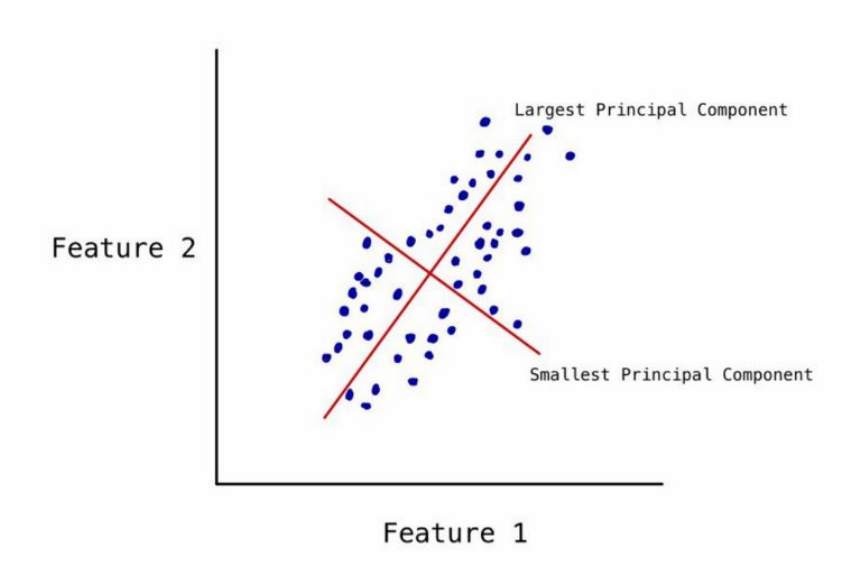


Figure 2.5: $2D$ PCA Components

$(P_{c1}, P_{c2}...P_{cn})$, form the basis of the new vector space. The goal of the transformation is to find new axes that better represent the data and cluster the data groups. The aim is to capture maximum data variance and simultaneously minimize the error between the data points and their projections. The importance of the components depends on the increasing order of their variances.

Figure 2.5 shows the distribution of the features along with $2D$ PC's. The data has a larger variance along the main diagonal and less along the other. The largest Principal Component captures the maximum variance of the data while the smallest component, orthogonal to the first, captures the lowest. These two components become the new basis for the two features. For n features, n components are computed.

The consequence of this transformation is dimensionality reduction or feature selection. We implement PCA to eliminate the curse of dimensionality of the data. The 2^{nd} order SPAM features of an image consist of 686 attributes or dimensions. This represents a very high-dimensional data, limiting the performance of machine learning algorithms. It is not easy to find patterns and cluster high-dimensional data. Moreover, these number of features represent a lot of computational burden. Therefore, we transform 686 SPAM features into 686 Principal Components. Then we make the feature selection; we keep only the essential components ($<<$ 686) depending on the accuracy of the classifier. Because of

the PCA, our classifier achieves very high accuracy and has a low computational cost.

Chapter 3

Related Work

In this chapter, we introduce the underlying problem and the recent advancements in the field of PRNU-based camera identification. Before introducing the problem, we highlight some terminologies to simplify the understanding of the problem.

Terminologies
The word *Model* refers to a particular smartphone, *e.g.,* Samsung $S10$ is a Smartphone Model.
The word *Devices* refers to the different exemplars of the same model, *i.e.,* Different Devices correspond to different instances of Samsung $S10$, each instance belonging to a different Flickr *User*. In other words, different users correspond to the different devices of the same model. Therefore, we interchangeably use the word Devices with Users.
Finally, the word *Resolution* refers to the size of the sensor, which shoots the image. Modern smartphones have more than one sensor to shoot the image. Throughout the thesis, we will make use of these terminologies.

3.1 Problem Statement

Camera Identification for modern smartphones turns out to be quite challenging. Ideally, PRNU patterns (fingerprints) of the images belonging to different devices are strongly uncorrelated. However, this no longer holds for the recent smartphones. The problem with modern smartphones is that correlation between PRNU patterns corresponding to the images of different devices (same model) results in unexpected distribution. The correlation values easily surpass

the threshold and overlap with auto-correlation values to some extent. In other words, the PRNU fingerprints of different devices collide with each other *i.e.*, the fingerprints are no longer unique; instead, they have an element of similarity. Therefore, high False Alarm Ratios (FAR's) are recorded for these devices. This type of inconsistent behavior leads to erroneous source identification. In such a case, one can wrongly link an image to multiple devices.

This unexpected behavior is verified for both the metrics *viz,* Normalized Correlation and PCE. These metrics reveal a high FAR. As far as normalized correlation is concerned, the distribution no longer follows a zero-mean Gaussian distribution. Moreover, the variance model for the H_0 hypothesis is not verified. An in-depth analysis reveals that the distribution can be modeled as a bi-modal distribution (Gaussian Mixture), with one mode close to zero mean and the other with a relatively higher mean.

This problem of fingerprint collision in modern smartphones raises a question on the uniqueness and reliability of the PRNU for Image Forensics. The use of PRNU as such in practical scenarios, *e.g.,* court evidence, will lead to misleading conclusions and drastic consequences.

Approach

This problem is under-researched. To the best of our knowledge, no solution to address the problem has been reported to the date of submitting the thesis. We propose two proactive solutions to address the problem of fingerprint collision in recent smartphones. The solutions are discussed in chapter 5. These solutions help us identify the images that produce unexpected distribution and, consequently, high FAR's. Those images are therefore disregarded as for as PRNU extraction is concerned. Instead, the PRNU is computed from the images producing expected distribution and no FAR's. The PRNU computed in such a way is without distortions and hence does not produce any erroneous results. In this way, Camera Identification can be performed reliably.

3.2 Recent Findings

The problem of fingerprint collision was first reported by Iuliani *et al.* (2021) in **A Leak in PRNU Based Source Identification—Questioning Fingerprint Uniqueness**. Iuliani and his team's work focuses on issues associated with PRNU-based SCI (Source Camera Identification). The experiments are performed on three datasets: VISION, CONTROL, and Flickr. VISION dataset is a

popular dataset used to analyze the efficiency of forensic algorithms. It consists of images from slightly old (released before 2016) smartphones. There are 35 different devices corresponding to 11 brands like Samsung, Apple, Huawei, Lenovo, LG, *etc.*

On the other hand, the CONTROL dataset consists of images of recent smartphones (released after 2016). The team first verifies the efficiency of PRNU source identification on the VISION dataset. They analyze 20 images per device. The fingerprint is estimated using a single image, and PCE is computed between every fingerprint couple. The team reports entirely consistent results as far as False Alarm Ratio is concerned. They obtain a significantly low FAR, less than 0.001. This strategy, where the fingerprint is estimated using a single image only, does not affect FAR; it only affects the detection rate. The same procedure is repeated for the CONTROL dataset. However, this time fingerprints are estimated from 5 flat-field images. On computing PCE between every fingerprint couple, the team reports quite inconsistent results. High false alarms are seen among Samsung, Xiaomi, and Huawei devices. This behavior is seen both at the model and brand levels, which means that different devices of the same model and different models of the same brand suffer from this behavior. The team then verifies this behavior on the Flickr dataset. They collect images of devices corresponding to the CONTROL dataset and images from other models as well. Repeating the same analysis on the Flickr dataset verifies the problem of fingerprint collision in modern smartphones. The correlation between Pair-wise Noise Residuals exposes a high percentage of false alarms for these smartphones. Iuliani and his team do not find any link between meta-data and this behavior. The team fails to provide any solution to counter the problem.

Joshi *et al.* (2020) evaluate the robustness of PRNU-based fingerprint considering different Imaging Pipelines in **Empirical Evaluation of PRNU Fingerprint Variation for Mismatched Imaging Pipelines**. The team analyses the problem of fingerprint mismatch across different pipelines. Imaging pipeline refers to the set of algorithms performed on an image to produce the final digital image. These algorithms are performed by Imaging Signal Processor, a DSP for Image Processing. Joshi and his team stress the growing use of neural nets in modern cameras and its impact on source identification. The team analyses 13 different pipelines that include 10 standard pipelines like Adobe Lightroom, Corel AfterShot Pro, *etc.,* and 3 neural network-based pipelines. The neural net-based architectures employ Dnet and Unet for color interpolation and denoising. They also include an Inet to produce images similar to standard imaging pipelines. The

team collects 120 raw sensor images from 4 Camera's: Nikon $D7000$, Canon EOS $40D$, Nikon $D90$ and Canon EOS $5D$. The images are collected at maximum resolution. These images are processed using the different pipelines mentioned above. Default settings and minimum post-processing are applied on each pipeline so that the resulting image is not significantly different across pipelines. The team aims at analyzing fingerprint mismatches locally and globally. PRNU fingerprints are estimated corresponding to the same images processed by different pipelines. Correlation is performed between these fingerprints. One expects to have high correlation scores because they compare the PRNU's corresponding to the same images.

On the contrary, the team reports significantly lower detection scores. The average PCE score is reduced by 62% between images processed by other pipelines. To assess the local impact, the group extracts non-overlapping square patches of varying size *viz,* 128×128, 256×256, and 1024×1024 from these images. The group observes a high reduction in detection scores for smaller image patches, 128 by 128 pixels, which are used to detect forgery, than for the larger ones 1024 by 1024 pixels. Also, lower detection rates are seen for neural network-based pipelines compared to standard pipelines. However, the team does not report any solution to counter the problem.

Liu *et al.* (2021) propose CNN-based Camera Identification in **Efficient Source Camera Identification with Diversity-Enhanced Patch Selection and Deep Residual Prediction**. The use of deep learning for Camera Identification is growing faster than ever. However, the cost of complexity and generalization remains the bottleneck for deep-learning-based methods, as Liu and his team report. To tackle the cost of complexity, the team presents a simplified model where only specific image patches of the size 64 by 64 pixels or higher are used for training. Instead of using all image patches for training, the team uses some arbitrary patches. The main focus of the research is the selection of these patches, *i.e.,* identifying the candidate patches for training. The team employs more than one criteria to select the patches *viz*, edge, and textual criteria based on statistical properties of patches and semantic content. This way, the representative patches better model the distribution of overall patches.

After selecting the patches, Noise Residuals for these patches are estimated with the help of Res2Net. Res2Net aims at reducing the image content in the residual. The final step involves using a classifier module to learn the camera-specific features. The tests are conducted on the Dresden dataset at the device, model, and brand levels. The results obtained are similar to the state of art

algorithms. However, this method is data-driven and computationally costly. Also, the Dresden dataset is old, and the corresponding devices use obsolete image processing algorithms. There is not any problem of fingerprint collision in the Dresden dataset. Therefore, given the complex in-camera processing modern smartphones employ, this method is likely not going to generalize. Moreover, this method gets tricky if dark, bright, and smooth patches are used for training.

de Roos & Geradts (2021) describe the factors that affect the efficiency of PRNU based SCI (Source Camera Identification) in **Factors that Influence PRNU-Based Camera-Identification via Videos** . The tests are performed on videos of iPhone 6. The identified factors are resolution, length, and compression. Two different resolutions are tested : $720p$ and $1080p$. To achieve video compression, Snapchat is used. The team observes that a higher resolution and video length guarantees reliable PRNU estimation. However, compression harms PRNU estimation.

Nowadays, more research is being done to protect users' privacy whose images are used for fingerprint estimation. Since PRNU's extraction requires several images of a specific camera, it leaks a considerable amount of user information, violating privacy. Fernández-Menduiña & Pérez-González (2021) present various strategies to counter the leakage of information associated with PRNU estimation. The methods include using flat-field images (sky, *etc.*), modifying the PRNU fingerprint, and estimating PRNU from encrypted images. However, this increases the computational complexity and also affects the performance.

Chapter 4

Dataset

In this brief chapter, we present the data that we collected and analyzed. The data consists of images. We also discuss the assumptions concerning the dataset.

4.1 Details

Manufacturer	Model	Resolution	Users	Images
Huawei	LYA-L29	3648×2736	10	700
	LYA-L29	5120×3840	7	490
	LYA-L29	7296×5472	2	140
	VOG-L29	3648×2736	10	700
	VOG-L29	7296×5472	2	139
Samsung	SM-G973u	4032×3024	6	378
	SM-G973u	4608×3456	4	279
	SM-G975u	4032×3024	8	558
	SM-G975u	4608×3456	2	140
	SM-A505u	4032×3024	3	209
	SM-G970u	4032×3024	5	350
Xiaomi	Redmi Note 7	4000×3000	8	560

Table 4.1: Details of the Dataset

Our dataset consists of images from 3 different smartphone manufacturers. These images correspond to Huawei, Samsung, and Xiaomi smartphones. We collect images of two models of Huawei , LYA-L29 (Mate 20 Pro) and VOG-L29 (P30 Pro) , four models of Samsung SM-G973u (S10), SM-G975u (S10+), SM-A505u

(A50) and SM-*G*970u (S10e) and one model of Xiaomi, Redmi Note 7. Since these smartphones feature more than one camera sensor, it is possible to shoot images at different resolutions. So, we have more than one resolution for the majority of these smartphones.

Table 4.1 shows the details of the dataset wherein, Manufacturer refers to the brand of the smartphone, the Model refers to a particular smartphone, Resolution refers to the size of the camera sensor. Users refer to the different devices or instances of the same model. Each user corresponds to a different device (same model). The overall structure of the dataset is as follows :

$$Manufacturer \rightarrow Model \rightarrow Resolutions \rightarrow Users \rightarrow Images$$

The images corresponding to a particular resolution are uniformly distributed among these users. For example, for LYA-L29, we have 700 images uniformly distributed among 10 different devices at 3648×2736 resolution. This is true for every model. The average number of images per user is 70. These number of images ensure diversity in the data primarily because it is improbable that these 70 images are shot under the same camera settings, *e.g.*, ISO, Exposure, Focal Length, *etc.* Meta-data analysis with MATLAB testifies the diversity of the data. Secondly, collecting images of different resolutions for most of the models allows us to draw broader conclusions. Finally, these images are rich in content as well. The images correspond to the random subjects, *e.g.*, sky, table, cat, *etc.*, which represent real-life situations. Therefore the dataset is quite diverse and realistic simultaneously. This variety ensures that whatever conclusions hold for this dataset can be generalized for any other realistic dataset.

4.2 Reference and Assumptions

This choice of the dataset is based on the results presented by Iuliani *et al.* (2021). Given the timing constraints of the project, we collect a total of 4643 images only, corresponding to 7 different models. The images are downloaded from the Flickr database using a Python script. Since we do not have physical possession of the devices, we resort to meta-data to ensure the genuineness of the data. Meta-data analysis is used to verify the authenticity of the image. Figure 4.1 reveals the Meta-Data Content of one of the images belonging to a Flickr user shot on Redmi Note 7. The authenticity of the image is verified by looking at the *Model* tag in the meta-data. Moreover, the resolutions of the images match the device's capabilities.

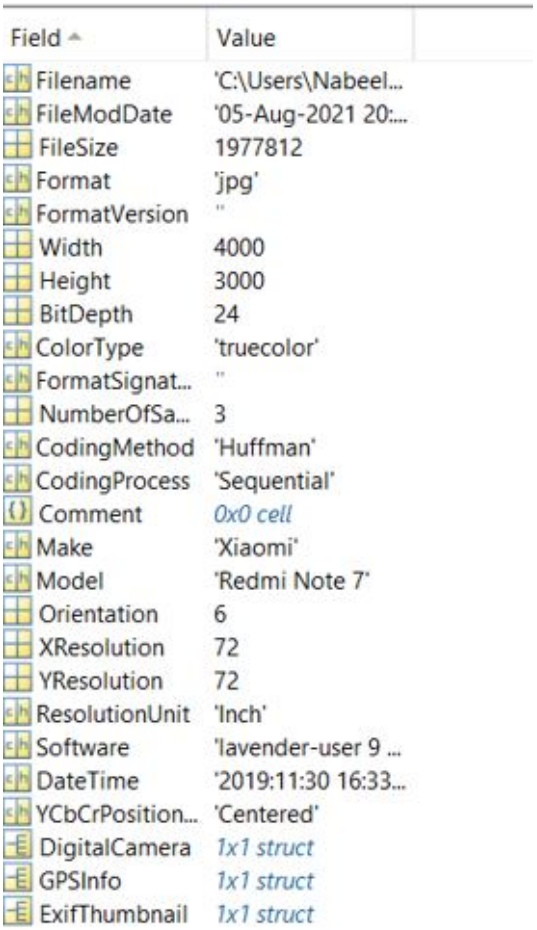


Figure 4.1: Meta-Data Content with in-built MATLAB tool

Most importantly, looking at *GPSInfo* tag of the meta-data, it is safe to say that different users correspond to different devices (of the same model) since the coordinates corresponding to different users represent different regions of the world. We verify this for Redmi Note 7 and believe that the same holds for other models. Images of the same user reveal similar GPS coordinates, while images from different users show significantly different coordinates. For example, for Redmi Note 7, the GPS coordinates of the images shot by one of the users reveal the location around the Mediterranean sea. In contrast, the GPS tag reveals locations around Poland for another user of the same model. This is one of the critical assumptions for the dataset. All our conclusions are based on this assumption. The dataset is then analyzed in MATLAB with the tools discussed

in chapter 5.

Chapter 5

Methodology and Algorithms

This chapter presents a model and the associated hypothesis that explains fingerprint collision in the modern smartphones. We discuss two problem verification tools that expose the unexpected behavior of modern smartphones. Finally, we discuss two algorithms and their pipelines to tackle this problem.

5.1 Methodology

Model Assumptions

The problem of fingerprint collision occurs in most modern smartphones. We suppose that this problem occurs due to complex in-camera processing, which is quite common these days. Camera features like Portrait, HDR10+, AI, Night Mode, EIS (Electronic image Stabilization), *etc.*, require a lot of processing and that too, within seconds. The portrait effect is achieved with the help of machine learning (edge detection) by separating the foreground and blurring the background. HDR is achieved by capturing n number of images with different shutter speeds and combining them into a single image to get a wide dynamic range close to a Human Eye. Because of this complex processing, final images are significantly different from the raw sensor images. We believe that the processing, *i.e.*, the algorithms behind these features, is somewhat common to the same model devices. *e.g.*, all the devices of a particular Samsung Model (*e.g.*, S10) likely use similar if not the same algorithms to produce these effects. In the context of our terminology, different users of the same model experience similar camera processing. Therefore, this common processing introduces non-unique artifacts (NUA's) in the images of these users and consequently distorts the actual fingerprint. As a result, PRNU patterns and Noise Residuals belonging to different devices exhibit

higher correlation metrics, leading to erroneous source identification.

Model

Given the above discussion, the PRNU and the Noise Residual terms are modified with respect to section 2.1.1 in the following way :

$$S' = S_0 + S_0 P'_N + N + NUA's \tag{5.1}$$
$$W' = S' - S_0 \tag{5.2}$$
$$W' = W + N + \epsilon \tag{5.3}$$
$$P''_N = P'_N + N' + \epsilon' \tag{5.4}$$

where S' represents the output image,
NUA's refers to the non-unique artifacts,
$W = S_0 P'_N$ and P'_N refer to the ideal Noise Residual and PRNU term, respectively,
N and N' represent i.i.d. Noise Components independent for each image,
ϵ and ϵ' represent small perturbation due to non-unique artifacts and
P''_N and W' refer to the modified PRNU and Noise Residual terms, respectively.

We estimate Noise Residuals and the PRNU terms from a single image. So, each estimation of the terms is affected by an independent noise component corresponding to the image. Since the PRNU estimation is slightly complex, the associated noise term N' is different from N. We correlate the PRNU terms and Noise Residuals of two different images. So the independent noise components corresponding to two different images cancel each other and do not contribute to false alarms. Instead, ϵ and ϵ' are responsible for fingerprint collision.

As described in section 2.1.1 the most crucial step in fingerprint computation is the estimation of the denoised image. We believe that the denoising filter cannot suppress the NUA's term completely. Therefore, the denoised image still contains small traces of the NUA's. These small perturbations manifest as ϵ and ϵ' in W' and P''_N, respectively. PRNU patterns and the Noise Residuals corresponding to the devices of a specific model suffer from the same non-unique perturbation. The presence of ϵ and ϵ' in all devices of the same model distorts the uniqueness of the W' and P''_N and increases the element of similarity among them. This results in unexpectedly high correlation scores between different devices. Since the PRNU extraction involves an additional step of demodulating the residual, it is likely that $\epsilon' < \epsilon$.

The first step involves verification of the problem. Given the simplicity and time constraints, we limit ourselves to intra-resolution and intra-model analysis,

i.e., we analyze different devices/users at the same resolution and the same model. Problem analysis is done in MATLAB using the tools discussed below.

5.2 Problem Verification Tools

To verify the problem of fingerprint collision, we propose two strategies: Pair-wise Noise Residual (Strategy 1) and Pair-wise Fingerprint Comparison (Strategy 2). The basics of these strategies are already discussed in section 2.1.2.
Under the strategy 1, we compute and correlate Noise Residuals (W') of two images at a time. Noise Residuals are computed using Denoising Filter. After calculating the Noise Residuals, Wiener Filtering is applied to remove artifacts (blockiness) due to JPEG. Then we correlate the Noise Residual of an image belonging to a specific user with the Noise Residuals of every image belonging to different users, including the former (auto-correlation).
Steps :

- Compute Noise Residual of 1^{st} Image
- Compute Noise Residual of 2^{nd} Image
- Normalized Correlation and PCE between the two Residuals
- Loop over all Images.

Under the strategy 2, we compute and correlate fingerprints (P''_N) of two images at a time. Fingerprint computation is described in section 2.1.1. Rest of the steps are same as strategy 1.
e.g., for Normalized Correlation (ρ), we do the following :

Strategy 1:

$$\rho = \frac{< W'_1, W'_2 >}{\|W'_1\|\|W'_2\|} \tag{5.5}$$

Strategy 2:

$$\rho = \frac{< P''_{N1}, P''_{N2} >}{\|P''_{N1}\|\|P''_{N2}\|} \tag{5.6}$$

Under each strategy, we obtain two symmetric matrices corresponding to Normalized Correlation (ρ) and PCE. These matrices reveal the auto-correlation and

cross-correlation scores between the devices of the same model. For Normalized Correlation, the diagonal elements are equal to 1, correlation of an image with itself. Due to the presence of ϵ and ϵ' in the Noise Residuals and the fingerprints, respectively, we see elevated correlation scores between different users. We analyze the problem for every model and every resolution. This analysis is discussed in detail in chapter 6.

5.3 Algorithms

We propose two algorithms to tackle the problem of fingerprint collision: SPAM Classifier and Meta-Data SceneType Tag Classifier.
The analysis tools discussed in section 5.2 testify the problem of fingerprint collision. However, an in-depth analysis reveals that some images do not suffer from fingerprint collision. The distribution of correlation values corresponding to these images is close to the ideal behavior. These images do not produce any false alarms. In our terminology, we call them *Good Images*. The ρ values corresponding to these images approximate a zero-mean Gaussian distribution. We believe that these images do not suffer from NUA's and therefore do not behave according to the model introduced in section 5.1. On the other hand, *Bad Images* are the ones that produce unexpected distribution and high false alarms. These bad images are responsible for fingerprint collisions and behave according to the model discussed in section 5.1.

Following observations hold for pair-wise correlations of good and bad images, corresponding to different devices (different users):

- Cross-correlation between PRNU patterns of two good images (different devices) results in a correlation value less than the threshold.
- Cross-correlation between PRNU patterns of one good image and one bad also yields a correlation value less than the threshold.
- However, the cross-correlation between two bad images produces an unexpected correlation value greater than the threshold.

Case 1: Two Good Images

$$P''_{N1} = P'_{N1} + N'_1 \tag{5.7}$$
$$P''_{N2} = P'_{N2} + N'_2 \tag{5.8}$$

Therefore,

$$\rho_{12} = \frac{< P''_{N1}, P''_{N2} >}{\|P''_{N1}\|\|P''_{N2}\|} < T \tag{5.9}$$

Case 2: One Good and One Bad Image

$$P''_{N2} = P'_{N2} + N'_2 \tag{5.10}$$
$$P''_{N3} = P'_{N3} + N'_3 + \epsilon' \tag{5.11}$$

Therefore,

$$\rho_{23} = \frac{< P''_{N2}, P''_{N3} >}{\|P''_{N2}\|\|P''_{N3}\|} < T \tag{5.12}$$

Case 3: Two Bad Images

$$P''_{N3} = P'_{N3} + N'_3 + \epsilon' \tag{5.13}$$
$$P''_{N4} = P'_{N4} + N'_4 + \epsilon' \tag{5.14}$$

Therefore because of ϵ',

$$\rho_{34} = \frac{< P''_{N3}, P''_{N4} >}{\|P''_{N3}\|\|P''_{N4}\|} > T \tag{5.15}$$

where indices 1 and 2 correspond to good images,
3 and 4 correspond to bad images,
T represents the detection threshold.
The same observations hold for the Noise Residuals. The crux of this discussion is that the correlation between a good image and any other always produces expected distribution. There are no false alarms at all. In contrast, due to the same perturbation (ϵ') in the bad images of different devices of the same model, the corresponding PRNU patterns exhibit elevated correlation values.

The intuition for the algorithms comes from the discussion above. The algorithms aim at identifying these good images. These images can then be used for unique and reliable fingerprint extraction. While the SPAM classifier is more accurate and applies to all models, the Meta-Data SceneType Classifier, on the other hand, is faster but does not apply to all models.

5.3.1 SPAM Classifier

This algorithm is based on the SPAM features discussed in chapter 2. These features capture complex dependencies between the pixels and identify the presence of hidden content in the images. However, here we are not interested in the traditional Steganalysis. Instead, we model NUA's (non-unique artifacts) as the confidential information in the images. So those images contaminated with NUA's are Stego Images, and those free from NUA's are Cover Images. We use the terms Bad Image and Good Image for the two classes, respectively, as described in section 5.3. Figure 5.1 shows the pipeline of the algorithm.

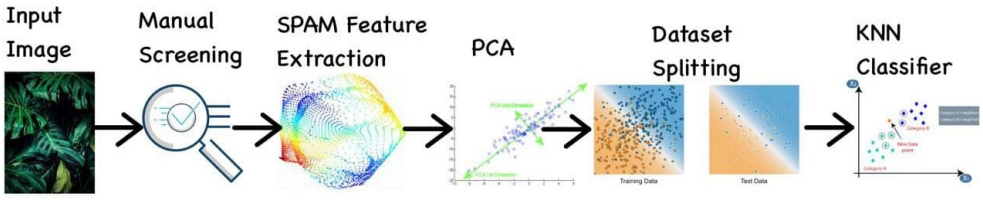
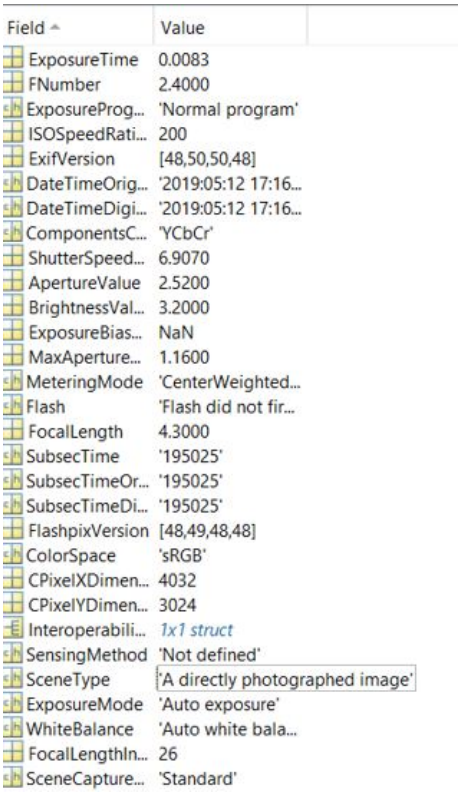


Figure 5.1: The pipeline of the SPAM Classifier

- The first step involves manual screening of good and bad images of the dataset. This is done by counting the number of false positives corresponding to an image. Those images which record 0 false positives are labeled as good images, while others are labeled as bad. These labels serve as *Ground-Truths* for the classifier.
- Then we extract the 2^{nd} order SPAM features from the good and bad images. These features capture the presence of NUA's. However, the SPAM features of each image consist of 686 dimensions. The dataset, therefore, does not scale with the dimensions of the SPAM features.
- We separately perform PCA on the SPAM features of the good and bad images to address the curse of dimensionality. We keep only N Principal Components depending on the accuracy of the classifier.
- The dataset is then divided into the training and test data.
- The fifth step involves training a classifier with the training data. The classifier outputs labels, good and bad, corresponding to the training data.

Finally, we test the accuracy of the classifier on the test dataset.

5.3.2 Meta-Data Screening: SceneType Tag Classifier



Field	Value
ExposureTime	0.0083
FNumber	2.4000
ExposureProg...	'Normal program'
ISOSpeedRati...	200
ExifVersion	[48,50,50,48]
DateTimeOrig...	'2019:05:12 17:16...
DateTimeDigi...	'2019:05:12 17:16...
ComponentsC...	'YCbCr'
ShutterSpeed...	6.9070
ApertureValue	2.5200
BrightnessVal...	3.2000
ExposureBias...	NaN
MaxAperture...	1.1600
MeteringMode	'CenterWeighted...
Flash	'Flash did not fir...
FocalLength	4.3000
SubsecTime	'195025'
SubsecTimeOr...	'195025'
SubsecTimeDi...	'195025'
FlashpixVersion	[48,49,48,48]
ColorSpace	'sRGB'
CPixelXDimen...	4032
CPixelYDimen...	3024
Interoperabili...	*1x1 struct*
SensingMethod	'Not defined'
SceneType	'A directly photographed image'
ExposureMode	'Auto exposure'
WhiteBalance	'Auto white bala...
FocalLengthIn...	26
SceneCapture...	'Standard'

Figure 5.2: Meta-Data of a Good Image

Meta-Data is a piece of text information automatically embedded in an image file. It is produced by the device capturing the image. The meta-data contains valuable information about the image such as Exposure, Shutter Speed, ISO, Focal Length, GPS info, Device ID, Manufacturer, Date, *etc.* Since the meta-data content contains all the image settings, we correlate these settings with the unexpected behavior. We observe the relation between the tags and correlation values in the scatter plots. Analysis of scatter plots reveals no link between the different tags and the unexpected behavior for Huawei and Xiaomi devices.
However, for Samsung Models, one of the meta-data tags, *SceneType: A directly photographed image,* correlates highly with the expected behavior. Figure 5.2 shows a typical meta-data content of an image with the presence of a SceneType tag. The images which record this tag do not suffer from false alarms. These images resemble good images. Therefore, the problem of fingerprint collision does

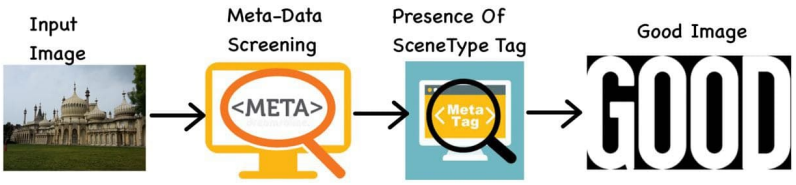


Figure 5.3: The pipeline of Meta-Data SceneType Classifier

not exist for such images. On the other hand, those images whose meta-data does not contain this tag produce unexpected distribution.

In comparison to the SPAM Classifier, the Meta-Data-based Classifier achieves lower accuracy. However, it does not require any training at all. Figure 5.3 shows the pipeline of the algorithm. We need to look for the tag in the meta-data. Once we find the presence of the tag, *A directly photographed image*, in the image, we are pretty confident that the image does not suffer from NUA's and can therefore be used for reliable fingerprint extraction. This type of screening is faster and requires less effort.
The accuracy of this algorithm is calculated by comparing its predictions with the ground-truths identified under manual screening (1^{st} step) of SPAM Classifier, presented in section 6.3. The results of this algorithm are described in subsection 6.5.3.

Chapter 6

Experiments

This chapter presents the experimental setup, the metric for evaluating the methods, and the analysis carried on the dataset. The critical goal of this chapter is to discuss the results concerning analysis tools and algorithms.

6.1 Setup

We perform the experiments in MATLAB, a famous work environment with excellent community support and tons of in-built functions. Moreover, for MATLAB, there exists a library of codes to compute the PRNU fingerprint and the SPAM features. Thanks to DDE Lab, Binghamton University, NY, for making these codes accessible to the community.

The data consists of images of recent smartphones whose details are presented in chapter 4. The first experiment involves an analysis of the problem with the verification tools proposed in chapter 5. Because of time constraints, we limit ourselves to intra-resolution and intra-model analysis, *i.e.,* we verify the issue at the resolution level within each model. In other words, we focus on inter-user analysis within each resolution. Then we analyze the meta-data content of the images. We also present a brief experiment with CFA (Color Filter Array) trace in the images. Finally, we discuss the investigations concerning the two methods.

6.1.1 Metric

To evaluate the performance of our methods, we use a standard metric, *Confusion Matrix*, popular for supervised machine learning tasks. Since the goal of our classifier is to predict the images correctly through supervised learning, the

confusion matrix is an ideal metric. The metric, just like other matrices, consists of rows and columns. The ground-truths and classifier-predicted labels are along the rows and columns, respectively. We can use it on the training as well as testing data. Usually, we use it on the test data. The size of the confusion matrix depends on the number of classes. Since we have only two classes, *viz,* good and bad, the size of the confusion matrix is 2×2.

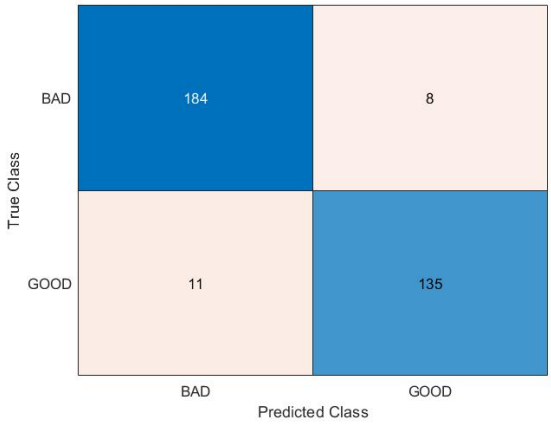


Figure 6.1: An example of Confusion Matrix

The matrix compares the classifier's results with the ground-truths and computes four quantities, two along one diagonal *viz,* True Positive Rate (TPR), True Negative Rate (TNR), and two along the other *viz,* False Negative Rate (FNR) and False Positive Rate (FPR). One of the two classes serves as a positive class; *e.g.,* in our case, the positive class refers to a set of good images. Therefore, TPR is the number of times the classifier can classify good images correctly.

On the other hand, the negative class refers to the bad images, and thus, TNR relates to the number of times the classifier can classify bad images correctly. FPR relates to the number of times the classifier incorrectly classifies bad images as good. FNR refers to the number of times the classifier incorrectly classifies good images as bad. TPR and TNR are accuracy measures, while FPR and FNR measure errors in the classifier's performance. Figure 6.1 shows an example of the confusion matrix. Rows indicate True Class, and columns indicate Predicted Class. In the example, TNR is $\frac{184}{192}$, TPR is $\frac{135}{146}$. On the other hand, the error rates corresponding to FPR and FNR are $\frac{8}{192}$ and $\frac{11}{146}$ respectively. In other words, 95.83% and 92.4% are the accuracies for each class. Approximately 4% and 8% are the associated error rates. The confusion matrix also helps us to understand

the balance of the dataset. In this case, the number of bad examples is more than good examples. Therefore the accuracy associated with the bad class surpasses the good. However, both accuracies are reasonably good, above 90%, so the data imbalance does not significantly impact the classifier's performance.

6.2 Initial Analysis

This section presents the results we get with the problem verification tools. In particular, we show the histograms and stats corresponding to Pair-wise Noise Residual and Pair-wise Fingerprint Comparison. This section exposes the problem of fingerprint collision.

6.2.1 Pair-wise Noise Residual Comparison: Strategy 1

Under this strategy, we correlate Noise Residuals corresponding to every image couple. Our focus is on the cross-correlation scores, *i.e.,* Noise Residuals corresponding to the images of different users. We use the two metrics, PCE and Normalized Correlation (ρ), to understand if the residuals are similar or not. Since the residuals are from different devices, we expect them to be different. We separately discuss the results concerning the Samsung Models.

Samsung Models

Figure 6.2 shows the histograms corresponding to four Samsung models. The first row of the figure depicts the histograms of SM-G973u (12 MP), SM-G973u (16 MP), and SM-G975u (12 MP), and the second row depicts histograms of the SM-G975u (16 MP), SM-G970u (12 MP), and SM-A505u (12 MP).

These histograms show the distribution of cross-correlation and auto-correlation scores in black and green, respectively. The X-axis depicts the Normalized Correlation values (ρ), and the y-axis represents frequency (ν). The histograms are probability normalized. A pink line parallel to the y-axis indicates the detection threshold, dependent on the size of the sensor.

In all the histograms, we see that many cross-correlation values surpass the threshold and overlap significantly with auto-correlation scores. Therefore, these models suffer from high false alarms. The cross-correlation values do not follow a zero-mean Gaussian distribution. Particularly, both the resolutions of Samsung $S10$ and $S10+$ suffer from the problem of fingerprint collision. We extract some statistical parameters to see the extent of fingerprint collision for each model.

6.2 Initial Analysis

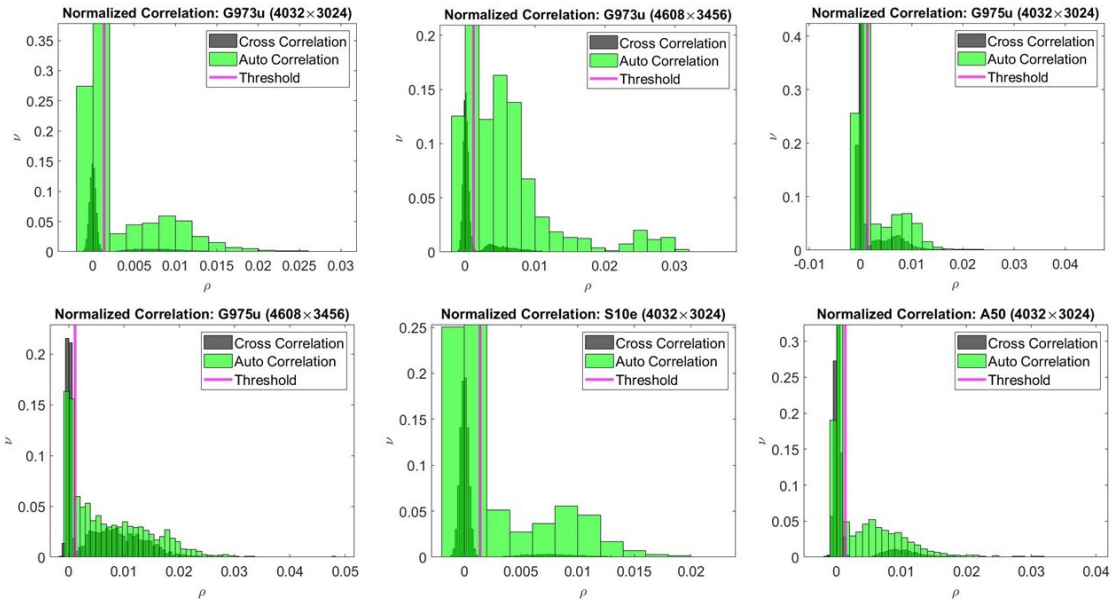


Figure 6.2: Normalized Correlation of Samsung Devices

Model	μ	Threshold	σ^2_{ex}	σ^2_{ac}	FAR
SM-G973u (12 MP)	0.0019	0.0014	$8.2e-08$	$1.75e-05$	21.6%
SM-G973u (16 MP)	0.0012	0.0012	$6.27e-08$	$7.931e-06$	19%
SM-G975u (12 MP)	0.0016	0.0014	$8.2e-08$	$1.178e-05$	22.6%
SM-G975u (16 MP)	0.0052	0.0012	$6.27e-08$	$3.80e-05$	54%
SM-G970u (12 MP)	$7.76e-04$	0.0014	$8.2e-08$	$6.5e-06$	9.44%
SM-A505u (12 MP)	$1.64e-04$	0.0014	$8.2e-08$	$1.85e-05$	14.67%

Table 6.1: Normalized Correlation Stats of the Samsung Models

Table 6.1 shows the cross-correlation stats of the Samsung Models, where μ refers to the mean value of cross-correlation, the Threshold is based on sensor size, σ^2_{ex} refers to the expected variance, σ^2_{ac} refers to the actual variance of the distribution and FAR refers to the false alarm ratio. MP refers to the Mega Pixels, *i.e.,* sensor size. The number of devices for each resolution is presented in chapter 4. We see that the actual variance is significantly higher than expected for each resolution of the model. The variance equation presented in section 2.1.2 is therefore not verified. Also, the mean of the distribution is substantially different from zero; it exceeds the threshold for all models except Samsung $S10e$

and $A50$. Above all, we record alarmingly high FAR's for all of these models. All the devices suffer from FAR's $> 9\%$. Therefore, the devices corresponding to each of these models exhibit significant non-unique artifacts in their images. Notably, Samsung G975u at a resolution of 4608×3456 (16 MP) suffers the most with FAR of 54%.

To have a deeper insight into the unexpected behavior of the Noise Residuals, we plot cross-correlation values only. Since we expect to see a Gaussian behavior, we fit a single-mode Gaussian distribution to the ρ values.

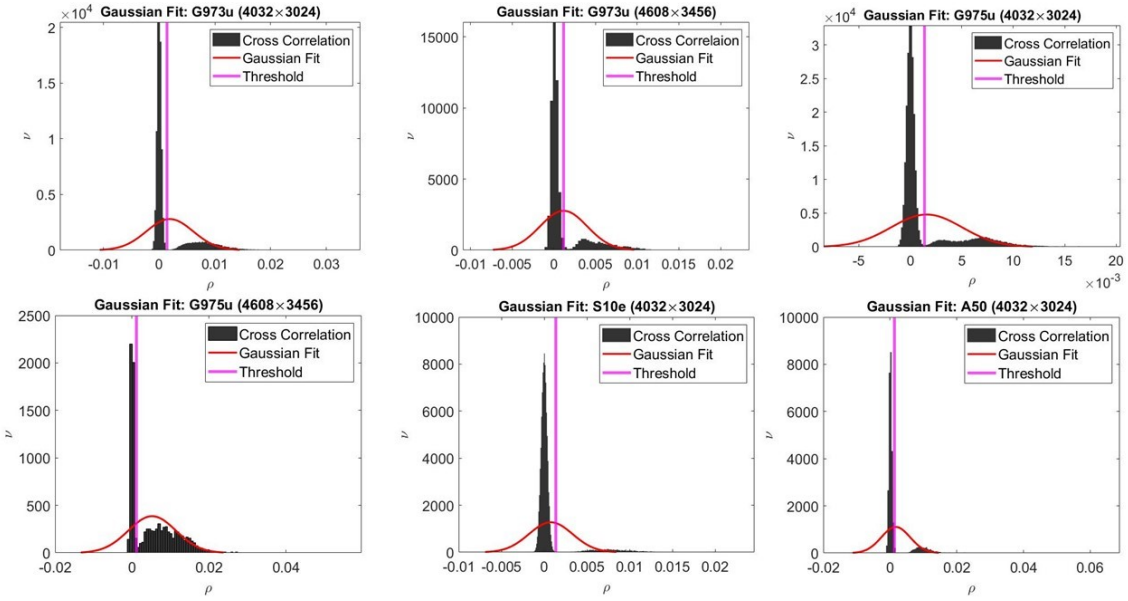


Figure 6.3: Gaussian Fit to Normalized Correlation

Figure 6.3 shows Normalized cross-correlation values with a Gaussian Fit distribution for each resolution of the model. The X-axis represents ρ values while the y-axis represents count or frequency. The mean and the variance of the Gaussian distribution more or less correspond to the values proposed in table 6.1. It can be seen that many correlation values exceed the pink line, which represents the threshold. Because of this unexpected distribution, a higher-mean Gaussian curve fits the correlation values instead of zero-mean. If we carefully inspect the correlation values of these models, we see that they suffer from a bi-modal distribution, one mode with nearly zero mean and the other with a higher mean. To visualize the two modes, we fit a bi-modal Gaussian curve to the ρ values.

6.2 Initial Analysis

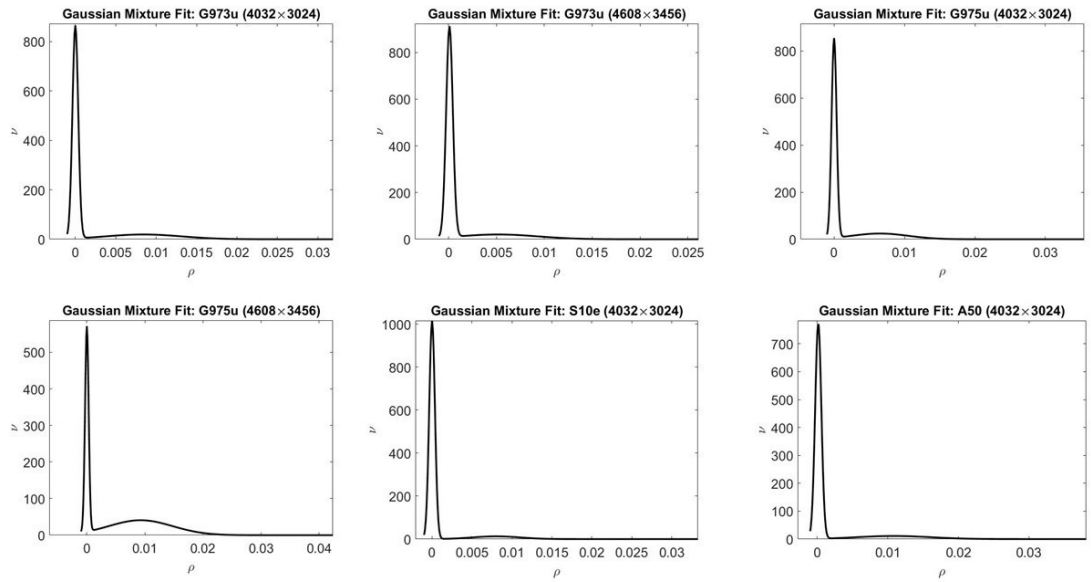


Figure 6.4: Gaussian Mixture Modelling

Figure 6.4 shows Gaussian Mixture Modelling (GMM) of ρ values for all the Samsung Models. GMM shows two peaks for each model: a more prominent peak close to 0 and a smaller one at a higher ρ value. The higher peak close

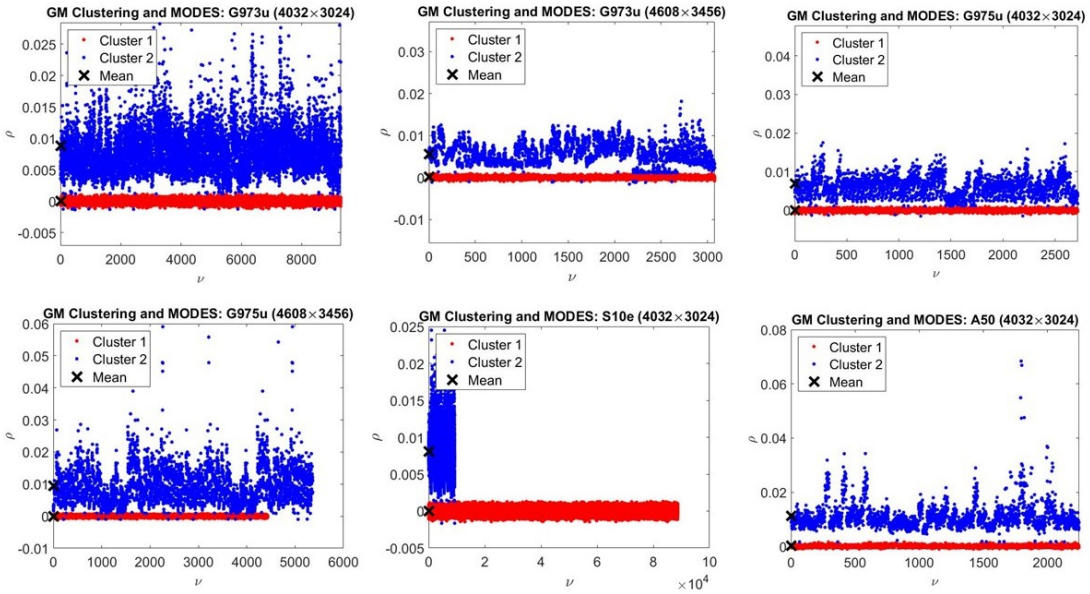


Figure 6.5: Gaussian Mixture Clustering

to the 0 value of ρ indicates that most cross-correlation values follow zero-mean Gaussian behavior. In comparison, a considerably lower peak suggests that some of the values do not behave as expected. Therefore, the two modes contribute in different proportions. The zero-mean mode contributes more to the Gaussian Mixture.

We can also see that the variance of the higher peak (lower mode) is lower and better approximates the expected variance. In contrast, variance corresponding to the higher mode is significantly higher. This unexpected higher-order mode leads to the problem of false alarms. All of the Samsung models, all resolutions suffer from this bi-modal distribution. We can easily visualize this with the help of clustering.

Figure 6.5 shows the associated clustering of ρ values using GMM. The red and blue dots indicate the two clusters, and the black crosses indicate the two modes of the Gaussian Mixture. The cluster corresponding to the higher mode has a larger variance. To further support our analysis, we extract mean and variance from the two clusters of the Gaussian Mixture. The details are shown in table 6.2, where indices 1 and 2 correspond to lower and higher modes. These stats support the above analysis.

Model	μ_1	μ_2	σ_1^2	σ_2^2
SM-G973u (12 MP)	$4.09e-06$	0.0084	$1.31e-07$	$1.98e-05$
SM-G973u (16 MP)	$9.18e-05$	0.0052	$1.24e-07$	$1.61e-05$
SM-G975u (12 MP)	$1.49e-06$	0.0065	$1.3e-07$	$1.44e-05$
SM-G975u (16 MP)	$1.20e-06$	0.0093	$9.84e-08$	$2.86e-05$
SM-G970u (12 MP)	$1.4729e-07$	0.0080	$1.2e-07$	$9e-06$
SM-A505u (12 MP)	$1.37e-04$	0.0108	$1.94e-07$	$2.7e-05$

Table 6.2: Stats of Gaussian Mixture Modelling

On the other hand, the PCE metric corresponding to the cross-correlation values also exposes the unexpected behavior. PCE values too suffer from high FAR's, although the percentage of false alarms is slightly less than ρ. This is expected since PCE is more robust than Normalized Correlation.

The histograms in figure 6.6 reveal the PCE cross-correlation scores for the same Samsung Models. A pink parallel to the y-axis, at a PCE value of 60, indicates the detection threshold. We can easily see that many values surpass the detection threshold leading to false positives. Though in some cases, the tails

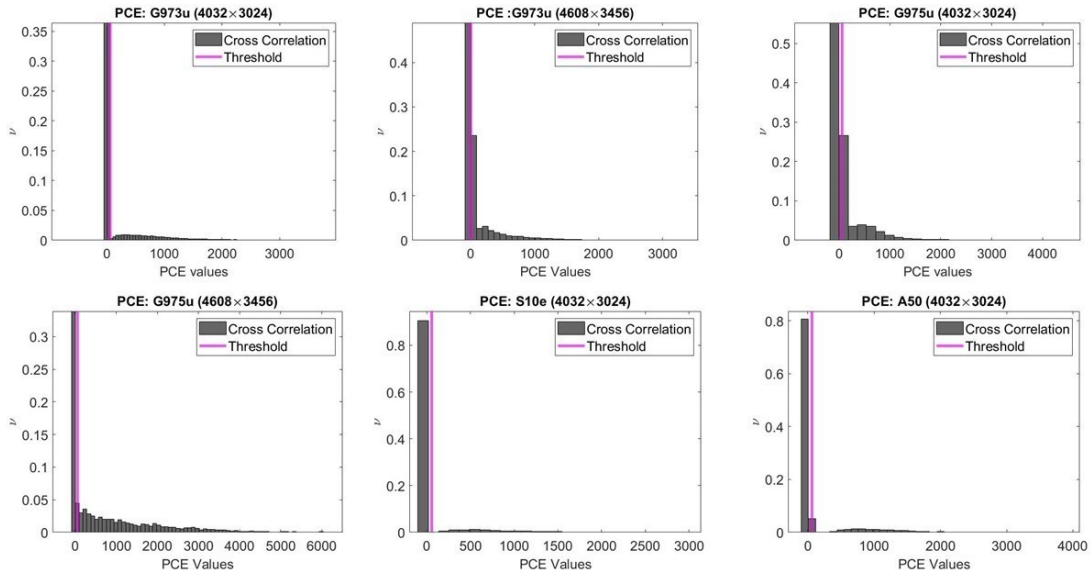


Figure 6.6: Distribution of PCE Values for Samsung Models

of the distribution may not be that evident due to the smaller proportion. To better appreciate the behavior of PCE, we extract some statistics.

Model	μ	Threshold	FAR
SM-G973u (12 MP)	228.5	60	21.5%
SM-G973u (16 MP)	135	60	18.5%
SM-G975u (12 MP)	154	60	21%
SM-G975u (16 MP)	886	60	53.2%
SM-G970u (12 MP)	76	60	9.42%
SM-A505u (12 MP)	220	60	14.08%

Table 6.3: PCE Stats for Samsung Models

Table 6.3 shows the PCE stats of the Samsung Models. Here also, we see a high percentage of FAR's. The results obtained with PCE further cement the problem of non-unique artifacts for these devices. Both PCE and ρ confirm the problem of fingerprint collision for the Samsung Models.
Now, we focus on Huawei models and Redmi note 7 to see the extent of the fingerprint collision.

Huawei Models and Xiaomi Redmi Note 7

Huawei Models, LYA-L29 (Huawei Mate 20 Pro) and VOG-L29 (Huawei P30 Pro) along with Redmi Note 7 also suffer from the unexpected distribution. However,

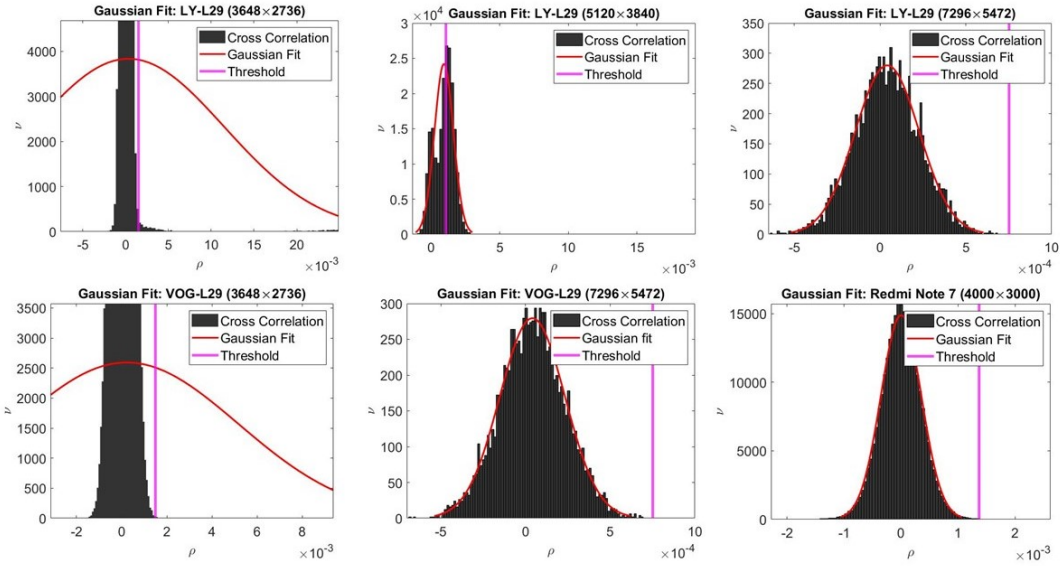


Figure 6.7: Normalized Correlation for Huawei Models and Redmi note 7

the behavior is not as prominent as the Samsung Models, meaning that they record significantly lower false alarms than the Samsung ones. Interesting to note that the highest resolutions of Huawei Models record 0 false positives, unlike the highest resolutions of Samsung Models (S10 and S10+).

Figure 6.7 shows the histograms for Huawei Models and Redmi Note 7. The first row of the figure consists of three resolutions of Huawei Mate 20 Pro starting from lowest to highest (10 MP, 20 MP, and 40 MP). The second row depicts two resolutions of Huawei P30 Pro (10 MP and 40 MP), followed by a single Redmi Note 7 (12 MP) resolution. These histograms reveal the distribution of ρ values. A pink link indicates the sensor-dependent threshold, and the red curve represents the Gaussian Fit to the distribution.

The histograms corresponding to Huawei Mate 20 Pro and P30 Pro at 40MP show the expected behavior, *i.e.,* not a single value of correlation exceeds the threshold. The ρ values corresponding to these resolutions follow the zero-mean Gaussian curve. Therefore, the devices corresponding to these two resolutions do not exhibit any non-unique artifacts.

6.2 Initial Analysis

On the other hand, the histograms corresponding to the lowest resolutions of both Huawei Models and Redmi Note 7 show that fewer correlation values surpass the threshold, unlike the Samsung Models, where a comparatively more extensive number of correlation values exceeded the threshold. However, the histogram corresponding to Huawei Mate 20 Pro at an intermediate resolution of 20 MP suffers from unexpected behavior. A considerable number of correlation values exceed the threshold. Therefore, different Mate 20 Pro devices at this resolution exhibit significant artifacts, leading to higher FAR's. To gain better insights, we extract statistical quantities for these models.

Model	μ	Threshold	σ_{ex}^2	σ_{ac}^2	FAR
LY-L29 (10 MP)	$3.43e-04$	0.0015	$1.0019e-07$	$1.249e-04$	0.85%
LY-L29 (20 MP)	$9.44e-04$	0.0011	$5.086e-08$	$4.744e-07$	48%
LY-L29 (40 MP)	$4.2e-05$	$7.5e-04$	$2.504e-08$	$3.49e-08$	0
VOG-L29 (10 MP)	$2.25e-05$	0.0015	$1.0019e-07$	$2.43e-05$	0.24%
VOG-L29 (40 MP)	$4.03e-05$	$7.5e-04$	$2.504e-08$	$3.72e-08$	0
Redmi Note 7 (12 MP)	$8.44e-07$	0.0014	$8.33e-08$	$1.255e-07$	0.55%

Table 6.4: Normalized Correlation Stats of Huawei Models and Redmi Note 7

Table 6.4 shows the statistics of the Huawei Models and Redmi Note 7. The σ_{ex}^2 and σ_{ac}^2 refer to the expected variance and the one actually found. All the models, except LY-L29 at 20 MP, record FAR's $< 1\%$. Particularly for Huawei's highest resolutions, the expected variance coincides with the actual. The devices corresponding to these resolutions behave ideally. However, a very high FAR of 48% can be seen for LY-L29 at the resolution of 5120×3840 or 40 MP.

We do a similar analysis for these devices as we did for the Samsung ones. We fit a Gaussian Mixture to understand if these devices suffer from bi-modal distribution. We see that only Mate 20 Pro at 20 MP resolution suffers from the bi-modal Gaussian distribution. For all other resolutions, a single-mode Gaussian curve fits the distribution.

Figure 6.8 shows the bi-modal distribution of ρ values for the Mate 20 Pro at 20 MP resolution. The behavior is similar to what we see for Samsung Models. The two peaks at ρ values of $-1.38e-05$ and 0.0012 correspond to two modes of the Gaussian Mixture. The latter mode leads to the problem of high false alarms. To further verify this behavior, we compute the PCE metric for these devices.

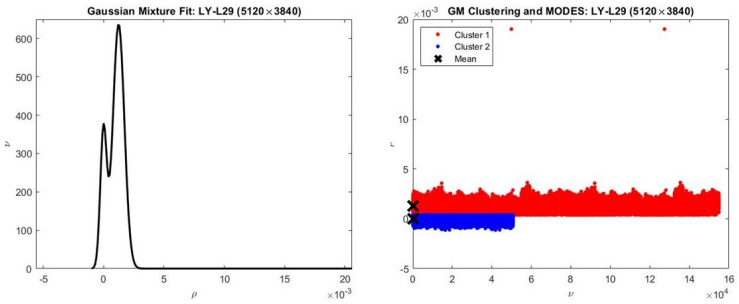


Figure 6.8: GMM for Huawei Mate 20 Pro @20 MP Resolution

Model	μ	Threshold	FAR
LY-L29 (10 MP)	$1.0605e+03$	60	0.69%
LY-L29 (20 MP)	24	60	7.4%
LY-L29 (40 MP)	0.48	60	0
VOG-L29 (10 MP)	145	60	0.23%
VOG-L29 (40 MP)	0.483	60	0
Redmi Note 7 (12 MP)	0.0034	60	$< 0.1\%$

Table 6.5: PCE Stats of the Huawei Models and Redmi Note 7

Table 6.5 shows the PCE stats for Huawei Models and Redmi Note 7. Interestingly, with PCE, we record a significantly lower FAR of 7.4% for LY-L29 at 20 MP against 48% for the Normalized Correlation metric. This shows the

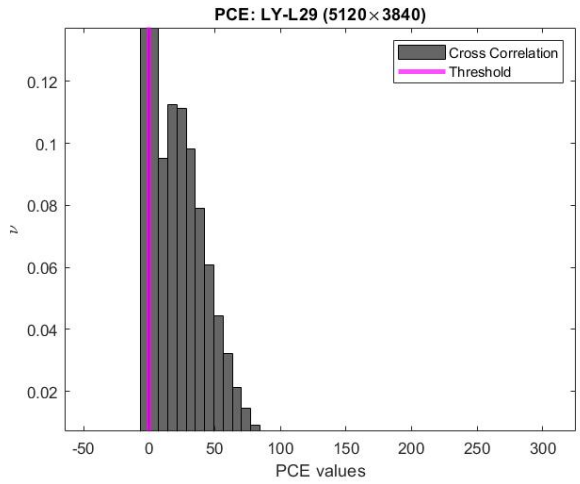


Figure 6.9: PCE metric for Huawei Mate 20 Pro @20 MP Resolution

superiority of PCE over ρ. However, the focus is that 7.4% of the false alarm ratio is still unexpected. Overall, PCE results are consistent with the Normalized Correlation. Lower false alarms are seen for these devices than the Samsung ones.

Since FAR is not significant for other devices, we plot PCE values only for Mate 20 Pro at 20 MP. Figure 6.9 shows the distribution of PCE values for Huawei mate 20 Pro at 20 MP. We can see that even with PCE, many values surpass the threshold value of 60.

6.2.2 Pair-wise Fingerprint Comparison: Strategy 2

Under this strategy, we correlate the PRNU terms for every pair of images. We expect to see slightly different results since the PRNU extraction is more complex than the Noise Residual. We do this analysis for a subset of models, which exhibit higher artifacts under Pair-wise Noise Residual Comparison. They are SM-G973u (Samsung S10) at 12 MP and 16 MP, SM-G975u (Samsung S10+) at 12 MP and 16 MP, and LY-L29 (Huawei Mate 20 Pro) at 20 MP.

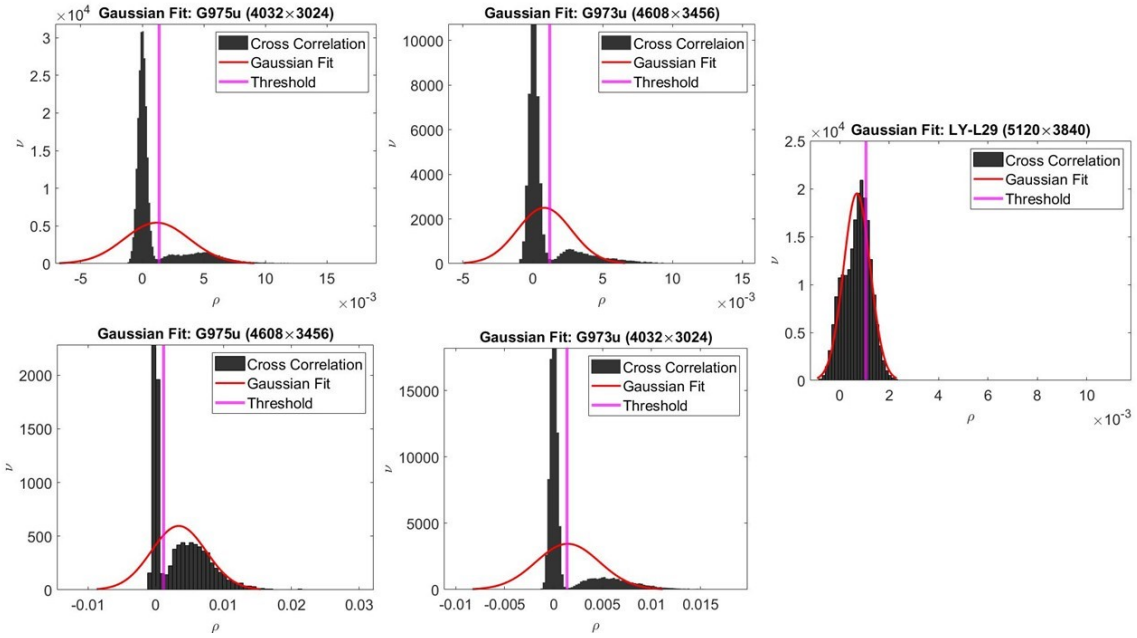


Figure 6.10: Normalized Correlation under Pair-wise Fingerprint Comparison

Figure 6.10 shows the ρ values corresponding to Pair-wise Fingerprint correlation. As with the strategy 1, many correlation values surpass the threshold. We

fit the distribution with Gaussian Mixture, and we find that both Samsung Models suffer from the same bi-modal Gaussian behavior. However, Huawei Mate 20 Pro does not suffer from two-mode distribution under this strategy. We extract some statistical parameters corresponding to these devices to see the differences from Pair-wise Noise Residual comparison.

Model	μ	FAR
SM-G973u (12 MP)	0.0014	21%
SM-G973u (16 MP)	$8.28e-04$	18.4%
SM-G975u (12 MP)	0.0012	22.18%
SM-G975u (16 MP)	0.0034	52%
Huawei LY-L29 (20 MP)	$6.9e-04$	25%

Table 6.6: Normalized Correlation Stats for Pair-wise Fingerprint Comparison

.

Table 6.6 shows stats of correlation values using fingerprint comparison, strategy 2. The problem of high FAR persists, though there is an advantage to Pair-wise Noise Residual comparison. For the Samsung Models, we record a slightly lower percentage of false positives. However, for Huawei LY-L29 at 20 MP, there is a massive difference to strategy 1. With the strategy 2, a dip of 23% in the FAR is seen for this resolution.

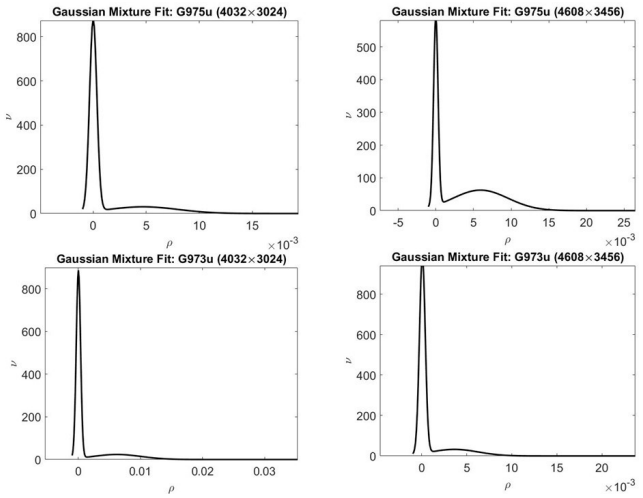


Figure 6.11: GMM for Pair-wise Fingerprint Comparison

We fit the GMM to the ρ values of Samsung Models. Figure 6.11 shows the

Model	μ	FAR
SM-G973u (12 MP)	127	20%
SM-G973u (16 MP)	61.5	16.8%
SM-G975u (12 MP)	86.41	19.67%
SM-G975u (16 MP)	355	51%
Huawei LY-L29 (20 MP)	13.09	0.65%

Table 6.7: PCE Stats for Pair-wise Fingerprint Comparison

bi-modal Gaussian behavior for the Samsung Models S10 and S10+ at both resolutions. These results are consistent with the Pair-wise Noise Residual Comparison. Therefore, with both strategies, the Samsung Models suffer from bi-modal distribution as far as Normalized Correlation is concerned.

PCE metric for Pair-wise Fingerprint comparison presented in table 6.7, further testifies a slight advantage of using strategy 2, in terms of FAR's. We see a slight dip in the percentage of false alarms for all the models. This is expected since fingerprint estimation involves an additional step of demodulating the Noise Residual. This confirms that the contribution of non-unique artifacts (ϵ') to the PRNU terms is less than the contribution (ϵ) to the Noise Residuals. More importantly, PCE significantly reduces FAR for Mate 20 Pro with strategy 2.With strategy 2, the Mate 20 Pro at 20 MP resolution records $< 1\%$ of FAR against 7.4% with the strategy 1. The above analysis shows that strategy 2 outperforms strategy 1. We present a table that highlights the superiority of Fingerprint Comparison over Noise Residual in terms of false alarms.

Model	Strategy 1 : Strategy 2 (ρ)	Strategy 1: Strategy 2 (PCE)
SM-G973u (12 MP)	21.6% : 21%	21.5% : 20%
SM-G973u (16 MP)	19% : 18.4%	18.5% : 16.8%
SM-G975u (12 MP)	22.6% : 22.18%	21% : 19.67%
SM-G975u (16 MP)	54% : 52%	53.2% : 51%
Huawei LY-L29 (20 MP)	48% : 25%	7.4% : 0.65%

Table 6.8: Comparison of two Strategies in terms of FAR's

Table 6.8 shows FAR's corresponding to both Normalized Correlation (ρ) and PCE. We can clearly see that strategy 2 always results in a lower percentage of false alarms for both metrics.

In this section, we exposed the problem of fingerprint collision for recent smartphones. Under both strategies, the unexpected issue of high false alarms is seen. Strategy 2 is seen to outperform strategy 1 in terms of false alarms. It can significantly reduce the number of false alarms (Mate 20 Pro at 20 MP). Also, as expected, PCE outshines the Normalized Correlation metric. So strategy 2 coupled with PCE achieves the best results as far as FAR is concerned. The Huawei Models and Redmi note 7 do not suffer significantly from the problem of fingerprint collision. However, the Samsung Models exhibit significant non-unique artifacts among their devices, which leads to alarmingly high FAR's with both the strategies.

6.3 Manual Screening: Cause of Unexpected Behavior

Model	Number of Good Images	Number of Bad Images
SM-G973u (12 MP)	181	197
SM-G973u (16 MP)	74	205
SM-G975u (12 MP)	217	341
SM-G975u (16 MP)	27	113
SM-A505u (12 MP)	44	165
SM-G970u (12 MP)	224	126
LY-L29 (10 MP)	437	263
LY-L29 (20 MP)	62	428
LY-L29 (40 MP)	140	0
VOG-L29 (10 MP)	598	102
VOG-L29 (40 MP)	139	0
Redmi Note 7 (12 MP)	452	108

Table 6.9: Number of Good and Bad Images Manually Identified

The problem verification tools reveal that some cross-correlation values surpass the threshold. However, other correlation values behave as expected, *i.e.* are well below the threshold. We analyze the ρ and PCE values to understand what triggers the unexpected distribution (correlation values greater than the threshold), in particular for ρ, what gives rise to the second mode of the Gaussian Mixture, as exposed in the section 6.2. Upon thorough analysis of the pair-wise correlation

values, we observe that the correlation values corresponding to some images exceed the threshold, while correlation values corresponding to others do not. The former are called Bad Images, while the latter as Good Images. Therefore, good images do not record any false positives, *i.e.,* none of the pair-wise correlation values (Normalized Correlation or PCE) exceed the threshold. In other words, the correlation between the PRNU patterns or Noise Residuals of two good images and one good and one bad always results in values less than threshold. However, the correlation between patterns or residuals corresponding to two bad images results in values greater than the threshold. Therefore, we conclude that the unexpected behavior or the second mode results from pair-wise correlations between bad images only. It is the subset of bad images, which lead to unexpected behavior as far as Normalized Correlation and PCE metric are concerned.

Based on the discussion above, we identify good and bad images for each model. Table 6.9 shows the number of good and bad images identified for each model and each resolution. A total of 4643 images are analyzed. As discussed in section 6.2, Huawei models at the highest resolution do not record any false positives. Therefore there is no bad image for these models. We record a total of 2595 good and 2048 bad images. These good and bad images serve as ground-truths for validating the performance of the two algorithms.

6.4 Meta-Data And CFA Analysis

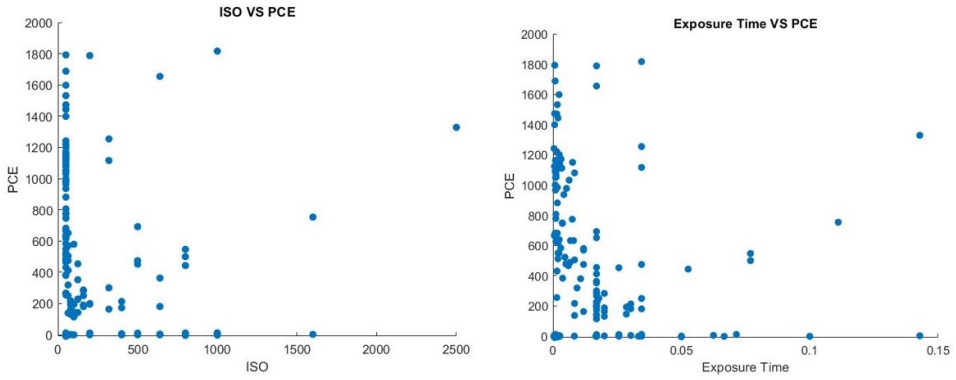


Figure 6.12: Meta-Data Settings VS PCE

After initial analysis, we investigate the meta-data of the images to see if the unexpected behavior can be linked to some specific settings. We examine the

scatter plots between each meta-data setting like Exposure, ISO, Aperture, *etc.,* with the correlation and PCE values to see if there is some pattern in the scatter plot. For the sake of readability, we report only Exposure and ISO; the same results hold for other settings.

Figure 6.12 shows scatter plots of Exposure and ISO against PCE metric for the images producing unexpected distribution for one of the Samsung Models. From the scatter plots, it is clear that there exists no pattern concerning the ISO and the Exposure. We find that there is not a particular setting that triggers the unexpected behavior. So, the same settings can produce PCE values below and above the threshold. This behavior is verified for all the models. Thus the problem of fingerprint collision can happen at any value of Exposure, ISO, Aperture, *etc.* Therefore, we rule out the link between unexpected behavior and the standard meta-data settings.

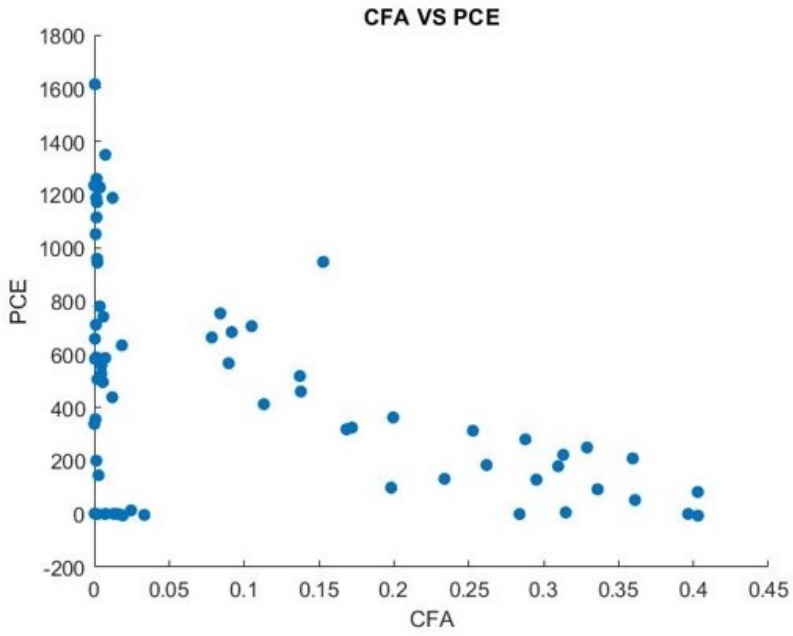


Figure 6.13: CFA Trace VS PCE

We also probe images to see the existence of CFA (Color Filter Array) trace. Since modern cameras employ complex in-camera processing, the trace might get destroyed. Such a trace is detected by exploiting periodicity in the second derivative. The purpose of analyzing this trace is to see if the absence of the CFA trace triggers unexpected behavior. This trace is analyzed at the native resolution of each model.

Figure 6.13 shows the CFA trace for images against PCE values for Samsung S10+ which has 140 pictures.We can see that the trace is present in many images. We record a stronger trace, higher values > 0.2 of CFA for many images. Even those images recording a stronger CFA trace also suffer from the unexpected PCE values. For these images, the PCE values are in the order of $1e2$. The same behavior is seen for all models, *i.e.*, no link between high cross-correlation values and CFA can be established.

This analysis confirms two things: CFA trace is not destroyed despite the complex in-camera processing of modern smartphones; second, the unexpected behavior can not be linked to the presence or absence of the CFA trace; there is no dependence whatsoever.

6.5 Performance of the Algorithms

This section discusses the experiments and the results concerning the algorithms.

6.5.1 SPAM Classifier

We follow the pipeline presented in chapter 5. First, we identify good and bad images for each model. The details of this step are presented in section 6.3. We remind that good images are those which record zero false positives. Then we extract SPAM features for each image. The SPAM function outputs the SPAM features. The output data is of the size 4643×686, where 4643 represents the number of images and 686 represents the number of SPAM dimensions or features for each image. SPAM features are pre-processed using PCA to scale down the number of features. We use a random number of PCA components in the first instance. The data is divided into training and testing data with a standard $70 : 30$ split.

MATLAB APP is used to train a classifier. With MATLAB APP, we use feature engineering to select the optimum number of PCA components regarding accuracy and complexity cost. The APP allows us to train many classifiers; we choose the one with the best accuracy. After multiple hit and trials, we find that 30 and Cosine KNN represents the optimum choice for the number of PCA components and the classifier, respectively. Cosine KNN uses the cosine function, *i.e.*, $cos(\theta)$ as the distance metric to find similarities between the neighbors. The angular difference decides whether to group two points together or not. With a $70 : 30$ split, the training data is of the size 3251×30 and the test data

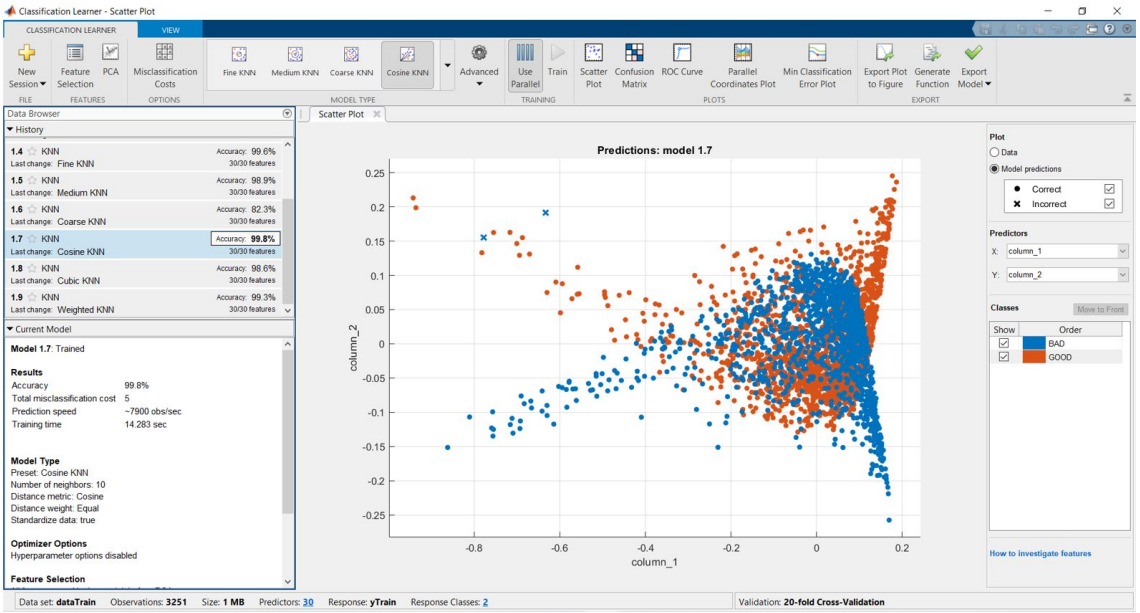


Figure 6.14: Classifier Training on MATLAB APP

1392×30. In order to split the data, we use an in-built function of MATLAB, which randomly splits the data.

The input to the classifier is the training data. A k-folds cross-validation is used to prevent the classifier from over-fitting. The training data is divided randomly into k partitions or folds, and the classifier is trained repeatedly with $k-1$ folds. The remaining 1 fold of the training data is always left out for validation. The left-out folds are used to tune the model. The results are averaged across the folds. 20 folds prove to be the best for our data in terms of classifier's accuracy. The test data is left unseen to the model.

Figure 6.14 shows the training process of the classifier. The X-axis and the Y-axis represent two PCA components since it is impossible to visualize > 3 dimensions. The red and blue dots correspond to the $2D$ PCA components of good and bad images, respectively. We can see the accuracy of different classifiers on the training data in figure 6.14. KNN outperforms decision trees and SVM's. Among KNN, Cosine KNN achieves the best accuracy. The number of neighbors chosen is 10, meaning that 10 nearest neighbors are used to classify a feature point. Moreover, the training is completed in just 14 seconds, thanks to dimensionality reduction. The classifier achieves a very high accuracy of 99.8% on the training data. There are only 5 miss-classifications.

In the subsequent experiments with this algorithm, we try to understand if the SPAM features are resolution-specific or model-specific. The goal is to see if the SPAM Classifier generalizes across different models or even resolutions of the same model. For this, we conduct two experiments. First, we train a classifier on all models, excluding one. For instance, we train on all models and leave S10 (all resolutions) out. The left-out model is used as the testing data. We do a round-robin, every-time we leave a different model out as the testing data. Since there are 7 different models in our dataset, we perform this experiment 7 times; in each, we leave one of the 7 models out. The second experiment involves training on one of the resolutions and leaving other resolutions of the same model out. For example, we train on features of all the resolutions, including S10 (16 MP), and leave S10 (12 MP) out as the test data. This experiment is done on models which feature more than one resolution. These two experiments help us to understand if the SPAM features are model-specific or resolution-specific. The results for the test data are discussed in subsection 6.5.3.

6.5.2 Meta-Data Screening: SceneType Tag Classifier

For Samsung Models, we find the existence of a meta-data tag, *SceneType: A directly photographed Image*. This tag can be used to identify good images directly. Most of the images which record this tag do not suffer from fingerprint collision. On the other hand, the images which do not contain this tag suffer from high false alarms. Both the strategies confirm this behavior. To visualize this behavior, we first identify the images with this tag for each Samsung model.

Model	Images With SceneType	Images Without SceneType
SM-G973u (12 MP)	167	211
SM-G973u (16 MP)	73	206
SM-G975u (12 MP)	252	306
SM-G975u (16 MP)	37	103
SM-G970u (12 MP)	203	147
SM-A505u (12 MP)	41	168

Table 6.10: Number of Images with and without SceneType Tag

Table 6.10 shows the number of images with and without the SceneType tag for each resolution. We plot the normalized correlation and PCE values corresponding to these images. For the sake of readability, we only show results from

Pair-wise Noise Residual comparison, the same holds for Pair-wise Fingerprint comparison.

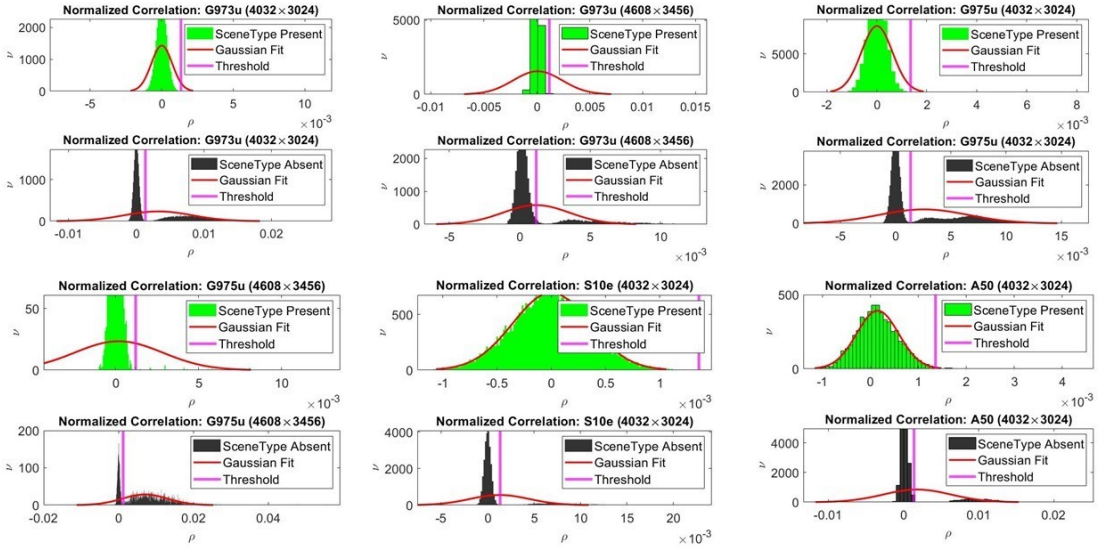


Figure 6.15: Dependence of Normalized Correlation on SceneType Tag

Figure 6.15 shows the distribution of ρ values in presence and absence of the SceneType tag for all the Samsung Models and all resolutions. The subplots correspond to S10 (12 MP), S10 (16 MP), S10+ (12 MP) in the first row and S10+ (16 MP), S10e (12 MP), and A50 (12 MP) in the second row. The green histograms show the ρ values corresponding to the images which contain SceneType tag. The pink line represents the threshold, and the red curve is a Gaussian Fit to the distribution. The black histograms depict the correlation scores of the images which don't contain the tag.

From the green histograms, it is clear that almost all of the correlation scores are below the pink line. The problem of high false alarms or bi-modal distribution is not seen for the images containing this tag. The correlation values follow the expected zero-mean Gaussian distribution. These images mostly record 0 false positives or very few in some cases.

On the other hand, those images which do not contain this tag suffer from the bi-modal distribution. The black histograms reveal that many correlation values surpass the threshold, leading to high false alarms. Due to the smaller proportion, the tails may not be that evident in some histograms. We extract

Model	SceneType Present (μ)	SceneType Absent (μ)
SM-G973u (12 MP)	$2.605e-06$	0.0032
SM-G973u (16 MP)	$8.37e-05$	0.0011
SM-G975u (12 MP)	$6.27e-06$	0.0025
SM-G975u (16 MP)	$7.43e-05$	0.0068
SM-G970u (12 MP)	$-3.16e-07$	0.0012
SM-A505u (12 MP)	$1.54e-04$	0.0018

Table 6.11: Mean value of Gaussian Fit to Green and Black Histograms

the mean value of the Gaussian curve that fits green and black histograms to highlight the difference between the two distributions.

Table 6.11 shows the mean value of the Gaussian curve that fits the green and black distribution. The Gaussian curve that matches the ρ values corresponding to the images with the SceneType tag for each resolution is nearly a zero-mean. On the other hand, a relatively higher mean Gaussian curve fits the ρ values of images without this tag.

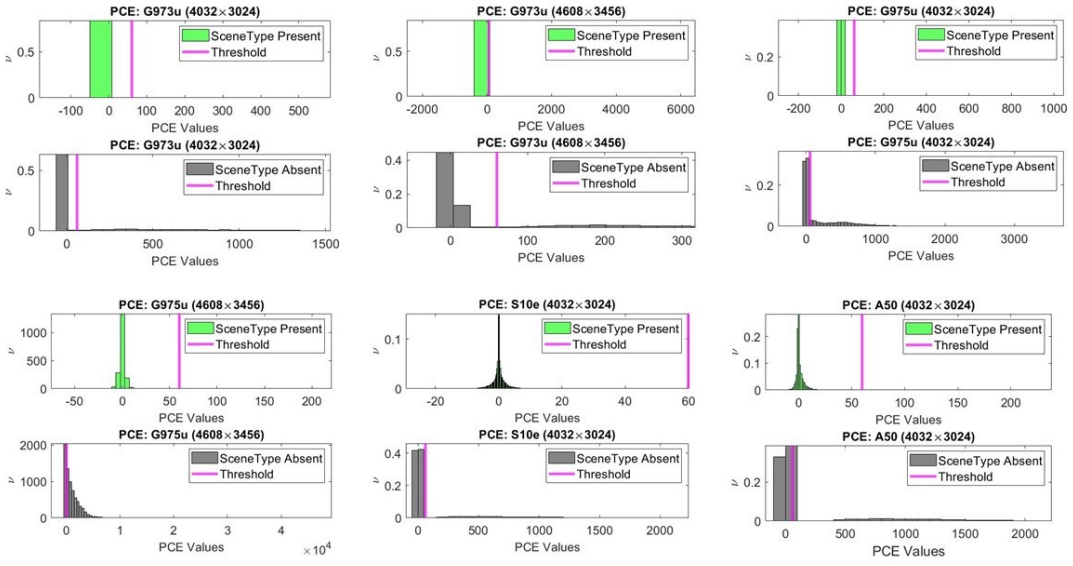


Figure 6.16: Dependence of PCE Values on SceneType Tag

The PCE metric also validates these results. The distribution of the PCE values shown in figure 6.16 corresponds to the images with and without the SceneType tag. The subplots reveal the PCE values corresponding to the Samsung

Models. The PCE values of images with the SceneType tag are well below the threshold. On the other hand, PCE values surpass the threshold for the images which do not contain this tag.

Therefore, both the metrics confirm the association of the SceneType tag to the expected distribution. To estimate the accuracy of this algorithm, we compare its performance with the ground-truths from the manual screening described in section 6.3.

6.5.3 Validation

In this subsection, we present the evaluation metric concerning the two algorithms. The evaluation metric helps us to understand the accuracies of each algorithm.

SPAM Classifier

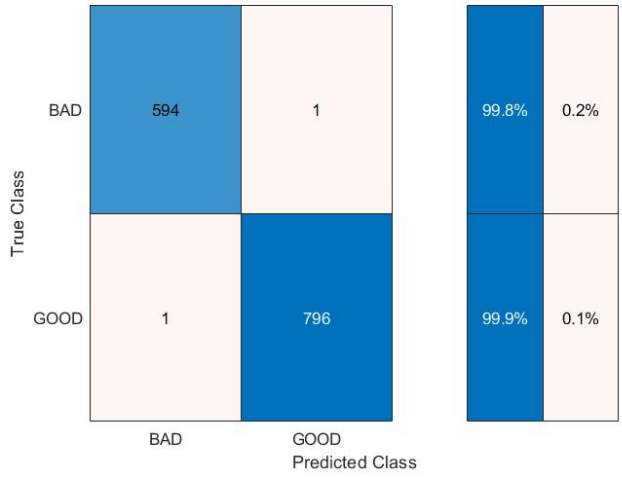


Figure 6.17: Confusion Matrix based on Test Data of SPAM Classifier

Figure 6.17 represents the confusion matrix concerning the test data of the SPAM Classifier. Out of 1392 test examples, the classifier incorrectly classifies only 2 of them. 99.8% accuracy is associated with the bad class and 99.9% with the good class. Therefore, our model can achieve very high accuracy in correctly identifying good and bad images corresponding to all the models and resolutions. Even if we use 50% of data for training the classifier, the model still performs exceptionally well with $> 99\%$ accuracy for each class. Therefore, the SPAM Classifier

can be a valuable tool in identifying good and bad images with relatively small data.

Generalization of SPAM Classifier

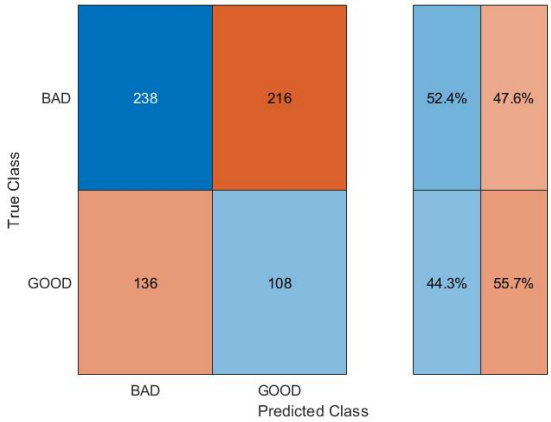


Figure 6.18: SPAM Classifier across different Models

Figure 6.18 shows the confusion matrix where the training data consists of features of all models except S10+. The features corresponding to S10+ (all resolutions *i.e.* 12 MP and 16 MP) are left-out to validate the performance of the classifier. They therefore represent the test data. We leave one of the models out to see if the SPAM Classifier is to able to generalize across different models.

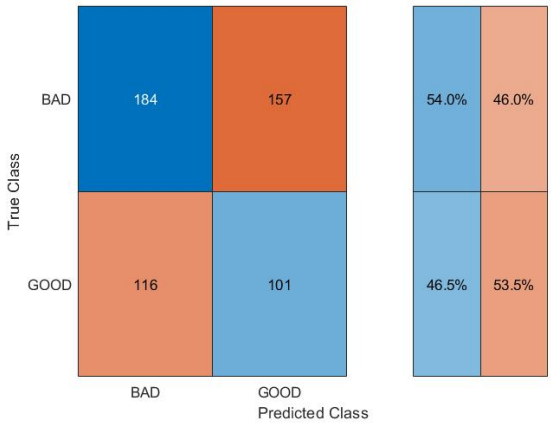


Figure 6.19: SPAM Classifier across Resolutions of the same Model

Figure 6.19 shows the confusion matrix where test data only consists of features corresponding to 12 MP resolution of S10+. This time the training data includes features of S10+ (16 MP). So, we train with one of the resolutions and leave the other resolutions of the same model out for testing. The goal is to see if the SPAM features generalize across the resolutions of the same model.

The confusion matrices depicted in figures 6.18 and 6.19 reveals poor accuracies and high miss-classification costs. The performance of the classifier is like a random guess. Therefore, the SPAM Classifier does not generalize across different models and resolutions of the same model. Within a model, SPAM features tend to be resolution-specific. The same behavior is seen for other models.

Meta-Data Screening: SceneType Tag Classifier

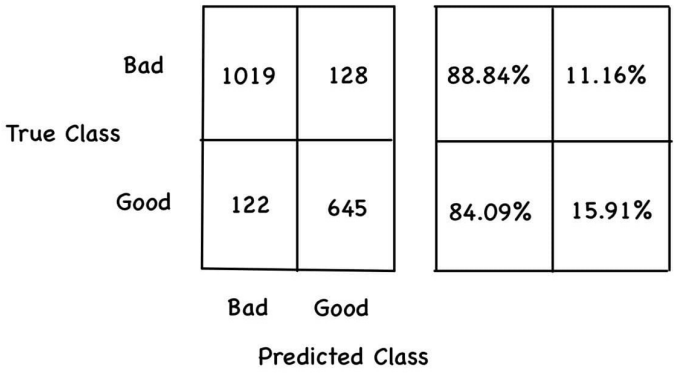


Figure 6.20: Confusion Matrix based on Meta-Data SceneType Tag Classifier

Figure 6.20 shows the confusion matrix for the SceneType Tag-based Classifier. A total of 1914 images corresponding to all Samsung models are analyzed. The algorithm achieves a decent accuracy of 84% for the good class and $\simeq$ 89% for the bad one. Compared to the SPAM Classifier, the accuracies are relatively low, and the miss-classification percentage is relatively high. 11.16% and 15.91% are the inaccuracies for each class. Nevertheless, SceneType-based Classifier can prove a handy tool in real-time identification of good and bad images.

Chapter 7

Conclusion: Limitations and Future Work

In this work, we verified the problem of fingerprint collision for recent smartphones. The results obtained are consistent with Iuliani *et al.* (2021). Since the number of images per device and number of users differs, the results vary slightly. However, the problem of fingerprint collision is verified in general. We saw that Huawei Models did not suffer from false alarms at the highest resolution. In contrast, the rest of Huawei's resolutions did not seem to cause a problem when strategy 2 was considered. We can therefore conclude that Huawei Models and Redmi Note 7 do not suffer significantly from the problem of fingerprint collision. On the other hand, Samsung Models revealed a high percentage of false alarms under both strategies. Notably, the highest resolutions of Samsung Models also recorded many false positives, unlike Huawei's highest resolutions. Thus, the corresponding Samsung Devices exhibit significant non-unique artifacts. Given the use of PRNU by various law enforcement agencies, this problem of fingerprint collision represents a threat and can lead to severe consequences.

We did not find any link between the settings and the unexpected behavior with the meta-data analysis, except for the Samsung Models. Also, some manufacturers not choosing to embed the tags related to computational photography adds to our woes.

We, for the first time, propose two algorithms to counter this problem. Our algorithms aim at identifying the images suffering from non-unique artifacts. One of the algorithms is based on SPAM features, and the other is based on the SceneType tag of meta-data. The algorithm based on SPAM features achieves a very high accuracy of 99.8% in identifying good and bad images. On the other

hand, the SceneType-based Classifier is relatively quick and achieves a decent accuracy ($\simeq$ 86.5%.). Once good and bad images are identified, the bad images are disregarded as far as PRNU extraction is concerned. Instead, good images can be used to extract a reliable fingerprint.

Limitations

In this section, we discuss the following limitations of our methods:

Algorithms

The SPAM Classifier-based algorithm does not generalize across different models and resolutions of the same model. The classifier needs to see the training examples corresponding to each model and all resolutions. In other words, SPAM features are resolution-specific and therefore do not generalize. This is because of the pixel-binning and some other processing that is resolution-specific.

On the other hand, Meta-Data-based Classifier is only valid for Samsung Models; it does not generalize for Huawei and Xiaomi Models. Though relatively quick, but with a lower accuracy than the SPAM Classifier.

Dataset

The dataset was collected from the Flickr database. Even though the meta-data settings ensure the authenticity and reliability of the images, we believe that having access to smartphones could have allowed us to perform additional experiments. Some manufacturers are not embedding the tags associated with the computational photography in the images. We could have manually kept track of the images shot under Portrait, Night mode, AI effect, *etc.* This also represents an area of future work.

Future Work

We discuss some of the possible areas for future research:

Denoising Filter

We believe that the denoising filter cannot suppress the non-unique artifacts (NUA's) in the images. The Noise Resdiuals and the PRNU terms contain traces of NUA's. Therefore, more powerful denoising filters are required. It is interesting to see the results with deep learning-based methods, which employ a pre-trained network to compute the Noise Residuals.

Large Dataset

We have tested our results on a small dataset. We could incorporate additional smartphones and other professional cameras to test the efficiency of the SPAM-based algorithm.

Deep Learning

Also, it is interesting to replace the PRNU pipeline entirely with the deep learning one for modern smartphones. Instead of extracting features from images, we could directly feed images or patches to the deep learning network.

References

Chen, M., Fridrich, J., Goljan, M. & Lukás, J. (2008). Determining image origin and integrity using sensor noise. *IEEE Transactions on information forensics and security*, **3**, 74–90. 6

Costa, H.B., Zampolo, R.F., Carmo, D.M., Castro, A.R. & Santos, E.P. (2012). On the practical aspects of applying the prnu approach to device identification tasks. In *International Conference on Multimedia Forensics, Surveillance and Security*. 9

de Roos, L. & Geradts, Z. (2021). Factors that influence prnu-based camera-identification via videos. *Journal of Imaging*, **7**, 8. 21

Fernández-Menduiña, S. & Pérez-González, F. (2021). On the information leakage quantification of camera fingerprint estimates. *EURASIP Journal on Information Security*, **2021**, 1–13. 21

Fridrich, J., Lukas, J. & Goljan, M. (2006). Digital camera identification from sensor noise. *IEEE Transactions on Information Security and Forensics*, **1**, 205–214. 5, 9

Iuliani, M., Fontani, M. & Piva, A. (2021). A leak in prnu based source identification—questioning fingerprint uniqueness. *IEEE Access*, **9**, 52455–52463. 2, 18, 24, 61

Joshi, S., Korus, P., Khanna, N. & Memon, N. (2020). Empirical evaluation of prnu fingerprint variation for mismatched imaging pipelines. In *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, 1–6, IEEE. 19

Liu, Y., Zou, Z., Yang, Y., Law, N.F.B. & Bharath, A.A. (2021). Efficient source camera identification with diversity-enhanced patch selection and deep residual prediction. *Sensors*, **21**, 4701. 20

REFERENCES

Pevny, T., Bas, P. & Fridrich, J. (2010). Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on information Forensics and Security*, **5**, 215–224. 12