Social Media Investigations Using Shared Photos

Flavio Bertini  
flavio.bertini2@unibo.it

Rajesh Sharma  
rajesh.sharma@unibo.it

Andrea Ianni  
andrea.ianni@studio.unibo.it

Danilo Montesi  
danilo.montesi@unibo.it

Department of Computer Science and Engineering  
Mura Anteo Zamboni, 7  
University of Bologna, Italy

ABSTRACT

In recent years, social networks (SNs) have revolutionized many aspects of society. The wide variety of available platforms meets almost every kind of users’ needs: socialization, professional connections and image sharing, to name a few. Smartphones have brought vital change in user’s behaviour towards sharing of multimedia content on online SNs. One of the noticeable behaviour is taking pictures using smartphone’s camera and sharing them with contacts through online social platforms. On the flip side, SNs have contributed to the growth of the cyber crime and one of the main aspect of social networks security is to identify fake profiles. In this paper, we present a method to verify and resolve user profiles in SNs using the images posted on them, assuming taken through user’s smartphone. Creating a sort of fingerprint of the user profile, the method is capable enough to resolve profiles in spite of the fact that the images get downgraded during the uploading/downloading process. Also, the method can compare different images belonging to different SNs without using the original images. To evaluate our method, we use three online SNs and five different smartphones. The results using real dataset, show profile verification can provide results of 96.48% and resolution of 99.49% on an average values, which shows the effectiveness of our approach.

KEYWORDS

Social Network Security, Profile Verification, Profile Resolution, User Profile Fingerprint, Pattern Noise.

1 INTRODUCTION

In the last decade, various types of social platforms, have been introduced on web. These various networks cater the specific needs of the users. For example, LinkedIn for professional networks, Facebook for more social interactions, Instagram for photo sharing, WhatsApp for instant messaging on mobiles to name a few. Smartphones have played an important role in users’ behaviour with respect to posting and sharing of multimedia content on social platforms [18].

A recent article has revealed that smartphones are the usual gateways for social media, especially for teenagers [22]. If SNs provide opportunities to socialize and share interests (for example, across political, economic and geographic borders), it is also true that they offer a medium for fraudulent activities.

This paper, address the issue of fake profiles problem which is an important problem in social networks security. Thus, our solution can also be useful to detect and prevent cyber crime on SN. In particular, we provide solution for user profile verification and resolution. By user profile verification we mean correctly verifying the person behind the user profile. In other words, the problem tries to verify if the profile indeed belongs to the person who is claiming to be in the profile. User profiles resolution addresses the task to verify if two different user profiles (for example, nicknames) belong to the same user. This can be performed within a same network (intralayer) or across different social networks (interlayer). Consider
a multilayer network consisting of Facebook, Google+ and WhatsApp (Figure 1). The intralayer profile resolution targets to match user profiles on the same layer as a user can create two profiles and out of two, one could be a fake profile. For example, in Figure 1 (Facebook layer, dotted line), one could be interested in determining if the left side girl (brown hair) is same as the right side girl (blonde hair). The interlayer profile resolution aims at matching user profiles on different layers. For example, in Figure 1 (Facebook and Google+ layers, dash-dot-dash line), one could be interested in understanding if the boy in Facebook layer is same as the boy in Google+ layer.

Figure 1. User profiles resolution problem: interlayer (dotted line) and intralayer (dash-dot-dash line).

The problems of user profile verification and resolution are important for two reasons. Firstly, they correspond to one of many kinds of missing data problem [23], [24]. Secondly, and most importantly are related to a significant increase in the number of fake profiles. For example, in a recent annual report [10], Facebook estimated that 8.35% of its total active profiles can be categorized as fake profiles. The company classifies such fake profiles into three groups:

- 75 million (6.1%) of accounts that users maintains in addition to their principal account;
- 18 million (1.45%) of business, organization or pet accounts;
- 10 million (0.8%) of undesirable profile accounts that have been set up to annoy other Facebook users.

Thus, the problems of user profile verification and resolution are more meaningful for resolving different profiles in digital forensic and criminal investigations.

1.1 Background

Most of the multimedia content posted on social networks is done through smartphones [18]. This intuitively suggests researchers to fingerprint smartphones by exploiting the sensor’s imperfections to identify users (in other words smartphones). The intuition behind using smartphone’s fingerprint for performing digital forensic activities is driven by two reasons. Firstly, due to hard bound phone contracts, smartphones are considered more personal than other digital devices for example, laptops or desktops. Secondly, and more importantly, it has been shown that the sensors (for example, GPS, microphone-speaker, camera etc.) in smartphones can be used for creating a fingerprint [3], [8] and [9].

We propose a solution for detecting fake user profiles by creating unique fingerprint of a smartphone, assuming that the pictures being captured and published on SNs are taken by the smartphone itself. The approach is partially inspired by a social trend of image sharing, which is one of the main activities on social network platforms [21]. In practice, we use defects in the images to create a unique fingerprint of the smartphones. We then use this fingerprint to match different user profiles. Several methods have been proposed to detect defects due to the manufacturing process of the camera [26] and it is possible to distinguish between defects introduced by the hardware components of the device (steps a, b and c in Figure 2) to those introduced by the software components (steps d and e in Figure 2).

Our proposed method is based on hardware imperfections as hardware based techniques
provide better results [16]. Thus, we propose a method based on hardware imperfection to extract the noise component (characteristic noise) systematically present in each image and to create the unique fingerprint of the smartphone. Moreover, we are able to perform this kind of analysis on the original as well downloaded images from social networks (see Figure 2), without conducting any hacking activity on the smartphone’s hardware.

![Figure 2. Consumer level digital cameras pipeline. Step (a): the lens collect the light from the scene; step (b): several hardware filters improve the acquisition; step (c): the sensor converts the light into electrical signal; step (d): the processing produces a visually pleasing image; step (e): the digital image is stored.](image)

Our aim is to identify the source and not the model being used to take a certain picture and thus our problem, source camera identification, should not be confused with the more general digital camera model identification. The idea behind finding the source is that the unique identification of the camera basically maps to the smartphones. Thus, fingerprint of a smartphone transitivity maps to the users, who are registered owner of that device. As a result, if two profiles’ images map to the same smartphone camera, with high probability we can assume that two profiles belong to the same user and thus, helping in user verification and resolution.

### 1.2 Problem Statement

We explore the possibility of uniquely matching two different user profiles (on SNs) based on the pictures being posted on them, assuming taken by the same smartphone’s camera. A smartphone can be uniquely identified by exploiting the characteristic noise present in the images taken by the smartphone. Let $\mathcal{UP} = \{U_1, U_2, \ldots, U_n\}$ defines the set of user profiles across social platforms. Let $\mathcal{I}_{U_i} = \{I_{i1}, I_{i2}, \ldots, I_{im}\}$ represents the set of images posted by the user $U_i \in \mathcal{UP}$ on his profile. We try to verify if any two specific user profiles $U_x, U_y \in \mathcal{UP}$ actually belongs to the same user. To resolve this, our approach uses the set of images belonging to the user profiles under investigation. That is $I_{U_x}$ and $I_{U_y}$.

Let $f$ represents the function that extracts the characteristic noise from set of images $I_{ij} \in U_i$. To create a characteristic noise for a particular user profile $U_x$, the function $f$ averages the characteristic noise of all the images belonging to a user profile.

$$\frac{1}{m} \sum_{j=1}^{m} f(I_{ij}) = CN(I_{U_i}) \equiv CN(U_i) \quad (1)$$

We then use the outcome of $f$ that can be considered the characteristic noise of the user profile $CN(U_i)$ for deciding if two user profiles belong to the same user. We define the function $g$, which takes as input $CN(U_x)$ and $CN(U_y)$ to decide if $U_x$ and $U_y$ belong to the same user.

$$g(CN(U_x), CN(U_y)) = \begin{cases} 
1 & \text{if } U_x \text{ and } U_y \text{ belong to the same user.} \\
0 & \text{otherwise} 
\end{cases} \quad (2)$$
1.3 Contribution

We propose a method to exploit the smartphone camera’s fingerprint to solve the fake profile problem. Our method, extracts fingerprint from images based on the characteristic noise present in them. The approach also overcomes the compression techniques used by various SNs.

In our analysis we have compared the original and unprocessed pictures taken by smartphones with the pictures processed by SNs. The method is capable enough to resolve images from different SNs. Also, the method does not require original images for verification and resolve as in some cases due to reasons like privacy original images might not be available.

The rest of the paper is organized as follows. In Section 2, we review literature work related to smartphone fingerprinting techniques and forensic investigations on SNs. In Section 3, we describe our methodology. The results of our evaluation along with the analysis of our results is discussed in Section 4. We conclude the article with future directions.

2 RELATED WORKS

In this section, we describe various literatures from three different domains, at the intersection of which our work lies. Firstly, in Section 2.1, we explain techniques to identify fingerprints of smartphone devices using various built in sensors. Then, in Section 2.2, we discuss various source camera identification approaches proposed in literature. In the end, in Section 2.3, we present methods to extract useful information from user profiles on SNs for investigation activities.

2.1 Fingerprinting the Smartphones

Various techniques for creating smartphones fingerprints have been proposed recently. These methods are based on built in sensors present in smartphones. For example, a technique by using microphones and speakers present in smartphones to create unique fingerprint is described in [8]. To track users through mobile phones, using smartphone accelerometers a method is proposed in [9]. A hybrid approach is proposed in [3], where authors exploit both i) speakerphone-microphone and ii) accelerometer for creating a uniquely fingerprint for the smartphones. We exploit this phenomena to verify users profile and resolve multiple profiles in SNs, which is the main focus of our study. The smartphone camera could be considered a built in sensor and in the next section, we present works related to source camera identification. This is important as we are using images captured by smartphones’ camera for our investigations.

2.2 Fingerprinting Source Camera Identification

Researchers have also proposed various techniques for source camera identification based on hardware and software defects, which gets introduced during the manufacturing process. A technique based on chromatic aberration is presented in [25] while in [19] a method to recognize a counterfeit image using the filters is presented. Mainly it is the Colour Filter Array (CFA) which, combined with a monochrome sensor, allows to capture multiple colors with a grid of pixels. However, methods based on the footprints left by the sensor are more suitable for the source camera identification problem. In [16], authors showed that Photo-Response Non-Uniformity (PRNU) is a unique feature of sensor which is capable enough to distinguish two cameras. However, this method only works with unscaled photos. This limitation was resolved by proposing a method which can work with images of different sizes by researchers in [11]. In another work in [5], the authors demonstrate that the robustness of the method based on PRNU can overcome the changes made to the images by the various SNs that commonly causes a loss of effectiveness. Compared to all these approaches of source camera identification, we are using the source camera of the smartphones using the PRNU-based method with the aim of resolving user profiles on SNs.
2.3 User Profiles Investigation on SNs

SNs often include strict regulations in their services: “You will not provide any false personal information on Facebook” or “You will not create more than one personal account” (from “Statement of Rights and Responsibilities” of Facebook Legal Terms). However, the large amount of data uploaded on SNs often does not follow these laws strictly [28]. In [1], authors analysed the user activities using the logs found in the smartphone devices which can be used to match user profiles. However, unlike Blackberry, not all the devices retain this information. In comparison, our method is device independent. An interesting solution has been proposed in [13] by exploiting the multimedia contents on SNs. The authors combine user ID and their tags to identify users across the social tagging system.

In [27] and [15] authors analysed users’ information from different SNs to identify and match two different user profiles. However, the method fails in the presence of false information which usually happens in case of fake profiles. Works like [20] and [2] address the matching of user profiles by providing a weight-based framework, while in [14] the authors combine content and network attributes with profile attributes to improve a traditional identity search algorithm. A machine learning techniques is proposed in [17] for solving the matching user profiles problem across multiple SNs. They use 27 features grouped into three types: name based features, general user info based features and SN topological based features. In [6] a weighted ontology-based user profile resolution technique is proposed for profile matching. Generally, these techniques assume that the two user profiles under investigation intentionally share attributes and features. In comparison to these methods, our approach exploits smartphone’s camera to verify and match user profiles, in order to find out fake profiles.

3 METHODOLOGY

In this section, we firstly present a brief background in image processing. This is important as it will help in understanding the reasons behind the selection of a particular method to extract the noise. Next, we describe the procedure followed to extract the characteristic noise from an image and the process to verify the image source. Finally, we present the evaluation results of our approach obtained using various SNs and smartphones.

Images captured by cameras\(^1\) has two components, namely signal and noise component. Technically signal represents the light which hits the camera sensor. In simple words, signal represents image being captured. However, there is always a portion of noise present in the images, which is unavoidable.

The noise component consists of two components. One of the component is called shot noise (or photonic noise). This component is introduced due to external factors such as temperature, brightness or humidity. Pattern noise, which is the second component, can be considered a system and regular one. The word systematic signify that it is present in each image and regular means, in all the images taken from the same source, it is present almost at the same location.

3.1 Pattern Noise Extraction

In this section, we define the function \( f \) useful to extract the smartphone camera’s fingerprint from a set of images. We exploit a PRNU-based method [16] to extract the pattern noise, systematically introduced by the sensor (step \( c \) in Figure 2), as it is more suitable to solve the problem of source camera verification. The pattern noise can be considered as average of large number of images. Denoising algorithms [4] can be employed to clean the representative component. By this way only the noise component is left in the image. Let \( I \) represents the original image, \( N \) the residual noise and \( d \) the denoising function, then formally \( N \) can be represented as:

\[
N = d(I)
\]  

\(^1\)This also apply to the images being taken by smartphones.
The fingerprint $FP_i$ of the camera $i$ (that is the pattern noise) can be approximated as the average residual noise of $n$ images of the camera $i$ [16]:

$$FP_i = \frac{1}{n} \sum_{j=1}^{n} N_j$$

The choice of the denoising algorithm can affect the pattern noise computation due to the presence of high-frequency details which generally belong to the signal component of the image. By increasing the number of samples, the errors can be reduced. However, difficulty in acquiring new samples makes the task impossible in several cases. We use the Block Matching 3D ($d \equiv d_{BM3D}$) denoising algorithm [7], which is sufficiently proficient in distinguishing between high frequencies of noise from the high frequencies of the images [12].

### 3.2 Source Verification

Let $S_i$ identifies the whole set of images taken by the source $i$, then:

- $P_i \subset S_i$ represents the subset of images used to generate $FP_i$ according to (4). $P_i$ represents the training set of our method;
- $C_i \subset S_i$ identifies the subset of images to be classified. $C_i$ represents the test set of the camera $i$;
- $P_i \cup C_i = S_i$;
- $P_i \cap C_i = \emptyset$
- $V = \bigcup C_i$ represents the whole test set of $m$ images to be correlated with each $FP_i$.

The goal is to determine for each image $I \in V$ if it belongs to the source $i$. In practice, we are defining the function $g$ according to (2) that allows to decide if two user profiles $U_x$ and $U_y$ belong to the same user exploiting theirs images. In the first step, according to (3), we extract the residual noise $N_k$ from each image $I_k \in V$ of the unknown source. Then we apply the standard normalized correlation $corr(N_k, FP_i)$ between $N_k$ and $FP_i$ as done in [16]:

$$corr(N_k, FP_i) = \frac{(N_k - \overline{N_k})(FP_i - \overline{FP_i})}{\|N_k - \overline{N_k}\|\|FP_i - \overline{FP_i}\|}$$

This similarity score is computed between each residual noise $N_k$ of the images $I_k \in V$, that belong to the unknown source, and $FP_i$ of each source $i$. Let $\mu_i$ be the mean correlation value for the source $i$, and $\sigma_i$ its standard deviation:

$$\mu_i = \frac{1}{m} \sum_{k=1}^{m} corr(N_k, FP_i)$$

$$\sigma_i = \sqrt{\frac{1}{m} \sum_{k=1}^{m} (corr(N_k, FP_i) - \mu_i)^2}$$

![Figure 3. Two examples of the correlation values distribution. In (a) the threshold value is computed for the iPhone 4S (1), while in (b) for the iPhone 5 (3) (see Section 4).](image)
Then in the second step we can define a threshold $T_i$ for each source $i$ as:

$$T_i = \mu_i + \sigma_i$$  (8)

All the images in $\mathcal{V}$ with a correlation value that exceed the threshold $T_i$ can be identified as taken from the camera $i$. In other words, this is the function $g$ that determines if the owner of the set $\mathcal{P}_i$ is the same user having the set of images identified in $\mathcal{V}$. Also, each time a new $\mathcal{P}_i$ is defined, a new threshold value must be computed. We define the threshold $T_i$ as the mean correlation value plus a standard deviation thanks to the distribution of the correlation values, as shown in Figure 3.

### 3.3 Experimental Setting

We performed our evaluations on five smartphones from two different brands. Table 1 summarizes characteristics of these devices. We collected 200 high-resolution photographs for each of these phones, taken under different conditions. As proposed in [5], we resize images to normalize the dimensions and, as suggested in [16], the crop is used to select a 512x512 central window small enough to cut out any lens aberrations.

<table>
<thead>
<tr>
<th>ID</th>
<th>Brand</th>
<th>Model</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Apple</td>
<td>iPhone 4s</td>
<td>3264x2448</td>
</tr>
<tr>
<td>2</td>
<td>Apple</td>
<td>iPhone 4s</td>
<td>3264x2448</td>
</tr>
<tr>
<td>3</td>
<td>Apple</td>
<td>iPhone 5</td>
<td>3264x2448</td>
</tr>
<tr>
<td>4</td>
<td>Samsung</td>
<td>Galaxy S4</td>
<td>4128x3096</td>
</tr>
<tr>
<td>5</td>
<td>Samsung</td>
<td>Galaxy S4</td>
<td>4128x3096</td>
</tr>
</tbody>
</table>

For our analysis, we select Facebook, Google+ and WhatsApp SNs. Each of these SNs has different compression characteristics of the image, summarized in Table 2. In comparison to Facebook and Google+, WhatsApp is an application specifically initially conceived for mobile users only.

The validation phase can be categorized in the following three classes:

**Test 1: Original-By-Original.** Before applying the method to downloaded images, we need to verify that it works on original ones. Thus, the goal of this test is to validate the proposed method by applying it to the original images taken by the smartphones. In practice, we try to verify the source camera of a set of taken images, which have not been uploaded on any SNs.

The test is carried out systematically varying the cardinality of each $\mathcal{P}_i$, in order to establish the minimum number of images useful to define $\mathcal{FP}_i$. The threshold $T_i$ is computed each time using the $\mathcal{FP}_i$ extracted from $\mathcal{P}_i$, and is used to classify the images in $\mathcal{V}$.

**Test 2: Social-By-Social.** In this test, the goal is to check the robustness of our approach for images whose resolution gets affected due to uploading and downloading process on the SNs. This is important to verify smartphones which has been used for capturing and uploading the images. From an investigative and forensic point of view, this test allows the mapping between a device and the images published on a SN. In general, it’s possible to match a user profile to a physical device and transitively to the user who is registered owner of the smartphone.

The classification procedure is the same of the previous test, however in this case, both image sets $\mathcal{P}_i$ and $\mathcal{V}$ are previously uploaded and downloaded on every SNs chosen, as shown in Figure 4.

**Test 3: Cross-Social.** The aim of this test is to demonstrate that using our approach it is possible to verify two different user profiles, from two or same SN, using the images posted on them (Figure 5). Also, this test can be used to verify the source of images posted on a SN, di-
Figure 4. Social-by-social test. Steps (a) and (b): the $5 P_i$ subsets, related to the $i=5$ sources, have been uploaded and downloaded on each SNs (Facebook, Google+, and WhatsApp). Step (c): the set $V$ has been composed using the sets $C_i$ of images downloaded from the user profiles on SNs. Step (d): the 5 different fingerprints have been extracted using the downloaded images $P_i$, to pair each source to the right user profile.

In the former case the significant outcome is the ability to match two different user profiles belonging to different SNs without using the smartphone. This is particularly useful in case the smartphone is not available. Thus the verification for a subject can be done using another SN account. Also in this case the sets $P_i$ and $V$ are previously uploaded and downloaded on each SNs. However, we use the subset $P_i$, downloaded from a SN, to extract the $FP_i$ for each source $i$ so as to classify the images in $V$ that have been downloaded from another SN. For example, a Facebook profile is linked to a Google+ profile exploiting their respective shared photos, as shown in Figure 5.

The latter case is also useful for performing another type of investigative activity. That is to verify the source (smartphone) of the published images. This is particularly useful in matching (fake) user profile with a smartphone and transitively with a user. In this case, the $V$ represents the set of uploaded and downloaded images on and from each SNs, and $P_i$ as set of posted images. We again use the subsets $P_i$ to extract the $FP_i$ for each source $i$ so as to classify the original images in $V$.

For this purpose we perform all the possible combinations exploiting the SNs chosen, as shown in Figure 6.

4 EVALUATION

In Section 3.3, we have described three scenarios in order to evaluate the capabilities of our method. For each of them, namely i) Original-By-Original, ii) Social-By-Social and iii) Cross-Social, we performed a specific test
Cross-social test. Each arrow from a point A to a point B means that we use the subset $P_i$ of A to extract $FP_i$ and classify the images in $V$ of B, for example the arrow (b2) means that we use the subset $P_i$ downloaded from Facebook to classify the image in $V$ downloaded from Google+. We perform all the twelve combinations.

whose results are presented in this section. In particular, we use two well known statistical indexes to assess the classification process, that is sensitivity (SEN) and specificity (SPE). In our specific context, the sensitivity identifies the capability of the method to rightly recognize the source of the given images, while the specificity is the capability to reject the images that does not belong to the source under evaluation.

Out of all the images $C_i$ to be classified as belonging to the source $i$, let $C^+_i$ be the subset of images that our algorithm has successfully assigned. We define sensitivity as:

$$SEN = \frac{|C^+_i|}{|C_i|}$$

Out of all the images $\hat{C}_i = V \setminus C_i$ to be classified as not belonging to the source $i$, let $\hat{C}^-_i$ the subset of images that our algorithm has successfully rejected. We define specificity as:

$$SPE = \frac{|\hat{C}^-_i|}{|\hat{C}_i|}$$

4.1 Test 1: Original-By-Original

We perform this first test to determine the minimum cardinality of each set $P_i$, with which we can correctly extract the pattern noise. We start with a single image, and then the cardinality is increased in the sequence of: 2, 3, 4, 5, 10, 15, 20, 40, 60, 80, 100, 120 and 140, where each cardinality defines a subtest $P_i$ for each device. We fix the limit to 140 due to the resource constraints. We perform two steps process to evaluate the specific cardinality. In the first one, we extract the pattern noise and calculate the threshold. While in the second one, we classify the image and evaluate the SEN and the SPE. For each cardinality under evaluation, this procedure is carried for all the devices.

Figure 7 shows the computed values of mean between sensitivity and specificity values with an increase in cardinality for each smartphone. The obtained results suggest to fix the cardinality of the set $P_i$ to 100 images. The motivation to select $P_i = 100$ is based on following two reasons: (i) starting from this value, the mean between the sensitivity and the specificity index for each source has a stability of over 97%; (ii) a larger cardinality value to help the BM3D function in discriminating the high frequencies. By using 100 images for each smartphone to define the fingerprint $FP_i$, the set $V$ is composed of 500 images.

The cardinality is fixed to 100 images and we compute the SEN and SPE indexes for each smartphone. We represent the sensitivity using the confusion matrix (see Table 3), which is a standard representation for statistical clas-
sifier to highlight the percentage error. Each column and row identifies the device ID in Table 1. The main diagonal shows the percentage of images correctly assigned, while the upper and lower triangles of the matrix show the incorrect assignments. Along with matrix, we augment two columns to show SEN and SPE values for each smartphone. As shown in the Table, the method reaches 100% sensitivity for each device. In other words, it has verified the source in the original–by–original test. Also, the result returns the confidence value of at least 95% for specificity. It means that the method correctly discards at least the 95% of images that belong to a different source compared to the right one.

Table 3. The original-by-original confusion matrix.

<table>
<thead>
<tr>
<th>ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>SEN</th>
<th>SPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>95.47</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>96.62</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>95.47</td>
</tr>
<tr>
<td>4</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
<td>97.80</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>100.00</td>
<td>95.01</td>
</tr>
</tbody>
</table>

4.2 Test 2: Social-By-Social

In the second test we classify the images downloaded from the three different SNs using the threshold computed after the downloading operation. The results we obtain for Facebook, Google+ and WhatsApp are shown in Tables 4, 5 and 6 respectively. As in the original-by-original case, we present the confusion matrix along with SEN and SPE indexes values. Compared to the other two SNs, Google+ does least compression of the images (see Table 2). This characteristic justifies the highest sensitivity and specificity average values obtained in case of Google+ for each smartphone, that is 100% and 97.56% respectively. Facebook compresses the images more than Google+. However, the method has obtained good sensitivity and specificity average values, that is 96.92% and 91.58% respectively. WhatsApp is a mobile application mainly conceived for smartphones. For this reason WhatsApp requires a higher compression level than Facebook and Google+, whose use is mainly done with large screen computers with medium/high quality definition. This higher compression level significantly alters the information content. As a result, WhatsApp returns worst sensitivity and specificity average results, that is 92.52% and 90.84% respectively.

Table 4. The Facebook confusion matrix.

<table>
<thead>
<tr>
<th>ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>SEN</th>
<th>SPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>95.01</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>93.46</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>93.88</td>
</tr>
<tr>
<td>4</td>
<td>4.23</td>
<td>2.82</td>
<td>2.82</td>
<td>85.92</td>
<td>4.23</td>
<td>85.92</td>
<td>90.91</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>100.00</td>
<td>91.32</td>
</tr>
</tbody>
</table>

Table 5. The Google+ confusion matrix.

<table>
<thead>
<tr>
<th>ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>SEN</th>
<th>SPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>99.01</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>99.50</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>97.32</td>
</tr>
<tr>
<td>4</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
<td>98.52</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>100.00</td>
<td>93.46</td>
</tr>
</tbody>
</table>

Table 6. The WhatsApp confusion matrix.

<table>
<thead>
<tr>
<th>ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>SEN</th>
<th>SPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>85.33</td>
<td>2.67</td>
<td>4.00</td>
<td>8.00</td>
<td>0.00</td>
<td>85.33</td>
<td>91.53</td>
</tr>
<tr>
<td>2</td>
<td>5.19</td>
<td>94.81</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>94.81</td>
<td>93.62</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>3.23</td>
<td>96.77</td>
<td>0.00</td>
<td>0.00</td>
<td>96.77</td>
<td>90.87</td>
</tr>
<tr>
<td>4</td>
<td>4.76</td>
<td>3.17</td>
<td>6.35</td>
<td>85.71</td>
<td>0.00</td>
<td>85.71</td>
<td>89.47</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>100.00</td>
<td>88.69</td>
</tr>
</tbody>
</table>

Table 4, 5 and 6 provide detailed information about sensitivity and specificity for social-by-social test. Also, we plot in the same 3D graph the SEN and SPE values of the original images with the downloaded ones. Figure 8 shows the comparison of original-by-original with social-by-social for each smartphone. On an average, the method is able to verify which device has taken and uploaded the images on the SNs with 96.48% sensitivity and 93.77% specificity.

4.3 Test 3: Cross-Social

The last test is the most interesting one. In this case the pattern noise, characteristic of the
source, has been extracted from the subsets $\mathcal{P}_i$ not belonging to the same category of images (original or published on SN) of the set $\mathcal{V}$.

Following the schema defined in Figure 6 we perform all the tests combining all possible ways using the images downloaded from the SNs and the original ones. Figures 9, 10, 11 and 12 present the SEN and SPE results obtained from each combination. In each triangular histogram the icon in the middle identifies the selected category for the subset $\mathcal{P}_i$. While the three icons on the sides represent the categories that contributed to form the set $\mathcal{V}$ of images to be classified.

As discussed above, intuitively, if $FP_i$ and $TP_i$ are computed using higher quality images the method obtains the best results. That is the case of original images and those downloaded from Google+, as shown in Figure 9 and Figure 11. Although Facebook compresses the images stricter than Google+, the results are still good. The sensitivity results lie on high values, however, the average specificity value decreases (see Figure 10).

WhatsApp has the highest compression level among all the three SNs however, our method has correctly resolved WhatsApp user profiles with the other two SNs profiles, with an average sensitivity of 98.78%, as shown in Figure 12.

The overall sensitivity values are really close to 100%, except in the Facebook – WhatsApp case. In this particular combination, the sensitivity has presented deterioration, due to the medium/low image quality level of both SNs. While the average specificity value is over 92% in all categories (original 95.80%, Facebook 92.39%, Google+ 95.09% and WhatsApp 94.04%). Based on the results of this test, we can conclude that the proposed method has a success rate over the 90% for matching profiles across SNs.

5 CONCLUSIONS AND FUTURE WORKS

Fake profile identification is an important research problem in the domain of cyber security. This is particularly meaningful for crime prevention and detection activities coupled with smartphone device [1]. In this paper, we have presented a method for matching operations of SNs’ user profiles using the images shared on these platforms, in order to address the fake profiles problem. We chose five smartphones of two different brands with two pairs of identical models and three SNs with different compression characteristics of the image. We have shown that it is possible to verify the device which has taken and published images on a specific user profile among Facebook, Google+ or WhatsApp. We used only images from the SNs to extract the pattern noise and verify the source camera. Finally, we demonstrate how distinct user profiles belonging to
The threshold is computed using the subset $P_i$ composed by the original images. It is used to classify the images downloaded from WhatsApp (top left), Facebook (top right) and Google+ (bottom), respectively (a2), (a1) and (a3) in Figure 6.

The threshold is computed using the subset $P_i$ downloaded from Google+. It is used to classify the images downloaded from WhatsApp (top left) and Facebook (top right) and the original images (bottom), respectively (c3), (c2) and (c1) in Figure 6.

The threshold is computed using the subset $P_i$ downloaded from Facebook. It is used to classify the images downloaded from Google+ (top left) and WhatsApp (bottom) and the original images (top right), respectively (b2), (b3) and (b1) in Figure 6.

The threshold is computed using the subset $P_i$ downloaded from WhatsApp. It is used to classify the images downloaded from Google+ (top left) and Facebook (top right) and the original images (bottom), respectively (d3), (d2) and (d1) in Figure 6.

different SNs can be mapped using the images from the SN. In practice, we are able to create a fingerprint of the user profile.

The method proposed in this article may fail with increase in the number of devices, user profiles and different characteristics of SNs. To overcome this we plan to explore the definition of a more general purpose methodology that allows to decide if two user profiles $U_x$ and $U_y$ belong to the same user. As a future work, we plan to include a clustering algorithm in our existing approach. In this way, we intend to solve several other problems, for example images of the same user taken from different sources and classification of images with different resolution taken from the same source (front and rear camera). Future works also include testing our approach on a large set of heterogenous de-
vices. As video sharing is also a common behavior on SNs, we would also like to test our method on frames extracted from videos.

6 ACKNOWLEDGMENTS

This work has been supported in part by the Italian Ministry of Education, Universities and Research IMPACT project (RBFR107725) and OPLON project (SCN_00176).

REFERENCES


