User profiles' image clustering for digital investigations

User Profiles' Image Clustering for Digital Investigations

Dr Rahimeh Rouhi[a,*], Dr Flavio Bertini[b] and Professor Danilo Montesi[b]

[a] *Université de Lorraine, CNRS, LORIA, F-54000 Nancy, France*
[b] *Department of Computer Science and Engineering, University of Bologna, Italy*

ARTICLE INFO

Keywords:
sensor pattern noise
camera fingerprinting
clustering
digital evidence analysis
social network
digital investigations

ABSTRACT

Sharing images on Social Network (SN) platforms is one of the most widespread behaviors which may cause privacy-intrusive and illegal content to be widely distributed. Clustering the images shared through SN platforms according to the acquisition cameras embedded in smartphones is regarded as a significant task in forensic investigations of cybercrimes. The Sensor Pattern Noise (SPN) caused by camera sensor imperfections due to the manufacturing process has been proved to be an effective and robust camera fingerprint that can be used for several tasks, such as digital evidence analysis, smartphone fingerprinting and user profile linking as well. Clustering the images uploaded by users on their profiles is a way of fingerprinting the camera sources and it is considered a challenging task since users may upload different types of images, i.e., the images taken by users' smartphones (*taken images*) and single images from different sources, cropped images, or generic images from the Web (*shared images*). The *shared images* make a perturbation in the clustering task, as they do not usually present sufficient characteristics of SPN of their related sources. Moreover, they are not directly referable to the user's device so they have to be detected and removed from the clustering process. In this paper, we propose a user profiles' image clustering method without prior knowledge about the type and number of the camera sources. The hierarchical graph-based method clusters both types of images, *taken images* and *shared images*. The strengths of our method include overcoming large-scale image datasets, the presence of shared images that perturb the clustering process and the loss of image details caused by the process of content compression on SN platforms. The method is evaluated on the VISION dataset, which is a public benchmark including images from 35 smartphones. The dataset is perturbed by 3000 images, simulating the *shared images* from different sources except for users' smartphones. Experimental results confirm the robustness of the proposed method against perturbed datasets and its effectiveness in the image clustering.

1. Introduction

In recent years, different Social Networks (SNs) have revolutionized the Internet and our society by providing different types of interaction, for instance by sending texts and sharing images and videos. Many SNs provide their own dedicated applications for major mobile devices (e.g. smartphones). This influences user habits regarding multimedia content on SNs, (Norouzizadeh Dezfouli et al., 2016). In particular, this has led users to take more digital images and share them across various SNs, (Liu et al., 2012), making it a challenging task to control image production and propagation and to use such images as a form of digital evidence.

Images shared by SN users can be considered as complementary clues used to detect evidence in digital investigations, e.g., identity theft, online sexual harassment, piracy, cyber stalking and cyber terrorism, (Huang et al., 2018). In tracing the history of an image, identifying the source which captured the image is of major interest and the task is more challenging when the original images and smartphones are not available. In practice, it cannot always be assumed that prior information about the original image is available. Important clues on the camera source can be easily found in Exchangeable Image File Format (EXIF) data, (Cox et al., 2002). However, since this information can be simply modified or can be removed by online SN platforms (due to user's privacy), it cannot always be applied to digital evidence analysis and digital investigations. Hence, blind analysis has to be applied to investigate the right source of an image. The blind analysis has attracted a growing interest of researchers during the last years. Particularly, blind techniques exploit traces left by different processing steps in source manufacturing and the image acquisition and storage phases.

The Sensor Pattern Noise (SPN), due to camera sensor imperfections, is considered as a unique characteristic to fingerprint a source camera and it remains as residual noised in the images (Lukas et al., 2006). Regarding the SPN,

*Corresponding author
rahimeh.rouhi@loria.fr (R. Rouhi); flavio.bertini2@unibo.it (F. Bertini); danilo.montesi@unibo.it (D. Montesi)
ORCID(s): 0000-0002-0067-7455 (R. Rouhi); 0000-0001-6925-5712 (F. Bertini); 0000-0002-4748-6867 (D. Montesi)

different techniques such as SPN-based image clustering, (Lin and Li, 2017) and user profile linking, (Bertini et al., 2015) have been presented in digital investigations.

In most real-life cases in digital investigations e.g., Internet child pornography, a large number of images on a set of user profiles are collected, but sources by which these images are taken are not available. Moreover, this set of images might be perturbed by images that may not directly referable to the user's device. Clustering the collected images based on the residual noised extracted from the corresponding images, into an unknown number of groups can be a way to associate different crime scenes. It can provide the investigators by more clues to link the evidence to the seized hardware that are owned by the suspects, in the future.

In data analysis, an outlier is an object that differs significantly from other objects in a dataset, (Grubbs, 1969). Typically, outliers are a minority of objects that are inconsistent with the pattern presented by the majority of objects in the same dataset, (Taha and Hadi, 2019). Cluster analysis and outlier detection are strongly coupled tasks. The obtained clusters can be simply destroyed by a few outliers. Outliers are defined by the concept of the cluster and they are recognized as the objects which are not assigned to any cluster. Most of the existing clustering methods, generally and also specifically presented for the application of SPN-based image clustering, assume that all the objects (images in our case) should be assigned to a cluster label, meaning that there are no phases for outlier detection in the clustering methods. This is not always true, especially for the unsupervised clustering. The outliers unavoidably degrade the clustering effectiveness. For instance, only few outliers can destroy the clusters computed from k-means algorithm and generate bizarre distributions of Gaussian mixture model, (Liu et al., 2018). Consequently, the clustering should be provided with an outlier detection phase.

In this paper, we propose a user profiles' image clustering method that does not need prior knowledge about the type and number of the camera sources. We cluster both types of images uploaded buy users on SN platforms, i.e., the images taken by users' smartphones (*taken images*) and images from different sources, cropped images, or images from the Web, such that they cannot be used in fingerprinting their sources (*shared images*). The proposed method detects and removes the *shared images* (i.e., the outliers) and then clusters the remaining images into an unknown number of clusters according to the number of cameras which leads to fingerprinting the users' smartphones.

Given a set of *taken images* and *shared images* uploaded by users on SN platforms, the main contribution of this paper is as follows:

- we apply outlier detection to detect and remove *shared images* that do not present sufficient residual noises to fingerprint their right sources.
- we cluster *taken images* into an unknown number of groups, each of them including RN extracted from images coming from the same source.
- we investigate if the number of *shared images* is increased, how it affects the effectiveness of clustering of *taken images*.

The method clusters the residual noises based on the combination of hierarchical and Markov clustering algorithms and an adaptive threshold, which is updated according to the quality of the resulted clusters in each iteration of the algorithm. This makes no need to compute full-pairwise correlation matrix in the clustering. The method is evaluated on the VISION dataset (Shullani et al., 2017), which is a public benchmark including images from 35 smartphones. The dataset is perturbed by 3000 images, simulating the *shared images* from different sources except for users' smartphones. Experimental results confirm the robustness of the proposed method against perturbed datasets and its effectiveness in the image clustering. We show that the proposed method is stable against the number of the *shared images*, the loss of image details caused by the process of image compression applied by SN platforms, that degrades the quality of the SPN, and it is scalable in the number of images.

The rest of the paper is organized as follows. In Section 2, some related clustering works are discussed. In Section 3, the proposed method is explained. Section 4 introduces the datasets, evaluation measures and the results of parameter setting and the proposed clustering method on different SN datasets. In Section 5, significance and limitations of the proposed method are discussed. Finally, in Section 6, conclusions is presented.

2. Related works

Generally speaking, any inherent traces left in the image by the processing components, either hardware or software, of the image acquisition pipeline, such as defective pixels (Kurosawa et al., 1999; Geradts et al., 2001), color filter array (CFA) interpolation artifacts (Bayram et al., 2005; Swaminathan et al., 2007), JPEG compression artifacts (Sorrell, 2009; Alles et al., 2009) lens aberration (Choi et al., 2006; Van et al., 2007) or the combination of several image

intrinsic characteristics (Kharrazi et al., 2005; Çeliktutan et al., 2008) could be applied to associate the images to their source camera. Apart from the above-mentioned techniques, the methods that attract the most attention may be those based on SPN (Lukas et al., 2006; Chen et al., 2008; Filler et al., 2008; Goljan et al., 2009; Li, 2010a; Wu et al., 2012), which mainly consists of the photo-response non-uniformity (PRNU) noise (Lukas et al., 2006) arising primarily from the manufacturing imperfections and the inhomogeneity of silicon wafers. The uniqueness to individual camera and stability against environmental conditions make SPN a feasible fingerprint for identifying and linking source cameras.

Many works have been done on SPN-based image clustering. As a pioneering work, (Bloy, 2008) presented a clustering algorithm, considering the residual noises as singleton clusters and hierarchically merging the similar clusters. The algorithm is based on the idea that the more images are clustered, the better the quality of SPNs can be obtained. As a drawback, the algorithm produces the threshold based on a quadratic model, which does not generalize well across various source cameras. In (Li, 2010b), Markov random fields are used to assign a class label to an image iteratively, according to the consensus of a small set of SPNs, called membership committee. This raises an issue on how to choose a suitable committee, in particular for the datasets with different cluster cardinalities, i.e., asymmetric datasets. In (Caldelli et al., 2010), the authors developed a faster algorithm by proposing a new enhancer applied to the extracted SPNs. The algorithm merges the clusters hierarchically, and a silhouette coefficient is calculated for each cluster. The silhouette coefficient estimates the separation among clusters as well as the cohesion within each cluster. The average of the silhouette coefficients related to the produced clusters in each iteration is considered as a merit of the clustering which shows its quality. In (Villalba et al., 2016), a similar clustering algorithm was proposed. The main difference is that the evolutionary process of the cluster formation is used in the calculation of the coefficient. The main problem of the hierarchical algorithms is that they are sensitive to noise and outliers as wrong assignments may propagate the error to the following iterations in the clustering. Also, their computational complexities are high especially for high dimensional residual noises because all the cluster pairs have to be checked for a merging.

The graph based algorithms have been applied successfully to SPN-based image clustering. In (Liu et al., 2010), a method based on k-nearest neighbor technique was proposed, where the clustering is regarded as a graph partitioning problem. Each image is considered as a node and the correlation values between the residual noises are considered as the weights of the edges. Next, the nodes are partitioned into disjointed sets by using spectral analysis. Besides the need for the user to provide the number of clusters, a major problem of this method is that its quality is dependent on the random initialization. In (Amerini et al., 2014), the problems were addressed by using the normalized cut graph partitioning algorithm, (Shi and Malik, 2000), which resulted better clustering results without providing the number of clusters as an input parameter. In (Marra et al., 2016), the clustering is performed based on correlation clustering, which formulates the graph partitioning problem as constrained energy minimization. The issue of this algorithm is that it needs a parameter set by the user according to preliminary analyses on an appropriate training set. The issue was handled in (Marra et al., 2017) by consensus clustering applied to all the cluster partitions obtained from correlation clustering, to extract a unique solution. Generally, the average correlations between SPNs of one camera may remarkably differ from that of other cameras, which makes the clustering more difficult. To tackle the issue, in (Li and Lin, 2017), shared nearest neighbors, (Ertöz et al., 2003), are applied to the full-pairwise correlation matrix, to find clusters with different sizes and densities. A common undesirable trait of the algorithms mentioned above is their need for full-pairwise correlation matrix, which may prevent their use for large-scale datasets in practical applications.

Only a few studies considered the scalability aspect of SPN-based image clustering. In (Lin and Li, 2017), large-scale clustering was handled by partitioning the dataset into small batches, which could fit in RAM efficiently, and applying a coarse-to-fine clustering method. An adaptive threshold was proposed for merging the obtained clusters based on the quality of the clusters iteratively updated during the clustering. The authors of (Phan et al., 2018) used the similar partitioning approach and exploited linear dependencies among SPNs in their intrinsic vector subspaces. It uses a training phase to generate an adaptive threshold for merging the obtained clusters. However, the training may lead to over-fitting in some datasets. Also, compared with the threshold proposed in (Lin and Li, 2017), it tends to be too radical for clusters with large size, which is a critical point in real-life applications. These scalable clustering algorithms were only tested on *native images*, not compressed by SN platforms, and their robustness on images compressed on Social Networks (SNs) is still in doubt.

Outlier detection, also known as anomaly detection, recognizes the objects deviated from the others and identifies these objects as outliers. In most of the existing works, unsupervised outlier detection was studied, in such a way that each object is given a score based on some criteria, and the objects with large scores are considered as the outlier candidates, (Liu et al., 2018). Some representative methods include density based Local Outlier Factor (LOF), (Breunig et al., 2000), Connectivity-based Outlier Factor (COF), (Tang et al., 2002), Local Distance-based Outlier Factor

(LODF), (Zhang et al., 2009), Frequent Pattern-based outlier detection (Fp-outlier), (He et al., 2005), ensemble-based isolation Forest (iForest), (Liu et al., 2008), Oversampling Principal Component Analysis (OPCA), (Lee et al., 2012), and cluster-based Text Outliers using Non-negative Matrix Factorization (TONMF), (Kannan et al., 2017). Recently, some methods based on deep learning were proposed, such as deep one-class SVM, (Ruff et al., 2018) and Generative Adversarial Networks (GAN)-based methods, (Li et al., 2018), learning a non-linear transformation to project the original data into hidden space, for more effective recognition. These methods are supervised and train the model only with accurate samples and predict the label of new samples whether they are outliers or not. Hence, the training phase is crucial and challenging. Although clustering and outlier detection are mostly a coupled task in the real-life applications, there are few works presented a unified framework for cluster analysis and outlier detection. For example, a developed version of k-means algorithm was proposed in (Chawla and Gionis, 2013), called k-means--, which detects outliers and groups the remaining objects into k clusters, where the objects with large distance from the nearest centroid are considered as outliers during the clustering process. Langrangian Relaxation (LP), presented in (Ott et al., 2014), formulates the clustering task with outliers as an integer programming problem, which requires the cluster creation costs as the input parameter. (Charikar et al., 2001) proposed a bi-criteria approximation algorithm for the facility location with outliers problem. Reference (Chen, 2008) proposed a constant factor approximation algorithm for the k-medoids clustering with outliers.
Gisolf et al., (Gisolf et al., 2014) introduced a high speed common source identification on a 'standard' desktop computer to tackle high computational cost in the comparison done to find out which images originate from the same source in the clustering. They demonstrated that by applying several time-saving methods – grayscale conversion, digestion, quantization and the modified NCC formula – one can analyze a large database in forensically relevant time, without resorting to large and expensive computer clusters. They focused on the optimization of the time needed per comparison.

Although some pioneering works introduced new approaches for joining clustering and outlier detection phases, none of these algorithms, except k-means--, are applicable to large-scale datasets. However, the spherical structure assumption of k-means-- and the original feature space limit its capability for clustering complex data. Liu et al. (Liu et al., 2018) introduced a Clustering with Outlier Removal (COR) method, which partitions an entire dataset into several clusters and one outlier cluster, separately. The COR method transforms the original feature space into the partition space, where according to Holoentropy, the COR is designed to provide simultaneous consensus clustering and outlier detection. To the best of our knowledge, only the work presented in (Phan et al., 2018) considered the outlier detection, based on Density-Based Spatial Clustering with Applications with Noise (DBSCAN) algorithm, (Ester et al., 1996), and evaluated their method robustness against the images which come from different sources and do not present sufficient characteristics of their sources. However, their method was not evaluated in such a case that the number of outliers exceeds the number of the other images. When dataset is perturbed by a large number of images shared form other sources except users' smartphones, an effective and efficient clustering algorithm is needed for detection and removal of these images, i.e., *shared images* and *taken images*, such that the clustering of the the images taken by user's smartphones is not affected negatively.

3. Proposed method

We present a hierarchical graph based clustering method to detect and remove the *shared images*, performed in the first steps of the clustering, and cluster the remaining images, i.e., the *taken images* based on their acquisition smartphones. The flowchart of the proposed method is presented in Figure 1.

3.1. Preparation

Camera sensor imperfections remain stable as the residual noises in the images. Each residual noise is the difference between the image content and its denoised version acquired by a de-noising filter $d()$. The residual noise of an image I is extracted as follows, (Lukas et al., 2006):

$$\mathrm{RN} = I - \mathrm{d}(I) \tag{1}$$

by averaging the residual noises extracted from n images taken by a given smartphone, the SPN, i.e., the camera fingerprint, can be approximated by:

$$\mathrm{SPN} = \frac{1}{n}\sum_{j=1}^{n} \mathrm{RN}_j \tag{2}$$

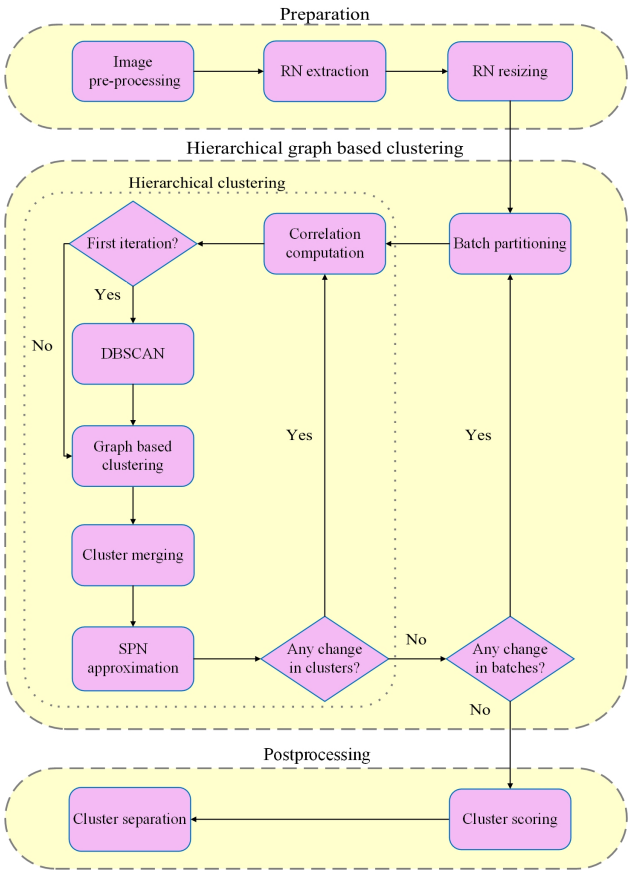

Figure 1: Flowchart of the proposed method.

Based on (1) and (2), it can be seen that the quality of the extracted residual noises and SPN is dependent on $d()$ and n. Block-Matching and 3D (BM3D) de-noising filter introduced by (Dabov et al., 2007) is an accurate way to extract the residual noises with better qualities. Through BM3D, non-unique artifacts are removed by using zero-meaning all columns and rows, and Wiener filtering in the Fourier domain, (Chen et al., 2008), (Lin and Li, 2016; Chierchia et al., 2010).

The extracted residual noises are re-sized to a specific resolution. It is needed for computation of similarities between the residual noises. In order to reduce the use of RAM and to make the algorithm scalable, we follow the approach presented in (Lin and Li, 2017). Let N be the total number of residual noises in the dataset. The pre-processed residual noises are randomly partitioned into t batches, i.e., $B = \{b_1, b_2, ..., b_t\}$, where $t = \lceil \frac{N}{q} \rceil$ and q is the batch size. The parameter q is determined regarding the available size of RAM. For each batch, a correlation matrix $\mathcal{A}$ is created, with each element $\mathcal{A}(i, j)$ being NCC similarity between any two SPNs in the batch, calculated by (3). The Normalized Cross Correlation (NCC) similarity between any two camera fingerprints $f_i = [x_1, ..., x_l]$ and $f_j = [y_1, ..., y_l]$ is calculated as follows:

$$\mathcal{A}(i,j) = \frac{\sum_{n=1}^{l}(x_n - \overline{f}_i)(y_n - \overline{f}_j)}{\sqrt{\sum_{n=1}^{l}(x_n - \overline{f}_i)^2 \sum_{n=1}^{l}(y_n - \overline{f}_j)^2}} \tag{3}$$

where $\overline{f}_i$ and $\overline{f}_j$ represent the means of the two fingerprints, respectively. Generally, the correlations between the fingerprints from the same camera are higher than those from different cameras (Fahad et al., 2014).

3.2. Shared images removal based on DBSCAN

Outlier detection is a pre-clustering step that is usually performed in many density based clustering algorithms. Through a density-based approach, the clusters of arbitrary shapes can be found. For instance, it can even find a cluster completely surrounded by (but not connected to) a different cluster. DBSCAN is a density-based clustering non-parametric algorithm proposed in (Ester et al., 1996). DBSCAN does not require to be provided with the number of clusters in the data a priori, unlike k-means, k-medoids and hierarchical clustering. Giving some objects, DBSCAN groups together the objects and marks these which lie alone in low-density regions as the ouliers. It finds the object's neighbors by density ϑ on an n-dimensional sphere with radius ϵ. The parameter ϑ is defined as the minimum number (a threshold) of the objects huddled together for a region to be considered dense, and ϵ is a parameter specifying the radius of the neighborhood. A cluster can be defined as the maximal set of density connected objects in the space. DBSCAN is a strong and effective method when the distribution of values in the feature space can not be assumed. Hence, it can handle the process of clustering the high dimensional residual noises specified by a large feature vector and detect outliers. DBSCAN is robust to outliers.

We apply DBSCAN to the clustering just before the Markov clustering is performed and for the first iteration of the processing of each batch, see Figure 1. Once the batch is purified by removing the discovered outliers, it is passed to the following stages to cluster the remained residual noises into unknown number of clusters, each of them including residual noises coming from the same smartphone. It is worth mentioning that the larger the size is considered for each batch the better the effectiveness of the outlier detection is concluded, as the residual noises are compared in a larger scale.

3.3. Hierarchical clustering

The clustering for each batch starts by considering each residual noise as a singleton cluster. It is performed in an agglomerative hierarchical way by iteratively merging the similar clusters. At the end of each iteration of the hierarchical clustering, the camera fingerprints corresponding to the merged clusters are updated according to (2). Then, the obtained clusters from all the batches are grouped, and they are hierarchically partitioned and clustered until no new cluster is found, see Figure 1. The hierarchical clustering has some drawbacks. A wrong assignment may propagate the error to the following iterations in the clustering. Moreover, its computational burden is high as it has to check all the pairs of clusters for merging, (Shirkhorshidi et al., 2014). In our proposed method, to handle the mentioned drawbacks, we combine the hierarchical algorithm with the Markov clustering algorithm. We also embed a cluster merging step based on an adaptive threshold to achieve precise and fast clustering.

3.4. Graph based clustering

Markov clustering is a fast and scalable graph based unsupervised learning algorithm, proposed in (Van Dongen, 2008). It has successfully been applied to different fields of science. It considers the objects as the vertices of a graph, e.g., Q, and groups them regarding the weights of the edges, i.e., the similarities between the objects (Mesa, 2012). The Markov matrix $\mathcal{M}$ corresponding to the graph Q is defined by normalizing all the columns of the graph adjacency matrix. A random walk is simulated over the vertices of the graph to increase and decrease the flow in strong and weak currents in Q, respectively, (Varia, 2013). The random walk can be modeled as a Markov chain on the graph Q. Starting from a vertex, a random walk is more likely to arrive at the vertices within the same cluster than those in different clusters. The vertices of Q are considered as a set of states $S = \{s_1, s_2, ..., s_n\}$, and the graph edges are associated with the transition probabilities in $\mathcal{M} = [\mathfrak{p}(i, j)] \in \mathbb{R}^{n\times n}$, where each element at index (i, j) is the transition probability from vertex i to vertex j and

$$\sum_{i=1}^{n} \mathfrak{p}(i, j) = 1,\ 0 \leq \mathfrak{p}(i, j) \leq 1 \tag{4}$$

By applying expansion and inflation operators to $\mathcal{M}$ alternatively, the clusters can be considered from the resulting transition matrix at the converged state. The expansion operator performed based on matrix multiplication simulates a random walk on the graph Q by:

$$\mathcal{M}_{exp} = \mathcal{M}^e \tag{5}$$

where e is the expansion parameter. The j^{th} column of $\mathcal{M}_{exp}$ can be interpreted as the probability distribution of the j^{th} of random walk. Subsequently, the inflation operator is performed on each element of the matrix $\mathcal{M}_{exp}$ as follows:

$$\mathcal{M}_{inf}(i,j) = \frac{\mathcal{M}_{exp}(i,j)^{\eta}}{\sum_{k=1}^{n} \mathcal{M}_{exp}(k,j)^{\eta}} \tag{6}$$

By the inflation operator, the elements of the matrix $\mathcal{M}_{exp}$ are raised to the power of the inflation parameter η, and then the columns are normalized. In each column, the elements which have very small values (less than a predefined value ς) are removed, and the remaining elements are re-scaled, to make the sum of each column equal to 1. This is called pruning which is defined as follows:

$$\mathcal{M}_{pru}(i,j) = \begin{cases} 0, & \mathcal{M}_{inf}(i,j) < \varsigma \\ \mathcal{M}_{inf}(i,j), & \text{otherwise} \end{cases} \tag{7}$$

The pruning decreases the number of non-zero elements in $\mathcal{M}_{inf}$, which subsequently reduces the memory usage and accelerates the clustering, (Satuluri, 2012). Global chaos $\mathcal{G}$ shows the rate of the changes in the probability values related to each of the two consecutive iterations. The algorithm stops once the global chaos approximately is zero. The value of $\mathcal{G}$ is calculated according to the maximum value of chaos denoted as $\mathcal{C}_j$ on every column j of $\mathcal{M}_{pru}$, (Bustamam et al., 2012).

$$\mathcal{C}_j = \frac{\max\limits_{i=1,2,..,n} \mathcal{M}_{pru}(i,j)}{\sum_{i=1}^{n} \mathcal{M}_{pru}(i,j)^2} \tag{8}$$

$$\mathcal{G} = \max_{j=1,2,..,n} \mathcal{C}_j \tag{9}$$

The details of the Markov clustering as the graph based clustering algorithm applied to the proposed method are presented in Algorithm 1.

Algorithm 1: Markov clustering algorithm.

input: Pairwise correlation matrix, $\mathcal{A}$
output: Probabilities matrix, $\mathcal{M}$
- expansion parameter: e
- inflation parameter: η
- global chaos: $\mathcal{G}$
- prune parameter: ς
- threshold for global chaos: ξ
- add self-loops to the graph $\mathcal{A}$, $\mathcal{A} = \mathcal{A} + \mathcal{I}$
- create the diagonal degree matrix of $\mathcal{A}$, $\mathcal{D}$
- create Markov matrix, $\mathcal{M} = \mathcal{A}\mathcal{D}^{-1}$
while $\mathcal{G} > \xi$ **do**
  - expansion on $\mathcal{M}$, based on (5)
  - inflation on $\mathcal{M}_{exp}$, based on (6)
  - pruning on $\mathcal{M}_{inf}$, based on (7)
  - update $\mathcal{G}$ based on (8) and (9)
  - $\mathcal{M}$=$\mathcal{M}_{pru}$
return $\mathcal{M}$

The Markov clustering receives the adjacency matrix $\mathcal{A}$, containing the similarities of the fingerprints, computed by (3), in one batch, as an input. It produces the probability transition matrix $\mathcal{M}$, such that each entry of the matrix represents the degree of the similarities between a pair of residual noises in the batch. The addition of self-loops to the input matrix $\mathcal{A}$ prevents the dependency of the flow distribution on the length of the random walk, which ensures the presence of at least one non-zero entry per column, (Satuluri, 2012). Considering two clusters c_i and c_j corresponding

to the vertices v_i and v_j in Q, if they share the same or similar fingerprint characteristics of a camera, the element $\mathcal{M}(i, j)$ is set to a non-zero value. Otherwise, it means the clusters are from different cameras and $\mathcal{M}(i, j)$ is set to 0. In every iteration of the hierarchical clustering, the Markov clustering is used to each batch, see Figure 1. By applying nearest neighboring to the columns of the obtained probability transition matrix, small cluster granularities are generated. These representative and precise clusters are considered as the candidate clusters, which are more likely to be from the same cameras. The candidate clusters are iteratively merged to discover larger clusters. This makes no need to compute full-pairwise correlation matrix.

3.5. Cluster merging

The matrix $\mathcal{M}$ of a batch may contain many sparse columns, i.e., the columns which are populated with many zero values. The reason could be partitioning the residual noises into batches randomly. So, only the clusters corresponding to non-sparse columns are kept, and the remaining clusters are passed to the next iterations to get a better chance for a merging, as the clusters are being evolved. In the implementation, we consider a column as non-sparse if the number of its non-zero elements is less than 20. For each non-sparse column corresponding to the cluster c_i, its nearest neighbor cluster, i.e., c_j is found based on the highest probability value existing in the column. Then, the clusters c_i and c_j are selected as the candidate clusters for a merging. Usually, the residual noises from the same model of smartphones present high correlations, and correspondingly high probabilities in $\mathcal{M}$ are produced. Accordingly, it is probable that they are selected as the candidate clusters. Therefore, to make the proposed method more precise, in the cluster merging, in addition to using the candidate clusters, we use an adaptive threshold. The adaptive threshold is updated based on the quality of the obtained clusters in each iteration of the hierarchical clustering. It exploits the idea that the more images from a given smartphone are precisely clustered, the better the quality of SPN can be estimated, (Bloy, 2008). As the cluster size grows, the inter-camera and intra-camera correlation distributions are normally more separable. Therefore, adaptively increasing the threshold can effectively prevent wrong merging of clusters, especially those from the same model of smartphones. The adaptive threshold $\mathcal{T}$ is defined as follows, (Lin and Li, 2017):

$$\mathcal{T} = \max(\tau, \frac{\psi\sqrt{n_{c_i} n_{c_j} \mu_{c_i}^2 \mu_{c_j}^2}}{\sqrt{[(n_{c_i} - 1)\mu_{c_i}^2 + 1][(n_{c_j} - 1)\mu_{c_j}^2 + 1]}}) \tag{10}$$

the parameter τ is a minimum threshold working as a trust boundary of $\mathcal{T}$. The terms n_{c_i} and n_{c_j} shows the number of the residual noises in the two clusters c_i and c_j, respectively, and ψ is a predefined scaling factor. The quality of the cluster c_i, that is μ_{c_i}, is defined as the mean of the correlation values between all the pairs of residual noises in the cluster. Given two candidate clusters c_i and c_j, if the correlation between the corresponding fingerprints f_i and f_j, which is calculated by (3), is greater than the adaptive threshold, i.e., $\mathcal{A}(f_i, f_j) > \mathcal{T}$, the clusters are merged. Otherwise, they are not merged and passed to the next iteration of the algorithm.

3.6. Post-processing

As the final phase of the proposed method, post-processing is applied to the resulted clusters. The method generates both fine and coarse clusters. While coarse clusters include a notable number of residual noises sharing the same camera fingerprints characteristics, the fine clusters include few residual noises which do not share sufficient similarities with the obtained camera fingerprints through the clustering. Due to the nature of the noise-like camera fingerprints, (Lin and Li, 2017), and particularly the low resolution of the uploaded images on SNs, the presence of the fine clusters are almost unavoidable. For the datasets with a variety of images coming from the same or different smartphone models and brands, merging the fine clusters into the coarse ones may cause a drop in the quality of the clustering. Accordingly, to present a more precise clustering tool for shared image analysis, we remove the fine clusters and preserve the coarse ones. To distinguish the coarse clusters from the fine ones, we introduce a size-based score ζ_i specified for each cluster as follows:

$$\zeta_i = \frac{|c_i|.|C|}{N} \tag{11}$$

where N is the number of RNs, and c_i is the i^{th} cluster in the resulted set of clusters, i.e., C, from the proposed method. If $\zeta_i \leq 1$, the cluster c_i is considered as a fine cluster, and it is excluded from C. Otherwise, we keep it as a coarse cluster. Given the explanations above, the algorithm of the proposed method is presented in Algorithms 2.

Algorithm 2: Proposed clustering algorithm.

input: pre-processed RNs
output: list of clusters, *shared images* C_O and C *taken images*
- number of RNs, N
- scaling factor, ψ in (10)
- minimum threshold, τ in (10)
- size of batches, q
- clustering initialization, $C_{old} = \{\}$
- considering a set of single clusters corresponding to the RNs, $C_{new} = \{c_1, c_2, ..., c_N\}$
- initializing a set of camera fingerprints with the residual noises corresponding to the clusters, $F = \{f_1, f_2, ..., f_N\}$
- partitioning initialization, $B_{old} = \{\}$
- $t = \lceil \frac{N}{q} \rceil$
- randomly partition C_{new} into t batches with size q, $B_{new} = \{b_1, b_2, ..., b_t\}$
- parameter for determining first iteration to apply DBSCAN, $flag = 1$
while $|B_{new}| \neq |B_{old}|$ **do**
  for $k = 1 : t$ **do**
    while $|C_{new}| \neq |C_{old}|$ **do**
      - compute correlation matrix $\mathcal{A}$ by (3)
      if $flag$ **then**
        - apply DBSCAN to $\mathcal{A}$
        - put the found *shared images* in the cluster C_O
        - extract the correlation matrix of normal RNs, i.e., $\mathcal{A}_N$
        - $\mathcal{A} = \mathcal{A}_N$
      - apply Markov clustering to $\mathcal{A}$ and generate the probability transition matrix $\mathcal{M}$ by Algorithm 1
      - put non-sparse column's indices in the list L
      for $i = 1 : |L|$ **do**
        - find the nearest cluster c_j to the cluster c_i from the list L
        - compute the adaptive threshold $\mathcal{T}$ by (10)
        if $\mathcal{A}(f_i, f_j) > \mathcal{T}$ **then**
          - merge clusters c_i and c_j
        else
          - continue
      - put the obtained clusters in C_{new}
      - update the camera fingerprints in F for the merged clusters by (2)
      - $C_{old} = C_{new}$
  - consider all the obtained clusters from batches as a new cluster C_{new}
  - $B_{old} = B_{new}$
  - $N = |C_{new}|$
  - update t, $t = \lceil \frac{N}{q} \rceil$
  - partition the clusters in C_{new} into t batches with size q, and form B_{new}
  - $flag = 0$
- post-processing
- $C = C_{new}$
return C_O and C

To summarize, the proposed clustering method starts with randomly partitioning the dataset into small batches. For each batch, the pairwise correlation matrix is calculated. DBSCAN is applied to each correlation matrix to detect outliers and each batch is purified. Then, Markov clustering algorithm is applied to the correlation matrix of the remaining RNs. By using the probability matrix, as the output of the Markov clustering, and nearest neighboring, the candidate clusters to be merged are selected, and subsequently an adaptive threshold is computed, for merging the clusters. The fingerprint for the merged clusters is updated through (2) and similar process is hierarchically performed on the merged clusters. The clustering stops once no new cluster is found. The resulted clusters are scored based on their sizes, and the coarse clusters, i.e., the clusters with a notable number of RNs sharing the same SPN characteristics, are stored as the final result.

4. Experiments and results

To validate the proposed method, we apply the VISION image dataset, (Shullani et al., 2017). After removing dark and saturated images which are not applicable to the clustering task, we have 7480 *Native* images taken by 35 smartphones. The images were uploaded and downlowded on the main SNs platforms such as *WhatsApp* (W), *Facebook High Resolution* (FH) and *Facebook Low Resolution* correspondingly we call them $\mathcal{V}^{\mathrm{NA}}$, $\mathcal{V}^{\mathrm{W}}$, $\mathcal{V}^{\mathrm{FH}}$, and $\mathcal{V}^{\mathrm{FL}}$. To see how the proposed method performs on the images coming from the identical models of smartphones, that represents a challenging task, we consider the subsets $\mathcal{V}_1^{\mathrm{N}} \subseteq \mathcal{V}^{\mathrm{N}}$, $\mathcal{V}_1^{\mathrm{W}} \subseteq \mathcal{V}^{\mathrm{W}}$, $\mathcal{V}_1^{\mathrm{FH}} \subseteq \mathcal{V}^{\mathrm{FH}}$ and $\mathcal{V}_1^{\mathrm{FL}} \subseteq \mathcal{V}^{\mathrm{FL}}$, each of them includes 2250 images from 11 smartphones, two iPhone 4S, two iPhone 5, three iPhone 5c, two iPhone 6, and two Samsung Galaxy S III models.

4.1. Experimental measures

The proposed algorithm is evaluated by different measures such as *Precision rate* $\mathcal{P}$, *Recall rate* $\mathcal{R}$ also known as *True Positive Rate* (TPR), *F1-measure* ($\mathcal{F}1$), *Rand Index* (RI), *Adjusted Rand Index* (ARI), *Purity*, *False Positive Rate* (FPR). Computing True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN), different measures can be obtained by (12) - (21):

$$\mathcal{P} = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FP}|} \tag{12}$$

$$\mathcal{R} = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FN}|} \tag{13}$$

$$\mathcal{F}1 = 2 \cdot \frac{\mathcal{P} \cdot \mathcal{R}}{\mathcal{P} + \mathcal{R}} \tag{14}$$

$$\mathrm{RI} = \frac{|\mathrm{TP}| + |\mathrm{TN}|}{|\mathrm{TP}| + |\mathrm{FP}| + |\mathrm{TN}| + |\mathrm{FN}|} \tag{15}$$

where $|.|$ shows the number of the pairs in the corresponding set defined in i-iv. The value of RI varies between 0 and 1, respectively showing no agreement and full agreement between the clustering results and the ground truth. For two random clusters, the average of $\overline{\mathrm{RI}}$ is a non-zero value. To get rid of this bias, ARI was proposed in (Hubert and Arabie, 1985):

$$\mathrm{ARI} = \frac{RI - \overline{RI}}{1 - \overline{RI}} \tag{16}$$

Also, we use *Purity* and FPR in the evaluations as follows:

$$Purity = \frac{\sum_{i=1}^{|C|} \frac{|\widehat{c}_i|}{|c_i|}}{|C|} \tag{17}$$

where $|C|$ is the number of the obtained classes, $\widehat{c}_i$ denotes the number of residual noises with the dominant class label in the cluster c_i, and $|c_i|$ is the total number of residual noises in c_i.

$$\mathrm{FPR} = \frac{|\mathrm{FP}|}{|\mathrm{FP}| + |\mathrm{TN}|} \tag{18}$$

In addition, in clustering, the ratio of the number of the obtained clusters that is n_h over the number of ground truth clusters denoted by n_g is calculated as follows:

$$\mathcal{N}_C = \frac{n_h}{n_g} \tag{19}$$

Table 1
Values of different parameters for proposed method.

Notation	Value	Description
q	1500	batch size for partitioning dataset
ϵ	0.95	minimum threshold for a dense region in DBSCAN
ϑ	20	threshold for neighborhood of each residual noise in DBSCAN
e	2	expansion parameter in (5)
η	1	inflation parameter in (6)
ς	0.005	prune parameter in (7)
$\mathcal{G}$	2	initial value of global chaos in (9) and Algorithm 1
ξ	0.3	threshold for global chaos in (9)
ψ	0.15	scaling factor in (10) for $\mathcal{V}_1^{\mathsf{NA}}$,$\mathcal{V}_2^{\mathsf{NA}}$ and $\mathcal{V}^{\mathsf{NA}}$
	0.09	scaling factor in (10) for $\mathcal{V}_1^{\mathsf{W}}$,$\mathcal{V}_2^{\mathsf{W}}$ and $\mathcal{V}^{\mathsf{W}}$
	0.07	scaling factor in (10) for $\mathcal{V}_1^{\mathsf{FH}}$,$\mathcal{V}_2^{\mathsf{FH}}$ and $\mathcal{V}^{\mathsf{FH}}$
	0.03	scaling factor in (10) for $\mathcal{V}_1^{\mathsf{FL}}$,$\mathcal{V}_2^{\mathsf{FL}}$ and $\mathcal{V}^{\mathsf{FL}}$
τ	0.004	minimum threshold for neighborhood in (10)

We calculate $\mathcal{N}_U$ for the clustering as well:

$$\mathcal{N}_U = \frac{n_u}{N} \tag{20}$$

where n_u is the number of unclustered RNs. Also, for the *shared images* detection, $\mathcal{N}_O$ is defined as follows:

$$\mathcal{N}_O = \frac{n_d}{n_o} \tag{21}$$

where n_d and n_o are the number of the detected and ground truth *shared images*, respectively. We evaluate the method by all the mentioned measures in (12) - (21). For the datasets covering a variety of smartphone models and brands, it is difficult to achieve the best values for all the mentioned measures. For example, merging the residual noises of different cameras into the same cluster increases FPR, which can propagate the error to the following iterations in the clustering. As a result, $\mathcal{P}$ and $Purity$ decrease, although $\mathcal{R}$ increases, (Lin and Li, 2017). We prefer to have the clusters with high values of $\mathcal{P}$, $Purity$, a low value of FPR, and accurate values of $\mathcal{N}_C$.

4.2. Experimental setting

Regarding the memory constraint, we choose the resolution of 1024×1024, for re-sizing all the residual noises. We need to set the parameters of the proposed method with the values which results in the best quality of *shared images* detection and subsequently the clustering. For example, for the parameters ϵ and ϑ, for the applied DBSCAN algorithm, we consider different ranges for ϵ and ϑ, which are respectively $[0.991, 0.993, 0.995, 0.997, 0.999]$ and $[10, 15, 20, 25, 30]$. We perform the parameter setting on the sample dataset $\mathcal{V}_0^{\mathrm{NA}} \subseteq \mathcal{V}^{\mathrm{NA}}$, including 3500 images, 100 images from each of 35 smartphones are considered. The dataset $\mathcal{V}_0^{\mathrm{NA}}$ is perturbed by 500 *shared images*. We evaluate the clustering, which is performed by selecting different values of the parameters, based on different measures defined by (12) - (21). From Figure 2, it can be seen if the values of ϵ and ϑ are set with 0.995 and 20, respectively, the best effectiveness for both clustering the *taken images* and detecting the *shared images* can be obtained. Setting ϵ with the values smaller than 0.995 results in the appropriate removal of residual noises of the *shared images*. Similar setting process was done for other parameters. The parameters and their values are listed in Table 1.

4.3. Experimental results

Table 2 presents the results of the proposed method on the datasets $\mathcal{V}_{\mathrm{NA}}$, $\mathcal{V}_{\mathrm{W}}$, $\mathcal{V}_{\mathrm{FH}}$ and $\mathcal{V}_{\mathrm{FL}}$ in terms of both clustering quality and outlier detection. The algorithm removed successfully the outliers with $\mathcal{N}_O > 90\%$ and also cluster the remained residual noises with high quality of different measures. We have set the number of outliers to 3000. To compare the results of DBSCAN algorithm, we have applied another outlier detection method which is Distance to K-Nearest Neighbor (DKNN), (Haque, 2019). Comparing the Tables 2 and 3, it can be seen that DBSCAN is more reliable in the detecting the *shared images*. In the literature of the outlier detection, e.g., in (Taha and Hadi, 2019), it

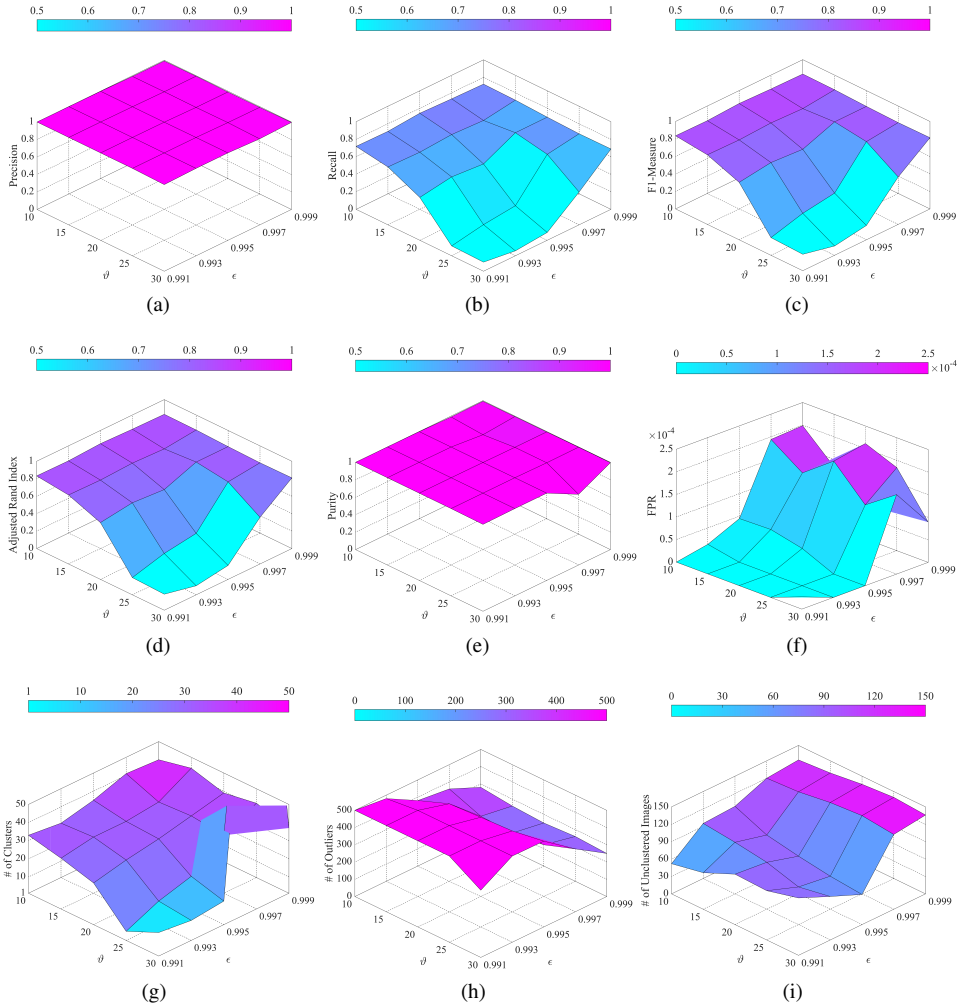

Figure 2: DBSCAN parameters ϵ and ϑ affect *shared images* detection and clustering on $\mathcal{V}_0^{\mathsf{NA}}$, perturbed by 500 *shared images*, (a) *Precision*, (b) *Recall*, (c) *F1-Measure*, (d) *Adjusted Rand Index*, (e) *Purity*, (f) *False Positive Rate*, (g) number of obtained clusters, (h) number of discovered *shared images*, and (i) number of unclustered images.

Table 2
Results (%) of the hierarchical graph based clustering method based on DBSCAN outlier detection, on different datasets perturbed by 3000 *shared images*.

Dataset	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}1$	*ARI*	*Purity*	*FPR*	$\mathcal{N}_C$	$\mathcal{N}_O$	$\mathcal{N}_U$
$\mathcal{V}^{\mathsf{NA}}$	0.996	0.754	0.858	0.854	0.996	0.000	37/35	2836/3000	595/10480
$\mathcal{V}^{\mathsf{W}}$	0.907	0.613	0.732	0.725	0.970	0.001	30/35	2686/3000	783/10480
$\mathcal{V}^{\mathsf{FH}}$	0.981	0.601	0.738	0.732	0.989	0.000	30/35	2834/3000	407/10480
$\mathcal{V}^{\mathsf{FL}}$	0.796	0.301	0.433	0.423	0.876	0.002	14/35	2608/3000	569/10480

has been mentioned that the number of outlier samples is much less than the number of other samples. However, in user profile image analysis, it cannot always be true as users may share more single, cropped or filtered images than those applicable to camera fingerprinting. So, in this challenging case, stability of the clustering algorithm is important. To test the stability, we increase the number of *shared images* gradually and perform clustering. We apply the datasets $\mathcal{V}_1^{\mathsf{NA}}$, $\mathcal{V}_1^{\mathsf{W}}$, $\mathcal{V}_1^{\mathsf{FH}}$ and $\mathcal{V}_1^{\mathsf{FL}}$. The datasets are perturbed by images from the Web to simulate the *shared images*. The

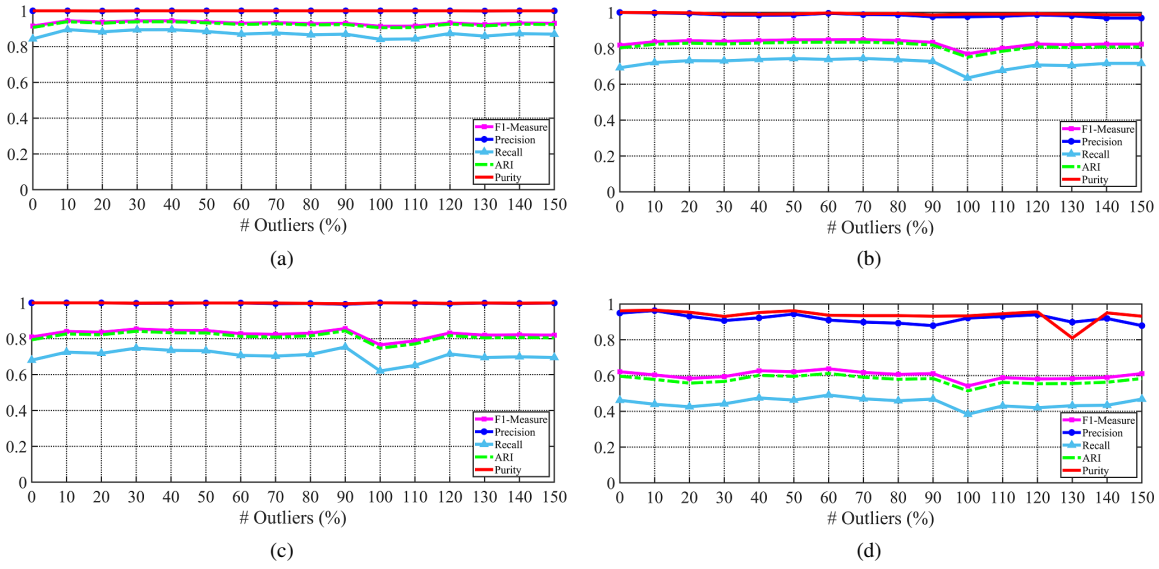

Figure 3: The hierarchical graph based clustering method is robust against the number of *shared images* in different datasets.: (a) $\mathcal{V}_1^{\mathsf{NA}}$, (b) $\mathcal{V}_1^{\mathsf{W}}$, (c) $\mathcal{V}_1^{\mathsf{FH}}$ and (d) $\mathcal{V}_1^{\mathsf{FL}}$.

Table 3
Results (%) of the hierarchical graph based clustering method based on DKNN outlier detection, on different datasets perturbed by 3000 *shared images*.

Dataset	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}1$	*ARI*	*Purity*	*FPR*	$\mathcal{N}_C$	$\mathcal{N}_O$	$\mathcal{N}_U$
$\mathcal{V}^{\mathsf{NA}}$	0.941	0.638	0.760	0.755	0.984	0.001	29/35	3000/3000	434/10480
$\mathcal{V}^{\mathsf{W}}$	0.885	0.575	0.697	0.690	0.949	0.002	27/35	3000/3000	745/10480
$\mathcal{V}^{\mathsf{FH}}$	0.994	0.467	0.635	0.628	0.992	0.000	26/35	3000/3000	913/10480
$\mathcal{V}^{\mathsf{FL}}$	0.700	0.281	0.382	0.375	0.790	0.005	12/35	3000/3000	1020/10480

Table 4
Results (%) of running time (seconds) of the proposed method for different datasets.

Dataset	I/O	Outlier removal	Correlation	Clustering	Total
$\mathcal{V}^{\mathsf{NA}}$	7078	26	351	577	8032
$\mathcal{V}^{\mathsf{W}}$	4512	30	397	658	5597
$\mathcal{V}^{\mathsf{FH}}$	7548	25	402	803	8778
$\mathcal{V}^{\mathsf{FL}}$	6837	23	402	732	7994

shared images are increasingly added with the order 10%, 20%, 30%, ..., 150% of the 2250 images in the datasets. The results of the clustering on the perturbed datasets are shown in Figure 3. The clustering is not affected, meaning that it is stable against the *shared images* even for the increase of 150% of the *taken images* by user's smartphones. The fine fluctuations in the graphs are related to the random selection of residual noises to fill in each batch in the batch partitioning step of the proposed method, see Figure 1.

Table 4 depicts the running time of the implementation of the proposed method on different datasets in terms of I/O, outlier removal, correlation computation and clustering phases, separately.

5. Discussion

Clustering the uploaded images by users on SN platforms is a challenging task as users may upload images from different sources. In this paper, we have categorized the uploaded images into two groups of *taken images* and *shared images*. By *taken images*, we mean images that contribute to fingerprinting their camera sources, while by *shared images*, we mean the images that are not directly referable to the user's device and do not present sufficient characteristics of sources and are not directly referable to the user's device. Besides, in the clustering of the images from a variety of smartphone models and brands, it is difficult to achieve the best values for all the mentioned measures in Section 4.1. For example, merging the residual noises of different cameras into the same cluster increases FPR, which can propagate the error to the following iterations in the clustering. As a result, $\mathcal{P}$ and $Purity$ decrease, although $\mathcal{R}$ increases (Lin and Li, 2017). We aimed to have the clusters with high values of $\mathcal{P}$, $Purity$, a low value of FPR, and an accurate value of $\mathcal{N}_C$, $\mathcal{N}_U$ and $\mathcal{N}_O$, since for this type of investigation, it is usually preferred to have the most accurate number of clusters with high values of precision and purity. Table 2 shows a recall of 0.754 for the native dataset compared to 0.301 for the FL dataset, meaning that the higher quality of the images we have, the better results of the clustering we get. Whereas the results from $\mathcal{V}^{\mathrm{W}}$ and $\mathcal{V}^{\mathrm{FH}}$ prove that the method is also robust regarding the process of image compression applied by SN platforms, that degrades the quality of the SPN. While, most of the running time of the proposed method is spent for I/O, for loading residual noises into RAM, see Table 4. This implies that the more capacity of RAM is provided, the faster implementation of the clustering is resulted. In this paper we presented an accurate analysis on how the *shared images* influence the clustering of the *taken images* and fingerprinting the smartphones of users in such a way that the number of *shared images* exceeds the number of *taken images*, while to the best of our knowledge, has not been before investigated by the state-of-the-art woks. Through the proposed clustering method, it is not needed to compute all the elements of full-pairwise correlation matrix. That is because of the Markov clustering that introduces candidate clusters for a merging. In doing so, only the correlation of the more similar residual noises are computed. Also, the adaptive threshold which is computed for the merging of the candidate clusters prevents from merging the clusters including images from the identical models of smartphones resulting in high values of $\mathcal{P}$, $Purity$ and accurate values of $\mathcal{N}_C$, see Table 2.

6. Conclusions

Clustering the images uploaded by users on their profiles is a way of fingerprinting the camera sources and it is considered a challenging task since users may upload different types of images, i.e., the images taken by their smartphones (*taken images*) and a variety images like single images from different sources, cropped images, or images from the Web (*shared images*). In this paper, we propose a user profiles' image clustering method without prior knowledge about the type and number of the camera sources. The *shared images* make a perturbation in the clustering task, so they have to be detected and removed from the clustering process. We have applied Density-Based Spatial Clustering of Applications with Noise (DBSCAN) technique to remove the *shared images*. The proposed method exploits hierarchical and graph based clustering algorithms, and an adaptive threshold to cluster the images. Through the clustering, Markov clustering introduces representative clusters, with a higher probability of coming from the same camera, for merging. This accelerates the clustering as there is no need to compute the full pairwise correlation matrix. The adaptive threshold for merging the representative clusters is updated during the hierarchical algorithm that prevents merging the images from the identical models of smartphones. The proposed method is scalable and applicable to large scale datasets as it partitions datasets into batches. Experimental results confirm the robustness of the method against perturbed datasets and its effectiveness in the image clustering. It has been shown that the method is stable against the *shared images* even for the increase of 150% of the *taken images*.

References

Alles, E.J., Geradts, Z.J., Veenman, C.J., 2009. Source camera identification for heavily jpeg compressed low resolution still images. Journal of forensic sciences 54, 628–638.

Amerini, I., Caldelli, R., Crescenzi, P., Del Mastio, A., Marino, A., 2014. Blind image clustering based on the normalized cuts criterion for camera identification. Signal Processing: Image Communication 29, 831–843.

Bayram, S., Sencar, H., Memon, N., Avcibas, I., 2005. Source camera identification based on cfa interpolation, in: Image Processing, 2005. ICIP 2005. IEEE International Conference on, IEEE. pp. III–69.

Bertini, F., Sharma, R., Iannì, A., Montesi, D., 2015. Smartphone verification and user profiles linking across social networks by camera fingerprinting, in: International Conference on Digital Forensics and Cyber Crime, Springer. pp. 176–186.

Bloy, G.J., 2008. Blind camera fingerprinting and image clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 532–534.

Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J., 2000. Lof: identifying density-based local outliers, in: ACM sigmod record, ACM. pp. 93–104.

Bustamam, A., Burrage, K., Hamilton, N.A., 2012. Fast parallel markov clustering in bioinformatics using massively parallel computing on gpu with cuda and ellpack-r sparse format. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 9, 679–692.

Caldelli, R., Amerini, I., Picchioni, F., Innocenti, M., 2010. Fast image clustering of unknown source images, in: 2010 IEEE International Workshop on Information Forensics and Security, IEEE. pp. 1–5.

Çeliktutan, O., Sankur, B., Avcibas, I., 2008. Blind identification of source cell-phone model. IEEE Trans. Information Forensics and Security 3, 553–566.

Charikar, M., Khuller, S., Mount, D.M., Narasimhan, G., 2001. Algorithms for facility location problems with outliers, in: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics. pp. 642–651.

Chawla, S., Gionis, A., 2013. k-means–: A unified approach to clustering and outlier detection, in: Proceedings of the 2013 SIAM International Conference on Data Mining, SIAM. pp. 189–197.

Chen, K., 2008. A constant factor approximation algorithm for k-median clustering with outliers, in: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms, Citeseer. pp. 826–835.

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.

Chierchia, G., Parrilli, S., Poggi, G., Sansone, C., Verdoliva, L., 2010. On the influence of denoising in prnu based forgery detection, in: Proceedings of the 2nd ACM workshop on Multimedia in Forensics, Security and Intelligence, ACM. pp. 117–122.

Choi, K.S., Lam, E.Y., Wong, K.K., 2006. Source camera identification using footprints from lens aberration. Digital photography II 6069, 172–179.

Cox, I.J., Miller, M.L., Bloom, J.A., Honsinger, C., 2002. Digital watermarking. volume 53. Springer.

Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K., 2007. Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on Image Processing 16, 2080–2095.

Ertöz, L., Steinbach, M., Kumar, V., 2003. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data, in: Proceedings of the 2003 SIAM International Conference on Data Mining, SIAM. pp. 47–58.

Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise., in: Kdd, pp. 226–231.

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A.Y., Foufou, S., Bouras, A., 2014. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. IEEE transactions on emerging topics in computing 2, 267–279.

Filler, T., Fridrich, J., Goljan, M., 2008. Using sensor pattern noise for camera model identification, in: 2008 15th IEEE International Conference on Image Processing, IEEE. pp. 1296–1299.

Geradts, Z.J., Bijhold, J., Kieft, M., Kurosawa, K., Kuroki, K., Saitoh, N., 2001. Methods for identification of images acquired with digital cameras, in: Enabling technologies for law enforcement and security, International Society for Optics and Photonics. pp. 505–512.

Gisolf, F., Barens, P., Snel, E., Malgoezar, A., Vos, M., Mieremet, A., Geradts, Z., 2014. Common source identification of images in large databases. Forensic science international 244, 222–230.

Goljan, M., Fridrich, J., Filler, T., 2009. Large scale test of sensor fingerprint camera identification, in: Media forensics and security, International Society for Optics and Photonics. p. 72540I.

Grubbs, F.E., 1969. Procedures for detecting outlying observations in samples. Technometrics 11, 1–21.

Haque, A., 2019. Twitch plays pokemon, machine learns twitch: unsupervised context-aware anomaly detection for identifying trolls in streaming data. arXiv preprint arXiv:1902.06208 .

He, Z., Xu, X., Huang, J.Z., Deng, S., 2005. Fp-outlier: Frequent pattern based outlier detection. Comput. Sci. Inf. Syst. 2, 103–118.

Huang, N., He, J., Zhu, N., Xuan, X., Liu, G., Chang, C., 2018. Identification of the source camera of images based on convolutional neural network. Digital Investigation 26, 72–80.

Hubert, L., Arabie, P., 1985. Comparing partitions. Journal of Classification 2, 193–218.

Kannan, R., Woo, H., Aggarwal, C.C., Park, H., 2017. Outlier detection for text data, in: Proceedings of the 2017 SIAM International Conference on Data Mining, SIAM. pp. 489–497.

Kharrazi, M., Sencar, H., Memon, N., 2005. Blind source camera identification, in: IEEE International Conference on Image Processing, pp. 69–72.

Kurosawa, K., Kuroki, K., Saitoh, N., 1999. Ccd fingerprint method-identification of a video camera from videotaped images, in: Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348), IEEE. pp. 537–540.

Lee, Y.J., Yeh, Y.R., Wang, Y.C.F., 2012. Anomaly detection via online oversampling principal component analysis. IEEE transactions on knowledge and data engineering 25, 1460–1470.

Li, C.T., 2010a. Source camera identification using enhanced sensor pattern noise. IEEE Transactions on Information Forensics and Security 5, 280–287.

Li, C.T., 2010b. Unsupervised classification of digital images using enhanced sensor pattern noise, in: Proceedings of 2010 IEEE International Symposium on Circuits and Systems, IEEE. pp. 3429–3432.

Li, C.T., Lin, X., 2017. A fast source-oriented image clustering method for digital forensics. EURASIP Journal on Image and Video Processing 2017, 69.

Li, D., Chen, D., Goh, J., Ng, S.k., 2018. Anomaly detection with generative adversarial networks for multivariate time series. arXiv preprint arXiv:1809.04758 .

Lin, X., Li, C.T., 2016. Preprocessing reference sensor pattern noise via spectrum equalization. IEEE Transactions on Information Forensics and Security 11, 126–140.

Lin, X., Li, C.T., 2017. Large-scale image clustering based on camera fingerprints. IEEE Transactions on Information Forensics and Security 12, 793–808.

Liu, B.b., Lee, H.K., Hu, Y., Choi, C.H., 2010. On classification of source cameras: A graph based approach, in: 2010 IEEE International Workshop

on Information Forensics and Security, IEEE. pp. 1–5.
Liu, F.T., Ting, K.M., Zhou, Z.H., 2008. Isolation forest, in: 2008 Eighth IEEE International Conference on Data Mining, IEEE. pp. 413–422.
Liu, H., Li, J., Wu, Y., Fu, Y., 2018. Clustering with outlier removal. arXiv preprint arXiv:1801.01899 .
Liu, Q., Li, X., Chen, L., Cho, H., Cooper, P., Chen, Z., Qiao, M., Sung, A., 2012. Identification of smartphone-image source and manipulation. Advanced Research in Applied Artificial Intelligence , 262–271.
Lukas, J., Fridrich, J., Goljan, M., 2006. Digital camera identification from sensor pattern noise. IEEE Transactions on Information Forensics and Security 1, 205–214.
Marra, F., Poggi, G., Sansone, C., Verdoliva, L., 2016. Correlation clustering for prnu-based blind image source identification, in: 2016 IEEE International Workshop on Information Forensics and Security (WIFS), IEEE. pp. 1–6.
Marra, F., Poggi, G., Sansone, C., Verdoliva, L., 2017. Blind prnu-based image clustering for source identification. IEEE Transactions on Information Forensics and Security 12, 2197–2211.
Mesa, U., 2012. Regularized markov clustering algorithm on protein-protein interaction networks using ellpack-r sparse data format. Undergraduated thesis, University of Indonesia .
Norouzizadeh Dezfouli, F., Dehghantanha, A., Eterovic-Soric, B., Choo, K.K.R., 2016. Investigating social networking applications on smartphones detecting facebook, twitter, linkedin and google+ artefacts on android and ios platforms. Australian Journal of Forensic Sciences 48, 469–488.
Ott, L., Pang, L., Ramos, F.T., Chawla, S., 2014. On integrated clustering and outlier detection, in: Advances in neural information processing systems, pp. 1359–1367.
Phan, Q.T., Boato, G., De Natale, F.G., 2018. Accurate and scalable image clustering based on sparse representation of camera fingerprint. IEEE Transactions on Information Forensics and Security .
Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M., 2018. Deep one-class classification, in: International Conference on Machine Learning, pp. 4393–4402.
Satuluri, V.M., 2012. Scalable clustering of modern networks. Ph.D. thesis. The Ohio State University.
Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 888–905.
Shirkhorshidi, A.S., Aghabozorgi, S., Wah, T.Y., Herawan, T., 2014. Big data clustering: a review, in: International Conference on Computational Science and Its Applications, Springer. pp. 707–720.
Shullani, D., Fontani, M., Iuliani, M., Al Shaya, O., Piva, A., 2017. Vision: a video and image dataset for source identification. EURASIP Journal on Information Security 2017, 15.
Sorrell, M.J., 2009. Digital camera source identification through jpeg quantisation, in: Multimedia forensics and security. IGI Global, pp. 291–313.
Swaminathan, A., Wu, M., Liu, K.R., 2007. Nonintrusive component forensics of visual sensors using output images. IEEE Transactions on Information Forensics and Security 2, 91–106.
Taha, A., Hadi, A.S., 2019. Anomaly detection methods for categorical data: A review. ACM Computing Surveys (CSUR) 52, 38.
Tang, J., Chen, Z., Fu, A.W.C., Cheung, D.W., 2002. Enhancing effectiveness of outlier detections for low density patterns, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer. pp. 535–548.
Van, L.T., Emmanuel, S., Kankanhalli, M.S., 2007. Identifying source cell phone using chromatic aberration, in: 2007 IEEE International Conference on Multimedia and Expo, IEEE. pp. 883–886.
Van Dongen, S., 2008. Graph clustering via a discrete uncoupling process. SIAM Journal on Matrix Analysis and Applications 30, 121–141.
Varia, S., 2013. Regularized Markov Clustering in MPI and Map Reduce. Ph.D. thesis. The Ohio State University.
Villalba, L.J.G., Orozco, A.L.S., López, R.R., Castro, J.H., 2016. Identification of smartphone brand and model via forensic video analysis. Expert Systems with Applications 55, 59–69.
Wu, G., Kang, X., Liu, K.R., 2012. A context adaptive predictor of sensor pattern noise for camera source identification, in: 2012 19th IEEE International Conference on Image Processing, IEEE. pp. 237–240.
Zhang, K., Hutter, M., Jin, H., 2009. A new local distance-based outlier detection approach for scattered real-world data, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer. pp. 813–822.