IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 29, 2019, accepted June 3, 2019, date of publication July 2, 2019, date of current version July 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2925102

Hybrid Clustering of Shared Images on Social Networks for Digital Forensics

RAHIMEH ROUHI[1], FLAVIO BERTINI[1], DANILO MONTESI[1], XUFENG LIN[2], YIJUN QUAN[3], AND CHANG-TSUN LI[2,3], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, University of Bologna, 40126 Bologna, Italy
[2]School of Information Technology, Deakin University, Geelong Waurn Ponds Campus, Waurn Ponds, VIC 3216, Australia
[3]Department of Computer Science, University of Warwick, Coventry CV4 7AL, U.K.

Corresponding author: Rahimeh Rouhi (rahimeh.rouhi2@unibo.it)

This work was supported in part by the H2020 Framework Project OPERANDUM under Grant 776848.

ABSTRACT Clustering the images shared through social network (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 the camera sensor imperfections during the manufacturing process can be extracted from the images and used to fingerprint the smartphones. The process of content compression performed by the SNs causes loss of image details and weakens the SPN, making the clustering task even more challenging. In this paper, we present a hybrid algorithm capable of clustering the images captured and shared through SNs without prior knowledge about the types and number of the acquisition smartphones. The hybrid method exploits batch partitioning, image resizing, hierarchical and graph-based clustering approaches to cluster the images. Using Markov clustering, the hierarchical clustering is conducted in such a way that the representative clusters with a higher probability of belonging to the same camera are selected for merging, which accelerates the clustering. For merging the clusters, the adaptive threshold updated iteratively through the hybrid clustering is used, which results in more precise clusters even for images from the same model of smartphones. The results on the VISION dataset, including both *native* and *shared images*, prove the effectiveness and efficiency of the hybrid method in comparison with the state-of-the-art SPN-based image clustering algorithms.

INDEX TERMS Sensor pattern noise, camera fingerprinting, image clustering, social networks, digital forensics.

I. INTRODUCTION

The ever increasing prevalence of smartphones and the popularity of social network (SN) platforms have facilitated instant sharing of multimedia content through SNs [1]. However, the ease in taking and sharing photos and videos through SNs also allows privacy-intrusive and illegal content to be widely distributed [2]. As such, images captured and shared by users on their profiles are considered as significant digital evidence providing investigators with important clues once a digital crime is reported. Generally, there are different scenarios in online digital forensics. Given a set of confiscated smartphones and a set of suspect profiles on SNs, the investigators may want to know whether the images shared on suspect profiles originated from the confiscated smartphones or not. It is usually called *smartphone verification*. More specifically, the verification task is a binary classification that carries out 1-by-1 matching between the pairs of the images and the smartphones. The second scenario is similar to the first, but instead of one, multiple cameras are available. It is known as *smartphone identification*, which deals with 1-to-m matching problem and determines which smartphone out of *m* took a given image [3]. However, in most real-life cases, the investigators collect a large number of shared images on suspect profiles, without any clue of the cameras, and they still intend to group the images into an unknown number of clusters, each of them includes the images taken by the same camera. As an important outcome, by the resulted clusters and the existing profile tags, the profiles sharing the images captured by the same camera source are linked, resulting in more clues to detect evidence references in a digital crime.

The associate editor coordinating the review of this manuscript and approving it for publication was Irene Amerini.

The Sensor Pattern Noise (SPN), due to camera sensor imperfections created during the manufacturing process, is considered as a unique characteristic to fingerprint a source camera. Given a set of images taken by a specific smartphone camera, the SPN can be approximated by averaging the Residual Noises (RNs) presented in the images [4]. Each RN is the difference between the image content and its denoised version acquired by a denoising filter. The clustering is typically performed based on the similarities among the extracted RNs.

A. PROBLEM STATEMENT

For real-life applications dealing with a large number of images, a clustering algorithm needs to be precise, scalable and feasible. In other words, it is desired to cluster the images into the right groups, to be applicable on any given dataset, and have a reasonable running time. Developing an algorithm meeting all these requirements has some challenges as follows:

- To extract the right SPN, the correct orientation of images has to be obtained. Some smartphone settings and SN platforms remove the orientation information from metadata of the image file.
- The extracted RN from an image can be severely contaminated by other interference. Besides, the process of content compression performed by SNs causes loss of image details and weakens the SPN.
- To compute the similarity among the RNs, they have to have the same spatial resolution. The typical way is to crop the central block of RNs which may cause loss of critical data.
- Though it is desirable to apply high resolutions of RNs, for having better quality of clustering, the heavy overhead on data storage and computation limits their usage.
- Images captured by different smartphones of the same model undergo the same imaging pipeline which introduces similar artifacts in the SPNs.
- Calculating the full-pairwise correlation matrix is a cumbersome and sometimes infeasible task especially for large-scale datasets.

With these challenges in the SPN-based image clustering, many works have been done. However, the literature lacks a precise, scalable, and feasible algorithm designed particularly for clustering the *shared images* on SN platforms.

B. CONTRIBUTIONS

We present a hybrid clustering approach by combining the hierarchical and graph-based algorithms. The hybrid algorithm is capable of clustering the images captured and shared through SNs without prior knowledge about the types and number of the acquisition smartphones, and it tackles most of the mentioned challenges. This makes it applicable to the real-life cases:

- Unlike most studies presented in the literature, to get better characteristics of SPN, we exploit resizing rather than cropping the RNs, which is more useful in the clustering, especially for *shared images* having low resolutions.
- To tackle the problem of the limited size of RAM for loading the RNs, the dataset is partitioned into small batches to ensure that each of them can be fit into the available RAM.
- By using Markov clustering, the hierarchical clustering is conducted in such a way that the representative clusters with a higher probability of belonging to the same camera are selected for a merging.
- To merge the candidate clusters, an adaptive threshold is used. The threshold generally increases as the size and quality of a cluster increase. This prevents wrong merging of clusters, especially the clusters from the same models of smartphones.
- Partitioning the dataset, exploiting the inherent sparseness of correlation matrix, and checking only the representative clusters in the merging result in the calculation of a small portion of the full-pairwise correlation matrix. This accelerates the clustering, which is particularly helpful for large-scale datasets.

The rest of the paper is organized as follows. In Section II, a summary of existing works on SPN-based image clustering is provided. The proposed hybrid algorithm is explained in Section III. In Section IV, The results of the hybrid algorithm as well as the comparisons with other state-of-the-art algorithms are presented. Finally, the conclusion is drawn in Section V.

II. RELATED WORKS

Many works have been done on SPN-based image clustering. As a pioneering work, Bloy in [5] presented an unsupervised clustering algorithm considering the RNs as singleton clusters and hierarchically merging the similar clusters. The pairs of the clusters are randomly selected and if their correlation exceeds a threshold, they are merged into a new cluster, which is represented by its centroid, i.e., the corresponding SPN. It 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 is not generalized well across various source cameras.

In [6], Markov random fields are used to assign iteratively a class label to an image, according to the consensus of a small set of SPNs, called membership committee. This raises an issue on how to choose an appropriate committee, particularly for the datasets with various cluster cardinalities, i.e., asymmetric datasets. The authors of [7] developed a faster algorithm by proposing a new enhancer applied to the extracted SPNs. A random subset, i.e., training set, of the entire dataset is used for clustering, which is followed by a classification step for the remaining fingerprints, i.e., test set. The algorithm merges the clusters hierarchically, and a silhouette coefficient is calculated for each cluster. Particularly, the silhouette coefficient estimates the separation among clusters as well as the cohesion within each cluster.

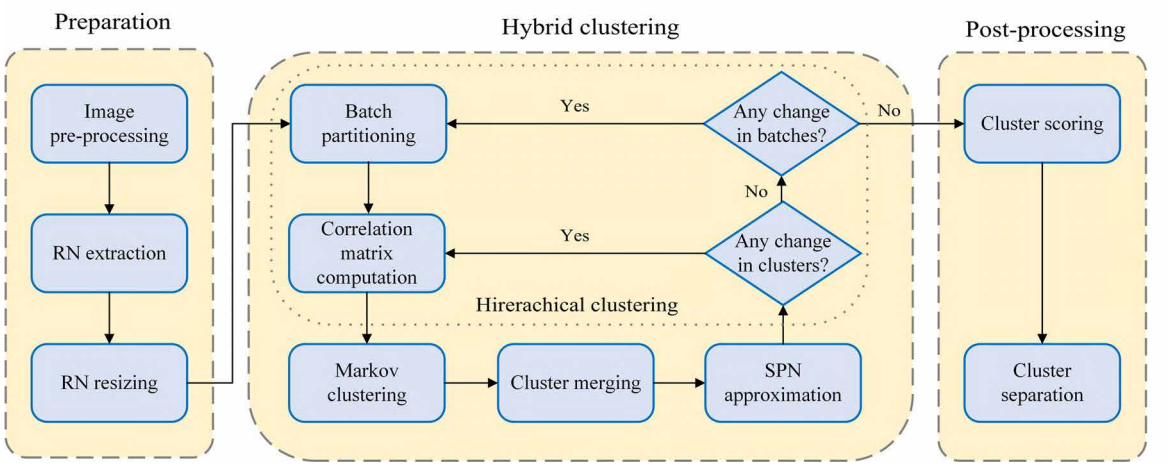


FIGURE 1. Flowchart of the proposed hybrid method.

The average of the silhouette coefficients corresponding to the produced clusters in each iteration is considered as a merit of the clustering, showing its quality. Once all fingerprints are merged into one cluster, a set of clusters with the highest value of the merit is selected as the optimal result of the clustering. In [8], a similar unsupervised 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 proposed 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 RNs because all the cluster pairs have to be checked for a merging.

The graph-based algorithms have successfully been applied to SPN-based image clustering by computing a small portion of the full-pairwise correlation matrix. In [9], an approach 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 RNs 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 depends on the random initialization. In [10], the problems were addressed by means of the normalized cut graph partitioning algorithm [11], which produced better clustering results without providing the number of clusters as an input parameter. However, the stopping condition is met once all the aggregation coefficients computed for the obtained clusters are greater than a pre-defined threshold, which is an important input parameter itself.

In [12], the clustering is performed based on correlation clustering formulating the graph partitioning problem as constrained energy minimization. A refinement step is applied to generate clusters with higher qualities. The disadvantage is that it needs a parameter set by the user according to preliminary analyses on an appropriate training set. The issue of parameter setting was handled in [13] 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. This makes the clustering more difficult. To tackle the issue, in [14], shared nearest neighbors [15] are applied to the full-pairwise correlation matrix, to find clusters with different sizes and densities. A common undesirable trait of the mentioned algorithms 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 [16], 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 approach. 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 [17] 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 [16], it tends to be too radical for clusters with large size, which is a critical point in real-life applications. However, these scalable clustering algorithms were only tested on *native images* and their robustness on *shared images* on SNs is still in doubt.

III. PROPOSED HYBRID METHOD

The proposed hybrid method mainly consists of preparation, hybrid clustering, and post-processing phases, as shown in Figure 1. We will look into each of the three phases to give an overview of the proposed method:

- **Preparation:** firstly, dark and saturated images are excluded. Then, the remaining images are aligned to the same orientation, depending on the availability of meta-data of the image files. Next, the grayscale version of the images is obtained. RNs are extracted from the pre-processed images, and they are resized to a specific resolution, i.e., 1024×1024 px.
- **Hybrid clustering:** the clustering starts with randomly partitioning the dataset into small batches. For each batch, a pairwise correlation matrix is calculated. Markov clustering algorithm is applied to the correlation matrix. By using the probability matrix, as the output of the Markov clustering, and nearest neighboring, the candidate clusters are selected, and subsequently an adaptive threshold is computed for merging the clusters. The SPN is updated for the merged clusters and similar process is hierarchically performed on the merged clusters. The clustering stops once no new cluster is found.
- **Post-processing:** in this phase, 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.

A. PREPARATION

Since dark and saturated images do not provide trustworthy RNs [18], including these images makes the clustering task unreliable and computationally cumbersome [16]. Therefore, we recognize these images and exclude them from our experiments. If 70% of the pixels in an image have the intensities smaller than 50 or greater than 250, we consider it as dark or saturated image, respectively. We align all the remaining images to the same orientation according to the Exchangeable Image File Format (EXIF) data. For some images without EXIF data, we only rotate the images to either portrait or landscape orientation based on the spatial resolution [19]. This may not completely resolve the orientation problem, but it can be alleviated. Any how, the clustering quality would be affected. Finally, to reduce the computational cost, we only use the grayscale version of each image.

We extract the RNs from the pre-processed images as follows:

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

where I and $d()$ are an image and a denoising filter, respectively. Subsequently, by averaging the RNs extracted from n images taken by a given smartphone, the SPN, i.e., the camera fingerprint, can be approximated by:

$$SPN = \frac{1}{n}\sum_{j=1}^{n} RN_j \tag{2}$$

Based on (1) and (2), it is obvious that the quality of the extracted RN and SPN depends on $d()$ and n. Block-Matching and 3D (BM3D) denoising filter introduced by [20] is applied to extract the RNs because it results in better quality of SPNs. Through BM3D, non-unique artifacts are removed by using zero-meaning all columns and rows, and Wiener filtering in the Fourier domain [18], [21] and [22].

After extracting the RNs from the full-size grayscale images, unlike many works, which cropped RNs (often from the center), we resize them to a specific resolution based on bicubic interpolation [23]. Indeed, resizing is a flexible way to calculate the correlations between the RNs with different resolutions. Assume that images in a dataset were captured by 2 smartphones and have different resolutions such as 2560×1920 and 960×720 px. To calculate correlations, all the RNs are typically cropped to the lowest resolution, i.e., 960×720 px in this case. Consequently, a large segment of RNs with the highest resolution, i.e., 2560×1920 px, is discarded. Conversely, by resizing, we can flexibly upscale the lowest resolution or downscale the highest resolution to a specific size for more efficient use of available information. More details about the impact of resizing versus cropping on the correlations and the clustering will be given in Section IV-C.1. Actually, zero-padding can be another option to handle the computation of correlations between RNs with different resolutions. It may give rise to other issues, e.g., memory usage, but it is worth further investigation.

B. HYBRID CLUSTERING

The proposed hybrid clustering groups the RNs based on the combination of hierarchical and Markov clustering algorithms and an adaptive threshold which is iteratively generated according to the quality of the resulted clusters. First, we explain batch partitioning and how the cluster similarities are computed. Then, we describe different steps in the hybrid clustering in the following subsections.

Due to the limited size of RAM, it is not applicable to load all the RNs into RAM at once. To tackle this issue and provide a scalable clustering algorithm, we follow the approach presented in [16]. Let N be the total number of RNs in the dataset. The pre-processed RNs are randomly partitioned into t batches, i.e., $B = \{b_1, b_2, \ldots, b_t\}$, where $t = \lceil \frac{N}{q} \rceil$ and q is the batch size. The parameter q is determined with respect to the available size of RAM. For each batch, a correlation matrix $\mathcal{A}$ is constructed, with each element $\mathcal{A}(i, j)$ being the Normalized Cross Correlation (NCC) similarity between any two SPNs, $f_i = [x_1, \ldots, x_l]$ and $f_j = [y_1, \ldots, y_l]$:

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

where $\bar{f}_i$ and $\bar{f}_j$ represent the means of the two SPN vectors, respectively.

1) HIERARCHICAL CLUSTERING

The clustering for each batch starts by considering each RN as a singleton cluster, and 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. Once a merging is done, this cannot be undone, and a wrong assignment may propagate the error to the following iterations in the clustering. Moreover, its computational burden is high because it has to investigate all the pairs of clusters for merging [24]. In our proposed clustering algorithm, we handle the mentioned drawbacks of the hierarchical clustering by combining it with the Markov clustering algorithm and a cluster merging step based on an adaptive threshold to achieve precise and fast clustering.

2) MARKOV CLUSTERING

Markov clustering is a fast and scalable unsupervised algorithm proposed by [25]. It has successfully been applied in different fields of science. It considers the objects as the vertices of a graph, e.g., G, and groups them with respect to the weights of the edges, i.e., the similarities between the objects [26]. The Markov matrix $\mathcal{M}$ associated with the graph G is defined by normalizing all columns of the similarity matrix. Then, a random walk is simulated over the vertices of the graph to increase and decrease the flow in strong and weak currents in G, respectively [27]. Starting from a vertex, a random walk is more likely to arrive at the vertices within the same cluster than those in different clusters. By applying expansion and inflation operators to $\mathcal{M}$ alternatively, with each element at index (i, j) being the transition probability from vertex i to vertex j, the clusters can be interpreted from the resulting transition matrix at the converged state. More specifically, the random walk can be modeled as a Markov chain on the graph G. The vertices of G are considered as a set of states $S = \{s_1, s_2, \ldots, s_n\}$, and the graph edges are associated with the transition probabilities represented in $\mathcal{M} = [\rho(i, j)] \in \mathbb{R}^{n\times n}$ with

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

The expansion operator performed based on matrix multiplication simulates a random walk on the graph G 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 a 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 rescaled to make the sum of each column equal to 1. This is called pruning 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 reduces the number of non-zero elements in the matrix $\mathcal{M}_{inf}$, and efficiently decreases the memory usage and accelerates the clustering [28]. A global chaos $\mathcal{G}$ shows the rate of the changes in the probability values in every two consecutive iterations. The algorithm stops when the global chaos approximately equals zero. The value of $\mathcal{G}$ is determined based on the maximum value of chaos denoted as $\mathcal{C}_j$ on every column j of the matrix $\mathcal{M}_{pru}$ [29].

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

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

The details of the Markov clustering as a part of our proposed hybrid algorithm 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)
  - $\mathcal{M} = \mathcal{M}_{pru}$
end
return $\mathcal{M}$

The Markov clustering receives the similarity matrix $\mathcal{A}$ computed by (3) as an input, and it produces the matrix $\mathcal{M}$, with each entry of the matrix representing the degree of the similarities between a pair of RNs. 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, ensuring at least one non-zero entry per column [28]. Given two clusters c_i and c_j corresponding to the vertices v_i and v_j in G, if they share the same or very similar fingerprint characteristics of a camera, the element $\mathcal{M}(i, j)$ is assigned to a non-zero value. Otherwise, it means that the clusters are from different cameras and $\mathcal{M}(i, j)$ is set to 0.

During the hybrid clustering, in every iteration of the hierarchical clustering, the Markov clustering is performed on

each batch, see Figure 1. By applying nearest neighboring to the columns of the resulted probability matrix, small cluster granularities can be obtained. We consider these representative and precise clusters as the candidate clusters, which are more likely to be from the same camera sources, and we merge them iteratively to discover larger clusters.

3) CLUSTER MERGING

Since the RNs were randomly partitioned into batches, the matrix $\mathcal{M}$ of a batch may contain many sparse columns, which are populated with many zero values. Hence, 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 RNs 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 hybrid clustering more precise, in the cluster merging, in addition to using the candidate clusters, we use an adaptive threshold.

The adaptive threshold is computed based on the quality of the obtained clusters, which is updated in each iteration of the algorithm. It exploits the idea that the more images from a given smartphone are precisely clustered, the better the quality of SPN can be estimated [5]. 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 [16]:

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

where τ is a minimum threshold working as a trust boundary of $\mathcal{T}$. The terms n_{c_i} and n_{c_j} denote the number of the RNs 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 RNs in the cluster. Given two candidate clusters c_i and c_j, if the correlation calculated by (3) between their corresponding fingerprints f_i and f_j is greater than the adaptive threshold, i.e., $\mathcal{A}(f_i, f_j) > \mathcal{T}$, the clusters are merged. Otherwise, they are kept separately and passed to the next iteration of the algorithm.

Algorithm 2 Proposed Hybrid Clustering Algorithm

input : pre-processed RNs
output: list of clusters, C
- 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, $C_{new} = \{c_1, c_2, \ldots, c_N\}$
- initializing a set of camera fingerprints with the RNs corresponding to the clusters, $F = \{f_1, f_2, \ldots, f_N\}$
- partitioning initialization, $B_{old} = \{\}$
- $t = \lceil \frac{N}{q} \rceil$
- randomly partition C_{new} into t batches with size q, $B_{new} = \{b_1, b_2, \ldots, b_t\}$
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)
      - apply Markov clustering to $\mathcal{A}$ and generate the probability matrix $\mathcal{M}$ by Algorithm 1
      - put non-sparse column's indices in the list $\mathcal{L}$
      for $i = 1 : |\mathcal{L}|$ **do**
        - find the nearest cluster c_j to the cluster c_i from the list $\mathcal{L}$
        - 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
        end
      end
      - put the obtained clusters in C_{new}
      - update the camera fingerprints in F for the merged clusters by (2)
      - $C_{old} = C_{new}$
    end
  end
  - 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}
end
- $C = C_{new}$
return C

4) COMPUTATIONAL COMPLEXITY

Given the explanations above, we present the pseudo code of the hybrid clustering in Algorithm 2. To compute the time complexity of the hybrid algorithm, we first compute the complexity for one batch and then generalize it over all the t batches. For each batch, different steps such as correlation

matrix computation, Markov clustering, finding non-sparse columns and nearest neighboring for cluster merging are applied. The correlation matrix computation for each batch with the size of q has the complexity $O(q^2)$. Then, Markov clustering is applied to the correlation matrix $\mathcal{A}$ with the complexity $O(qz^2)$, where z is the maximum number of non-zero elements in each column in $\mathcal{A}$. Finding non-sparse columns in $\mathcal{M}$, has the complexity $O(q)$. In addition, selecting the nearest neighbors of non-sparse columns in $\mathcal{M}$ requires the complexity $O(w^2 \log w)$, such that w is the number of non-sparse columns in $\mathcal{M}$. Totally, the complexity of the clustering for each batch is $O([q^2+qz^2+q+w^2 \log w])$. The agglomerative hierarchical clustering has the cost $O(C'^2 \log C')$ where C' is the number of the obtained clusters after the first iteration. Eventually, the total time complexity of clustering t batches is approximated as $O(t[q^2+qz^2+q+w^2 \log w]+C'^2 \log C') \approx O(q^2)$. With respect to q, the complexity of the hybrid algorithm is almost quadratic which is promising for SPN-based image clustering.

C. POST-PROCESSING

As the final phase, post-processing is applied to the result of clustering, both coarse and fine clusters. The aim is to preserve the coarse clusters, including a notable number of RNs sharing the same camera fingerprints characteristics. In contrast, fine clusters include few RNs which do not share sufficient similarities with the obtained camera fingerprints during the clustering. Due to the nature of the noise-like camera fingerprints [16], and especially the low resolution of the *shared 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 in the application of digital investigations, we remove the fine clusters. 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 c_i is the i^{th} cluster in the resulted set of clusters C from the hybrid clustering. 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.

IV. EXPERIMENTS

A. DATASET

In the experiments, we use the VISION dataset [19], composed of images in both *native* format and *shared* version. More specifically, the dataset contains *flat* and *generic* images taken by 35 smartphones. The former is a set of images of walls and skies, while the latter is a set of images without limitations on orientation or scenario. The images were shared through *WhatsApp* and *Facebook* (in both high and low resolutions). We use only *generic* images in our experiments. To see the functionality of the proposed algorithm on the images from the same or completely different models of smartphones, we divide the entire dataset $\mathcal{D}$ into two subsets $\mathcal{D}_1$ and $\mathcal{D}_2$. After excluding dark and saturated images, $\mathcal{D}_1$ includes 2250 images from 11 smartphones with 5 different models, while $\mathcal{D}_2$ covers 5230 images from 24 smartphones with completely different models. The characteristics of the applied smartphones and their corresponding number of images in our experiments are listed in Tables 1 and 2. For each dataset $\mathcal{D}_i$, both types of images *native* and *shared* are considered, so we have $\mathcal{D}_i^{\mathrm{N}}$, $\mathcal{D}_i^{\mathrm{W}}$, $\mathcal{D}_i^{\mathrm{FH}}$ and $\mathcal{D}_i^{\mathrm{FL}}$ corresponding to *Native*, *WhatsApp*, *Facebook high-resolution* and *Facebook low-resolution* images, respectively. Finally, to test the generalization of the proposed algorithm, we run it on the whole dataset, i.e., $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$. In particular, we have $\mathcal{D}^{\mathrm{N}} = \mathcal{D}_1^{\mathrm{N}} \cup \mathcal{D}_2^{\mathrm{N}}$, $\mathcal{D}^{\mathrm{W}} = \mathcal{D}_1^{\mathrm{W}} \cup \mathcal{D}_2^{\mathrm{W}}$, $\mathcal{D}^{\mathrm{FH}} = \mathcal{D}_1^{\mathrm{FH}} \cup \mathcal{D}_2^{\mathrm{FH}}$, and $\mathcal{D}^{\mathrm{FL}} = \mathcal{D}_1^{\mathrm{FL}} \cup \mathcal{D}_2^{\mathrm{FL}}$, each of them

TABLE 1. Smartphone's characteristics in Dataset $\mathcal{D}_1$.

ID	Brand	Model	Original resolution	#images
S_1	Apple	iPhone 4S	3264×2448	178
S_2	Apple	iPhone 4S	3264×2448	200
S_3	Apple	iPhone 5	3264×2448	203
S_4	Apple	iPhone 5	3264×2448	223
S_5	Apple	iPhone 5c	3264×2448	201
S_6	Apple	iPhone 5c	3264×2448	206
S_7	Apple	iPhone 5c	3264×2448	333
S_8	Apple	iPhone 6	3264×2448	129
S_9	Apple	iPhone 6	3264×2448	227
S_{10}	Samsung	Galaxy S III Mini GT-I8190	2560×1920	150
S_{11}	Samsung	Galaxy S III Mini GT-I8190N	2560×1920	200

TABLE 2. Smartphone's characteristics in Dataset $\mathcal{D}_2$.

ID	Brand	Model	Original resolution	#images
S_1	Apple	iPad2	960×720	170
S_2	Apple	iPad mini G	2592×1936	157
S_3	Apple	iPhone 4	2592×1936	217
S_4	Apple	iPhone 6 Plus	3264×2448	256
S_5	Asus	Zenfone	3264×1836	208
S_6	Huawei	Ascend G6-U10	3264×2448	153
S_7	Huawei	Honor 5C	4160×3120	271
S_8	Huawei	P8 GRA-L09	4160×2336	265
S_9	Huawei	P9 EVA-L09	3968×2976	237
S_{10}	Huawei	P9 Lite VNS-L31	4160×3120	234
S_{11}	Lenovo	P70-A	4784×2704	216
S_{12}	LG	D290	3264×2448	224
S_{13}	Microsoft	Lumia 640 LTE	3264×2448	180
S_{14}	OnePlus	A3000	4640×3480	284
S_{15}	OnePlus	A3003	4640×3480	236
S_{16}	Samsung	Galaxy S3 GT-I9300	3264×2448	207
S_{17}	Samsung	Galaxy S4 Mini GT-I9195	3264×1836	208
S_{18}	Samsung	Galaxy S5 SM-G900F	5312×2988	254
S_{19}	Samsung	Galaxy Tab 3 GT-P5210	2048×1536	166
S_{20}	Samsung	Galaxy Tab A SM-T555	2592×1944	154
S_{21}	Samsung	Galaxy Trend Plus GT-S7580	2560×1920	163
S_{22}	Sony	Xperia Z1 Compact D5503	5248×3936	216
S_{23}	Wiko	Ridge 4G	3264×2448	249
S_{24}	Xiaomi	Redmi Note 3	4608×2592	305

TABLE 3. Lowest and highest image resolution in different datasets.

Dataset	Lowest resolution	Highest resolution
$\mathcal{D}_1^{\mathrm{N}}$	2560×1920	3264×2448
$\mathcal{D}_1^{\mathrm{W}}$	1280×960	1280×960
$\mathcal{D}_1^{\mathrm{FH}}$	2048×1536	2048×1536
$\mathcal{D}_1^{\mathrm{FL}}$	960×720	1224×918
$\mathcal{D}_2^{\mathrm{N}}$	960×720	4608×2592
$\mathcal{D}_2^{\mathrm{W}}$	960×720	2048×1536
$\mathcal{D}_2^{\mathrm{FH}}$	960×720	1280×960
$\mathcal{D}_2^{\mathrm{FL}}$	1040×584	1312×984
$\mathcal{D}^{\mathrm{N}}$	960×720	5248×3936
$\mathcal{D}^{\mathrm{W}}$	960×720	1280×960
$\mathcal{D}^{\mathrm{FH}}$	960×720	2048×1536
$\mathcal{D}^{\mathrm{FL}}$	1040×584	1312×984

includes 7480 images. In Table 3, the lowest and the highest resolutions of the images for each dataset are listed.

To empirically set a specific resolution for resizing the RNs and the required parameters for the clustering algorithm, we consider the sample datasets $\mathcal{D}_0^{\mathrm{N}} \subseteq \mathcal{D}^{\mathrm{N}}$, $\mathcal{D}_0^{\mathrm{W}} \subseteq \mathcal{D}^{\mathrm{W}}$, $\mathcal{D}_0^{\mathrm{FH}} \subseteq \mathcal{D}^{\mathrm{FH}}$ and $\mathcal{D}_0^{\mathrm{FL}} \subseteq \mathcal{D}^{\mathrm{FL}}$. From each of 35 smartphones, 100 images are randomly selected, so each dataset includes 3500 images. The sample datasets make the setting facilitative and they still include images from a variety of smartphone models and brands. Hence, the set values for each dataset can be generalized to the entire datasets.

B. EXPERIMENTAL MEASURES

The quality of the hybrid clustering algorithm is evaluated by different measures. There are four definitions based on the agreement between two sets of labels, i.e., ground truth or target clusters $T = \{t_1, t_2, \ldots, t_g\}$ and the resulted clusters $C = \{c_1, c_2, \ldots, c_h\}$, where g and h are the number of the target and the resulted clusters, respectively. Given two samples d_i and d_j in a dataset, e.g., $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$, we have the following definitions:

i. True Positive: $TP = |\{(d_i, d_j) : t_i = t_j \text{ and } c_i = c_j\}$ meaning that the two samples d_i and d_j belong to the same cluster in T, and they are also in the same output cluster in C.
ii. False Negative: $FN = |\{(d_i, d_j) : t_i = t_j \text{ and } c_i \neq c_j\}$ meaning that the two samples d_i and d_j belong to the same cluster in T, while they are not in the same output cluster in C.
iii. False Positive: $FP = |\{(d_i, d_j) : t_i \neq t_j \text{ and } c_i = c_j\}$ meaning that the two samples d_i and d_j do not belong to the same cluster in T, but they are in the same output cluster in C.
iv. True Negative: $TN = |\{(d_i, d_j) : t_i \neq t_j \text{ and } c_i \neq c_j\}$ meaning that the two samples d_i and d_j do not belong to the same cluster in T, and they are also not in the same output cluster in C.

Having the above definitions, *Precision rate* $\mathcal{P}$, *Recall rate* $\mathcal{R}$ also known as *True Positive Rate* (*TPR*), *F1-measure* $\mathcal{F}$, *Rand Index* (*RI*), *Adjusted Rand Index* (*ARI*), *Purity*, *False Positive Rate* (*FPR*), and *ratio of the cluster numbers* $\mathcal{N}$ are depicted by (12)-(19):

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

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

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

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

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 $\overline{RI}$ is a non-zero value. To get rid of this bias, *ARI* was proposed by [30]:

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

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 clusters, $\widehat{c_i}$ denotes the number of RNs with the dominant cluster label in the cluster c_i, and $|c_i|$ is the total number of RNs in c_i.

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

In addition, the ratio of the number of the obtained clusters that is n_o over the number of ground truth clusters denoted by n_g is calculated:

$$\mathcal{N} = \frac{n_o}{n_g} \tag{19}$$

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 RNs 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 [16]. We prefer to have the clusters with high values of $\mathcal{P}$, *Purity*, a low value of *FPR*, and an accurate value of $\mathcal{N}$. However, we evaluate the proposed algorithm by all the mentioned measures in (12)-(19) to have a reasonable comparison with the other clustering algorithms.

C. EXPERIMENTAL RESULTS

In this section, firstly we explain how we do pre-processing on the RNs and parameter setting for the proposed algorithm. Next, we present the results of the algorithm on the different datasets. Finally, the hybrid clustering algorithm is compared with other state-of-the-art SPN-based clustering algorithms.

TABLE 4. Clustering results for resizing versus cropping the RNs, by testing different image resolutions on $\mathcal{D}_0^N$.

	Resizing							Cropping*						
Size	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	*ARI*	*Purity*	*FPR*	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	*ARI*	*Purity*	*FPR*	$\mathcal{N}$
1280×1024	**0.997**	0.758	0.861	0.858	**0.997**	**0.000**	**35/35**	——	——	——	——	——	——	– /35
1024×1024	**0.997**	**0.765**	**0.866**	**0.863**	0.996	**0.000**	**35/35**	——	——	——	——	——	——	– /35
960×720	0.986	0.725	0.835	0.831	0.992	**0.000**	34/35	0.693	0.614	0.651	0.641	0.966	0.007	31/35
512×512	0.953	0.440	0.602	0.595	0.964	**0.000**	48/35	0.676	0.497	0.572	0.561	0.962	0.007	37/35
256×256	0.508	0.027	0.051	0.048	0.596	**0.000**	280/35	0.654	0.303	0.411	0.401	0.882	0.004	60/35
128×128	0.031	0.146	0.050	0.004	0.176	0.132	26/35	0.487	0.138	0.212	0.203	0.598	0.004	159/35

* The highest resolution for cropping RNs is 960×720 px, based on Table 3.

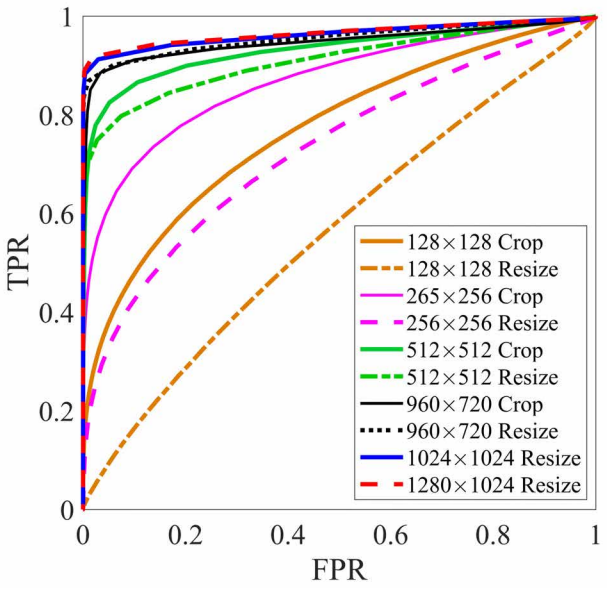


FIGURE 2. ROC curves obtained from resizing versus cropping with different resolution of RNs in $\mathcal{D}_0^N$.

1) RN RESIZING

Resizing the images involves upscaling or downscaling the images to a specific resolution. To select the specific resolution resulting in the best quality of the clustering, we consider a variety of image resolutions including 128×128, 256×256, 512×512, 960×720, 1024×1024, and 1280×1024 px. While for cropping, we crop each image from the center to the resolutions including 128×128, 256×256, 512×512, and 960×720 px, which is the lowest resolution of the *native* images in the dataset, see Table 3. The impact of resizing versus cropping is evaluated based on both correlation gain and clustering quality. The related results are presented in Figure 2 in terms of Receiver Operating Characteristics (ROC) showing the relationship between *TPR* and *FPR* measures. Each ROC curve is obtained by selecting a threshold from the range [-1,1] and comparing it with the full-pairwise correlation matrix of the resized/cropped RNs.

TABLE 5. Parameter values for the hybrid proposed method.

Notation	Value	Description
q	1000	batch size for partitioning dataset
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 $\mathcal{D}_1^N$,$\mathcal{D}_2^N$ and $\mathcal{D}^N$
	0.09	scaling factor in (10) for $\mathcal{D}_1^W$,$\mathcal{D}_2^W$ and $\mathcal{D}^W$
	0.07	scaling factor in (10) for $\mathcal{D}_1^{FH}$,$\mathcal{D}_2^{FH}$ and $\mathcal{D}^{FH}$
	0.03	scaling factor in (10) for $\mathcal{D}_1^{FL}$,$\mathcal{D}_2^{FL}$ and $\mathcal{D}^{FL}$
τ	0.004	minimum threshold in (10) for adaptive threshold

The results of clustering by using the correlation matrices obtained from resizing and cropping RNs in $\mathcal{D}_0^N$, with different resolutions, are also presented in Table 4. The highest value of each measure is highlighted in bold. The results in Figure 2 and Table 4 show that for the resolutions smaller than 960×720, cropping generates better results. In contrast, for the resolutions equal to or greater than 960×720, resizing delivers far better clustering results compared with cropping. As it can be easily seen from Table 4, resizing the RNs to the resolution 1024×1024 concludes the highest values of $\mathcal{P}$, $\mathcal{R}$, $\mathcal{F}$, *ARI*, *FPR*, and $\mathcal{N}$. Accordingly, in the experiments, we resize all the RNs to the resolution 1024×1024 px.

2) PARAMETER SETTING

There are few parameters should be set for the proposed algorithm. Using the sample datasets $\mathcal{D}_0^N$, $\mathcal{D}_0^W$, $\mathcal{D}_0^{FH}$, and $\mathcal{D}_0^{FL}$, we set the parameters empirically to the values resulting in the highest quality of the clustering. Due to the space limitation, we only present the results of setting the inflation parameter η in (6) and ψ for the adaptive threshold in (10), for $\mathcal{D}_0^N$. Based on the graphs presented in Figure 3, setting ψ to a value in the range $[0.11, 0.17]$ and $\eta = 1$ generates reasonable clustering results. With $\eta = 1$, if ψ is set to a high

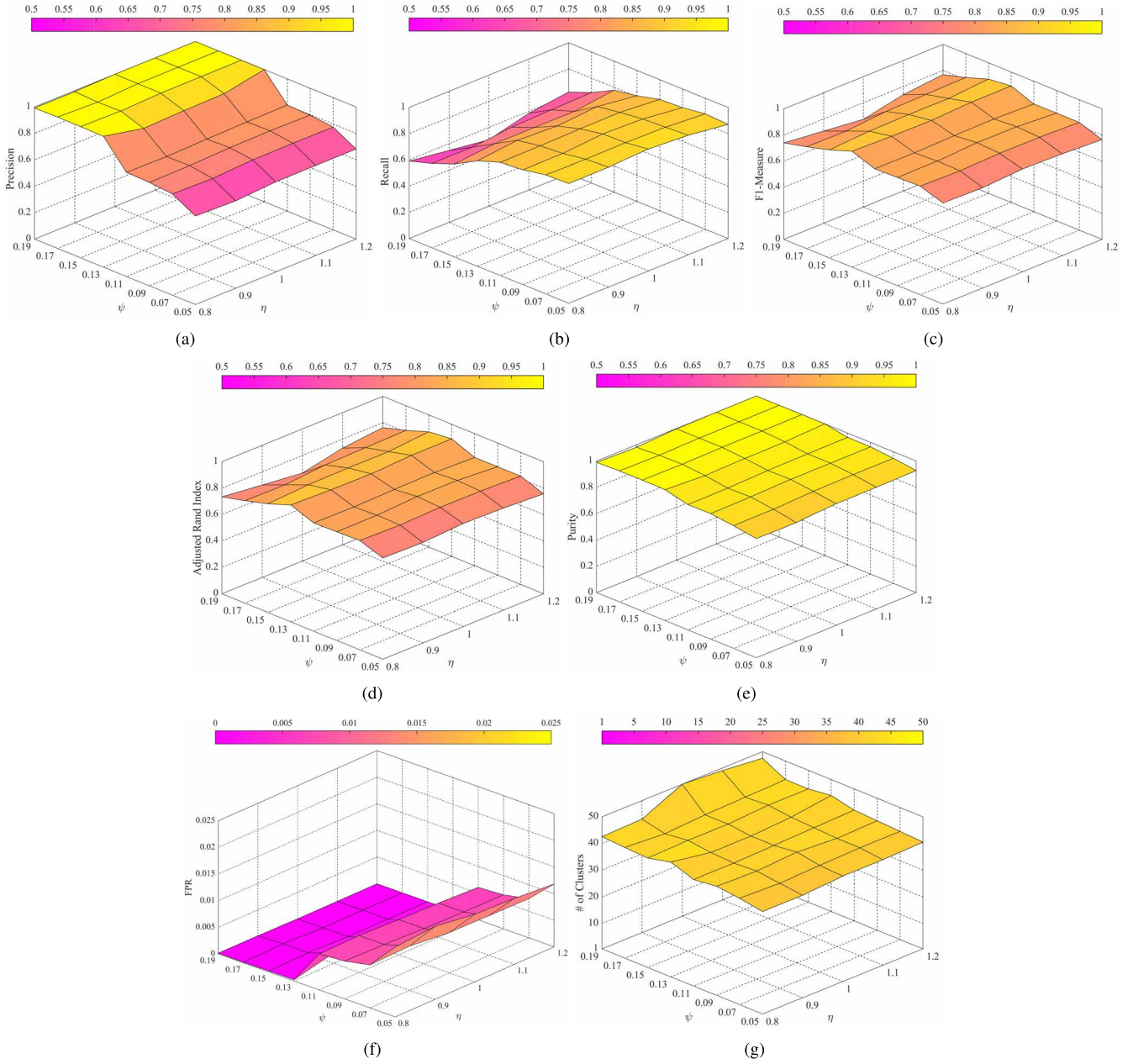


FIGURE 3. How the inflation parameter η and threshold ψ affect the clustering on $\mathcal{D}_0^{\mathbf{N}}$, (a) *Precision*, (b) *Recall*, (c) *F1-Measure*, (d) *Adjusted Rand Index*, (e) *Purity*, (f) *False Positive Rate*, and (g) number of the obtained clusters.

value, e.g., 0.19, the clustering generates a large number of fine clusters up to 50. Accordingly, to have the best values of $\mathcal{P}$, *purity*, $\mathcal{N}$ and *FPR*, we set $\psi = 0.15$ for *native images*. Similar parameter setting is performed for the *shared images*. In Table 5, all the required parameters and their set values are listed. It can be seen that while ψ depends on the quality of the images and it needs to be set to different values for *shared* and *native images*, the values of other parameters are set equally for all the images in the datasets. So, the appropriate values of ψ for the images of *Native*, *WhatsApp*, *Facebook high-resolution* and *Facebook low-resolution* datasets are set to 0.09, 0.07, and 0.03, respectively.

3) PROPOSED HYBRID CLUSTERING ALGORITHM

In this section, we present results of the hybrid clustering on different datasets $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}$, covering 11, 24, and 35 smartphones, respectively, for both *native* and *shared images*. We try to figure out how the proposed algorithm is capable of clustering images in different datasets. Since the algorithm randomly partitions the RNs into some batches, the results from multiple running of the algorithm on a given dataset may vary. Thereby, for each dataset, we run the algorithm 10 times and report the average results for the different measures. The results for the dataset D_1 which includes smartphones from the same models are shown in Table 6.

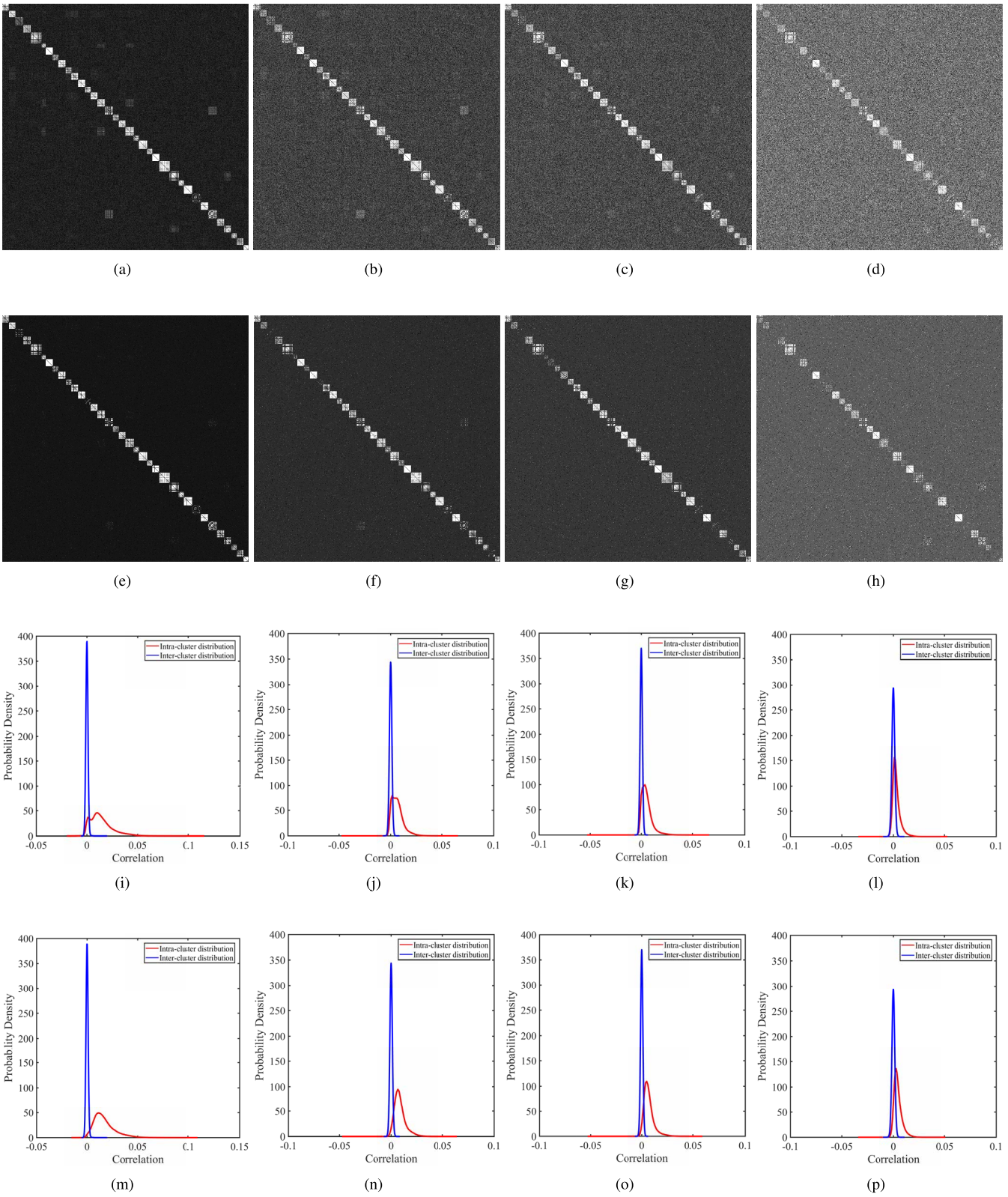


FIGURE 4. Effectiveness of the proposed algorithm in calculating correlation matrix: (a), (b), (c) and (d) are full-pairwise correlation matrices of RNs of the images in $\mathcal{D}^{N}$, $\mathcal{D}^{W}$, $\mathcal{D}^{FH}$ and $\mathcal{D}^{FL}$, respectively, and (e), (f), (g) and (h) are the corresponding correlation matrices calculated by the algorithm. The graphs (i)-(p) are probability distributions of the corresponding correlation matrices.

TABLE 6. Results of clustering on $\mathcal{D}_1$.

Dataset	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	*ARI*	*Purity*	*FPR*	$\mathcal{N}$
$\mathcal{D}_1^{\mathrm{N}}$	1.000	0.826	0.905	0.896	1.000	0.000	11/11
$\mathcal{D}_1^{\mathrm{W}}$	0.975	0.775	0.863	0.850	0.993	0.002	13/11
$\mathcal{D}_1^{\mathrm{FH}}$	0.994	0.720	0.835	0.821	0.994	0.001	12/11
$\mathcal{D}_1^{\mathrm{FL}}$	0.866	0.601	0.705	0.680	0.914	0.009	9/11

TABLE 7. Results of clustering on $\mathcal{D}_2$.

Dataset	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	*ARI*	*Purity*	*FPR*	$\mathcal{N}$
$\mathcal{D}_2^{\mathrm{N}}$	0.992	0.672	0.801	0.794	0.992	0.000	27/24
$\mathcal{D}_2^{\mathrm{W}}$	0.970	0.610	0.750	0.741	0.972	0.000	24/24
$\mathcal{D}_2^{\mathrm{FH}}$	0.958	0.627	0.758	0.749	0.966	0.001	25/24
$\mathcal{D}_2^{\mathrm{FL}}$	0.798	0.543	0.647	0.634	0.876	0.006	29/24

TABLE 8. Results of clustering on $\mathcal{D}$.

Dataset	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	*ARI*	*Purity*	*FPR*	$\mathcal{N}$
$\mathcal{D}^{\mathrm{N}}$	0.992	0.720	0.834	0.830	0.994	0.000	37/35
D^{W}	0.964	0.600	0.733	0.727	0.975	0.000	33/35
$\mathcal{D}^{\mathrm{FH}}$	0.962	0.610	0.746	0.740	0.975	0.000	33/35
$\mathcal{D}^{\mathrm{FL}}$	0.750	0.513	0.609	0.599	0.847	0.005	33/35

Due to the high resolution of *native images* in D_1^{N}, as shown in Table 3, the clustering results in the highest values of $\mathcal{P}$, $\mathcal{R}$, $\mathcal{F}$, *ARI*, *Purity*, *FPR*, and $\mathcal{N}$. Interestingly, the algorithm can detect all the 11 clusters corresponding to the smartphones in $\mathcal{D}_1^{\mathrm{N}}$. For the datasets $\mathcal{D}_1^{\mathrm{W}}$ and $\mathcal{D}_1^{\mathrm{FH}}$, the results are better than those for $\mathcal{D}_1^{\mathrm{FL}}$. It is because of low-resolution images in $\mathcal{D}_1^{\mathrm{FL}}$, see Table 3, but the algorithm is still capable to cluster the images with $\mathcal{P} = 0.866$, *Purity* $= 0.914$, *FPR* $= 0.009$, and $\mathcal{N} = 9/11$. In Table 7, similar trends can be observed for $\mathcal{D}_2$, with the best and worst results for $\mathcal{D}_2^{\mathrm{N}}$ and $\mathcal{D}_2^{\mathrm{FL}}$, respectively. However, clustering the images of $\mathcal{D}_2$ generates lower quality than that of $\mathcal{D}_1$. Obviously, the reason is that $\mathcal{D}_2$ covers images from completely different smartphone models and brands as well as various image resolutions. In Table 8, the results for *native* and *shared images* taken by all the 35 smartphones are shown. We consider it as the most challenging test for the proposed algorithm. Comparing the results in Tables 7 and 8, we can see that the algorithm can effectively cluster the images even in the larger datasets, i.e., $\mathcal{D}^{\mathrm{N}}$, $\mathcal{D}^{\mathrm{W}}$, $\mathcal{D}^{\mathrm{FH}}$ and $\mathcal{D}^{\mathrm{FL}}$.

To see why the proposed clustering is effective by calculating only a portion of the full-pairwise correlation matrix, in Figure 4, sub-figures (a)–(h), we present both full-pairwise correlation matrix and the correlation matrix calculated by the algorithm for each dataset. Indeed, the hybrid clustering exploits the sparseness of the full-pairwise correlation matrix and applies the Markov clustering, through which the trivial correlation values between the clusters in each batch are pruned. Subsequently, only the candidate clusters, which are selected based on the obtained transition matrix from the Markov clustering and the nearest neighboring, are checked in the merging. Therefore, to produce the adaptive threshold in (10), intra-cluster correlations and inter-cluster correlations of the candidate clusters need to be calculated, besides a few correlations between the clusters randomly partitioned in batches. Accordingly, for the datasets $\mathcal{D}^{\mathrm{N}}$, $\mathcal{D}^{\mathrm{W}}$, $\mathcal{D}^{\mathrm{FH}}$, and $\mathcal{D}^{\mathrm{FL}}$, the algorithm calculates 14.75%, 14.52%, 14.66%, and 14.59% of the corresponding full-pairwise correlation matrices, respectively. In sub-figures (i)–(p) of Figure 4, the probability distributions of the correlation matrices are also presented. The more the inter-camera and intra-camera correlation distributions are separable, the better the quality of the clustering is obtained. Comparing sub-figures (m)–(p) with (i)–(l), it can be seen that the calculated correlation matrices by the algorithm provide more separation between the probability distributions. In other words, the decrease in the overlapping areas of the probability distributions prevents wrong merging in the clustering, so more precise clusters are achieved.

TABLE 9. Comparison of clustering methods on $\mathcal{D}^{\mathrm{N}}$.

Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	*ARI*	*Purity*	*FPR*	$\mathcal{N}$
HMC	**0.992**	0.720	0.834	0.830	**0.994**	**0.000**	**37/35**
CC	0.987	0.863	**0.921**	**0.919**	0.915	**0.000**	46/35
FC	0.952	0.759	0.845	0.841	0.982	0.001	63/35
HC	0.205	**0.949**	0.338	0.304	0.828	0.112	23/35

4) COMPARISON WITH OTHER CLUSTERING ALGORITHMS

The proposed hybrid algorithm is compared with other SPN-based image clustering algorithms. We refer to the proposed hybrid algorithm, which combines the hierarchical and Markov clustering algorithms, as HMC. All the clustering algorithms including the proposed HMC, Correlation Clustering (CC) [13], Fast Clustering (FC) [14], and Hierarchical Clustering (HC) [8] are performed on the datasets $\mathcal{D}^{\mathrm{N}}$, $\mathcal{D}^{\mathrm{W}}$, $\mathcal{D}^{\mathrm{FH}}$, and $\mathcal{D}^{\mathrm{FL}}$, and the results are analyzed based on both clustering quality and running time. The comparison results in Tables 9–12 show that the proposed algorithm achieves the outstanding values of $\mathcal{P}$, *Purity*, *FPR*, and $\mathcal{N}$, apart from only *Purity* for $\mathcal{D}^{\mathrm{FL}}$. The values of $\mathcal{F}$ and *ARI* obtained by CC, and $\mathcal{R}$ by HC are the highest for the datasets $\mathcal{D}^{\mathrm{N}}$, $\mathcal{D}^{\mathrm{W}}$ and $\mathcal{D}^{\mathrm{FH}}$. Although FC does not show any improvement for the mentioned datasets, it generates the highest values of $\mathcal{F}$, *ARI* and *Purity* for $\mathcal{D}^{\mathrm{FL}}$. Also, for all the datasets, HC concludes the highest $\mathcal{R}$. Compared with the other algorithms, HMC has achieved the best values of $\mathcal{P}$, *Purity*, *FPR*, and $\mathcal{N}$.

In addition to the quality of clustering, the running time of a clustering algorithm is another important factor that should be taken into account for real-life applications. Hence, we compare the running time of the algorithms on the datasets $\mathcal{D}^{\mathrm{N}}$, $\mathcal{D}^{\mathrm{W}}$, $\mathcal{D}^{\mathrm{FH}}$, and $\mathcal{D}^{\mathrm{FL}}$ as shown in Figure 5. The time

TABLE 10. Comparison of clustering methods on $\mathcal{D}^{\mathbf{W}}$.

Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	ARI	$Purity$	FPR	$\mathcal{N}$
HMC	**0.964**	0.600	0.733	0.727	**0.975**	**0.000**	**33/35**
CC	0.952	0.787	**0.862**	**0.858**	0.856	0.001	56/35
FC	0.919	0.722	0.809	0.804	0.974	0.001	65/35
HC	0.245	**0.863**	0.382	0.352	0.813	0.081	32/35

TABLE 11. Comparison of clustering methods on $\mathcal{D}^{\mathbf{FH}}$.

Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	ARI	$Purity$	FPR	$\mathcal{N}$
HMC	**0.962**	0.610	0.746	0.740	**0.975**	**0.000**	**33/35**
CC	0.955	0.793	**0.866**	**0.863**	0.841	0.001	58/35
FC	0.913	0.758	0.828	0.824	0.974	0.002	64/35
HC	0.475	**0.823**	0.602	0.587	0.793	0.027	30/35

TABLE 12. Comparison of clustering methods on $\mathcal{D}^{\mathbf{FL}}$.

Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	ARI	$Purity$	FPR	$\mathcal{N}$
HMC	**0.750**	0.513	0.609	0.599	0.847	**0.005**	**33/35**
CC	0.712	0.632	0.669	0.660	0.776	0.007	42/35
FC	0.665	0.690	**0.717**	**0.680**	**0.915**	0.011	52/35
HC	0.031	**0.999**	0.061	0.003	0.520	0.941	2/35

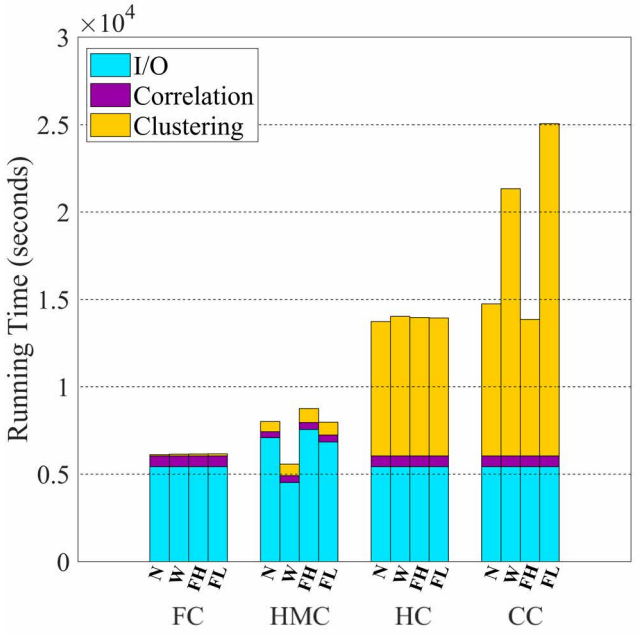


FIGURE 5. Running time of different clustering algorithms on $\mathcal{D}^{\mathbf{N}}$, $\mathcal{D}^{\mathbf{W}}$, $\mathcal{D}^{\mathbf{FH}}$, and $\mathcal{D}^{\mathbf{FL}}$ with image resolution 1024 × 1024.

for each algorithm is separately depicted by I/O, correlation calculation, and clustering operations. In terms of the overall running time, FC and CC are the fastest and the slowest algorithms, respectively. FC, HC, and CC have to be provided with the full-pairwise correlation matrix, so they have equal I/O and correlation calculation time. Despite the shorter clustering time of FC, HC and CC, they cannot be applicable to large-scale datasets due to the unavoidable time needed for the computation of full-pairwise correlation matrix. In contrast, by the proposed HMC, a small portion of the correlation matrix is calculated. This accelerates the clustering and also results in more precise clusters.

V. CONCLUSION

In this paper, a hybrid algorithm has been proposed to cluster the images captured and shared by the users of SN platforms, without prior knowledge about the types and number of the acquisition smartphones. The clustering results on the VISION image dataset confirm the effectiveness and efficiency of the proposed algorithm in comparison with the state-of-the-art algorithms. Particularly, the hybrid algorithm exploits image resizing, hierarchical and graph-based clustering algorithms, and an adaptive threshold to group the images. We have shown that resizing the RNs to a specific resolution can tackle the problem of low resolution of the *shared images* compressed by SN platforms. Using Markov clustering, the hierarchical clustering is conducted in such a way that the representative clusters with a higher probability of belonging to the same camera are selected for merging. The adaptive threshold for merging the representative clusters depends on the quality of the obtained clusters. The threshold that is updated during the hierarchical approach can produce more precise clusters even for images from the same model of smartphones. The scalability of the algorithm by partitioning the dataset into batches addresses the problem of memory constraint. Partitioning the dataset, exploiting the inherent sparseness of correlation matrix, and checking only the representative clusters in the merging result in the calculation of a small portion (about 15%) of the full-pairwise correlation matrix, accelerating the clustering. In our future work, we will consider clustering of different types of *shared images* on user profiles, not only from smartphones but also from the web.

ACKNOWLEDGMENT

The authors would like to thank the faculty and staff in the School of Computing and Mathematics, Charles Sturt University, Australia, for their support during the implementation of the work. They would also like to thank the anonymous reviewers for their invaluable comments and suggestions towards improving the paper.

REFERENCES

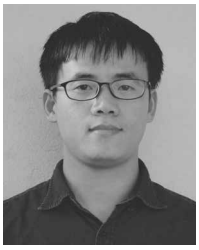
[1] F. N. Dezfouli, A. Dehghantanha, B. Eterovic-Soric, and K.-K. R. Choo, "Investigating social networking applications on smartphones detecting Facebook, Twitter, LinkedIn and Google+ artefacts on Android and iOS platforms," *Austral. J. Forensic Sci.*, vol. 48, no. 4, pp. 469–488, 2016.
[2] Q. Liu, X. Li, L. Chen, H. Cho, P. A. Cooper, Z. Chen, M. Qiao, and A. H. Sung, "Identification of smartphone-image source and manipulation," *Advanced Research in Applied Artificial Intelligence*. Springer, 2012, pp. 262–271.
[3] N. Yager and A. Amin, "Fingerprint classification: A review," *Pattern Anal. Appl.*, vol. 7, no. 1, pp. 77–93, 2004.

[4] J. Lukáš, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 205–214, Jun. 2006.
[5] G. J. Bloy, "Blind camera fingerprinting and image clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 532–534, Mar. 2008.
[6] C.-T. Li, "Unsupervised classification of digital images using enhanced sensor pattern noise," in *Proc. IEEE Int. Symp. Circuits Syst.*, May/Jun. 2010, pp. 3429–3432.
[7] R. Caldelli, I. Amerini, F. Picchioni, and M. Innocenti, "Fast image clustering of unknown source images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Dec. 2010, pp. 1–5.
[8] L. G. Villalba, A. S. Orozco, and J. R. Corripio, "Smartphone image clustering," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1927–1940, Sep. 2015.
[9] B.-B. Liu, H.-K. Lee, Y. Hu, and C.-H. Choi, "On classification of source cameras: A graph based approach," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Dec. 2010, pp. 1–5.
[10] I. Amerini, R. Caldelli, P. Crescenzi, A. D. Mastio, and A. Marino, "Blind image clustering based on the Normalized Cuts criterion for camera identification," *Signal Process. Image Commun.*, vol. 29, no. 8, pp. 831–843, 2014.
[11] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[12] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Correlation clustering for PRNU-based blind image source identification," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2016, pp. 1–6.
[13] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Blind PRNU-based image clustering for source identification," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 9, pp. 2197–2211, Sep. 2017.
[14] C.-T. Li and X. Lin, "A fast source-oriented image clustering method for digital forensics," *EURASIP J. Image Video Process.*, vol. 2017, no. 1, p. 69, 2017.
[15] L. Ertöz, M. Steinbach, and V. Kumar, "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data," in *Proc. SIAM Int. Conf. Data Mining*, 2003, pp. 47–58.
[16] X. Lin and C.-T. Li, "Large-scale image clustering based on camera fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 4, pp. 793–808, Apr. 2017.
[17] Q.-T. Phan, G. Boato, and F. G. B. De Natale, "Accurate and scalable image clustering based on sparse representation of camera fingerprint," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 7, pp. 1902–1916, Jul. 2019.
[18] M. Chen, J. Fridrich, M. Goljan, and J. Lukás, "Determining image origin and integrity using sensor noise," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 74–90, Mar. 2008.
[19] D. Shullani, M. Fontani, M. Iuliani, O. A. Shaya, and A. Piva, "VISION: A video and image dataset for source identification," *EURASIP J. Inf. Secur.*, vol. 2017, no. 1, p. 15, 2017.
[20] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.
[21] X. Lin and C.-T. Li, "Preprocessing reference sensor pattern noise via spectrum equalization," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 1, pp. 126–140, 2016.
[22] G. Chierchia, S. Parrilli, G. Poggi, C. Sansone, and L. Verdoliva, "On the influence of denoising in prnu based forgery detection," in *Proc. 2nd ACM Workshop Multimedia Forensics, Secur. Intell.*, 2010, pp. 117–122.
[23] R. E. Carlson and F. N. Fritsch, "An algorithm for monotone piecewise bicubic interpolation," *SIAM J. Numer. Anal.*, vol. 26, no. 1, pp. 230–238, 1989.
[24] A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah, and T. Herawan, "Big data clustering: A review," in *Proc. Int. Conf. Comput. Sci. Appl.* Springer, 2014, pp. 707–720.
[25] S. Van Dongen, "Graph clustering via a discrete uncoupling process," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 1, pp. 121–141, 2008.
[26] U. Mesa, "Regularized markov clustering algorithm on protein-protein interaction networks using ELLPACK-R sparse data format," M.S. thesis, Fac. Math. Natural Sci., Univ. Indonesia, Depok, Indonesia, 2012.
[27] S. Varia, "Regularized Markov clustering in MPI and map reduce," Ph.D. dissertation, Dept. Comput. Sci. Eng., Graduate School, Ohio State Univ., Columbus, OH, USA, 2013.
[28] V. M. Satuluri, "Scalable clustering of modern networks," Ph.D. dissertation, Dept. Comput. Sci. Eng., Graduate School, Ohio State Univ., Columbus, OH, USA, 2012.
[29] A. Bustamam, K. Burrage, and N. A. Hamilton, "Fast parallel Markov clustering in bioinformatics using massively parallel computing on GPU with CUDA and ELLPACK-R sparse format," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 9, no. 3, pp. 679–692, May 2012.
[30] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985.

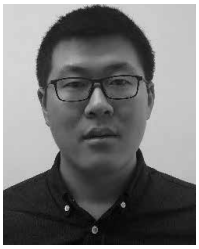
RAHIMEH ROUHI received the B.Sc. degree in computer science from the Shahid Bahonar University of Kerman, Iran, in 2010, and the M.Sc. degree in computer engineering (artificial intelligence) from the Science and Research Branch, Islamic Azad University, Iran, in 2013. She is currently pursuing the Ph.D. degree with the SmartData Research Group, Department of Computer Science and Engineering, University of Bologna, Italy. She is also a Research Assistant with the SmartData Research Group, Department of Computer Science and Engineering, and a Ph.D. Fellow with the Institute for Advanced Studies (ISA), University of Bologna. Her research interests include digital forensics, bioinformatics, image processing, machine learning, deep learning, and computer vision.

FLAVIO BERTINI received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the University of Bologna, Italy, in 2008, 2011, and 2015, respectively. In 2013, he was a Visiting Researcher with the Inria Bordeaux - Sud-Ouest. In 2018, he was a Visiting Researcher with the Stanford University under Marie Skłodowska-Curie Action. He is currently a Research Fellow with the SmartData Research Group, Department of Computer Science and Engineering, University of Bologna. His research interests include online fingerprinting, machine learning, and large-scale data analytics.

DANILO MONTESI received the Ph.D. degree in computer science from the University of Pisa, Italy. In 1993, he became a Fellow of the European Research Consortium for Informatics and Mathematics (ERCIM) and moved to the University of Lisboa, Portugal. Under the same fellowship, he later worked with the Rutherford Appleton Laboratory, Didcot, U.K., and the Department of Computer Systems and Telematics, University of Trondheim, Norway. Under a Senior NATO Fellowship, he worked with the Department of Computing, Purdue University, USA, in 1996. From 1996 to 2000, he was an Assistant Professor with the Department of Computer Science, University of Milano, Italy, and taught at the School of Information Systems, University of East Anglia, U.K. He visited British Telecom Labs, with a Senior Fellowship, in 1997, and Stanford University, in 2018, under Marie Skłodowska-Curie Action. In 2000, he became an Associate Professor with the University of Bologna, Italy. In 2002, he became a Full Professor with the University of Camerino. Since 2005, he has been a Full Professor of database and information systems with the University of Bologna. He is also the Founder of the SmartData Research Group. His principal research interests include the area of data and information management. In 1994, he was awarded the Human Capital and Mobility (HCM) Fellowship to work at Imperial College, London, U.K.



XUFENG LIN received the B.E. degree in electronic and information engineering from the Hefei University of Technology, Hefei, China, in 2009, the M.E. degree in signal and information processing from the South China University of Technology, Guangzhou, China, in 2012, and the Ph.D. degree in computer science from the University of Warwick, Coventry, U.K., in 2017. He is currently a Research Fellow with the School of Information and Technology, Deakin University, Australia. His research interests include digital forensics, multimedia security, machine learning, and data mining.



YIJUN QUAN received the B.A. degree in natural science from the Trinity College, University of Cambridge, U.K., in 2015, and the M.Sc. degree in computer science from the University of Warwick, U.K., in 2016, where he is currently pursuing the Ph.D. degree with the Department of Computer Science. He was a Visiting Scholar with the South China University of Technology (SCUT) under Marie Skłodowska-Curie Fellowship, in 2018. His research interests include multimedia forensics and security, machine learning, image processing, and computational photography.

CHANG-TSUN LI received the B.Sc. degree in electrical engineering from National Defence University (NDU), Taiwan, in 1987, the M.Sc. degree in computer science from the U.S. Naval Postgraduate School, USA, in 1992, and the Ph.D. degree in computer science from the University of Warwick, U.K., in 1998. He was an Associate Professor with the Department of Electrical Engineering, NDU, from 1998 to 2002. He was a Visiting Professor with the Department of Computer Science, U.S. Naval Postgraduate School, in 2001. He was a Professor with the Department of Computer Science, University of Warwick, until 2017. He was a Professor with Charles Sturt University, Australia, from 2017 to 2019. He is currently a Professor with the School of Information Technology, Deakin University, Australia. His research interests include multimedia forensics and security, biometrics, data mining, machine learning, data analytics, computer vision, image processing, pattern recognition, bioinformatics, and content-based image retrieval. The outcomes of his multimedia forensics research have been translated into award-winning commercial products protected by a series of international patents and have been used by a number of police forces and courts of law around the world. He is currently an Associate Editor of IEEE Access, *EURASIP Journal of Image and Video Processing* (JIVP), and *IET Biometrics*. He was involved in the organization of many international conferences and workshops and also served as a member of the international program committees for several international conferences. He is also actively contributing in keynote speeches and talks at various international events.