Computer Communications 127 (2018) 75–85

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Privacy-preserving image retrieval for mobile devices with deep features on the cloud

Nasir Rahim[a], Jamil Ahmad[a], Khan Muhammad[a], Arun Kumar Sangaiah[b], Sung Wook Baik[a,*]

[a] *Intelligent Media Laboratory, Digital Contents Research Institute, Sejong University, Seoul, Republic of Korea*
[b] *School of Computing Science and Engineering, Vellore Institute of Technology, Vellore, India*

ARTICLE INFO

Keywords:
Artificial intelligence
Image retrieval
Deep features
Cloud
Privacy preservation
Features extraction

ABSTRACT

With the prevalent use of mobile cameras to capture images, the demands for efficient and effective methods for indexing and retrieval of personal image collections on mobile devices have also risen. In this paper, we propose to represent images with hash codes, which is a compressed representation of deep convolutional features using deep auto-encoder on the cloud. To ensure user's privacy, the image is first encrypted using a light-weight encryption algorithm on mobile device prior to offloading it to the cloud for features extraction. This approach eliminates the computationally expensive process of features extraction on resource constrained devices. A pre-trained convolutional neural network (CNN) is used to extract features which are then transformed to compact binary codes using a deep auto-encoder. The hash codes are then sent back to the mobile device where they are stored in a hash table along with image location. Approximate nearest neighbor (ANN) search approach is utilized to efficiently retrieve the desired images without exhaustive searching of the entire image collection. The proposed method is evaluated against three different publicly available image datasets namely Corel-10K, GHIM-10K, and Product image dataset. Experimental results demonstrate that features representation using CNN and auto-encoder shows much better results than several state-of-the-art hashing schemes for image retrieval on mobile devices.

1. Introduction

With the proliferation of smartphones, tablet PCs, and smart wearable devices huge amount of multimedia data such as images, videos, and voice are generated and distributed every day. “How to manage such massive data on a resource-constrained device”, is a serious issue. One natural solution to handle such problem might be adapting cloud based services due to its tremendous advantages, such as on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk [1]. According to a report, Facebook is the largest growing image storage and sharing cloud service today [2]. Additionally, the efficient retrieval of image related information in enormous image datasets is another challenging issue [3,4]. There are numerous cloud based image service providers such as Amazon Cloud Drive, Flicker, iCloud by Apple, and Google that support efficient indexing and retrieval of multimedia data.

Despite the fact that cloud computing seems natural solution to manage large scale image repositories, new challenges regarding data control and privacies have also arisen. Cloud based photo management systems are still struggling to handle the issue of efficient searching with user's privacy. For example Facebook was criticized by huge community when they introduced an auto image recognition service in the year 2011 through which faces and objects in any photo can be easily recognized and searched [5,6]. Anyone could easily stalk and track anyone else using various image search engines such as Google and Yahoo image search engines. After the prolonged controversy of one year, Facebook agreed to remove the automatic face recognition mechanism from their system. But this functionality has included once again due to the requirement of efficient image searching and retrieval despite significant disapproval of community. Google removed face recognition functionality from Google Glasses to handle similar privacy issues. The primary reason of rejecting automatic face recognition services by vast community is that they can be stalked and illegally searched by malicious hackers from anywhere, especially if the search is performed by the system automatically. However this feature could cause the system to generate image searching results more intelligently such as retrieving a list of images being captured with a specific friend. By modifying only the access control mechanism from public to private does not guarantee that the uploaded image is totally safe on a cloud

[*] Corresponding author.
E-mail addresses: nrahim3797@gmail.com (N. Rahim), jamilahmad@sju.ac.kr (J. Ahmad), khanmuhammad@sju.ac.kr (K. Muhammad), arunkumarsangaiah@gmail.com (A.K. Sangaiah), sbaik@sejong.ac.kr (S.W. Baik).

https://doi.org/10.1016/j.comcom.2018.06.001
Received 15 October 2017; Received in revised form 8 May 2018; Accepted 1 June 2018
Available online 03 June 2018

platform. Furthermore, disabling the automatic object recognition or encrypting sensitive contents of the image are not even the proper solution, because it also decreases the efficiency of image searching functionality.

Although user prefers systems offering both functionalities i.e., efficient and privacy preserved image retrieval. The challenging task is outsourcing the content-based image search on cloud platform without letting the cloud anything know about the image contents during the processing phase. Usually, similarity between two images is computed by measuring the distance between the feature vectors. Such process involves both addition and multiplication operations. One ideal solution for handling such issue might be using fully homomorphic encryption techniques such as [7], but such approach is not easily adoptable due to its larger computational complexity. Another approach might be secure multi-party computation which supports privacy-preserved vector similarity between two images [8,9]. However, this process requires both parties i.e., image owner and user to interact constantly, which is not always possible for the image owner.

Recently, hashed based searching schemes have attracted significant attention due to the growing demands of efficient access in large data repositories [10]. Hash-based search schemes aim at efficient access to the relevant data points in large scale datasets using approximate nearest neighbor (ANN) search methods. They follow the mechanism of locality sensitivity hashing functions which perform dimensionality transformation from high to low dimension to preserve the original neighbors in the hamming space [11]. These compact codes are used to directly access the nearest neighbors of the query image without performing linear search. Numerous hashing schemes have been presented in the recent past which transform high dimension feature vectors of image to low dimension compact binary hash codes. These hashing methods include locality sensitivity hashing (LSH) [12], principle component analysis hashing (PCAH) [13], spectral hashing (SH) [14], spherical hashing (SpH) [15] and density sensitivity hashing (DSH) [16] etc. Apart from these approaches, several other techniques have been presented for transforming deep features to binary codes including [17,18] where the high dimensional features are directly transformed to binary codes using fast Fourier transform.

The aforementioned methods exploited various approaches for efficient and privacy preserved image retrieval. However, none of these methods produce a win-win situation that is both computationally less expensive as well as less prone to security issues. Some existing approaches such as [7,19] provide high-level encryption algorithms but require huge processing units, which makes them less suitable for real-time applications. Other techniques were computationally efficient but lack acceptable privacy, decreasing its applicability in various areas of interest [20]. Considering all these concerns, there is a need to come up with a method, maintaining a balance between privacy and its flexibility with limited resources.

In this article, we propose an efficient image retrieval approach for mobile devices that ensures the privacy of the user's data using an encryption algorithm, which is computationally less complex and easily adaptable by mobile phones. In addition, our method shifts the computationally heavy process to the cloud that can efficiently perform features extraction and hash codes generation using VGG network and deep Autoencoder. The generated hash codes allow the mobile device to retrieve the identical images by processing only N number of bits locally. Therefore, our method successfully improves the retrieval performance on smartphone devices, while avoiding the excessive use of device's resources to perform heavy computations. The main contributions of this work are summarized as follows:

1. We propose a privacy preserving image retrieval framework for mobile devices. Our framework ensures the privacy of user's data by encrypting images using an energy-friendly encryption algorithm. The encryption algorithm is optimized considering both the limited resources of smart phones and security of user's data, providing a better balance between them. This unique characteristic makes our framework more suitable for smart phones.
2. Avoiding the time-consuming efforts of the features engineering, we use a pre-trained CNN model i.e., VGG-16 which automatically learns rich features from the user's data.
3. Learned features are compressed to pre-defined length of hash codes using deep Autoencoder, making this process of image retrieval highly efficient. Further, the computationally expensive process of features extraction and hash codes generation is offloaded to the cloud, saving the resources of smart phones.

The rest of the paper is organized as follows: Section 2 explains the relevant literature in the field of image retrieval. Section 3 covers the technical detail of the proposed approach. Section 4 discusses the experimental results. The paper is concluded with future research directions in Section 5.

2. Related work

Searchable encryption techniques enable the user to search for specific information in an encrypted data collection. Majority of the existing searchable encryption techniques are utilized for textual information extraction. The basic cryptographic schemes in earlier stages were used to search for the query term in the encrypted text document with additional struggle not to let the server to learn anything from outsourced data [21,22]. Thereafter, a large number of methods in various thread models were proposed in the literature, such as multi-keyword ranked search [23–27], similarity search [28,29], and dynamic search [30,31] to attain various search functionalities. However, many of these schemes are reliable and easily adoptable for secure image retrieval task. Shashank et al. [32] proposed private content-based image retrieval scheme where the query image is protected before shifting it to cloud, while the image database remains unencrypted on the server. Other researchers such as [33,34] outsourced the computationally expensive task of features extraction to the cloud sever in privacy-preserving manner. Running cloud based query in a privacy-preserved manner is the key technique in CBIR outsourcing. Furthermore, homomorphic based techniques require high computation resources which make it impractical to adopt for smartphone devices. The first privacy-preserving CBIR scheme over encrypted image was proposed by Lu et al. [35] where images are represented by its visual features stored on the cloud server. Furthermore, Jaccard similarity between the visual features of the query image and features database is calculated to perform similarity matching between the two corresponding images. Feature vectors of the image are kept secure by employing order-preserving encryption and min-hash algorithm. In another work, Lu et al. [36] explored three features protection schemes and compared it in terms of its security, retrieval performance and computational complexity. The authors showed that Hamming distance can be easily calculated for those feature vectors encrypted with bitplane randomization, and randomized unary encoding. However features encrypted with random projection scheme can be utilized for calculating L1 distance in encryption domain. Cheng et al. [37] designed a secure CBIR system by utilizing bitplane randomization, and randomized unary encoding same as discussed in [36]. Ferreira et al. [19] introduced a new cryptographic scheme, called “IES-CBIR” which is particularly designed for privacy-preserving image indexing and retrieval in large image repositories. In their work they extracted texture from color component and encrypted each component using different encryption schemes. Texture component was encrypted using probabilistic cryptosystem while the color component was encrypted using deterministic cryptosystem to perform CBIR using color property. Cheng et al. [38] proposed an image retrieval scheme for stream cypher based encrypted image. According to this technique Markov features are extracted from encrypted image and is classified using support vector machine [39,40].

Large scale image datasets demand efficient indexing and retrieval of relevant images to the query image. Recently, ANN based searching schemes such as locality sensitivity hashing have shown very encouraging results. Images are typically represented as high dimensional features vectors where the Euclidean distance between the feature vectors corresponds to image similarity. The main goal of hashing schemes is to generate a low-dimensional embedding in hamming space while preserving the neighborhood. Hash based image retrieval involves efficiently accessing the nearest neighbors of the query image by calculating hamming distance. Numerous hashing schemes have been presented in literature such as PCAH [41] which uses principle direction of data to convert features vectors to binary codes. LSH [11] uses randomized algorithm to generate hash codes from the features vector using a random threshold. Theoretically, the hamming distance between LSH codes seems highly correlated to the Euclidean distance between features vector however, in reality this process yields to a very inefficient codes. Spectral hashing (SH) [14] choose binary codewords on the basis of minimum distance between same data points, where similarity is defined by an approximate proximity matrix. SH performs better than LSH, however, its optimization and generalization for new data points is difficult. SpH [15] addresses generalization problem by utilizing Eigen functions of weighted Laplace-Beltrami operations which efficiently generate hash codes than SH. However, this process requires very large memory for optimization. Similarly, other hashing schemes such as Kernelied LSH (KLSH) [42], PCA with random rotation (PCA-PR) [43] and iterative Quantization (ITQ) [44], Circulant binary embedding with optimization (CBE-opt) [45] and compact quantization (CQ) [46] have also been proposed in recent years.

3. Proposed framework

In this section we present an efficient image retrieval scheme on smartphone devices that performs reasonable amount of computation on local CPU while providing high rank image searching accuracy in large-scale datasets. The proposed method consists of light-weight image encryption algorithm that allows user's privacy while offloading query image to the cloud. Hash codes are generated on cloud using VGG network followed by Autoencoder that allows the process of mobile image searching efficient for large-scale datasets. Details about image encryption, features extraction, hash codes generation, indexing and retrieval processes are provided in the subsequent sections.

3.1. Image encryption

For encryption, we use an image encryption method which is both computationally efficient and resilient against various attacks. Our image encryption scheme consists of two main steps: secret keys generation and encryption. The details of these steps are provided in the sub-sequent sections.

3.1.1. Secret keys generation

In this work, the initial key is hashed using SHA-256 hash function to a fixed length for ensuring its maximum sensibility. Its output is further used for producing the initial values of the chaotic system to generate the required cryptographic keys. Previously, we used Zaslavsky chaotic map to generate appropriate encryption keys for the cipher algorithm [47,48] but in this work, we modify the chaotic map to another more secure and less complicated map. The detailed steps for initialization of secret keys are given in Algorithm 1.

In Algorithm 2, the steps for generating the encryption keys for our method using the secret key and an initial [32,32] matrix are given. This work uses the same matrix K_{in}as used in our previous work [48]. First, we set up the initial values (x_0, y_0, r) for the chaotic map using Algorithm 1. Then, we generate the random sequences *S1, S2,* and *S3* based on the 2D-logistic map as given in Algorithm 2. The confusion and diffusion operations are manipulated with each block [32,32] and therefore, our encryption approach can be applied to any digital with size format of [N*32, M*32], where M and N are integer numbers. The encrypted image has similar size to the decrypted image. An example of sample images from the datasets along with their encrypted and decrypted versions are shown in Fig. 1.

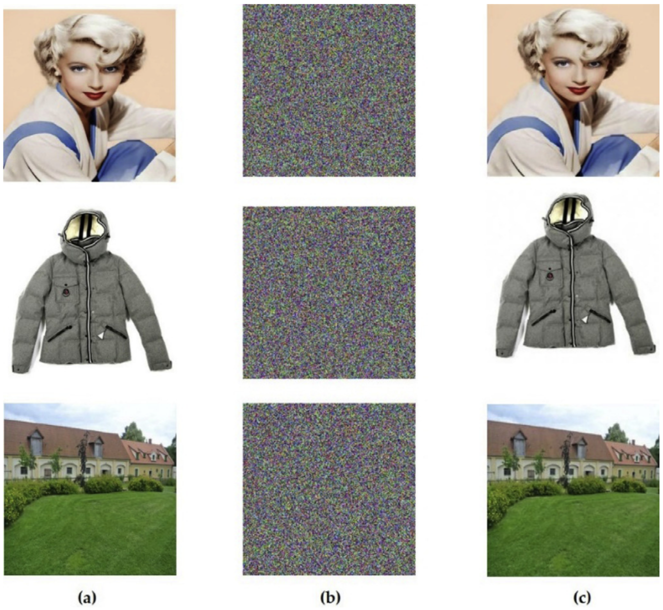

Fig. 1. Sample test images from the dataset. (a), original images, (b), encrypted images, and (c) decrypted images.

3.1.2. Encryption process

In this procedure, the input image is converted to one matrix. Next, some random bits are generated using true random number generator. The generated bits sequence are added the input image pixels using bitwise-XOR operation. It should be noted that the generated bits noise should have the same image size. From Fig. 1, it is clear that adding random bits is not affecting the visual presentation of the images after its decryption. In our scheme, any modification to the initial values completely changes the ciphered image. Therefore, the encrypted image produced is completely different despite the same input and secret keys. The main steps of the image encryption method are given in Fig. 2.

3.2. Features extraction

There are two main phases involved in automatic classification: feature extraction phase and classification phase. Generally, the feature extractor is composed of hand-crafted transformation of the input image with, main focus of making the classification phase more efficient. To date, a variety of feature extraction methods are presented in the literature which include Scale-Invariant Feature Transform (SIFT) [6], Speeded-up Robust Feature (SURF) [49], Local Binary Pattern (LBP) [50], Histogram of Oriented Gradient (HOG) [51] and Oriented FAST and Rotated BRIEF (ORB) [52]. Support vector machine (SVM) is one of the well-known classification algorithms used to solve many computer vision problems. However, random forest, decision tree, and multi-layer perceptron (MLP) can also be used for such purposes. Generally, choosing strong features may lead to high accuracy result during classification phase.

Extraction of hand-crafted features requires a human expert to decide which features better matches for the solution of a problem in a specific domain. To overcome this drawback, an automated tool is needed to extract the features from the given data and suggest a solution using the extracted features for the specified problem.

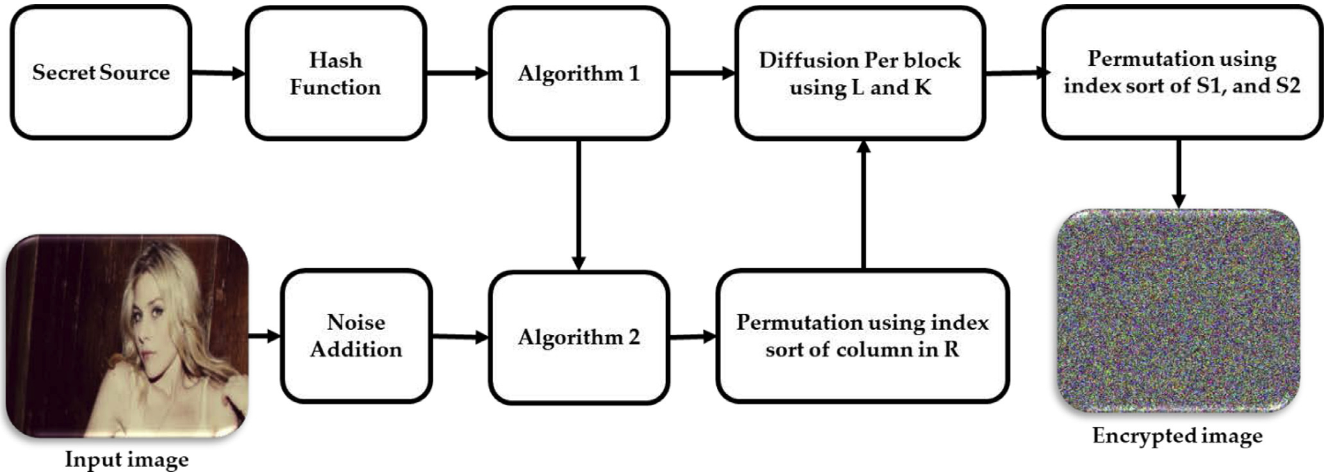
Secret Source
Hash Function
Algorithm 1
Diffusion Per block using L and K
Permutation using index sort of S1, and S2
Input image
Noise Addition
Algorithm 2
Permutation using index sort of column in R
Encrypted image

Fig. 2. Main steps of the image encryption method.

Convolutional Neural Network (CNN) is one of the best solutions to address such problems [53]. CNN has multiple convolutional layers; each layer is followed by a ReLU and a pooling layer. At the end we have fully connected layers followed by a Softmax layer. The architecture of a CNN is designed to have the ability of learning variety of different features at each layer which is more abstract representation of the image data. It can perform both operations i.e., features extraction and classification.

3.3. CNN architecture

For features extraction, we used Visual Geometry Group (VGG) [54] model as shown in Fig. 3. The main reason of using this model is the higher accuracy due to increased architectural depth and large number of learned parameters as compared to other similar models i.e., AlexNet [55]. The input image to the VGG network is of fixed size i.e., 224 × 224 × 3. The only preprocessing performed on the input image is subtracting mean image, formed during the training phase of the network. In VGG network, input image is passed through a stack of various convolutional layers of different receptive fields. The primary focus of VGG network is to investigate the effect of increased convolutional network depth over the accuracy. As the network goes deeper and deeper, the number of learned filters increase. The stride rate for convolutional layers and pooling layers remains the same throughout the network which is 3 × 3 with stride 1 in convolutional phase and 2 × 2 with stride 2 in pooling phase. In the initial two convolutional layers, 64 and 128 kernels are learned respectively. The rest of the layers include 256, 512 and 512 kernels, respectively. To preserve the features maps size similar as the input during the convolutional phase, border pixels are mirrored before each convolutional operation. Convolutional layers are followed by three fully connected layers. The first two FC layers consist of 4096 neurons while the final FC layer compresses the features values to 1000 channels because of ILSVRC classification problem.

It has been shown in the past that the CNNs trained on huge datasets like ImageNet can serve as generic feature extractors. We used global sum pooling of the last convolutional layer as features to represent images. This layer consists of 1000 activation maps where each map refers to one particular class of the ImageNet dataset. By taking the individual sum of each map in this layer, we get 1000 values which are quantized to obtain compact representations. Each value is quantized using four bits which reduced the memory requirements without any significant loss in retrieval accuracy.

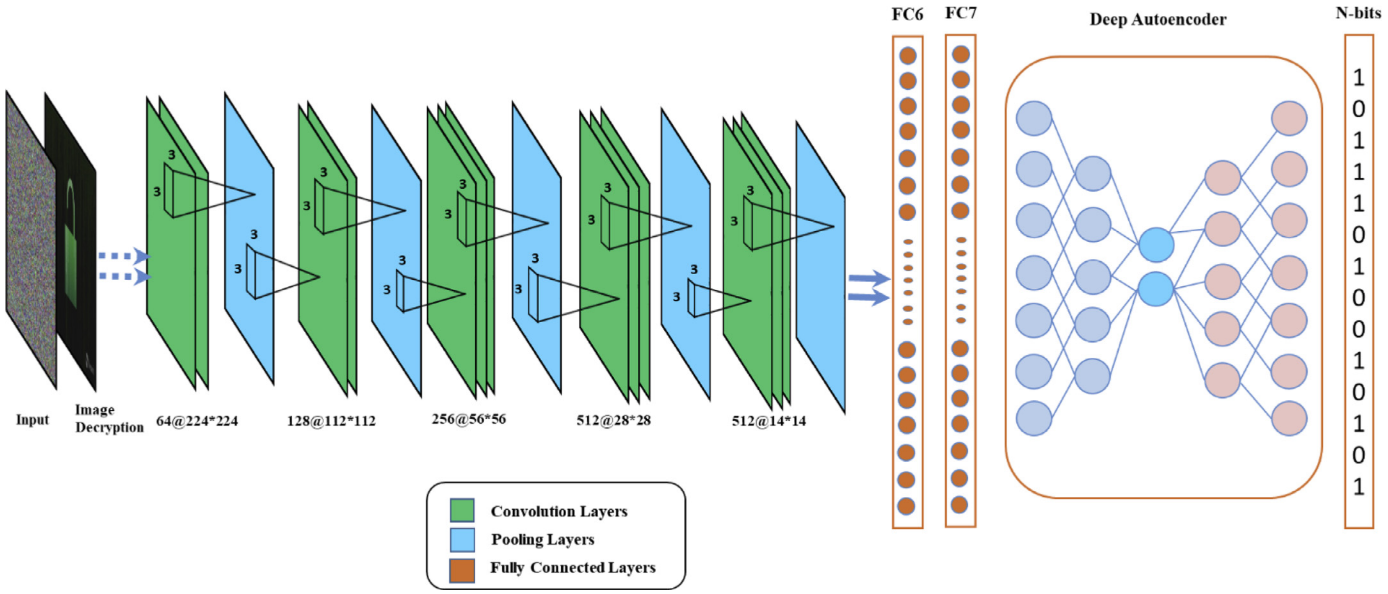

Fig. 3. CNN architecture for features extraction followed by deep Autoencoder for N-bit hash codes generation.

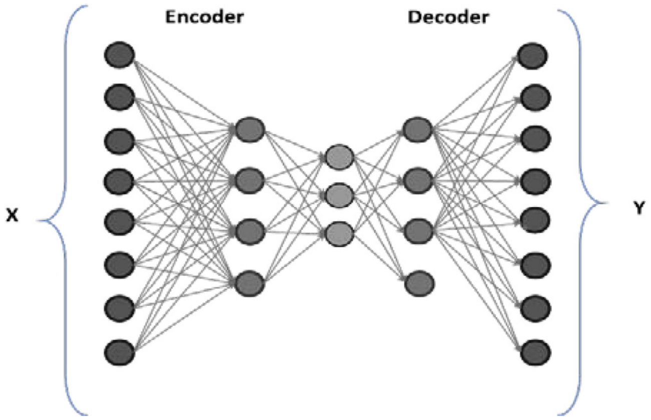

Fig. 4. Deep Autoencoder.

3.4. Autoencoder

The main architecture of Autoencoder is a feed forward neural network with an input layer, an output layer and one or more hidden layers between them. Fig. 4 depicts the architecture of deep Autoencoder. An Autoencoder framework usually includes the encoding and decoding processes. Given input x is encoded by Autoencoder across one or more hidden layers by employing several encoding processes, which is decoded again to reconstruct an output $\hat{x}$. The main focus of Autoencoder is to reduce the deviation of $\hat{x}$ from the given features value x. The summarized representation of deep Autoencoder is given as follows:

$$\text{Encoding}\xi = f(M_1 x + C_1) \tag{1}$$

$$\text{Decoding}: \hat{x} = f(M_1'\xi + C'_1) \tag{2}$$

Where f is a nonlinear sigmoid activation function, $M_1 \in R^{\mathrm{p}\times\mathrm{q}}$ and $M_1' \in {}^{\mathrm{q}\times\mathrm{p}}$ are parameter matrices, $C_1 \in \mathrm{R}^{\mathrm{p}\times 1}$ and $C_1' \in \mathbb{R}^{\mathrm{q}\times 1}$ are bias vectors, and $\xi \in \mathbb{R}^{\mathrm{p}\times 1}$ shows output of the hidden layer. For an input features vector $\{x_i\}\overset{n}{i} = 1$, the reconstruction cost can be computed using squared error cost function i.e., $\sum_{i=1}^{n} ||\hat{x}_i - x_i||^2$. The main goal of Autoencoder is to learn weight matrices M_1 and M_1', and bias vectors C1 and C_1' by minimizing the reconstruction error as follows,

$$\underset{M_1,\, C_1,\, M_1',\, C_1'}{\min} \sum_{i=1}^{n} ||\hat{x}_1 - x_i||^2 \tag{3}$$

In the proposed method we used deep Autoencoder to compress the 4096 dimensions feature vector into binary codes of 512, 256 and 128 dimensions. For this purpose, we trained an Autoencoder with more than one hidden layers. Number of neurons in the given hidden layers were defined in a hierarchical order. For our problem the designed Autoencoder was composed of five hidden layers. First layer contained 2048 neurons, followed by 1024, 512, 256, and 128 neurons, respectively. These layers were trained using the feature vectors obtained from VGG network. The proposed system was evaluated using three different length hash codes of 512, 256 and 128 bits. To transform 512 dimension feature vector to hash codes, 0.5 threshold value was found the optimum value, whereas for 256 and 128, 0.05 threshold value was chosen. Further, the generated hash codes were used to perform image retrieval. The process of database population, and image retrieval using generated hash codes are described in the subsequent sections.

3.5. Image indexing

After capturing an image using smart phone's camera, the image is forwarded to encryption module, where it is resized to 256 × 256 pixels and is encrypted using a light-weight encryption algorithm. The encrypted image is then offloaded to the cloud via internet connection. The uploaded image is decrypted on the cloud and is then pass through VGG network. The network generates 4096 dimension features vector at FC7 layer which is further compressed to 512, 256 and 128 bits hash codes using Autoencoder as discussed in Section 3.4. Generated hash codes are returned to the mobile device. Once the hash code is received by mobile app, the local path of the captured image and its corresponding hash code is stored to local database. Since the proposed method has been evaluated using three different length hash codes, this makes the total length of the hash table to 2^{512}, 2^{256} and 2^{128} .The entire process of image indexing using the cloud and smart phone is depicted in Fig. 5.

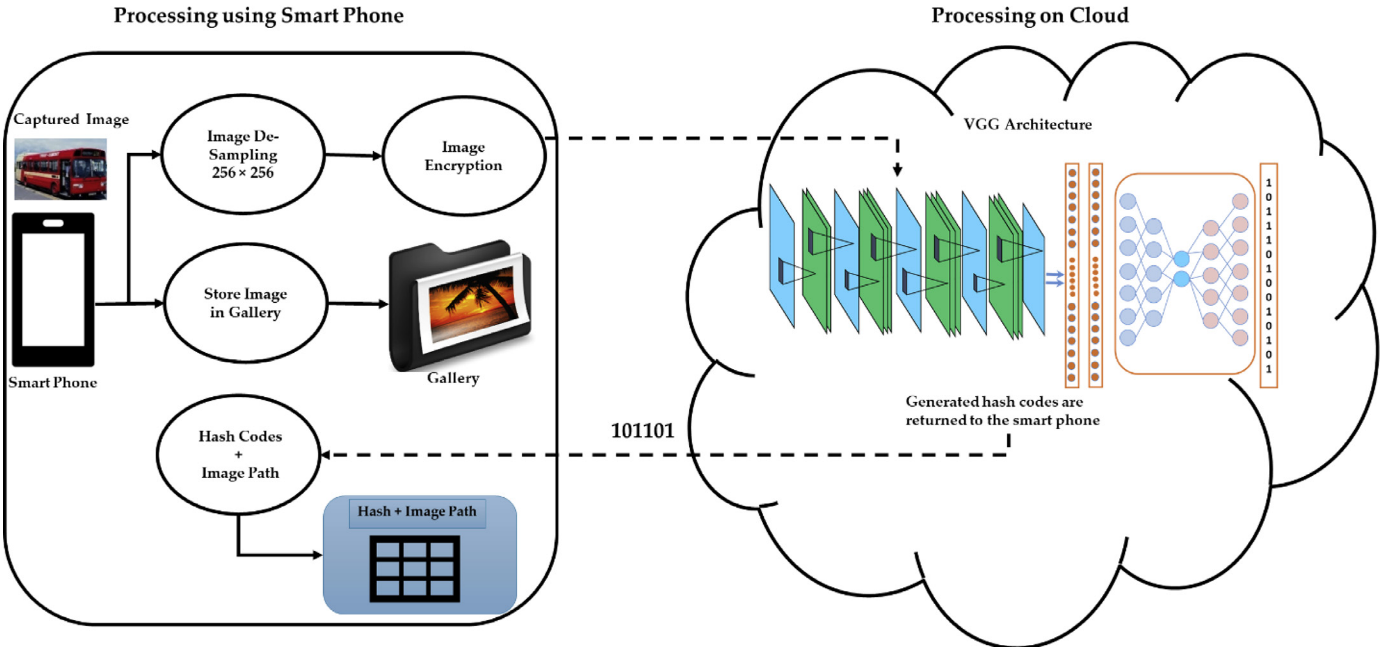

Fig. 5. Image indexing using smart phone with hash generation using cloud.

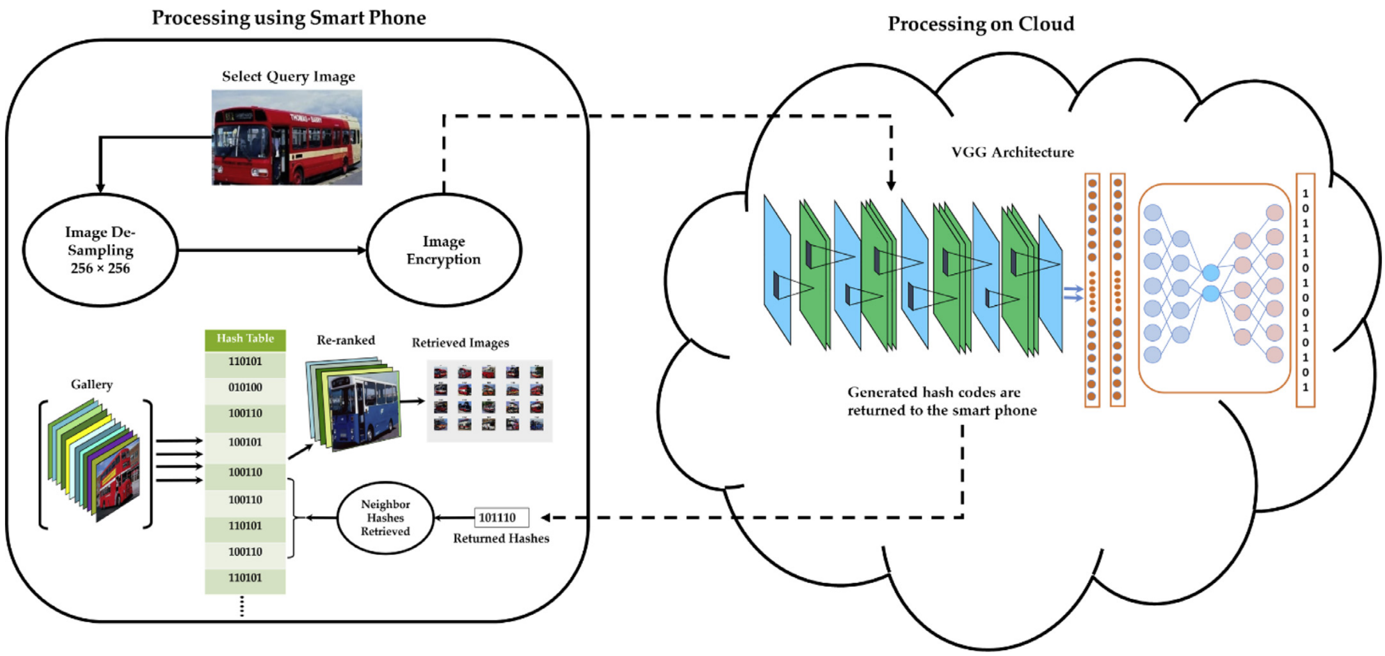

Fig. 6. Image retrieval from smart phone's gallery.

Table 1
Execution time (sec) for encryption and decryption of sample images using our method.

Image number	1	2	3	4	5	6	7	8	9	10	Mean
Encryption	0.15	0.16	0.15	0.15	0.17	0.15	0.15	0.75	0.16	0.16	0.215
Decryption	0.15	0.15	0.15	0.15	0.15	0.15	0.17	0.77	0.15	0.16	0.215

Table 2
Comparison based on encryption speed (Kb/s).

Algorithm	Encryption
Our method	1166
Zhou et al. [58]	≈ 390
Belazi et al. [59]	≈ 200
Yao et al. [60]	≈ 440
Hamza et al. [47,48].	≈ 205

3.6. Image retrieval

In retrieval phase, the query image is first de-sampled to 256 × 256 and is then encrypted using the algorithm given in Section 3.1. The encrypted image is then offloaded to the cloud where it is decrypted and converted to various length hash codes using VGG network and Autoencoder. The generated codes are returned to the device via internet. Once the hash code is received by the mobile app, hamming distance is calculated between the returned codes and existing codes of database. The top ranked images with minimum distance from the gallery images are displayed to the user. In this way, the proposed framework minimizes the overall computation complexity, network traffic, saves bandwidth, preserves user's privacy, and improves the retrieval performance. Fig. 6 shows the overall procedure of image retrieval for smart phones.

4. Experiments and results

We performed several experiments using actual smart phone device as an emulator i.e., SONY Xperia C3, having Android Lollipop 5.1.1 version installed on it. The processing components of the device include Quad-core 1.2 GHz Cortex-A7 processing unit and 1 GB of RAM and 8 MP camera. For emulating cloud infrastructure, we used apache's famous distribution XAMPP to make our local desktop as server to user's query. Experiments are conducted from two perspectives: image encryption and retrieval.

4.1. Evaluation of our image encryption method

In this section, the image encryption approach is experimentally tested using its execution time and performance comparison with other methods. For real-time applications, the algorithm should ensure both fast speed and security [56,57], which is one of the distinguishing properties of our method. The execution time for a set of images for encryption and decryption is given in Table 1. The comparative results are shown in Table 2 from where the superior speed of our method can

Table 3
Comparative results.

	Size Image	Key space	Speed (ms)	Correlation	NPCR	UACI
Our	[256, 256,3]	10^{90}	≅150	0.0035	99.615	33.4658
[62]	[1024,1024,1]	2^{624}	2513	0.0129	99.6177	33.6694
[63]	[256,256,1]	0.25×10^{64}	1320	0.0060	99.6200	33.5100
[64]	[256,256,1]	10^{56}	547	0.0722	> 99	≅33.43

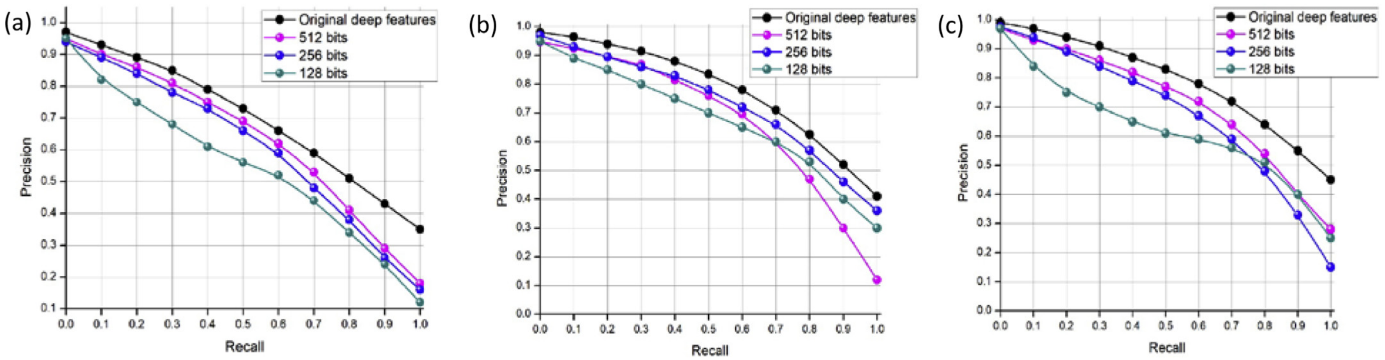

Fig. 7. Retrieval results using the proposed hashing scheme and original deep features for (a) Corel-10K (b) Product image and (c) GHIM-10K image datasets.

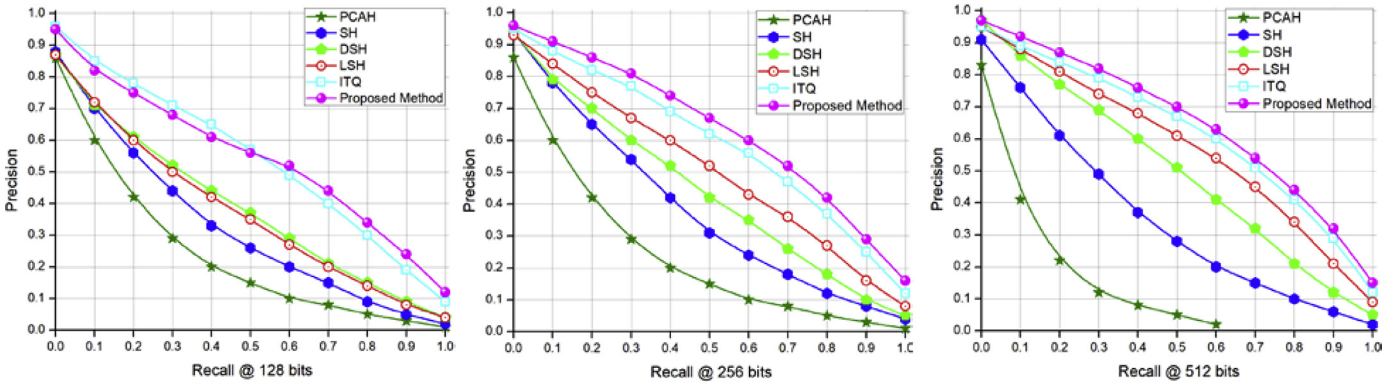

Fig. 8. Retrieval performance of the proposed method in comparison with the other state-of-the-art hash code schemes for Corel-10K image dataset.

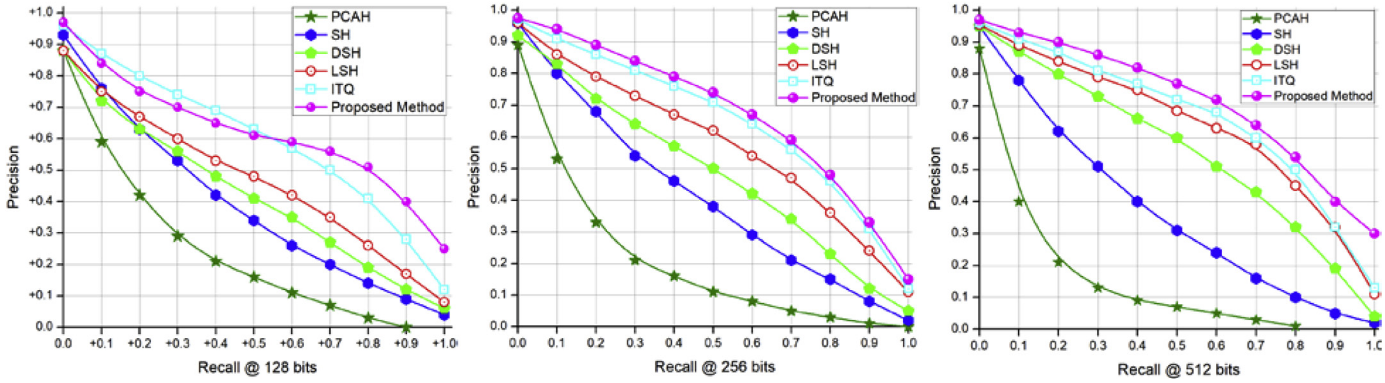

Fig. 9. Retrieval performance of the proposed method in comparison with the other state-of-the-art hash code schemes for GHIM-10K image dataset.

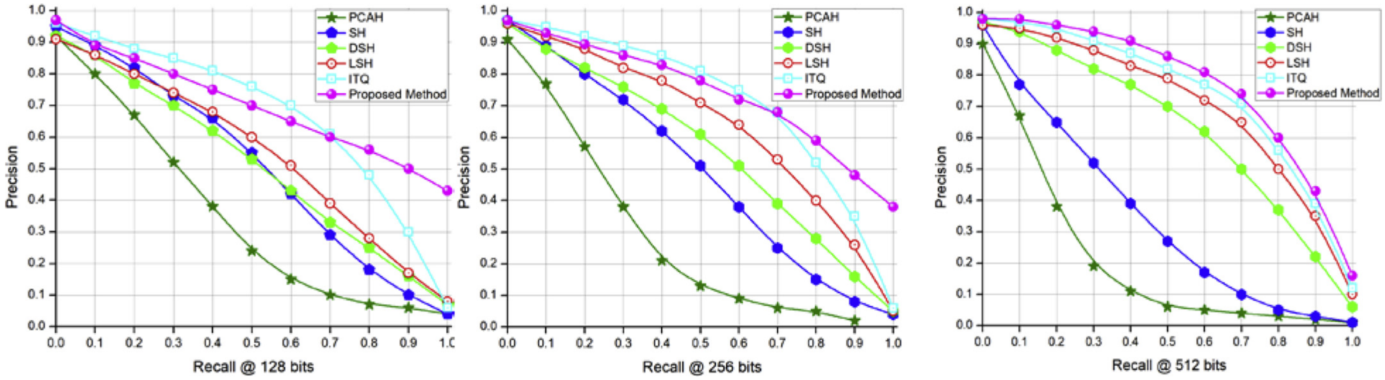

Fig. 10. Retrieval performance of the proposed method in comparison with the other state-of-the-art hash code schemes for Product image dataset.

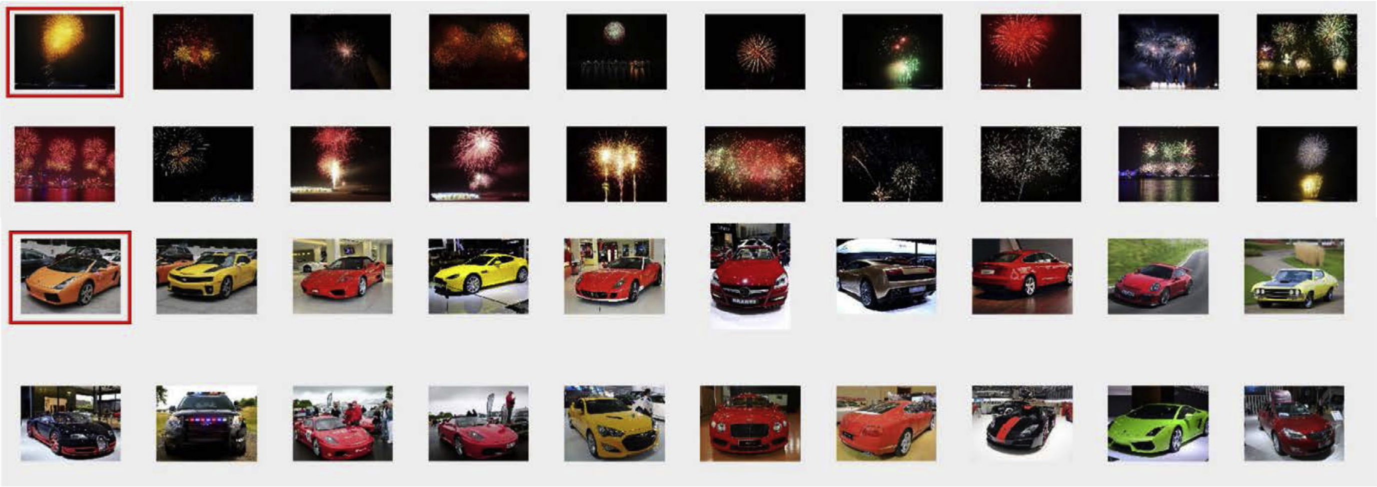

Fig. 11. Retrieval results of the proposed methods using 512-bit hash codes for GHIM-10K image dataset.

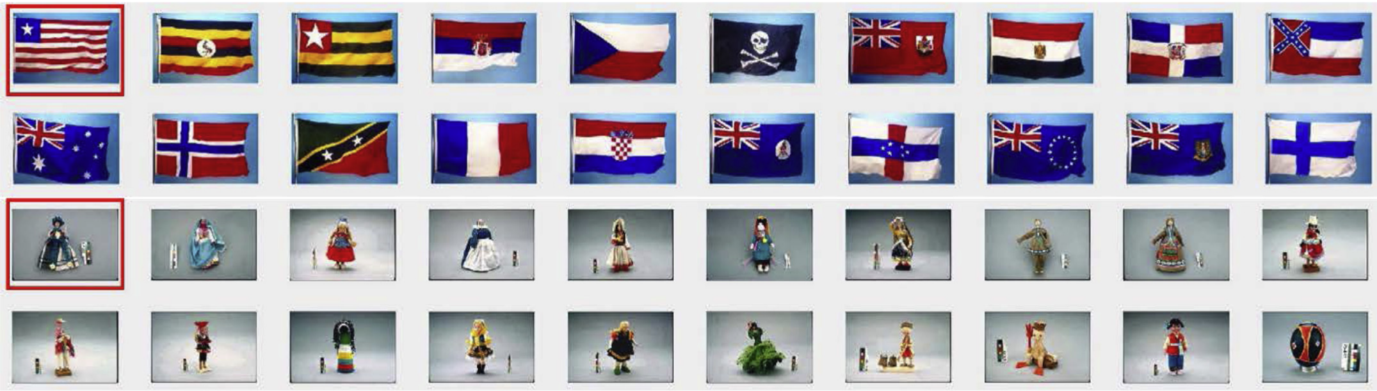

Fig. 12. Retrieval results of the proposed methods using 512-bit hash codes for Corel-10 image dataset.

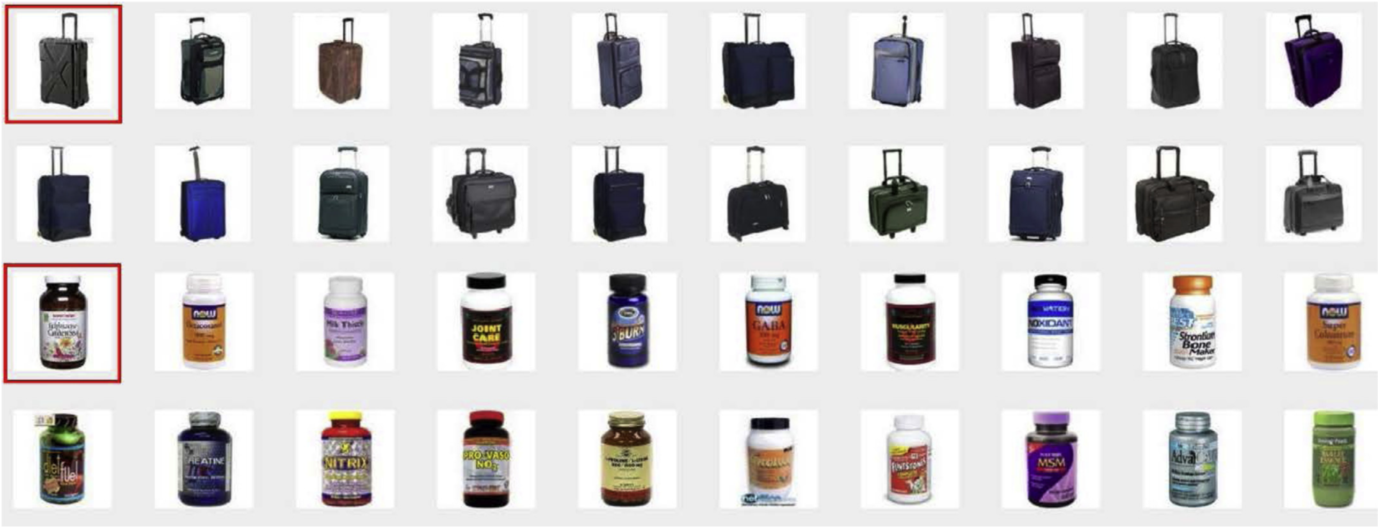

Fig. 13. Retrieval results of the proposed methods using 512-bit hash codes for Product image dataset.

be observed.

Further, we compared our method with state-of-the-art algorithms using four metrics as shown in Table 3. The metrics are standard metrics for performance evaluation of image encryption methods. The details of metrics can be found in [61]. The tabulated results indicate a better balance of the proposed method among the competing metrics.

4.2. Datasets

A wide variety of datasets exist for evaluating retrieval performance of CBIR systems. We tested our proposed method against three different benchmark datasets i.e., Corel 10K, GHIM-10K, and product image dataset. Corel-10K image dataset consists of hundred different classes having a total of 10,000 different images. This dataset contains diverse

Table 4
Processing time of the proposed method and deep features.

Method	Processing time (ms)
Original features	215 ± 4.04
512 bits	0.340 ± 0.170
256 bits	0.330 ± 0.190
128 bits	0.193 ± 0.005

Table 5
Time required for features extraction on different platforms.

Platform	Processing time (ms)
Cloud	49 ± 2.64
Smartphone	1268 ± 21.7

Algorithm 1
Initialization of secret keys.

```
Input: Sec
x = $\sum_{i=1}^{16} Sec/\sum_{i=1}^{32} Sec$,  y = $\sum_{i=17}^{32} Sec/\sum_{i=1}^{32} Sec$
  if Sec(1) < 32
    r = 1.11 + abs(x-round(x,2)) + abs(y-round(y,2))
  Elseif Sec(1 ) > = 32 And Sec (1) < 64
    r = 1.12 + abs(x-round(x,2)) + abs(y-round(y,2))
  Elseif Sec(1) > = 64 And Sec (1) < 96;
    r = 1.13 + abs(x-round(x,2)) + abs(y-round(y,2))
  Elseif Sec(1) > = 96 And Sec(1) < 128
    r = 1.14 + abs(x-round(x,2)) + abs(y-round(y,2))
  Elseif Sec(1 ) > = 128 And Sec(1) < 160
    r = 1.15 + abs(x-round(x,2)) + abs(y-round(y,2))
  Elseif Sec(1) > = 160 And Sec(1) < 192
    r = 16 + abs(x-round(x,2)) + abs(y-round(y,2))
  Elseif Sec(1) > = 192
  r = 1.17 + abs(x-round(x,2)) + abs(y-round(y,2))
  End
Output:x,y,r.
```

Algorithm 2
The pseudo keys.

```
Input: File source, and $K_{in}$
1: Sec ← $HF$ ($Source$)
2: [x,y,r] ← Algorithm 1 (Sec)
3: $Ve \leftarrow Logistic\,map\,2D(x,y,r)$
4: S1 ← $Ve(10{:}\ h + 9)$
5: S2 ← $Ve(10{:}\ w \times e + 9)$
6: S3 ← $reshape$ ($Ve$ (10:1024 + 9),32,32)
7: $[\neg, V] \leftarrow Sort(S1)[\neg, V'] \leftarrow Sort(S2)[\neg, R] \leftarrow Sort(S3)$
8: $\alpha = |\sum S1 + \sum S2 + \sum S3|$mod256
9: IF $\alpha = 0$ then
$\alpha = |\sum S1 + \sum S2 + \sum S3|$mod255
End
10: $K \leftarrow \alpha \cdot K_{in}$, $L \leftarrow K^{-1}$
Output: $V, V', R, \alpha, L, K$.
```

contents such as mountains, buildings, animals, sunset, achieves, horses, vegetables, trees, food, and flowers. Each class consists of hundred 192 × 128 images. The second dataset is GHIM-10K image dataset. This dataset is comprised of images collected from web as well as camera captured images [65]. It contains 20 categories where each category has 500 images of size 400 × 300 or 300 × 400 of JPEG format. This dataset has 10,000 overall images covering heterogeneous contents from natural environment such as sunset, flowers, mountains, cars, buildings, tiger, and fish etc. Most of the images are zoomed into size 400 × 300 or 300 × 400 from its high resolution state. The product image dataset contains 5000 images of various household items including laptops, cameras, chairs, hats, and bicycles.

In our experiments, we chose 5 random queries from each class of Corel-10K, GHIM-10K, and product dataset. Precision score is calculated for 10 recall levels for every input query. Same is the case for the number of queries conducted from other datasets. Lastly, mean precision-recall is computed to report retrieval performance. A commonly adopted method for evaluating image retrieval system is precision-recall pair. Precision is defined as the number of similar images retrieved per query divided by the total number of images retrieved. Recall can be calculated as the total number of images we want to retrieve, divided by the total number of similar images in the dataset [65]. Recall defines the robustness of the retrieval process. Precision and recall are calculated as follows:

$$Precision = \frac{Number\ of\ accurate\ images\ retrieved}{Total\ number\ of\ images\ retrieved} \tag{4}$$

$$Recall = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ number\ of\ relevant\ image\ in\ specified\ dataset} \tag{5}$$

4.3. Retrieval performance with original feature values vs. hash codes

In this section the retrieval performance is compared using deep features as well as with different length of binary codes generated by the proposed method. Fig. 7 shows the comparison of the retrieval performance of the proposed method using hash codes and original deep features. It can be noticed that all three representation of deep features provide good accuracies for initial recalls. Especially, retrieval performance achieved by 512 bit codes have shown very similar results to original deep features.

4.4. Retrieval performance on various length of hash codes

In this section, the retrieval performance of the proposed method is compared with other state-of-the-art hash encoding schemes. These schemes include LSH [11,66], SH [14], PCAH [41], DSH [16] and ITQ [44]. Experiments conducted using random queries chosen from dataset and the retrieval performance were evaluated using variable length hash codes including 128, 256, and 512 bits. Results have been reported in precision recall standard format. Fig. 8 represents the retrieval performance of the proposed method in comparison with other hashing schemes for Corel-10K image dataset. The proposed method performed better than various hashing schemes using 128 bits. However, the retrieval seems weaker for initial recalls as compared to ITQ but gets improved for later recalls. For 256 and 512 bits hash codes, the proposed method outperformed all other methods.

In GHIM-10K dataset, the proposed method significantly outperforms overall hashing schemes using 256 and 512 bit codes as given in Fig. 9. At 128 bits it performs better than PCAH, SH, DSH and LSH. However, ITQ performs better than our method for initial recalls but the accuracy of our method improves as the recall goes higher. The proposed method also outperformed all methods in product image dataset using 512 bit hash codes as given in Fig. 10. However, due to the challenging nature of the image dataset, its initial retrieval result for 256 and 512 bits seem weaker than ITQ, which improved for later recalls.

It can be concluded that the proposed method performs well compared to other methods using 512 and 256 bits hash codes. Furthermore, the proposed method generates more significant results than other competing methods as the size of the hash codes increases. It is much simpler than the other competing methods while its results are promising. The given results show that the proposed method can transforms high dimensional feature vectors to compact binary hash codes very efficiently and requires very low processing power at cloud. It is recommended to use 256 or 512 bit hash codes for efficient indexing and retrieval in large scale dataset. Furthermore, the higher length hash codes can also be obtained from early hidden layer of deep Autoencoder with same compactness and efficiency which may lead to more improved performance.

4.5. Visual results

In this experiment, visual results of randomly chosen images from different datasets and the retrieval performance of the proposed method using 512-bit hash codes have been shown in Figs. 11–13. Each dataset has queried for two random query images and the top 20 retrieved images have been shown. Visual results reveal that the proposed method is capable of retrieving highly similar images despite the huge volume and challenging nature of the consisted images in the given datasets i.e., Corel-10K, GHIM-10K and Product image dataset. In Fig. 11 top two images represent the query image from GHIM-10K dataset where the entire retrieved images belong to the same category. Fig. 12 represents results of the proposed method for Corel-10K dataset where the top ranked images for both queries belong to its right category. In the second query, the last image is irrelevant, however it has very high resemblance with the query image in terms of its visual appearance. Fig. 13 shows the retrieval performance on Product image dataset where in both cases the most relevant images have been successfully retrieved at top ranks. These results show encouraging performance of the proposed method.

4.6. Efficiency analysis

In this section the computation time of the proposed method has been evaluated using original deep features and binary hash codes. Table 4 reveals that processing 4096 dimension features vector of the FC layer took 215 ms which has been reduced significantly by the proposed method. By processing hash codes, time reduces to only 0.340 ms, while the retrieval performance remains almost same as processing original deep features. In Table 5, features extraction time on different platforms has been shown where the features extraction time on cloud is only 49 ms. However, this process took more than one second on smartphone device to perform features extractions.

5. Conclusion and future work

Due to the excessive use of smart phones, the amount of captured image data has increased significantly, needing efficient indexing and retrieval methods. Additionally, data privacy is another main concern that comes into user's mind while using cloud based services. Therefore, in this paper, we propose an approach for fast image retrieval in a variety of datasets on smart phone devices by compressing the visual features of CNN model using deep Autoencoder. The proposed method offloads the computationally extensive phase of the image retrieval process to the cloud to smoothen the image retrieval process on smart phones. Our method is evaluated on three benchmark datasets: Corel-10K, GHIM-10K and Product image dataset and results indicate that features representation using CNN reports better performance than several hand-crafted and deep features for image retrieval using mobile devices. Furthermore, our framework ensures the privacy of user's data based on a computationally efficient encryption method, providing a better balance between execution time and security level.

In future work, we plan to use probabilistic image encryption schemes [61] to further improve the security. The retrieval performance can be further improved by introducing more compact deep features in the current framework. In addition, the current system can be merged with authentication mechanisms [67–70] and can be extended to smart cities [71,72].

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIP) (No.2016R1A2B4011712).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.comcom.2018.06.001.

References

[1] P. Mell and T. Grance, The NIST Definition of Cloud Computing (v15) http://csrc.nist.gov/groups, *SNS/cloud-computing*, 2009.
[2] Scoial Network Ranked by Number of Usres, 2018.
[3] J. Ahmad, K. Muhammad, S. Bakshi, S.W. Baik, Object-oriented convolutional features for fine-grained image retrieval in large surveillance datasets, Fut. Gen. Comput. Syst. 81 (2018) 314–330.
[4] J. Ahmad, M. Sajjad, I. Mehmood, S.W. Baik, SiNC: Saliency-injected neural codes for representation and efficient retrieval of medical radiographs, PLoS One 12 (2017) e0181707.
[5] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (SURF), Comput. Vis. Image Understand. 110 (2008) 346–359.
[6] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2004) 91–110.
[7] M.-R. Ra, R. Govindan, A. Ortega, P3: toward privacy-preserving photo sharing, NSDI, (2013), pp. 515–528.
[8] V. Deshpande, L.B. Schwarz, M.J. Atallah, M. Blanton, K.B. Frikken, Outsourcing manufacturing: secure price-masking mechanisms for purchasing component parts, Product. Oper. Manag. 20 (2011) 165–180.
[9] L. Zhang, X.-Y. Li, Y. Liu, T. Jung, Verifiable private multi-party computation: ranging and ranking, INFOCOM, 2013 Proceedings IEEE, 2013, pp. 605–609.
[10] J. Wang, W. Liu, S. Kumar, S.-F. Chang, Learning to hash for indexing big data—a survey, Proc. IEEE 104 (2016) 34–57.
[11] A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing, VLDB, (1999), pp. 518–529.
[12] M. Datar, N. Immorlica, P. Indyk, V.S. Mirrokni, Locality-sensitive hashing scheme based on p-stable distributions, Proceedings of the Twentieth Annual Symposium on Computational Geometry, 2004, pp. 253–262.
[13] X. Yu, S. Zhang, B. Liu, L. Zhong, D.N. Metaxas, Large scale medical image search via unsupervised PCA hashing, Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on, 2013, pp. 393–398.
[14] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, Advances in Neural Information Processing Systems, 2009, pp. 1753–1760.
[15] J.-P. Heo, Y. Lee, J. He, S.-F. Chang, S.-E. Yoon, Spherical hashing: Binary code embedding with hyperspheres, IEEE Trans. Pattern Anal. Mach. Intell. 37 (2015) 2304–2316.
[16] Z. Jin, C. Li, Y. Lin, D. Cai, Density sensitive hashing, IEEE Trans. Cybern. 44 (2014) 1362–1371.
[17] J. Ahmad, K. Muhammad, J. Lloret, S.W. Baik, Efficient conversion of deep features to compact binary codes using fourier decomposition for multimedia big data, IEEE Trans. Ind. Inf. (2018), http://dx.doi.org/10.1109/TII.2018.2800163.
[18] J. Ahmad, K. Muhammad, S.W. Baik, Medical image retrieval with compact binary codes generated in frequency domain using highly reactive convolutional features, J. Med. Syst. 42 (December 19 2017,) 24.
[19] B. Ferreira, J. Rodrigues, J. Leitão, and H. Domingos, Towards an image encryption scheme with content-based image retrieval properties, in Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance, ed: Springer, 2015, pp. 311–318.
[20] Z.A. Balouch, M.I. Aslam, and I. Ahmed, Energy efficient image encryption algorithm, in *Innovations in Electrical Engineering and Computational Technologies (ICIEECT), 2017 International Conference on*, 2017, pp. 1–6.
[21] D.X. Song, D. Wagner, A. Perrig, Practical techniques for searches on encrypted data, Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on, 2000, pp. 44–55.
[22] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky, Searchable symmetric encryption: improved definitions and efficient constructions, J. Comput. Secur. 19 (2011) 895–934.
[23] Z. Fu, X. Sun, Q. Liu, L. Zhou, J. Shu, Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing, IEICE Trans. Commun. 98 (2015) 190–200.
[24] Z. Fu, X. Wu, C. Guan, X. Sun, K. Ren, Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement, IEEE Trans. Inf. Forensics Secur. 11 (2016) 2706–2716.
[25] H. Li, D. Liu, Y. Dai, T.H. Luan, X.S. Shen, Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage, IEEE Trans. Emerg. Topics Comput. 3 (2015) 127–138.
[26] Y. Yang, M. Ma, Conjunctive keyword search with designated tester and timing enabled proxy re-encryption function for e-health clouds, IEEE Trans. Inf. Forensics Secur. 11 (2016) 746–759.
[27] Y. Zheng, B. Jeon, D. Xu, Q. Wu, H. Zhang, Image segmentation by generalized hierarchical fuzzy C-means algorithm, J. Intell. Fuzzy Syst. 28 (2015) 961–973.
[28] C. Wang, K. Ren, S. Yu, K.M.R. Urs, Achieving usable and privacy-assured similarity search over outsourced cloud data, INFOCOM, 2012 Proceedings IEEE, 2012, pp. 451–459.
[29] Z. Xia, Y. Zhu, X. Sun, L. Chen, Secure semantic expansion based search over encrypted cloud data supporting similarity ranking, J. Cloud Comput. 3 (2014) 8.
[30] Z. Xia, X. Wang, X. Sun, Q. Wang, A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data, IEEE Trans. Parallel Distrib. Syst. 27

(2016) 340–352.

[31] S. Kamara, C. Papamanthou, Parallel and dynamic searchable symmetric encryption, International Conference on Financial Cryptography and Data Security, 2013, pp. 258–274.

[32] J. Shashank, P. Kowshik, K. Srinathan, C. Jawahar, Private content based image retrieval, Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, 2008, pp. 1–8.

[33] Z. Qin, J. Yan, K. Ren, C.W. Chen, C. Wang, Towards efficient privacy-preserving image feature extraction in cloud computing, Proceedings of the 22nd ACM International Conference on Multimedia, 2014, pp. 497–506.

[34] Q. Wang, J. Wang, S. Hu, Q. Zou, K. Ren, Sechog: privacy-preserving outsourcing computation of histogram of oriented gradients in the cloud, Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, 2016, pp. 257–268.

[35] W. Lu, A. Swaminathan, A.L. Varna, M. Wu, Enabling search over encrypted multimedia databases, Media Forensics and Security, 2009.

[36] W. Lu, A.L. Varna, A. Swaminathan, M. Wu, Secure image retrieval through feature protection, Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on, 2009, pp. 1533–1536.

[37] C.-Y. Hsu, C.-S. Lu, S.-C. Pei, Secure and robust SIFT, Proceedings of the 17th ACM International Conference on Multimedia, 2009, pp. 637–640.

[38] H. Cheng, X. Zhang, J. Yu, F. Li, Markov process-based retrieval for encrypted JPEG images, EURASIP J. Inf. Secur. 2016 (2016) 1.

[39] B. Gu, V.S. Sheng, K.Y. Tay, W. Romano, S. Li, Incremental support vector learning for ordinal regression, IEEE Trans. Neural Netw. Learn. Syst. 26 (2015) 1403–1416.

[40] Z. Xia, R. Lv, Y. Zhu, P. Ji, H. Sun, Y.-Q. Shi, Fingerprint liveness detection using gradient-based texture features, Signal, Image Video Process. 11 (2017) 381–388.

[41] X.-J. Wang, L. Zhang, F. Jing, W.-Y. Ma, Annosearch: Image auto-annotation by search, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, 2006, pp. 1483–1490.

[42] B. Kulis, K. Grauman, Kernelized locality-sensitive hashing, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 1092–1104.

[43] C. Ma, Y. Gu, W. Liu, J. Yang, X. He, Unsupervised video hashing by exploiting spatio-temporal feature, International Conference on Neural Information Processing, 2016, pp. 511–518.

[44] Y. Gong, S. Lazebnik, A. Gordo, F. Perronnin, Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 2916–2929.

[45] F. Yu, S. Kumar, Y. Gong, S.-F. Chang, Circulant binary embedding, International Conference on Machine Learning, 2014, pp. 946–954.

[46] T. Zhang, C. Du, J. Wang, Composite quantization for approximate nearest neighbor search, ICML, (2014), pp. 838–846.

[47] R. Hamza, K. Muhammad, Z. Lv, F. Titouna, Secure video summarization framework for personalized wireless capsule endoscopy, Pervasive Mob. Comput. 41 (2017) 436–450.

[48] R. Hamza, F. Titouna, A novel sensitive image encryption algorithm based on the Zaslavsky chaotic map, Inf. Secur. J. 25 (2016) 162–179 /12/01 2016.

[49] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, Image Vision Comput. 22 (2004) 761–767.

[50] Z. Guo, L. Zhang, D. Zhang, A completed modeling of local binary pattern operator for texture classification, IEEE Trans. Image Process. 19 (2010) 1657–1663.

[51] R. Hu, J. Collomosse, A performance evaluation of gradient field hog descriptor for sketch based image retrieval, Comput. Vision Image Understand. 117 (2013) 790–806.

[52] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: an efficient alternative to SIFT or SURF, Computer Vision (ICCV), 2011 IEEE International Conference on, 2011, pp. 2564–2571.

[53] J. Wan, D. Wang, S.C.H. Hoi, P. Wu, J. Zhu, Y. Zhang, et al., Deep learning for content-based image retrieval: a comprehensive study, Proceedings of the 22nd ACM International Conference on Multimedia, 2014, pp. 157–166.

[54] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, 2014.

[55] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems, (2012), pp. 1097–1105.

[56] K. Muhammad, M. Sajjad, S.W. Baik, Dual-level security based cyclic18 steganographic method and its application for secure transmission of keyframes during wireless capsule endoscopy, J. Med. Syst. 40 (2016) 114.

[57] R. Hamza, K. Muhammad, A. Nachiappan, G.R. González, Hash based Encryption for keyframes of diagnostic hysteroscopy, IEEE Access (2017), http://dx.doi.org/10.1109/ACCESS.2017.2762405.

[58] Y. Zhou, L. Bao, C.P. Chen, A new 1D chaotic system for image encryption, Signal Process. 97 (2014) 172–182.

[59] A. Belazi, M. Khan, A.A.A. El-Latif, S. Belghith, Efficient cryptosystem approaches: S-boxes and permutation–substitution-based encryption, Nonlinear Dyn. (2016) 1–25.

[60] W. Yao, F. Wu, X. Zhang, Z. Zheng, Z. Wang, W. Wang, et al., A fast color image encryption algorithm using 4-pixel feistel structure, PLoS One 11 (2016) e0165937.

[61] K. Muhammad, R. Hamza, J. Ahmad, J. Lloret, H.H.G. Wang, S.W. Baik, Secure Surveillance Framework for IoT systems using probabilistic image encryption, IEEE Trans. Ind. Inf. (2018), http://dx.doi.org/10.1109/TII.2018.2791944.

[62] A. Belazi, A.A.A. El-Latif, S. Belghith, A novel image encryption scheme based on substitution-permutation network and chaos, Signal Process. 128 (2016) 155–170.

[63] L. Xu, Z. Li, J. Li, W. Hua, A novel bit-level image encryption algorithm based on chaotic maps, Opt. Lasers Eng. 78 (2016) 17–25.

[64] X. Huang, Image encryption algorithm using chaotic Chebyshev generator, Nonlinear Dyn. 67 (2012) 2411–2417.

[65] G.-H. Liu, J.-Y. Yang, Z. Li, Content-based image retrieval using computational visual attention model, Pattern Recognit. 48 (2015) 2554–2566.

[66] M. Slaney, M. Casey, Locality-sensitive hashing for finding nearest neighbors [lecture notes], IEEE Signal Process. Mag. 25 (2008) 128–131.

[67] M. Sajjad, S. Khan, T. Hussain, K. Muhammad, A.K. Sangaiah, A. Castiglione, et al., CNN-based anti-spoofing two-tier multi-factor authentication system, Pattern Recognit. Lett. (2018), http://dx.doi.org/10.1016/j.patrec.2018.02.015.

[68] K. Muhammad, M. Sajjad, I. Mehmood, S. Rho, S.W. Baik, Image steganography using uncorrelated color space and its application for security of visual contents in online social networks, Fut. Gen. Comput. Syst. 86C (2018) 951–960, http://dx.doi.org/10.1016/j.future.2016.11.029.

[69] K. Mahmood, S.A. Chaudhry, H. Naqvi, S. Kumari, X. Li, A.K. Sangaiah, An elliptic curve cryptography based lightweight authentication scheme for smart grid communication, Fut. Gen. Comput. Syst. 81 (2018) 557–565.

[70] X. Li, F. Wu, S. Kumari, L. Xu, A.K. Sangaiah, K.-K.R. Choo, A provably secure and anonymous message authentication scheme for smart grids, J. Parallel Distrib. Comput. (2017).

[71] M. Sajjad, M. Nasir, K. Muhammad, S. Khan, Z. Jan, A.K. Sangaiah, et al., Raspberry Pi assisted face recognition framework for enhanced law-enforcement services in smart cities, Fut. Gen. Comput. Syst. (2017), http://dx.doi.org/10.1016/j.future.2017.11.013.

[72] S. Kumar, D. Datta, S.K. Singh, A.K. Sangaiah, An intelligent decision computing paradigm for crowd monitoring in the smart city, J. Parallel Distrib. Comput. (2017).