Secure Representation of Images Using Multi-layer Compression

Sohrab Ferdowsi[1], Svyatoslav Voloshynovskiy[1], Dimche Kostadinov[1], Marcin Korytkowski[2], and Rafał Scherer[2]

[1]University of Geneva, Department of Computer Science,
Battle Bât. A, 7 route de Drize, 1227 Carouge, Switzerland
[2]Institute of Computational Intelligence, Częstochowa University of Technology
Al. Armii Krajowej 36, 42-200 Częstochowa, Poland

Abstract. We analyze the privacy preservation capabilities of a previously introduced multi-stage image representation framework where blocks of images with similar statistics are decomposed into different codebooks (dictionaries). There it was shown that at very low rate regimes, the method is capable of compressing images that come from the same family with results superior to those of the JPEG2000 codec. We consider two different elements to be added to the discussed approach to achieve a joint compression-encryption framework. The first visual scrambling is the random projections were the random matrix is kept secret between the encryption and decryption sides. We show that for the second approach, scrambling in the DCT domain, we can even slightly increase the compression performance of the multi-layer approach while making it safe against de-scrambling attacks. The experiments were carried out on the *ExtendedYaleB* database of facial images.

Keywords: image compression, image scrambling, dictionary learning, rate-distortion theory, privacy preservation

1 INTRODUCTION

Representation of visual data is a fundamental problem in many areas of artificial intelligence. In general, one seeks a representation which is as concise as possible in terms of memory storage, as fast as possible in terms of computation and as precise as possible in terms of fidelity to the original data.

Moreover, in many applications like medical imaging, biometric data and multimedia management, the security and privacy preservation issues are also among the major factors that should be seriously taken into account. In these applications, the image data could be of sensitive nature that should not be revealed. As an assumption, one can consider that a set of images and their encoded versions, also possibly some information about their encoding algorithm are available in a public domain. Therefore, a safe and privacy preserving representation scheme should impede an attacker who knows the general structure of the representation method being used and aims at reconstructing the original

images from their representation by increasing the computational cost and the necessary data resources needed for the attack.

In this work, we consider the framework introduced in [1] for image representation and in particular image compression when the images have a similar source. We investigate the privacy preserving capabilities of this approach and add visual scrambling elements to make the representation more secure.

The paper is organized as follows. In section 2 we review the multi-layer image representation framework and in section 2.1 we analyze the security issues with this framework. In section 3 we introduce two different methods to increase the privacy preserving capabilities of the discussed framework and analyze their behavior. The first method is based on random projections and the second method scrambles the images in their DCT domain. Section 4 discusses the experimental setup and the results. We conclude the paper in section 5.

2 Multi-layer Image Representation

A framework to represent images into multi layers of decomposition was recently introduced in [1]. The method is based on qunatizing the image patches in the direct gray-scale domain into several codeworks (atoms) for the first stage. In the next stages, the quantization residual is quantized to a new set of codewords.

Fig. 1 shows this idea. $\mathbf{X} \in \mathcal{R}^n$ is the random vector representing the vectorized patches of images and $\mathbf{x}(j)$ is its j^{th} realization. $\hat{\mathbf{x}}_1(j), \cdots \hat{\mathbf{x}}_L(j)$ are the results of quantization of the j^{th} patch at each stage and $\hat{\mathbf{x}}(j) = \hat{\mathbf{x}}_1(w_1(j)) + \cdots + \hat{\mathbf{x}}_L(w_L(j))$ is the final estimation of $\mathbf{x}(j)$. An index vector $\mathbf{w}(j) = [w_1(j), \cdots, w_L(j)]^T$ will be used to represent the estimation of the j^{th} patch.

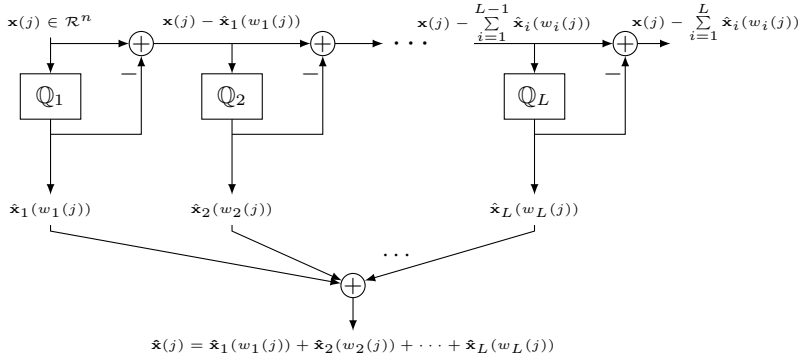

Fig. 1: Multi-layer quantization of a vectorized image patch $\mathbf{x}(j)$ to the vector $\hat{\mathbf{x}}(j)$. $\mathbb{Q}_i(\cdot)$, $1 \leq i \leq L$ is the quantizer of the i^{th} layer and $w_i(j)$ is the the corresponding index at that layer. Quantizer $\mathbb{Q}_i(\cdot)$ has 2^{nR_i} codewords, where R_i is the rate of compression at i^{th} stage.

Fig. 2 illustrates the compression stages to be carried out in this representation scheme. The decompression procedure is simply the inverse of compression.

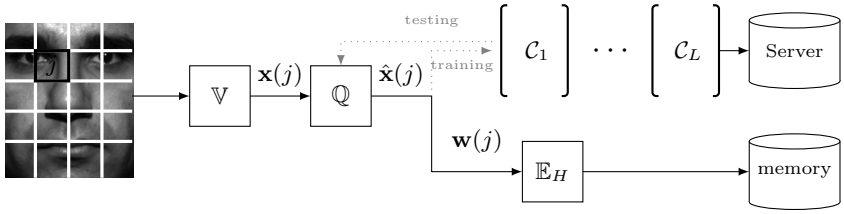

Fig. 2: Compression of an image from its blocks. $\mathbb{V}$ is the vectorizing operator, $\mathbb{Q}$ is the multi-layer quantizer and $\mathbb{E}_H$ is the entropy encoding. The codewords in each of the codebooks $\mathcal{C}_1, \cdots \mathcal{C}_L$ will be kept on a server while the encoded indices will be stored in memory as the representation of images.

In this decomposition, each codeword of $\mathcal{C}_i$, $\hat{\mathbf{X}}_i$ has the same dimension as $\mathbf{X}$. Each codebook $\mathcal{C}_i$ contains 2^{nR_i} codewords, or equivalently the corresponding indices have an alphabet of $|\mathbf{W}_i| = 2^{nR_i}$ in each layer. Therefore, the equivalent alphabet size of the corresponding indices of the final estimation, $\hat{\mathbf{X}}$ is upper bounded as

$$|\hat{\mathbf{W}}| \leq 2^{nR_1} \times \cdots \times 2^{nR_L} = 2^{n(R_1 + \cdots R_L)}.$$

This means that, by this multi-layer decomposition, the image patch could be represented by an equivalent dictionary with a maximum number of atoms equals to $2^{nR_c} = 2^{2(R_1 + \cdots R_L)}$.

This is very appealing since the equivalent alphabet size is growing exponentially with the number of layers. If a one-layer structure was to be used, like in k-means based VQ, we would need to store all these atoms in memory which would practically be infeasible, while in fact, in the multi-layer structure, we only store $2^{nR_1} + \cdots + 2^{nR_L}$ atoms. Moreover, the search complexity and also the sample complexity, the amount of training data required to achieve a certain performance in a machine learning setup will also be reduced the same as memory.

2.1 Security Analysis

We assume that an attacker has access to a number of training data from the same family of images, along with their respective representations (indices). To analyze the security of this scheme, one should ask the question, given the representation of a probe image, how can the attacker reconstruct the content of the compressed image? In particular, we are interested in estimating the amount of training data needed for the attacker to do the reconstruction and also the computational complexity of this attack procedure.

We imagine that the attacker has a database $\mathcal{X} = [\mathbf{x}(1)^T, \cdots \mathbf{x}(M)^T]$ of training data which could be similar to the database used to train the codebooks, where $\mathbf{x}(j)^T \in \mathcal{R}^n$ is a column vector of an image patch. For any $\mathbf{x}(j), 1 \leq j \leq M$, the attacker has also a set of indices $\mathbf{w}_1(j), \cdots \mathbf{w}_L(j)$, where $\mathbf{w}_i(j)$ is a column vector consisting of k_i elements with only one element equal to one(active) and the rest equal to zero and where k_i is the number of codewords in an unknown codebook $\mathcal{C}_i$ with $1 \leq i \leq L$.

Therefore, the attacker can write the system of equations as below.

$$\begin{aligned} \mathbf{x}(1)^T &= \mathcal{C}_1\mathbf{w}_1(1) + \cdots + \mathcal{C}_L\mathbf{w}_L(1) \\ \mathbf{x}(2)^T &= \mathcal{C}_1\mathbf{w}_1(2) + \cdots + \mathcal{C}_L\mathbf{w}_L(2) \\ &\vdots \\ \mathbf{x}(M)^T &= \mathcal{C}_1\mathbf{w}_1(M) + \cdots + \mathcal{C}_L\mathbf{w}_L(M) \end{aligned} \tag{1}$$

This can be written in a matrix form as:

$$\mathcal{X}^T = \mathcal{C}W \tag{2}$$

Where $\mathcal{X}$ is defined as above and $\mathcal{C}_{n\times K}$ is the concatenation of all codebooks with $K = \sum_{i=1}^{L} k_i$ and $W_{K\times M}$ is the concatenation of all $\mathbf{w}_j(i)$'s corresponding to one $\mathbf{x}(i)$ in its $i^{\mathbf{th}}$ column.

The pseudo-inverse of W does not exist since its sparsity pattern imposes the rank of WW^T to be not complete for the multi-layer case. Therefore, this equation cannot be solved to derive the value of unknown $\mathcal{C}$.

However, instead of directly solving equation (2), the attacker can estimate the values of $\mathcal{C}_j$ sequentially, starting from the first layer and then continuing to the next layers. For example, one can gather all $\mathbf{x}(j)$'s that have the same $\mathbf{w}_1$ and sum them up. As long as the number of $\mathbf{x}$'s is enough, assuming that $\mathcal{C}_i$'s are zero mean, an estimation of the corresponding codeword of $\mathcal{C}_1$ will be derived. Having estimated the codewords of C_1, then other codebooks could also be estimated accordingly.

Therefore, one can conclude that this data representation scheme is not safe against attacks. In the next section, we investigate two different modifications to this approach to boost its privacy preservation capabilities.

3 Privacy Preserving Multi-layer Image Representation

In this section we consider the problem of joint compression-scrambling. In particular, we search for an image representation which is as compact as possible, while it should be able to cope with the distortions introduced by the compression and also not prone to security attacks. In section 3.1 we investigate the idea of Random Projections as an element to be added to the previous setup. Then in 3.2 we consider randomizing the images in the DCT domain.

Fig. 3 sketches the general idea of joint compression-scrambling.

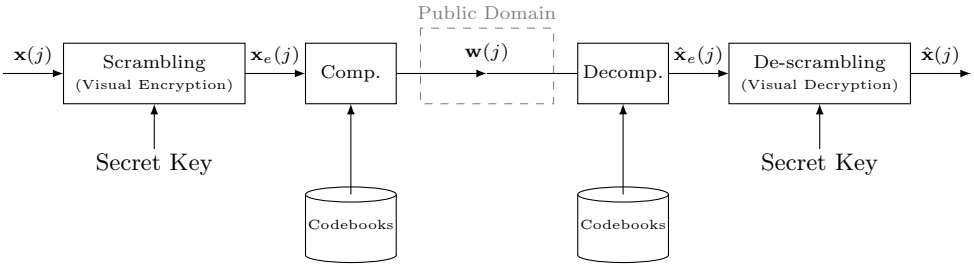

Fig. 3: The joint compression-scrambling scenario where the image data are scrambled(visually encrypted) using a secret key and then compressed. The image representations(indices), along with the details of the compression algorithm may be available to public.

3.1 Random Projections based scrambling

Random projections are extensively used in dimensionality reduction due to their capabilities to preserve pairwise distances of data points [2]. Although we are not interested here in dimensionality reduction, random projections are appealing for us. Because, while keeping the essentials of the original data, we can have a randomization in their structure and thus more security.

Fig. 4 shows the structure of the system with random projections. $A_{b\times b}$ is a random matrix with elements independently drawn from an identical zero-mean Gaussian distribution and then orthogonalized, where b is the size of square patches from images. $x(i)$ is an image patch in the form of a square matrix before vectorization.

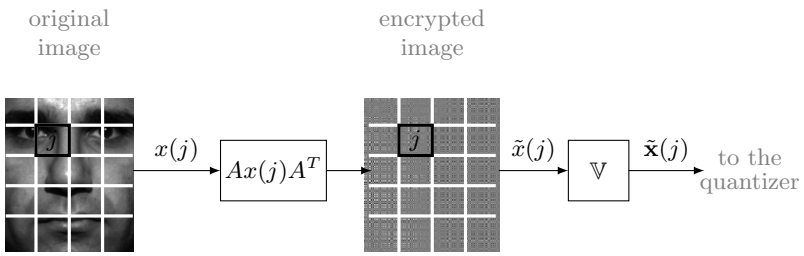

Fig. 4: Encryption of the images with random projections. $\mathbb{V}$ is the vectorizing operator.

After the compression and decompression stages, the de-scrambling of the data is simply the reverse operations of those in Fig. 4. However, having the

random matrix A as a secret key between the encryption and decryption sides and not disclosed to the public domain, the attacker cannot practically reconstruct the data given their representation and a set of training images. The reason is that in the construction of $\tilde{x}(j) = Ax(j)A^T$, the random matrix A is multiplied by the image patch from both sides. Therefore, even in the case where only one stage of quantization is used where equation (2) can be solved in the non-encrypted case, the attacker cannot infer anything from the structure of the codebooks and indices, even by having access to an increasing number of training data.

A drawback of this encryption method, however, is a reduction in compression performance. Random projections are shown to decrease correlation in data [3]. In fact, the entropy of the transformed coefficients is increased. As a result, more rate is required to achieve the same distortion. Therefore, the performance of quantizers will be decreased, especially in the initial stages where the images are considerably correlated in nature. In section 4, we show experimentally how this method is decreasing the compression performance.

3.2 DCT Domain Scrambling

The randomization can also be performed in the DCT domain. Fig. 5 shows the proposed DCT-based encryption, the main contribution of this paper.

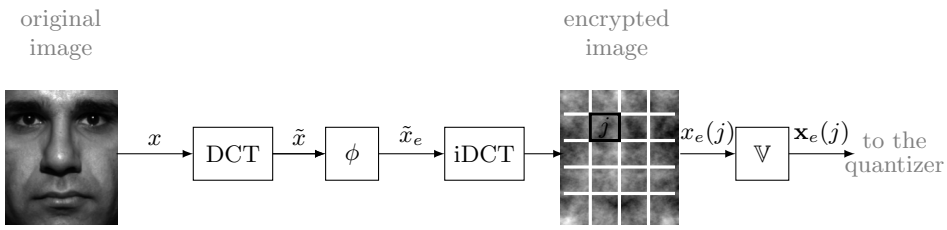

Fig. 5: Scrambling of the images in the DCT domain. ϕ is the randomizer in the DCT domain and $\mathbb{V}$ is the vectorizing operator.

After computing the DCT coefficients of an image, a random matrix $B_{b\times b}$ with values drawn equi-probably from a binary alphabet $\{\pm 1\}$ will be multiplied element-wise (Hadamard product) with the DCT coefficients of that image and then converted back to the pixel value domain. The matrix $B_{b\times b}$ could be generated for each block specifically, or it can be generated universally for all blocks.

We can write:

$$\begin{aligned} \tilde{x} &= MxM^T \\ \tilde{x}_e &= \phi(\tilde{x}) = \tilde{x} \circ B \\ x_e &= M^T \tilde{x}_e M, \end{aligned} \tag{3}$$

where M is the orthogonal DCT matrix.

As in the random projection based encryption, the attacker cannot reconstruct the images as long as the random matrix B is kept secret, or, even if only one universal matrix is used for all blocks, he has to guess its values among the 2^n possible values with $n = b^2$, which makes it computationally intractable.

It should be pointed out that in this way of scrambling, the magnitude information of images is revealed. However, as can be seen in Fig. 5, the scrambled images do not have any meaningful visual resemblance to the original images since in the images, the phase information is very important rather than the magnitude information. The reader is refered to [4] for more explanations on this method of scrambling. The DCT-based scrambling is also closely related to the DFT scrambling based on phase randomization [5].

An important advantage of this encryption method is that the correlation among the data is not reduced here as much as in the random projection based encryption. In fact, entropy of the scrambled data in this case is the same as the original data, since changing the sign of the DCT coefficients does not increase their entropy. Therefore, the performance of quantization is expected to be superior. Moreover, as will be shown in section 4, the blocking artifacts present in the previous methods are completely removed since every image block $x(j)$ contains information about the whole image that it comes from and hence causes a smoothness, although the blocks will be processed independently afterwards.

4 Experimental Results

In this section we experimentally study the proposed methods and validate them on the *CroppedYaleB* [6], a database of facial images with varying lighting conditions. We randomly chose 1600 images for training the codebooks and tested on another 400 images from the same database.

Fig. 6 shows the average Peak Signal-to-Noise Ratio (PSNR) for different methods versus their corresponding Bits Per Pixel (BPP) values. $L = 20$ layers were used in the methods with $k_i = 256, 128, 32, 16$ codewords for the first, second, third and forth quarters of the 20 layers, respectively.

As can be seen from the curves, the joint compression-scrambling in the DCT domain does not decrease the quality of compression of the method discussed in [1]. It should be noted, however, that the quality of compression increases less rapidly with the increase of rate compared to the JPEG2000 codec. The reason is, in the current simple experimental setup, many optimizations are neglected. For example, the number of codewords in each codebook, the design of different codebooks for different rate regimes, variable length code allocation for different

patches based on their variance are among the immediate points to be considered in future versions of this family of methods.

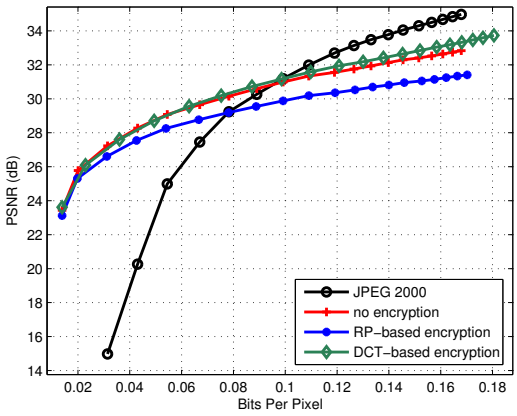

Fig. 6: PSNR vs. BPP, averaged over 400 test images.

Fig. 7 shows the results of compression of these methods for two different rates.

It is interesting to point out that the blocking artifacts are removed in the DCT-based scrambling. Fig. 8 compares the rate-distortion behavior of the methods on the test data. It is verified here that the DCT-based scrambling, unlike its random projection counterpart, does not decrease the rate-distortion performance.

5 Conclusions

In this paper, we addressed the problem of joint compression-scrambling for images. Specifically, in a previous work, we were considering the case were the images to be compressed come from the same family and thus have similar statistics. In this case, unlike conventional image compression methods which try to capture redundancy in a given image by considering local image properties, we tried to learn these patterns from an ensemble of images. In this work, we focused on the encryption of this representation and considered two remedies for the problem of security and privacy preservation. We first discussed random projections to make the representation less vulnerable to attacks. We then considered the scrambling in the DCT domain and showed that, without any decrease in the performance, we can have a framework for the joint compression and scrambling of images.

Fig. 7: Visual comparison of different methods for two different BPP values.

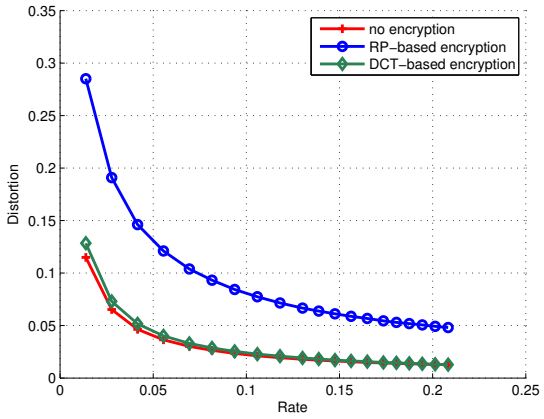

Fig. 8: Rate-Distortion curve, averaged over the normalized data from 400 images. The rate here can also be interpreted as bits per pixel since their values are the same.

Acknowledgments

The research has been partially supported by a grant from Switzerland through the Swiss Contribution to the enlarged European Union PSPB-125/2010.

References

1. Ferdowsi, S., Voloshynovskiy, S., Kostadinov, D.: Sparse Multi-Layer image approximation: Image compression. In: European Signal Processing Conference 2015 (EUSIPCO 2015)(submitted to), Nice, France (August 2015)
2. Johnson, W.B., Lindenstrauss, J.: Extensions of lipschitz mappings into a hilbert space. conference in modern analysis and probability (new haven, conn., 1982), 189–206. Contemp. Math **26** (1984)
3. Farhadzadeh, F., Voloshynovskiy, S., Koval, O.J.: Performance analysis of content-based identification using constrained list-based decoding. IEEE Transactions on Information Forensics and Security **7**(5) (2012) 1652–1667
4. Diephuis, M., Voloshynovskiy, S., Koval, O., Beekhof, F.: Robust message-privacy preserving image copy detection for cloud-based systems. In: CBMI 2012,10th Workshop on Content-Based Multimedia Indexing. (2012)
5. Grytskiv, Z., Voloshynovskiy, S., Rytsar, Y.: Cryptography and steganography of video information in modern communications. Facta Universitatis **11**(1) (1998) 115–125
6. Lee, K., Ho, J., Kriegman, D.: Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans. Pattern Anal. Mach. Intelligence **27**(5) (2005) 684–698