SECURE OMP BASED PATTERN RECOGNITION THAT SUPPORTS IMAGE COMPRESSION

Takayuki Nakachi[1] and Hitoshi Kiya[2]

[1]Nippon Telegraph and Telephone Corporation,Kanagawa, Japan

[2]Tokyo Metropolitan University, Tokyo, Japan

ABSTRACT

In this paper, we propose a secure Orthogonal Matching Pursuit (OMP) based pattern recognition scheme that well supports image compression. The secure OMP is a sparse coding algorithm that chooses atoms sequentially and calculates sparse coefficients from encrypted images. The encryption is carried out by using a random unitary transform. The proposed scheme offers two prominent features. 1) It is capable of pattern recognition that works in the encrypted image domain. Even if data leaks, privacy can be maintained because data remains encrypted. 2) It realizes Encryption-then-Compression (EtC) systems, where image encryption is conducted prior to compression. The pattern recognition can be carried out using a few sparse coefficients. On the basis of the pattern recognition results, the scheme can compress selected images with high quality by estimating a sufficient number of sparse coefficients. We use the INRIA dataset to demonstrate its performance in detecting humans in images. The proposal is shown to realize human detection with encrypted images and efficiently compress the images selected in the image recognition stage.

KEYWORDS

Surveillance Camera, Pattern Recognition, Secure Computation, Sparse Coding, Random Unitary Transform

1. INTRODUCTION

With the increase in threats and criminal activity, security is seen as a major public concern. Image/video surveillance is one approach to addressing this issue. Many image/video surveillance systems are now widely deployed in many public spaces such as airports, banks, shopping streets, and public streets, and they are recording huge amounts of image/video every day. Fortunately, edge/cloud computing offers an efficient way of handling and analyzing the huge amounts of image/video data. However, edge/cloud computing poses some serious issues for end users, such as unauthorized use, data leaks, and privacy failures due to the unreliability of providers and accidents [1].

Many studies have examined the processing of encrypted data; most proposals use homomorphic encryption (HE) and secure multiparty computation (MPC) [2]-[7]. Even though service providers cannot directly access the native content of encrypted signals, they can still apply HE and MPC. In particular, fully homomorphic encryption (FHE) allows arbitrary computation on encrypted data [6][7]. However, these methods impose high communication costs, high computational complexity, or large cipher text sizes, so further advances are needed for attractive applications such as big data analysis and advanced image/video processing. We take a random unitary transform approach, which has much lower communication, lower computational complexity, and a smaller ciphertext size than either FHE or MPC has. Secure computation methods based on the random unitary transform have been reported for biometric template protection [8]-[11].

DOI: 10.5121/sipij.2020.11202

We proposed secure sparse coding based on the random unitary transform for pattern recognition [12]-[15], and Encryption-then-Compression (EtC) methods [16]-[18]. Early work on sparse coding was based on the efficient coding hypothesis, which states that the goal of visual coding is to faithfully reproduce visual input while minimizing neural effort [20][21]. It effectively represents observed signals as the linear combination of a small number of atoms. Sparse dictionary learning has been successfully applied to various image/video and audio processing applications [22]-[30]. The effectiveness of sparse coding has been reported for pattern recognition/image classification [24]-[26] and image compression [27]-[30]. For example, the experiments of Ref. [30] show that rate-distortion based sparse coding outperforms JPEG and JPEG2000 by up to 6+ dB and 2+ dB, respectively.

In this paper, we propose a secure pattern recognition scheme that extends the previously proposed EtC methods. The secure pattern recognition methods and EtC systems mentioned above were proposed separately. The current proposal offers not only image pattern recognition but also image compression. The integrated system is realized by performing pattern recognition in the compressed signal domain. 1) It is capable of efficient pattern recognition in the encrypted image domain. Even if data leaks, privacy is maintained because the data remains encrypted. 2) It works as an EtC system. Pattern recognition and image compression can be carried out seamlessly in the same compressed signal domain. In the proposed scheme, Orthogonal Matching Pursuit (OMP) [19] is executed in the compressed signal domain. OMP is a sparse coding algorithm that chooses the atoms of sparse coding sequentially and calculates the sparse coefficients. Pattern recognition employs a few sparse coefficients. On the basis of the pattern recognition results, additional enhancement atoms are chosen and used to compress the selected images. Finally, we employ the INRIA person dataset [31] to evaluate the human detection performance of the proposed method. Detecting humans in images is essential for not only image/video surveillance but also many applications such as automatic driver assistance.

The organization of this paper is as follows. In Sec. 2, we explain related work. Section 3 describes sparse coding for image patches. In Sec. 4, we propose secure OMP based pattern recognition that supports image compression. Section 5 shows experimental results. Conclusions and future work are given in Sec. 6.

2. Related Work

In this section, we review the conventional secure pattern recognition methods and EtC systems.

2.1. Secure Pattern Recognition

We previously proposed secure sparse coding for pattern recognition [12]-[15]. Feeding encrypted images into the secure OMP computation yields sparse coefficients used for pattern recognition. We verified that by adopting the random unitary transform, the pattern recognition performance is not degraded, which proves that this proposed framework operates securely with no performance degradation. Furthermore, compared with deep-learning based methods such as Stacked PCA network (SPCANet) [32], our sparse coding based method has several prominent advantages such as 1) low computational complexity and less data needed for training and 2) transparent machine learning; the algorithm is interpretable as the optimization problem is written in closed form. Refs. [12]-[15] detail the experiments and results.

2.2. Encryption-then-Compression (EtC) Systems

Encryption-then-Compression (EtC) systems [16]-[18] [33]-[35] have been proposed to securely transmit and compress images through an untrusted channel provider; the traditional technique is

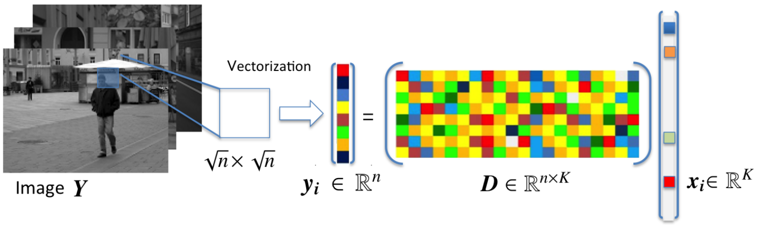

Vectorization
=
$\sqrt{n}\times\sqrt{n}$
Image $\boldsymbol{Y}$
$\boldsymbol{y_i} \in \mathbb{R}^n$
$\boldsymbol{D} \in \mathbb{R}^{n\times K}$
$\boldsymbol{x_i} \in \mathbb{R}^K$

Figure 1: Sparse coding for image patches.

to use Compression-then-Encryption (CtE) systems. EtC systems allow us to close non-encrypted images to SNS providers because encrypted images can be directly compressed even when the images are recompressed by SNS providers multiple times. Well-known EtC systems are block scrambling-based encryption schemes that are compatible with international standards, e.g., JPEG and JPEG2000 [33]-[35]. While sparse coding based EtC systems [16]-[18] are not compatible with international compression standards, their coding performance is high because they form dictionaries that fit the observed signals.

3. Sparse Coding for Image Patches

In this section, we overview sparse coding for image patches, which is the basis of secure pattern recognition and EtC systems.

3.1. Basic Formulation

We consider image patches of size $\sqrt{n} \times \sqrt{n}$ pixels that are ordered lexicographically as column vectors $\boldsymbol{y_i} = \{y_1, , ..., y_n\}^T \in \mathbb{R}^n$. The patches are extracted from image $\boldsymbol{Y}$ as shown in Fig. 1. We assume that every image patch $\boldsymbol{y_i}$ can be represented sparsely given the over-complete dictionary $\boldsymbol{D} = \{\boldsymbol{d}_1, ..., \boldsymbol{d}_K\} \in \mathbb{R}^{n\times K}$ whose columns contain K prototype atoms $\boldsymbol{d}_i$:

$$\boldsymbol{y_i} = \boldsymbol{D}\boldsymbol{x_i}, \tag{1}$$

where $\boldsymbol{x_i} = \{x_1, ..., x_K\}^T \in \mathbb{R}^K$ are sparse coefficients, $i = 1, \cdots, N$, and N is the total number of patches. In advance, the dictionary $\boldsymbol{D}$ is designed for the images by training algorithms such as MOD [36] and K-SVD [37].

If $n < K$ and $\boldsymbol{D}$ is a full-rank matrix, an infinite number of solutions to the representation problem are available. The solution with the least number of nonzero coefficients is certainly an appealing representation. This sparsest representation is the solution given by

$$(P_0) \quad \min_{\boldsymbol{x}_i} \|\boldsymbol{x}_i\|_0 \quad \text{subject to} \quad \boldsymbol{y}_i = \boldsymbol{D}\boldsymbol{x}_i, \tag{2}$$

where $\|\cdot\|_0$ is the l_0-norm, counting the nonzero entries of the vector. Extraction of the sparsest representation is, however, an NP-hard problem [38].

3.2. Selection of Dictionary Atoms

Dictionary atoms are typically estimated by a "pursuit algorithm" that finds the following approximate solution:

$$\boldsymbol{x}_i = arg\,\underset{\boldsymbol{x}_i}{min} \left\|\boldsymbol{y}_i - \boldsymbol{D}\boldsymbol{x}_i\right\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{x}_i\|_0 < \epsilon_i. \tag{3}$$

We assume that the dictionary $\boldsymbol{D}$ is fixed. Well-known pursuit algorithms include Orthogonal Matching Pursuit (OMP) [19]. OMP is a greedy, step-wise regression algorithm. At each stage, OMP chooses the dictionary atom having the maximal projection onto the residual signal. After each selection, the representation coefficients w.r.t. the atoms selected so far are found via least-squares search.

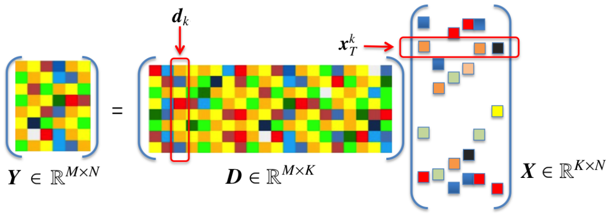

d_k
x_T^k
=
$Y \in \mathbb{R}^{M\times N}$
$D \in \mathbb{R}^{M\times K}$
$X \in \mathbb{R}^{K\times N}$

Figure 2: One atom $\boldsymbol{d}_k$ and corresponding sparse coefficient vector $\boldsymbol{x}_T^k$.

3.3. Dictionary Learning

An over-complete dictionary $\boldsymbol{D}$ is designed by adapting its content to fit a given set of images. Given the set $\boldsymbol{Y}$=$\{\boldsymbol{y}_i\}_{i=1}^N$, we assume that there exists a dictionary, $\boldsymbol{D}$, that can recreate the given images via sparse combinations. The overall mean square error of a representation is given by

$$E = \|\boldsymbol{Y} - \boldsymbol{DX}\|_2^2 . \tag{4}$$

Method of Optimal Direction (MOD) [36] and K-Singular Value Decomposition (K-SVD) [37] are well-known dictionary learning algorithms. Assuming that $\boldsymbol{X}$=$\{\boldsymbol{x}_i\}_{i=1}^N$ is fixed, the MOD algorithm allows us to seek an update to $\boldsymbol{D}$ such that the above error is minimized. Taking the derivative of (4) with respect to $\boldsymbol{D}$ yields

$$\boldsymbol{D} = \underset{\boldsymbol{D}}{arg\ min}\,\|\boldsymbol{Y} - \boldsymbol{DX}\|_F^2 = \boldsymbol{YX}^T(\boldsymbol{XX}^T)^{-1}. \tag{5}$$

K-SVD is an iterative method that uses singular value decomposition; it alternates between sparse coding based on the current dictionary and the process of updating the dictionary atoms to better fit the data. It has been shown to perform very well for image processing tasks. Here, we use K-SVD for pattern recognition and image compression because of its ability to extract the features of image data. Unlike MOD, K-SVD updates atoms sequentially. Figure 2 shows the k-th atom $\boldsymbol{d}_k$ and the corresponding sparse coefficient vector $\boldsymbol{x}_T^k$. For each atom $\boldsymbol{d}_k$ ($k = 1, 2, \cdots, K$ in $\boldsymbol{D}$), update it with the following steps.

1) Compute the overall representation error matrix $\boldsymbol{E}_k$ with

$$\boldsymbol{E}_k = \boldsymbol{Y} - \sum_{j\neq k}^{K} \boldsymbol{d}_j \boldsymbol{x}_T^j. \tag{6}$$

2) Define the group of indexes that satisfy:

$$\omega_k = \{i \mid 1 \leq i \leq K,\ \ \boldsymbol{x}_T^k(i) \neq 0\}. \tag{7}$$

Define $\boldsymbol{\Omega}_k$ as a matrix of size $N \times |\omega_k|$ with ones on the $(\omega_k(i), i)$th entries and zeros elsewhere. Multiplication $\boldsymbol{E}_k^R = \boldsymbol{E}_k\boldsymbol{\Omega}_k$ creates a matrix that includes a selection of error columns that use the atom $\boldsymbol{d}_k$.

3) Apply Singular Value Decomposition (SVD) to $\boldsymbol{E}_k^R$:

$$\boldsymbol{E}_k^R = \boldsymbol{U\Delta V^T} = \sum_{i=1}^{n} \boldsymbol{u}_i \cdot \sigma_i \boldsymbol{v}_i^T . \tag{8}$$

Choose the updated dictionary atom $\boldsymbol{d}_k$ to be the first column $\boldsymbol{u}_1$. Update coefficient vector $\boldsymbol{x}_R^k$ to be the first column multiplied by the first eigenvalue $\sigma_1 \boldsymbol{v}_1^T$.

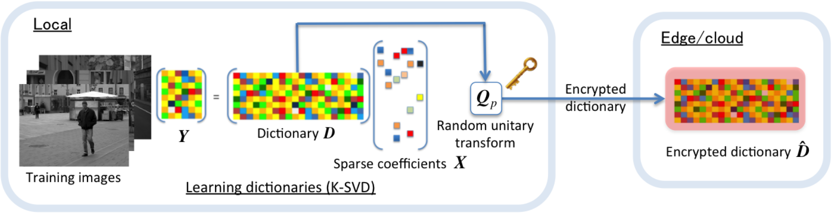
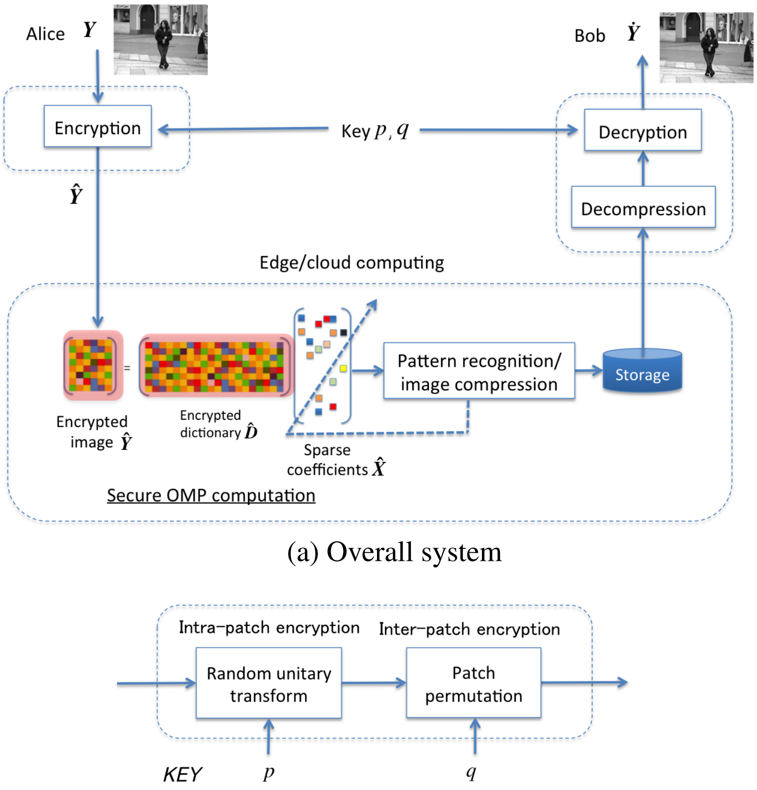


Figure 3: Architecture of training process.



(a) Overall system

(b) Encryption process

Figure 4: Architecture of running process: pattern recognition that supports image compression.

4. Secure OMP Based Pattern Recognition That Supports Image Compression

In this section, we propose a secure pattern recognition method that offers image compression as an integrated component. The integrated system is realized by performing pattern recognition in the compressed signal domain.

4.1. Secure Computation Architecture

Figure 3 shows the training step. The dictionary $\boldsymbol{D}$ is designed by the K-SVD algorithm at the local site. Feeding the training images to the learning algorithm yields the dictionary $\boldsymbol{D}$. Next, we apply random unitary transform function $T(\cdot)$ to the dictionary $\boldsymbol{D}$ to generate an encrypted dictionary $\hat{\boldsymbol{D}}$. The encrypted dictionary $\hat{\boldsymbol{D}}$ is sent to the appropriate edge/cloud site and stored in a database.

Figure 4 illustrates the running step. Figure shows (a) the overall system and (b) its encryption process. The local site applies the same random unitary transform function $T(\cdot)$ to test image

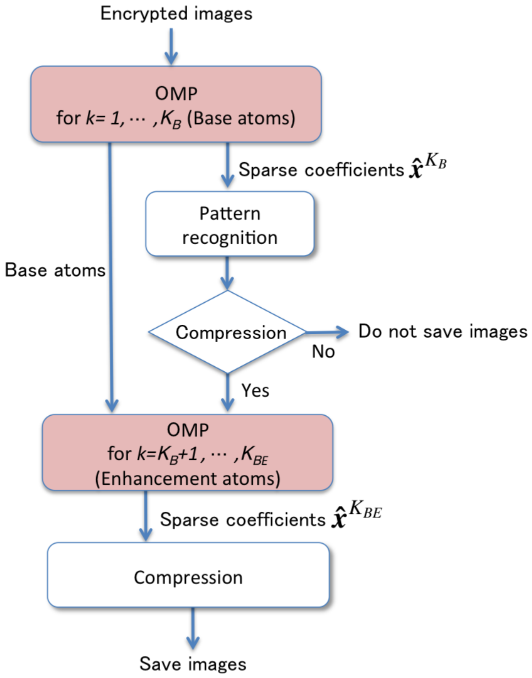

Encrypted images
OMP
for $k= 1, \cdots ,K_B$ (Base atoms)
Sparse coefficients $\hat{\boldsymbol{x}}^{K_B}$
Pattern recognition
Base atoms
Compression
Do not save images
No
Yes
OMP
for $k=K_B+1, \cdots ,K_{BE}$
(Enhancement atoms)
Sparse coefficients $\hat{\boldsymbol{x}}^{K_{BE}}$
Compression
Save images

Figure 5: Flow chart of secure OMP computation.

$\boldsymbol{Y}$ to generate encrypted image $\hat{\boldsymbol{Y}}$. Then, encrypted image $\hat{\boldsymbol{Y}}$ is sent to the edge/cloud site. The edge/cloud site uses encrypted image $\hat{\boldsymbol{Y}}$ and the encrypted dictionary $\hat{\boldsymbol{D}}$ to perform secure OMP computation. Secure OMP chooses the atoms sequentially and calculates the sparse coefficients $\hat{\boldsymbol{X}}$ from the encrypted $\hat{\boldsymbol{Y}}$ and $\hat{\boldsymbol{D}}$.

Figure 5 shows a flow chart of secure OMP computation. First, secure OMP chooses a few base atoms and estimates a corresponding set of a few sparse coefficients $\hat{\boldsymbol{x}}^{K_B}$ for pattern recognition. The pattern recognition is carried out in the compressed signal domain using a set of the sparse coefficients $\hat{\boldsymbol{x}}^{K_B}$. Then, on the basis of the pattern recognition results, only the selected images are compressed. For example, if a user wants to store images of humans, image compression is performed only for those images. Secure OMP chooses additional enhancement atoms in order to store images in high quality. Then, a set of sparse coefficients $\hat{\boldsymbol{x}}^{K_{BE}}$ corresponding to both the base and the enhancement atoms are calculated. $\hat{\boldsymbol{x}}^{K_{BE}}$ is used for compression.

4.2. Random Unitary Transform

The encrypted images and dictionary are generated by using the random unitary transform approach. A vector $\boldsymbol{f}_i$ $(i = 1, \cdots , L) \in \mathbb{R}^N$ is encrypted by a random unitary matrix $\boldsymbol{Q}_p \in \mathbb{C}^{N\times N}$ with a private key p:

$$\hat{\boldsymbol{f}}_i \quad = T(\boldsymbol{f}_i, p) = \boldsymbol{Q}_p \boldsymbol{f}_i, \tag{9}$$

where $\hat{\boldsymbol{f}}_i$ is an encrypted vector; L is the number of vectors. Note that the random unitary matrix $\boldsymbol{Q}_p$ satisfies

$$\boldsymbol{Q}_p^* \boldsymbol{Q}_p = \boldsymbol{I}, \tag{10}$$

where $[\cdot]^*$ and $\boldsymbol{I}$ mean the Hermitian transpose operation and the identity matrix, respectively. In addition to unitarity, $\boldsymbol{Q}_p$ must have randomness for generating the encrypted signal. Gram-Schmidt orthogonalization is a typical method for generating $\boldsymbol{Q}_p$. Security analyses on using the random unitary matrix have been considered in terms of brute-force attack, diversity, and irreversibility [8]-[10]. Furthermore, the encrypted vector has the following properties.

· Property 1: Conservation of Euclidean distances.

$$\|\boldsymbol{f}_i - \boldsymbol{f}_j\|_2^2 = \|\hat{\boldsymbol{f}}_i - \hat{\boldsymbol{f}}_j\|_2^2 \tag{11}$$

· Property 2: Norm isometry.

$$\|\hat{\boldsymbol{f}}_i\|_2^2 = \|\boldsymbol{f}_i\|_2^2 \tag{12}$$

· Property 3: Conservation of inner products.

$$\boldsymbol{f}_i^* \boldsymbol{f}_j = \hat{\boldsymbol{f}}_i^* \hat{\boldsymbol{f}}_j \tag{13}$$

The proposed secure sparse coding computation generates encrypted signal $\hat{\boldsymbol{y}}_i$ and dictionary $\hat{\boldsymbol{D}}$ with two transforms:

$$\hat{\boldsymbol{y}}_i \quad = T(\boldsymbol{y}_i, p) = \boldsymbol{Q}_p \boldsymbol{y}_i \tag{14}$$

$$\hat{\boldsymbol{D}} \quad = T(\boldsymbol{D}, p) = \boldsymbol{Q}_p \boldsymbol{D}. \tag{15}$$

Then, the encrypted image patches $\hat{\boldsymbol{y}}_i$ $(i = 1, 2, \cdots, N)$ are randomly permuted using a random integer generated by a private key q. Finally, the permuted patches are combined to form an encrypted image $\hat{\boldsymbol{Y}}$, which is fed to the OMP computation.

4.3. Secure OMP Computation

The sparse coefficient $\hat{\boldsymbol{x}}_i$ is estimated for each image patch $\hat{\boldsymbol{y}}_i$. Instead of Eq. (3), we consider the following optimization problem in which $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{D}}$ are assumed to be given:

$$\hat{\boldsymbol{x}}_i = \underset{\boldsymbol{x}}{arg\ min} \parallel \hat{\boldsymbol{y}}_i - \hat{\boldsymbol{D}}\boldsymbol{x}_i \parallel_2^2 \quad subject\ to \quad \|\boldsymbol{x}_i\|_0 < \epsilon. \tag{16}$$

The sparse coefficient $\hat{\boldsymbol{x}}_i$ yielded by secure OMP computation is the same result as that created by the non-encrypted version [16]. The algorithm is shown below (prefix i of $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{y}}_i$ is omitted for notation simplicity):

Secure OMP computation for pattern recognition that supports image compression

Initialization: $k = 0$, and set

· The initial solution $\boldsymbol{x}^0 = \boldsymbol{0}$

· The initial residual $\hat{\boldsymbol{r}}^0 = \hat{\boldsymbol{y}} - \hat{\boldsymbol{D}}\boldsymbol{x}^0 = \hat{\boldsymbol{y}}$

· The initial solution supports $\boldsymbol{S}^0 = \emptyset$.

Main Iteration:

Increment k by 1 and perform the following steps:

· **S**weep: Compute the errors

$$\hat{\epsilon}(i) \quad = \quad \|\hat{\boldsymbol{r}}^{k-1}\|_2^2 - \frac{(\hat{\boldsymbol{d}}_i \cdot \hat{\boldsymbol{r}}^{k-1})^2}{\|\hat{\boldsymbol{d}}_i\|_2^2}. \tag{17}$$

Here, we define an atom $\hat{\boldsymbol{d}}_i$ as

$$\hat{\boldsymbol{d}}_i = \hat{\boldsymbol{D}}\delta_i, \tag{18}$$

where $\boldsymbol{\delta}_i = [(0, \cdots, 0, \delta(i), 0, \cdots, 0)]^T$ has all elements equal to 0 except one (i.e., the i-th element is 1).
· **U**pdate Support: Find the minimizer

$$i_0 = arg \underset{i \notin \mathbf{S}^{k-1}}{min} \{\hat{\epsilon}(i)\}, \boldsymbol{S}^k = \boldsymbol{S}^{k-1} \cup \{i_0\},$$

where $\boldsymbol{S}^k$ is a support that is the set of indexes corresponding to non-zero elements of the sparse coefficient vector $\boldsymbol{x}$ at the k-th iteration.
· **U**pdate Provisional Solution: compute

$$\hat{\boldsymbol{x}}^k = \{(\hat{\boldsymbol{D}}_{\boldsymbol{S}^k})^T \hat{\boldsymbol{D}}_{\boldsymbol{S}^k}\}^{-1}\{(\hat{\boldsymbol{D}}_{\boldsymbol{S}^k})^T \hat{\boldsymbol{y}}\}, \tag{19}$$

where $\hat{\boldsymbol{D}}_{\boldsymbol{S}^k}$ is a submatrix of $\hat{\boldsymbol{D}}$ consisting of the columns $\hat{\boldsymbol{d}}_i$ with $i \in \boldsymbol{S}^k$, and $\hat{\boldsymbol{x}}^k$ is the set of columns of $\boldsymbol{x}$ corresponding to the support $\boldsymbol{S}^k$.
· **U**pdate Residual: compute

$$\hat{\boldsymbol{r}}^k = \hat{\boldsymbol{y}} - \hat{\boldsymbol{D}}_{\boldsymbol{S}^k} \hat{\boldsymbol{x}}^k. \tag{20}$$

· **S**topping Rule: For the pattern recognition,

$$k = K_B, \tag{21}$$

where K_B is the number of a few specified atoms, e.g., $K_B = 1$ or 2. Iteration is repeated until the number of chosen atoms reaches K_B. Then, by using a set of the estimated sparse coefficients $\hat{\boldsymbol{x}}^k = \hat{\boldsymbol{x}}^{K_B}$, the pattern recognition is performed with the processing steps described in Sec. 4.4.

On the basis of the pattern recognition results, further iteration is needed for image compression of the selected image. For image compression,

$$k = K_{BE}, \tag{22}$$

where $K_{BE}(> K_B)$ is the sufficient number of atoms that can be used to store images in high quality. K_{BE} is directly specified by the user. A set of the sparse coefficients $\hat{\boldsymbol{x}}^k = \hat{\boldsymbol{x}}^{K_{BE}}$ is used for image compression. An alternative stopping rule is that if

$$||\hat{\boldsymbol{r}}^k||_2 < \epsilon, \tag{23}$$

stop. Here, ϵ is a threshold specified by the user. The corresponding set of the sparse coefficients $\hat{\boldsymbol{x}}^k = \hat{\boldsymbol{x}}^{K_{BE}}$ is used for image compression. $k = K_{BE}$ is indirectly controlled by ϵ, which is generally a different value from Eq. (22). Until satisfaction is achieved, commence another iteration.

Output: The proposed solution $\hat{\boldsymbol{x}}$ is obtained after k iterations.

4.4. Pattern Recognition

The pattern recognition consists of two steps: feature extraction and classification. The secure OMP algorithm chooses atoms sequentially and calculates the corresponding sparse coefficients for each image patch. We use just a set of the few sparse coefficients $\hat{\boldsymbol{x}}^{K_B}$ (calculated using only $k = 1$ *or* 2 iterations) for pattern recognition and classification.

1. Feature Extraction
   Figure 6 shows the procedure of feature extraction and classification. The sparse coefficients for each image patch are used for formatting the feature vector. To reduce the

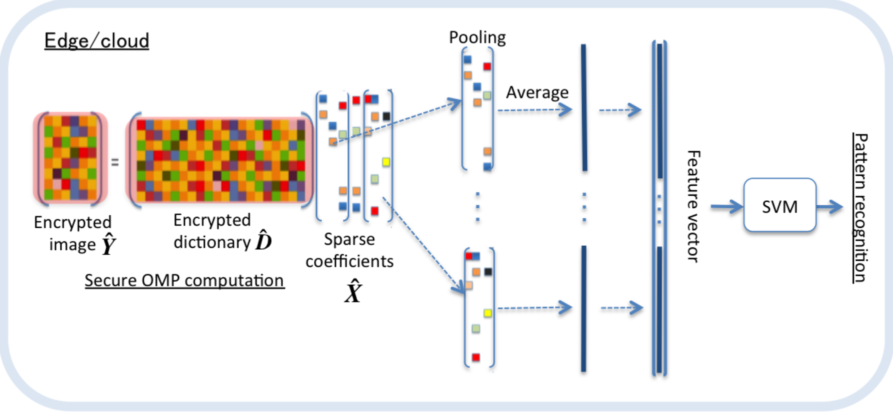


Figure 6: Pattern recognition: feature extraction and classification.

dimension of the feature, we take the statistics of the spatially local sparse coefficients as the feature, which corresponds to local spatial pooling. Multiple sparse coefficients $\hat{\boldsymbol{x}}_i^{K_B}$, which correspond to local $B \times B$ image patch $\boldsymbol{y}_i$, are grouped into the averaged sparse coefficient $\bar{\boldsymbol{x}}_j(j =, 1, 2, \cdots, N/B^2)$, where B is block size. The averaged sparse coefficients $\bar{\boldsymbol{x}}_j$ are vectorized to produce a feature vector $\vec{\boldsymbol{x}}$.

2. Classification
Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification or regression tasks, but it is mostly used for the former. In SVM, we input the feature vector $\vec{\boldsymbol{x}}$ to the discriminant function as

$$(\vec{\boldsymbol{x}}) = sign(\omega^T \vec{\boldsymbol{x}} + b) \tag{24}$$

with

$$sign(u) = \begin{cases} 1(u > 1) \\ -1(u \leq 1), \end{cases} \tag{25}$$

where ω is a weight parameter, and b is a bias. SVM also has a technique called the kernel trick, which is a function that takes a low dimensional input space and transforms it into a higher dimensional space. This can be used for non-linear classification. For the pattern recognition task, classification is performed using a linear SVM. The SVM is trained using task data from training subjects.

4.5. Image Compression

Feeding the encrypted dictionary $\hat{\boldsymbol{D}}$ and the encrypted image $\hat{\boldsymbol{Y}}$ into the secure OMP computation yields a set of the sparse coefficients $\hat{\boldsymbol{x}}_i^{K_{BE}}$ for each image patch $\boldsymbol{y}_i$. Then, quantization is applied to a set of the sparse coefficients $\hat{\boldsymbol{x}}_i^{K_{BE}}$ and entropy encoded. The rate-distortion tradeoff between the compression ratio and decoded image quality of each image patch can be controlled by altering the number of atoms K_{BE} or the threshold ϵ_i without decoding the encrypted image. Rate-distortion control can be done gracefully by adding atoms sequentially. To keep the image quality of each image patch, the same threshold is set: ϵ_i = constant $(i = 1, \cdots, N)$.

The decompression and decryption processes are the reverse processes of compression and encryption. The decoded image $\dot{\boldsymbol{y}}_i$ for each image patch can be obtained by $\dot{\boldsymbol{y}}_i = \boldsymbol{Q}_p^* \hat{\boldsymbol{D}} \hat{\boldsymbol{x}}_i^{K_{BE}}$. Only the authorized user can decrypt the encrypted images.

5. Experimental Results

We carried out experiments on detecting humans in images from the INRIA person dataset [31]. Here, we assume that we compress only those that include human(s) captured by surveillance systems.

(a) Human images

(b) Non-human images

Figure 7: Several samples of 480 × 640-pixel human and non-human images in INRIA person datasets [31].

5.1. INRIA Person Dataset and Parameters

The INRIA person dataset is one of the most popular and widely used pedestrian detection benchmark datasets. It contains images of various sizes with and without humans. We evaluated the performance of the proposed method by challenging it with 480 × 640-pixel human and non-human images. Several samples of the INRIA person dataset are shown in Fig. 7. The upper rows are human images, and the lower rows are non-human ones. The parameter settings are:

1. Designing K-SVD: We applied K-SVD and trained a dictionary of size 64 × 256. The training data consisted of a set of image patches of size 8 × 8 pixels, randomly taken from 20 human images.
2. Creating the random unitary transform: We generated a 64 × 64 random unitary transform by using the Gram-Schmidt orthogonalization method.
3. Designing and running the SVM: Block size B=20 for local pooling of the sparse coefficients. For the human detection task, two-class classification was performed using a linear SVM. In the training step, the SVM was trained using 100 images (50 human images and 50 non-human images).

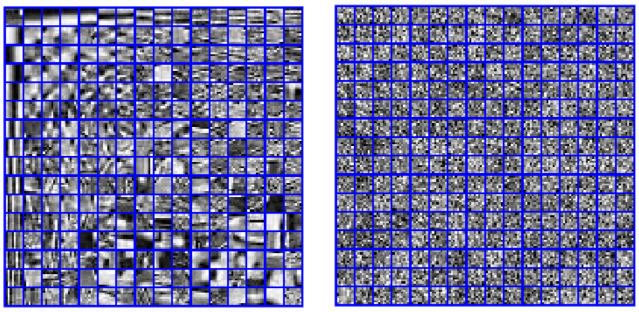
In the evaluation, we used 10-fold cross-validation. One-hundred images were partitioned into 10 sub-samples (a single sub-sample contained 5 human and 5 non-human images). Of the 10 sub-samples, a single sub-sample was retained as the validation data for testing, and the remaining 9 subsamples were used as training data. The cross-validation process was then repeated 10 times, with each of the 10 subsamples used exactly once as the validation data. The 10 results were then averaged to produce a single estimate.

5.2. Results

The trained dictionary $\boldsymbol{D}$ and corresponding encrypted dictionary $\hat{\boldsymbol{D}}$ are shown in Fig. 8. Figures 9 and 10 show the original $\boldsymbol{Y}$ and corresponding encrypted images $\hat{\boldsymbol{Y}}$ for a sample of human and non-human images, respectively. It can be seen that the encrypted dictionary and the encrypted images provided no visible information. Feeding the encrypted dictionary $\hat{\boldsymbol{D}}$ and the encrypted images $\hat{\boldsymbol{Y}}$, the secure OMP computation was performed.

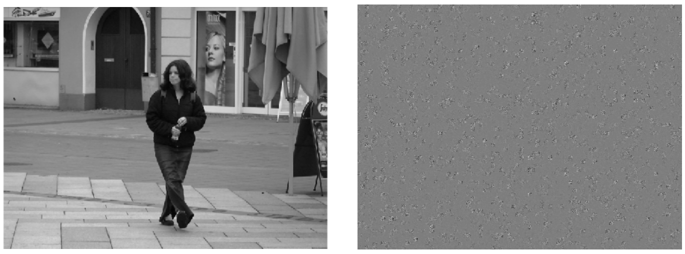
A. Pattern Recognition

The detection rate of the proposed privacy-preserving pattern recognition method is shown in Table 1. We evaluated two cases: the number of atoms $K_B = 1$ and $K_B = 5$. The detection rate is



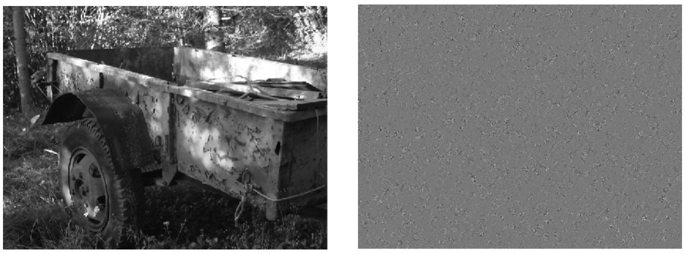
(a) Dictionary $\boldsymbol{D}$ (b) Encrypted dictionary $\hat{\boldsymbol{D}}$

Figure 8: Trained dictionary and corresponding encrypted dictionary for human images.



(a) Original $\boldsymbol{Y}$ (b) Encrypted image $\hat{\boldsymbol{Y}}$

Figure 9: Sample of original and encrypted human images.



(a) Original $\boldsymbol{Y}$ (b) Encrypted image $\hat{\boldsymbol{Y}}$

Figure 10: Sample of original and encrypted non-human images.

calculated by

$$Detection\ rate = \frac{Number\ of\ images\ correctly\ detected}{Total\ number\ of\ test\ images}. \tag{26}$$

Table 1 shows that the proposed method achieved a detection rate of around 80 [%]. Note that the results were obtained from encrypted images. Setting the number of atoms to $K_B = 1$ or $K_B = 5$ yielded almost the same performance. This means that a small number of sparse coefficients was enough for pattern recognition. Figures 11 and 12 show feature vectors (reshaped to matrix forms) for the human image of Fig. 9 and the non-human image of Fig. 10, respectively. These figures show that the feature vectors of the human image were more sparse than that of the non-human image. Regarding the difference in the number of atoms ($K_B = 1$ and $K_B = 5$), the feature vector was almost the same. This also supports the assumption that a small number of sparse coefficients is sufficient for pattern recognition.

For comparison, we evaluated a pattern recognition method with the input being the non-encrypted version of OMP. The detection rate of the non-encrypted version is shown in Table 2. The 10-

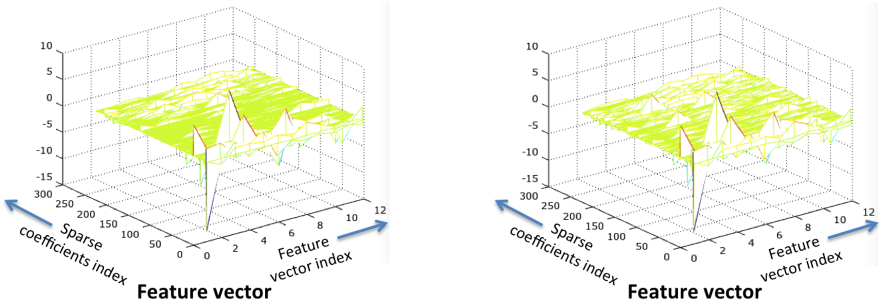
Table 1: Detection rate (DR) [%] of proposed method.

(a) Number of atoms: $K_B = 1$

Test	1	2	3	4	5	6	7	8	9	10	Ave.
DR	100	70	80	70	90	90	80	60	90	70	80

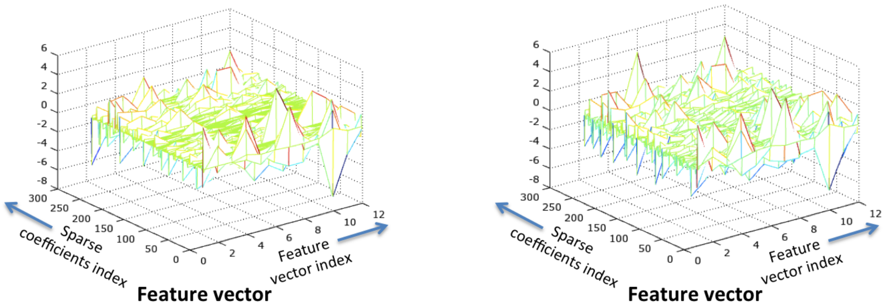
(b) Number of atoms: $K_B = 5$

Test	1	2	3	4	5	6	7	8	9	10	Ave.
DR	90	60	90	70	90	90	80	50	100	70	79



(a) Number of atoms: $K_B = 1$ (b) Number of atoms: $K_B = 5$

Figure 11: Feature vectors (reshape to matrix forms) for human image of Fig. 9.



(a) Number of atoms: $K_B = 1$ (b) Number of atoms: $K_B = 5$

Figure 12: Feature vectors (reshape to matrix forms) for non-human image of Fig. 10.

fold cross-validation used the same training and testing datasets as for the non-encrypted version of OMP and the secure OMP. The results show that the proposal had exactly the same detection performance as the non-encrypted version of the pattern recognition method.

B. Image Compression

Figure 13 plots the coding efficiency (number of atoms K_{BE} vs. decoded image quality PSNR [dB]) for the selected human images. We controlled the image quality of the human images for each patch by setting the number of atoms $K_{BE} = \{1, 2, 3, 4, 5\}$. For practical use, we set the number of atoms according to the condition $K_{BE} > K_B$. Here, K_{BE} was set without following this condition in order to evaluate the coding efficiency. This figure shows that the proposed method increased the decoded image quality by adding atoms sequentially. It can be seen that when the number of atoms was 5, high quality images were obtained. Note that there is no need to decompress and decrypt images when running the secure OMP algorithm.

Table 2: Detection rate (DR) [%] of the non-encrypted method.

(a) Number of atoms: $K_B = 1$

Test	1	2	3	4	5	6	7	8	9	10	Ave.
DR	100	70	80	70	90	90	80	60	90	70	80

(b) Number of atoms: $K_B = 5$

Test	1	2	3	4	5	6	7	8	9	10	Ave.
DR	90	60	90	70	90	90	80	50	100	70	79

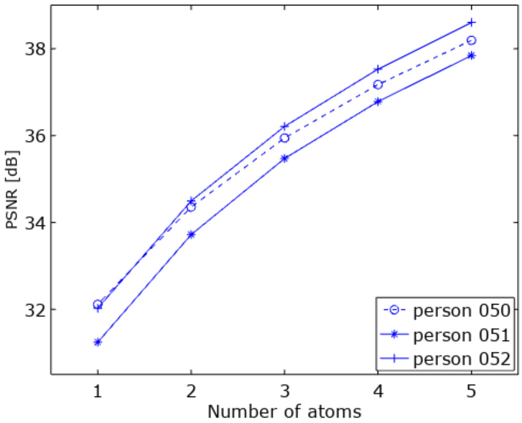


Figure 13: Coding efficiency (Number of atoms K_{BE} vs. decoded image quality).

C. Security Evaluation

Finally, we evaluated the security of secure OMP from the viewpoint of objective image quality (PSNR) and the visibility of decoded images. We considered both (a) access by an authorized user and (b) access by an unauthorized user. Tables 3 shows the decoded image quality obtained by the authorized and unauthorized users for the encrypted human image of Fig. 9. From this table, we can see that the decoded image quality obtained by the unauthorized user was very low regardless of the number of atoms K_{EB}. Figures 14 and 15 show decoded image examples obtained by the authorized and unauthorized users for the encrypted human image of Fig. 9. These results show that encrypted images cannot be decrypted by an unauthorized user.

6. Conclusion and Future Work

In this paper, we proposed an OMP based pattern recognition scheme that well supports image compression. Pattern recognition and image compression can be carried out seamlessly in the same compressed signal domain. The proposed scheme offers two prominent features. 1) It is capable of pattern recognition that works in the encrypted image domain. Even if data leaks, privacy can be maintained because data remains encrypted. 2) It also realizes EtC systems, where image encryption is conducted prior to compression. We confirmed its performance by detecting humans in the INRIA dataset. In terms of estimation accuracy for pattern recognition, these experiments are merely the first step. Further study is required to enhance the proposal's performance.

References

[1] C. T. Huang, L. Huang, Z. Qin, H. Yuan, L. Zhou, V. Varad-harajan, and C-C. J. Kuo, "Survey on securing data storage in the cloud," *APSIPA Transactions on Signal and Information Processing*, vol. 3, e7, 2014.

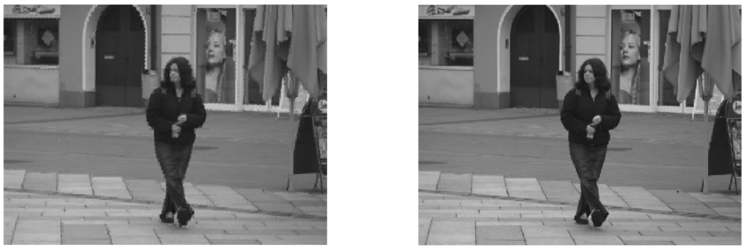
Table 3: Decoded image quality obtained by authorized and unauthorized users for encrypted human image of Fig. 9.

(a) Authorized user

Number of atoms K_{BE}	1	2	3	4	5
PSNR [dB]	31.24	33.72	35.47	36.78	37.84

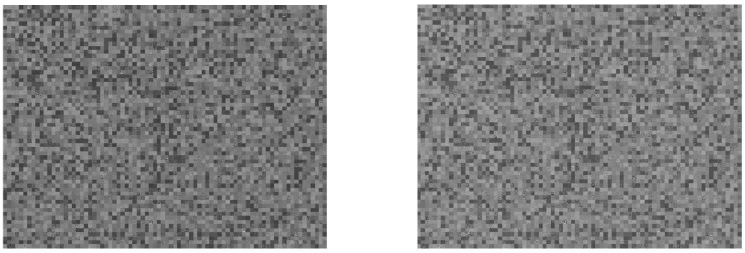
(b) Unauthorized user

Number of atoms K_{BE}	1	2	3	4	5
PSNR [dB]	13.24	13.21	13.20	13.19	13.19



(a) Number of atoms: $K_{BE} = 1$ (b) Number of atoms: $K_{BE} = 5$

Figure 14: Images decoded by authorized user for encrypted human image of Fig. 9.



(a) Number of atoms: $K_{BE} = 1$ (b) Number of atoms: $K_{BE} = 5$

Figure 15: Images decoded by non-authorized user for encrypted human image of Fig. 9.

[2] R. L. Lagendijk, Z. Erkin, and M. Barni, “Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation,” *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 82-105, Jan. 2013.

[3] T. Araki, J. Furukawa, Y. Lindell, A. Nof, and K. Ohara, “High-throughput semi-honest secure three-party computation with an honest majority, ” *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, pp. 805-817, 2016.

[4] T. Araki, A. Barak, J. Furukawa, T. Lichter, Y. Lindell, A. Nof, K. Ohara, A. Watzman, and O. Weinstein, “Optimized honest-majority mpc for malicious adversaries - breaking the 1 billion-gate per second barrier, ” *IEEE Symposium on Security and Privacy (SP)*, pp. 843-862, 2017.

[5] Y. Aono and T. Hayashi and L. Phong and L. Wang, “Privacy-preserving logistic regression with distributed data sources via homomorphic encryption, ” *IEICE Transactions on Information and Systems*, vol. E99-D, no. 8, pp. 2079-2089, 2016.

[6] W. Lu, S. Kawasaki, and J. Sakuma, “Using fully homomorphic encryption for statistical analysis of categorical, ordinal and numerical data, ” *IACR Cryptology ePrint Archive*, p. 1163, 2016.

[7] Z. Brakerski, “Fundamentals of fully homomorphic encryption - A survey,” *Electronic Colloquium on Computational Complexity*, report no. 125, 2018.

[8] W. Yongjin and K. N. Plataniotis, “Face based biometric authentication with changeable and privacy preservable templates,” *Biometrics Symposium*, pp. 1-6, 2007.

[9] I. Nakamura, Y. Tonomura, and H. Kiya, “Unitary transform-based template protection and its application to l2-norm minimization problems,” *IEICE Transactions on Information and Systems,* vol. E99-D, no.1, pp. 60-68, Jan. 2016.

[10] Y. Saito, I. Nakamura, S. Shiota, and H. Kiya, “An efficient random unitary matrix for biometric template protection,” *2016 Joint 8th International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems (ISIS)*, pp. 366-370, 2016.

[11] T. Maekawa, A. Kawamura, T. Nakachi, and H. Kiya, “Privacy-preserving support vector machine computing using random unitary transformation,” *IEICE Transactions on Fundamentals*, vol. E102-A, no. 12, pp. 1849-1855, Dec. 2019.

[12] Y. Wang, T. Nakachi, and H. Ishihara, “Edge and cloud-aided secure sparse representation for face recognition,” *27th European Signal Processing Conference (EUSIPCO 2019)*, Sept. 2019.

[13] Y. Wang and T. Nakachi, “Towards secured and transparent AI technologies in hierarchical computing networks,” *NTT Technical Review*, <https://www.ntt-review.jp/archive/2019/201909.html>, vol. 9, 2019.

[14] Y. Wang and T. Nakachi, “Secure face recognition in edge and cloud networks: From the ensemble learning perspective,” *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2020)*, May 2020.

[15] T. Nakachi, Y. Wang, and H. Kiya, “Privacy-preserving pattern recognition using encrypted sparse representations in L0 norm minimization,” *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2020)*, May 2020.

[16] T. Nakachi and H. Kiya, “Practical secure OMP computation and its application to image modeling,” *Proceedings of the 2018 International Conference on Information Hiding and Image Processing (IHIP2018)*, Sept. 2018.

[17] T. Nakachi, Y. Bandoh, and H. Kiya, “Secure dictionary learning for sparse representation,” *27th European Signal Processing Conference (EUSIPCO 2019)*, Sept. 2019.

[18] T. Nakachi and H. Kiya, “Secure sparse representations in L0 norm minimization and its application to EtC systems,” *13th International Conference on Signal Processing and Communication Systems (ICSPCS2019)*, d13, pp. 61-67, Dec. 2019.

[19] Y. C. Pati, R. Rezaiifar, Y. C. P. R. Rezaiifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition,” *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pp. 40-44, 1993.

[20] H. B. Barlow, “Possible principles underlying the transformation of sensory messages,” *Sensory Communication*, pp. 217-234, 1961.

[21] B. A. Olshausen and D. J. Field, “Emergence of simple-cell receptive-field properties by learning a sparse code for natural images, ” *Nature*, vol. 381, pp. 607-609, 1996.

[22] M. Elad, “Sparse and redundant representations: From theory to applications in signal and image processing,” *Springer*, 2010.

[23] M. Elad, “Sparse and redundant representation modeling - what next?,” *IEEE Signal Processing Letters*, vol. 19, no. 12, pp. 922-928, Dec. 2012.

[24] Z. Jiang, Z. Lin, and L. S. Davis, “Label consistent K-SVD: Learning a discriminative dictionary for recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651-2664, Nov. 2013.

[25] Q. Wang, Y. Guo, J. Guo, and X. Kong, “Synthesis K-SVD based analysis dictionary learning for pattern classification,” *Multimedia Tools and Applications*, vol. 77, pp. 17023-17041, 2018.

[26] Y. Song, Y. Liu, Q. Gao, X. Gao, F. Nie, and R. Cui, “Euler Label Consistent K-SVD for image classification and action recognition,” *Neurocomputing*, vol. 310, no. 8. pp. 277-286, 2018.

[27] K. Skretting and K. Engan, “Image compression using learned dictionaries by RLS-DLA and compared with K-SVD,” *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2011)*, pp. 1517-1520, 2011.

[28] O. Bryt and M. Elad, “Compression of facial images using the K-SVD algorithm,” *Journal of Visual Communication and Image Representation*, vol. 19, issue 4, pp. 270-282, 2008.

[29] Y. Sun, X. Tao, Y. Li, and J. Lu, “Dictionary learning for image coding based on multisample sparse representation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 11, pp. 2004-2010, Nov. 2014.

[30] X. Zhang, W. Lin, Y. Zhang, S. Wang, S. Ma, L. Duan, and W. Gao, “Rate-distortion optimized sparse coding with ordered dictionary for image set compression,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 12, pp. 3387-3397, Dec. 2018.

[31] “INRIA Person Dataset,” http://pascal.inrialpes.fr/data/human/.

[32] L. Tian, C. Fan, Y. Ming, and Y. Jin, “Stacked PCA network (SPCANet): An effective deep learning for face recognition,” *2015 IEEE International Conference on Digital Signal Processing (DSP)*, pp. 1039-1043, July 2015.

[33] K. Kurihara, M. Kikuchi, S. Imaizumi, S. Shiota, and H. Kiya, “An Encryption-then-Compression system for JPEG/Motion JPEG standard,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E98-A, no. 11, pp. 2238-2245, 2015.

[34] W. Sirichotedumrong and H. Kiya, “Grayscale-based block scrambling image encryption using YCbCr color space for Encryption-then-Compression systems,” *APSIPA Trans. Signal and Information Processing*, vol. 8, no. E7, Feb. 2019.

[35] T. Chuman, W. Sirichotedumrong, and H. Kiya, “Encryption-then-Compression systems using grayscale-based image encryption for JPEG images,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 6, pp. 1515-1525, June 2019.

[36] K. Engan, S. O. Aase, and J. Hakon Husoy, “Method of optimal directions for frame design,” *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP1999)*, pp. 2443-2446, 1999.

[37] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionary for sparse representation,” *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311-4322, Nov. 2006.

[38] B. K. Natarajan, “Sparse approximate solutions to linear systems,” *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227-234, 1995.

Authors

Takayuki Nakachi received a Ph.D. degree in electrical engineering from Keio University, Tokyo, Japan, in 1997. Since joining the Nippon Telegraph and Telephone Corporation (NTT) in 1997, he has been engaged in research on super-high-definition image/video coding and media transport technologies. From 2006 to 2007, he was a visiting scientist at Stanford University. Dr. Nakachi is a member of the Institute of Electrical and Electronics Engineers (IEEE) and the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.

Hitoshi Kiya received his B.E and M.E. degrees from the Nagaoka University of Technology in 1980 and 1982, respectively, and his Dr. Eng. degree from Tokyo Metropolitan University in 1987. In 1982, he joined Tokyo Metropolitan University, where he became a Full Professor in 2000. From 1995 to 1996, he attended the University of Sydney, Australia as a Visiting Fellow. He is a Fellow of IEEE, IEICE, and ITE. He currently serves as President-Elect of APSIPA, and he served as Inaugural Vice President (Technical Activities) of APSIPA from 2009 to 2013 and as Regional Director-at-Large for Region 10 of the IEEE Signal Processing Society from 2016 to 2017.