IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 26, 2020, accepted July 18, 2020, date of publication July 22, 2020, date of current version August 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3011112

A Privacy-Preserving Learning Framework for Face Recognition in Edge and Cloud Networks

YITU WANG, (Student Member, IEEE), AND TAKAYUKI NAKACHI, (Member, IEEE)
NTT Network Innovation Laboratory, NTT Corporation, Yokosuka 239-0847, Japan

Corresponding author: Yitu Wang (yitu.wang.dp@hco.ntt.co.jp)

This work is financially supported by NTT Network Innovation Laboratory, NTT Corporation.

ABSTRACT Offloading the computationally intensive workloads to the edge and cloud not only improves the quality of computation, but also creates an extra degree of diversity by collecting information from devices in service. Nevertheless, significant concerns on privacy are raised as the aggregated information could be misused without the permission by the third party. Sparse coding, which has been successful in computer vision, is finding application in this new domain. In this paper, we develop a secured face recognition framework to orchestrate sparse coding in edge and cloud networks. Specifically, 1). To protect the privacy, a low-complexity encrypting algorithm is developed based on random unitary transform, where its influence on dictionary learning and sparse representation is analysed. Furthermore, it is proved that such influence will not affect the accuracy of face recognition. 2). To fully utilize the multi-device diversity and avoid big data transmission between edge and cloud, a distributed learning framework is established, which extracts deeper features in an intermediate space, expanded according to the dictionaries from each device. Classification is performed in this new feature space to combat the noise and modeling error. Finally, the efficiency and effectiveness of the proposed framework is demonstrated through simulation results.

INDEX TERMS Information security, edge and cloud networks, face recognition, sparse representation.

I. INTRODUCTION

Face Recognition (FR) has been a prominent biometric technique for identity authentication in a wide range of areas and applications, e.g., public security and virtual reality [2]. Due to both the scientific challenge and its practical implementation value, FR has been a long-standing research topic, where significant theoretical and experimental research has been done to promote the accuracy of FR. With the inspiration from the sparsity mechanism of human vision system and the success of sparse coding in image processing, the sparse representation based FR algorithms have received large attention and achieved excellent performance [3]. In [4], K-Singular Value Decomposition (K-SVD) algorithm is adopted to learn a discriminative dictionary, then applies Orthogonal Matching Pursuit (OMP) to find the sparse representation for FR. In [5], Fisher discrimination criterion is adopted to further enhance the discrimination capability of the learned dictionary. With the advances in machine learning and Artificial Intelligence (AI) techniques, deep learning based classification algorithms have established themselves as strong competitors to sparse representation based algorithms [6], while suffering from lack of insight. In [7], [8], deep convolutional neural network is adopted to extract high-level visual features, which boosts the performance of face recognition. In [9], Deep Convolutional Neural Network (DCNN) is adopted to exploit the intraspectrum discrimination and interspectrum correlation to promote the FR accuracy. To better cope with the image variations in practical, such as change of pose, illumination, etc., an innovative FR algorithm is proposed in [10]. Moreover, algorithm for color face recognition is proposed in [11]. However, these complex and well-engineered approaches require substantial effort in parameter training, e.g., huge amount of training data is required for the well training, and retraining is preferred when users are added or removed. Thus, they pose exigent requirements on computational capability, which cannot be easily satisfied by solely relying on devices due to their limited resources, e.g., constrained computation capability and memory size.

A cloud built on top of the data center, which seamlessly integrates storage and computation, seems to be an ideal

The associate editor coordinating the review of this manuscript and approving it for publication was Tai-Hoon Kim.

platform for implementing the FR algorithms. It however faces significant challenges from data collection and service distribution over the network, given the devices in service are globally and remotely distributed [12]. One promising solution to address these limitations is to make use of the hierarchically distributed computing structure consisting of the edge, cloud and devices. In this stand, not only the tension between computation-intensive applications and resource-limited mobile devices can be significantly released, but also the long latency incurred due to the information exchange between devices and the cloud in wide area networks can be totally avoided [13], [14]. In the literature, [15], [16] improve the computational efficiency of FR by distributing the computation tasks to the edge and cloud, respectively. However, such strategies suffer from one major drawback, i.e., they use the edge and cloud simply to accelerate the computation, while neglecting the extra degree of diversity generated by aggregating information from multiple devices in service at the cloud.

To exploit more dimensions of the edge and cloud resources for not only fulfilling the computation demands, we allow the cloud to produce a joint FR result based on the dictionaries from each device, and try to study *What is the fundamental benefit of exploiting the multi-device diversity?* Our objective is to construct a learning framework to reduce the computation demands at each mobile device, while taking advantage of this benefit to produce a more accurate FR result. However, this may lead to serious privacy concerns, especially when the sharing of such biometric information at the cloud is allowed, as it could be collected and misused without the permission by the third party.

In this paper, we develop a privacy preserving framework for FR in edge and cloud networks based on sparse representation. The motivation and main contributions of this paper are summarized as follows,

1) **Preserve the privacy by random unitary transform:** Commonly adopted encrypting algorithms, such as Advanced Encryption Standard (AES) and Secure Hash Algorithm (SHA), make it computationally difficult to directly extract the original images from the encrypted ones. However, they cannot render the FR algorithms valid, i.e., dictionaries/recognition results cannot be trained/drawn from the encrypted images. Even though algorithms as Homomorphic Encryption (HE) and secure Multi-Party Computation (MPC) allow computation on cipher-texts [17], they are faced with the curse of dimensionality regarding the size of images, and thus incur significant computational complexity. To address this challenge, we develop a low-complexity encrypting algorithm based on random unitary transform, which enables that dictionaries/FR results can be trained/drawn from the encrypted images. Moreover, it is proved both theoretically and through simulation that such encryption will not affect the accuracy of FR.
2) **Exploit multi-device diversity by ensemble learning:** The performance of the sparse representation-based FR algorithms relies on the number of training samples. However, the excessive cost of bandwidth, computation and storage makes it difficult to gather all the training samples from devices at the cloud for dictionary training. When the development of a powerful single classifier requires considerable efforts, alternatively, it is possible to train several different classifiers with adequate performance at the edge servers, and then combine the computing results to produce a final output at the cloud. The main premise of this paradigm is to create several classifiers with similar bias and then combining their outputs to reduce the variance. In this stand, not only the communication between the edge servers and the cloud can be substantially reduced, but also a classifier with better scalability and flexibility can be constructed. To this end, we first obtain the decision templates for each class, which extracts deeper features in the intermediate space expanded according to each of the dictionaries. Then, upon receiving a testing sample, the FR decision is carried out according to the pairwise similarity between its decision profile and each of the decision templates, which proves to be effective in combating the noise and modeling error.

The rest of this paper is organized as follows. Section II discusses the related works. Section III presents the system model. In Section IV, we propose the secured FR framework based on sparse representation. Following this, the performance of the proposed algorithm is evaluated in Sections V. Finally, this paper concludes with Section VI.

II. RELATED WORKS

This paper develops an analytical framework for face recognition in cloud and edge networks from a privacy preserving perspective. In this section, we briefly review the existing works on FR in cloud and edge networks, and dictionary learning for FR.

A. FR IN CLOUD AND EDGE NETWORKS

Machine learning based FR algorithms are widely applied due to its superior performance compared with traditional methods. However, it is still a consensus that the memory space and computational capability requirements for the training/testing process are quite high. To reduce the cost and satisfy such high demand, may application operators prefer to outsource their intensive computation and extreme volume of data to the edge and cloud servers [18]. Especially, bulk amount of data are generated with the development of Internet of Things, which far exceed the processing capability of local devices.

The face data for machine learning, such as in [19], are usually transmitted to the edge and cloud servers without encryption, and thus, at the risk of being eavesdropped and abused. Many publications address the privacy preserving aspect of FR in edge and cloud computing scenario.

In [20], Convolution Neural Network (CNN) is adopted for FR, and a privacy-preserving protocol is designed based on secure nearest neighbor algorithm. In [21], a lightweight FR algorithm is proposed based on depthwise separable convolutions and weight evaluation module, and Bayesian Generative Adversarial Networks (GAN) is adopted to address the privacy-preserving issue. However, such strategies use the edge and cloud simply to accelerate the computation, while neglecting the extra degree of diversity generated by aggregating information from multiple devices in service at the cloud. Different from the above works, we try to exploit more dimensions of the edge and cloud resources for not only fulfilling the computation demands, but also produce a more accurate FR result by exploiting the multi-device diversity.

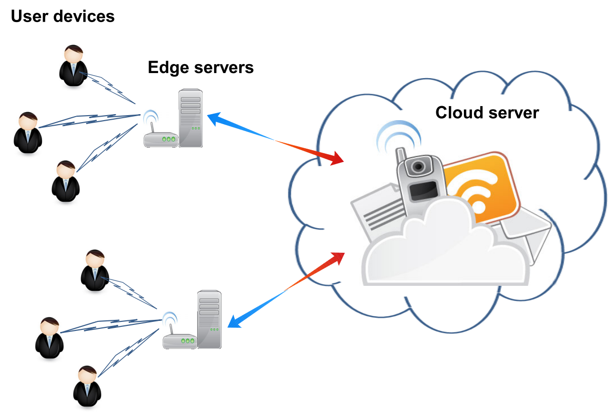


FIGURE 1. System model.

B. DICTIONARY LEARNING FOR FR

Dictionary learning has long been an active research area for pattern recognition. In [4], the dictionary and classifier are learned in a joint manner to enhance the discriminability, and thus results in an improved recognition accuracy. Following this, the idea of sparse embedded dictionary learning is proposed to reduce the computational cost. Also, the performance can be further improved due to the joint optimization of the data structure and the dictionary [22]. Recently, to better apply FR in practical, structured sparse representation is utilized to deal with occlusion and illumination variation [23]. To address the scarcity of training samples, an algorithm based on sparse discriminative multi-manifold embedding is proposed for one training sample per person face recognition problem [24].

To apply dictionary learning in edge and cloud networks, the privacy of the face data for machine learning should be strictly preserved. However, the complicated key management/distribution and the associated encryption/decryption process could be time consuming. For most of the FR applications, it is required that thousands of users can get the feedback in seconds [18]. To bypass this problem, we encrypt the data only once at each user device by adopting random unitary transform, and consider how to orchestrate sparse representation-based FR algorithm with the encrypted domain.

III. SYSTEM MODEL

In this section, we first introduce the architecture of the edge and cloud network. After discussing the method for training discriminative dictionary and obtaining the sparse representation, the FR problem under privacy preserving constraints is formulated.

A. EDGE AND CLOUD NETWORK

We first introduce the communication model for the edge and cloud network as shown in Fig. 1, where N single core user devices, denoted as set $\mathcal{N}$, are assisted by M edge servers, denoted as set $\mathcal{M}$, and one remote cloud. The mobile devices are running applications featuring FR,[1] such as interactive gaming and virtual reality applications [13], [25]. Among the user devices in $\mathcal{N}$, L classes of individuals are to recognize, denoted as $\mathcal{L}$, and the training set for class $i \in \mathcal{L}$ and device $j \in \mathcal{N}$ is denoted as $\boldsymbol{B}_i^j$ [26]. Each edge server in $\mathcal{M}$ is a light-weight computing center deployed at a wireless access point, i.e., only with limited computation and storage resources, while the remote cloud is equipped with a much stronger processor and connects with each edge server using the backbone network [27].

In the edge and cloud network, a device will offload its computation tasks to the edge server in close proximity via wireless channels,[2] the edge server together with the cloud will execute the computation tasks on behalf of the device. It is widely expected that most of data generated by the devices must be processed locally, either at the devices or at the edge servers, for otherwise the total amount of data for a centralized cloud would overwhelm the network bandwidth [14]. In addition, such distributed computing hierarchy provides extra degree of freedom for system flexibility, and could respond with a much smaller delay.

B. SPARSE REPRESENTATION BASED FACE RECOGNITION

The FR problem is defined as, using labeled training samples from L distinct classes to determine the class to which a new testing sample belongs. For this purpose, we adopt sparse representation for FR, which has two benefits for solving the problem in our work,

1) It is flexible enough to capture much of the variation in real datasets, and provides insights into the features extracted from the training dataset.

[1] All the captured images are pre-processed by the mobile devices, i.e., images are segmented into fine-grained head pictures [26].

[2] Some physical layer access scheme, e.g., Code Division Multiple Access (CDMA), is adopted to allow multiple devices to share the same edge server simultaneously and efficiently. There are also existing algorithms for the matching between mobile devices and edge servers based on their channel quality [27]. In this paper, we assume such matching is accomplished during network setup period.

2) The algorithm is with low computational complexity, and does not rely on domain-specific knowledge, which make it possible to apply to very general and large-scale scenarios.

To classify according to the sparse representation of face images, we proceed by two steps:

1) DICTIONARY TRAINING

Given an m-dimensional training set $\boldsymbol{B}^j \in \mathbb{R}^{m\times|\boldsymbol{B}^j|}$ of device $j \in \mathcal{N}$, learning a reconstructive dictionary with K^j atoms for obtaining the sparsest representation can be accomplished by solving the following optimization problem,

$$\begin{aligned}&\min_{\boldsymbol{D}^j} ||\boldsymbol{B}^j - \boldsymbol{D}^j\boldsymbol{X}^j||_2^2\\ &s.t.\ ||\boldsymbol{x}_i^j||_0 \le \epsilon, \quad \forall i \in \{1,2,\cdots,|\boldsymbol{B}^j|\},\end{aligned} \tag{1}$$

where $\boldsymbol{D}^j \in \mathbb{R}^{m\times K^j}$ is the learned dictionary, $\boldsymbol{X}^j \in \mathbb{R}^{K^j\times|\boldsymbol{B}^j|}$ is the sparse representation of $\boldsymbol{B}^j$, and ϵ is the sparsity constraint factor.

To jointly learn dictionary and classifier for FR, a label consistent regularization term and a classification error term are included to both embrace the reconstructive and discriminative capability as in [28]. The optimization problem in Eq. (1) can be modified into

$$\begin{aligned}P(\boldsymbol{B}^j) = &\min_{\boldsymbol{D}^j,\boldsymbol{W}^j,\boldsymbol{A}^j} ||\boldsymbol{B}^j - \boldsymbol{D}^j\boldsymbol{X}^j||_2^2 + \alpha||\boldsymbol{C}^j - \boldsymbol{A}^j\boldsymbol{X}^j||^2\\ &\qquad + \beta||\boldsymbol{H}^j - \boldsymbol{W}^j\boldsymbol{X}^j||_2^2\\ &\qquad s.t.\ ||\boldsymbol{x}_i^j||_0 \le \epsilon, \quad \forall i \in \{1,2,\cdots,|\boldsymbol{B}^j|\},\end{aligned} \tag{2}$$

where α and β are the weights for the label consistent term and reconstruction error term, $\boldsymbol{C}^j$ is the discriminative sparse code of the input training samples in $\boldsymbol{B}^j$, in which the nonzero entries occur at those indices where the input and the dictionary share the same label, $\boldsymbol{A}^j$ is a linear transformation matrix, which transforms the original sparse codes to be the most discriminative in sparse feature space $\mathbb{R}^{K^j}$, $\boldsymbol{W}^j$ denotes the classifier parameters, and $\boldsymbol{H}^j$ is the class labels of training samples in $\boldsymbol{B}^j$. To make the problem trackable, Eq. (2) is rewritten as

$$\begin{aligned}P(\boldsymbol{B}^j) = &\min_{\boldsymbol{D}^j,\boldsymbol{W}^j,\boldsymbol{A}^j} ||\boldsymbol{Z}^j - \boldsymbol{T}^j\boldsymbol{X}^j||_2^2\\ &\qquad s.t.\ ||\boldsymbol{x}_i^j||_0 \le \epsilon, \quad \forall i \in \{1,2,\cdots,|\boldsymbol{B}^j|\},\end{aligned} \tag{3}$$

where

$$\boldsymbol{Z}^j = \begin{bmatrix}\boldsymbol{B}^j\\ \sqrt{\alpha}\boldsymbol{C}^j\\ \sqrt{\beta}\boldsymbol{H}^j\end{bmatrix}, \quad \boldsymbol{T}^j = \begin{bmatrix}\boldsymbol{D}^j\\ \sqrt{\alpha}\boldsymbol{A}^j\\ \sqrt{\beta}\boldsymbol{W}^j\end{bmatrix}. \tag{4}$$

This optimization problem can be solved efficiently using K-SVD [4].

2) FACE RECOGNITION

Given a testing sample $\boldsymbol{y}^k$, the sparse representation based on the trained dictionary $\boldsymbol{D}^j$ can be calculated according to

$$\begin{aligned}S(\boldsymbol{y}^k, \boldsymbol{D}^j) = &\min_{\boldsymbol{x}_j^k} ||\boldsymbol{y}^k - \boldsymbol{D}^j\boldsymbol{x}_j^k||_2^2\\ &s.t.\ ||\boldsymbol{x}_j^k||_0 \le \epsilon.\end{aligned} \tag{5}$$

The above optimization problem can be efficiently solved using Orthogonal Matching Pursuit (OMP) [4].

Then, the class label for this testing sample $\boldsymbol{y}^k$ can be estimated using the linear predictive classifier $\boldsymbol{W}^j$ as

$$l_j^k = \arg\max\{\boldsymbol{W}^j\boldsymbol{x}_j^k\}. \tag{6}$$

Under the conventional single device setting [28], Eq. (6) is reduced into $l = \arg\max\{\boldsymbol{W}\boldsymbol{x}\}$. Because of the joint optimization of the dictionary and classifier, such a scheme has been demonstrated to be effective in FR. However, there are still two major drawbacks,

- The classify decision is made based on local dictionary only, which makes this method vulnerable to noise and modeling error.
- The inherent structure of the class label vector $\boldsymbol{W}\boldsymbol{x}$ is not fully exploited due to the arg max operation.

C. PROBLEM FORMULATION

In the edge and cloud network, observing that the cloud could obtain the full-knowledge of the dictionaries and classifiers of each device by collecting the information from the edge servers, it is possible to produce a joint FR result to exploit more dimensions of such hierarchically distributed computing structure and further enhance the performance. However, the sharing of dictionaries and classifiers brings significant security concerns regarding the information passing during the process. To this end, our objective is to construct a privacy preserving framework and improve the recognition accuracy by exploiting the multi-device diversity, which can be formally formulated as

$$\begin{aligned}&\min_{g(\cdot)} \left|\left|\boldsymbol{G}^k - g\big(\tilde{\boldsymbol{X}}_1^k, \overline{\boldsymbol{W}}^1, \cdots, \tilde{\boldsymbol{X}}_N^k, \overline{\boldsymbol{W}}^N\big)\right|\right|_0\\ &s.t.\ \overline{\boldsymbol{B}}^j = f(p, \boldsymbol{B}^j)\\ &\qquad \overline{\boldsymbol{Y}}^k = f(p, \boldsymbol{Y}^k)\\ &\qquad <\overline{\boldsymbol{D}}^j, \overline{\boldsymbol{W}}^j> = P(\overline{\boldsymbol{B}}^j)\\ &\qquad \tilde{\boldsymbol{X}}_j^k = S(\overline{\boldsymbol{Y}}^k, \overline{\boldsymbol{D}}^j)\\ &\qquad ||g(\tilde{\boldsymbol{X}}_1^k, \overline{\boldsymbol{W}}^1, \cdots, \tilde{\boldsymbol{X}}_N^k, \overline{\boldsymbol{W}}^N)\\ &\qquad - g(\boldsymbol{X}_1^k, \boldsymbol{W}^1, \cdots, \boldsymbol{X}_N^k, \boldsymbol{W}^N)||_0 = 0,\end{aligned} \tag{7}$$

where $\boldsymbol{G}^k$ contains the class labels of the testing images, $g(\cdot)$ is the classify function producing the estimated class label matrix, $f(\cdot)$ is the encrypting function, p is the private key, and $\tilde{\boldsymbol{X}}_j^k$ represents the sparse representation of $\overline{\boldsymbol{Y}}^k$ under $\overline{\boldsymbol{D}}^j$. The first two constraints in Eq. (7) guarantee that the original training and testing images are difficult to be recognized from the encrypted ones without the knowledge of p, and the last constraint makes sure the recognition results can be drawn from the encrypted images without performance loss.

IV. SECURED FACE RECOGNITION FRAMEWORK IN EDGE AND CLOUD NETWORKS

In this section, a secured framework for FR is proposed based on sparse representation. To satisfy the privacy preserving constraints in Eq. (7), we briefly introduce random unitary transform and outline three important properties, based on which its influence on dictionary learning and sparse representation is analyzed. It is proved that the results of FR will not be affected. To fully exploit the multi-device diversity, an ensemble learning framework is proposed by two steps,

1) By deploying a small ensemble training set, we extract the decision templates for each class in the space expanded by the estimated class label vectors according to the dictionaries of each device, so as to fully exploit the structure of the class label vectors in a holistic manner.
2) We estimate the degree of similarity between the decision profile of the testing sample and each of the decision templates in a pairwise manner, in order to combat the noise and modeling error for a refined FR result.

A. RANDOM UNITARY TRANSFORM

In order to not only preserve the privacy of the system, but also enable computing on cipher-texts, the random unitary transform is one promising method, which proves to be effective for biometric template protection [29]. Moreover, random unitary transform provides us with a desired low computational complexity, which makes it possible to apply the proposed algorithm to the scenarios with a large cipher-text size. Therefore, the encrypted training and testing samples are generated using random unitary transform.

Any vector $\boldsymbol{v} \in \mathbb{R}^{m\times 1}$ encrypted by random unitary matrix $\boldsymbol{Q}_p \in \mathbb{C}^{m\times m}$ with private key p can be expressed as follows,

$$\bar{\boldsymbol{v}} = f(p, \boldsymbol{v}) = \boldsymbol{Q}_p \boldsymbol{v}, \tag{8}$$

where $\bar{\boldsymbol{v}}$ is the encrypted vector, and the unitary matrix $\boldsymbol{Q}_p$ satisfies

$$\boldsymbol{Q}_p^* \boldsymbol{Q}_p = \boldsymbol{I}, \tag{9}$$

where $[\cdot]^*$ and $\boldsymbol{I}$ represents the Hermitian transpose and identity matrix, respectively. Gram-Schmidt orthogonalization can be adopted for generating $\boldsymbol{Q}_p$.[3] The encrypted vector has three properties [30] as follows,

- Conservation of the Euclidean distances

$$||\boldsymbol{v}_i - \boldsymbol{v}_j||_2^2 = ||\bar{\boldsymbol{v}}_i - \bar{\boldsymbol{v}}_j||_2^2, \tag{10}$$

- Norm isometry

$$||\boldsymbol{v}||_2^2 = ||\bar{\boldsymbol{v}}||_2^2, \tag{11}$$

- Conservation of inner products

$$\boldsymbol{v}_i \times \boldsymbol{v}_j^T = \bar{\boldsymbol{v}}_i \times \bar{\boldsymbol{v}}_j^T, \tag{12}$$

where $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are two distinct vectors with the same dimension.

[3] Such encrypting technique has been proved to be robust in terms of brute-face attack, diversity and irreversibility [29].

B. SECURED FACE RECOGNITION

According to random unitary transform, the encrypted training samples $\overline{\boldsymbol{B}}^j \in \mathbb{R}^{m\times|\boldsymbol{B}^j|}, j \in \mathcal{N}$ and testing samples $\overline{\boldsymbol{Y}}^k$, $k \in \mathcal{N}$ are generated as follows,

$$\begin{aligned} \overline{\boldsymbol{B}}^j &= f(p, \boldsymbol{B}^j) = \boldsymbol{Q}_p \boldsymbol{B}^j \\ \overline{\boldsymbol{Y}}^k &= f(p, \boldsymbol{Y}^k) = \boldsymbol{Q}_p \boldsymbol{Y}^k. \end{aligned} \tag{13}$$

To obtain the dictionary and classifier parameter based on encrypted training samples, the following optimization problem should be solved,

$$\begin{aligned} <\overline{\boldsymbol{D}}^j, \overline{\boldsymbol{W}}^j> &= \min_{\boldsymbol{D}^j, \boldsymbol{W}^j} ||\overline{\boldsymbol{Z}}^j - \overline{\boldsymbol{T}}^j \boldsymbol{X}^j||_2^2 \\ & s.t.\ ||\boldsymbol{x}_i^j||_0 \le \epsilon, \quad \forall i \in \{1, 2, \cdots, |\boldsymbol{B}^j|\}, \end{aligned} \tag{14}$$

where

$$\overline{\boldsymbol{Z}}^j = \begin{bmatrix} \overline{\boldsymbol{B}}^j \\ \sqrt{\alpha}\boldsymbol{C}^j \\ \sqrt{\beta}\boldsymbol{H}^j \end{bmatrix}, \quad \overline{\boldsymbol{T}}^j = \begin{bmatrix} \overline{\boldsymbol{D}}^j \\ \sqrt{\alpha}\overline{\boldsymbol{A}}^j \\ \sqrt{\beta}\overline{\boldsymbol{W}}^j \end{bmatrix}. \tag{15}$$

In the following theorem, the influence of random unitary transform on the trained dictionary and classifier parameter is demonstrated.

Theorem 1: The trained dictionary $\overline{\boldsymbol{D}}^j$ and classifier parameter $\overline{\boldsymbol{W}}^j$ based on the encrypted training samples $\overline{\boldsymbol{B}}^j$ should satisfy

$$\begin{aligned} \overline{\boldsymbol{D}}^j &= \boldsymbol{Q}_p \boldsymbol{D}^j \\ \overline{\boldsymbol{W}}^j &= \boldsymbol{W}^j \end{aligned} \tag{16}$$

where $\boldsymbol{D}^j$, $\boldsymbol{W}^j$ are the trained dictionary and classifier parameter using the original training samples as in Eq. (3). ■

Proof: Please refer to Appendix. □

To obtain the sparse representation of encrypted testing samples $\overline{\boldsymbol{Y}}^k$ based on $\overline{\boldsymbol{D}}^j$, the following optimization problem should be solved,

$$<\tilde{\boldsymbol{X}}_k^j> = \min_{\boldsymbol{X}_k^j} ||\overline{\boldsymbol{Y}}^k - \overline{\boldsymbol{D}}^j \boldsymbol{X}_k^j||_2^2 \quad s.t.\ ||\boldsymbol{x}_k^j||_0 \le \epsilon, \tag{17}$$

In the following theorem, it is demonstrated that the sparse representation is not affected by random unitary transform.

Theorem 2: Given the encrypted dictionary $\overline{\boldsymbol{D}}^j$ and testing samples $\overline{\boldsymbol{Y}}^k$, its sparse representation $\tilde{\boldsymbol{X}}_k^j$ satisfies,

$$\tilde{\boldsymbol{X}}_k^j = \boldsymbol{X}_k^j, \tag{18}$$

where $\boldsymbol{X}^j$ is the sparse representation of $\boldsymbol{Y}^k$ under $\boldsymbol{D}^j$. ■

Proof: Please refer to *Sparse coding step* in Appendix. □

Remark 1: According to the above two theorems, the classifier parameter $\overline{\boldsymbol{W}}^j$ and sparse representation $\tilde{\boldsymbol{X}}_k^j$ under the encrypted training/testing images are identical to those under the original unencrypted training/testing images. To this end, we have

$$g(\tilde{\boldsymbol{X}}_1^k, \overline{\boldsymbol{W}}^1, \cdots, \tilde{\boldsymbol{X}}_N^k, \overline{\boldsymbol{W}}^N) = g(\boldsymbol{X}_1^k, \boldsymbol{W}^1, \cdots, \boldsymbol{X}_N^k, \boldsymbol{W}^N), \tag{19}$$

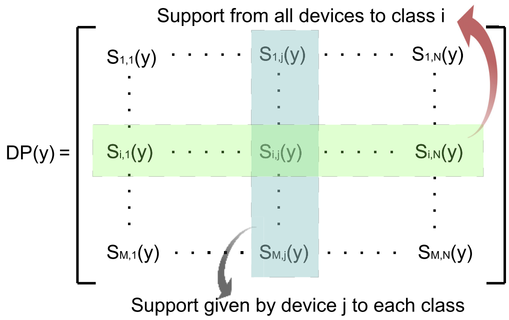


FIGURE 2. Decision profile.

which indicates that by adopting unitary random transform, the privacy can be preserved without any performance degradation.

C. ENSEMBLE LEARNING FRAMEWORK

We propose an ensemble learning framework to fully exploit the multi-device diversity by exploring the structure of the estimated label vectors, and design the function $g(\cdot)$ in two steps:

1) Ensemble Training: The class label vectors for ensemble training samples are estimated according to each $\boldsymbol{D}^j$ and $\boldsymbol{W}^j$, based on which the structurized decision profile in the intermediate space can be obtained. Then, the decision template for each class is extracted, which records the most typical decision profile for that class.
2) Recognizing: Upon a testing sample, its decision profile is calculated according to each $\boldsymbol{D}^j$ and $\boldsymbol{W}^j$, then the classification is identified according to the pairwise similarity between its decision profile and each of the decision templates.

Before going into the details of the proposed framework, we first introduce the concept of decision profile and decision template.

- **Decision Profile (DP)**: An example of decision profile with the size $\mathbb{R}^{L\times N}$ is shown in Fig. 2. The j-th column represents the normalized class label vector according to the dictionary of device j, which can be interpreted as the degree of support given by device j to each class. The i-th row represents the degree of support from all devices to class i. An intuitive idea is that, the larger the support, the more likely to identify the classification to that class. However, such intuition does not consider the inter-dependency among those class label vectors, leading to the failure of fully exploiting the multi-device diversity. To better exploit the structure of the class label vectors in a holistic manner, we ignore the context of the DP, and treat the values as features in a new feature space, i.e., the *intermediate feature space*. The final decision is made by an ensemble classifier that takes the intermediate feature space as input, and produces a class label.
- **Decision Template (DT)**: A DT is to remember the most typical DP for each class. Then, given a pairwise similarity measurement, it is possible to compare the current DP of a testing sample with each of the DTs for each class, and the closest match will produce the classification result. The decision is dependent on neither the absolute value of reconstruction error nor the exact support in the original feature space, but the relative value based on the comparison between the current DP and DT. Therefore, such a method can efficiently combat the noise and modeling error, which are already included in DT.

The proposed framework mainly consists of two stages,

1) TRAINING

The ensembler and classifiers are trained in three steps, as shown in Fig. 3.

- *Training Set Division*: Upon receiving the training samples $\boldsymbol{B}^j$ for device j at the edge server, which can be partitioned into two parts, namely, one for dictionary training $\boldsymbol{B}^{j,D}$ and the other for ensemble training.[4]
- *Dictionary and Classifiers Training*: A discriminative dictionary $\boldsymbol{D}^j$ and classifier parameter $\boldsymbol{W}^j$ are jointly

[4]The partition method is discussed and evaluated in Section V through simulation.

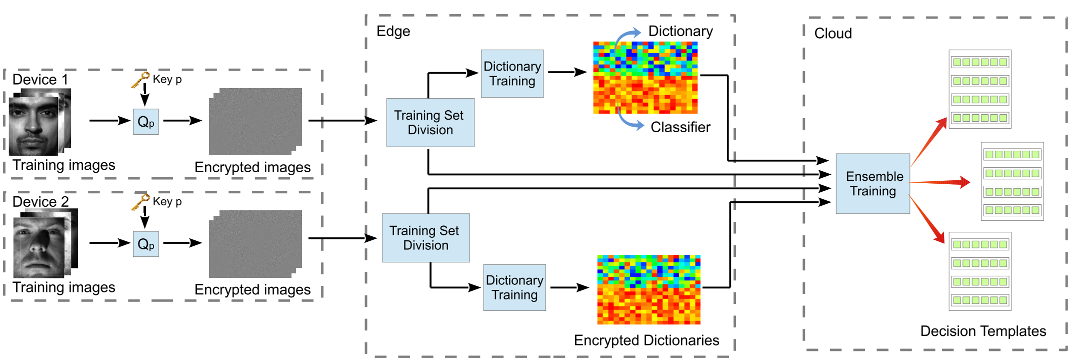


FIGURE 3. Ensembler and classifiers training.

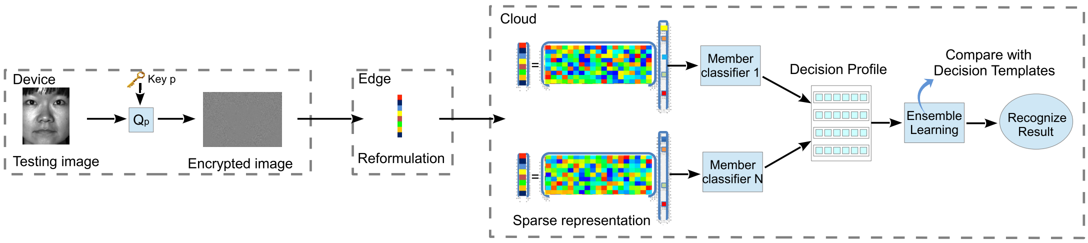


FIGURE 4. Recognizing.

trained based on the dictionary training set $\boldsymbol{B}^{j,D}$ as in Eq. (14), which are then transmitted to the cloud.

- *Ensembler Training*: After the edge servers transmit all the ensemble training samples to the cloud, the ensemble training set can be formulated accordingly as $\boldsymbol{B}^E$. The class label vectors for each sample can be estimated using the linear predictive classifiers $\boldsymbol{W}^j$, $\forall j \in \mathcal{N}$ as
$$\boldsymbol{L}^j = \{\boldsymbol{W}^j \boldsymbol{y}, \quad \forall j \in \mathcal{N}, \ \forall y \in \boldsymbol{B}^E\}. \tag{20}$$
Next, the DPs are formulated as
$$\boldsymbol{DP}(y) = [\boldsymbol{L}^1, \boldsymbol{L}^2, \cdots, \boldsymbol{L}^N], \quad \forall y \in \boldsymbol{B}_i^E. \tag{21}$$
Finally, the DT for class $i \in \mathcal{L}$ can be calculated as
$$\boldsymbol{DT}_i = \frac{1}{|\boldsymbol{B}_i^E|} \sum_{y \in \boldsymbol{B}_i^E} \boldsymbol{DP}(y). \tag{22}$$

Remark 2: Since most of the training samples are not directly transmitted to the cloud, the amount of required network bandwidth between the edge servers and the cloud can be notably reduced.

2) RECOGNIZING

The recognizing stage is illustrated in Fig. 4. First, upon receiving a testing image $\boldsymbol{y}^k$, the edge server only transmits the feature descriptor extracted by random faces [4] to the cloud. Then, we estimate the class label vectors $\boldsymbol{L}_k^j$ based on each $\boldsymbol{W}^j$ according to Eq. (20), and further obtain its $\boldsymbol{DP}(\boldsymbol{y}^k)$ according to Eq. (21). Finally, the pairwise similarity between $\boldsymbol{DP}(\boldsymbol{y}^k)$ and each of the DTs can be calculated as

$$\mu(\boldsymbol{DP}(\boldsymbol{y}^k), \boldsymbol{DT}_i) = 1 - \frac{1}{L \times N} \sum_{l \in \mathcal{L}} \sum_{j \in \mathcal{N}} \left(\boldsymbol{DT}_i(l,j) - S_{l,j}(y^k)\right)^2, \tag{23}$$

which is the squared Euclidean distance. The classification is identified as the class label i^*, where $\mu(\boldsymbol{DP}(\boldsymbol{y}^k), \boldsymbol{DT}_{i^*}) > \mu(\boldsymbol{DP}(\boldsymbol{y}^k), \boldsymbol{DT}_i), \forall i \neq i^*$.

We provide the details of the proposed framework using pseudo codes in Algorithm 1. In the pseudo-codes, Lines 1–3 provide the initialization for training sets, Lines 4–5 provide the training process for encrypted dictionary, classifiers and ensembler, Lines 6–9 calculate the sparse representation for the testing sample, and Line 10 obtains the joint recognition result by embracing ensemble learning.

Algorithm 1 Secure Face Recognition in Edge and Cloud Network Based on Ensemble Learning

1: **[Training Stage]**
2: Training sample sets $\boldsymbol{B}^j$, $\forall j \in \mathcal{N}$ are encrypted according to Eq. (13), and transmitted to the designated edge server.
3: The edge server divides $\boldsymbol{B}^j$ into a dictionary training set $\boldsymbol{B}^{j,D}$ and a part of the ensemble training set $\boldsymbol{B}^{j,E}$.
4: The edge server trains the dictionary $\boldsymbol{D}^j$ and the classifier parameter $\boldsymbol{W}^j$ based on $\boldsymbol{B}^{j,D}$ using K-SVD algorithm, and uploads both $\boldsymbol{B}^{j,D}$ and $\boldsymbol{B}^{j,E}$ to the cloud.
5: The cloud formulates the ensemble training set $\boldsymbol{B}^D$, and obtains the DTs according to Eq. (22).
6: **[Testing Stage]**
7: Testing image $\boldsymbol{y}^k$ is encrypted according to Eq. (13), and transmitted to the designated edge server.
8: The edge server uploads the formulated encrypted testing vector $\bar{\boldsymbol{y}}^k$ to the cloud.
9: The cloud calculates its sparse representation using OMP algorithm, and formulates its DP according to Eq. (21).
10: The cloud further obtain the similarity values according to Eq. (23), based on which the classification result is identified.

On the computational complexity of Algorithm 1, the most time-consuming parts are the K-SVD and OMP algorithms in Line 4 and Line 9, in which the running time is $O(mK\epsilon)$ and $O(Nm^2K)$ [31], respectively. The complexity brought by the enseble learning is $O(L(m+N))$ and $O(L^2N)$ for the training and testing phases, respectively. L is the number of classes, N denotes the number of training samples, m denotes the dimension of the dictionary, K represents the number of rows in the dictionary, and ϵ is the sparsity of the sparse representation.

Remark 3: Note that the training samples are differently and independently chosen according to different devices, and such relative uniqueness of the information available in each training sets prompts the dictionaries to capture different patterns along the system dynamics. Since the class label vectors are adaptive to the underlying structure of the dictionaries, it renders the multi-device diversity valid in the designed ensemble learning based framework, which accounts for the performance improvements.

D. DISCUSSIONS ON FURTHER IMPROVING THE PERFORMANCE

In this subsection, we discuss two possible approaches to further improve the performance, i.e., incorporating pre-processing and making use of local correlation inside multi-dimensional data.

To apply the proposed algorithm in practical, it is necessary to handle images collected in unconstrained environment, where misalignment, scale variations, in-plane as well as out-of-plane rotations are included. To avoid performance degradation, we may pre-process the data before recognition, such as performing face alignment beforehand to decrease the intra-class difference of each subject, further making the classifier more discriminative [41].

Currently, the data is presented in the form of vectors (one dimension), and then perform pattern recognition. However, such modeling ignores the inherent structure and breaks the local correlation inside multi-dimensional data. To address this issue, it is possible to adopt tensor sparse representation to process multi-dimensional data directly [32]. However, the influence of the encryption algorithm, i.e., random unitary transform, on tensor sparse representation is not straightforward, because complicated operators are involved, e.g., Kronecker product. It is an interesting problem to derive a secured pattern recognition algorithm for high-dimensional data, which we are willing to investigate in our future work.

V. SIMULATION RESULTS

In this section, the performance of the proposed framework is investigated by simulation. First, the characteristics of the proposed framework is analyzed, including the influence of key parameters, the influence of different ensemble techniques, and several properties. Second, the accuracy of the proposed framework is compared with those of the existing algorithms on several publicly available datasets. Finally, the computational complexities are compared both analytically and through simulation.

A. CHARACTERISTICS OF THE PROPOSED FRAMEWORK

In the proposed ensemble learning framework, it is nontrivial to design the combiner, because an inadequate combiner could result in an even larger variance (lower recognition accuracy). An improvement over the single best classifier or even on the average of the individual accuracies is not generally guaranteed. To address this issue, we carefully design four possible combiners,

- *Similarity-based*: The pairwise similarity between the DTs and DP of the testing sample is calculated according to Eq. (23), and the class label is determined according to the largest similarity.
- *Difference-based*: The difference, based on the fuzzy set theory, between the DTs and DP of the testing sample is calculated as in [33], and the class label is determined according to the smallest difference.
- *Max-based*: The class label receiving the largest amount of support from all the devices is selected as the result.
- *SVM-based*: The SVM model is trained according to DPs of the ensemble training samples using a linear core. The class label is determined according to the relationship between the position of the DP of the testing sample and the formulated hyperplanes.

In this simulation, the Extended YaleB database is adopted, which is one of the commonly used database for FR. The cropped and normalized face images are captured according to different angles and lighting conditions [37], [38]. There are in total 38 individuals and approximately 64 images for each individual in the database. We randomly select 32 images for each individual as the dictionary training set, 5 images for each individual as a part of the ensemble training set, while the rest for testing. For each device, the training samples are randomly selected from the dictionary training set. Such randomness makes sure each member classifier show different classification properties, and ensures our results and conclusions do not rely on special choice of the training data.

Fig. 5 demonstrates the influence of several key parameters on recognition accuracy, including the number of devices, the number of dictionary training samples, the number of ensemble training samples and its way of constitution.

Fig. 5 (a) and Fig. 5 (b) verify the performance improvement brought by exploiting the multi-device diversity through ensemble learning. We have the following observations,

1). The performance improvement is significant when the number of devices is large due to the extra degree of freedom.

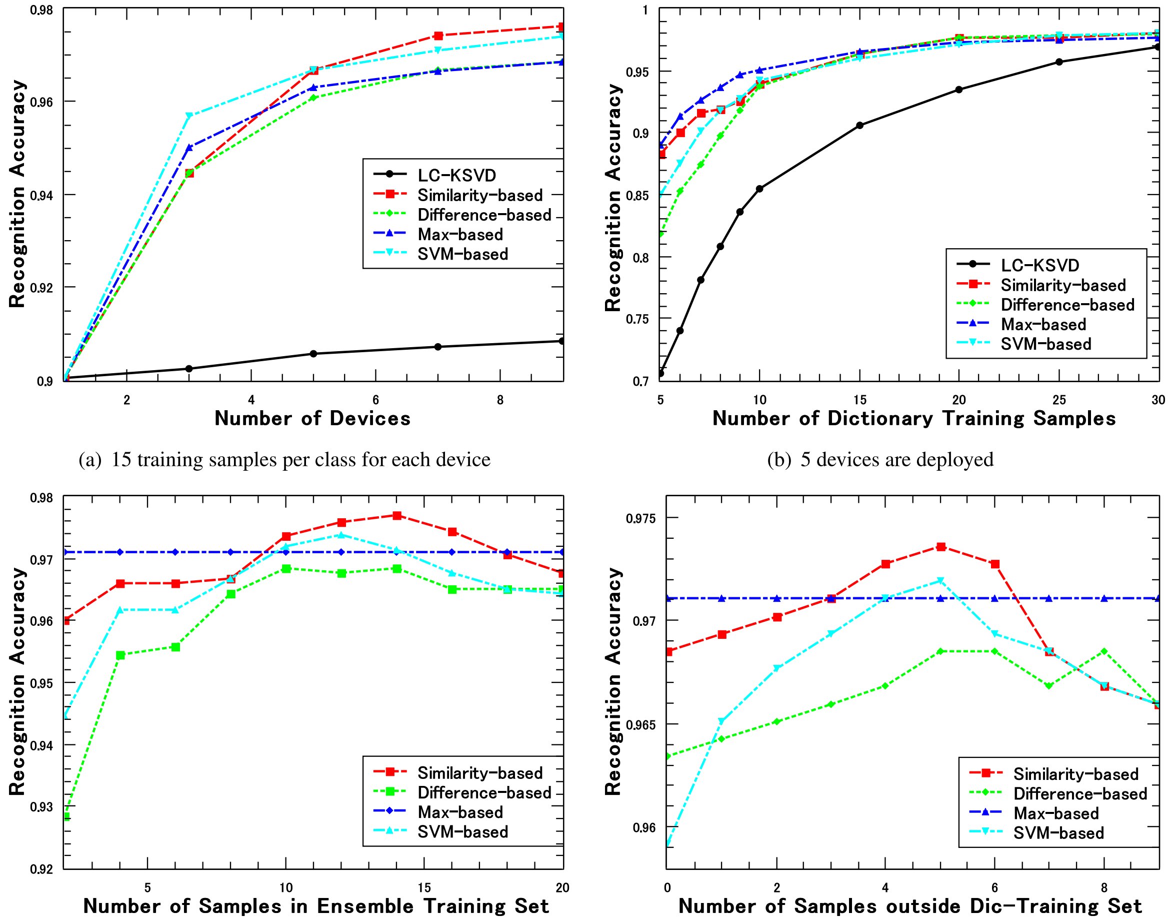
2). When the number of devices is small, e.g., smaller than 5, the SVM-based ensemble learning algorithm dominates due to the joint consideration of mean value and variance. When the number of devices is large, e.g., larger than 6, the Similarity-based ensemble learning algorithm dominates due to the bias-variance trade-off, i.e., the SVM-based algorithm enforces a low bias in parameter estimation while having a higher variance of the parameter estimates across samples.

3). The performance improvement is significant, even if there is only few dictionary training sample, which signifies the potential for practical implementation of the proposed framework.

Fig. 5 (c) and Fig. 5 (d) illustrate the influence of the choice of ensemble training set, we have

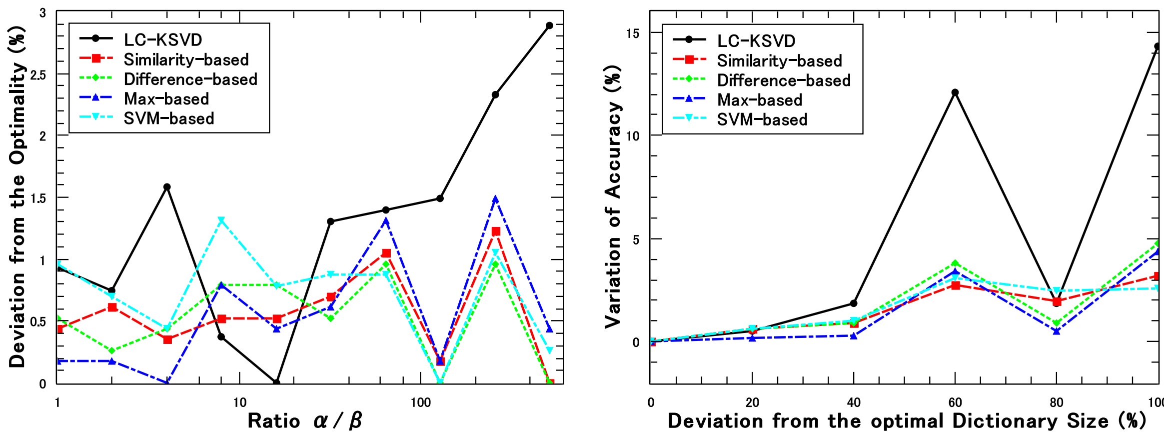
1). When the number of ensemble training samples for each class is too small, e.g., smaller than 8, the bias is so high that the algorithm misses the relevant relations between features and target outputs. When the number of ensemble training samples for each class is too large, e.g., larger than 15, the variance is so high that the algorithm models the random noise in the training data, rather than the intended outputs. Therefore, it is important to optimize the size of ensemble training set to find a balance between bias and variance to produce the best performance.

2). As for the constitution of the ensemble training set, the performance is enhanced when partly overlapping with the dictionary training set. We find that better results are obtained



(a) 15 training samples per class for each device
(b) 5 devices are deployed
(c) Half of the ensemble training samples are new
(d) 10 ensemble training samples per class

FIGURE 5. **Influence of key parameters.**

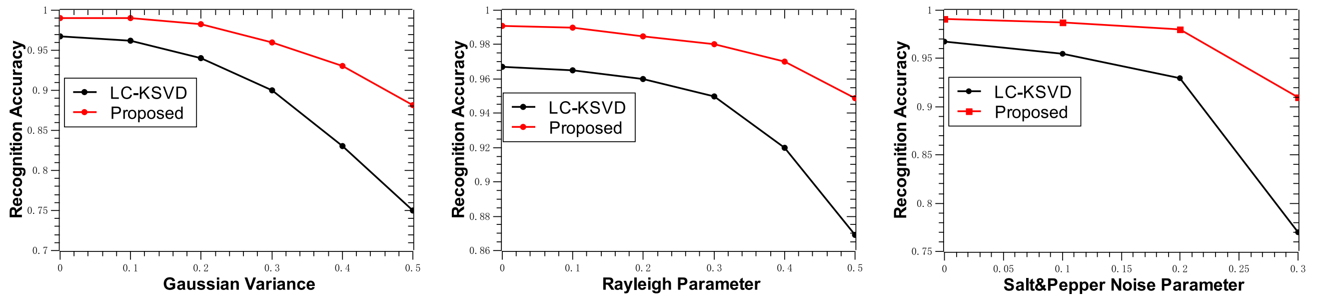


(a) $\beta = 1$
(b) 5 devices and 15 dictionary training samples per class

FIGURE 6. **Robustness of the proposed framework.**

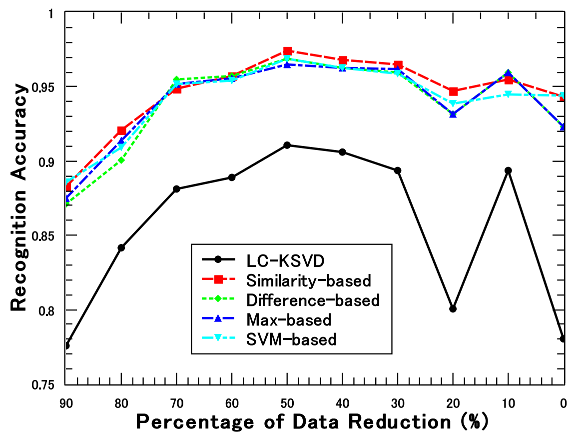
when half of the ensemble training samples are selected from the dictionary training set.

Fig. 6 demonstrates the robustness of the proposed framework towards the selection of system parameters. α is the weight for label consistent term, β is the weight for reconstruction error term, the optimal dictionary size should be selected as around half of the number of training samples for LC-KSVD [28] according to numerical experiment. All

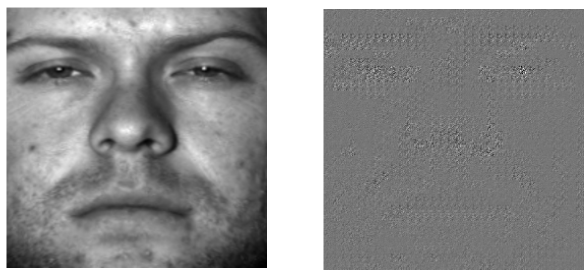


(a) Gaussian noise (b) Rayleigh noise (c) Salt&Pepper noise

FIGURE 7. Sensitivity to noises.



(a) 5 devices and 15 dictionary training samples per class



(b) Original image (c) Encrypted image

FIGURE 8. Data reducing and secure properties.

these three parameters should be given in prior to obtain the dictionaries. However, the optimality of these parameters is not only coupled, but also varies according to the selection of training samples, the size of the training set, as well as the deployed ensemble algorithms. Therefore, it is necessary to analyze the robustness of the proposed framework towards the variation of these parameters.

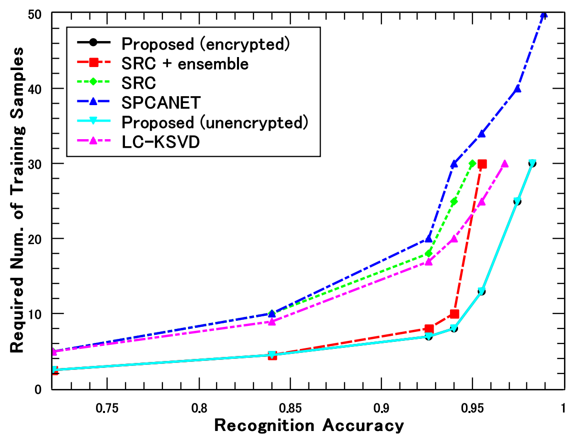
1). The ensemble learning algorithms tend to produce the optimal result with a carefully selected large α, e.g., 128 or 512 in the simulated settings. However, observing that the deviation from the optimality could be large when $\alpha = 256$, it would be safe to select a small α, e.g., 1, 2 or 4, to produce a reasonable result at the cost of slightly deviating from the optimality.

2). When the choice of the size of dictionary deviates from the optimality, the performance of the ensemble learning based frameworks only slightly deviates from the optimality, which indicates the strong compatibility with different settings.

In order to verify the scalability, as well as the robustness towards practical random noises, we conduct sensitivity analysis based on a larger dataset, i.e., the CMU PIE face dataset [35], consisting of 41,368 front-face images of 68 persons, and the face images of each person are captured under 13 different poses, 43 different illumination conditions, and 4 different facial expressions. We choose the five near-frontal poses (C05, C07, C09, C27, C29) of each subject and use all the images under different illumination conditions and facial expressions. Thus we get 170 images for each individual. Every image is normalized to the size of 32×32. We randomly select 20 images of each person as training samples. Besides high recognition accuracy, Fig. 7 illustrates the insensitivity of the proposed framework to practical noises. By adopting ensemble learning, the proposed framework becomes much more robust than the base algorithm. Especially, when the noises are small (i.e., Gaussian noise with a variance smaller than 0.2, Rayleigh noise with a parameter smaller than 0.3, Salt&Pepper noise with a parameter smaller than 0.2), the performance of the proposed framework is only slightly influenced. When the noises become larger, the performance of the proposed framework is much better than that of the base algorithm.

Fig. 8 verifies the communicational efficiency and the privacy-preserving property of the proposed framework. The data reduction ratio represents the ratio between the size of training samples and the size of the trained dictionary.

1). We find that better results are obtained when the size of dictionary is chosen as around 50% of the training



(a) Similarity-based algorithm is deployed

FIGURE 9. Performance comparison.

samples, which is consistent with the result for the base algorithm.

2). The trade-off between recognition accuracy and the size of dictionary demonstrates the flexibility of the proposed framework, e.g., 40% more storage can be saved at the cost of only near 2% performance loss.

3). By adopting random unitary transform, the training and testing images are encrypted using a private key. It is shown that the original images are difficult to be recognized from the encrypted ones, and it would be computationally expensive to obtain the original images from the encrypted ones without the knowledge of the private key.

B. PERFORMANCE COMPARISON

For performance comparison, we adopt four baseline algorithms,

- *Baseline 1 (SPCANET) [34]*: Deep learning is adopted for FR, in which a 5-layer Convolutional Neural Network (CNN) is adopted, and Principal Component Analysis (PCA) is deployed to learn filter kernels in order to extract more discriminative features.
- *Baseline 2 (SRC) [26]*: Face subspace model is adopted for sparse representation-based classification, in which the l^1-minimization problem is solved to obtain the sparsest solution to exploit the discriminative property.
- *Baseline 3 (LC-KSVD) [28]*: The label consistent term and the reconstruction error term are added to the l^1-minimization problem to both enforce the representational power as well as improve the discrimination capability.
- *Baseline 4 (SRC + ensemble) [36]*: SRC is combined with ensemble learning to exploit the multi-device diversity in order to produce an enhanced FR result.

Fig. 9 provides the performance comparison among the proposed and baseline algorithms. First, it is demonstrated that the proposed framework outperforms all the sparse-representation based baseline algorithms, because we take the full advantage of the multi-device diversity. Since the result produced by SRC is not structurized, the combination of SRC and ensemble learning framework cannot fully exploit the multi-device diversity, which leads to inferior performance. Second, it is verified that by adopting random unitary transform, the result of FR is not influenced, which proves that the proposed framework could operate on secured plane without any performance degradation. Finally, even though the deep learning based algorithm SPCANET outperforms the proposed framework by 0.7% in terms of recognition accuracy, it requires 67% more dictionary training samples for each class, which is over 80% of the entire available samples for both training and recognizing. Especially, when there are 30 dictionary training samples for each class, which is the most common setting when evaluating the performance on YaleB dataset, the proposed framework outperforms SPCANET by over 4%. When there are only 10 training samples per class, which is reasonable in real-world settings due to the scarcity of fine-grained and manually labeled data, the proposed framework outperforms SPCANET by over 12%. By combining sparse coding with ensemble learning, the performance of such hybrid algorithm becomes much closer to that of deep learning based algorithms.

TABLE 1. Performance comparison on AR and LFWcrop.

Algorithm	AR	LFWcrop
Proposed	95.99%	75.13%
SPCANET	95.00%	83.94%
LC-KSVD	90.83%	62.11%
SRC	66.50%	44.96%
SRC+ensemble	79.74%	54.09%

Moreover, we compare the performance based on AR dataset [39] and LFW dataset [40]. Regarding the AR dataset, we choose a subset consisting of 50 classes of male and female, respectively, each of which includes 14 images captured according to different facial expressions, illumination conditions and disguises. In the experiment, the images are cropped with dimension 645 and converted to gray scale. For each class, we randomly select 9 images as the entire training set, where each device randomly select 6 images as the dictionary training set, and 4 images as the ensemble training set, while the rest for testing. The LFW dataset contains more than 13,000 images of faces collected in an unconstrained environment, and they are labeled with the names of different individuals. LFW is more challenging than AR and YaleB datasets since it includes various uncontrolled variations. LFWcrop dataset is a cropped version of the LFW, keeping only the center portion of each image (i.e. the face), which exhibits real-life conditions, including misalignment, scale variations, in-plane as well as out-of-plane rotations. In the experiment, following the settings in [40], we choose person who has more than 20 photos but less than 100 photos as the sub-dataset, which contains 57 classes and 1883 images. For each person, we randomly choose ninety percent of images for training, and the remaining images for testing.

The results are given in Table. 1. Regarding the AR dataset, since the number of training samples is rather limited,

TABLE 2. Computational complexity comparison.

Algorithm	Training	Testing
Proposed	$O(mKe + L(m + N))$	$O(Nm^2K + L^2N)$
LC-KSVD	$O(mKe)$	$O(Nm^2K)$
SRC	$O(N)$	$O(N^2m^2)$
SRC+ensemble	$O(L(m + N))$	$O(N^2m^2 + L^2N)$

TABLE 3. Executing time (s).

	Training Time (s)	Testing Time (s)
Proposed	7.29	1.64×10^{-3}
SPCANET	5780	1.20
LC-KSVD	4.84	1×10^{-4}
SRC	0.30	1.07
SRC+ensemble	2.70	1.11

it is demonstrated that the proposed framework outperforms SPCANET, which is consistent with the result in the Extended YaleB dataset. Regarding the LFW dataset, it is observed that 1). the performance gain of adopting ensemble learning is valid. 2). Deep learning based algorithm could extract non-linear features, and thus performs better when handling unconstrained face images. To fill in the gap, it is possible to pre-process the data before recognition, such as performing face alignment beforehand [41].

C. COMPUTATIONAL COMPLEXITY COMPARISON

We analytically compare their computational complexities in Table. 2, where N denotes the number of training samples, m denotes the dimension of the dictionary, K represents the number of rows in the dictionary, L is the number of classes, and ϵ is the sparsity of the sparse representation. Note that it is satisfied that $N > K$. As illustrated in Fig. 8, we may choose $K = N/2$ to achieve the best performance, and $K = N/10$ to achieve an adequate performance with reduced computational complexity. Therefore, the testing complexity of the proposed algorithm is smaller than that of SRC. Regarding SPCANET, it is difficult to express the computational complexity mathematically. Therefore, we evaluate/compare the computational complexities through simulation.

Table. 3 provides the simulation results on computational complexity. As for the dictionary training time, SRC based algorithms are the fastest, because they simply down-sample the training samples, stack their columns, and align those vectors to formulate a dictionary, which does not demonstrate neither enough discrimination capability nor any representational power. LC-KSVD and the proposed algorithm need more time due to adopting K-SVD for dictionary computation. Even though SPCANET adopts 1). PCA instead of stochastic gradient descent to learn filter kernels, 2). Hashing method to simplify the nonlinear processing layer, in order to reduce the computation complexity, it still requires very long training time under such a small database. As for the testing time, LC-KSVD and the proposed algorithm are extremely fast, which makes it possible to support real-time FR related applications. SRC based algorithms are slow due to the large size of the dictionary, and SPCANET is the slowest among these algorithms due to the deep learning framework. Especially, when various FR featured applications from a variety of devices produce large amount testing samples, a large recognizing time will significantly jeopardize the Quality of Experience (QoE) of users in service. Therefore, the proposed framework demonstrates strong potential to support real-time applications.

VI. CONCLUSIONS

In this paper, we develop a secured framework for FR in edge and cloud networks based on sparse representation. To guarantee the privacy, random unitary transform is adopted, where it is proved that the accuracy of recognition will not be affected. To reduce both the computation demands at each device and the communication requirements between the edge and cloud, the dictionary and classifier learning is conducted at each edge server, and the recognition is accomplished at the cloud. To exploit the multi-device diversity, we proceed by three steps. 1). The training samples are divided into two parts, one for dictionary and classifier training, and the other for ensemble training. 2). The DT is extracted for each class in the intermediate space, expanded by the estimated the label vectors based on the ensemble training set. 3). The recognition is identified according to the pairwise similarity between the DP of the testing sample and each of the DTs. Finally, the simulation results verify the privacy preserving property as well as the superiority of the proposed framework.

ACKNOWLEDGMENT

This article was presented in part at the IEEE ICASSP 2020 [1], Barcelona, Spain.

APPENDIX

PROOF OF THEOREM 1

In the following, we analyze the relationship between the dictionary and classifier parameter learned from the original training samples and those learned from the encrypted training samples by adopting K-SVD. Specifically, K-SVD consists of two steps,

A. SPARSE CODING STEP

The objective is to estimate the sparse coefficient $\boldsymbol{X}^j$ given $\boldsymbol{T}^j$, according to the following optimization problem

$$< \boldsymbol{X}^j >= \min_{\boldsymbol{X}^j} ||\boldsymbol{Z}^j - \boldsymbol{T}^j\boldsymbol{X}^j||_2^2 \quad s.t. \ ||\boldsymbol{x}^j||_0 \leq \epsilon. \tag{24}$$

It is possible to apply OMP to find the solution to Eq. (24), where the detail can be found in [42].

When the encrypted training samples are utilized, Eq. (24) is modified into

$$< \tilde{\boldsymbol{X}}^j >= \min_{\boldsymbol{X}^j} ||\overline{\boldsymbol{Z}}^j - \overline{\boldsymbol{T}}^j\boldsymbol{X}^j||_2^2 \quad s.t. \ ||\boldsymbol{x}^j||_0 \leq \epsilon. \tag{25}$$

In the following, it is proved that the solution to Eq. (25) is identical to the solution to Eq. (24).

First, in the *Sweep* step, we compute the error $\bar{\epsilon}^j(i)$, where i denotes the label of atom, as

$$\begin{aligned}\bar{\epsilon}^j(i) &= \min_{x_i^j} ||\bar{\boldsymbol{t}}_i^j \boldsymbol{x}_i^j - \bar{\boldsymbol{Z}}^j||_2^2 = ||\bar{\boldsymbol{Z}}^j||_2^2 - \frac{((\bar{\boldsymbol{t}}_i^j)^T \bar{\boldsymbol{Z}}^j)^2}{||\bar{\boldsymbol{t}}_i^j||_2^2} \\ &= ||\boldsymbol{Q}_p\boldsymbol{B}^j||_2^2 + ||\sqrt{\alpha}\boldsymbol{C}^j||_2^2 + ||\sqrt{\beta}\boldsymbol{H}^j||_2^2 \\ &\quad - \frac{(\boldsymbol{Q}_p\boldsymbol{B}^j\boldsymbol{Q}_p\boldsymbol{D}_i^j + \alpha\boldsymbol{C}^j\boldsymbol{A}_i^j + \beta\boldsymbol{H}^j\boldsymbol{W}_i^j)^2}{||\boldsymbol{Q}_p\boldsymbol{D}^j||_2^2 + ||\sqrt{\alpha}\boldsymbol{A}^j||_2^2 + ||\sqrt{\beta}\boldsymbol{W}^j||_2^2}. \quad (26)\end{aligned}$$

According to the norm isometry property, we have $||\boldsymbol{Q}_p\boldsymbol{B}^j||_2^2 = ||\boldsymbol{B}^j||_2^2$ and $||\boldsymbol{Q}_p\boldsymbol{D}^j||_2^2 = ||\boldsymbol{D}^j||_2^2$. Based on the property of conservation of inner product, we have $\boldsymbol{Q}_p\boldsymbol{B}^j\boldsymbol{Q}_p\boldsymbol{D}_i^j = \boldsymbol{B}^j\boldsymbol{D}_i^j$. Therefore,

$$\begin{aligned}\bar{\epsilon}^j(i) &= ||\boldsymbol{B}^j||_2^2 + ||\sqrt{\alpha}\boldsymbol{C}^j||_2^2 + ||\sqrt{\beta}\boldsymbol{H}^j||_2^2 \\ &\quad - \frac{(\boldsymbol{B}^j\boldsymbol{D}_i^j + \alpha\boldsymbol{C}^j\boldsymbol{A}_i^j + \beta\boldsymbol{H}^j\boldsymbol{W}_i^j)^2}{||\boldsymbol{D}^j||_2^2 + ||\sqrt{\alpha}\boldsymbol{A}^j||_2^2 + ||\sqrt{\beta}\boldsymbol{W}^j||_2^2} \\ &= \epsilon^j(i). \quad (27)\end{aligned}$$

Hence, the minimizer of Eq. (25) is the same as that of Eq. (24), which accounts for the same support S_i.

Second, in the *Update Provisional Solution* step, the sparse representation ia updated according to the atom i with the minimum error, as

$$\begin{aligned}\bar{E}_i^j &= ||\bar{\boldsymbol{Z}}^j - \bar{\boldsymbol{T}}_{S_i}^j\boldsymbol{X}_{S_i}^j||_2^2 \\ &= ||\boldsymbol{Q}_p\boldsymbol{B}^j - \boldsymbol{Q}_p\boldsymbol{D}_{S_i}^j\boldsymbol{X}_{S_i}^j||_2^2 + ||\sqrt{\alpha}\boldsymbol{C}^j - \sqrt{\alpha}\boldsymbol{A}_{S_i}^j\boldsymbol{X}_{S_i}^j||_2^2 \\ &\quad + ||\sqrt{\beta}\boldsymbol{H}^j - \sqrt{\beta}\boldsymbol{W}_{S_i}^j\boldsymbol{X}_{S_i}^j||_2^2. \quad (28)\end{aligned}$$

According to the property of norm isometry,

$$\begin{aligned}\bar{E}_i^j &= ||\boldsymbol{B}^j - \boldsymbol{D}_{S_i}^j X_{S_i}^j||_2^2 + ||\sqrt{\alpha}\boldsymbol{C}^j - \sqrt{\alpha}\boldsymbol{A}_{S_i}^j\boldsymbol{X}_{S_i}^j||_2^2 \\ &\quad + ||\sqrt{\beta}\boldsymbol{H}^j - \sqrt{\beta}\boldsymbol{W}_{S_i}^j\boldsymbol{X}_{S_i}^j||_2^2 \\ &= E_i^j. \quad (29)\end{aligned}$$

Therefore, the provisional solution $\tilde{\boldsymbol{X}}_{S_i}^j$ to Eq. (25) is the same as $\boldsymbol{X}_{S_i}^j$ to Eq. (24).

Finally, in the *Stopping Rule* step, similarly,

$$||\bar{r}_i^j||_2^2 = ||\bar{\boldsymbol{Z}}^j - \bar{\boldsymbol{T}}_{S_i}^j\boldsymbol{X}_{S_i}^j||_2^2 = ||r_i^j||_2^2 \leq \epsilon. \quad (30)$$

Therefore, the above analysis proves that random unitary transform does not affect the sparse coding step, i.e., the solution $\tilde{\boldsymbol{X}}^j$ to Eq. (25) is exact the same as the solution $\boldsymbol{X}^j$ to Eq. (24) under the condition $\bar{\boldsymbol{D}}^j = \boldsymbol{Q}_p\boldsymbol{D}^j$.

B. DICTIONARY UPDATE STEP

The objective is to find the dictionary that best describes the training samples according to the optimization problem in Eq. (3), where K-SVD can be applied to find the solution. The detail of the algorithm can be found in [42]. When the encrypted training samples are utilized, Eq. (3) should be modified into Eq. (14). In the following, it is proved that the solution to Eq. (3) has deterministic relationship with that to Eq. (14).

In the *Compute the residual matrix* step, the representation error matrix $\boldsymbol{E}_d^j$ for the d-th column in Eq. (3) is given by

$$\boldsymbol{E}_d^j = \boldsymbol{Z}^j - \sum_{i \neq d}^{K^j} \boldsymbol{t}_i^j(\boldsymbol{x}_i^j)^T, \quad (31)$$

where $\boldsymbol{t}_i^j$ represents the i-th column in $\boldsymbol{T}^j$, $(\boldsymbol{x}_i^j)^T$ represents the i-th row in $\boldsymbol{X}^j$. To minimize the l^2-norm of $\boldsymbol{E}_d^j$ while keeping the cardinalities of all the representations fixed, we restrict $\boldsymbol{E}_d^j$, by choosing only the columns where the entries in the row are non-zero, into $(\boldsymbol{E}_d^j)^R$. Then apply SVD to find an approximate solution,

$$\begin{aligned}(\boldsymbol{E}_d^j)^R &= \begin{bmatrix} (\boldsymbol{B}^j - \sum_{i \neq d}^{K^j} \boldsymbol{D}_i^j(\boldsymbol{x}_i^j)^T)^R \\ (\sqrt{\alpha}\boldsymbol{C}^j - \sum_{i \neq d}^{K^j} \sqrt{\alpha}\boldsymbol{A}_i^j(\boldsymbol{x}_i^j)^T)^R \\ (\sqrt{\beta}\boldsymbol{H}^j - \sum_{i \neq d}^{K^j} \sqrt{\beta}\boldsymbol{W}_i^j(\boldsymbol{x}_i^j)^T)^R \end{bmatrix} \\ &= \begin{bmatrix} U_A^j & 0 \\ 0 & U_B^j \end{bmatrix} \begin{bmatrix} S_A^j & 0 \\ 0 & S_B^j \end{bmatrix} \begin{bmatrix} V_A^j \\ V_B^j \end{bmatrix} \\ &= \boldsymbol{U}^j\boldsymbol{\Delta}^j\boldsymbol{V}^j = \sum_{i=1}^{m} \boldsymbol{u}_i^j \cdot \boldsymbol{\sigma}_i^j(\boldsymbol{v}_i^j)^T, \quad (32)\end{aligned}$$

where it is satisfied that $(\boldsymbol{B}^j - \sum_{i \neq d}^{K^j} \boldsymbol{D}_i^j(\boldsymbol{x}_i^j)^T)^R = U_A^j S_A^j (V_A^j)^T$, and $[(\sqrt{\alpha}\boldsymbol{C}^j - \sum_{i \neq d}^{K^j} \sqrt{\alpha}\boldsymbol{A}_i^j(\boldsymbol{x}_i^j)^T)^R; (\sqrt{\beta}\boldsymbol{H}^j - \sum_{i \neq d}^{K^j} \sqrt{\beta}\boldsymbol{W}_i^j(\boldsymbol{x}_i^j)^T)^R] = U_B^j S_B^j (V_B^j)^T$.

The representation error matrix $\bar{\boldsymbol{E}}_d^j$ of Eq. (14) is given by

$$\bar{\boldsymbol{E}}_d^j = \bar{\boldsymbol{Z}}^j - \sum_{i \neq d}^{K^j} \bar{\boldsymbol{t}}_i^j(\boldsymbol{x}_i^j)^T, \quad (33)$$

where $\bar{\boldsymbol{t}}_i^j$ represents the i-th column in $\bar{\boldsymbol{T}}^j$.

To apply SVD to minimize the l^2-norm of $\bar{\boldsymbol{E}}_d^j$,

$$\begin{aligned}(\bar{\boldsymbol{E}}_d^j)^R &= \begin{bmatrix} (\boldsymbol{Q}_p\boldsymbol{B}^j - \sum_{i \neq d}^{K^j} \bar{\boldsymbol{D}}_i^j(\boldsymbol{x}_i^j)^T)^R \\ (\sqrt{\alpha}\boldsymbol{C}^j - \sum_{i \neq d}^{K^j} \sqrt{\alpha}\bar{\boldsymbol{A}}_i^j(\boldsymbol{x}_i^j)^T)^R \\ (\sqrt{\beta}\boldsymbol{H}^j - \sum_{i \neq d}^{K^j} \sqrt{\beta}\bar{\boldsymbol{W}}_i^j(\boldsymbol{x}_i^j)^T)^R \end{bmatrix} \\ &= \begin{bmatrix} Q_p U_A^j & 0 \\ 0 & U_B^j \end{bmatrix} \begin{bmatrix} S_A^j & 0 \\ 0 & S_B^j \end{bmatrix} \begin{bmatrix} V_A^j \\ V_B^j \end{bmatrix} \\ &= \bar{\boldsymbol{U}}^j\bar{\boldsymbol{\Delta}}^j\bar{\boldsymbol{V}}^j = \sum_{i=1}^{m} \bar{\boldsymbol{u}}_i^j \cdot \bar{\boldsymbol{\sigma}}_i^j(\bar{\boldsymbol{v}}_i^j)^T. \quad (34)\end{aligned}$$

In this case,

$$\left(\boldsymbol{Q}_p\boldsymbol{B}^j - \sum_{i \neq d}^{K^j} \bar{\boldsymbol{D}}_i^j(\boldsymbol{x}_i^j)^T\right)^R = \boldsymbol{Q}_p U_A^j S_A^j (V_A^j)^T, \quad (35)$$

Because $\boldsymbol{D}_i^j$ is the optimal minimizer for $(\boldsymbol{B}^j - \sum_{i \neq d}^{K^j} \boldsymbol{D}_i^j(\boldsymbol{x}_i^j)^T)^R$ given $\boldsymbol{B}^j$ and $\boldsymbol{x}_i^j$, which is a linear problem. When $\boldsymbol{B}^j$ is multiplied by $\boldsymbol{Q}_p$, according to the scaling property, we have $\overline{\boldsymbol{D}}_i^j = \boldsymbol{Q}_p\boldsymbol{D}_i^j$. Moreover, it is satisfied that

$$\begin{bmatrix} (\sqrt{\alpha}\boldsymbol{C}^j - \sum_{i\neq d}^{K^j} \sqrt{\alpha}\overline{\boldsymbol{A}}_i^j(\boldsymbol{x}_i^j)^T)^R \\ (\sqrt{\beta}\boldsymbol{H}^j - \sum_{i\neq d}^{K^j} \sqrt{\beta}\overline{\boldsymbol{W}}_i^j(\boldsymbol{x}_i^j)^T)^R \end{bmatrix} = U_B^j S_B^j (V_B^j)^T. \quad (36)$$

Because observing that 1). $\boldsymbol{x}_i^j$ is the optimal minimizer for both $(\boldsymbol{Q}_p\boldsymbol{B}^j - \sum_{i\neq d}^{K^j} \boldsymbol{Q}_p\boldsymbol{D}_i^j(\boldsymbol{x}_i^j)^T)^R$ and $(\boldsymbol{B}^j - \sum_{i\neq d}^{K^j} \boldsymbol{D}_i^j(\boldsymbol{x}_i^j)^T)^R$. 2). $\boldsymbol{A}_i^j$ and $\boldsymbol{W}_i^j$ are the optimal minimizer for $[(\sqrt{\alpha}\boldsymbol{C}^j - \sum_{i\neq d}^{K^j} \sqrt{\alpha}\boldsymbol{A}_i^j(\boldsymbol{x}_i^j)^T)^R; (\sqrt{\beta}\boldsymbol{H}^j - \sum_{i\neq d}^{K^j} \sqrt{\beta}\boldsymbol{W}_i^j(\boldsymbol{x}_i^j)^T)^R]$ given $\boldsymbol{x}_i^j$. It is satisfied that $\overline{\boldsymbol{A}}_i^j = \boldsymbol{A}_i^j$ and $\overline{\boldsymbol{W}}_i^j = \boldsymbol{W}_i^j$ when minimizing the l^2-norm of $\overline{\boldsymbol{E}}_d^j$ given $\tilde{\boldsymbol{x}}_i^j = \boldsymbol{x}_i^j$.

Therefore,

$$\begin{aligned} \overline{\boldsymbol{u}}_1^j &= Q_p\boldsymbol{u}_1^j \\ \overline{\boldsymbol{u}}_{K^j+1}^j &= \boldsymbol{u}_{K^j+1}^j \\ \overline{\boldsymbol{\Delta}}^j &= \boldsymbol{\Delta}^j \\ \overline{\boldsymbol{V}}^j &= \boldsymbol{V}^j, \end{aligned} \quad (37)$$

and update the dictionary, classifier and coefficient vector as

$$\begin{aligned} \overline{\boldsymbol{D}}_d^j &= \overline{\boldsymbol{u}}_1^j = Q_p\boldsymbol{u}_1^j = Q_p\boldsymbol{D}_d^j \\ <\overline{\boldsymbol{A}}_d^j, \overline{\boldsymbol{W}}_d^j> &= \overline{\boldsymbol{u}}_{K^j+1}^j = \boldsymbol{u}_{K^j+1}^j = <\boldsymbol{A}_d^j, \boldsymbol{W}_d^j> \\ \tilde{\boldsymbol{x}}_d^j &= \overline{\boldsymbol{\sigma}}_1^j(\overline{\boldsymbol{v}}_1^j)^T = \boldsymbol{\sigma}_1^j(\boldsymbol{v}_1^j)^T = \boldsymbol{x}_d^j, \end{aligned} \quad (38)$$

which finishes the proof.

REFERENCES

[1] Y. Wang and T. Nakachi, "Secure face recognition in edge and cloud networks: From the ensemble learning perspective," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 1–5.

[2] Y. Duan, J. Lu, J. Feng, and J. Zhou, "Context-aware local binary feature learning for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1139–1153, May 2018.

[3] Y. Xu, Z. Li, J. Yang, and D. Zhang, "A survey of dictionary learning algorithms for face recognition," *IEEE Access*, vol. 5, pp. 8502–8514, 2017.

[4] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2691–2698.

[5] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 543–550.

[6] M. M. Kasar, D. Bhattacharyya, and T. H. Kim, "Face recognition using neural network: A review," *Int. J. Secur. Appl.*, vol. 10, no. 3, pp. 81–100, Oct. 2016.

[7] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," *Proc. CVPR*, Jun. 2014, pp. 1891–1898.

[8] R. Ranjan, V. M. Patel, and R. Chellappa, "HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 121–135, Jan. 2019.

[9] F. Wu, X.-Y. Jing, X. Dong, R. Hu, D. Yue, L. Wang, Y.-M. Ji, R. Wang, and G. Chen, "Intraspectrum discrimination and interspectrum correlation analysis deep network for multispectral face recognition," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1009–1022, Mar. 2020.

[10] F. textcolor black Wu, X. Y. Jing, S. Wu, G. Gao, Q. Ge, and R. Wang, "Like charges repulsion opposite charges attraction' law based multilinear subspace Anal. for face recognition," *Knowl.-Based Syst.*, vol. 149, pp. 76–87, Jun. 2018.

[11] F. Wu, X.-Y. Jing, X. Dong, Q. Ge, S. Wu, Q. Liu, D. Yue, and J.-Y. Yang, "Uncorrelated multi-set feature learning for color face recognition," *Pattern Recognit.*, vol. 60, pp. 630–646, Dec. 2016.

[12] X. Fan, Y. Huang, X. Ma, J. Liu, and V. C. M. Leung, "Exploiting the edge power: An edge deep learning framework," *CCF Trans. Netw.*, vol. 12, pp. 1–8, Dec. 2018.

[13] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[14] S. Teerapittayanon, B. McDanel, and H. T. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 328–339.

[15] P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang, and X. Yao, "Security and privacy preservation scheme of face identification and resolution framework using fog computing in Internet of Things," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1143–1155, Oct. 2017.

[16] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-Vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *Proc. CC*, Jul. 2012, pp. 1–6.

[17] R. B. Romdhane, H. Hammami, M. Hamdi, and T.-H. Kim, "At the cross roads of lattice-based and homomorphic encryption to secure data aggregation in smart grid," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 1067–1072.

[18] Z. Ma, Y. Liu, X. Liu, J. Ma, and K. Ren, "Lightweight privacy-preserving ensemble classification for face recognition," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5778–5790, Jun. 2019.

[19] M. Z. Khan, S. Harous, S. U. Hassan, M. U. Ghani Khan, R. Iqbal, and S. Mumtaz, "Deep unified model for face recognition based on convolution neural network and edge computing," *IEEE Access*, vol. 7, pp. 72622–72633, 2019.

[20] X. Wang, H. Xue, X. Liu, and Q. Pei, "A privacy-preserving edge computation-based face verification system for user authentication," *IEEE Access*, vol. 7, pp. 14186–14197, 2019.

[21] Y. Li, Y. Wang, and D. Li, "Privacy-preserving lightweight face recognition," *Neurocomputing*, vol. 363, pp. 212–222, Oct. 2019.

[22] Y. Chen and J. Su, "Sparse embedded dictionary learning on face recognition," *Pattern Recognit.*, vol. 64, pp. 51–59, Apr. 2017.

[23] W. Ou, X. You, D. Tao, P. Zhang, Y. Tang, and Z. Zhu, "Robust face recognition via occlusion dictionary learning," *Pattern Recognit.*, vol. 47, no. 4, pp. 1559–1572, Apr. 2014.

[24] P. Zhang, X. You, W. Ou, C. L. Philip Chen, and Y.-M. Cheung, "Sparse discriminative multi-manifold embedding for one-sample face identification," *Pattern Recognit.*, vol. 52, pp. 249–259, Apr. 2016.

[25] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2510–2523, Dec. 2015.

[26] J. Wright, A. Y. Yang, A. Ganesh, S. Shankar Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[27] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[28] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 1697–1704.

[29] I. Nakamura, Y. Tonomura, and H. Kiya, "Unitary transform-based template protection and its application to l2-norm minimization problems," *IEICE Trans. Inf. Syst.*, vol. E99.D, no. 1, pp. 60–68, 2016.

[30] T. Maekawa, T. Nakachi, S. Shiota, and H. Kiya, "Privacy-preserving SVM computing by using random unitary transformation," in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst. (ISPACS)*, Nov. 2018, pp. 1–5.

[31] S.-H. Hsieh, C.-S. Lu, and S.-C. Pei, "Fast OMP: Reformulating OMP via iteratively refining ℓ_2-norm solutions," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, Aug. 2012, pp. 189–192.

[32] N. Qi, Y. Shi, X. Sun, and B. Yin, "TenSR: Multi-dimensional tensor sparse representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5916–5925.

[33] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ, USA: Wiley, 2004.
[34] L. Tian, C. Fan, Y. Ming, and Y. Jin, ''Stacked PCA network (SPCANet): An effective deep learning for face recognition,'' in *Proc. IEEE Int. Conf. Digit. Signal Process. (DSP)*, Jul. 2015, pp. 1039–1043.
[35] D. Cai, X. He, J. Han, and H. J. Zhang, ''Orthogonal Laplacian faces for face recognition,'' *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3608–3614, Oct. 2006, doi: 10.1109/TIP.2006.881945.
[36] Y. Wang, T. Nakachi, and H. Ishihara, ''Edge and cloud-aided secure sparse representation for face recognition,'' in *Proc. 27th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2019, pp. 1–5.
[37] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, ''From few to many: Illumination cone models for face recognition under variable lighting and pose,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.
[38] K.-C. Lee, J. Ho, and D. J. Kriegman, ''Acquiring linear subspaces for face recognition under variable lighting,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.
[39] A. Martinez and R. Benavente, ''The AR face database,'' CVC Tech. Rep., Jun. 1998.
[40] Y. Liu, S. Zhao, Q. Wang, and Q. Gao, ''Learning more distinctive representation by enhanced PCA network,'' *Neurocomputing*, vol. 275, pp. 924–931, Jan. 2018.
[41] Z. An, W. Deng, Y. Zhong, Y. Huang, and X. Tao, ''APA: Adaptive pose alignment for robust face recognition,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1–9.
[42] T. Nakachi and H. Kiya, ''Practical secure OMP computation and its application to image modeling,'' *Proc. ACM ICIHIP*, Sep. 2018, pp. 25–29.

YITU WANG (Student Member, IEEE) received the B.S. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2013 and 2018, respectively. From August 2014 to November 2014, he was a Visiting Student with the University of Paris-Sud, Orsay, France. He is currently a Researcher at the NTT Network Innovation Laboratory, NTT Corporation, Japan. His research interests include communication networks and artificial intelligence.

TAKAYUKI NAKACHI (Member, IEEE) received the Ph.D. degree in electrical engineering from Keio University, Tokyo, Japan, in 1997. Since joining Nippon Telegraph and Telephone Corporation (NTT), 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. He also actively participates in MPEG international standardization activities. His current research interests include communication science, information theory, and signal processing. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE), Japan. He received the 26th TELECOM System Technology Award, the 6th Paper Award of *Journal of Signal Processing* and the Best Paper Award of the IEEE ISPACS 2015.