IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.2020/ACCESS.2020.DOI

Wi-Fi based User Identification using In-air Handwritten Signature

JUNSIK JUNG[1], HAN-CHEOL MOON[2], JOOYOUNG KIM[3], DONGHYUN KIM[3] AND KAR-ANN TOH[3] (Senior Member, IEEE).

[1]Korea Advanced Institute of Science and Technology, School of Computing, Daejeon, Republic of Korea (e-mail: junsik.jung@kaist.ac.kr)
[2]Nanyang Technological University, Singapore (e-mail: hancheol001@e.ntu.edu.sg)
[3]Yonsei University, School of Electrical and Electronic Engineering, Republic of Korea (harrykim@yonsei.ac.kr, kimd@yonsei.ac.kr, katoh@yonsei.ac.kr)

Corresponding author: Kar-Ann Toh.

The authors are thankful to the editor and the anonymous reviewers for their constructive comments to improve the manuscript. This work was performed when Junsik Jung and Han-Cheol Moon were with Yonsei University. This research was supported by the National Research Foundation of Korea (NRF) under the program of Basic Research Laboratory (BRL) (NRF-2019R1A4A1025958).

ABSTRACT This paper conducts a feasibility study regarding the use of the Wi-Fi channel state information for user recognition based on in-air handwritten signatures. A novel system for identity recognition is thus proposed to observe for distinctive signal distortions along the propagation path for different users. The system capitalizes on the vast availability of Wi-Fi signals for signal analysis without needing additional hardware infra-structure. Since the patterns of the raw Wi-Fi signals are sensitive to the signer's location, a transfer learning has been adopted to cope with the positional variation. Specifically, features trained at one position are transferred to classify signals collected at another position via a single shot retraining. A kernel and range space projection has been adopted for the single shot retraining. Our experiments show encouraging results for the proposed system.

INDEX TERMS Biometrics, Wi-Fi, CSI, In-air signature recognition, Transfer learning, Kernel and range space projection, Score-level fusion.

I. INTRODUCTION

According to [1]–[3], the biometric traits have played a central role for user identity authentication in the networked society. Generally, biometric based systems can be divided into two categories depending on the adopted information: 1. one that based on the user's physiological characteristics such as face, fingerprint, palmprint, iris, and 2. one that utilizes the user's behavioral traits such as signature, gait, and keystroke. Comparing with the physiological characteristics based systems, the behavioral traits based systems are more robust to the spoofing attack because the physiological characteristics could be easily cloned or imitated due to their static nature [4], [5].

Among those behavioral traits, the handwritten signature has been widely used as means for large scale person identification [6], [7]. The handwritten signature uses personalized patterns while the hand gestures include more general patterns such as waving [8]. Hence, the handwritten signature can be considered as one particular type of hand gestures. The handwritten signature has several advantages over other behavioral biometrics. These advantages include 1. the acquisition procedure is non-invasive. 2. signatures are replaceable even if the biometric information is stolen or compromised. For example, it is very challenging to have the gait replaced by a new walking pattern upon compromise since the walking pattern has become a habit throughout one's life. In contrast, the handwritten signature can overcome this issue by creating a new signature pattern to replace the compromised signature. Systems which adopt the handwritten signatures can be classified into two categories according to the signature acquisition method. They are namely, those which utilized the offline signatures and those which used the online signatures [4], [6]. The offline signature can be acquired by having a signing instrument directly in contact with a scanning device. This method has several problems compared with the online signature. Firstly, it is vulnerable to imitation because it leaves a traceable print on the surface [4]. Secondly, it contains fewer features than online approaches. For example, online signature contains dynamic information such as signing velocity, degree of pressure, and degree of tilt while the offline signature only contains the shape information [4]. Finally, it could distract the user activities in practice. A user has to stop what he/she was doing and move the focus onto the system to write his/her signature on the restricted surface where the sensing module is placed. Moreover, setting up the contact-based sensing modules takes up space and could be costly in practice.

In view of the above advantages of the online signature systems, the online based in-air signature recognition systems have been gaining attention recently [4], [9]–[13]. These systems were designed to recognize the user's signature that was written in the air. In [9], the in-air hand signature was recognized by the trajectory image of the hand through a visual sensor. In [10], a movement sensor in mobile phone was utilized to capture the in-air signature. Although these systems could avoid the costly setup issue by utilizing built-in devices, these systems either require the users to hold a certain type of device (e.g., mobile phone) or standing towards a specific direction where the sensing module is placed during the authentication process.

Recent studies present a variety of potential applications of the Wi-Fi signal based human activity recognition system. This includes the human fall detection [14], [15], keystroke recognition [16], respiration monitoring [17], etc. The fundamental mechanism that forms the basis of these systems is the identifiable interference caused by the human activities along the Wi-Fi signal propagation. Utilizing the raw Wi-Fi signals for activity recognition has several characteristics. Firstly, the system is able to utilize the widely available Wi-Fi devices as its platform. This helps to alleviate the struggle associated with implementing the system in new devices where an immediate use of the widely available Wi-Fi devices becomes possible. Secondly, the system does not require the user to hold or wear any additional device or sensor. Thirdly, the system is able to identify human activities from various locations and directions without needing to position the sensors at specific locations. Finally, the Wi-Fi based system is robust to illumination conditions compared with vision based systems [17].

In this work, we propose to utilize the widely available Wi-Fi signals for user identification based on the in-air handwritten signature. As the user identification/recognition consists of N number of one-to-one matching, the verification accuracy is also included to supplement the recognition performance. Here we note that the adoption of Wi-Fi signals for human activity recognition poses new challenges. For example, the Wi-Fi signals can be disturbed by distractors such as the presence of movements other than the desired signature patterns. Moreover, apart from contamination by noises, the Wi-Fi signals can be attenuated physical distance and barriers. Lastly, different from the consistency of offline signature images acquired from contact based devices [9], the Wi-Fi based system is observed to be sensitive to the position of objects that reflect or interfere with the Wi-Fi Channel State Information (CSI) signals. When the same person draws an in-air signature at different locations, the captured Wi-Fi signal shows different patterns (see Figure 1). Our task in this work is to establish the feasibility and then design a system for user recognition.

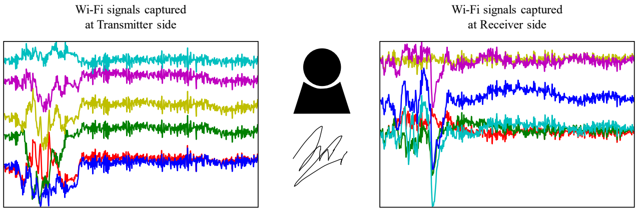


FIGURE 1. Samples of captured Wi-Fi signals at different positions from the same user. These signals from the same user show different patterns.

Our proposed system addresses the last challenge by adopting transfer learning [18]. Instead of training from scratch at each position, we firstly train a Convolutional Neural Network (CNN) [19] using the Wi-Fi signature signals collected from one position. Subsequently, the trained feature extractor is transferred to recognize the Wi-Fi signature signals collected at another position. In view of the time consuming retraining process by the gradient descent (GD) algorithm [20], [21], we adopt a fast single shot algorithm for fine tuning the transferred features. The single shot training algorithm, called Kernel and the Range (KAR) space projection learning [22]–[25], shows an impressive learning speed for optimizing relatively shallow networks. Our experiments on the proposed system show up to 99.875% identification accuracy. Apart from the pioneering effort in addressing the issue of positional sensitiveness of Wi-Fi based in-air signature for user identification, the proposed system delivers a reasonable prediction output based on a relatively small training dataset. This is an important advantage over many deep learning methods which require a large pool of data for effective learning.

The main distinctive attributes of our contribution are as follows:

- Upon establishing the feasibility, we propose a system for user identification based on the Wi-Fi handwritten signature signals. The system can leverage on existing Wi-Fi system for signal acquisition.
- We address the positional sensitiveness of Wi-Fi based in-air handwritten signature signals by adopting transfer learning. A single shot learning algorithm based on kernel and range space projection has been adopted for fine tuning the transferred features for computational efficiency. A score-level fusion is subsequently employed for accuracy enhancement.
- In view of the lack of public datasets, we have constructed a dataset of 100 subjects for experimentation.

The paper is organized as follows: Section II provides a brief review of related works. Section III introduces our proposed system for user identity recognition. In Section IV we present our experimental settings and evaluation of the system. Concluding remarks are presented in Section V.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3071228, IEEE Access

TABLE 1. Overview of existing literature for in-air signature verification and identification.

	(a) Vision/handheld-device based	(b) Wi-Fi based
1) 2D in-air signature	i) Handcrafted feature based systems: Katagiri et al. [26], Sajid et al. [27], Khoh et al. [4]. ii) Deep learning based systems: Nil.	i) Handcrafted feature based systems: Moon et al. [11]. ii) Deep learning based systems: Our preliminary work in conference [28]. **Our extended work in this paper.**
2) 3D in-air signature	i) Handcrafted feature based systems: Casanova et al. [29], Bailador et al. [10], Jeon et al. [9], Diep et al. [30], Fang et al. [31], Malik et al. [13] ii) Deep learning based systems: Behera et al. [12].	i) Handcrafted feature based systems: Nil. ii) Deep learning based systems: Nil.

II. RELATED WORKS

A. IN-AIR HUMAN SIGNATURE RECOGNITION

According to [2], the problem of identity recognition using biometrics (e.g., signature) can be divided into two types based on their operation. The first type is known as identification (also known as recognition) and the second type is known as verification (also known as authentication). Identification is a 1-to-n matching problem, which means that the given individual biometric template is compared with all others in the database to locate the identity. In contrast, verification is a 1-to-1 matching problem, which means that the given biometric template is matched against a specific biometric template to see whether it is a match or a non-match. In other words, the available answer to verification is either "match" or "non-match" (or am I who I claim I am?). For identification, the answer to the 1-to-n is "who" is the owner of the biometric template (or who am I?). Table 1 provides an overview of existing literature for in-air signature recognition systems where their modes of operation (verification or identification) can be found in Table 2. Although these systems have different measurements in their respective setups, their performances can nevertheless be quantified by a common measure of either the accuracy of identification or the Equal Error Rate (EER) of verification [8], [32].

As seen from Table 1, based on the signal capturing method, these works on handwritten signatures can be categorized into two groups, namely (a) vision/handheld-device based and (b) Wi-Fi based systems. The vision/handheld-device based systems utilized either imaging sensors such as RGB, NIR, ToF sensors or a handheld device for signature motion capture or imaging. The Wi-Fi based systems utilized either the CSI or the RSSI signals for capturing of signature patterns. Based on the captured signature type, the related works can be further classified into two categories, namely 1) 2D signature and 2) 3D signature. In terms of the adopted methodology, we further divide the related works into i) handcrafted feature based systems and ii) deep learning based systems. Based on this categorization, we observed that most of the earlier works utilized handcrafted features for in-air signatures while only a few recent works used deep learning technique where the feature extraction was performed in an end-to-end manner. Our proposed system belongs to the category of 2D handwritten signature recognition utilizing the Wi-Fi CSI signals. Table 2 provides greater details of existing literature in terms of adopted modality, data type, methodology and the mode of operation. The following subsections provide a brief account of these existing technologies.

1) Vision/handheld-device Based Systems

In [26], Katagiri et al. conducted a preliminary study of in-air handwritten signature for person verification. In their system, the user was prompted to draw his/her signature in the air using a pen that came with a light-emitting diode (LED). The highlighted trajectory of the in-air signature was subsequently captured by a video camera. In [10], Bailador et al. made use of the movement sensor in the mobile device to capture in-air signatures. They also conducted experiments under spoofing attack scenarios. Similarly, in [29] and [30], an accelerometer embedded in the mobile device was utilized to perform the verification task. Casanova et al. achieved an EER of 2.5% by fusing the features of gesture accelerations at decision level. In [30], Diep et al. achieved an EER of 0.8% by using the Support Vector Machine (SVM) to differentiate between the genuine users and the impostors.

In [9], Jeon et al. proposed a vision based in-air signature verification system. The system utilized a depth camera (Kinect) to track the finger together with the palm motion of the user. Unlike [10], [26], this system did not require a person to hold any additional devices. In [27], Sajid et al. proposed a vision based in-air signature verification system. In this work, the Google-Glass was used to recording in-air handwritten signatures. The extracted 2D coordinates are matched and authenticated by Dynamic Time Warping (DTW). In [31], Fang et al. proposed a video-based in-air signature verification system. The time and space information of in-air signature were fused by DTW and Fast Fourier Transform (FFT), and this fusion resulted in a robust and accurate verification system. In [13], Malik et al. utilized a depth sensor to capture the 3D hand joint positions from in-air handwritten signatures. They employed a Multi-dimensional DTW algorithm to match between the preprocessed test signature and the corresponding feature. Behera et al. [12] proposed a deep learning based matching system. They employed a CNN based sequential classifier to perform the identification task on the 3D in-air signature dataset captured by a leap motion sensor. Recently, Khoh et

TABLE 2. Existing works sorted according to their adopted modality, data type, methodology and operation modes.

Reference	Modality	Data type (#subjects /#samples)	Methodology	Operation mode
(a) Vision/handheld-device based previous work				
Katagiri et al. [26]	Pen with a LED, video camera	2D in-air signature (105/ 315)	Commercial signature verification engine	Verification
Bailador et al. [10]	Accelerometer	3D in-air signature (96 / 768)	HMM[1], Bayes classifiers and DTW[2]	Verification
Casanova et al. [29]	Accelerometer	3D in-air signature (34 / 942)	Euclidean distance, DTW	Verification
Jeon et al. [9]	Kinect camera	3D in-air signature (50 / 500)	Euclidean distance, DTW	Verification
Sajid et al. [27]	Google Glass	2D in-air signature (10 / 96)	Euclidean distance, DTW	Verification
Diep et al. [30]	Motion sensor	3D in-air signature (30 / 1800)	SVM[3]	Verification
Fang et al. [31]	High-speed camera	3D in-air signature (14 / 560)	FFT[4], DTW	Verification
Behera et al. [12]	Leap motion sensor	3D in-air signature (50 / 700)	CNN[5]	Identification
Malik et al. [13]	Depth sensor	3D in-air signature (15 / 600)	MD-DTW[6]	Verification
Khoh et al. [4]	Kinect camera	2D in-air signature (100 / 2000)	HOG[7], Euclidean distance	Both
(b) Wi-Fi based previous work				
Moon et al. [11]	Wi-Fi (CSI)[5]	1D in-air signature (13 / 1040)	Euclidean distance, PCA[8]	Verification
Jung et al. [28]	Wi-Fi (CSI)	2D in-air signature (50 / 4000)	CNN	Identification
Ours	Wi-Fi (CSI)	1D&2D in-air signature (100 / 8000)	CNN	Both

[1] Hidden Markov Model (HMM)
[2] Dynamic Time Warping (DTW)
[3] Support Vector Machine (SVM)
[4] Fast Fourier Transform (FFT)
[5] Convolutional Neural Network (CNN)
[6] Multi Directional DTW (MD-DTW)
[7] Histogram of Oriented Gradient (HOG)
[8] Principal Component Analysis (PCA)

al. [4] proposed an in-air signature recognition system using the Kinect camera to capture not only the RGB images but also the depth information. They utilized several distance measures to compute the dissimilarities among the extracted features. Both the identification mode and the verification mode were experimented to show the potential of the system.

2) Wi-Fi Based Systems

Comparing with the vision/handheld-device based systems mentioned above, the Wi-Fi based in-air signature recognition system has less spatial constraint due to the vast accessibility of the Wi-Fi signal via commercial devices. Recently, Moon et al. [11] constructed a preliminary in-air signature recognition system for user verification. The system was able to authenticate in-air signatures which were captured through the Wi-Fi CSI signals.

The above review shows that none of the previous systems are designed to address the location sensitivity issue of Wi-Fi based systems except the gesture recognition work of Ohara et al. [33] where the Doppler shift effect was adopted to extract position-independent features. However, only features parallel to the direction of the Wi-Fi signals can account for the position change. The current paper reports an extension of the preliminary study in [28] on an extended dataset from [11], [28] using 100 subjects and incorporates score-level fusion of multi-samples (i.e., 1-dimensional, 2-dimensional Wi-Fi signals) for performance enhancement.

B. TRANSFER LEARNING

When there is insufficient data for training a deep network, the technique of transfer learning [18], [34] can be exploited to adapt the features of a pretrained network to another under the similar domain. In [35], Oquab et al. proposed a CNN which was pretrained using available labeled source and then used it as a fixed feature extractor. To address the distribution difference between the source used for pretraining and the novel target domain, an adaptation layer was utilized to fine tune the network to the novel target. In [36], Shin et al. transferred an ImageNet [37] pretrained CNN and then retrain the network using a sub-domain of medical images to perform skin lesion detection. They showed that this approach can effectively address the problem of insufficient medical images and achieved the state-of-the-art performance. Wang et al. [38] proposed an online training method to improve the visual tracking performance. They transferred deep features that was trained on a large visual dataset and was shown to remarkably improve the performance on the small dataset. Our proposed system has been inspired by these works. Since our in-air Wi-Fi handwritten signatures captured at different positions and orientations belong to a common domain, we can utilize transfer learning to address the limited number of samples acquired from a single orientation at each position.

C. MOORE-PENROSE INVERSE BASED NETWORK LEARNING

The backpropagation algorithm using gradient descent [20], [21] has achieved great success in deep learning. However, gradient descent based learning algorithm comes at the expense of several costs such as selection of hyperparameters, having local minima with time consuming iterations, and gradient vanishing. The Moore-Penrose inverse based learning for training the shallow (less than three layers) networks has been recently reviewed in [39]. This approach aims to find the global minima of the reformulated system in a single learning shot without training iterations. Moreover, it does not require hyper-parameters such as the learning rate and the momentum setting. However, the learning has been limited to shallow networks. Recently, a novel method to learn neural networks with more than three layers was seen in [22]–[25]. This learning approach aims to learn all the parameters of the deep neural network in a single operating pass by manipulating the KAR space of the reformulated system. We shall utilize this learning method in the stage of retraining to avoid iterative search. The details shall be provided in Section

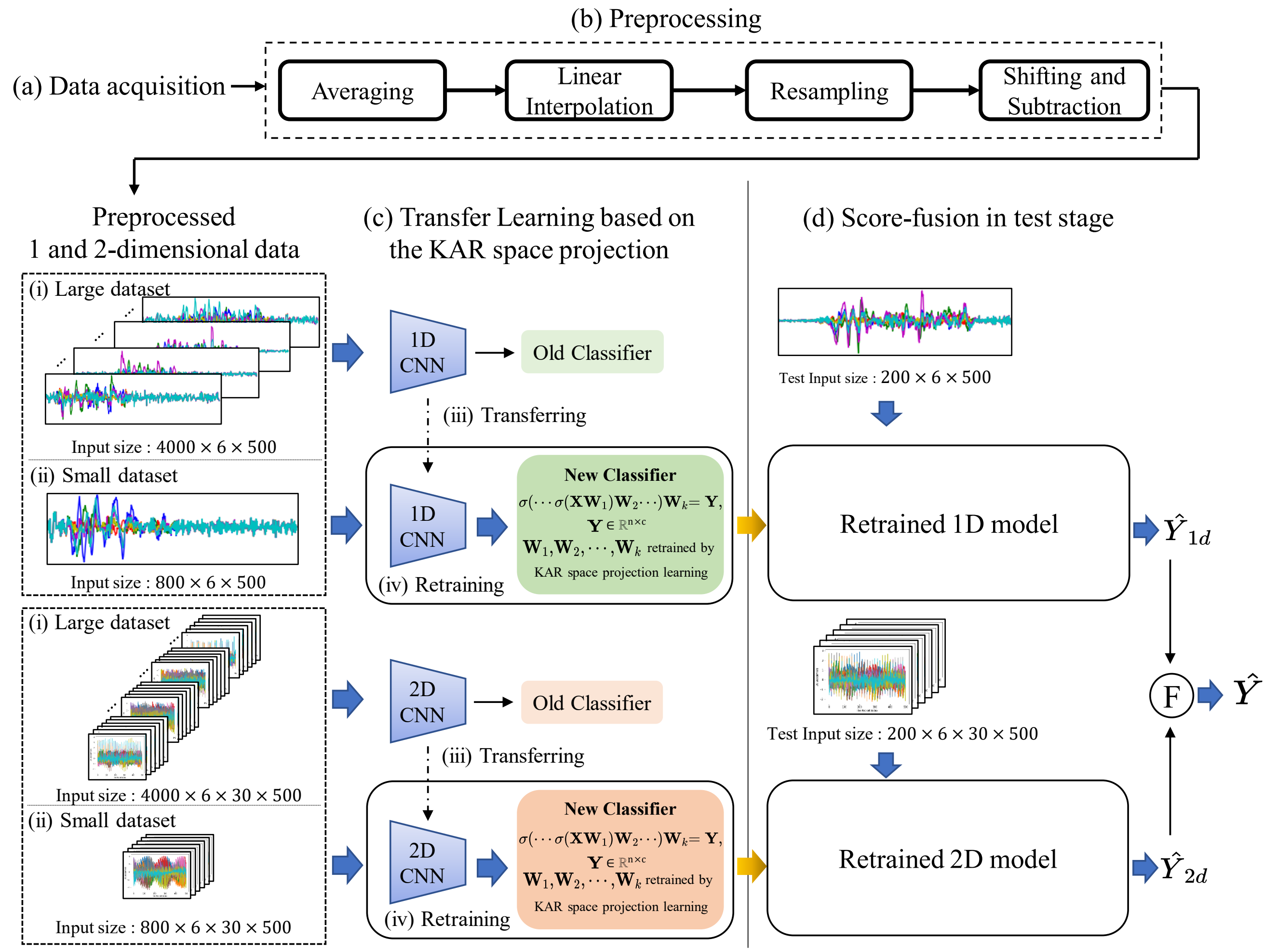

(b) Preprocessing
(a) Data acquisition
Averaging
Linear Interpolation
Resampling
Shifting and Subtraction
Preprocessed 1 and 2-dimensional data
(c) Transfer Learning based on the KAR space projection
(d) Score-fusion in test stage
(i) Large dataset
Input size : 4000 × 6 × 500
(ii) Small dataset
Input size : 800 × 6 × 500
1D CNN
Old Classifier
(iii) Transferring
New Classifier
$\sigma(\cdots\sigma(\mathbf{X}\mathbf{W}_1)\mathbf{W}_2\cdots)\mathbf{W}_k = \mathbf{Y}$,
$\mathbf{Y} \in \mathbb{R}^{n\times c}$
$\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_k$ retrained by
KAR space projection learning
(iv) Retraining
Test Input size : 200 × 6 × 500
Retrained 1D model
$\hat{Y}_{1d}$
(i) Large dataset
Input size : 4000 × 6 × 30 × 500
(ii) Small dataset
Input size : 800 × 6 × 30 × 500
2D CNN
Old Classifier
(iii) Transferring
New Classifier
$\sigma(\cdots\sigma(\mathbf{X}\mathbf{W}_1)\mathbf{W}_2\cdots)\mathbf{W}_k = \mathbf{Y}$,
$\mathbf{Y} \in \mathbb{R}^{n\times c}$
$\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_k$ retrained by
KAR space projection learning
(iv) Retraining
Test Input size : 200 × 6 × 30 × 500
Retrained 2D model
$\hat{Y}_{2d}$
F
$\hat{Y}$

FIGURE 2. Overview of the proposed system. (a) Capture of in-air handwritten signature gesture using Wi-Fi CSI signals (b) Data preprocessing steps (c) User identification process using transfer learning based on the KAR space projection learning (d) Score-fusion in the testing stage for accuracy enhancement. The symbol “F" refers to fusion. Each of these processing steps is described in greater details in Section III.

III-B.

III. PROPOSED SYSTEM

In this section, we introduce our system for user identification where its flow diagram is shown in Figure 2. Essentially, the processing steps in the system can be divided into three stages namely, i) data acquisition and preprocessing (see labels (a) and (b) in the figure), ii) network pretraining and transfer learning (see label (c) in the figure), and iii) decision fusion (see label (d) in the figure). These processing stages are described in greater detail in the following subsections. The abbreviations and acronyms in this section are explained in Table 3.

TABLE 3. Meaning of abbreviations/acronyms in Section III.

Section	Abbreviation/Acronym	Meaning
III-A	NIC CSI DC component LTS Tx, Rx	Network Interface Controller Channel State Information Mean amplitude of the signal Long Term Supported Transmitter, Receiver
III-B	KAR learning GD algorithm	Kernel and Range space projection learning Gradient Descent algorithm
III-C	SVM TER	Support Vector Machine Total Error Rate minimization

A. DATA ACQUISITION AND PREPROCESSING

1) Device and Environmental Setup

A VPCSB16FK VAIO laptop with Intel Core i5-2410M 2.3GHz 64bit CPU and 8Gb memory was used as the Wi-Fi signal receiver. The receiver was equipped with the Linux 802.11n CSI tool [40] and an Intel 5300 Network Interface Controller (NIC) to collect the CSI of the signals. The aforementioned tool was implemented on an Ubuntu 12.04.05 LTS laptop. The receiver was set to receive each CSI at every 0.0001 second (10kp/s) in view of higher packet loss at lower packet rates as well as the relatively long range

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3071228, IEEE Access

between the transceivers in our setting [1] [41]–[43]. In order to gather as many packets as possible in multiple paths, 3 external antennas at the receiver side were deployed. An IpTime A1004 router with 2 transmission antennas was used as the Wi-Fi signal transmitter. The transmitter was set to transmit Wi-Fi signals at 2.4 GHz frequency band. To the best of our knowledge, the primary difference between the 2.4GHz and 5GHz signals are the speed and the range of coverage. A 2.4GHz Wi-Fi signal is able to cover a larger area but it sacrifices the speed. In contrast, the 5GHz band provides a faster speed for a smaller area. We observed higher noise in the collected signals via the 5GHz band than that via the 2.4GHz band during the data collection process. In view of the noise issue, we have utilized the 2.4GHz signals in this study. Under this device setup, subjects were requested to write their signatures in the air while sitting at two different positions with four writing directions (see Figure 3 (a)). Since our router utilizes the omnidirectional antennas, it can suffer from wireless interference caused by other devices such as microwave ovens and cordless phones. To minimize such interference, we either turned off related devices during data collection process or having them removed from the data collection environment. The signals based on four facing directions of the user were recorded to observe the impact of user orientation.

The layout of the experimental environment is shown in Figure 3 (b). The data collection was performed in a typical office room of dimension 4m×6m with desktop, tables, chairs, and bookshelves. The locations of the transmitter (Tx) and the receiver (Rx) were placed near the two ends of the room. The distance between Rx and Tx was approximately 4.84m and they were placed about 82cm and 110cm above the floor, respectively. The 3 external antennas at the receiver end were placed at approximately 45cm apart. During data acquisition, each user was asked to sit at one of the positions labelled as Tx-side or Rx-side and perform his/her personal gesture of in-air handwritten signature while the transmitter and the receiver were put in operation. Some samples of the received signals containing the disturbances caused by the in-air signatures are shown in Figure 1.

2) Data Collection and Preprocessing

The above setup was used to collect the Wi-Fi CSI signals, which contained signals distorted by the in-air handwritten signatures, from 100 subjects. For each subject, 10 samples were collected for each of the 2 sitting positions with each position having 4 signature facing orientations. In order to enact a realistic usage environment during the data collection process, the users were seated at a comfortable posture at each of the four orientations for each of the two positions without any other restrictions of the body pose. The users can use either the left or the right hand, with or without specific finger posture, to perform the signature gesture according to their habits or preferences. Moreover, there is neither the need to use a specific language nor the need to use their real signature for the experimentation so long as the same signature pattern has been used for each orientation and position. Each of the collected samples is of approximately 12 second duration. A total of 8000 signal sequences were collected to form the database for experimentation.

The collected Wi-Fi CSI signals consist of several sub-carriers for each packet in the Tx-Rx antenna pair. The CSI of sub-carrier can be modeled as

$$h_c = |h_c|e^{\angle\theta}, \tag{1}$$

where $c \in \{1, 2, \cdots, C\}$ denotes the sub-carrier index, with C being the maximum number of sub-carriers, $|h_c|$ and θ respectively denote the amplitude and the phase of the sub-carrier. Since our Intel 5300 NIC had a firmware issue in extracting the phase information from sub-carriers at 2.4GHz frequency band, we only exploited the amplitude of the CSI of the collected signals. Under our 2×3 Tx-Rx configuration, there are 6 (2×3) streams and each stream has 7,000 to 15,000 packets on average, and each packet has 30 sub-carriers.

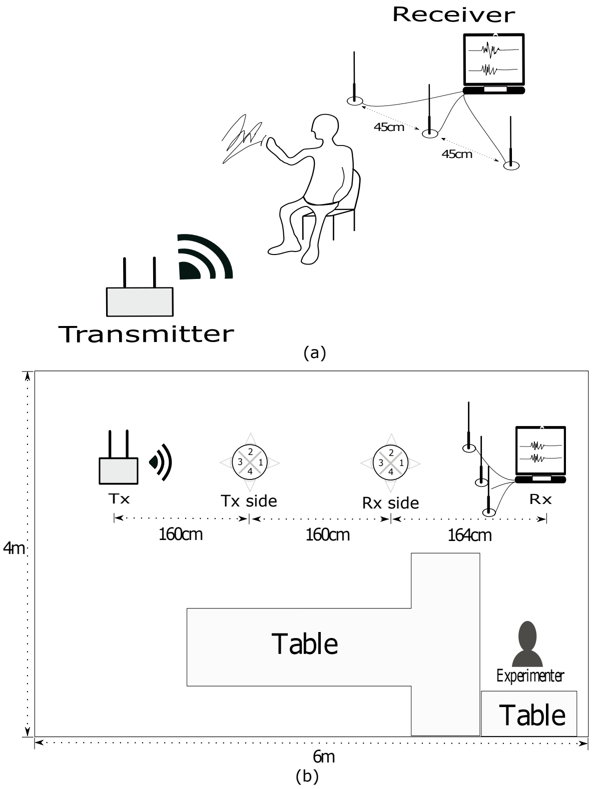


FIGURE 3. (a) An illustration of the data acquisition process, where the subject is to draw the in-air signature while sitting in between Tx and Rx. (b) The environment layout for data collection. The user is positioned at either the Tx-side or the Rx-side, each with four writing directions or orientations, during data collection process. The experimenter is located outside the data collecting zone.

[1] In [41], a finger-grained gestures based on the American sign language has been reported to work well for transceivers located within 1 meter. However, when the distance between the transceivers goes beyond 1 meter, the patterns caused by finger gestures in CSI stream were reported to nearly disappear.

Since the collected signals at fast receiving rate are computationally expensive and likely to contain missing values, we adopt several preprocessing steps to reduce the computational load and enhance the signal quality for our user identification system. Our preprocessing steps include linear-interpolation, resampling, low-pass filtering and removal of DC components. To reduce the computational complexity, the C sub-carrier streams of each packet in the Tx-Rx antenna pair were averaged into a single stream and this resulted in an 1-dimensinal signal. The averaging process can be written as:

$$\tilde{h}_c = \frac{1}{C}\sum_{c=1}^{C} |h_c|, \tag{2}$$

and the averaged signal stream can be expressed as:

$$\tilde{h}_s = [\tilde{h}_c^{(1)}, \tilde{h}_c^{(2)}, \cdots, \tilde{h}_c^{(P)}], \tag{3}$$

where $s \in \{1, 2, \cdots, S\}$ denotes the streaming index, with S denoting the maximum number of streams, and P denotes the maximum number of collected CSI packets. Subsequently, the signal streams are packed as:

$$\hat{h}_s = \begin{bmatrix} |h_1^{(1)}| & |h_1^{(2)}| & \ldots & |h_1^{(P)}| \\ |h_2^{(1)}| & |h_2^{(2)}| & \ldots & |h_2^{(P)}| \\ \vdots & \vdots & \ddots & \vdots \\ |h_C^{(1)}| & |h_C^{(2)}| & \ldots & |h_C^{(P)}| \end{bmatrix}. \tag{4}$$

The collected raw data tends to have missing packets and shows irregular data collection packet sizes P (i.e., 7000 to 15000 under our configuration) despite the fixed packet receiving rate. To address this issue, we firstly apply a linear-interpolation [44] to fill the missing values. After the linear interpolation process, the CSI packets are filtered by a Butterworth filter [45] to smoothen the highly jagged signal. The Butterworth filter has been chosen for this application since our signals of interest (signal distortions caused by signatures) mainly lie in the low frequency range but containing much high frequency components due to environmental noises. We have observed that the Butterworth filter has high efficiency in handling a large number of packets in the collected signals while showing good system performance. Subsequently, we perform resampling [46] to uniformly adjust all the packet sizes from P to P', where P' denotes the length of the resampled stream.

Since the Wi-Fi signals can be easily affected by surrounding terrain (e.g., location of stuffs on the table), the collected Wi-Fi signal contains DC component arising from minor changes of the data acquisition process. A shifting and subtraction algorithm described in [47] has been applied to remove the DC component. Figure 4 shows an illustration of the shifting and subtraction process.

Without and with consideration of the subcarrier variations, two types of CSI streams (called 1, 2-dimensional signals respectively) have been considered and packed into $\tilde{H}_{1d} \in \mathbb{R}^{S \times P'}$ and $\tilde{H}_{2d} \in \mathbb{R}^{S \times C \times P'}$ as:

$$\tilde{H}_{1d} = \begin{bmatrix} \tilde{h}_1 & \tilde{h}_2 & \cdots & \tilde{h}_S \end{bmatrix}^T, \tag{5}$$

$$\tilde{H}_{2d} = \begin{bmatrix} \hat{h}_1 & \hat{h}_2 & \cdots & \hat{h}_S \end{bmatrix}^T. \tag{6}$$

Considering the sampling instance, the data can be further packed into tensor form $\tilde{X}_{1d} \in \mathbb{R}^{N \times S \times P'}$ and $\tilde{X}_{2d} \in \mathbb{R}^{N \times S \times C \times P'}$:

$$\tilde{X}_{1d} = \begin{bmatrix} \tilde{H}_{1d}^{(1)} & \tilde{H}_{1d}^{(2)} & \cdots & \tilde{H}_{1d}^{(N)} \end{bmatrix}^T, \tag{7}$$

$$\tilde{X}_{2d} = \begin{bmatrix} \tilde{H}_{2d}^{(1)} & \tilde{H}_{2d}^{(2)} & \cdots & \tilde{H}_{2d}^{(N)} \end{bmatrix}^T, \tag{8}$$

where N denotes the maximum number of samples. Under our 2×3 Tx-Rx configuration, the S, C respectively become 6, 30. P' and shifting distance k are empirically set at 500 and 5, respectively. A sample set of results for each preprocessing step is shown in Figure 5.

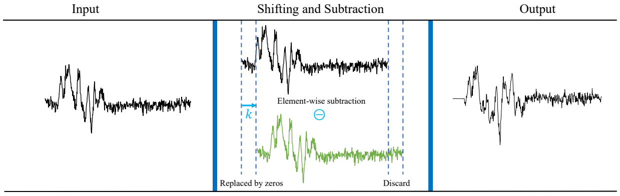


FIGURE 4. An illustration of the shifting and subtraction preprocessing. For visual simplicity, only a single averaged CSI stream is shown.

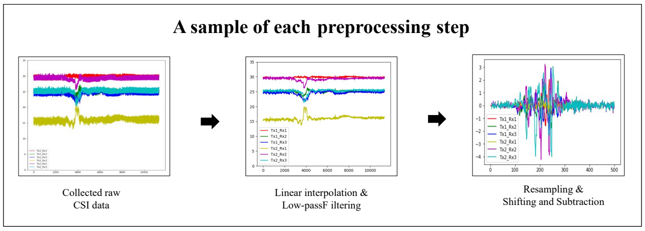


FIGURE 5. Sample result of each preprocessing step. For visual simplicity, only the 1-dimensional signal is shown.

B. NETWORK PRETRAINING AND TRANSFER LEARNING

Part (c) of Figure 2 illustrates our transfer learning based on the Kernel and the Range space projection (KAR learning). As explained in Section II-B, transfer learning is a methodology where the network weights trained on a certain source domain are taken and used in another target domain for learning refinement. It has been studied to address the problem of insufficient training data in deep learning [18]. Since our Wi-Fi in-air signature dataset is not large enough to train a complex network structure, the conventional transfer learning may not generalize well for prediction. Moreover, most of the works that adopted the Gradient Descent Algorithm to retrain the transferred model (called GD-retrain hereafter) are iterative and time-consuming. To avoid the iterative learning process in the retraining, we adopt the KAR learning [22]–[25] in this work. The KAR learning aims to solve the network in a single operating pass where no iterative search is needed. The main advantage of this approach is that neither descent nor gradient computation is needed for network learning. Moreover, It does not require learning of hyper-parameters such as the learning rate and the momentum setting.

1) Pretraining

Firstly, the model is pretrained utilizing every orientation at say, the Tx (or Rx) position based on the gradient descent algorithm [20], [21] where this position is not used for identity prediction. Based on the CNN structure described in [19], the feature extractor of the proposed pretraining model consists of three convolutional layers with activation functions followed by pooling layers. The activation function and the pooling layer are the commonly used ReLU and Max-pooling. Each convolution layer has several filters to be trained. Each filter convolves with the input and forward its result to the next layer. A Fully Connected Feedforward Neural Network (FCN) serves as a classifier for pretraining. The prevalent technique for CNN such as Batch Normalization (BN) [48] can be utilized for the pretraining. The pretrained model which shows the best result is saved and the parameters of the saved feature extractor are transferred for use in identity recognition at a different position. Let us denote the pretrained weights by $\hat{\mathbf{W}}_k$, $k = 2, ..., l$, and these pretrained weights will be used to perform weights update in the retraining stage that utilizes data from a different position.

2) Retraining

After the pretraining stage, we have the feature extractor pretrained with the Wi-Fi signature signals collected from every orientation at the Tx side. Since the Wi-Fi signature signals obtained from different positions are in the similar data domain [11], we can treat the pretrained convolutional layers as feature extractor and retrain only the classification layers for identity recognition. We shall describe the procedure of KAR learning for retraining our transferred model in the sequel.

Next, consider the new input Wi-Fi signature signals collected from say, the Rx (or Tx) side (for each single orientation) or even from a different collection environment. These novel signals are fed into the pretrained model to generate the transferred features. Suppose $\mathbf{X} \in \mathbb{R}^{n\times m}$ and $\mathbf{Y} \in \mathbb{R}^{n\times q}$ denote respectively the input features to the FCN (in other words, features generated from the transferred feature extractor) and the corresponding one-hot encoded matrix indicating the class label from the new position. The symbols n, m, q respectively denote the number of input samples, the number of output features, and the number of identities. By utilizing the pretrained weight matrices $\hat{\mathbf{W}}_2, \cdots, \hat{\mathbf{W}}_l$ obtained from the above stage, the network can be retrained according to [22], [24] utilizing the forward propagation of the network and the functional inverse based solution for $\mathbf{W}_1$ as follows:

$$\sigma(\cdots\sigma(\mathbf{X}\mathbf{W}_1)\hat{\mathbf{W}}_2 \cdots)\hat{\mathbf{W}}_l = \mathbf{Y}, \tag{9}$$

$$\sigma(\cdots\sigma(\mathbf{X}\mathbf{W}_1)\hat{\mathbf{W}}_2 \cdots) = \mathbf{Y}\hat{\mathbf{W}}_l^\dagger, \tag{10}$$

$$\sigma(\mathbf{X}\mathbf{W}_1) = (\cdots\sigma^{-1}(\mathbf{Y}\hat{\mathbf{W}}_l^\dagger)\cdots)\hat{\mathbf{W}}_2^\dagger, \tag{11}$$

$$\mathbf{W}_1 = \mathbf{X}^\dagger\sigma^{-1}(\cdots\sigma^{-1}(\mathbf{Y}\hat{\mathbf{W}}_l^\dagger)\cdots)\hat{\mathbf{W}}_2^\dagger), \tag{12}$$

where $\sigma(\cdot)$, $\dagger$ respectively denote an invertible activation function and the Moore-Penrose inverse operation [49].

After $\mathbf{W}_1$ is retrained following (12), it is back-substituted into (9) to retrain $\mathbf{W}_2$ based on:

$$\sigma(\mathbf{X}\mathbf{W}_1)\mathbf{W}_2 = (\cdots\sigma^{-1}(\mathbf{Y}\hat{\mathbf{W}}_l^\dagger)\cdots), \tag{13}$$

$$\mathbf{W}_2 = (\sigma(\mathbf{X}\mathbf{W}_1))^\dagger(\cdots\sigma^{-1}(\mathbf{Y}\hat{\mathbf{W}}_l^\dagger)\cdots). \tag{14}$$

This functional inverse based learning is repeated until $\mathbf{W}_l$ is retrained as shown in (15):

$$\mathbf{W}_l = (\sigma(\cdots\sigma(\mathbf{X}\mathbf{W}_1)\mathbf{W}_2\cdots)\mathbf{W}_{l-1})^\dagger\mathbf{Y}. \tag{15}$$

After $\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_l$ have been learned, the one-hot output prediction matrix can be estimated as follows:

$$\sigma(\cdots\sigma(\mathbf{X}\mathbf{W}_1)\mathbf{W}_2 \cdots)\mathbf{W}_l = \hat{\mathbf{Y}}, \tag{16}$$

where $\hat{\mathbf{Y}} \in \mathbb{R}^{n\times q}$. We shall call this learning process for our retraining as KAR-retrain.

C. DECISION FUSION

According to [50], multibiometrics can be utilized to overcome limitations inherent in each unibiometric system. According to [50], [51], multibiometric fusion can be performed at different levels namely, the data level, the feature level, the score level, the rank level and the decision level. Among these fusion levels, the score-level fusion is among the most commonly used due to the ease of accessing scores generated by commercial matchers [50], [51]. Moreover, it is known to produce the best classification accuracy performance [47], [50], [51]. For example, simple non-learning based algorithms such as the SUM-rule, the MAX-rule and the MIN-rule were performed and compared in [52]–[55]. Apart from the above fusion means, learning based algorithms, such as SVM [56] and TER [57], can be adopted for score level fusion [47], [58]–[60]. However, learning based fusion requires additional computational cost for training. For simplicity, we adopt the score-level fusion based on the rule-based operation. To take advantage of multibiometric system, we prepare two different kinds of signals in the preprocessing stage of the proposed system, which can be regarded as multi-samples of the same biometric modality captured with certain variations. As illustrated in part (d) of Figure 2, let $\hat{\mathbf{Y}}_{1d} \in \mathbb{R}^{n\times q}$ and $\hat{\mathbf{Y}}_{2d} \in \mathbb{R}^{n\times q}$ respectively denote the predicted score matrices of the 1-dimensional and the 2-dimensional Wi-Fi signals from the same user in our proposed system. After

TABLE 4. Overview of experiments

Experiments	Description	Model/Algorithm	Objective
Transfer Learning	Evaluation of several pretraining models	CNN	Selection of CNN architecture for each type of data
	Evaluation of retraining algorithm	For identification / verification: i) KAR retrain, ii) GD algorithm	To observe the performance of each method
Non-transfer learning	Evaluation of training from scratch using several learning models	For identification / verification: i) SVM, ii) ELM, iii) CNN	
Practical scenario	Evaluation of environmental factors such as interference and receiving packet rates	KAR-retrain	To observe the impacts of interference and receiving packet rates
Comparison	Comparison of all the methods in terms of: i) identification accuracy ii) Equal error rate iii) Elapsed training time	Transfer learning: i) KAR retrain, ii) GD algorithm: Non-transfer learening: i) SVM, ii) ELM, iii) CNN	To compare the system performances

a SoftMax normalization [61], the ranges of the scores for the 1-deimensional and the 2-dimensional Wi-Fi signals are both within [0, 1]. We fuse these two normalized scores by an element-wise rule-based operation, such as SUM, MAX, and MIN operations to get the final score matrix $\hat{\mathbf{Y}} \in \mathbb{R}^{n \times q}$:

$$\mathcal{L}(\hat{\mathbf{Y}}_{1d}, \hat{\mathbf{Y}}_{2d}) = \hat{\mathbf{Y}}, \tag{17}$$

where $\mathcal{L}$ denotes an element-wise rule-based operation, such as using SUM, MAX, and MIN. After score fusion, the classification accuracy can be calculated by comparing the fused one-hot label with the ground truth label (i.e., $\mathbf{Y}$).

IV. EXPERIMENTS

The main goal of this study is to verify the effectiveness of the proposed system for user identification and verification using the Wi-Fi based in-air handwritten signature signals.

Firstly, we introduce the details of our dataset and parameter settings in Section IV-A. Secondly, we provide the experimental results and discussion in Section IV-B. Essentially, we evaluate several pretraining models to utilize the best architecture for each of the two differently preprocessed data (i.e., the 1- and 2-dimensional data). Subsequently, we compare the result of KAR learning with that of the GD algorithm at the retraining stage. We also show results of non-transfer learning based methods (i.e., training from scratch at each orientation using SVM, ELM and CNN). Next, the impacts of ambient wireless interference and packet rates are evaluated. Finally, we compare all the methods in terms of the verification and identification accuracies as well as the elapsed training time. Table 4 summarizes our experiments with brief descriptions.

A. EXPERIMENTAL SETTINGS

1) Datasets

Table 5 summarizes the dataset for our experimentation. An expanded version of the Wi-Fi in-air signature dataset from [11] is utilized. Under the acquisition setup described in Section III-A, the dataset consists of 8000 samples collected from 100 subjects. The subjects were requested to draw their signature in the air while sitting at 2 different positions (i.e., the Tx side and the Rx side as shown in Figure 3 (b)). At each position, the subjects were asked to face 4 different directions (i.e., front, right, left and back). For each orientation, 10 samples were collected for each subject, resulting in 1000 samples per orientation (i.e., 10 samples × 8 orientations × 100 subjects in total).

In order to evaluate the impact of ambient wireless signal interference, another dataset has been collected with and without ambient interference. The ambient interference has been produced by two smartphones, which were carried by each subject and the experimenter. This dataset has been collected at a different location with the user sitting in between the Tx side and the Rx side in order to observe also the impact of geographical locations. Following the same acquisition protocol described above, this dataset consists of 200 samples from 5 subjects (i.e., 10 samples × 4 orientations × 5 subjects). According to [16], [62], [63], the performance of a Wi-Fi CSI-based recognition system highly depends on the granularity of the captured CSI signal. Therefore, in order to study also the impact of receiving packet rates on the recognition accuracy, the data has been collected using two different receiving packet rates (1kp/s and 10kp/s). As the set of genuine-users under the identification mode has been relatively small for representative learning, only the verification mode is studied here. Table 6 summarizes the four subsets of data collected for this study.

TABLE 5. The acquired dataset without consideration of ambient Wi-Fi interference

Number (#) of subjects	Position	Direction	Number (#) of samples
100	Tx side	Front	1000
		Right	1000
		Left	1000
		Back	1000
	Rx side	Front	1000
		Right	1000
		Left	1000
		Back	1000
Total		8000 samples	

TABLE 6. The acquired dataset for interference evaluation

Number (#) of subejcts		5 subjects			
Mobile wireless interference		w/ interference		w/o interference	
Packet rate		1kp/s	10kp/s	1kp/s	10kp/s
Direction	Front	50	50	50	50
	Right	50	50	50	50
	Left	50	50	50	50
	Back	50	50	50	50
Total		200	200	200	200

2) Evaluation Protocol

The experiments were performed using a desktop computer equipped with an i7 processor (3.70GHz), together with a NVIDIA Geforce GTX 1080 Ti GPU and 32GB of RAM. Both transfer-based learning and non-transfer-based learning algorithms were evaluated.

For evaluations of transfer-based learning methods, in the pretraining stage the dataset was divided into two groups, namely a training set and a test set. For example, the dataset (4000 samples) on four writing directions at the Tx side was separated into a training set (3200 samples, 80%) for training, and a test set (800 samples, 20%) for evaluation. A 5% portion from the training set (i.e., validation set) was used to fine tune the hyper-parameters. In the retraining stage, the performances were evaluated and averaged from five-fold cross-validation tests. For example, the dataset (1000 samples) of single direction at the Rx side was divided into a training set (800 samples, 80%), and a test set (200 samples, 20%). Subsequently, these two positions were swapped for pretraining (Rx side) and retraining (Tx side). For non-transfer-based learning methods, the five-fold cross-validation tests have been utilized for performance measure.

For the identification task, the rank-1 identification accuracy is adopted as the metric for performance evaluation. As mentioned in Section III-C, the system produces the final score matrix $\hat{Y} \in \mathbb{R}^{n \times q}$, where n, q respectively denote the number of input samples and the number of identities. The predicted identity label $\hat{y} \in \mathbb{R}^n$ can be obtained from $\hat{Y}$ by determining the rank-1 (the highest possibility of) identity for each sample. The identification accuracy is calculated by comparing between the predicted label $\hat{y}$ and the ground truth label y:

$$\text{Identification accuracy} = \frac{n(\hat{y} \cap y)}{\text{Total number of input samples } n}. \tag{18}$$

For the verification task, the degree of matching between two biometric templates is measured. In our implementation, the known target labels (1 for genuine-users and 0 for impostors) of the template pairs from the training set has been used to learn the networks (KAR-retrain, GD-retrain, CNN-scratch) [64]. The genuine-users refer to matching of templates drawn from the same user while the impostors refer to matching of templates drawn from different users. The verification accuracy is subsequently computed based on the population of test matches obtained from the genuine-users and the impostors in terms of the EER [32]. The EER is obtained based on the intersection of the False Acceptance Rate (FAR) and the False Rejection Rate (FRR) curves. The FAR is the number of false accept counts over the size of the impostor population. The FRR is the number of false reject counts over the size of the genuine-user population. Since the ratio of the genuine-user and the impostor populations is highly imbalanced, we randomly subsample the impostor pairs to make it having a balanced size with that of the genuine pairs. Also, limited by our memory constraint for KAR learning, only the 1-dimensional data is utilized for the verification task. All the experiments are implemented using the Pytorch [65] deep learning framework.

3) Parameter Settings for Transfer-based Learning.

Table 7 summarizes the parameter settings adopted in our experiments. For preprocessing, the packet size for resampling in (2) and (4) has been empirically fixed at 500 and the shifting distance k in Figure 4 has been empirically selected as 5. For the transfer learning (i.e., pretraining + retraining), there are several parameters to be set. At the pretraining stage of our proposed system, the CNN was trained starting with a learning rate (η) of 0.0001 utilizing the Cross Entropy Loss [66]. The Adam optimizer [67] with an L2 penalty of 0.0001 was adopted for the training. After half of the total epoch had elapsed, the η value was halved. The training epochs and batch size were empirically set at 120 and 64, respectively. The trained feature extractor which showed the best classification accuracy during training iterations was subsequently saved and transferred.

In order to compare with the adopted KAR retrain at the retraining stage, the GD algorithm was experimented starting with an η value of 0.0001 for the 1-dimensional data and 0.0005 for the 2-dimensional data. Similar to the pretraining stage, the η value was halved when the training iterations passed half of the total epoch. The training epochs were empirically set at 50. For the GD algorithm, all other settings including the loss function, the optimizer and the batch size were chosen to be the same as that of the pretraining stage. At the testing stage, a score level fusion with element-wise rule-based operations (i.e., SUM, MAX, and MIN) was adopted.

4) Parameter Settings for Non-transfer-based Learning.

For non-transfer-based learning, we evaluate three learning models, namely the SVM [56], the Extreme Learning Machine (ELM) [68], and a CNN training from scratch which utilized a similar structure to that used in the pretraining stage.

Before applying the SVM, the dimension of input signals was reduced to d dimension using the Principal Component Analysis (PCA) [69] in view of the heavy computational overhead. Several reduced features $d \in \{50, 100, \cdots, 300\}$ were selected and compared in Section IV-B2. Subsequently, common kernel functions such as linear, polynomial, and Radial Basis Function (RBF) kernels were also investigated for the SVM. The degree of the polynomials and the RBF kernel coefficients were empirically set as shown in Table 7.

In a similar manner, for ELM, the input features were reduced to d dimension by PCA. Several sizes of the hidden neurons $\alpha \in \{2000, 4000, 6000, 8000\}$ were compared in Section IV-B2. The sigmoid function was used for the activation function in ELM.

For training from scratch using CNN, the training epochs and learning rate were empirically set at 100 and 0.0001, respectively. All other settings (i.e., the loss function, the optimizer and the batch size) were chosen to be similar to that in the pretraining stage.

B. RESULTS AND DISCUSSION

1) Results of Transfer Learning Models

a: Evaluation of Pretraining Models.

Six CNN models have been experimented to figure out the best architecture for pretraining each of the two different types of experimental signals. Table 8 details the architectures of the CNN models utilized for pretraining study. As shown in the table, CNN_1 and CNN_3 have no BN layer

TABLE 7. Summary of parameter settings for pretraining, transfer learning, non-transfer learning, and testing.

Stage	Algorithm	Parameter	Meaning	Attribute
Preprocessing	Resampling	P'	Length of resampled CSI stream	500
	Shifting and subtraction	k	Shifting amount to right columns	5
Transfer learning (Pretraining)	Gradient descent (CNN)	epochs	Number (#) of training iterations	120
		batch size	Number (#) of training samples present in a single batch	64
		η	Learning rate	0.00001
		Optimizer	-	Adam
		Loss function	-	Cross Entropy Loss
Transfer learning (Retraining)	Gradient descent (FCN)	epochs	-	50
		η	Learning rate	0.0001 (1D), 0.0005 (2D)
Non-transfer learning	PCA	d	Number (#) of reduced features	50, 100, $\cdots$, 300 (Identification), 5 (Verification)
	SVM	Kernel function	-	linear, polynomial, RBF
		degree	The degree of the polynomial	2

		γ	Kernel coefficient of the polynomial and the RBF	0.01(poly, 1D), 0.001(RBF, 1D)
				0.1(poly, 2D), 0.00001(RBF, 2D)

	ELM	α	Number (#) of hidden neurons	2000, 4000, 6000, 8000
	Gradient descent (CNN)	epochs	-	100
		η	Learning rate	0.0001
Test	Score fusion	Rules for score fusion	-	SUM, MAX, MIN

[48] while CNN_2, CNN_4, and CNN_5 have. CNN_6 down-samples the feature-map with strided convolution instead of a pooling layer.

In order to increase the size of the dataset, a random horizontal flipping and a random vertical flipping with probability 0.5 have been used for the 2-dimensional (2D) Wi-Fi data while only a horizontal flipping is utilized for the 1-dimensional (1D) data. All the models are trained from scratch using the dataset from every orientation at position Tx side (or Rx side). The identification accuracies are presented in Table 9. For the 1D data, CNN_6 shows the best averaged identification accuracy while CNN_4 shows the best result for the 2D data. Following this observation, we use CNN_6, CNN_4 as a pretraining model for the 1D and the 2D data, respectively.

Under the verification mode, only the 1D data has been experimented for the convolutional network method due to our memory constraints. Therefore, only CNN_6 for 1D data is used for the pretraining. The EER performance of CNN_6 is 4.3% at the Tx side and 2.7% at the Rx side.

b: Evaluation of Retraining Algorithms.

Table 10 shows the average identification accuracies of the two retraining algorithms (KAR-retrain and GD-retrain) evaluated under the parameter settings described in Section IV-A. For the 1D data, the KAR-retrain yields higher average accuracy (98.05% at Tx side and 98.65% at Rx side) than that of the GD-algorithm (96.55% at Tx side and 98.175% at Rx side) at the retraining stage (see the “1D” columns of Table 10). For the 2D data, the GD-retrain (97.85% at Tx side and 98.25% at Rx side) outperforms the KAR-retrain (95.25% at Tx side and 96.45% at Rx side) as shown in the “2D” columns of Table 10.

The KAR-retrain shows comparable or slightly higher accuracy in case of utilizing score fusion (see “Score fusion” columns of Table 10). Among the three fusion means (i.e., SUM, MAX and MIN), both learning algorithms achieve the highest accuracy using the SUM-rule operation (named fusion-SUM hereafter). With fusion-SUM, the KAR-retrain shows marginally higher accuracy (99.75% at Tx side and 99.875% at Rx side) than that of the GD-retrain in the case of using fusion-SUM.

Table 13 shows the EER performance of KAR-retrain and GD-retrain under the verification mode. The GD-retrain shows a slightly lower EER than that of the KAR-retrain at both the Tx and the Rx sides (2.9% at Tx side and 2.2% at Rx side).

2) Results of Non-transfer Learning Models

a: Evaluation of Training from scratch using SVM.

Since applying the SVM directly on the raw dataset is computationally expensive, we utilize the PCA to reduce the number of features (called PCA-SVM hereafter). Figure 6 shows the average identification accuracies plotted over the reduced feature dimensions. For the 1D data, the SVM with RBF kernel (SVM-RBF) shows the best performance over all feature sizes except for the case of the reduced features of 50 dimensions where the SVM with linear kernel (SVM-linear) shows marginally better accuracy. For the 2D data, the SVM-linear shows the best performance in all cases. Among these studied cases, applying the SVM-linear to the reduced features of 50 dimensions shows the best performance on

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3071228, IEEE Access

TABLE 8. List of CNN models for pretraining study. The convolutional layer has four parameters (k, s, p, c) where the symbols c and k respectively indicate the number of outputs and the channel size. We have k^1 and k^2 for respectively the 1D and the 2D signals. s and p respectively denote the stride and the padding. The pooling layer has three parameters given by (k, s, p).

Layer	CNN_1	CNN_2	CNN_3	CNN_4	CNN_5	CNN_6
Feature extractor	Conv1, ReLU (3, 1, 1, 16)	Conv1, BN, ReLU (3, 1, 1, 16)	Conv1, ReLU (3, 1, 1, 16)	Conv1, BN, ReLU (3, 1, 1, 16)	Conv1, BN, ReLU (3, 1, 1, 32)	Conv1, ReLU (3, 1, 1, 32)
	MaxPool (2, 2, 0)	MaxPool (2, 2, 0)	MaxPool (2, 2, 0)	MaxPool (2, 2, 0)	MaxPool (2, 2, 0)	Conv2, ReLU (3, 2, 1, 64)
	Conv2, ReLU (3, 1, 1, 32)	Conv2, BN, ReLU (3, 1, 1, 32)	Conv2, ReLU (3, 1, 1, 32)	Conv2, BN, ReLU (3, 1, 1, 32)	Conv2, BN, ReLU (3, 1, 1, 64)	Conv3, ReLU (3, 1, 1, 128)
	MaxPool (2, 2, 0)	MaxPool (2, 2, 0)	MaxPool (2, 2, 0)	MaxPool (2, 2, 0)	MaxPool (2, 2, 0)	Conv4, BN, ReLU (3, 2, 1, 256)
	N/A	N/A	Conv3, ReLU (3, 1, 1, 64)	Conv3, BN, ReLU (3, 1, 1, 64)	Conv3, BN, ReLU (3, 1, 1, 128)	Conv5, BN, ReLU (3, 1, 1, 256)
			MaxPool (2, 2, 0)	MaxPool (2, 2, 0)	MaxPool (2, 2, 0)	Conv6, BN, ReLU (3, 2, 1, 256)
			N/A	N/A	Conv4, BN, ReLU (3, 1, 1, 256)	Conv7, BN, ReLU (3, 1, 1, 512)
					MaxPool (2, 2, 0)	N/A
Classifier	1024, Tanh					
	512, Tanh					
	100, Softmax					

TABLE 9. Test identification accuracy (%) of each pretrained CNN model at each position.

Data	Position	CNN_1	CNN_2	CNN_3	CNN_4	CNN_5	CNN_6
1D	Tx side	87	87	84	90	95	96
	Rx side	92	92	91	94	96	97
Average		89.5	89.5	87.5	92	95.5	**96.5**
2D	Tx side	92	93	94	94	91	92
	Rx side	92	94	94	95	93	93
Average		92.5	93.5	94	**94.5**	92	92.5

both types of data (i.e., 1D and 2D data) in terms of the average identification accuracy while the SVM with polynomial kernel (SVM-poly) shows the worst performance. Therefore, we utilize the former case for score fusion to compare with the proposed system. The "PCA-SVM" column of Table 11 shows the results of score fusion. Among the three fusion rules, fusion-SUM achieves the highest result (96.075% at Tx side and 96.825% at Rx side).

For the verification task, the PCA-SVM shows an average EER of 12.5% at the Tx side and 7.8% at the Rx side (see Table 13). The number of reduced features adopted was five and the kernel of SVM adopted was RBF.

b: Evaluation of Training from scratch using ELM.

Similar to the SVM case, we reduce the number of features using PCA before applying ELM on the dataset (named PCA-ELM hereafter). Figure 7 shows the average accuracies of PCA-ELM plotted over the reduced dimensions. For the 1D data, the ELM with 8,000 hidden neurons (ELM-8000) shows the best performance in all cases. Among these studied cases, applying the ELM-8000 to the reduced features of 50 dimensions shows the best result. For the 2D data, the ELM with 6,000 hidden neurons (ELM-6000) shows comparable results with that of ELM-8000. However, applying the ELM-6000 to the reduced features of 50 dimensions shows the best performance over the other cases. Therefore, similar to applying the SVM, we utilize the former case for score fusion to compare with the proposed system. The results of score fusion are presented in the "PCA-ELM" column of Table 11. Among the three fusion rules, fusion-SUM achieves the highest performance (96.275% at Tx side and 96.725% at Rx side).

For the verification task, the PCA-ELM shows an average EER of 11.7% at the Tx side and 8.8% at the Rx side. Similar to the above case (SVM), five reduced features with 8000 hidden neurons have been adopted.

c: Evaluation of Training from scratch using CNN.

We train the CNN from scratch using data collected at each direction as described in Section IV-A. The "CNN" column of Table 11 shows the results of training from scratch using CNN (called CNN-scratch hereafter). The selected CNN architecture is indicated next to each data type. As seen from the table, the average accuracy at the Tx side is 95.1% for the 1D data and 94.7% for the 2D data while the average accuracy at the Rx side is 97.35% for the 1D data and 97.05% for the 2D data. Similar to the SVM and ELM cases, we can see that using score fusion with rule-based operation improves the performance. Among the three fusion rules, the highest result can be found for the case of using fusion-SUM (99.45% at the Tx side and 99.85% at the Rx side).

The accuracy of verification for CNN_6 is reported in Table 13 in terms of the EER. The results show an average EER of 3.0% at the Tx side and 2.3% at the Rx side for CNN_6.

3) Results of KAR-retrain under interference and packet rate considerations

Table 12 shows the average accuracy results (in terms of the verification EER%) recorded based on 5-fold cross-validation tests. Under the scenarios with wireless interference, the data subset collected at the higher packet rate

TABLE 10. The identification accuracy (%) of each learning algorithm at the retraining stage evaluated from all four facing directions and two subject positions. All the performances are averaged results of 5-fold cross validation tests.

Pretraining Position	Retraining Position	Direction	KAR learning					Gradient descent				
			1D	2D	Score fusion			1D	2D	Score fusion		
					SUM	MAX	MIN			SUM	MAX	MIN
Rx side	Tx side	Front	97.5	92.3	99.5	97.9	97.2	95.4	96.9	99.9	99.5	98
		Right	97.9	97.4	99.9	99.4	99.4	96.9	98	100	99.8	99.5
		Left	98.7	95.9	100	99.6	99.8	98	98.5	98.9	98.7	98.7
		Back	98.1	95.4	99.6	99.5	98.7	95.9	98	99.9	99.2	99.4
Average			98.05	95.25	**99.75**	99.1	98.775	96.55	97.85	99.675	99.3	98.9
Tx side	Rx side	Front	98.8	94.7	99.8	99.6	99	98	97.7	99.6	99.6	98.8
		Right	98.8	97.9	99.9	99.4	99.6	98.8	98.6	99.8	99.1	99.2
		Left	98	96.6	99.9	99.4	99.3	98	98.3	99.5	99.3	99.5
		Back	99	96.6	99.9	99.6	99.6	97.9	98.4	100	99.9	100
Average			98.65	96.45	**99.875**	99.5	99.375	98.175	98.25	99.725	99.475	99.375

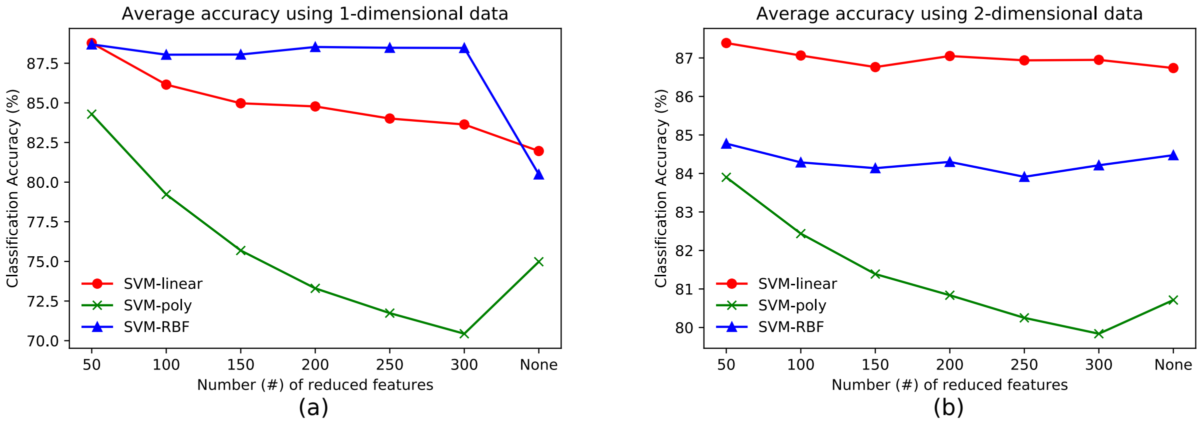


FIGURE 6. Results of Non-transfer Learning Model (SVM): identification accuracies (%) averaged from all the four facing directions and two subject positions plotted over number (#) of reduced features. "None" means no PCA is applied. (a) Average accuracy plot using 1-dimensional data (b) Average accuracy plot using 2-dimensional data

(10kps, 0.56%) shows a lower average EER than that at the lower packet rate (1kps, 3.94%). Similarly, with no wireless interference, a lower average EER is observed at the higher packet rate (10kps, 0.13%) than that at the lower packet rate (1kps, 2.56%). In terms of the total average, the performance without interference (1.34%) is seen to have a lower EER (better performance) than that of the case with interference (2.25%).

4) Comparison of all the Methods

a: Comparison of Identification Accuracy and Equal Error Rate.

Figure 8 plots the average identification accuracies (from all four facing directions at each position) of all the compared learning algorithms over each signature orientation. According to Figure 8(a), the PCA-SVM and the PCA-ELM show comparable average accuracies (96.075% and 96.275% at Tx side and 96.825% and 96.725% at Rx side). These accuracies are significantly lower than that of KAR-retrain, GD-retrain and CNN-scratch. Figure 8(b) shows an enlarged plots for KAR-retrain, GD-retrain and CNN-scratch. These results show either better or comparable accuracy of KAR-retrain relative to GD-retrain and CNN-scratch.

Table 13 shows the EER of all the compared methods. Similar to identification case, the PCA-SVM and the PCA-ELM show similar average EERs (12.5% and 11.7% at Tx side and 7.8% and 8.8% at Rx side) which is worse than that of KAR-retrain, GD-retrain and CNN_6. However, KAR-retrain shows comparable EER with that of the GD-retrain and CNN_6 (about 0.8% higher EER on average) in verification mode.

b: Comparison of Elapsed Training Time.

Table 14 shows the time taken to train the compared algorithms in seconds. All the compared algorithms have been trained using the GPU except for PCA-SVM which has been trained using the CPU due to the unavailability of codes for GPU.

Non-transfer learning methods: among the non-transfer learning methods in the table, the CNN-scratch shows the longest training time among the three evaluated methods. Attributed to the shadow architecture, both PCA-SVM and PCA-ELM show a much faster training time than that of the CNN-scratch. However, this speed comes with the price of a compromised identification accuracy.

Transfer learning methods: for the two evaluated methods based on transfer learning, the retraining time is compared

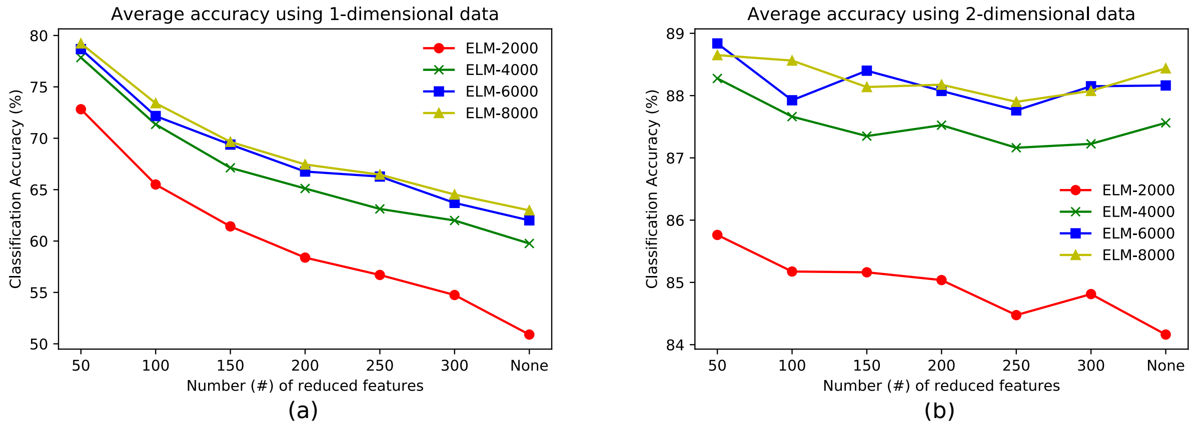


FIGURE 7. Results of Non-transfer Learning Model (ELM): identification accuracies (%) averaged from all the four facing directions and two subject positions plotted over number (#) of reduced features. "None" means no PCA is applied. (a) Average accuracy plot using 1-dimensional data (b) Average accuracy plot using 2-dimensional data

TABLE 11. The identification accuracy (%) of training from scratch using PCA-SVM with fusion-SUM and CNN evaluated from all four facing directions and two subject positions. All the performances are averaged from 5-fold cross validation tests.

Training Position	Direction	PCA-SVM			PCA-ELM			CNN				
		Score fusion			Score fusion			1D(CNN_6)	2D(CNN_4)	Score fusion		
		SUM	MAX	MIN	SUM	MAX	MIN			SUM	MAX	MIN
Tx side	Front	94.3	90.1	88.6	97	95.5	92.3	92.2	91.2	99	98	89.1
	Right	97.5	92.5	93.3	96.4	95.2	88	96.2	96	99.8	99.3	98.9
	Left	96.4	93.7	91	95.9	94.7	88	95.2	94.8	99.4	98.9	98.7
	Back	96.1	91.5	93	95.8	94.5	86.5	96.8	96.8	99.6	98.4	99
Average		96.075	91.95	91.475	96.275	94.975	88.7	95.1	94.7	99.45	98.65	96.425
Rx side	Front	97.8	94.9	93.2	97.1	95.1	87.8	98	97.6	99.8	99.6	98.9
	Right	96.9	91.8	94.9	96.9	94.9	88.1	97.4	97.2	99.9	99.8	99.2
	Left	96.4	92.4	91.9	97	95.4	86.1	97	96.6	99.9	99.6	99
	Back	96.2	92.2	92.3	95.9	94.4	85.6	97	96.8	99.8	99.6	99.2
Aveage		96.825	92.825	93.075	96.725	94.95	86.9	97.35	97.05	99.85	99.65	99.075

TABLE 12. Verification accuracy (EER, %) of KAR-retrain with and without ambient interference at a different location at two different packet rates. The results have been recorded based on 5-fold cross-validation tests.

Presence of wireless interference		w/ interference		w/o interference	
Packet rate		1kp/s	10kp/s	1kp/s	10kp/s
Orientation	Front	3.50	0.25	0	0
	Right	1.25	1.00	1.25	0
	Left	4.00	0.50	4.00	0.25
	Back	7.00	0.50	5.00	0.25
Average		3.94	0.56	2.56	0.13
Total Average		2.25		1.34	

between that of KAR-retrain and GD-retrain. The results in Table 14 show a much faster training speed of KAR-retrain than that of GD-retrain, both of which have been built upon a pretaining overhead (with pretraining time = 64.49 secs (1D), 162.41 secs (2D) utilizing data of all four directions).

Comparing between the transfer learning methods and the non-transfer learning methods, the former shows faster adaptation when there are more unseen user positions (apart from the studied Tx and Rx positions) for retraining. In other words, the transfer learning in our system trains only the last three layers in a single shot manner whenever a new user position is added while not compromising the identification accuracy. Such a fast adaptation to new user positions is a clear advantage over training from scratch for real world applications.

TABLE 13. The verification performance (EER, %) of each learning algorithm evaluated from all four facing directions and two subject positions. All the performances are averaged results of 5-fold cross validation tests.

Training Position	Direction	Transfer learning		Non-transfer learning		
		KAR-retrain	GD-retrain	PCA-SVM	PCA-ELM	CNN-scratch
Tx side	Front	5.1	4.5	15.0	16.4	3.8
	Right	6.9	2.6	12.5	10.2	2.9
	Left	6.8	2.8	12.7	11.4	2.4
	Back	8.8	2.2	9.9	8.8	2.5
Average		6.9	2.9	12.5	11.7	3.0
Rx side	Front	3.5	2.1	9.1	7.3	2.6
	Right	3.3	2.2	7.3	14	2.1
	Left	2.9	2.3	7.7	5.9	2.4
	Back	2.4	2.2	7.1	7.8	1.9
Average		3.025	2.2	7.8	8.8	2.3

5) Summary of results

The results are summarized and discussed as follows:

- As observed from Table 9, making the CNN architecture deeper does not help to improve the performance for the 2D data while it shows enhanced performance for the 1D data. We infer that this is because the 2D data has the same number of samples as that of the 1D data, but with

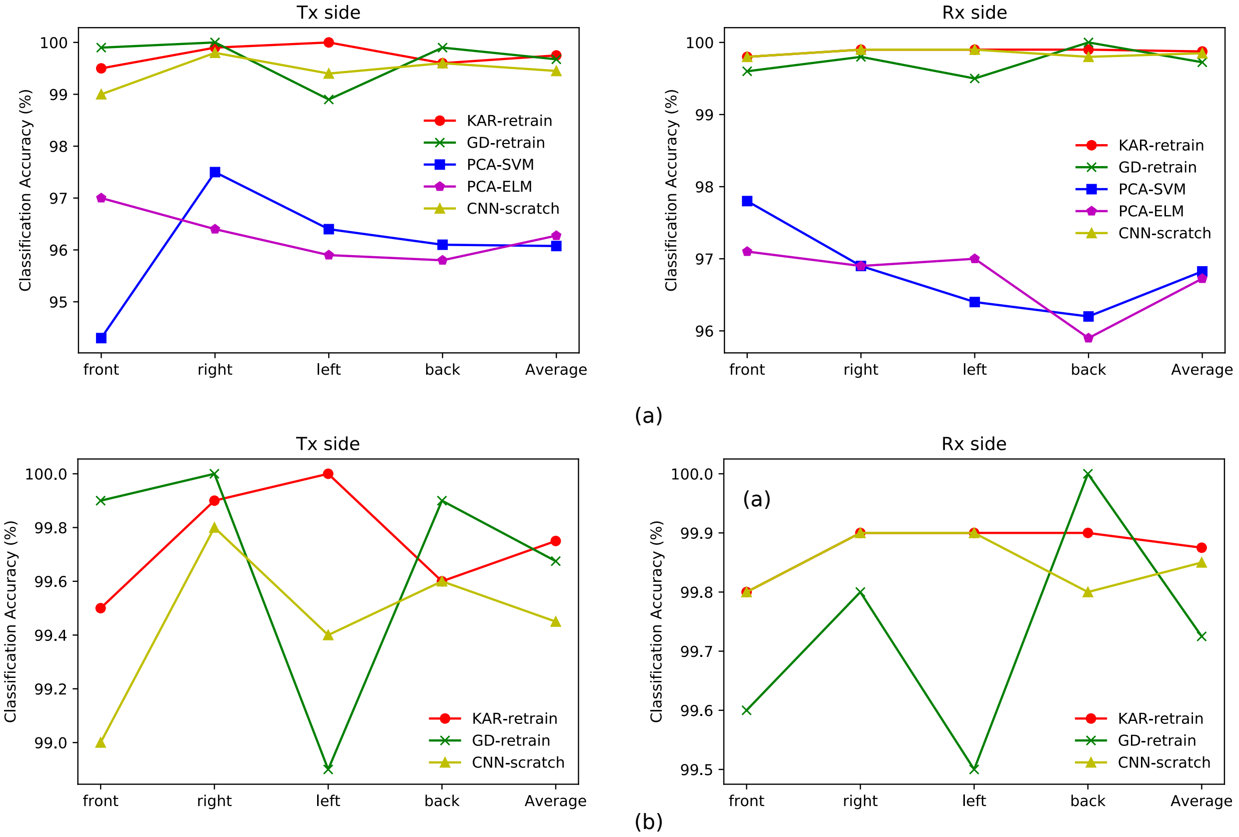


FIGURE 8. Plots of identification accuracies (%) of all the methods in case of using fusion-SUM. "Average" means the classification accuracy averaged from all four facing directions (i.e., front, right, left and back). (a) Plots of identification accuracies at two positions. (b) A zoom-in view of (a).

richer features. This may indicate that training CNN on 2D data needs much more number of samples in view of the higher dimensional features. Hence, training the deeper CNN using 2D data shows lower performance than that of the shallower CNN.

- From Figure 8, the proposed KAR-retrain outperforms the compared methods in terms of the average identification accuracy. Although the GD-retrain shows comparable accuracy compared with the KAR-retrain (see Table 10 for details), the KAR-retrain is shown to learn the transferred features with much lower computational cost (see the "Transfer learning" column of Table 14). This is because learning a classifier (i.e., FCN) using the GD algorithm requires iterative search while the KAR learning learns the weights of classifier in a single operating pass.
- Among the compared methods, the CNN-scratch also shows a competitive accuracy (see Figure 8). However, its computational cost is about 10 to 16 times heavier than that of the KAR-retrain when the pretraining overhead is not considered. The main reason is due to the iterative search of the GD algorithm to train the CNN-scratch.
- Although the PCA-SVM and the PCA-ELM classify the data with high computational efficiency (see Table 14), they achieve the lowest identification accuracy among the compared methods. Besides, the two learning algorithms incur an additional cost of using PCA for dimension reduction. Comparing with GD-retrain and CNN-scratch, the KAR-retrain provides a balance between identification accuracy and computational efficiency.
- Since the proposed system trains only the final three layers of the network for transfer learning, the adaptation of the network to new user positions and orientations is much faster than that of training from scratch.
- Under the verification mode, the EER performance for KAR-retrain is observed to be much better than that of PCA-SVM and PCA-ELM and comparable to that of GD-retrain and CNN_6 (see Table 13).
- The difference in experimental location does not have apparent impact on the verification performance (see Table 12).
- A higher receiving packet rate at 10kp/s shows a better verification accuracy than that at a lower receiving packet rate of 1kp/s. We have observed that even though severe packet loss has occurred during data acquisition, the higher packet rate at 10kp/s can capture about 1.5 to 2 times more packets than that at 1kp/s. Hence, setting the receiving packet rates as high as possible is advantageous for capturing accurate in-air signature.

TABLE 14. The elapsed training time (sec) for the case of using fusion-SUM for all the learning algorithms measured based on each of the four facing directions and the two positions. All the training times are averaged from 5-fold cross validation tests except pretraining time.

Training Position	Direction	Transfer learning		Non-transfer learning		
		KAR-retrain	GD-retrain	PCA-SVM*	PCA-ELM	CNN-scratch
Tx side	Front	1.404	9.731	2.33 + 0.141	2.33 + 0.153	28.149
	Right	1.386	6.942	2.32 + 0.134	2.32 + 0.157	18.168
	Left	1.384	10.995	2.26 + 0.132	2.26 + 0.164	18.249
	Back	1.385	8.901	2.27 + 0.127	2.27 +0.155	20.214
Total time		5.559	36.569	9.714	9.809	84.780
Rx side	Front	1.376	8.435	2.29 + 0.135	2.29 + 0.151	16.054
	Right	1.394	7.609	2.28 + 0.141	2.28 + 0.152	11.061
	Left	1.372	7.654	2.31 + 0.142	2.31 + 0.142	14.026
	Back	1.368	8.095	2.28 + 0.145	2.28 + 0.145	13.427
Total time		5.510	31.793	9.723	9.750	54.568

* Training on CPU

- Under the presence of wireless interference, the verification accuracy is observed to be slightly degraded comparing with that without ambient interference (see Table 12). This shows the feasibility of the proposed system for application under practical scenario.
- In this feasibility study, the improvements in terms of recognition accuracy and training time have been obtained based on the data sets captured under a similar setting. Moreover, as shown in Table 12, the proposed KAR-retrain performs favorably well on the dataset captured at a different location (0.13% at most in terms of EER). This is a clear sign that our system can effectively handle the varying patterns of captured signals at a different location by adopting the functional inverse based transfer learning.

6) Future works

Comparing with the studied methods, the proposed KAR-retrain shows promising performance with low retraining time. The study using different packet rates shows improved verification accuracy for the higher transmission rate. Moreover, the interference study shows minor degradation of verification accuracy with ambient interference. In order to improve the robustness of the system to work in the production environment, several aspects can be investigated in future. These investigations include recognition of multiple in-air signatures and handling of spoofing attacks. In other words, as the signature gesture can be visible during authentication, a challenging topic for future research would be whether any forgery of the personal signature gesture can be detected. Also, recognition of multiple users could be another challenging topic.

V. CONCLUSION

In this paper, a novel Wi-Fi enabled system for in-air handwritten signature was proposed for user identification after the feasibility has been established. The proposed system utilized the variation of CSI amplitude information caused by the in-air writing movements as patterns for recognition. To address the sensitiveness of captured patterns towards different user positions, the transfer learning was adopted to avoid training from scratch for each position. A KAR learning was adopted in the retraining stage to reduce the learning computational cost. Subsequently, two differently preprocessed features were fused at the score level for performance enhancement. Our experimental results based on a moderate size of in-air signature dataset showed promising identification accuracy of the proposed system with relatively low computational cost during transfer.

REFERENCES

[1] A. K. Jain, A. A. Ross, and K. Nandakumar, Introduction to biometrics. Springer Science & Business Media, 2011.

[2] A. K. Jain, P. Sharath, and B. Ruud, Biometrics personal identification in networked society. Springer US, 2006.

[3] A. K. Jain, K. Nandakumar, and A. Ross, "50 years of biometric research: Accomplishments, challenges, and opportunities," Pattern recognition letters, vol. 79, pp. 80–105, 2016.

[4] W. H. Khoh, Y. H. Pang, and A. B. J. Teoh, "In-air hand gesture signature recognition system based on 3-dimensional imagery," Multimedia Tools and Applications, vol. 78, no. 6, pp. 6913–6937, 2019.

[5] H. Manabe, Y. Yamakawa, T. Sasamoto, and R. Sasaki, "Security evaluation of biometrics authentications for cellular phones," in 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP). IEEE, 2009, pp. 34–39.

[6] R. V. Yampolskiy and V. Govindaraju, "Behavioural biometrics: a survey and classification," International Journal of Biometrics, vol. 1, no. 1, pp. 81–113, 2008.

[7] M. Diaz, M. A. Ferrer, D. Impedovo, M. I. Malik, G. Pirlo, and R. Plamondon, "A perspective analysis of handwritten signature technology," ACM Computing Surveys (CSUR), vol. 51, no. 6, pp. 1–39, 2019.

[8] S. W. Shah and S. S. Kanhere, "Recent trends in user authentication–a survey," IEEE Access, vol. 7, pp. 112 505–112 519, 2019.

[9] J.-H. Jeon, B.-S. Oh, and K.-A. Toh, "A system for hand gesture based signature recognition," in 2012 IEEE international Conference on Control Automation Robotics & Vision (ICARCV). IEEE, 2012, pp. 171–175.

[10] G. Bailador, C. Sanchez-Avila, J. Guerra-Casanova, and A. de Santos Sierra, "Analysis of pattern recognition techniques for in-air signature biometrics," Pattern Recognition, vol. 44, no. 10-11, pp. 2468–2478, 2011.

[11] H.-C. Moon, S.-I. Jang, K. Oh, and K.-A. Toh, "An in-air signature verification system using Wi-Fi signals," in Proceedings of the 2017 4th International Conference on Biomedical and Bioinformatics Engineering. ACM, 2017, pp. 133–138.

[12] S. K. Behera, A. K. Dash, D. P. Dogra, and P. P. Roy, "Air signature recognition using deep convolutional neural network-based sequential model," in 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018, pp. 3525–3530.

[13] J. Malik, A. Elhayek, S. Ahmed, F. Shafait, M. I. Malik, and D. Stricker, "3DAirSig: A framework for enabling in-air signatures using a multi-modal depth sensor," Sensors, vol. 18, no. 11, p. 3872, 2018.

[14] H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li, "RT-Fall: A real-time and contactless fall detection system with commodity WiFi devices," IEEE Transactions on Mobile Computing, vol. 16, no. 2, pp. 511–526, 2016.

[15] Y. Wang, K. Wu, and L. M. Ni, "WiFall: Device-free fall detection by wireless networks," IEEE Transactions on Mobile Computing, vol. 16, no. 2, pp. 581–594, 2016.

[16] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Recognizing keystrokes using WiFi devices," IEEE Journal on Selected Areas in Communications, vol. 35, no. 5, pp. 1175–1190, 2017.

[17] X. Liu, J. Cao, S. Tang, J. Wen, and P. Guo, "Contactless respiration monitoring via off-the-shelf WiFi devices," IEEE Transactions on Mobile Computing, vol. 15, no. 10, pp. 2466–2479, 2015.

[18] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," arXiv preprint arXiv:1808.01974, 2018.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[20] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern classification. John Wiley & Sons, 2012.

[21] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint arXiv:1609.04747, 2016.

[22] K.-A. Toh, "Learning from the kernel and the range space," in 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS). IEEE, 2018, pp. 1–6.

[23] K.-A. Toh, Z. Lin, Z. Li, B. Oh, and L. Sun, "Gradient-free learning based on the kernel and the range space," arXiv preprint arXiv:1810.11581, 2018.

[24] K.-A. Toh, "Analytic network learning," arXiv preprint arXiv:1811.08227, 2018.

[25] K. A. Toh, "Kernel and range approach to analytic network learning," International Journal of Networked and Distributed Computing, vol. 7, no. 1, pp. 20–28, 2018.

[26] M. Katagiri and T. Sugimura, "Personal authentication by free space signing with video capture," in The 5th Asian Conference on Computer Vision (ACCV), vol. 6. Citeseer, 2002.

[27] H. Sajid and S. C. Sen-ching, "VSig: Hand-gestured signature recognition and authentication with wearable camera," in 2015 IEEE International Workshop on Information Forensics and Security (WIFS). IEEE, 2015, pp. 1–6.

[28] J. Jung, J. Kim, and K.-A. Toh, "Transfer learning of Wi-Fi handwritten signature signals for identity verification based on the kernel and the range space projection," in 2019 IEEE International Conference on Image Processing (ICIP). IEEE, 2019, pp. 2586–2590.

[29] J. G. Casanova, C. S. Ávila, A. de Santos Sierra, G. B. del Pozo, and V. J. Vera, "A real-time in-air signature biometric technique using a mobile device embedding an accelerometer," in International Conference on Networked Digital Technologies. Springer, 2010, pp. 497–503.

[30] N. N. Diep, C. Pham, and T. M. Phuong, "SigVer3D: Accelerometer based verification of 3-d signatures on mobile devices," in Knowledge and Systems Engineering. Springer, 2015, pp. 353–365.

[31] Y. Fang, W. Kang, Q. Wu, and L. Tang, "A novel video-based system for in-air signature verification," Computers & Electrical Engineering, vol. 57, pp. 1–14, 2017.

[32] A. Jain, L. Hong, and S. Pankanti, "Biometric identification," Communications of the ACM, vol. 43, no. 2, pp. 90–98, 2000.

[33] K. Ohara, T. Maekawa, S. Sigg, and M. Youssef, "Preliminary investigation of position independent gesture recognition using Wi-Fi CSI," in 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). IEEE, 2018, pp. 480–483.

[34] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," Journal of Big data, vol. 3, no. 1, p. 9, 2016.

[35] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2014, pp. 1717–1724.

[36] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," IEEE transactions on medical imaging, vol. 35, no. 5, pp. 1285–1298, 2016.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in 2009 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, 2009, pp. 248–255.

[38] L. Wang, W. Ouyang, X. Wang, and H. Lu, "STCT: Sequentially training convolutional networks for visual tracking," in Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2016, pp. 1373–1381.

[39] J. Wang, P. Guo, and X. Xin, "Review of pseudoinverse learning algorithm for multilayer neural networks and applications," in International Symposium on Neural Networks (ISNN). Springer, 2018, pp. 99–106.

[40] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," ACM SIGCOMM Computer Communication Review, vol. 41, no. 1, pp. 53–53, 2011.

[41] H. Li, W. Yang, J. Wang, Y. Xu, and L. Huang, "WiFinger: Talk to your smart devices with finger-grained gesture," in Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 2016, pp. 250–261.

[42] H. Yan, Y. Zhang, Y. Wang, and K. Xu, "WiAct: A passive WiFi-based human activity recognition system," IEEE Sensors Journal, vol. 20, no. 1, pp. 296–305, 2019.

[43] C. Xiao, D. Han, Y. Ma, and Z. Qin, "CsiGAN: Robust channel state information-based activity recognition with GANs," IEEE Internet of Things Journal, vol. 6, no. 6, pp. 10 191–10 204, 2019.

[44] E. Meijering, "A chronology of interpolation: from ancient astronomy to modern signal and image processing," Proceedings of the IEEE, vol. 90, no. 3, pp. 319–342, 2002.

[45] S. Butterworth, "On the theory of filter amplifiers," Wireless Engineer, vol. 7, no. 6, pp. 536–541, 1930.

[46] J. Eric, O. Travis, P. Pearu et al., "SciPy: Open source scientific tools for Python," https://docs.scipy.org/doc/scipy-0.16.0/reference/generated/scipy.signal.resample.html, 2019, [accessed 19 Dec. 2019].

[47] J. Kim, K. Oh, B.-S. Oh, Z. Lin, and K.-A. Toh, "A line feature extraction method for finger-knuckle-print verification," Cognitive Computation, vol. 11, no. 1, pp. 50–70, 2019.

[48] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.

[49] A. Albert, Regression and the Moore-Penrose pseudoinverse. Elsevier, 1972.

[50] A. Ross and A. Jain, "Information fusion in biometrics," Pattern recognition letters, vol. 24, no. 13, pp. 2115–2125, 2003.

[51] A. K. Jain, P. Flynn, and A. A. Ross, Handbook of biometrics. Springer Science & Business Media, 2007.

[52] A. K. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," Pattern recognition, vol. 38, no. 12, pp. 2270–2285, 2005.

[53] N. Poh and S. Bengio, "Database, protocols and tools for evaluating score-level fusion algorithms in biometric authentication," Pattern Recognition, vol. 39, no. 2, pp. 223–233, 2006.

[54] M. He, S.-J. Horng, P. Fan, R.-S. Run, R.-J. Chen, J.-L. Lai, M. K. Khan, and K. O. Sentosa, "Performance evaluation of score level fusion in multimodal biometric systems," Pattern Recognition, vol. 43, no. 5, pp. 1789–1800, 2010.

[55] M. B. Yılmaz and B. Yanıkoğlu, "Score level fusion of classifiers in off-line signature verification," Information Fusion, vol. 32, pp. 109–119, 2016.

[56] C. Cortes and V. Vapnik, "Support-vector networks," Machine learning, vol. 20, no. 3, pp. 273–297, 1995.

[57] K.-A. Toh and H.-L. Eng, "Between classification-error approximation and weighted least-squares learning," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 4, pp. 658–669, 2008.

[58] B.-S. Oh, K.-A. Toh, K. Choi, A. B. J. Teoh, and J. Kim, "Extraction and fusion of partial face features for cancelable identity verification," Pattern Recognition, vol. 45, no. 9, pp. 3288–3303, 2012.

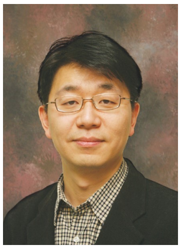
[59] B.-S. Oh, K.-A. Toh, and A. B. J. Teoh, "Direct extraction and fusion of image pixel features for face identity verification," in 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS). IEEE, 2010, pp. 1–6.

[60] B.-S. Oh and K.-A. Toh, "Fusion of structured projections for cancelable face identity verification," in 2011 International Joint Conference on Biometrics (IJCB). IEEE, 2011, pp. 1–8.
[61] C. M. Bishop, Pattern recognition and machine learning. springer, 2006.
[62] R. Nandakumar, B. Kellogg, and S. Gollakota, "Wi-Fi gesture recognition on existing devices," arXiv preprint arXiv:1411.5394, 2014.
[63] Y. Lu, F. Wu, S. Tang, L. Kong, and G. Chen, "Pushing the limit of CSI-based activity recognition: An enhanced approach via packet reconstruction," in 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). IEEE, 2019, pp. 1–9.
[64] N. Liu, M. Zhang, H. Li, Z. Sun, and T. Tan, "DeepIris: Learning pairwise filter bank for heterogeneous iris verification," Pattern Recognition Letters, vol. 82, pp. 154–161, 2016.
[65] "Pytorch," http://pytorch.org/, Facebook, 2019, [accessed 19 Dec. 2019].
[66] S. Mannor, D. Peleg, and R. Rubinstein, "The cross entropy method for classification," in Proceedings of the 22nd International Conference on Machine learning (ICML). ACM, 2005, pp. 561–568.
[67] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[68] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," Neurocomputing, vol. 70, no. 1-3, pp. 489–501, 2006.
[69] M. Turk and A. Pentland, "Eigenfaces for recognition," Journal of cognitive neuroscience, vol. 3, no. 1, pp. 71–86, 1991.

JUNSIK JUNG received the B.S. degree in electrical computer engineering from University of Seoul, South Korea, in 2018 and the M.S. degree in electrical and electronic engineering from Yonsei University, South Korea, in 2020. He is currently pursuing the Ph.D. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. His research interests include biometrics, machine learning, and video understanding

HANCHEOL MOON received the M.S. degree from Yonsei University, South Korea. He is currently pursuing the Ph.D. degree in computer science from Nanyang Technological University (NTU), Singapore. His current research interests include natural language processing and machine learning.

JOOYONG KIM received the B.S. degree in electrical and electronic engineering from Yonsei University, South Korea, in 2015. He is currently pursuing the Ph.D. degree in electrical and electronic engineering from Yonsei University, South Korea. His research interests include biometrics, pattern recognition, machine learning and artificial intelligence.



DONGHYUN KIM received B.S. with summa cum laude and M.S. from Seoul National University in 1993 and 1995 in electronics engineering. He graduated from the Massachusetts Institute of Technology in 2001 with Ph.D. in electrical engineering in the area of novel multi-dimensional display technologies and smart optical filters. He worked on next generation fiber-optic access communication systems at Corning Inc. as a senior research scientist and then investigated cellular biophotonic sensors for cell-based assays at Cornell University as a postdoctoral fellow. Since 2004, He has been leading Biophotonics Engineering Laboratory of Yonsei University, Seoul, Korea. The main theme of his research at Yonsei has been focused on sensing and imaging applications in biomedical engineering based on plasmonic techniques. He serves as Undergraduate Chair of the School of Electrical Engineering of Yonsei University and Vice President in the Board of Directors of the Optical Society of Korea. He is Fellow of the SPIE.

KAR-ANN TOH (Senior Member, IEEE) received the Ph.D. degree from Nanyang Technological University (NTU), Singapore. He then worked for two years in the aerospace industry, prior to his postdoctoral appointments at research centers in NTU, from 1998 to 2002. He was with the Institute for Infocomm Research, Singapore, from 2002 to 2005. He then joined the School of Electrical and Electronic Engineering, Yonsei University, South Korea where he was elevated to full Professor in 2009. During his sabbatical year in 2020, he worked with the Department of Electrical and Computer Engineering, National University of Singapore (ECE-NUS) as a Visting Professor. He is a co-inventor of two U.S. patents and has made several PCT filings related to biometric applications. His research interests include biometrics, pattern classification, machine learning, optimization, and neural networks. Besides being active in research and publications, he has served as a member of advisory board and technical program committee for international conferences related to biometrics and artificial intelligence. He has served as an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY from 2013 to 2016 and the Journal of The Franklin Institute from 2018 to 2020. He is currently an Associate Editor/Editorial Board of IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR, AND IDENTITY SCIENCE, Pattern Recognition Letters, and IET Biometrics.