

# Deep triplet network adopting kernel and the range space learning for Wi-Fi signature verification

Young-Woong Kwon, Jooyoung Kim, and Kar-Ann Toh

School of Electrical and Electronic Engineering, Yonsei University, 50 Yonsei-ro,  
Seodaemun-gu, Seoul 03722, Republic of Korea  
{herokwon, harrykim, katoth}@yonsei.ac.kr

**Abstract.** In this paper, we proposed a novel identity verification system based on Wi-Fi signature signals. In order to train the triplet network effectively, triplet input is mined from Wi-Fi signature signals by the kernel and the range space learning. Subsequently, ConvNet structure is adopted for feature extractor which generates feature vectors from input triplets. Finally, the triplet network is trained by matching between extracted feature vectors based on L-2 distance comparison. Our networks show good feature extracting performance, especially in low-dimension outputs and shows encouraging experimental results in terms of verification accuracy.

**Keywords:** Wi-Fi CSI signal · in-air signature verification · ConvNet · triplet loss · kernel and the range space learning.

## 1 Introduction

Along with the development of computer technology, researchers have investigated into using various handwritten signatures for identity verification in the past few decades. Among those handwritten signatures verification methods [11], it has been shown that In-air signature/gesture [6], [9], [4] has the potential to be used as a identifier. However, they adopted a depth camera or mobile sensor came with an additional cost which limited the use in commodity applications. Recently, In-air signature verification using Wi-Fi CSI signal has developed [10]. signature verification system using commodity Wi-Fi devices has advantages: 1) Cheap: Use commercial device, 2) Easy: User-friendly, 3) Secure: Hard to forgery. [1], [8] uses Wi-Fi CSI signal for in-air gesture classification, but their work is limited in a preset hand gesture. [15] used deep learning methods which have good ability to extract features. In this work, we focus particularly on the in-air signature verification using commodity Wi-Fi devices. We propose a novel yet simple feature extraction method for Wi-Fi CSI signature identity verification. Based on ConvNet with Triplet loss and KAR learning, we extract distinctive features from Wi-Fi CSI signatures. The proposed method achieved promising verification performance with the advantages of no additional devices.

The main contributions of this paper can be summarized as follows:

- Proposal of a system for identity verification based on the multiple-direction Wi-Fi signature signals.
- Effective ConvNet learning thanks to mining triplet input by KAR space projection learning.
- Experimental studies on in-house Wi-Fi signature signal dataset collected from 50 subjects.

The paper is organized as follows: our proposed method will be discussed in Section 2. Section 3 describes our experimental results and analysis. Some concluding remarks are given in Section 4.

## 2 Related Works

### 2.1 Triplet Network

A Triplet network is comprised of 3 instances of the same feedforward network (with shared parameters)[5]. It is widely used in face recognition [12] and person re-identification, which solves the matching problem of individuals and identity between camera images[3, 2]. The distinction between classes may be ambiguous in the above tasks since the targets may be looking in different directions or taking different positions. to achieve good ability to distinguish between classes, a triplet network works by extract feature vectors which expand margins between same and different class.

The loss function of triplet network is calculated as the difference in L-2 distance between anchor to each positive and negative triplet outputs. Optimizing the training process is important because gradient vanishing can occur during training. It happens when the negative class distance is much greater than positive class distance, the output of the loss function can down to zero as a result.

There are many ways to optimize training. In [12], selects inputs from large mini-batch at each training iteration using the network during training. [3] adds second margin for triplet loss to expand intra-class distance. Following in [2], quadruplet loss function makes the distance between the same classes is less than the distance between any different classes.

In this paper, we adopt multilayer feedforward networks trained by the kernel and the range space learning to select good triplet input for training the triplet network. It has advantage for reducing computing resources as well as boosting the training optimization.

### 2.2 kernel and the range space learning

Multilayer feedforward neural networks is widely known to be good for feature extractor and classifier. This network is generally trained by the gradient descent method and backpropagation. However, setting the learning parameters such as learning rate or momentum value is important to use the gradient descent method.

Recently, gradient-free learning framework based on series of kernel and range (KAR) space manipulation has developed [13, 14]. since it is trained by algebraic methods, no learning parameters are needed to train the networks. using this novel framework, we can train multilayer feedforward neural networks with any numbers and size of layers.

Let the training dataset  $\mathbf{X} \in \mathbb{R}^{m \times (n+1)}$  and  $\mathbf{G} \in \mathbb{R}^{m \times n}$  is network outputs. Multilayer neural network structure is shown below:

$$\mathbf{G} = \sigma \left( [\mathbf{1}, \sigma (\dots [\mathbf{1}, \sigma ([\mathbf{1}, \sigma (\mathbf{X}\mathbf{W}_1)] \mathbf{W}_2)] \dots \mathbf{W}_{(i-1)})] \mathbf{W}_i \right), \quad (1)$$

where  $\mathbf{W}_1 \in \mathbb{R}^{(n+1) \times h_1}$ ,  $\mathbf{W}_2 \in \mathbb{R}^{(h_1+1) \times h_2}$ ,  $\dots$ ,  $\mathbf{W}_i \in \mathbb{R}^{(h_{(i-1)}+1) \times n}$ ,  $\mathbf{1} = [1, \dots, 1]^T \in \mathbb{R}^{m \times 1}$  and  $\sigma(\cdot)$  is activation function.

When all weights have been learned, for given anchor signal, hard negative samples can be mined among the distance of network output of desired signal  $\mathbf{G}$  is below the threshold.

Network learning is archived using one-hot encoded target  $\mathbf{Y} \in \mathbb{R}^{m \times n}$  instead of network output  $\mathbf{G}$ . After that, we have the weight matrix  $\mathbf{W}_1 \dots \mathbf{W}_i$  to train. the weight matrix can be separated into weights and bias term as  $\mathbf{W}_2 \dots \mathbf{W}_i = \begin{bmatrix} \mathbf{w}_2^T \\ W_2 \end{bmatrix} \dots \begin{bmatrix} \mathbf{w}_i^T \\ W_i \end{bmatrix}$ .

after assign random weights to  $\mathbf{W}_1 \dots \mathbf{W}_i$ , we get  $\mathbf{W}_1$ . it is solved as follows:

$$\begin{aligned} & [\sigma^{-1}(\mathbf{Y}) - \mathbf{1} \cdot \mathbf{w}_i^T] W_i^\dagger = \sigma (\dots [\mathbf{1}, \sigma ([\mathbf{1}, \sigma (\mathbf{X}\mathbf{W}_1)] \mathbf{W}_2)] \dots \mathbf{W}_{(i-1)}) \\ \Rightarrow & \left[ \sigma^{-1} \left( \dots \left[ \sigma^{-1} \left( [\sigma^{-1}(\mathbf{Y}) - \mathbf{1} \cdot \mathbf{w}_i^T] W_i^\dagger \right) - \mathbf{1} \cdot \mathbf{w}_{(i-1)}^T \right] W_{(i-1)}^\dagger \dots \right) \right. \\ & \quad \left. - \mathbf{1} \cdot \mathbf{w}_2^T \right] W_2^\dagger = \sigma (\mathbf{X}\mathbf{W}_1) \\ \Rightarrow & \mathbf{X}^\dagger \sigma^{-1} \left( \left[ \sigma^{-1} \left( \dots \left[ \sigma^{-1} \left( [\sigma^{-1}(\mathbf{Y}) - \mathbf{1} \cdot \mathbf{w}_i^T] W_i^\dagger \right) - \mathbf{1} \cdot \mathbf{w}_{(i-1)}^T \right] W_{(i-1)}^\dagger \dots \right) \right. \right. \\ & \quad \left. \left. - \mathbf{1} \cdot \mathbf{w}_2^T \right] W_2^\dagger \right) = \mathbf{W}_1. \end{aligned} \quad (2)$$

After deriving the  $\mathbf{W}_1$ , the  $\mathbf{W}_2$  also can be optimized as:

$$\begin{aligned} \Rightarrow & (\sigma (\mathbf{X}\mathbf{W}_1))^\dagger \left( \dots \left[ \sigma^{-1} \left( [\sigma^{-1}(\mathbf{Y}) - \mathbf{1} \cdot \mathbf{w}_i^T] W_i^\dagger \right) - \mathbf{1} \cdot \mathbf{w}_{(i-1)}^T \right] W_{(i-1)}^\dagger \dots \right) \\ & = \mathbf{W}_2. \end{aligned} \quad (3)$$

By repeating this process recursively until all weight matrix values are obtained, the  $\mathbf{W}_i$  can be obtained as follows:

$$\mathbf{W}_i = [\mathbf{1}, \sigma (\dots [\mathbf{1}, \sigma ([\mathbf{1}, \sigma (\mathbf{X}\mathbf{W}_1)] \mathbf{W}_2)] \dots \mathbf{W}_{(i-1)})]^\dagger \sigma^{-1}(\mathbf{Y}). \quad (4)$$

### 3 Proposed System

In this section, we propose an identify verification system based on the Wi-Fi based in-air handwritten signature (will be called Wi-Fi signature signals hereafter). An overview of the proposed system utilizing the ConvNet [7] and the kernel and the range space projection learning (KAR space learning) [13, 14] is shown in Fig.1.

Essentially, the Wi-Fi signature signals are randomly divided to create the training and the validation set. Subsequently, the KAR space projection learning generates the triplet data by mining the hard positive and the hard negative samples for the given anchor signal from the training dataset.

To train the ConvNet structure, generated triplet data flows into the predefined ConvNet structure. The ConvNet extracts feature vectors from given triplet input. ConvNet training is achieved by the triplet loss, which is formulated by the L-2 distance metrics of feature vectors. The overall ConvNet network is trained by the backpropagation scheme. The following subsections detail the each step of the proposed system.

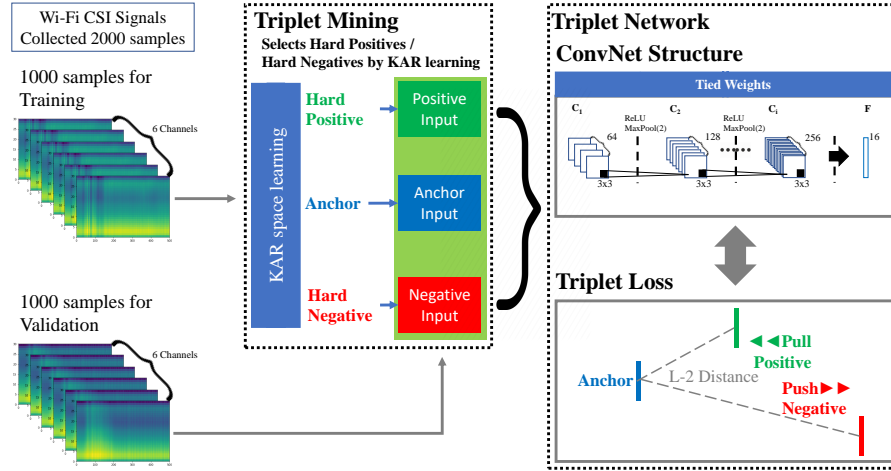


Fig. 1. Structure of the network

#### 3.1 Triplet loss

Triplet loss is used to train ConvNet to extract feature vector from triplet input. For the  $i_{th}$  anchor input signal  $\mathbf{X}_{anc,i}$ , triplet input is generated by grouping the positive input signal  $\mathbf{X}_{pos,i}$  drawn from the same identity with anchor signal and the negative input signal  $\mathbf{X}_{neg,i}$  is chosen from the another identity with

the anchor signal. Generated triplet input  $\{\mathbf{X}_{anc,i}, \mathbf{X}_{pos,i}, \mathbf{X}_{neg,i}\}$  is fed into the ConvNet structure one by one and respectively makes anchor, positive and negative feature vectors.

Let  $\mathbf{v}_{anc,i} \in \mathbb{R}^{d \times 1}$ ,  $\mathbf{v}_{pos,i} \in \mathbb{R}^{d \times 1}$  and  $\mathbf{v}_{neg,i} \in \mathbb{R}^{d \times 1}$  be three feature vectors extracted from the ConvNet structure. the triplet loss is calculated by comparing positive distance (L-2 distance from the anchor vector to the positive feature vector) and negative distance (L-2 distance from the anchor vector to the negative feature vector). ConvNet is trained to maximize gap between positive distance to negative distance more than margin  $\alpha$ . The triplet loss for the inputs can be calculated as follows:

$$loss = \sum_i^N \max \left( \left[ \|\mathbf{v}_{anc,i} - \mathbf{v}_{pos,i}\|_2^2 - \|\mathbf{v}_{anc,i} - \mathbf{v}_{neg,i}\|_2^2 + \alpha \right], 0 \right), \quad (5)$$

where N is size of the mini-batch. Note that the triplet loss can down to 0 when the negative distance is bigger than the positive distance more than preset margin  $\alpha$ . If we select proper triplet for training, we can avoid this problem and make the convergence of the loss function faster.

### 3.2 Triplet mining by kernel and the range space learning

According to [12], it is important to take hard positive and hard negative sample for faster convergence of the loss when training triplet networks. we adopted KAR space learning methods to approximate the ConvNet structure and mine the hard positive and the hard negative samples from the training dataset.

Hard positive sample means distance between feature vectors from positive signal to anchor signal is the greatest, while hard negative sample means distance between feature vectors from negative signal to anchor signal is the smallest.

Since we don't know which is the hard sample before training the network, we propose to train multilayer feedforward neural network to mine the hard positive and the hard negative from the training dataset. To train the multilayer neural network, we adopted the kernel and the range (KAR) space projection learning (see Section 2.2 for details).

### 3.3 ConvNet Structures

To design the proposed networks, we firstly need to select the feature extracting networks which convert the triplet data into a feature vector. In this work, we utilize the ConvNet structure [7] as a feature extractor since the three-dimensional data format of our preprocessed input signal can be regarded as an image data format with multiple channels.

Our ConvNet structure (See Fig 1 (b)) for the network consists of  $i$  convolutional layers  $\mathbf{C}_i$  and one fully-connected layer  $\mathbf{F}$ . The number of convolutional filters to be trained in each layer is empirically chosen as  $\{64, 128, \dots, 2^{6+i}\}$ , with fixed filter size of  $3 \times 3$  and stride of 1. The Rectified Linear (ReLU) function

as an activation function and the max-pooling layers are applied between each convolutional layers. The features from the last convolutional layer are directly flattened into a single vector. We used the sigmoid activation at ConvNet output and L-2 normalize output vectors to prevent negative distance when calculating triplet loss.

## 4 Experiments

**Dataset** In order to evaluate the verification performance of proposed system, Wi-Fi CSI signature dataset from [10] was used.

Wi-Fi CSI dataset consists of 2000 Wi-Fi CSI signature signals (4 directions  $\times$  50 identities  $\times$  10 samples) each signal is size of (500 packets  $\times$  30 subcarriers  $\times$  6 antennas).

Since every Wi-Fi signature signal has different data size, we firstly adopted the gradient operation with respect to the time instance to measure the short time energy. Data points with the highest short-time energy within the time period are then manually selected as the starting and the ending points of the in-air signature action.

Subsequently, the Fast Fourier Transform based re-sampling method [10] is implemented to unify the length of the signals. As a result, three-dimensional Wi-Fi signature signals with unified data size are obtained as the input of the ConvNet structure in the network.

### Experimental Settings

**1) Performance Evaluation** The proposed method is evaluated under comparison of the proposed method with other verification methods based on verification accuracy. verification performance is compared with existing linear methods in literature such as LSE, SVM and TER. Moreover, proposed method is compared by ConvNet based Siamese networks, we used same ConvNet model for our proposed method and Siamese network.

Validation performance of the proposed and other linear deep learning methods were evaluated in terms of Equal Error Rates (EER, %) averaged from five runs of two-fold cross-validation tests.

**2) Network Structure** We impose a triplet loss as objective function on our classifier. This objective is combined with standard backpropagation algorithm. The structure of the ConvNet network is specified in Table 1. we train the network starting with a learning rate of 0.00001 via Cross Entropy loss with a mini-batch sized of 32. We optimize the loss by the Adam optimizer with L-2 penalty of 0.0002 except for output layer. Output layer is regularized with L-2 penalty of 0.0001. We initialized all network weights in the convolutional layers from a normal distribution with zero-mean and a standard deviation of 0.01. Biases were also initialized from a normal distribution of standard deviation of

0.01, but with 0.5 mean. Training is conducted with 2000 iterations, which set empirically.

**Table 1.** The structure of ConvNet

Layer	Activation	Kernel / Stride	Input Size
Conv 1	ReLU	3x3x64 / 1	500x30x6
MaxPool 1		2x2 / 1	500x30x64
Conv 2	ReLU	3x3x128 / 1	250x15x64
MaxPool 2		2x2 / 1	250x15x128
Conv 3	ReLU	3x3x256 / 1	125x8x128
MaxPool 3		2x2 / 1	125x8x256
Dense	Sigmoid	128	63x4x256
L-2 Norm			1x1x128
Concat			1x1x384

For KARnet trained multilayer neural networks is shown below (Table 2), we set 2 layer and number of neurons are [1024,16]. initialized as uniform distribution over [0, 1). Used arctangent as activation function.

**Table 2.** The structure of KARnet

Layer	Size	Activation
Input	500x30x6	
Dense 1	1x1x1024	ArcTan
Dense 2	1x1x128	ArcTan
Output	1x1x50	

**3) Parameter Settings** For making triplet loss, 0.1 of alpha value is utilized for the proposed method. we choose output vector size of 16. For linear methods such as LSE,SVM and TER, we reduced dimension of input signal to 500x30, by averaging through subcarrier axis. Due to limitation of hardware memory. For PCA-LSE, input dimension is reduced by 40 sized features using PCA. For Triplet CNN without KAR learning, we choose signature randomly for making input triplets and used triplet loss as used in proposed methods. for Siamese Networks, we put 2 random signal as input and used binary loss for training the networks.

**Results and Discussion** Table 3 shows the best average EER performances obtained by the compared methods. The test EER was averaged from five runs of two-fold cross-validation. Comparing with ConvNet based Siamese and Triplet CNN without KAR learning, proposed methods showed the best EER perfor-

mance. Due to negative mining process included in making triplet loss. The proposed method showed much better EER than linear methods, such as SVM, LSE and TER. Since linear methods use reduced input signal.

Loss curves of proposed method are shown in Fig.2. This shows that KAR learning not only helps ConvNet to train faster, but also makes it to extract better features. As the proposed method shows steeper loss curve when training, and the variance of the validation loss is smaller.

**Table 3.** Verification Performance

Methology	Best EER (%)	Condition
Proposed(TripletCNN with KAR learning)	19.09	lr=0.00005, alpha=0.1
Triplet CNN without KAR learning	20.08	lr=0.00005, alpha=0.1
Siamese CNN	24.09	lr=0.00005, alpha=0.1
SVM(RBF)	24.31	c=1, gamma=0.01/3000
SVM(Linear)	28.23	c=1
LSE(Dimension Reduction by PCA)	30.79	40 dimension input
TER	35.75	order=1
LSE	48.44	

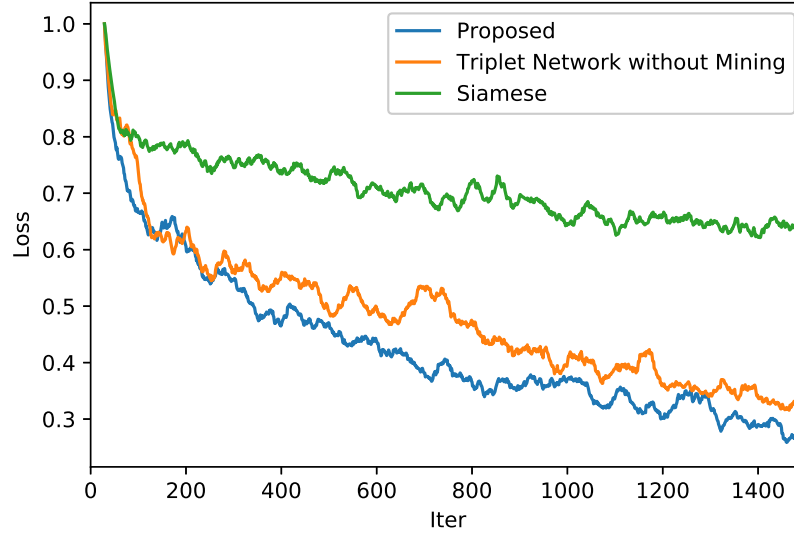
## 5 Conclusion

In this paper, we proposed a novel identity verification system based on Wi-Fi signature signals. In order to train the triplet network effectively, triplet input is mined from Wi-Fi signature signals by the kernel and the range space learning. Subsequently, ConvNet structure is adopted for feature extractor which generates feature vectors from input triplets. Finally, the triplet network is trained by matching between extracted feature vectors based on L-2 distance comparison. Our networks show good feature extracting performance, especially in low-dimension outputs and shows encouraging experimental results in terms of verification accuracy.

## References

1. Al-qaness, M., Li, F.: Wiger: Wifi-based gesture recognition system. *ISPRS International Journal of Geo-Information* **5**(6), 92 (2016)
2. Chen, W., Chen, X., Zhang, J., Huang, K.: Beyond triplet loss: a deep quadruplet network for person re-identification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 403–412 (2017)
3. Cheng, D., Gong, Y., Zhou, S., Wang, J., Zheng, N.: Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 1335–1344 (2016)





**Fig. 2.** Training Loss Curve of the Proposed Method

4. Galbally, J., Diaz-Cabrera, M., Ferrer, M.A., Gomez-Barrero, M., Morales, A., Fierrez, J.: On-line signature recognition through the combination of real dynamic data and synthetically generated static data. *Pattern Recognition* **48**(9), 2921–2934 (2015)
5. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: *International Workshop on Similarity-Based Pattern Recognition*. pp. 84–92. Springer (2015)
6. Jeon, J.H., Oh, B.S., Toh, K.A.: A system for hand gesture based signature recognition. In: *2012 12th International Conference on Control Automation Robotics & Vision (ICARCV)*. pp. 171–175. IEEE (2012)
7. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**(11), 2278–2324 (1998)
8. Li, H., Yang, W., Wang, J., Xu, Y., Huang, L.: Wifinger: talk to your smart devices with finger-grained gesture. In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. pp. 250–261. ACM (2016)
9. Malik, J., Elhayek, A., Ahmed, S., Shafait, F., Malik, M., Stricker, D.: 3dairsig: A framework for enabling in-air signatures using a multi-modal depth sensor. *Sensors* **18**(11), 3872 (2018)
10. Moon, H.C., Jang, S.I., Oh, K., Toh, K.A.: An in-air signature verification system using wi-fi signals. In: *Proceedings of the 2017 4th International Conference on Biomedical and Bioinformatics Engineering*. pp. 133–138. ACM (2017)
11. Sanmorino, A., Yazid, S.: A survey for handwritten signature verification. In: *2012 2nd International Conference on Uncertainty Reasoning and Knowledge Engineering*. pp. 54–57. IEEE (2012)

12. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
13. Toh, K.A.: Learning from the kernel and the range space. In: 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS). pp. 1–6. IEEE (2018)
14. Toh, K.A., Lin, Z., Li, Z., Oh, B., Sun, L.: Gradient-free learning based on the kernel and the range space. arXiv preprint arXiv:1810.11581 (2018)
15. Zhou, Q., Xing, J., Li, J., Yang, Q.: A device-free number gesture recognition approach based on deep learning. In: 2016 12th International Conference on Computational Intelligence and Security (CIS). pp. 57–63. IEEE (2016)