End-to-End Deep Learning-Based Human Activity Recognition Using Channel State Information

Chaur-Heh Hsieh[1], Jen-Yang Chen[2], Chung-Ming Kuo[3], Ping Wang[1]

[1] College of Artificial Intelligence, Yango University, China
[2] Department of Electronic Engineering, Ming Chuan University, Taiwan
[3] Department of Information Engineering, I-Shou University, Taiwan
chaoho1204@qq.com, jychen@mail.mcu.edu.tw, kuocm@isu.edu.tw, 51063978@qq.com

Abstract

The automatic recognition of human activities in the house using Channel State Information (CSI) has received wide attention due to the potential use of wide range of intelligent services. Two-dimensional Convolutional Neural Network (2D-CNN) is one of the most popular approaches for human activity recognition (HAR). This method first applies signal transform to convert a time-series CSI signal into a 2D image, and then uses the image to train a complex 2D-CNN model. In this paper we will present simple deep neural networks including multi-layer perceptron (MLP) and one-dimensional Convolutional Neural Network (1D-CNN) for HAR. Our proposed networks are fully end-to-end automatic learning from feature extraction/selection and classification, and do not require extra signal transform and denoising. Experimental results indicate that the proposed networks not only achieve much better recognition performance but reduces the network complexity significantly, as compared to the existing methods.

Keywords: Human activity recognition, Deep learning, Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), Channel State Information (CSI)

1 Introduction

With the rapid development of the Internet of Things and Artificial Intelligence technology, smart home and smart elderly care center have been gradually developing. The automatic identification of the activities of people in the house are very helpful for the caring purposes.

There are many daily activities at home, such as eating, taking medicine, sleeping, making phone calls, using computers, walking, cooking and cleaning. If the activities can be automatically recorded, identified and indexed, it will produce many important applications. For instance, establishment of a security monitoring system that can monitor the behaviors of elderly or children to deal with accidents such as falls (or fainting). It is also possible to build a smart search system that help users with poor memory to recall something. The system may provide answers of users to the questions such as "Where is my glasses (mobile phones, watches, etc.)?", "Is this medicine I have eaten?" This is especially important for people with poor memory. In addition, activity recognition is also important in Human-Computer Interface (HCI), such as gesture recognition for remote control of home smart products. Therefore, in recent years, the automatic analysis and identification of human activity behavior have received wide attention [1].

The methods for activity recognition in the literature can be roughly divided into two categories: device-based and device-free. In the former approach, human needs to wear devices such as accelerators or RFID. In the latter approach, it is only to build a transmitter and a receiver in the environment, and the subject does not need to wear any device at all. The device-free approach is cheaper than the device-based method. In addition, wearing a device is troublesome for most people, especially for the elderly or children. Therefore, the device-free method has received wide attention in recent years [2].

In device-free approach, the most widely studied is the visual method, which uses cameras to capture image and applies computer vision technique to identify activity types. However, the image quality is affected by many environmental factors. For example, the coverage area for detection must be within line-of-sight (LOS). Moreover, despite of improvements in image processing algorithms, the performance can degrade under low lightning conditions. In addition, from a user perspective the presence of cameras can affect privacy.

The research in HAR using WiFi signals has become active recently [3] due to its advantages in the following. WiFi signals can propagate through furniture, door and walls, and do not require line of sight (LOS), thus enabling a larger detection area.

*Corresponding Author: Jen-Yang Chen; E-mail: jychen@mail.mcu.edu.tw
DOI: 10.3966/160792642021032202004

Secondly, WiFi devices are now widely available in the house, hence no additional equipment is required to build WiFi-based HAR system, which reduces the cost of system deployment significantly. Finally, since no camera exists, there is no privacy concern.

Using wireless signals to HAR is much later than the use of video, and it can be said that it is still in its infancy. Earlier researches focus on the detection of human fall. The earliest study using CSI to detect fall activity is Wi-Fall [4], which applies CSI amplitude to distinguish it from the other three activities (sit down, stand up and walk). However, the algorithm does not take into account the various fall-like activities that occur in everyday life. To solve this problem, Anti-Fall [5] includes various fall-like activities in the dataset and uses phase of CSI signal as a feature to improve classification. In [6], an improved model named Rt-fall is proposed, which uses the phase difference of two receiving antennas to distinguish between fall events (including falls and similar falls) and general activities. Compared to Wi-Fall, Rt-fall achieves higher sensitivity and specificity.

Regarding activity recognition, Y. Wang et al. [7] propose the system E-eyes, which can identify 8 activities in the in-place activity, and 4 activities in the walking activity. It uses the CSI histogram as the fingerprint in the database. Preprocessing is also utilized, including low pass filtering and modulation and coding scheme (MCS) index filtering. Although the system performs well and the computational cost is low, the histogram technique is sensitive to environmental changes and may not be suitable for different environments.

W. Wang et al. [8] propose the CARM system, which establishes the correlation between the dynamic values of CSI and human activities to identify eight human activities of a single subject. The CARM performs PCA (Principal Component Analysis) according to the correlation between the CSI data of different sub-carriers to remove the associated noise components. It uses 12 levels of DWT (Discrete Wavelet Transform) to extract five main components as features, and then applies HMM (Hidden Markov Model) for classification.

[9] proposes the system HuAc, which combines WiFi features and Kinect skeleton joints to recognize human activity in an indoor environment with occlusion, weak light, and different perspectives. They apply the hand-crafted features into various classification models such as KNN, decision tree, random forest and SVM. HuAc achieves an average accuracy of greater than 93% using WiAR dataset that the authors created.

Yan et al. propose WiAct system [10], which can identify ten actions. In the system, a novel activity cutting algorithm is developed to detect the start and end of an activity. The Doppler shift correlation extracted from the correlation of the WiFi device's antennas is used as features to train the Extreme Learning Machine (ELM) for activity classification.

Gao et al. [11] propose an image processing framework based on deep learning for extracting discriminative deep image features from radio images. Specifically, the amplitude and phase information of CSI signals are converted into radio images, and the texture features, called raw image features, are extracted from the radio image using conventional image processing techniques. Then, an autoencoder network is designed to learn optimized deep features from the raw image features. Finally, the deep features are applied to a softmax regression model to estimate the state of the person.

The works stated above can be roughly classified into two types of methods. One is traditional machine learning method, in which it extracts features in a hand-design manner, and then trains a classification model. The other is combinational method, in which it utilizes deep learning and image processing to automatically extract features and then uses the features to learn a classification model. In addition, most of both types of methods above apply extra processing before feature extraction, including noise reduction and signal transformation [12]. Raw CSI measurements contain noises and outliers that could significantly reduce WiFi sensing performance. Noise reduction aims to reduce high frequency noises by using various filters such as moving average, median or low pass filters.

The raw CSI measurement is a time-series data. Signal transform is often used to obtain time-frequency representation of the time-series data. The typical signal transform techniques for Wifi sensing include Fast Fourier Transform (FFT), Short Time Fourier Transform (STFT), Discrete Hilbert Transform (DHT), and Discrete Wavelet Transform (DWT). The signal transform converts 1-D signal into 2-D (or more) representation.

Both types of methods above include quite complex processes. This not only reduces system processing speed but also increases implementation cost. To attack the drawbacks, this work proposes an approach based on end-to-end deep neural network (DNN), which is fully automatic from feature extraction, feature selection and classification. It does not require manual intervention (handcraft), and needs no signal transform and noise reduction. The proposed approach reduces the system complexity and achieves better recognition accuracy. The contributions of this work are summarized as follows:

(1) We investigate various deep neural networks using CSI signals for device-free human activity recognition, and evaluates their performance in terms of recognition accuracy and network complexity.

(2) The proposed MLP and 1D-CNN for activity recognition are proved to achieve much better performance with significantly lower network

complexity, as compared to the existing 2D-CNN based and SVM-based methods.

(3) The proposed approach is a fully end-to-end deep learning architecture, which does not need hand-crafted feature extraction and extra signal processing such as denoising filters and signal transforms.

The remainder of this paper is organized as follows. Section 2 describes the proposed method in details; we first explain how to construct a CSI dataset, and then illustrate the design procedure of MLP and 1D-CNN separately using the dataset. The optimization of the DNN configurations, classification error analyses, and performance comparison are described in Section 3. Finally, the conclusion is drawn in Section 4.

2 Proposed Method

The proposed HAR method aims to predict the activity type of a person using CSI signal between the wireless access point (AP) and the network interface. The method is basically an end-to-end learning architecture. Specifically, the features of CSI signals and classification model are automatically learned from input raw data. The design of this method consists of two phases, offline training and online prediction, as shown in Figure 1. During the offline training phase, the proposed DNN learns the optimal parameters of the network (called model) using a training set. The model includes the functions of feature extraction and classification, which are jointly learned from the input raw data. During the online prediction the system receives real-time data from a network interface card and then predicts what kind of activity that the person performs using the learned model.

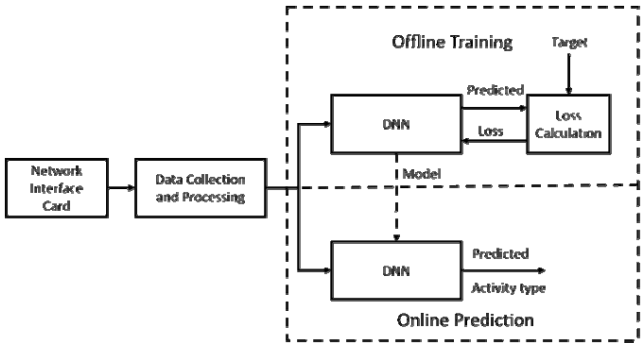

Figure 1. Proposed HAR system architecture

2.1 CSI Dataset Construction

Channel state information (CSI) is often used to measure the quality of the channel in wireless communication. It describes how a signal propagates from the transmitter to the receiver and represents the combined effect of, for example, scattering, fading, and power decay with distance [13]. Quantitative analysis of signal propagation behavior within WiFi-covered area can identify different types of disturbances. This feature allows CSI to be used in a variety of applications, such as detecting a person's location, the recognition of human activity and behavior, gesture recognition, and vital sign monitoring [14].

The channel frequency response (CFR) is often used to model the channel, which consists of amplitude response and phase response in frequency domain. Let X and Y be the frequency domain representations of transmitted and received signal vector, respectively, with carrier frequency $\boldsymbol{f_k}$. They are related by

$$Y = H \times X + N, \tag{1}$$

where N the additive white Gaussian noise, and H is the complex valued CFR, which can be estimated from X and Y. The channel frequency response of the kth subcarrier (channel) can be expressed as

$$H_k = |H_k| \exp\{j \angle H_k\}, \tag{2}$$

where $|H_k|$ is the amplitude response of the kth subcarrier, and $\angle H_k$ is its phase response.

We construct a CSI dataset which collects CSI signals reflected by 11 activities in a meeting room, as listed in Table 1. To evaluate the robustness of the proposed methods, the room we choose is with many desks in order to increase the complexity of the environment. The layout of the room is shown in Figure 2. It is a closed space with a length of 9.3 meters and a width of 5.9 meters. A total of 16 anchor points are marked in this space. The horizontal spacing between two adjacent anchor points is 1.1 meters and the vertical spacing is 0.7 meters. The router and receiver are located on the front desktop and the rear desktop, respectively, 1 meter above the ground. Each activity in Table 1 is performed 8 times by 7 volunteers with ages ranging from 20-30 years old and height from 150-180 cm.

Table 1. Human Activity Types

Activity index	Activity name	Activity index	Activity name
1	Lie	7	Stand up
2	Squat	8	Walk to right
3	Bend over	9	Walk to left
4	Standing	10	Raise hand
5	Sitting	11	Fall down
6	Sit down		

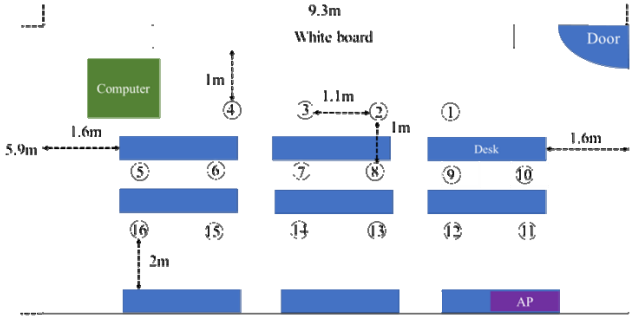

Figure 2. Experimental Environment Map

To collect CSI data of each activity, we adopt a notebook computer that installs an Intel Wi-Fi Link 5300 network interface card with three receiving antennas, and a wireless network router with two antennas. Since CSI is at the physical layer, in addition to hardware support, it requires software tool to help to extract CSI data. In this work, we apply the packet data-parsing tool developed in our previous work [15], which strengthens Linux 802.11n CSI Tool.

The number of CSIs in a packet is the product of the transmitter antenna, the receiver antenna, and the number of channels (sub-carriers). The Intel 5300 network interface card used in this work has 30 channels. Thus, the total number of CSI data per packet is 3×2×30 =180. Using Intel 5300 wireless network, we can obtain CFR containing 30 frequency responses defined in (2). In our work, only amplitude response is utilized for activity recognition. Hence, a CSI vector contains 180-D values of amplitude response.

Figure 3(a) to Figure 3(d) show the amplitude responses of CSI corresponding to four different activities including standing, fall down, walk to right and walk to left; different colors in the figures indicate signals measured by different receiving antennas. It can be seen that the differences surely exist between different activities. Therefore, it is possible to identify activities from the CSI waveforms.

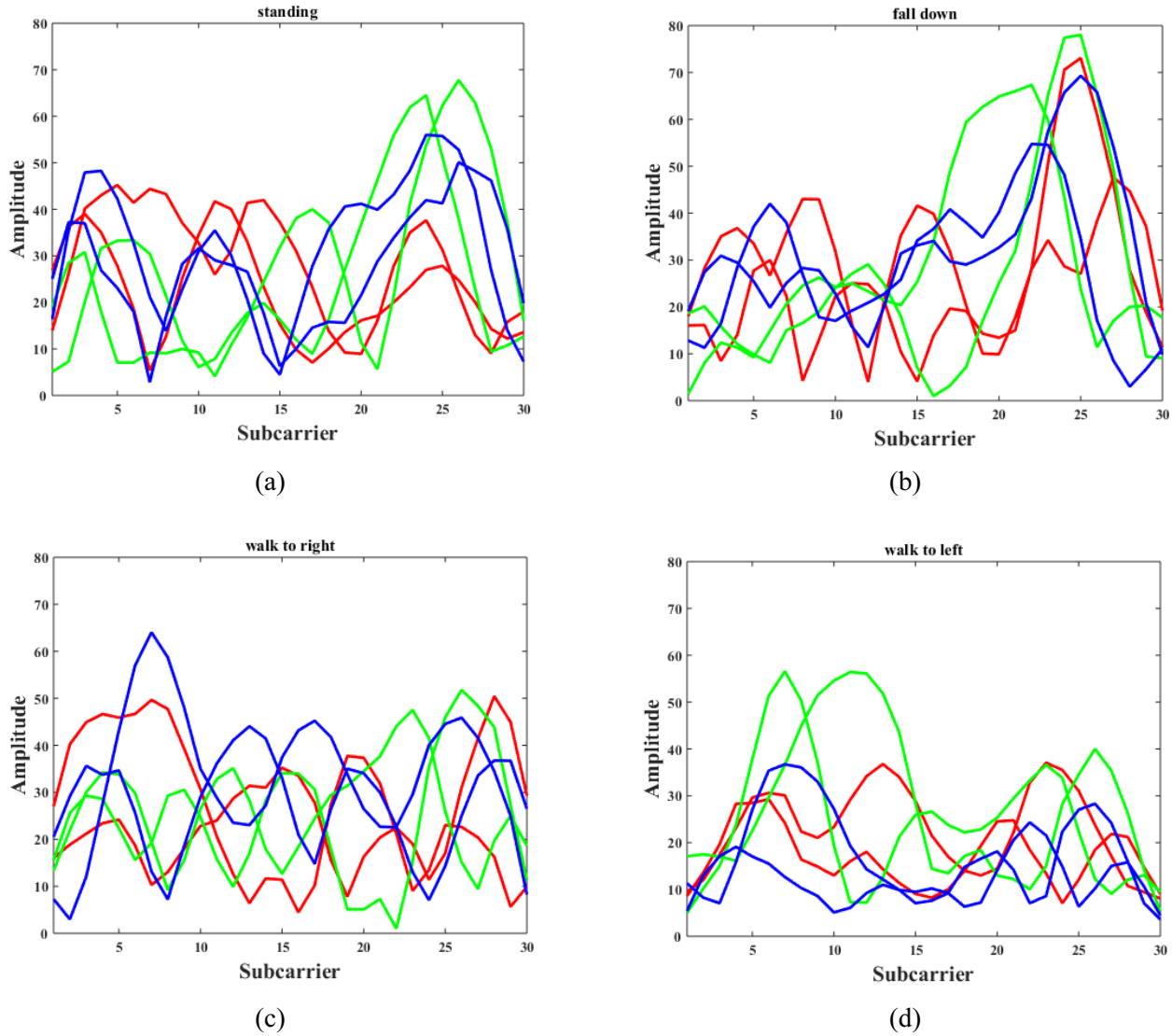

Figure 3. Amplitude responses of CSI corresponding to four different activities; different colors in the figures indicate signals measured by different receiving antennas

Some packets received are invalid, which could be caused by random transmission error. Therefore, the number of the valid packets of every activity is different. In other words, the length of activity sequences may not be the same. Due to the difficulty of putting variable-length data into a deep neural network, length normalization, which makes each activity sequence into segments with a fixed length, should be performed. Besides, the common problem in deep learning is it often needs a large amount of data for training to avoid overfitting. In our application, generating a large amount of activity samples is very difficult. For example, it is inhumane to ask voluteers to do hundreds of times of falling-down actions in

order to generate enough samples.

In order to solve the varible-length problem and increase amount of training samples to avoid overfitting, we apply a simple length normalization scheme for acivity sequences as follows. We divide a record of each activity into several segments with approximate 75% overlap between adjacent segments, and each segment contains 30 CSI vectors. For example, the first segment contains packets 1 to 30, the second segment contains packets 9 to 38, and so on. The segments from the same activity record are assigned the same label. This scheme solves the variable-length problem and is efficient for data augmentation. Table 2 shows the number of valid packets and the number of segements after length normalization for each activity. 70% of the total segments are randomly chosen for training, and 30% for testing.

Table 2. Number of Valid Packets and Number of Segments

Activity index	Total number of packets	Total number of segments
1	16114	787
2	15485	731
3	15612	757
4	14687	683
5	15236	727
6	15801	767
7	16291	793
8	15212	735
9	14905	703
10	15273	723
11	16524	804

2.2 Deep Neural Network Design

Two types of deep neural networks are considered in this work: MLP and 1D-CNN. A MLP consists of an input layer and an output layer with several hidden layers of nonlinearly-activating nodes. Since MLPs are fully connected, each node in one layer connects with a certain weight to every node in the following layer, as described by

$$y^{(i)} = f^{(i)}(W^{(i)}w^{(i)} + B^{(i)}), \quad \textbf{(3)}$$

where $x^{(i)}$ is the input, $y^{(i)}$ is the output, $W^{(i)}$ is the weight matrix, $b^{(i)}$ is the bias vector, and $f^{(i)}$ is the activation function [16]. The output of the previous layer is the input of the current layer, i.e., $x^{(i)} = y^{(i-1)}$. The first layer $x^{(1)}$ is the original input, and the last layer $y^{(N)}$ is the final output, i.e., the classification results. The weights and biases form the parameter set θ of the network, which are learnt in the training process.

In general, a CNN is composed of a number of convolutional layers for feature extraction, where each layer is usually followed by a pooling layer which is also followed by one or more fully connected layers for classification. CNN is effective for learning local features. As compared to MLP, CNN involves low computational complexity during training and prediction due to shared convolution kernels. The weights of the kernels construct the parameter set to be learned. 2D-CNN has been widely used and has achieved great success in image feature extraction and classification problems in the literature [17-20]. The CSI belongs to 1-D time-series signal, thus 1-D CNN is more suitable for the representation of the CSI than 2-D CNN. In addition, the model of 1-D CNN is simpler than that of 2-D CNN, which will reduce the system complexity. Furthermore, it needs extra signal transform if 2D-CNN is employed to process time-series CSI signal. Therefore, we select 1-D CNN instead of 2-D CNN. We use 1-D CNN to extract the hierarchical features of CSI from low-level to high-level features. The high-level features extracted are then used to train the softmax classifier in the final layer of the network.

The proposed 1D-CNN consists of several convolution units to extract hierarchical features from low level to high level. Each convolution unit contains 1-D convolution, batch normalization, and activation function. Batch normalization is applied after convolution and before activation since it is helpful for improving the performance and the stability of deep neural networks [21]. Note that the polling is not used in our 1D-CNN since it degrades the performance in our experiment.

In our work, total number of classes is 11, hence the size of the output layer of MLP or 1D-CNN is thus 11. The size of the input layer is 180, which is determined by the dimension of CSI signal.

For either MLP or CNN, the ReLu function in (4) is adopted as the activation function in the hidden layers to avoid vanishing gradient problem [16]. And the softmax activation function, as defined in (5), is employed in the output layer, which maps the real-value input into prediction probability in the range of [0, 1].

$$\sigma(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, \; j = 1, 2, \ldots, K. \quad \textbf{(4)}$$

In addition, the dropout is also employed between the two hidden layers to avoid overfitting [16]. For MLP and 1D-CNN, several configurations are implemented and evaluated, which will be discussed in the next section.

We apply the mini-batch gradient descent (MBGD) algorithm [5] to train MLP and 1D-CNN model. MBGD computes the gradient of the loss function J with respect to the parameter set θ for the every mini-

batch of n training examples, and then performs an update iteratively by

$$\theta_t = \theta_{t-1} - \rho \nabla_\theta J(\theta; x^{(i:x+n)}; y^{(i:y+n)}), \quad \textbf{(5)}$$

where x and y denote the target output and the predicted output vectors of the network; ∇_θ is gradient operator; ρ is the learning rate. The MBGD utilizes the backpropagation (BP) scheme to compute gradient of the loss function. BP is an efficient gradient computation scheme which calculates the loss (error) function between the ground truth and network prediction, and back propagates the error of each layer from output to input.

Several loss functions have been presented in the literature such as mean square error (MSE), Kullback Leibler (KL) Divergence and cross-entropy [16]. In this work, we choose cross-entropy as loss function. The learning of deep neural networks is to minimize the loss function $J(\theta, x, y)$ defined in (6) with respect to network parameter set θ

$$J(\theta, x, y) = -\sum_{i=1}^{N} x_i \log y_i. \quad \textbf{(6)}$$

The cross-entropy indicates the difference between the amount of information contained in x and the amount of information contained in y. If the value of the cross-entropy is small, then the predicted output is close to the target output. It is noted that L_1 or L_2 regularization is usually included into the loss function to avoid overfitting [16]. However, in our case, the regularization does not improve prediction performance, thus we do not use it for simplicity.

In MBGD training, choosing a proper fixed learning rate can be difficult. A learning rate that is too small results in slow convergence, while a learning rate that is too large can hinder convergence and cause the loss function to fluctuate around the minimum or even to diverge. To solve the problem, several gradient descent optimization algorithms with different learning rate schedules have been presented such as Adagrad, Adadelta, Adam and RMSprop [22]. The Adam (Adaptive Moment Estimation) algorithm is employed in this work since it has been experimentally proven to work well in practice [22]. The method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. The update algorithm of the parameter set of the network is given by [22]

$$\theta_t \to \theta_{t-1} - \frac{\eta \hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}, \quad \textbf{(7)}$$

where η is a fixed learning step size; ε is a very small constant; $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected first moment estimate and the biased-corrected second moment estimate, which are calculated by

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \hat{v}_t = \frac{v_t}{1-\beta_2^t}, \quad \textbf{(8)}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \quad \textbf{(9)}$$

$$v_t = \beta_2 m_{t-1} + (1-\beta_2) g_t^2, \quad \textbf{(10)}$$

where g_t is the gradient of lost function at time t; β_1 and β_2 are the attenuated constants for the first moment and second moment, respectively.

3 Numerical Analysis

3.1 Optimization of DNN Configurations

In this subsection, we explain how to obtain the optimal architecture for MLP and 1D-CNN. We compare the performance of different network configurations in terms of two metrics: classification accuracy and total number of parameters. The second metric reveals the complexity of a network. High complex network often needs a large amount of training data to avoid overfitting, and also requires high computational load. We apply the popular deep learning platform Keras to evaluate the two metrics of all deep learning architectures discussed in this paper.

The design of architectures of DNNs need to set a large variety of hyper-parameters including the number of hidden layers, nodes of every layer, learning rate, batch size, etc. This is so called hyperparameter optimization. The goal of the hyperparameter optimization is to find the best set of hyper-parameters which achieves the greatest performance given network complexity. Although some general strategies such as grid search and random search have been presented [16], the optimization is still very challengeable due to the required huge computation. In this work, we apply the process of try and error based on our experiences to obtain a better network configuration in an efficient way.

Five MLP configurations are designed in our experiment. The first four configurations utilize five hidden layers, but they have different number of nodes: 180,360,720 and 960. The last configuration has five hidden layers with different numbers of neurons in each layer. The results shown in Table 3 indicate that the architecture of 960x5 achieves the best classification accuracy.

For 1D-CNN, three hidden convolution blocks with filters of 16, 32, 32 are designed. In such architecture, the performance metrics under different kernel sizes are evaluated. As shown in Table 4, kernel size of 7 achieves the best classification accuracy.

Table 3. Comparison of MLP Configurations

MLP Configurations	Total Number of Parameters	Classification Accuracy (%)
180 x 5	1,107,371	86.48
360 x 5	2,473,931	86.76
720 x 5	5,984,651	87.62
960 x 5	8,901,131	89.00
720,620,520,420,320	5,023,251	87.49

Table 4. Comparison of 1D-CNN Configurations

Kernel sizes	Number of filters	Total Number of Parameters	Classification Accuracy (%)
3	55-55-55	66,836	91.39
5	55-55-55	98,736	91.76
7	55-55-55	130,636	93.22
9	55-55-55	162,536	92.08
11	55-55-55	194,436	91.84
13	55-55-55	226,336	91.15
15	55-55-55	258,236	90.46
17	55-55-55	290,136	90.82
19	55-55-55	322,036	90.99
21	55-55-55	353,936	90.50
23	55-55-55	385,836	90.46
25	55-55-55	417,736	91.55

As stated before, we adopt Adam, which is an adaptive learning rate algorithm. However, our experiences indicate that the initial learning rate in Adam will affect the convergence performance. Therefore, we try various initial values of learning rate in the best architectures of MLP and 1D-CNN, respectively. The results are shown in Table 5. It is seen that the initial learning rates of 0.2 and 0.01 yield the best classification accuracy for MLP and 1D-CNN, respectively In such case, MLP achieves better classification accuracy of 0.24% than 1D-CNN. However, the total number of parameters of MLP is more than 68 times of that of 1D-CNN. This indicates that the 1D-CNN has much less network complexity and thus it reduces implementation cost significantly.

Table 5. Comparison of MLP and 1D-CNN over Initial Learning Rates

Learning Rate	Total Number of Parameters		Classification Accuracy (%)	
	MLP	1D-CNN	MLP	1D-CNN
0.2	8,901,131	130,636	93.46	92.49
0.15	8,901,131	130,636	92.28	91.96
0.1	8,901,131	130,636	91.96	91.92
0.01	8,901,131	130,636	89.00	93.22
0.001	8,901,131	130,636	84.90	88.47
0.0001	8,901,131	130,636	63.46	56.47

3.2 Error Analysis

We use the best configurations above for classification error analysis and comparison. Table 6 shows the confusion matrix of MLP, which can be used to analyze the misclassification error. It is seen that the maximal prediction error comes from the situation that class 3 (bend over) is miss-classified into class 2 (squat). Table 7 displays the confusion matrix of 1D-CNN. Same as MLP, the biggest classification error comes from the fact that predicting "bend over" into "squat".

To figure out the reasons of the misclassification, we measure the similarity between each CSI sample of the two classes by calculating their correlation coefficients. Assume that the n-dimensional sample vectors of two classes are denoted as $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$, respectively, the sample correlation coefficient of two vectors is defined as [23]

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}, \quad \textbf{(11)}$$

where $\overline{x}$ and $\overline{y}$ are the sample means of x and y, respectively.

Figure 4 shows the CSI sample waveforms corresponding to the top three of correlation coefficient values between class 2 and class 3. It is seen that the waveforms on the top are very similar to corresponding waveforms below respectively such that the correlation coefficient values are rather high even the waveforms come from different classes, which likely results in the misclassification. Figure 5 demonstrates the CSI sample waveforms corresponding to the top three of correlation coefficient values between class 1 and class 11. It is obvious that the waveforms from different classes are not similar, and the correlation coefficients are low. Therefore, the misclassification rates are zero or very small for the two classes, as shown in Table 6 and Table 7.

3.3 Evaluation of Different Methods

In this section, to prove the effectiveness of the proposed MLP and 1D-CNN, we implement two categories of existing methods: (a) 1D-CNN plus SVM, and (b) 2D-CNN with various signal transformations. The first category extracts the features of 1D time-series CSI signal using 1D-CNN, and then trains SVM classification model using the extracted features. Here, the method is abbreviated as CNN-SVM. The second category applies various signal transformations to convert 1D CSI data into a 2D image, and then uses the 2D image to train a 2D-CNN model. The transformations [24] implemented here include Short-Time Fourier transform (STFT), Continuous Wavelet Transform (CWT), and Discrete Wavelet Transform (DWT). These methods are denoted as STFT-2DCNN, CWT-2DCNN and DWT-2DCNN, respectively.

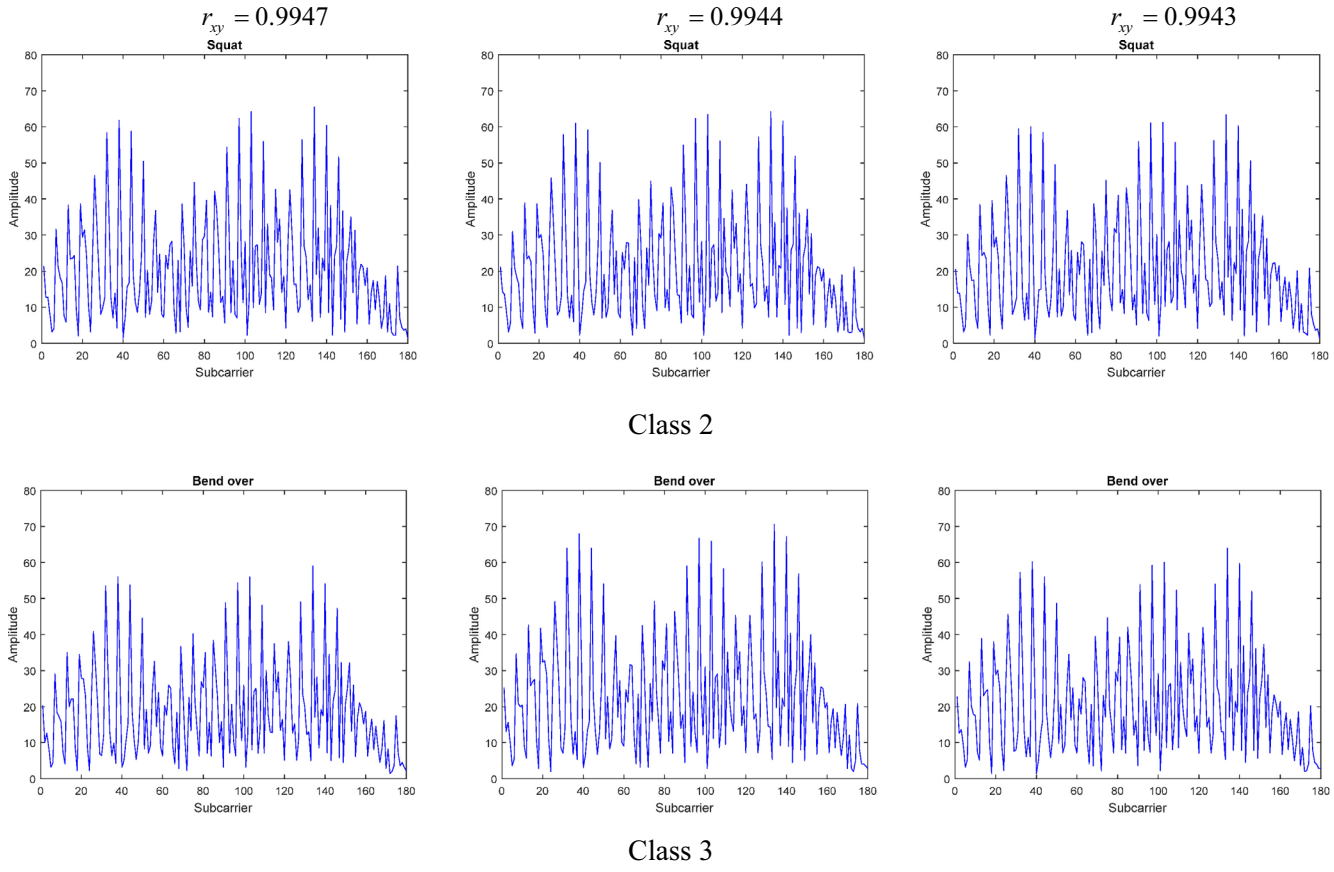

Figure 4. CSI waveforms corresponding to the top three of correlation coefficient values (from left column to right column) between class 2 and class 3

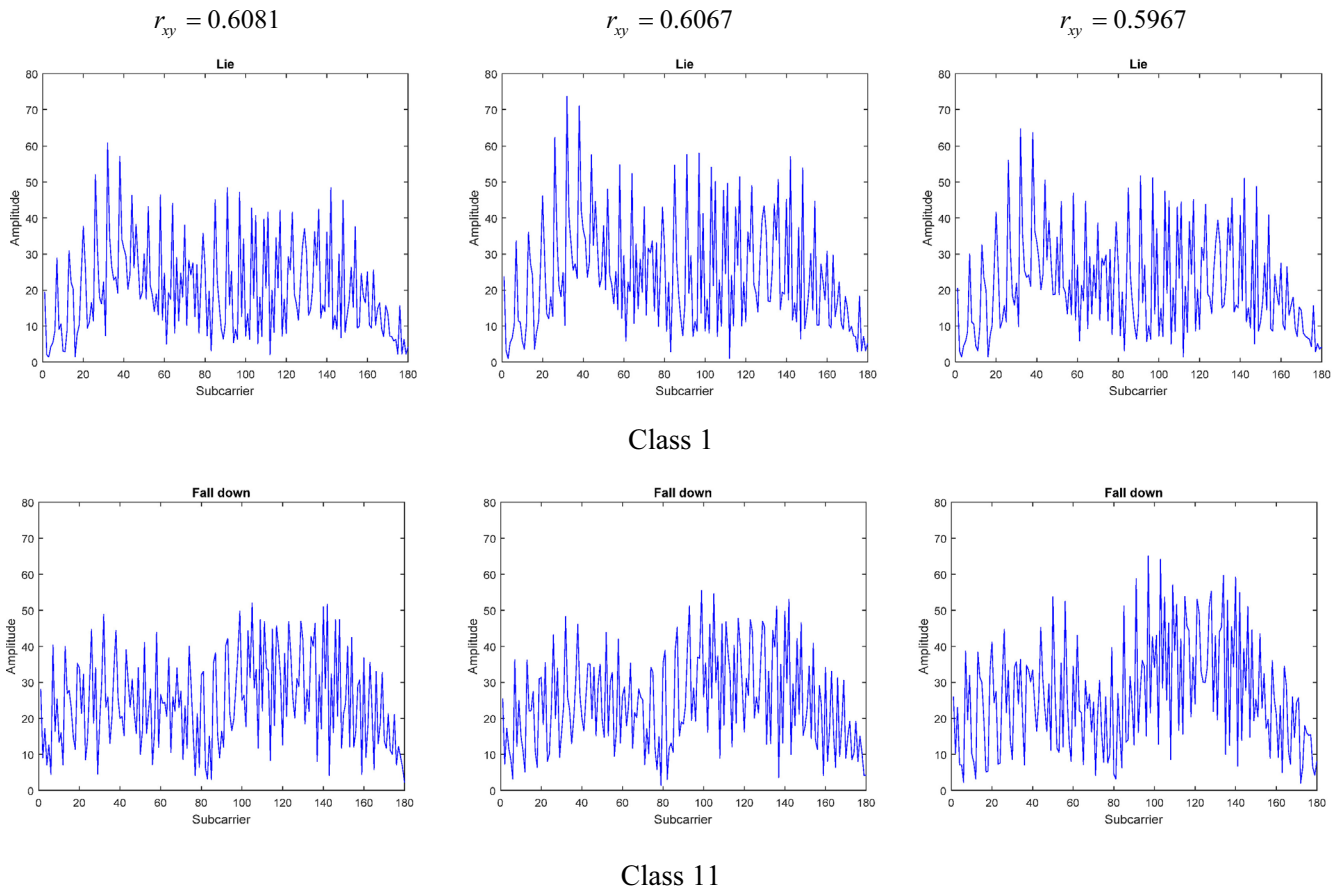

Figure 5. CSI waveforms corresponding to the top three of correlation coefficient values (from left column to right column) between class 1 and class 11

Table 6. Confusion Matrix of MLP

Predict \ Label	1	2	3	4	5	6	7	8	9	10	11
1	99.21	0.00	0.40	0.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.44	86.73	8.85	1.77	0.00	0.00	0.88	0.88	0.44	0.00	0.00
3	2.39	4.78	86.12	0.96	0.00	0.00	0.48	2.39	2.87	0.00	0.00
4	0.00	1.40	1.40	96.28	0.00	0.47	0.00	0.47	0.00	0.00	0.00
5	0.00	0.00	0.00	0.47	98.13	0.00	0.00	0.00	0.00	1.40	0.00
6	0.00	0.00	2.17	0.00	0.43	94.78	2.17	0.00	0.00	0.43	0.00
7	0.00	0.00	0.80	0.40	0.40	4.00	93.20	0.00	0.40	0.80	0.00
8	0.96	0.00	0.00	1.92	0.00	0.00	0.48	88.94	6.73	0.96	0.00
9	0.51	2.56	3.08	1.03	0.00	0.00	0.00	5.13	86.15	1.03	0.51
10	0.00	0.41	0.41	0.41	1.24	0.41	0.41	0.00	0.00	96.69	0.00
11	0.00	0.00	0.00	0.00	0.00	0.45	0.00	0.00	0.00	0.00	99.55

Table 7. Confusion Matrix of 1D-CNN

Predict \ Label	1	2	3	4	5	6	7	8	9	10	11
1	99.60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.40	0.00	0.00
2	1.38	83.94	10.09	1.38	0.00	0.00	0.46	0.92	1.38	0.46	0.00
3	0.99	3.45	95.07	0.00	0.00	0.00	0.00	0.00	0.49	0.00	0.00
4	0.00	0.93	0.00	96.73	0.00	0.00	0.93	1.40	0.00	0.00	0.00
5	0.00	0.00	0.44	0.00	97.81	0.88	0.88	0.00	0.00	0.00	0.00
6	0.00	0.45	0.45	0.00	0.00	93.67	4.52	0.00	0.00	0.90	0.00
7	0.42	1.27	0.42	0.42	0.00	4.64	91.98	0.42	0.00	0.42	0.00
8	0.00	3.00	0.00	2.15	0.00	0.00	0.00	85.41	9.01	0.43	0.00
9	0.45	3.15	1.35	0.90	0.00	0.00	0.45	5.86	86.94	0.90	0.00
10	0.48	1.45	0.48	0.00	1.45	0.48	1.45	0.00	0.00	94.20	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.00	99.56

The three 2D-CNN based methods are designed with three 2D convolutional blocks whose 2D filter structures are extended from the proposed 1D-CNN. We evaluate the performance of the three architectures under different 2D kernel sizes, and the results are shown in Tables 8 to 10, respectively. For each method, we select the best configuration which gives the largest classification accuracy, as marked in bold. It is seen that among the three methods, STFT-2DCNN performs similar to CWT-2DCNN, and DWT-2DCNN is the worst.

Table 8. Comparison of STFT-2DCNN over Kernel Sizes

Kernel sizes	Number of filters	Total Number of Parameters	Classification Accuracy(%)
5 x 5	16-32-32	14446011	84.00
7 x 7	16-32-32	14448072	84.38
9 x 9	16-32-32	14534715	84.50
11 x 11	16-32-32	14598075	85.11
13 x 13	**16-32-32**	**14674107**	**85.12**
15 x 15	16-32-32	14762811	85.02

Table 9. Comparison of CWT-2DCNN over Kernel Sizes

Kernel sizes	Number of filters	Total Number of Parameters	Classification Accuracy(%)
5 x 5	16-32-32	14446011	85.14
7 x 7	16-32-32	14448072	84.81
9 x 9	16-32-32	14534715	85.18
11 x 11	16-32-32	14598075	85.08
13 x 13	**16-32-32**	**14674107**	**85.25**
15 x 15	16-32-32	14762811	85.00

Table 10. Comparison of DWT- 2DCNN over Kernel Sizes

Kernel sizes	Number of filters	Total Number of Parameters	Classification Accuracy(%)
5 x 5	16-32-32	14446011	72.27
7 x 7	16-32-32	14448072	72.42
9 x 9	**16-32-32**	**14534715**	**72.50**
11 x 11	16-32-32	14598075	72.27
13 x 13	16-32-32	14674107	72.15
15 x 15	16-32-32	14762811	72.04

The comparison of all methods is listed in Table 11. It is obvious that our proposed MLP and 1D-CNN methods performs much better than 2D-CNN-based methods either from classification accuracy or system complexity. In addition, the proposed methods reduce the overhead of 2D-CNN that requires converting time-series data into 2D images.

Table 11. Performance Comparison of Proposed Methods and 2D-CNN-Based Methods

Methods	Total Number of Parameters	Classification Accuracy (%)
MLP	8,901,131	93.46
1D-CNN	130,636	93.22
STFT-2DCNN	14,674,107	85.12
CWT-2DCNN	14,674,107	85.25
DWT-2DCNN	14,534,715	72.50

SVM-based methods are very popular traditional machine learning approaches for HAR. Here we also implement pure SVM and CNN-SVM. The pure SVM means that the raw CSI data is directly used to train a SVM model. It obtains classification accuracy of 80.83%. The CNN-SVM, which combines CNN for feature extraction and SVM for classification, obtains classification accuracy of 90.90%. Again, the proposed 1D-CNN and MLP outperform the SVM-based methods.

4 Conclusions

In this paper, we have proposed end-to-end deep learning-based methods for the recognition of human activities using raw CSI signals. Through careful design and optimization of hyperparameters, the proposed MLP and 1D-CNN achieves good performance with classification accuracy more than 93%, which outperform significantly the existing 2D-CNN methods and SVM-based methods. In addition, the network complexity of the proposed neural networks is much lower than that of 2D-CNN methods. Compared to MLP, the 1D-CNN achieves slightly lower classification accuracy but with much less network complexity. Development of effective and efficient deep neural networks for the joint task of indoor localization and human activity recognition [2] will be investigated in the future

Acknowledgments

This work was supported in part by Ministry of Science and Technology, under the grant MOST 108-2221-E-130-009.

References

[1] M. A. A. Al-qaness, M. A. Elaziz, S. H. Kim, A. A. Ewees, A. A. Abbasi, Y. A. Alhaj, A. Hawbani, Channel State Information from Pure Communication to Sense and Track Human Motion: A Survey, *Sensors*, Vol. 19, No. 15, Article 3329, pp.1-27, July, 2019.

[2] J. Wang, X. Zhang, Q.-H. Gao, H. Yue, H.-Y. Wang, Device-Free Wireless Localization and Activity Recognition: A Deep Learning Approach, *IEEE Transactions on Vehicular Technology*, Vol. 66, No. 7, pp. 6258-6267, July, 2017.

[3] H. Liu, H. S. Darabi, P. Banerjee, J. Liu, Survey of Wireless Indoor Positioning Techniques and Systems, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 37, No. 6, pp. 1067-1080, November, 2007.

[4] Y.-X. Wang, K.-S. Wu, L.-M. Ni, Wifall: Device-free Fall Detection by Wireless Networks, *IEEE Transactions on Mobile Computing*, Vol. 16, No. 2, pp. 581-594, February, 2017.

[5] D.-Q. Zhang, H.-S. Wang, Y.-S. Wang, J.-Y. Ma, Anti-fall: A Non-intrusive and Real-time Fall Detector Leveraging CSI from Commodity Wifi Devices, *International Conference on Smart Homes and Health Telematics*, Springer, Geneva, Switzerland, 2015, pp. 181-193.

[6] H. Wang, D.-Q. Zhang, Y.-S. Wang, J.-Y. Ma, Y.-X. Wang, S.-G. 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, February, 2017.

[7] Y. Wang, J.-A. Liu, Y.-Y. Chen, M. Gruteser, J. Yang, H.-B. Liu, E-eyes: Device-free Location-oriented Activity Identification using Fine-grained Wi-fi Signatures, *20th annual international conference on Mobile computing and networking*, Maui, HI, USA, 2014, pp. 617-628.

[8] W. Wang, A.-X. Liu, M. Shahzad, K. Ling, S.-L. Lu, Understanding and Modeling of Wi-fi Signal based Human Activity Recognition, *International Conference on Mobile Computing and Networking*, Paris, France, 2015, pp. 65-76.

[9] L.-L. Guo, L. Wang, J.-L. Liu, W. Zhou, B.-X. Lu, HuAc: Human Activity Recognition using Crowdsourced WiFi Signals and Skeleton Data, *Wireless Communications and Mobile Computing*, Vol. 2018, pp. 1-15, January, 2018.

[10] H. Yan, Y. Zhang, Y.-J. Wang, K.-G. Xu, WiAct: A passive WiFi-based Human Activity Recognition System, *IEEE Sensors Journal*, Vol. 20, No. 1, pp. 296-305, January, 2020.

[11] Q.-H. Gao, J. Wang, X.-R. Ma, X.-Y. Feng, H.-Y. Wang, CSI-based Device-free Wireless Localization and Activity Recognition using Radio Image Features, *IEEE Transactions on Vehicular Technology*, Vol. 66, No. 11, pp. 10346-10356, November, 2017.

[12] Y.-S. Ma, G. Zhou, S.-Q. Wang, WiFi Sensing with Channel State Information: A Survey, *ACM Computing Surveys*, Vol. 52, No. 3, Article 46, June, 2019.

[13] X.-Y. Wang, L.-J. Gao, S.-W. Mao, S. Pandey, CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach, *IEEE Transactions on Vehicular Technology*, Vol. 66, No. 1, pp. 763-776, January, 2017.

[14] J.-A. Liu, H.-B. Liu, Y.-Y. Chen, Y. Wang, C. Wang, Wireless Sensing for Human Activity: A Survey, *IEEE Communications Surveys & Tutorials (Early Access)*, August, 2019.

[15] C.-H. Hsieh, J.-Y. Chen, B.-H. Nien, Deep Learning-based Indoor Localization using Received Signal Strength and Channel State Information, *IEEE Access*, Vol. 7, pp. 33256-33267, March, 2019.

[16] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning,* MIT Press, 2016.

[17] W. Rawat, Z.-H. Wang, Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review, *Neural Computation*, Vol. 29, No. 9, pp. 1-98, September, 2017.

[18] Z. Li, D. Hoiem, Learning without Forgetting, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40, No. 12, pp. 2935-2947, December, 2018.

[19] K.-M. He, X.-G. Zhang, S.-Q. Ren, J.-A. Sun, Deep Residual Learning for Image Recognition, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, LasVegas, NV, USA, 2016, pp. 770-778.

[20] C. Szegedy, W. Liu, Y.-Q. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going Deeper with Convolutions, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 1-9.

[21] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, *32nd International Conference on International Conference on Machine Learning*, Lille, France, 2015, pp. 448-456.

[22] S. Ruder, An Overview of Gradient Descent Optimization Algorithms. arXiv:1609.04747 [cs.LG], 2016. Available: https://arxiv.org/abs/1609.04747

[23] S. Boslaugh, *Statistics in Nutshell*, 2nd Edition, O'Reilly Media, 2012.

[24] S. Yousefi, H. Narui, S. Dayal, S. Ermon, S. Valaee, *IEEE Communications Magazine*, Vol. 55, No. 10, pp. 98-104, October, 2017.

Biographies

Chaur-Heh Hsieh received Ph.D. degree in Electrical Engineering in 1990 from Chung Cheng Institute of Technology, Taiwan, R.O.C. He is currently a Professor of College of Artificial Intelligence, Yango University, Fuzhou, China. He is an IET Fellow. His current research interests include signal and image processing, computer vision, and deep learning.

Jen-Yang Chen received the M.S. degree in electrical engineering from the Tatung University, Taipei, Taiwan, R.O.C., in 1992 and the Ph.D. in electrical engineering from Tamkang University, in 2000. He is currently a full professor in the Department of Electronic Engineering at Ming Chuan University. His research interests include deep learning and intelligent control.

Chung-Ming Kuo received the M.S. and Ph.D. degrees all in electrical engineering from Chung Cheng Institute of Technology, Taiwan, in 1988 and 1994, respectively. He is currently a Professor in the Department of Information Engineering at I-Shou University. His current research interests include image/video processing, computer vision, and machine/deep learning.

Ping Wang was born in Fuzhou City, China, in July 1984. She received the Master in software engineering from Fuzhou University in 2012. She is an experienced senior engineer in big data development and application. Her research interests include deep learning, graphics and image processing, big data and Internet of Things.