T2R2 東京科学大学 リサーチリポジトリ
Science Tokyo Research Repository

論文 / 著書情報
Article / Book Information

Title	User Adaptation of Convolutional Neural Network for Human Activity Recognition
Authors	Shinya Matsui, Nakamasa Inoue, Yuko Akagi, Goshu Nagino, Koichi Shinoda
Citation	2017 25th European Signal Processing Conference (EUSIPCO), pp. 753-757
Pub. date	2017, 10
DOI	http://dx.doi.org/10.23919/EUSIPCO.2017.8081308
Note	(c) 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.
Note	This file is author (final) version.

Powered by T2R2 (Science Tokyo Research Repository)

User Adaptation of Convolutional Neural Network for Human Activity Recognition

Shinya Matsui*, Nakamasa Inoue†, Yuko Akagi*, Goshu Nagino* and Koichi Shinoda†
*Synergistic Solutions Initiative, Asahi Kasei Corporation, Kanagawa, Japan
Email: {matsui.sk, akagi.yh, nagino.gb}@om.asahi-kasei.co.jp
†Tokyo Institute of Technology, Tokyo, Japan
Email: {inoue, shinoda}@ks.cs.titech.ac.jp

***Abstract*—Recently, monitoring human activities using smartphone sensors, such as accelerometers, magnetometers, and gyroscopes, has been proved effective to improve productivity in daily work. Since human activities differ largely among individuals, it is important to adapt their model to each individual with a small amount of his/her data. In this paper, we propose a user adaptation method using Learning Hidden Unit Contributions (LHUC) for Convolutional Neural Networks (CNN). It inserts a special layer with a small number of free parameters between each of two CNN layers and estimates the free parameters using a small amount of data. We collected smartphone data of 43 hours from 9 users and utilized them to evaluate our method. It improved the recognition performance by 3.0% from a user-independent model on average. The largest improvement among users was 13.6%.**

***Index Terms*—Human activity recognition, User adaptation, Convolutional neural network, Learning hidden unit contributions.**

I. Introduction

To increase the productivity of office workers, it is important to improve their time-management skills. As its first step, we should monitor their activities in their daily work. Recently, wearable sensors have become cheaper, smaller, and more precise, and can be used for this purpose. Many studies have been done in Human Activity Recognition (HAR) using them, where signal processing and pattern recognition technology are utilized to automatically classify human activities such as walking, running, sitting. Examples of these sensors include accelerometers, magnetometers, and gyroscopes.

As it is annoying for workers to wear extra devices for HAR, we would like them to use their own smartphones which are recently equipped with those sensors. But workers wear their smartphones quite differently. Some hold them in one hand, others put them in their pockets, or in their bags. Such differences in holding styles make sensor outputs significantly different, and thus, would degrade the HAR performance.

Here, we assume that, even if the holding styles have a large variety, they are quite limited for each worker; a worker wears his/her smartphone in the same way during most of their working hours. User adaptation techniques, where we modify user-independent (UI) model into user-dependent (UD) model with a small amount of data from a user, are expected to be effective to improve HAR performance.

Recently, deep learning has achieved high performance for recognizing multimedia data including natural language, image, video, and speech. They utilize various kinds of neural networks including Deep Neural Networks (DNNs), Convolutional Neural Networks (DNNs), and Recurrent Neural Networks (RNNs). Several studies applied them to HAR and proved their effectiveness [3], [4], [5], [6]. However, their studies used wearable sensors differently from smartphones, and did not utilize such model adaptation techniques. In the speech recognition field, on the other hand, there have been many studies of speaker adaptation in which a speech model is updated by using a small number of utterances from each user. Such techniques can be used for user adaptation in HAR.

In this paper, we propose a user adaptation method for HAR using Convolutional Neural Networks (CNN) [1]. It employs Hidden Unit Contributions (LHUC) [2] recently proposed for speaker adaptation of DNNs. LHUC inserts a special layer with a small number of free parameters between each of two hidden layers and estimates the free parameters using a small amount of data. We apply it to the CNN for HAR. We collected smartphone data of 43 hours from 9 users and used them for evaluating the proposed method.

This paper is organized as follows: Section 2 introduces some related works. Section 3 explains a CNN-based HAR framework. Section 4 proposes LHUC to this framework. Section 5 presents and examines the experimental results, and Section 6 concludes this paper.

II. Related Work

Several HAR methods using machine learning have been proposed [3], [4], [5], [6]. These methods extract useful features from sensor data and then recognize human activities by classifiers such as Support Vector Machines (SVMs), Gaussian Mixture Models (GMMs) or Hidden Markov Models (HMMs). In these methods, however, researchers or developers have had to decide adequate features for the task by trial and error.

Recently, deep learning has obtained high performances for recognizing multimedia data such as natural language, image, video, and speech. One of its advantages is that it includes feature extraction process in its modeling. Several HAR methods using deep learning have been also studied [7], [8], [9]. They were evaluated by using public datasets [10], [11] recorded by smartphones, but their locations in a body were fixed. For example, they were located in a pocket [10], or on the users waist [11]. Since smartphones can be held in

various positions, it is unclear whether these methods can be used in real situations.

User adaptation has been studied in speech recognition. Especially, user adaptation by changing parameters of acoustic models for each user has been extensively studied [12], [13]. In recent years, several user adaptation techniques for DNNs have been proposed. For example, a feature-domain transform-based approach, feature-space Maximum Likelihood Linear Regression (fMLLR) has been proposed [18]. Swietojanski et al. [2] proposed an effective model-based neural network adaptation technique that learns speaker-specific hidden unit contributions given adaptation data. Their research modifies the speaker-independent model by estimating a set of speaker-dependent parameters with a some amount of adaptation data from the specific user. However, there have been no researches solving the user adaptation problems for activity recognition.

III. Data Collection

In order to verify our proposed method, we collected evaluation data. Table I lists the conditions for recording evaluation data. The activities we recorded are walking (walk), running (run), stationary state (still), riding a train (train), riding a car or a bus (car/bus), and cycling (cycle). As the sensor for recording, we used an accelerometer, a magnetometer and a gyroscope. We recorded evaluation data with many holding styles: in a chest pocket, in a trousers pocket, in bag, in a bicycle basket, hand held with operations such as scrolling screen or tapping screen, hand held without operations such as just viewing a screen or swinging arm. The number of users is 9, and the amount of recorded total data was about 43 hours (4-5 hours for each). In order to prevent data bias for learning or testing, we collected fixed time-length data for each activity class and for holding style in each user.

IV. CNN based HAR

This section presents our user-independent (UI) model using a CNN for HAR. Here, the UI model is a model trained on sensor data from many users and is used as a base model for user adaptation in the next section. Unlike a feed-forward neural network, a CNN is a neural network composed not only of fully connected (FC) layers but also of convolutioned layers and pooling layers. A CNN can extract the features of local parts of the input.

TABLE I
Recording conditions of evaluation data

Number of users	9	**Data amount (hour)**	43.0
Activity class (6)	Train, Car/Bus, Bicycle, Walk, Run, Still		
Holding style (5)	Hand held with operation, Hand held without operation, In pocket, In bag, In bicycle basket		
Sensors (3)	accelerometer, magnetometer, gyroscope		

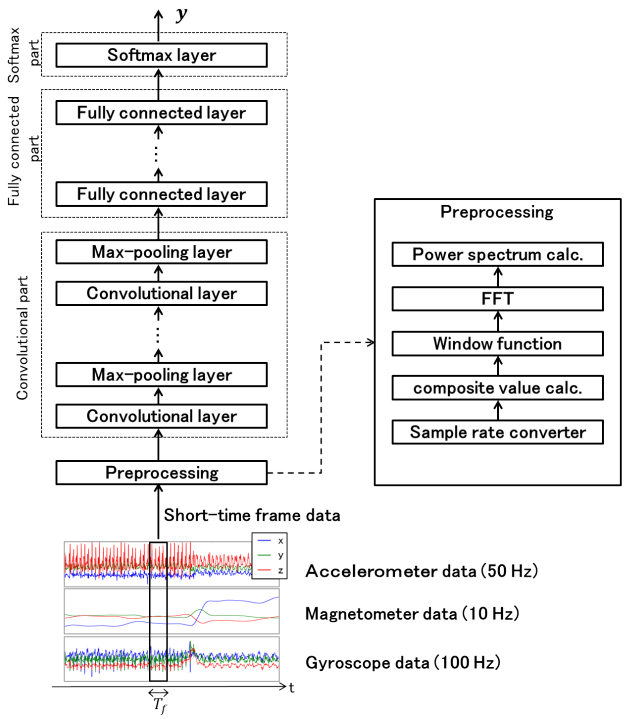


Fig. 1. Block diagram of UI model.

A. Pre-processing

Short-time Fourier transform (STFT) for time-series sensor data is calculated in pre-processing and its power spectrum is used as an input to a CNN.

Input of preprocessing is a short-time frame data of sensors with a frame period T_f, and a shift period $T_f/2$. If the sampling frequency of each sensor is different, the number of sampled signals in T_f is also different. In such a case, we converted sampling frequencies of each sensor to a certain sampling frequency to align the number of samples N_f.

An accelerometer, a magnetometer and a gyroscope, all output three dimensioned values. Let $a(i) = (a_i^{(x)}, a_i^{(y)}, a_i^{(z)})$ $(i = 1, \cdots, N_f)$ be an output of a sample rate converter, where i is a discrete-time variable. A composite of acceleration in three axes in each i: $\sqrt{(a_i^{(x)})^2 + (a_i^{(y)})^2 + (a_i^{(z)})^2}$ is calculated to reduce an influence from smartphone's postures. Then a power spectrum is calculated from a windowing function and Fourier Transform. The output is $\mathbf{x} = x_{i,c}$ $(i = 1, \cdots, N_p; c = 1, \cdots, N_s)$, where $N_p = (N_f/2 + 1)$ is the number of power spectrum bins, N_s is the number of sensors. The preprocessing is done for each frame segment.

B. Convolutional Neural Network for Sensor Data

A CNN for sensor data is composed from three parts: a convolutional part, a FC part and a softmax part as shown in Fig. 1. An input of the first convolutional part is $\mathbf{x}$. The CNN gives probabilities for each activity class.

The convolutional part is composed of convolutional and pooling layers, which are able to be stacked several times. Let

$u^{(l-1)} = \left[[u_{1,1}^{(l-1)}, \cdots, u_{N,1}^{(l-1)}], \cdots, [u_{1,C}^{(l-1)}, \cdots, u_{N,C}^{(l-1)}]\right]$ be an input of the l-th convolutional layer, where N s the number of samples and C is the number of channels. The output of the convolutional layer is then given by:

$$u_{i,k}^{(l)} = \phi^{(l)} \left(\sum_{c=1}^{C} \sum_{h=1}^{H} w_{h,c,k}^{(l)} u_{i+h,c}^{(l-1)} + b_{i,k}^{(l)} \right), \tag{1}$$

where H is the filter size, and $w_{h,c,k}^{(l)}$ is a weight for the k-th channel in the l-th layer, b is the bias term, ϕ is an activation function. $u_{i,c}^{(0)}$ is $x_{i,c}$ at the first layer. To reduce the sensitivity of the output to shifts and distortions, the output of convolutional layer is fed to an additional layer, called a max-pooling layer.

The next FC part placed on the top of the convolutional part consists of several FC layers. The output of the FC layer is:

$$\mathbf{u}^{(l)} = \phi^{(l)} \left({\mathbf{w}^{(l)}}^{\mathrm{T}} \mathbf{u}^{(l-1)} + \mathbf{b}^{(l)} \right), \tag{2}$$

where $\mathbf{w}^{(l)}$ is a weight matrix, $\mathbf{b}^{(l)}$ is a bias vector, and $\mathbf{u}^{(l-1)}$ is an input vector for the l-th FC layer.

The topmost softmax layer outputs the score of an activity class s given:

$$P(s|\mathbf{x};\theta) = \frac{\exp(u_{s_t}^{(L)})}{\sum_{j=1}^{K} \exp(u_j^{(L)})}, \tag{3}$$

where $\theta = w^{(1)}, b^{(1)}, \cdots, w^{(L)}, b^{(L)}$ is a set of UI model parameters. UI model parameters are learned by using a large amount of data $\{(\mathbf{x}_t, s_t)\}_{t=1}^{T}$, where t is an index of frame segmentation, $\mathbf{x}_t$ is expressed as $x_{i,c}$ in each frame segmentation and, T is a total number of frames.

The parameter θ is trained to minimize the following cost function with Adam [14], which is one of the stochastic gradient descent algorithms. The cost function $E(\theta)$ is expressed by taking the logarithm of the posterior distribution and inverting the sign:

$$E(\theta) = -\sum_{t}^{T} \log P(s_t|\mathbf{x}_t;\theta). \tag{4}$$

We used the dropout method [15] that involves choosing a probability p (commonly p = 0.5), and randomly deactivating hidden nodes with probability p during training time.

V. USER-DEPENDENT CNN

In this session, we introduce a user adaptation method for HAR using LHUC (Fig. 2). UD hidden unit layer is added on the top of each FC layer. The parameters of each UD hidden unit layers are estimated for each user. These parameters are learned by using a small amount of adaptation data $\{(\mathbf{x}_{t,m}, s_{t,m})\}_{t=1}^{T_m}, T_m \ll T$ for user m in order to refine the model such that it better approximates the posterior distribution for a given user. The other parameters are fixed in the training phase of UD model.

Let us focus on the l-th FC layer. Its corresponding UD hidden unit layer has a new model parameter $r_m^{(l)} = [r_{1,m}^{(l)}, \cdots, r_{J,m}^{(l)}]$, where J is the number of units and m represents an user learned and Eq. (2) for UI model is replaced by:

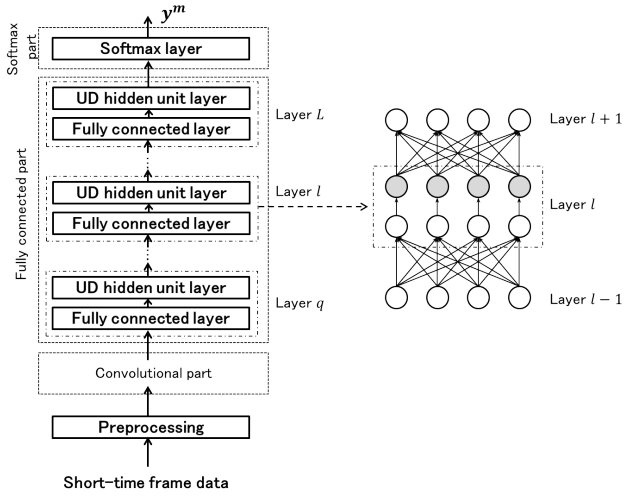


Fig. 2. Block diagram of UD model.

$$\mathbf{u}^{(l)} = \alpha(\mathbf{r}_m^{(l)}) \circ \phi^{(l)} \left({\mathbf{w}^{(l)}}^{\mathrm{T}} \mathbf{u}^{(l-1)} + \mathbf{b}^{(l)} \right), \tag{5}$$

where α is an activity function and "$\circ$" is an element-wise multiplication. Each $\alpha(\mathbf{r}_m^{(l)})$ operates as a weight of a corresponding node in FC layer. Here we use sigmoid function with amplitude 2:

$$\alpha(c) = \frac{2}{1 + \exp(-c)}. \tag{6}$$

If the value of $\alpha(\mathbf{r}_m^{(l)})$ is 1.0, the UI model and the UD model are equivalent. Therefore, we set an initial value of $\mathbf{r}_m^{(l)}$ as a vector with all values 1.0.

VI. EXPERIMENTS

We used the dataset explained in Chapter 2. For each user, the data from the other eight users are used for training the UI model. One fourth of the evaluation users data was used for adaptation (Adaptation data) and the three fourths were used for testing (Test data). Adaptation data is about 1 hour and Test data is about 3 hours.

The sampling frequencies of recorded data are 50 Hz in accelerometer, 10 Hz in magnetometer and 100 Hz in gyroscope, and are adjusted to 100 Hz by converting sample rate. The sliding window is 240 samples, and its step size is 120 samples. FFT size is 256pt. The structure of UI model is shown in Table II. It has 3 CNN layers and 3 FC layers. Therefore, HD hidden unit layers are added on each 3 FC layer of the UI model. As an activation function, we used Exponential Linear Units (ELU) [16]. During training, Adam was used to the target regression value and the learning rate has an initial value of 0.2 and attenuates by 1/2 every 50 epochs. Mini-batch size was 100 and training epochs was 500. All of the models described in this paper were implemented using the Chainer toolkit [17].

TABLE II
PARAMETERS FOR UI MODEL

Attribute	Channel	Filter	Stride	Unit	Function
Input	3	-	-	-	-
conv1	32	15×1	1	-	ELU
pool1	32	2×1	1	-	-
conv2	64	10×1	1	-	ELU
pool2	64	2×1	1	-	-
conv3	128	4×1	1	-	ELU
pool3	128	2×1	1	-	-
fc4	-	-	-	384	ELU
fc5	-	-	-	400	ELU
fc6	-	-	-	7	ELU

VII. RESULT

Recognition results are shown in Table III. The average recognition rate with the UI model is 85.1%, but that of user ID3 is considerably worse than that of the other users, only 74.9%. The behavior of User ID3 is different from behavior of other users. By applying UD model, the result of User ID3 was greatly improved by 13.6%. The average recognition rate with UD model was 88.2%, improved by 3.1%. No users had a degradation in recognition performance. We also evaluated another method which re-trains the UI model with Adaptation data (Re-train UI). In this case, the UI model parameters are set as initial values and are trained again by using Adaptation data. As in Table III, its recognition rate was worse than the UI model. It became unstable because all parameters were re-trained with a small amount of data. Although there were many users who have improved recognition performance compared to UI model there were also users with degradation.

Fig. 3 shows how UD hidden units are trained at the fc4 layer. We compare User ID3 whose performance is greatly improved by adaptive learning (upper figures) and User ID4 whose performance did not improve so much (lower figures). The figure on the left ((a), (c)) is the output value of UD hidden units added to the fc4 layer after learning. The vertical axis is the output value of UD hidden unit [0-2]. The horizontal axis is the HD hidden unit ID in fc5 layer. ((b), (d)) are their histograms. As the output value deviates from 1.0, the UD model adapts to be different from the UI model, which means the UD model is adapted to evaluation user. In Fig. 3 (b), the outputs of user ID3 are more deviated from 1.0. Therefore the shape of user ID3's histogram (b) becomes wider than that of user ID4's histogram (d). Fig. 4 also shows how HD hidden units are trained in the fc5 layer. The tendency of HD hidden units training is the same as the fc4 layer.

Furthermore, we investigated how the amount of Adaptation data affects the recognition rate. Fig. 5 shows average recognition rate with different amounts of Adaptation data. Even when using a small amount of Adaptation data, improves the recognition rate for many users.

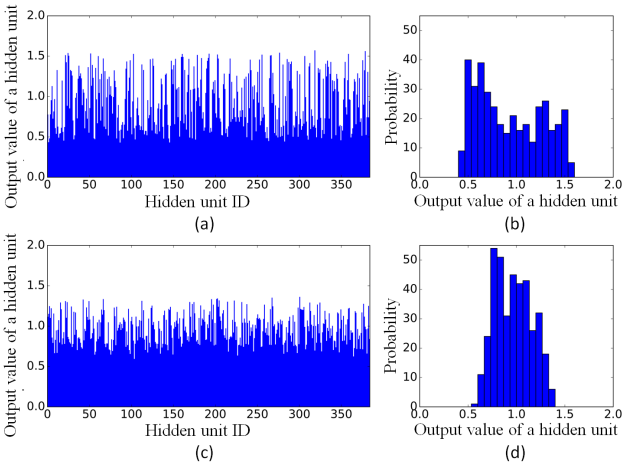


Fig. 3. (a), (c) are the output values of hidden units added to the fc4 layer. The vertical axis is the output value of hidden unit [0-2]. The horizontal axis is the hidden unit ID in fc5 layer. (b), (d) are their histograms. Upper is user ID3 and lower is user ID4.

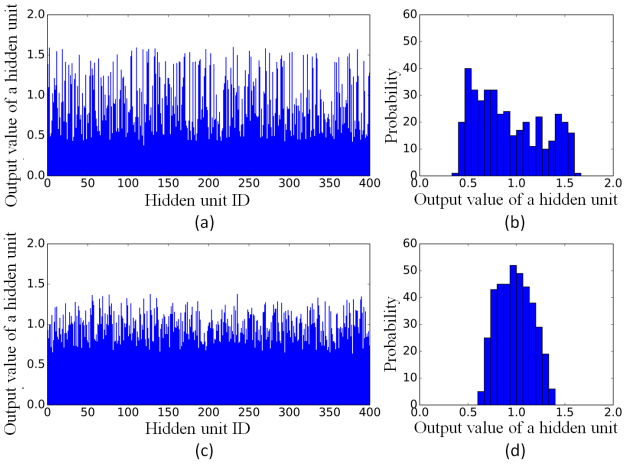


Fig. 4. (a), (c) are the output values of hidden units added to the fc5 layer. (b), (d) are its histogram. Upper is user ID3 and lower is user ID4.

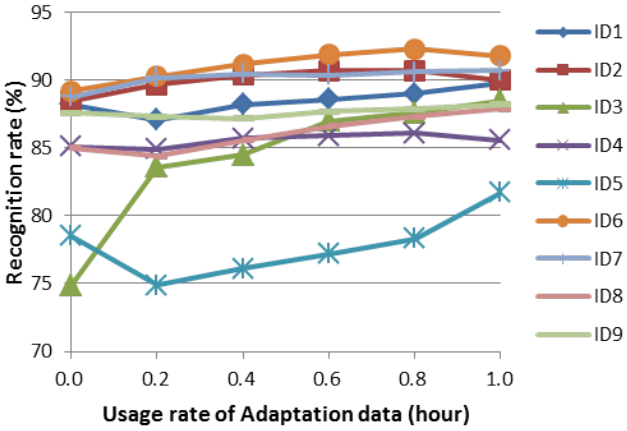


Fig. 5. Recognition rate with different amounts of Adaptation data.

TABLE III

RECOGNITION RATE (%) OF HAR FOR EACH USER AND ITS AVERAGE. THE TOP IS UI MODEL (BASE LINE). THE BOTTOM IS UD MODEL (PROPOSED). THE MIDDLE IS AN RE-TRAIN UI MODEL WHICH IS BASICALLY USED IN DEEP LEARNING. UD MODEL IMPROVED RECOGNITION PERFORMANCE FOR EVERY USER COMPARE TO UI MODEL.

	Evaluation user ID									
	ID1	**ID2**	**ID3**	**ID4**	**ID5**	**ID6**	**ID7**	**ID8**	**ID9**	**Average**
UI	88.2	88.5	74.9	85.1	78.5	89.2	88.8	85.0	87.6	85.1
Re-train UI	86.7	87.1	87.1	77.5	70.2	85.0	89.3	85.1	83.1	83.5
UD (proposed)	89.8	90.0	88.5	85.6	81.7	91.8	90.7	87.9	88.2	88.2

VIII. CONCLUSION

We have proposed a user adaptation method using LHUC for CNN-based HAR. It can effectively improve its performance by using a small amount of data from users. We collected sensor data from smartphones and utilized them for evaluating it. The recognition performance was improved by 3% from the case when we used UI models before adaptation. The largest improvement among users was 13.6%. In the future, we will improve the proposed method so that it can be effective using fewer amounts of data, and develop unsupervised adaptation methods where the class labels for adaptation data are not available.

REFERENCES

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, pp. 541-551, 1989.

[2] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," *Spoken Language Technology Workshop (SLT)*, pp. 171-176, 2014.

[3] L. Bao, and S. S. Intille, "Activity recognition from user-annotated acceleration data," *In International Conference on Pervasive Computing*, pp. 1-17, Springer Berlin Heidelberg, 2004.

[4] J. R. Kwapisz, G. M. Weiss and S. A. Moore, "Activity recognition using cell phone accelerometers," *CM SigKDD Explorations Newsletter*, vol. 12, pp. 74-82, 2011.

[5] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys and Tutorials*, vol. 15, pp. 1192-1209, 2013.

[6] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," In *Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference*, pp. 197-205, 2014.

[7] J. Yang, M. N. Nguyen, P. P. San, X. Li and S. Krishnaswamy, "Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition," In *IJCAI*, pp. 3995-4001, 2015.

[8] C. A. Ronao and S. B. Cho, "Deep convolutional neural networks for human activity recognition with smartphone sensors," In *International Conference on Neural Information Processing*, pp. 46-53, Springer International Publishing, 2015.

[9] J. W. Lockhart, G. M. Weiss, J. C. Xue, S. T. Gallagher, A. B. Grosner and T. T. Pulickal, "Design considerations for the WISDM smart phone-based sensor mining architecture," In *Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data*, pp. 25-33, 2011.

[10] D. Anguita, A. Ghio, L. Oneto, X. Parra and J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition using Smartphones," In *ESANN*, 2013.

[11] C. H. Lee, and J. L. Gauvain, "Speaker adaptation based on MAP estimation of HMM parameters," In *Acoustics, Speech, and Signal Processing. , ICASSP-93.*, Vol. 2, pp. 558-561, 1993.

[12] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech & Language*, Vol. 9, pp. 171-185, 1995.

[13] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[14] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, Vol. 15, pp. 1929-1958, 2014.

[15] D. A. Clevert, T. Unterthiner and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," arXiv preprint arXiv:1511.07289, 2015.

[16] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning,"

[17] Hinton, L Deng, D Yu, GE Dahl, A Mohamed, N Jaitly, A Senior, V Vanhoucke, P Nguyen, TN Sainath, and B Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,," *Signal Processing Magazine*, vol. 29, no. 6, pp. 8297, 2012.