Prototype Similarity Learning for Activity Recognition

Lei Bai[1(✉)], Lina Yao[1], Xianzhi Wang[2], Salil S. Kanhere[1], and Yang Xiao[3]

[1] University of New South Wales, Sydney, Australia
baisanshi@gmail.com, {lina.yao,salil.kanhere}@unsw.edu.au
[2] University of Technology Sydney, Sydney, Australia
xianzhi.wang@uts.edu.au
[3] Xidian University, Xi'an, China
yxiao_052126@stu.xidian.edu.cn

Abstract. Human Activity Recognition (HAR) plays an irreplaceable role in various applications such as security, gaming, and assisted living. Recent studies introduce deep learning to mitigate the manual feature extraction (i.e., data representation) efforts and achieve high accuracy. However, there are still challenges in learning accurate representations for sensory data due to the weakness of representation modules and the subject variances. We propose a scheme called Distance-based HAR from Ensembled spatial-temporal Representations (DHARER) to address above challenges. The idea behind DHARER is straightforward—the same activities should have similar representations. We first learn representations of the input sensory segments and latent prototype representations of each class, using a Convolution Neural Network (CNN)-based dual-stream representation module; then the learned representations are projected to activity types by measuring their similarity to the learned prototypes. We have conducted extensive experiments under a strict subject-independent setting on three large-scale datasets to evaluate the proposed scheme, and our experimental results demonstrate superior performance of DHARER to several state-of-the-art methods.

Keywords: Activity recognition · Deep learning · Similarity comparison · Spatial-temporal correlations

1 Introduction

Human activity recognition (HAR) is a significant step towards human computer interaction and enables a series of promising applications such as assistant living, skills training, health monitoring, and robotics [6]. Existing HAR techniques are either video- or sensor-based. In particular, sensor-based HAR aims at inferring human activities from a set of sensors (e.g., accelerometer, gyroscope, and magnetometer), which generate data streams over time. This approach is generally known to have several advantages over video-based HAR including: ease of deployment, low cost and less invasive from a privacy perspective [7].

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12084, pp. 649–661, 2020.
https://doi.org/10.1007/978-3-030-47426-3_50

Previous studies on sensor-based HAR focus on designing powerful handcrafted features in time (e.g., mean, variance) and frequency domain (e.g., power spectral density) to represent segments of raw sensory streams [9]. Traditional machine learning models such as Support Vector Machine (SVM) and Random Forest are employed to project the feature vector to activity labels [2]. The performance of these methods normally depends on the effectiveness of the extracted features where are heuristic, task-independent, and not specially designed for HAR [12]. Since designing powerful task-specific features require significant domain knowledge, and are labour intensive and time consuming, recent research introduces deep learning methods, which have exceptional data representation ability to expedite feature extraction. These works utilize deep neural networks, such as Convolution Neural Networks (CNN) [5,16] and Long-Short Term Memory (LSTM) [8,11], as feature extractors to learn the representation of the input sensory segments automatically, and then map the representation to labels using another neural network (normally a basic fully-connected layer).

Although deep learning methods have achieved significant progress, it is still difficult to learn accurate representations for the input segments due to the complex spatial correlations among sensors and temporal correlations between time periods. Considering the sensitivity of neural networks to noise, the biases in the representations further prevent neural network-based classifiers from making correct activity classification. In addition, subject variances inherently exist in HAR, where people tend to perform activities that are heavily influenced by personal characteristics, such as gender, height, weight, and strength. For example, men usually perform activities at a larger magnitude than women. Such divergence introduces deviations to the representations among subjects and thus prevent the model from getting accurate classification for new subjects (haven't appeared in the training set).

We propose to solve this problem from three perspectives: 1) Representation Stage: It is necessary to jointly capture the spatial and temporal correlations to achieve more accurate feature extraction. 2) Classification Stage: Intuitively, representations of the same activities should be similar. Therefore, using a distance metric which can infer the type of an input segment from labels of the most similar prototype is likely to make the classification module less susceptible to the preciseness of the data representations (compared to neural network based classification). 3) Training Stage: the subject variance can be explicitly modeled and minimized in the training stage to enhance the generalization ability of the approach.

The main contributions of this work are as follows:

- We propose a novel end-to-end deep learning framework for HAR to deal with the bias and deviations in the representations due to inaccurate learning and subject-variances.
- We design a dual-stream CNN network to jointly capture the spatial and temporal correlations in the multivariate sensory data, which can achieve more accurate representation and decrease the bias.

- We introduce a distance-based classification module to classify the segments by comparing their similarity to the learned prototypes of each class in the representation space, which is less susceptible to representation bias. We also introduce a cross-subject training strategy to train the module for minimizing the deviation caused by subject-variance.
- We conduct extensive experiments on three large-scale datasets under a strict subject-independent setting and demonstrate the superior performance of our model in new subjects. Our method consistently outperforms state-of-the-art methods by at least 3%.

2 Related Works

The recent work in HAR has moved towards designing deep learning models for more accurate recognition, given the exceptional representation ability of deep learning techniques. Most deep learning-based HAR methods focus on capturing the temporal correlations in the sensory streams. Jian Bo et al. [16] tackle the problem with convolutional neural networks, in which the convolution and pooling filters are designed along the temporal dimensions to process the readings of all sensors. Their work can capture long-term temporal correlation by stacking multiple CNN layers. Ordóñez et al. [12] further extend this model to DeepConvLSTM by integrating LSTM after CNN layers. The proposed DeepConvLSTM framework contains four CNN layers and two LSTM layers to capture the short-term and long-term temporal correlations, separately. One drawback of the DeepConvLSTM is that it potentially assumes the signals in all time steps are relevant and contribute equally to the target activity, which may not true. Murahari et al. [11] propose to solve the problem by integrating the temporal attention module to DeepConvLSTM. The attention module aligns the output vector at the last time step with other vectors at earlier steps to learn a relative importance score for each previous time step. Different from these methods, Guan et al. [8] propose to achieve more robust data representation ability with the ensemble method. They employ the Epoch-wise Bagging scheme in the training procedure and select multiple LSTMs in different training epochs as basic learners to form a powerful model. However, these methods neglect the spatial correlations among the different sensors, which cannot represent the sensory data precisely. Besides, they directly classify the learned representations to activity type with basic NN-based classifier, which could lead to misguided result due to the learning deviation and subject variances in the representations.

3 Problem Definition

The typical scenario for sensor-based HAR involves multiple devices attached to different parts of the human body. Each device carries multiples sensors, e.g., an inertial measurement unit (IMU) typically contains nine sensors: 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer. In this work,

we consider each 3-axis device as three sensors for capturing spatial correlations, e.g., 3-axis accelerometer contains x-accelerometer, y-accelerometer, and z-accelerometer. Thus, an IMU with 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer contains nine sensors. Let M be the total number of sensors embedded in multiple body-worn devices, and s_i $(1 \leq i \leq M)$ be the reading from the i_{th} sensor. Then, at each time point, the sensors, together, generate a vector of readings: $\boldsymbol{s} = [s_1, s_2, ..., s_M]^T$. Thus, a segment with the sliding window size $\boldsymbol{T}$ can be represented by $\boldsymbol{Seg} = [\boldsymbol{s_1}, \boldsymbol{s_2}, ..., \boldsymbol{s_T}]$.

Let there be N potential activities to be recognized, $\mathcal{C} = \{c_1, c_2, ..., c_N\}$, HAR aims to learn a function, $\mathcal{F}(\boldsymbol{Seg}, \bullet)$, to infer the correct activity label for the given segment, where $\bullet$ represents all learnable parameters.

4 Methodology

In this section, we elaborate our proposed methods for more accurate HAR, which contains three components: a dual-stream representation module to learn more accurate representations of the input segment, a distance-based classification module to recognize human activities, and a cross-subject training strategy to minimizing the subject-divergence.

4.1 Dual-Stream Representation Module

We first introduce the CNN-based dual-stream representation module (DARM) (shown in Fig. 1), which contains a spatial CNN network and a temporal CNN network. The two CNN networks learn two sub-representations capturing the spatial correlations and temporal correlations within the input segment, respectively, which can be regarded as an image of $M \times T$ (as denoted in Sect. 3). Then the two sub-representations are merged by summing to get the final joint representation of the input segment. Compared to the previous data representation models, the dual-stream representation module is more accurate by encapsulating both spatial and temporal correlations jointly. Besides, it is more light-weight and easy-to-train compared to LSTM-based approaches [8,11,12].

As shown in Fig. 1, the overall architectures of the temporal CNN and spatial CNN are the same. Both of them contain three consecutive CNN blocks to extract prominent patterns in the segment from different perspectives. The difference between the temporal CNN and spatial CNN mainly lays in the size of CNN kernels. More specifically, the temporal CNN applies the CNN kernels with size $1 \times k_T^l$ in the l_{th} T-CNN block, which operate the data along the time axis to capture the temporal correlations between different time points. As a contrast, the spatial CNN applies the CNN kernels with size $k_S^l \times k_S^l$ in the l_{th} S-CNN block to capture the spatial correlations between different sensor series. Besides the kernel size, either of the T-CNN block and the S-CNN block comprises a convolutional layer with a rectified linear units (ReLu) activation function, a max_pooling layer, and a batch-normalization layer. The convolutional layer performs the main function of pattern extraction, which employs several kernels

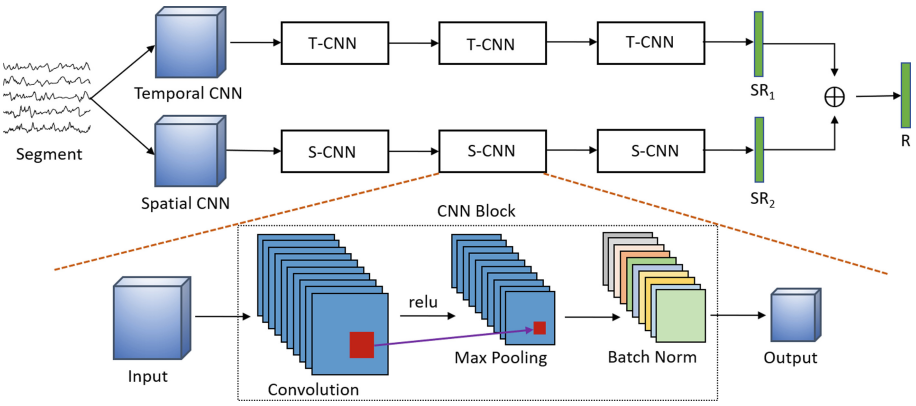

Fig. 1. The proposed dual-stream data representation module based on CNN networks

of the same shape to filter the input data X and extract meaningful patterns. We calculate a convolution layer with the ReLu activation function as follows:

$$X_j^{l+1} = \sigma(\sum_{i=1}^{i=F^l} W_{i,j} \times X_i^l + b_j^l) \tag{1}$$

where X_i^l is the i_{th} channel of the input for the l_{th} convolutional layer, F^l is the feature map (channel) numbers, $W_{i,j}$ is the j_{th} kernel, b_j^l is the bias and $\sigma(\cdot)$ is the ReLu function defined as: $\sigma(X^{l+1}) = max(0, X^{l+1})$. Then, the max pooling layer is employed as the sampling method to down-sampling the extracted representations while keeping the most protrusive patterns. We further integrate the batch-normalization layer to the CNN block to achieve faster and more stable training. The batch-normalization layer normalizes the layer's input with batch mean and batch variance to force the input of every layer to have approximately the same distribution [10].

4.2 Distance-Based Classification Module

Based on the representation module, we then propose to learn to recognition human activities by distance based classification module (DCAM) (Fig. 2), which is based on the Prototypical Networks [14]. Different from the general HAR process, which first learns a representation for the input segment and then maps the representation to the corresponding activity with classifiers, DCAM first learns a representation for the input segment and a latent prototype representation (a vector) for each class together. The prototypes are used to represent the embedding of each class. Then, DCAM recognizes the segment representation by comparing its similarity with the prototypes, which follows the same idea with the nearest neighbour methods. For clarity, we denote the data used to learning the prototypes as the support set and the segments to be recognized as queries

(see Fig. 2). In the training period, both support set and queries come from the training dataset. In the testing phase, we extract the support set from the training dataset and the queries from the testing dataset to avoid the information leakage.

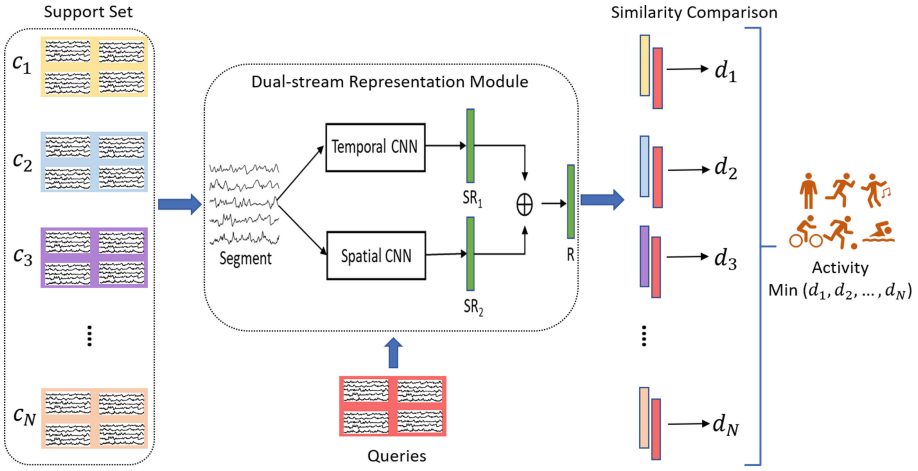

Fig. 2. The proposed distanced-based classification module

To learn the prototypes, we randomly select N_s samples from each class to form the support set for a batch of queries. Then, these support samples are fed into our dual-stream representation module to get their representations. The prototype of each class is the mean vector of the learned representations in the support set belonging to the corresponding class. Take $f(\cdot)$ denote the transformation of representation module, X_i^j as the i_{th} support sample in the j_{th} class, then the prototype of class j can be calculated as:

$$C_j = \frac{1}{N_s} \sum_{i=1}^{N_s} f(X_i^j) \tag{2}$$

Similarly, the query instances are also mapped to the embedding space by our representation module. DCAM can then learn a distribution of a query x over classes based on the softmax of its distances to the learned prototypes $\{C_1, C_2, \ldots, C_N\}$ in the representation space [14]:

$$p_f(y = c_j|x) = \frac{exp(-d(f(x), C_j)}{\sum_{j'=1}^{N} exp(-d(f(x), C_{j'}))} \tag{3}$$

where $d(\cdot, \cdot)$ is a distance function to measure the similarity of two given vectors. There are multiple widely used choices for calculating distance in the literature, such as the Cosine distance, Mahalanobis distance, Euclidean distance and so on.

In this work, we employ the squared Euclidean distance as the distance function as it is proved to be more effective than others in [14].

The whole model can be easily trained in an end-to-end manner by minimizing the negative log-probability $-log(p_f(y = c_j|x))$ according to the true label c_j of the query segment x via back-propagation strategy. Thus, We define the loss function as follows:

$$\mathcal{L}_x = d(f(x), C_j) + log(\sum_{j'=1}^{N} exp(-d(f(x), C_{j'}))) \tag{4}$$

4.3 Cross-Subject Training

We further propose the cross-subject training strategy to alleviate the influence of subject variances to the representations. Instead of random sampling support samples and queries from the training set, our cross-subject training strategy intentionally select queries from one subject and support set from other subjects for each batch during the training process. Thus, we can decrease the divergence between different subjects in the representation space through training iteration by minimizing the distance between queries representations and prototypes, which are learned from different subjects separately. Besides, the cross-subject training strategy also harmonizes the training stage and testing stage under the subject-independent setting, where the support set from the training dataset and queries from the testing dataset come from different subject inherently. Algorithm 1 describes the method's overall training procedure.

Algorithm 1. Training and Optimization

Require: the training dataset $\boldsymbol{L} = \{(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{U})\}$ ($\boldsymbol{U}$ is the subjects set in training), number of samples in queries N_q, number of samples in the support set for each class N_s, maximum training iteration $Iter$.

```
random initialize the network parameters
for iter = 0; iter < Iter do
    randomly choose query subjects u_i from U
    load N_q query samples from subject u_i as Q
    load N_s support samples for each class from U − u_i as support set S
    calculate representations of the queries and support samples with DARM
    for c_i in {c_1, c_2, ..., c_n} do
        calculate prototype C_i of class c_i according to Equation 2 with represented S
    end for
    Init loss L = 0
    for x, y in represented Q do
        calculate loss L_x with Equation 4
        update loss with L = L + L_x
    end for
    Back-propagate L and update the network parameters
end for
```

Table 1. Statistics of datasets (# denotes the "number").

Dataset	Subject#	Activity#	Frequency	Window	Devices#	Sensors#	Sample#
MHEALTH	10	12	50 Hz	20 (0.4 s)	3	23	34 097
PAMAP2	8	12	100 Hz	20 (0.2 s)	3	36	191 309
UCIDSADS	8	19	25 Hz	20 (0.8 s)	5	45	113 848

5 Experiments

5.1 Datasets

While several datasets are publicly available for HAR, many of them are limited in the scale of subjects (e.g. the Skoda dataset [15] only has one subject) or activities (e.g. the UCI dataset [1] only contains six activities). To evaluate the performance of our method in classifying activities and dealing with subject divergence more comprehensively, we select the following three datasets with relatively more activities and subjects:

MHEALTH Dataset. This dataset [3] contains body motion and vital signs for ten volunteers of diverse profiles. Each subject performed 12 activities in an out-of-lab environment with no constraints.

PAMAP2 Dataset. The PAMAP2 dataset [13] was designed to benchmark daily physical activities. It contains data collected from nine subjects related to 18 daily activities such as vacuum cleaning, ironing, and rope jumping.

UCIDSADS Dataset. The UCIDSADS dataset [4] was specially designed for daily and sports activities. It comprises motion sensor data of 19 sports activities such as walking on a treadmill and exercising on a stepper. Each activity was performed by eight subjects for 5 min without constraints.

Data Pre-processing. For the MHEALTH and UCIDSADS dataset, we use all the data from all subjects for experiments. For the PAMAP2 dataset, we remove six activities (watching TV, computer work, car driving, folding laundry, house cleaning, and playing soccer) as they are only executed by one subject. As a result, 12 activities from eight subjects are kept for our experiments in PAMAP2. Only the basic data segmentation and normalization methods are applied to the dataset. More specially, we first divide the raw sensory data streams into small segments with a fixed-sized sling window and an overlap of 50% for all the three dataset. Each window contains 20 time points, resulting the window lengths for MHEALTH, PAMAP2, and UCIDSADS are 0.4 s, 0.2 s, and 0.8 s, respectively. Then, we normalize the segments with the standard normalization methods. Table 1 gives the statistics of the three datasets.

5.2 Evaluation Settings

The main parameters in our evaluation includes network parameters and training parameters. For the temporal CNN part, we use 128 kernels in all three layers

shaped $(1 \times 5) \rightarrow (1 \times 5) \rightarrow (1 \times 2)$ respectively. For the spatial CNN part, we user 128 kernels in all three layers shaped $(6 \times 5) \rightarrow (6 \times 5) \rightarrow (2 \times 2)$, $(5 \times 5) \rightarrow (5 \times 5) \rightarrow (5 \times 2)$, and $(6 \times 5) \rightarrow (7 \times 5) \rightarrow (5 \times 2)$ for MHEALTH, PAMAP2 and UCIDSADS respectively. In learning the queries representations, we set the Batch_size (N_q) to 240 to accelerate the training speed and the length of the learned segment representations is 64. For learning the prototypes, we sample five samples from each class (N_s) as the support set in each iteration. We initialize the network parameters with Xavier Normal initialization and optimize them by Adam optimizer at the learning rate of 0.0005 for all three datasets.

To thoroughly evaluate the performance of our proposed model, we assess it iteratively with LOSO protocol on every subject separately. In each experiment, we train the model from scratch and test the model with one subject's data. Finally, we will get $subject_{number}$ results for each model. Considering the space limitation, we mainly report the mean result, worst result, and best result of all subjects as $mean[worst, best]$, which reflects both the overall performance and the generalization ability of a model. Besides, the weighted Precision (P_w) and weighted F_{score} (F_w) are used as the performance metrics for comparison.

5.3 Overall Comparison

To verify the overall performance of the proposed model, we compare our method with the following baseline and SOTAs: 1) the support vector machine (SVM), 2) MC-CNN [16], 3) b-LSTM-S [9], 4) ConvLSTM [12], 5) Ensem-LSTM [8], 6) AttConvLSTM [11], 7) Multi-Agent [5]. These SOTAs vary from CNN-based, LSTM-based to CNN-LSTM hybrid model and also include ensemble and attention methods. We replicated each method with the same settings as introduced in the original papers, except for the data pre-processing steps, where we use the same window size and overlap as ours. We also evaluate them with the LOSO evaluation protocol iteratively to achieve a fair and thorough comparison.

Table 2 shows the experimental results, from which we can observe the following points: 1) all the SOTAs deep learning models perform better than SVM, showing the superior ability of deep learning models in extracting complex nonlinear temporal patterns in the sensory streams. 2) the MC-CNN model outperforms LSTM-based methods in the MHEALTH dataset and PAMAP2 dataset, but fails in the UCIDSADS dataset. Recall the window length of each dataset, we interpret the results as the admirable ability of temporal CNN in capturing accurate temporal correlations with only a short time period of data. As a contrast, LSTM-based methods need data from longer period of time. 3) the complex reinforcement learning-based Multi-agent model does not work very well as reported in [5], where only six basic activities are selected for experiments. The result indicates the difficulty of selecting important modalities for numerous and more complex activities. 4) Last but not the least, our method consistently beats all the comparison models on three datasets with a significant margin. The mean recognition F_{score} achieves 4.52%, 4.78%, and 3.17% absolute improvements over the best SOTA in the MHEALTH, PAMAP2 and UCIDSADS datasets, respectively. The comparison demonstrates the effectiveness of our proposed model.

Table 2. Overall comparison with SOTAs on three datasets. Each cell consists of the mean score of a method in one evaluation metric, followed by the corresponding minimum and maximum scores in brackets. The best performance values are in bold.

MHEALTH	Method	SVM	MC-CNN	Bi-LSTM-S	ConvLSTM
	P_w	79.53 [65.29, 94.47]	93.51 [84.11, 98.18]	87.16 [76.51, 95.41]	89.37 [81.48, 99.21]
	F_w	76.73 [59.33, 92.80]	92.17 [85.34, 97.98]	87.90 [79.01, 94.85]	89.89 [81.49, 99.22]
	Method	Ensem_LSTM	AttConvLSTM	Multi-Agent	DHARER
	P_w	84.81 [74.57, 98.59]	89.96 [78.30, 98.21]	91.87 [80.51, 98.06]	**97.05 [94.31, 99.59]**
	F_w	84.64 [70.32, 98.55]	90.75 [80.36, 98.17]	91.20 [81.12, 98.01]	**96.69 [93.55, 99.58]**
PAMAP2	Method	SVM	MC-CNN	Bi-LSTM-S	ConvLSTM
	P_w	70.77 [41.69, 88.76]	80.64 [57.65, 93.82]	71.12 [29.01, 92.21]	73.04 [36.42, 92.95]
	F_w	68.11 [36.72, 86.68]	78.05 [52.09, 93.37]	68.65 [32.34, 91.94]	72.36 [41.67, 92.65]
	Method	Ensem_LSTM	AttConvLSTM	Multi-Agent	DHARER
	P_w	73.90 [36.88, 90.93]	73.92 [50.40, 85.02]	73.35 [36.22, 89.88]	**83.32 [60.25, 94.38]**
	F_w	71.98 [42.09, 88.84]	71.83 [44.79, 86.58]	71.39 [31.70, 87.14]	**82.83 [56.09, 94.32]**
UCIDSADS	Method	SVM	MC-CNN	Bi-LSTM-S	ConvLSTM
	P_w	70.60 [63.19, 78.84]	87.18 [64.01, 95.42]	89.72 [74.29, 95.25]	89.58 [79.88, 95.27]
	F_w	67.74 [60.25, 78.33]	85.52 [66.57, 94.53]	87.73 [75.36, 93.28]	88.42 [77.95, 94.08]
	Method	Ensem_LSTM	AttConvLSTM	Multi-Agent	DHARER
	P_w	84.06 [72.65, 93.51]	88.24 [74.57, 94.78]	87.45 [79.48, 92.91]	**93.72 [89.71, 96.59]**
	F_w	81.09 [71.48, 90.19]	86.75 [74.64, 94.22]	84.26 [73.03, 90.70]	**91.59 [82.77, 96.22]**

5.4 Ablation and Case Study

We further conduct an ablation study to evaluate the performance of the basic modules in our method. Figure 3 gives the weighted F_{score} of the spatial CNN module with two-layer MLP as classifier (S-CNN), temporal CNN module with two-layer MLP as classifier(T-CNN), our dual-stream representation module with two-layer MLP as classifier (Dual-CNN), and our dual-stream representation module with distance-based classification module (DHARER) on three datasets. We can observe that the dual-CNN is better than both S-CNN and T-CNN, indicating that ensembling T-CNN and S-CNN to capture both spatial and temporal correlations is useful. Besides, our DHARER further improves the dual-CNN significantly, which demonstrates the effectiveness of our distance-based classification module and the cross-subject training strategy.

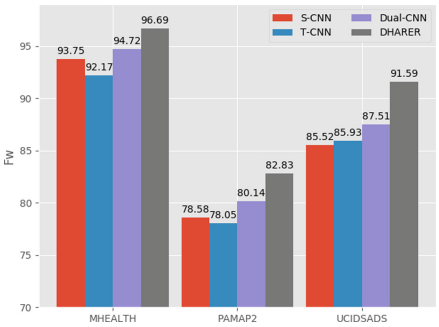

Fig. 3. Ablation study results

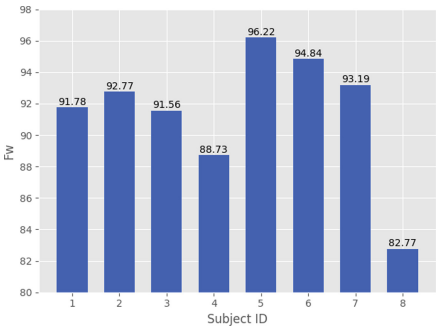

Fig. 4. Results of all subjects on UCIDSADS dataset

Considering the space limitation, we only shows the case study results on the UCIDSADS dataset in Fig. 4 and Fig. 5, which present the testing weighted F_{score} for each subject and the confusion matrix of subject 5 (achieve best performance among UCIDSADS) and subject 8 (achieve worst performance). As we can see, the results of different subjects and different activities vary seriously. Our method can achieve impressive performance on some subjects and most of the activities. But there still exist some hard-to-distinguish subjects and hard-to-distinguish activities (e.g. activity 7 which represents standing in an elevator still). In our future work, we will focus on improving the model's performance on these hard-to-distinguish subjects/activities.

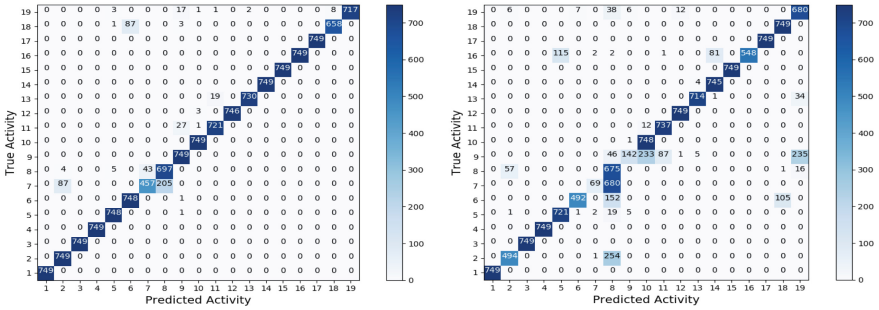

Fig. 5. Confusion Matrix of subject 5 (a) and subject 8 (b) from UCIDSADS dataset

6 Conclusion

In this work, we propose DHARER – a novel human activity recognition scheme based on similarity comparison and ensembled convolutional neural networks to

deal with the representation bias and deviation problem. We first design a dual-stream networks based on CNN to represent the sensory streams more accurately by integrating both spatial and temporal correlations. Then, a distance-based classification model is introduced, which classify the segments by comparing their similarity to the learned prototypes of each class in the representation space. Comparing to the NN-based classification module, the distance-based classification model is less susceptible to the bias in the segment representations. Moreover, we propose the cross-subject training strategy to deal with the deviations caused by subject-variance. Extensive experiments on three datasets demonstrate the superior of our proposed method over several strong SOTAs.

Acknowledgements. This research was partially supported by grant ONRG NICOPN 2909-19-1-2009.

References

1. Anguita, D., et al.: A public domain dataset for human activity recognition using smartphones. In: ESANN (2013)
2. Bai, L., et al.: Automatic device classification from network traffic streams of internet of things. In: IEEE 43rd Conference on Local Computer Networks (LCN). IEEE (2018)
3. Banos, O., et al.: mHealthDroid: a novel framework for agile development of mobile health applications. In: Pecchia, L., Chen, L.L., Nugent, C., Bravo, J. (eds.) IWAAL 2014. LNCS, vol. 8868, pp. 91–98. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13105-4_14
4. Barshan, B., Yüksek, M.C.: Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units. Comput. J. **57**(11), 1649–1667 (2014)
5. Chen, K., et al.: Multi-agent attention activity recognition. In: IJCAI (2019)
6. Chen, K., et al.: Deep learning for sensor-based human activity recognition: overview, challenges and opportunities. arXiv preprint arXiv:2001.07416 (2020)
7. Davoudi, H., Li, X.-L., Nhut, N.M., Krishnaswamy, S.P.: Activity recognition using a few label samples. In: Tseng, V.S., Ho, T.B., Zhou, Z.-H., Chen, A.L.P., Kao, H.-Y. (eds.) PAKDD 2014. LNCS (LNAI), vol. 8443, pp. 521–532. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06608-0_43
8. Guan, Y., Plötz, T.: Ensembles of deep LSTM learners for activity recognition using wearables. Proc. ACM Interact. Mob. Wearable Ubiquit. Technol. **1**(2), 11 (2017)
9. Hammerla, N.Y., et al.: Deep, convolutional, and recurrent models for human activity recognition using wearables. In: IJCAI, pp. 1533–1540. AAAI Press (2016)
10. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: ICML, pp. 448–456 (2015)
11. Murahari, V.S., Plötz, T.: On attention models for human activity recognition. In: The 2018 ACM International Symposium on Wearable Computers. ACM (2018)
12. Ordóñez, F., Roggen, D.: Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors **16**(1), 115 (2016)
13. Reiss, A., Stricker, D.: Introducing a new benchmarked dataset for activity monitoring. In: 16th International Symposium on Wearable Computers. IEEE (2012)

14. Snell, J., et al.: Prototypical networks for few-shot learning. In: NIPS. pp. 4077–4087 (2017)
15. Stiefmeier, T., et al.: Wearable activity tracking in car manufacturing. IEEE Pervasive Comput. **2**, 42–50 (2008)
16. Yang, J., et al.: Deep convolutional neural networks on multichannel time series for human activity recognition. In: IJCAI (2015)