Shallow Convolutional Neural Networks for Human Activity Recognition Using Wearable Sensors

Wenbo Huang, Lei Zhang, Wenbin Gao, Fuhong Min, and Jun He, *Member, IEEE*

***Abstract*— Due to rapid development of sensor technology, human activity recognition (HAR) using wearable inertial sensors has recently become a new research hotspot. Deep learning, especially convolutional neural network (CNN) that can automatically learn intricate activity features have gained a lot of attention in ubiquitous HAR task. Most existing CNNs process sensor input by extracting channel-wise features, and the information from each channel can be separately propagated in a hierarchical way from lower layers to higher layers. As a result, they typically overlook information exchange among channels within the same layer. In this article, we first propose a shallow CNN that considers cross-channel communication in HAR scenario, where all channels in the same layer have a comprehensive interaction to capture more discriminative features of sensor input. One channel can communicate with all other channels by graph neural network to remove redundant information accumulated among channels, which is more beneficial for deploying lightweight deep models. Extensive experiments are conducted on multiple benchmark HAR datasets, namely UCI-HAR, OPPORTUNITY, PAMAP2 and UniMib-SHAR, which indicates that the proposed method enables shallower CNNs to aggregate more useful information, and surpasses baseline deep networks and other competitive methods. The inference speed is evaluated via deploying the HAR systems on an embedded system.**

***Index Terms*— Convolutional neural networks (CNNs), cross-channel communication, deep learning, human activity recognition (HAR), sensor.**

I. Introduction

With rapid development of the Internet-of-Things and sensor technology, human activity recognition (HAR) using wearable inertial sensors has become a new research hotspot due to its extensive use in a large variety of application domains such as health-care [1], sports tracking [2], [3], fitness, game console design [4] and smart homes [5]. Deep learning [6], [7] has gained a lot of attention in sensor-based HAR scenario. Especially, convolutional neural networks (CNNs) have started delivering their advantages over feature learning and achieved state-of-the-art performance for HAR [8], [9]. Traditionally, various methods from the field of signal processing [10], [11] have been widely leveraged to distill collected sensor data, which requires domain-specific expert knowledge to process raw data. Statistical and machine learning models are then trained on the version of processed data [12]. That is to say, feature engineering is required to fit a model, which is expensive and not scalable. CNNs are capable of performing automatic feature learning, which significantly surpasses models fit on hand-crafted domain-specific features. Ideally, CNNs with automatic feature extraction provide the ability to learn features from raw sensor data with little pre-processing involved in feature engineering.

However, most existing CNNs for HAR typically overlook information exchange [13], [14] among channels within the same layer. When recognizing one human activity, the information between different channels at the same layer will not be exchanged. As far as we know, CNNs are composed of neurons that have a set of learnable weights and biases (i.e., *filter*). Based on these weights and biases, each neuron receives sensor input, performs a dot product, which is optionally followed by a non-linearity activation. Regarding each channel in CNN as a single neuron, the neurons at each layer typically respond to sensor input independently, which do not share any connections. Most existing CNNs process sensor input by extracting channel-wise features, and the information from each channel can be separately propagated in a hierarchical way from lower layers to higher layers. As a result, there is lots of redundant information accumulated between channels for the same layer, which leads an inefficient deep learning for HAR.

In this article, we, for the first time, consider cross-channel communication (C3) for CNN-based HAR. During training stage, the information at the same layer can be fully exchanged across different channels. That is to say, our method encourages all channels at the same layer to have a comprehensive interaction in order to capture more discriminative features of sensor input. When communicating with each other, the feature responses of all channels could be calibrated more explicitly to remove accumulated redundant information, which then are passed to next layers. The sketch map of C3 block in HAR is presented in Fig. 1, which consists of three parts. The first part is used for feature encoding. This module can encode feature responses from each channel via flattening them through MLP

Manuscript received March 16, 2021; revised May 27, 2021; accepted June 14, 2021. Date of publication June 24, 2021; date of current version July 9, 2021. This work was supported in part by the National Science Foundation of China under Grant 61203237, in part by the Industry-Academia Cooperation Innovation Fund Projection of Jiangsu Province under Grant BY2016001-02, and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20191371. The Associate Editor coordinating the review process was David Ayllon. *(Corresponding author: Lei Zhang.)*

Wenbo Huang, Lei Zhang, Wenbin Gao, and Fuhong Min are with the School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing 210023, China (e-mail: leizhang@njnu.edu.cn).

Jun He is with the School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China.

Digital Object Identifier 10.1109/TIM.2021.3091990

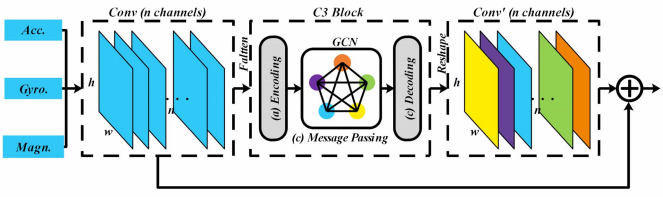


Fig. 1. Overview of C3 block.

with two fully connected (FC) layers. The second part is used for message passing by graph neural network which guarantees channels to interact with each other. Each channel's feature maps will then be updated. The third module is in charge of feature decoding, which reshapes the flattened features to the same size as original input. The decoding module then uses standard convolution operation and passes information to the next layers.

We conduct extensive experiments on multiple benchmark HAR datasets, namely UCI-HAR, OPPORTUNITY, UniMib-SHAR, and PAMAP2, which are publicly available. The effect of C3 block is evaluated by the baseline CNN with 3 or 6 convolutional layers. The experimental results show that the proposed method allows shallower CNNs to aggregate more useful information, which significantly surpasses baseline deep networks and other competitive methods. The inference speed is evaluated via deploying the HAR systems on a Raspberry Pi 3 B plus system. In comparison with previous CNN approaches used for HAR that overlook cross-channel interaction, the contributions of our work are threefold.

1) We propose a novel shallow CNN with C3 block for sensor-based HAR, which encourages all channels at the same layer to have a comprehensive interaction in order to capture more discriminative feature representation for raw sensor input.

2) When inserting the C3 block to shallower CNNs, we may obtain even better performance to baseline deep networks at much smaller memory and computational overhead, which shows that the learned features are more diverse and discriminative.

3) The effect of C3 block is evaluated by extensive ablation studies. Regardless of classification performance, we consider actual running time in a Raspberry Pi 3 B plus embedded system, which further verifies the efficiency of the proposed C3 method.

The rest of this article is organized as follows. In Section II, we will introduce recent related works. Section III will describe the details of C3 block and introduce three parts within it. In Section IV, we show our experimental results and conduct ablation experiments for further deep study. In Section V, we draw a conclusion.

II. Related Works

The idea of cross-channel interaction has obvious advantages compared with some recently proposed literature works. We review a few most related works and discuss their drawbacks from three aspects.

Traditional filters are usually handcrafted, which is hard to extract intricate features in complex HAR tasks, while the innovation of CNNs is the ability to automatically learn a large number of filters in parallel. During recent years, deep learning has become a dominant technique in HAR researches since it can automatically learn intricate activity features. For example, Wang *et al.* [15] applied a soft attention on CNN to locate and recognize weakly labeled sensor signals. Ordóñez and Roggen [13] presented a deep learning method called as DeepConvLSTM, which combines long short-term memory (LSTM) units with CNN to improve classification performance in HAR. Zeng *et al.* [16] and Ma *et al.* [17] adopted attention to focus on these channels which have more contribution to activity recognition. Zeng *et al.* [18] proposed a new CNN composed of convolutional layers and pooling layers, where each axis of three axial acceleration signals can be seen as one channel. Jiang and Yin [14] used a 2-D ConvNet to classify 2-D images which were converted from raw sensor data. However, unlike imagery data, raw sensor signals not only have correlation across temporal dimension but also have connection among different sensor modalities. Although there has been a significant amount of works on CNNs in HAR scenario, most existing networks typically overlook information exchange among channels within the same layer.

Deep models often require lots of computing resources, which is not available for wearable devices. In addition, the models are often trained OFF-line which cannot be executed in real-time. However, less complex models such as shallow networks and conventional machine learning methods could not achieve good performance. Therefore, it is necessary to develop lightweight deep models to perform HAR. In computer vision field, many research studies have been devoted to reducing model complexity. For example, Liu *et al.* [19] slimmed the network structure during training stage and Han *et al.* [20] proposed filter pruning technique to compress network. CNNs apply a set of filters on input to create output feature maps, which are tensors with a shape: feature map height$\times$feature map width$\times$feature map channels. In essence, the channel number is equal to the number of filters (i.e., *neurons*). He *et al.* [21] proposed a channel selection based least absolute shrinkage and selection operator (LASSO) regression, which can accelerate CNNs with the effect of least square reconstruction. At run-time, Gao *et al.* [22] used feature boosting and suppression (FBS) to skip unimportant channels. Jeong and Shin [23] replaced normal convolution with channel-selective convolution to reform existing CNNs. These operations can enable shallow networks to achieve an excellent performance comparable to deeper networks. To our knowledge, lightweight models are more suitable for wearable HAR computing. The research studies of reducing model complexity are rare to be seen in ubiquitous HAR scenario. How to design shallow CNNs that have better feature representation capacity deserves deeper investigations.

In computer vision field, a non-local network (NLN) used by Wang *et al.* [24] can easily model long-range spatial-temporal location's dependency. Unfortunately, NLN primarily works for video data because it captures long-range interactions

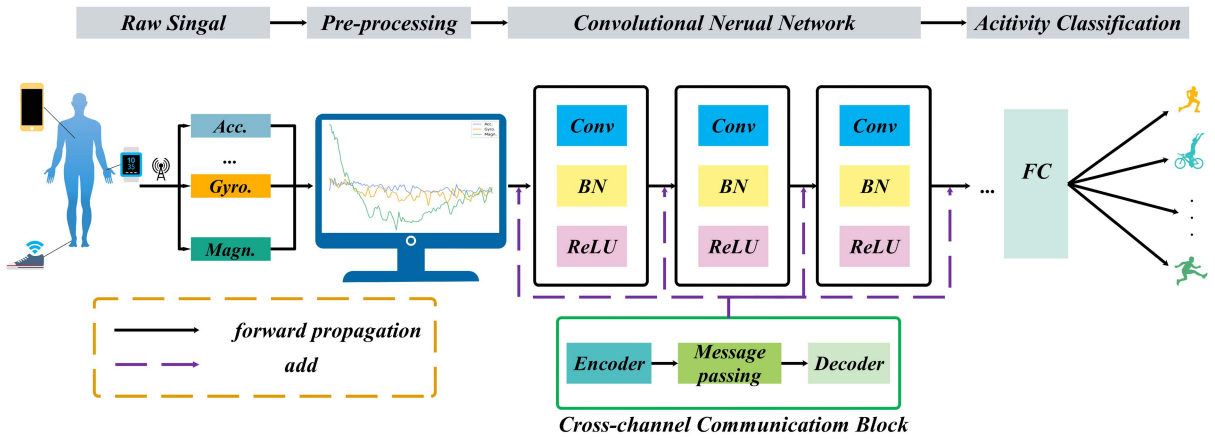


Fig. 2. Sketch map of the network with cross-channel communication (C3) block. The curves represent sensor time series. Sliding window is used to generate sensor examples.

via computing correlations between any two locations. To be specific, in video data, it needs to consider long-range interactions between distant pixels in space as well as time. Thus, it is not suitable for HAR works. A model which can establish the interaction between channels is very important. In another research line, Hu *et al.* [25] proposed a squeeze-and-excitation (SE) network, which can calibrate channel feature responses. Chen *et al.* [26] and Dai *et al.* [27] used channel-wise attention for semantic segmentation and image captioning. Wu and He [28] used group normalization. This model can be seen as a special model equipped with channel-wise communication. However, their interactions across different channels are too simple, where only the mean and standard deviation of feature maps are computed. Yang *et al.* [29] proposed cross-channel interaction at the same layer in computer vision area, which encourages the same layer's channel communication with each other to produce performance gain. As far as we know, the C3 block has demonstrated a good number of advantages, but it has rarely been exploited in a ubiquitous HAR scenario, which has a great potential to improve the representation ability of shallow networks.

III. Model

A. Formulation

In Section III, we detail the C3 structure within a CNN, in which related formulations of cross-channel interaction between channels is illustrated. Fig. 2 is the sketch map of the network. Actually, it is a very crucial step to segment sensor time series in the activity recognition process. A sliding window approach has been extensively leveraged to perform segmentation at fixed window size, in which streams of sensor data are usually split into continuous subs-sequences called windows, and each window is associated with a specific activity. We then may insert the C3 block to a few convolutional layers to perform activity prediction. If one neural network has L layers and there are n_l filters in each layer. We use $X_l = \{x_l^1, \ldots, x_l^{n_l}\}$ to represent the feature responses of the lth layer. In general, after the channel-wise interaction, the updated response can be formulated by

$$\bar{x}_l^i = x_l^i + f_l^i\left(x_l^1, \ldots, x_l^{n_l}\right). \tag{1}$$

In this formulation, f_l^i is a function that is used to accumulate feature responses of all channels. At the same time, it updates a particular channel's encoded features. The information exchange between different channels can be named as cross-channel communication (C3), which is realized via f_l^i. In SE block [25], the function f_l^i can be deemed as a simple FC layer, which is used to realize simple communications among all channels. Compared with SE block, the function f_l^i plays a similar role in the C3 block, but it allows for a more comprehensive communication between channels via graph neural network [29], [30], where every channel can be seen as a node in the C3 block. This cross-channel communication allows an all-side communication across the whole network. The details of the model are discussed in the next section.

B. Architecture

The feature encoding, message passing, and feature decoding are the three main parts of cross-channel communication network.

1) Feature Encoder: All channels' feature responses are extracted by this module. To be specific, for a given response map x_l^i, it is first flattened to a 1-D feature, and then passed to two FC [6], [29] layers

$$\begin{aligned} y_l^i &= f_{\text{enc}}^{\text{in}}\left(x_l^i\right), \\ z_l^i &= f_{\text{enc}}^{\text{out}}\left(\sigma\left(y_l^i\right)\right). \end{aligned} \tag{2}$$

In the network, two kinds of linear function are $f_{\text{enc}}^{\text{in}}$ and $f_{\text{enc}}^{\text{out}}$. A rectified linear unit (ReLU) can be represented by σ. In order to reduce feature dimension by a factor of $\alpha > 1$, we add a bottleneck after $f_{\text{enc}}^{\text{in}}$ in this module of feature encoder.

2) Message Passing: In order to encode different feature response's representation, the message passing module is used to guarantee all channels' interaction with each other. At the same time, each channel's feature response is updated.

Graph convolutional network (GCN) [30] is a representative approach to learn such cross-channel interaction. In particular, graph attention network has been proposed, in which a soft attention mechanism is embedded into the GCN. Cross-channel interaction's formulation is similar to graph attention network. To be specific, in an undirected graph, $Z = \{z_l^i\}$ are nodes. Between two nodes, we denote edge strength by $s_{ij} = f_{att}(z_l^i, z_l^j)$. Recently, various methods have been adopted to learn f_{att} [24], [29], [31]. We select a simple yet effective method which can easily compute the edge strength

$$\bar{z}_l^i = \sum_{k=1}^{h_l w_l} z_l^i[k]/(h_l w_l),$$
$$s_{ij} = -\left(\bar{z}_l^i - \bar{z}_l^j\right)^2 \tag{3}$$

in which h_l and w_l denote the height and the width of z_l^i at the l th layer, respectively. The flattened vector z_l^i's kth element can be represented as $z_l^i[k]$. In the formulation, the feature encoder's average output can be used for increasing message passing period's robustness. We then compute the negative square distance which allows more communication between similar channels. In this way, we make group of similar channels, which becomes more complementary and diverse. The normalized attention scores can be obtained after a softmax layer. a_{ij} denotes the normalized attention scores. The output $Z = \{z_l^i\}$ is formulated as

$$Z_j^i = \sum_{j=1}^{n_l} a_{ij} z_l^j \tag{4}$$

n_l denotes the number of channels at the lth layer as indicated above.

3) Feature Decoder: This module is in charge of obtaining the information of all corrected channels and reshape this information to the same size of the original input. The feature decoder use a normal convolution operation, which passes the information to the next layers. The feature decoder goes to effect after acquiring channel-wise updated output Z.

Feature encoder, message passing, and feature decoder allow all neurons at the same level to communicate for complementing with each other. In the C3 block, each neuron's information from the same layer is first computed by the encoder. These neurons interact with each other via message passing module, which uses a neural network to pass the information of one neuron to all other neurons. Finally, the information is collected by the decoder and sent to subsequent layers. All in all, feature encoder, message passing, and feature decoder allow all neurons at the same level to communicate for complementing with each other.

4) Complexity of Computing and Model: The computational burden of C3 block is very light. There are only two FC layers in each C3 block, which is not involved in the number of channels. As a result, the number of parameters in network is low. In effect, through many experiments, we find that adding the C3 block after all the layers is unnecessary. Adding the C3 block after only a few layers is able to further reduce memory or computational burden without compromising accuracy, which is very useful in HAR scenario. The part will be explored in the following section.

IV. Experiment

A. Datasets

We utilize the same preprocessing techniques that have been well-established in the four benchmark HAR datasets [37]–[40]. However, for various datasets, there is still no clear consensus on the optimal window size, which should be preferably employed. So far, it has been rarely and vaguely investigated. For larger window size, the HAR system has to "wait" the longer for a new window to be available for predicting. According to common intuition, decreasing window size is more beneficial for a faster activity detection, as well as reduced computing resources. Instead, raising window size is normally motivated for the recognition of complex activities, which takes place for a long period of time. In fact, most designs normally depend on window size used in previous researches, but there are no strict studies to explore them. For fair comparison, we still select the same values used in previous successful cases [13], [18], [32], [33], [35] for each dataset. The details of how to preprocess datasets can be seen in Table I. For each dataset, the heterogeneous sensor values are normalized into zero mean and unit variance via subtracting the mean and dividing by the standard deviation.

1) UCI-HAR Dataset [37]: It was collected by University of California Irvine to test machine learning algorithms on HAR task. Thirty Volunteers between 19 and 48 were chosen to join in the data collection. They were all equipped with Samsung Galaxy S2 on their waists. The six activities of daily living (ADLs) performed in a supervised scenario were standing, lying, walking, walking upstairs, and walking downstairs. The data were collected by triaxial angular velocity and acceleration sensors. The activity recognition begins with the acquisition of the sensor signals, which are subsequently pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 s and 50% overlap.

2) OPPORTUNITY Dataset [38]: The project was conducted by Daniel *et al.* in University of Sussex. They built a rich sensor environment that consists of 15 wireless and wired networked sensor systems. The sensor system has 72 sensors of ten modalities in it. On body, each subject was equipped with a rich number of sensors for machine recognition of human activities. As a result, they collected 17 morning activity data from four subjects. The issue of missing sensor values can be handled by either by interpolation or repetition of previous values. Among them, using interpolation consistently performs better.

3) PAMAP2 Dataset [39]: This dataset was collected by the Department of Augmented Vision German Research Center of artificial intelligence. Within the physical activity monitoring for aging people (PAMAP) project, the researchers recorded 18 activities from nine subjects that consist of walking, cycling, rope jumping, etc. All subjects wore three inertial measurement units (IMUs) and a heart-rate-monitor. A simple linear interpolation method is used to handle missing sensor data. In order to avoid dealing with eventual transient activities, 10-s data at the beginning and the end of each labeled activity instance is removed, respectively.

TABLE I
SIMPLE DESCRIPTION OF BENCHMARK HAR DATASETS

Attribute \ Dataset	UCI-HAR	OPPORTUNITY	PAMAP2	UniMib-SHAR
Sampling Rates	50Hz	30Hz	100Hz	50Hz
Number of Categories	6	17	12	17
Proportion of Training Data	70%	70%	80%	70%
Proportion of Testing Data	30%	30%	20%	30%
Sliding Window Size	128	64	512	151
Overlap Rates	50%	50%	50%	50%

TABLE II
SIMPLE DESCRIPTION OF CNN

Simple Description \ Dataset	UCI-HAR	OPPORTUNITY	PAMAP2	UniMib-SHAR
Layer1	C(64)	C(64)	C(128)	C(64)
Layer2	C(128)	C(256)	C(256)	C(256)
Layer3	C(256)	C(384)	C(384)	C(384)
FC	✓	✓	✓	✓
Softmax	✓	✓	✓	✓
Training time(epoch)	200	200	200	200
Batch size	64	1024	128	64
Learning rate	0.001	0.001	0.001	0.001
Reduction Ratio(α)	8	16	16	8

TABLE III
ACCURACY(%)&PARAMETERS(M)&FLOPS(M) OF MODELS ON VARIOUS DATASETS

Model + Method \ Dataset	UCI-HAR	OPPORTUNITY	PAMAP2	UniMib-SHAR
3 layers CNN (baseline)	96.13&0.341&20.64	77.86&1.347&24.5	90.23&0.869&6.76	73.91&1.522&40.64
3 layers CNN + C3	**96.98**&0.342&20.7	**80.23**&1.347&24.52	**91.93**&0.869&6.77	**75.42**&1.524&40.67
3 layers CNN + C3 (5-Fold Cross Validation)	96.96&0.342&20.7	80.14&1.347&24.52	91.76&0.869&6.77	75.16&1.524&40.67
6 layers CNN	96.84&0.599&46.69	79.59&3.02&56.34	91.11&2.935&23.6	74.75&2.213&75.66
Other Researchers' Results	96.27 [32]&1.3&34.99	**79.32** [33]&-&-	91.4 [32]&2.86&79.13	74.41 [32]&1.87&40.73
	96.97 [33]&0.35&-	75.54 [13]&-&-	**92.21** [33]&2.6&-	**74.66** [34]&-&-
	95.75 [35]&-&-	76.83 [18]&-&-	85.4 [36]&-&-	-
	95.18 [14]&-&-	-	89.96 [16]&-&-	-

4) UniMib-SHAR Dataset: Micucci *et al.* [40] in University of Milan-Bicocca collected a new acceleration dataset. The samples were acquired by a smartphone with Android operating system. The whole dataset was designed for monitoring human activity and detecting falling. Thirty volunteers ranging from 18 to 60 years contributed to 11 771 samples. In order to preprocess acceleration signals, we need to remove low-frequency gravitational component. A Butterworth low-pass filter with a 20-Hz cutoff frequency is adopted to generate the accelerometer data without gravitational component.

B. Setup of Networks

In this section, we will demonstrate the details of C3 block. The optimizer we choose is Adam. Taking into account the peculiarities of different datasets, we set different learning rates for various datasets. The learning rate is set to decay exponentially to speed training process. The deep learning framework we used is Pytorch. We conduct our experiments on a deep learning server machine (GPU: RTX 3090 with 24 GB, CPU: Intel i7 6850K, RAM: 64 GB). Table II shows the detailed description of networks. C(L_s) means a convolutional layer that has L_s feature maps. For different datasets, we set $\alpha = 8$ or 16, which is in charge of compressing feature dimension to reduce computational cost.

C. Quantitative Comparison

In this section, we will show our experiment results. The representative baseline networks contain three layers or six layers. We also compared our results with recent state-of-the-art performance on several benchmark HAR datasets. The performance improvements are presented in Table III.

1) Improvement on UCI-HAR Dataset: Using cross-channel interaction idea, we insert C3 block after the third layer. We compared the three models, and the test accuracy curves are shown in Fig. 3. The size of feature maps fed into C3 block is $4 \times 15 \times 256$ (W $\times$ H $\times$ N). From results in Table III, it can be observed that the three-layer

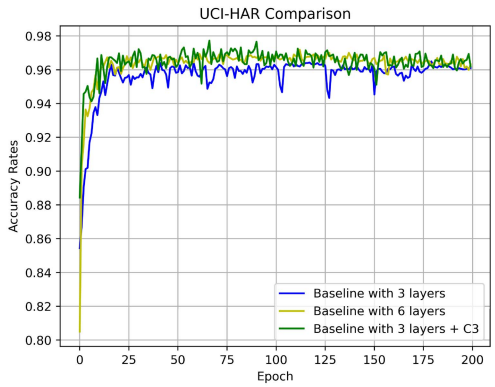


Fig. 3. Test accuracies with different models on UCI-HAR.

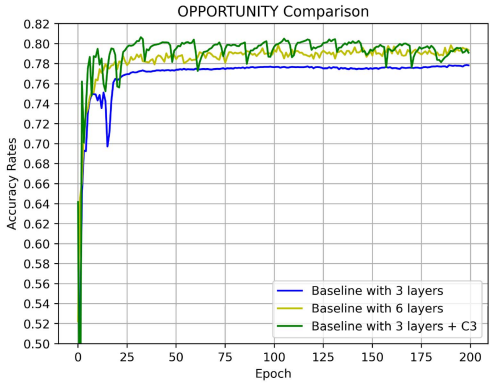


Fig. 4. Test accuracies with different models on OPPORTUNITY.

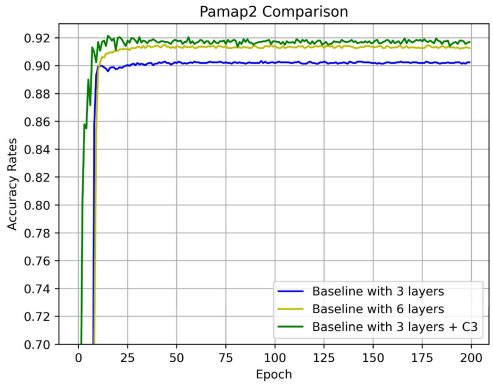


Fig. 5. Test accuracies with different models on PAMAP2.

CNN with C3 block outperforms the baseline three-layer and six-layer CNN by an accuracy improvement of 0.85% and 0.14%, respectively. For efficient consideration, the three-layer CNN with C3 block have almost the same parameters and Flops with the baseline. Compared with the six-layer CNN, it has much lower memory and computational overhead.

In Table III, we also compare our results with other competitive methods. According to classification accuracy, our method is superior to [14], [32], and [35] by 1.8%, 0.71%, and 1.23%, respectively. As far as we know, the best result on UCI-HAR dataset is obtained by [33]. The C3 block has almost the same performance with [33] at a much lower computational burden. The C3 block is more beneficial for the HAR task, because it can strike a more reasonable trade-off between accuracy and computational budget.

2) Improvement on Opportunity Dataset: We continue to perform comparison experiments on OPPORTUNITY dataset by inserting C3 block after the third layer. The input feature maps with the size $14 \times 1 \times 384$ (W $\times$ H $\times$ N) are fed into C3 block. The accuracy improvement caused by C3 block is shown in Table III, and the test accuracy curves are presented in Fig. 4. When compared with the baseline three-layer and six-layer CNN, the three-layer CNN with C3 block can produce 2.37% and 0.64% performance gain, respectively, without any extra cost. Based on efficient consideration, it can be concluded that shallow networks with C3 block have obvious advantages over baseline deep networks.

The proposed C3 method is compared with other recent researches, and the comparison results are shown in Table III. In terms of classification accuracy, our method surpasses [13], [16], and [33] by 4.69%, 3.4%, and 0.91%, respectively. To our knowledge, the best result on OPPORTUNITY dataset is [33]. As a comparison, the C3 block is more effective and efficient, which indicates a lightweight advantage with comparable classification performance.

3) Improvement on PAMAP2 Dataset: We add the C3 block after the third layer and send the $11 \times 2 \times 384$ (W $\times$ H $\times$ N) sized feature maps into the C3 block for communication. We train three models and test accuracy curves are illustrated in Fig. 5. According to the results in Table III, we find that the *three*-layer CNN with C3 block yield 1.7% and 0.82% performance gain, respectively, when compared with the baseline *three*-layer and *six*-layer CNN. In terms of accuracy, it is significantly superior to the baseline CNN at almost the same cost. In addition, it also has an obvious advantage over a deep baseline network with *six*-layer from efficient consideration.

We also compare our method with other state-of-the-art literature works in Table III. For example, our method outperforms [16], [32], and [36] by an accuracy improvement of 1.97%, 0.53%, and 6.35%, respectively. As far as we know, Teng *et al.* [33] surpassed our results by 0.28%, which is a negligible performance enhancement. However, their method requires much more memory burden than ours. It is evident that the lightweight C3 block provides a better choice for deep model design in ubiquitous HAR scenario.

4) Improvement on UniMib-SHAR Dataset: We investigate the effectiveness and efficacy of the proposed C3 in three representative networks. Without loss of generality, we insert the C3 block at the third layer. The size of input feature maps sent to C3 block for communication is $11 \times 2 \times 384$ (W $\times$ H $\times$ N). The test accuracy curves of three networks are given in Fig. 6, and the comparison results are presented in Table III. It can be seen that the C3 block brings up an accuracy improvement of 1.51% over the *three*-layer baseline with a minimal increase in computational burden. The C3 network even improves accuracy by 0.67% over baseline deep network with *six*-layer. It is worthwhile to note that shallower network even surpasses deeper network, indicating that the learned features under C3 block are more representative.

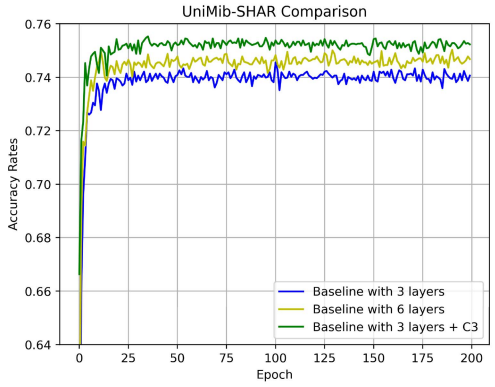


Fig. 6. Test accuracies with different models on UniMib-SHAR.

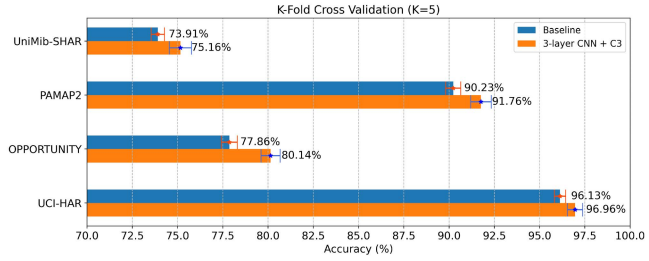


Fig. 7. K-fold cross validation ($K = 5$) on benchmark datasets.

We continue to compare the proposed C3 method with other recent researches. As can be seen in Table III, our proposed C3 outperforms [32] and [34] by a large margin accompanied by little increase of computational overhead. Concretely, it acquires an accuracy increase of 1.01% and 0.76%, respectively. To the best of our knowledge, the best result on UniMib-SHAR dataset is reported in [32]. Our accuracy is very close to theirs, but our resource consumption is much smaller. The ubiquitous HAR computing usually requires a lightweight model that has a better performance (i.e., higher accuracy, smaller resource consumption). Obviously, the C3 block is more suitable for HAR task.

5) K-Fold Cross-Validation ($k = 5$): We conduct K-fold cross-validation to verify the robustness of the proposed model. The datasets will be randomly divided into K equally sized parts. We will then train our model K times. For each run, a single fold from K folds is used as hold-out test set, while the rest folds are set aside for training. A total of K models is fit and evaluated on test sets. For deep learning, if K is raised, we have to train more models, which is a tedious and time-consuming process. Without loss of generality, K is set to 5, and the mean accuracy is reported in Fig. 7. It can be seen that the C3 block can reliably produce performance gain for each HAR dataset.

D. Analyzing the C3 Block

From different aspects, we analyze the C3 block in detail via several ablation experiments.

1) Can C3 Block Reduce the Depth of Network?: Theoretically speaking, the CNNs with six layers should have a better representation ability than those with three layers. According to our results in Fig. 8, the accuracy improvement

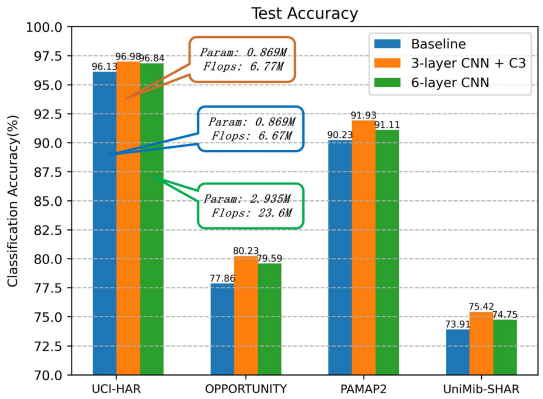


Fig. 8. Performance comparisons between shallow networks and deep networks.

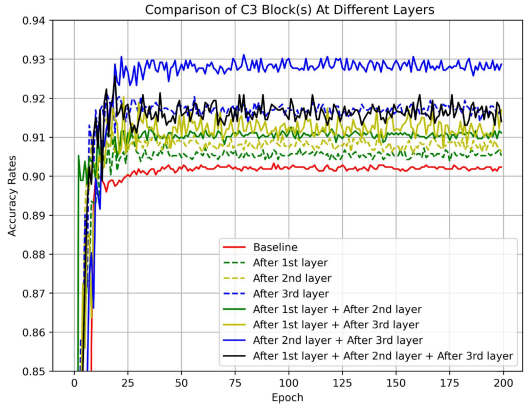
caused by the *six*-layer CNN is accompanied by a large increase in the number of parameters and Flops, which is not applicable in ubiquitous HAR scenario. As shown in Table III, the *three*-layer CNN with C3 block is able to acquire a comparable or even better accuracy than the *six*-layer CNN at a much smaller memory and computation budget. That is to say, with the help of C3 block, shallow networks can show their lightweight advantage and have a better classification accuracy than baseline deep CNNs, which is more beneficial for the HAR task.

2) Where Is the Best Place to Add C3 Block?: We conduct ablation experiments on PAMAP2 and UCI-HAR dataset to investigate the effect of C3 block at different layers. As indicated in Fig. 9 and Table IV, the most effective way is to add the C3 block after the second or third layer. According to recent literatures [41], it can be attributed to the reason that high-level semantic information is often encoded at higher layers, in which the neurons can use the C3 block to get more high-level and diverse feature responses. As a result, adding C3 block at higher layers can learn more useful information to perform activity recognition. Due to more filters at the beginning layer, there are more parameters compared with the second or third layer. Therefore, merely adding the C3 block at the final few layers can further reduce model complexity without compromising classification accuracy.

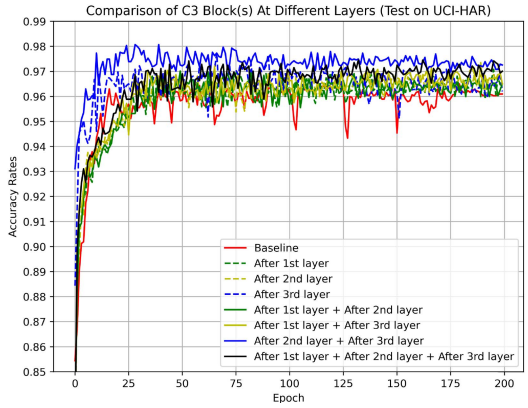
3) Which Part of C3 Block Works?: We specifically remove the encoder/decoder and message passing part from C3 block to analyze their independent contribution to performance improvement. We conduct the ablation experiment on PAMAP2 dataset. As shown in Table V and Fig. 10, it can be seen that either encoder/decoder or message passing is able to raise performance. For the former, the encoder/decoder alone is able to capture the corresponding channel's feature responses without the help of message passing, which leads to performance improvement. In comparison, the message passing outperforms the encoder/decoder. For the latter, all neurons have a better feature representation capacity because the Message Passing can encourage information exchange across different channels. When the encoder/decoder and message passing are combined, the performance is remarkably improved.

TABLE IV
ACCURACY(%)&PARAMETERS(M)&FLOPS(M) OF C3 BLOCK(S) AT DIFFERENT LAYERS

After 1st layer	After 2nd layer	After 3rd layer	PAMAP2	UCI-HAR
✓	-	-	90.52&0.869&6.77	96.23&0.341&20.7
-	✓	-	90.67&0.869&6.77	96.64&0.341&20.7
-	-	✓	**91.93&0.869&6.77**	**96.98&0.341&20.7**
✓	✓	-	91.16&0.870&6.78	96.97&0.399&20.91
✓	-	✓	91.69&0.870&6.78	96.71&0.399&20.91
-	✓	✓	**92.91&0.870&6.78**	**97.45&0.399&20.91**
✓	✓	✓	91.79&0.871&6.79	97.07&0.401&21.06



(a)



(b)

Fig. 9. Effect of C3 block at different layers. (a) PAMAP2. (b) UCI-HAR.

TABLE V
TEST ACCURACY OF C3 BLOCK WITH DIFFERENT PARTS

Encoder-decoder	Message passing	Accuracy(%)
-	-	90.23
✓	-	90.62
-	✓	91.11
✓	✓	**91.93**

E. Visualizations

1) Confusion Matrices: The confusion matrices have been computed in Fig. 11 to visually show performance

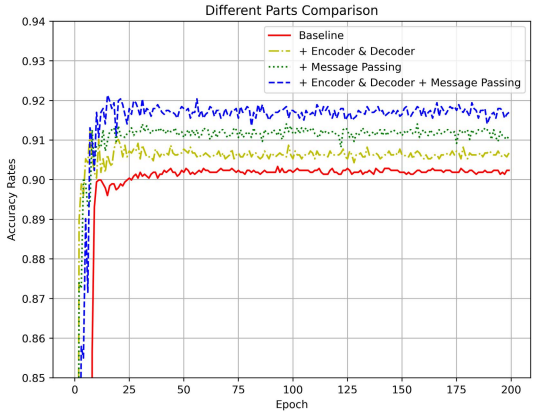
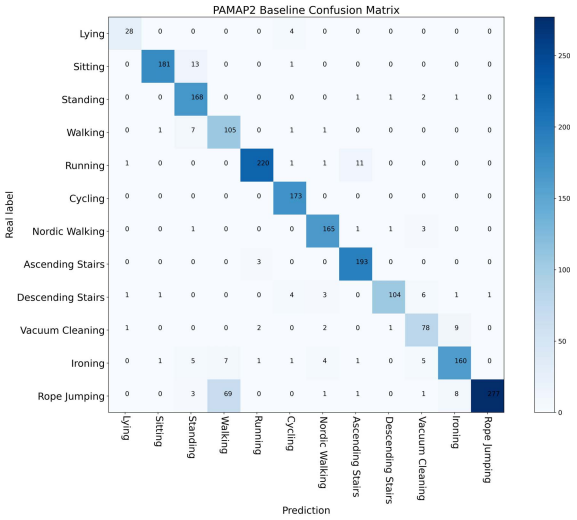


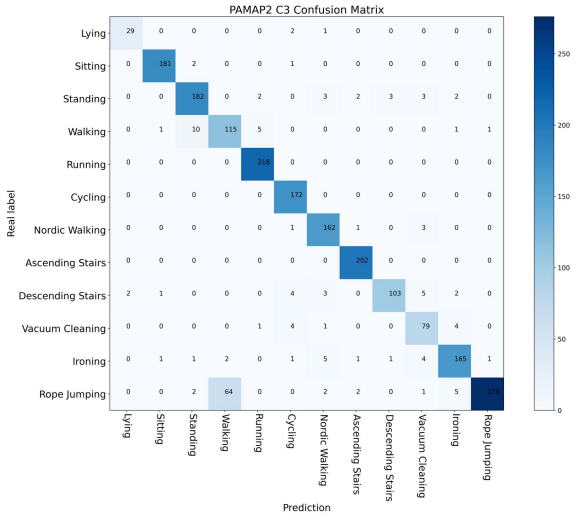
Fig. 10. Accuracy improvement caused by different parts within C3 block.

improvement. All the tests are conducted on PAMAP2 dataset. Compared with the *six*-layer CNN, it can be seen that the recognition performance (*three*-layer CNN + C3) on every activity nearly have an improvement. The number of correctly classified activity examples can be seen along the diagonal line of confusion matrices. The darker color represents a larger number. To be specific, due to the effect of C3 block, the number of correct classifications provided by shallow CNN is larger than the baseline deep network.

2) Visualization of Class Activation Maps: In order to find how the C3 block enhance each channel's feature responses, we try to build a visualization of class activation map (CAM) [42]. The CAM is an efficient yet clear method to detect changes across channels. We train the CNN with three layers on PAMAP2 dataset. In Fig. 12, we extract heat maps from our model's CAM. The filters could always locate salient sensors in sensor time series. For example, Fig. 12(a) shows a relatively simple activity: sitting. We can directly oberve that the IMUs (hand: Z; ankle: X; chest: Y) are more excited points than others. Fig. 12(b)'s lying is a little more complex activity than sitting. Thus its IMUs (hand: Y Z; ankle: X; chest: X Y) have more excited points than sitting. We continue to investigate a more complex activity: walking. Compared with sitting and lying, walking has more movement on subjects' ankles. So more IMUs (hand: Y; ankle: X Y Z; chest: X) become excited points. Among Fig. 12, the activity cycling in Fig. 12(d) is the most complex activity, where the subjects have various actions on their ankles, hands, and chest, that are even more complex than walking. As a result, we can



(a)



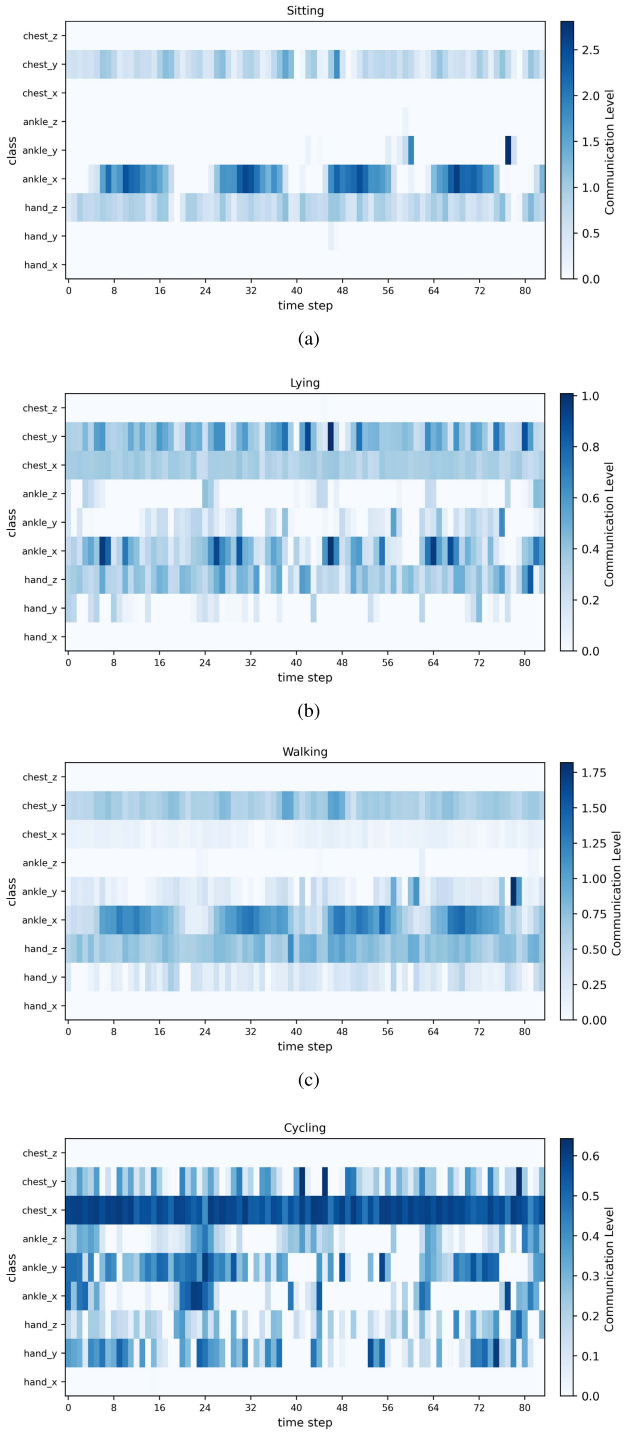
(b)

Fig. 11. Confusion matrices on PAMAP2 dataset. (a) Six-layer CNN. (b) Three-layer CNN + C3.

see different IMUs (hand: Y Z; ankle: X Y Z; chest: X Y) have become excited points. These visualizations demonstrate the effect of C3 block to feature extraction in CNN-based HAR. With the help of the C3 block, each channel's filters are able to learn in a more comprehensive way.

F. Prediction on Real-Time Platform (Raspberry Pi 3 B Plus)

For efficient consideration, we continue to evaluate the actual running time of the proposed method in resource-limited embedded platform. In order to realize a real-time HAR system, we implement two main steps as follows: 1) train the deep model with C3 block on training set from UCI-HAR; 2) install this trained model into embedded platform and run it to acquire sensor input and output a real-time prediction. The Raspberry Pi 3 B plus with ARM Cortex-A53 and 1GB SDRAM is chosen as our test platform, because the PyTorch deep learning library has a good combability with Raspberry PI operating system. A Raspberry Pi-based program



(a) (b) (c) (d)

Fig. 12. Visualization of feature responses. (a) Sitting. (b) Lying. (c) Walking. (d) Cycling.

is developed for real-time activity recognition, and its user interface is shown in Fig. 13. We perform timing after the model is loaded and starts to output a prediction. The two networks with and without C3 block are compared in Table VI. It can be seen that the baseline network without C3 block takes around 73.39–79.73 ms to process one window. The inference speed reaches 80.47–95.49 ms per each window for the same network structure with the C3 block. According

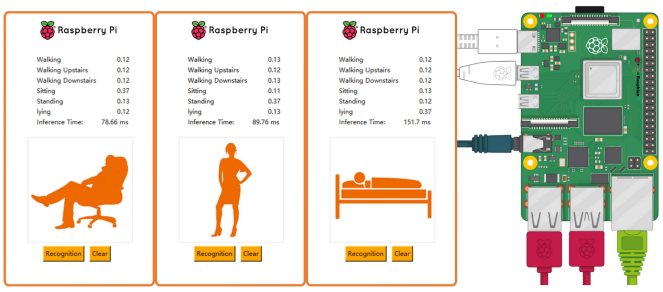


Fig. 13. Actual operation on Raspberry Pi 3 B plus system.

TABLE VI
INFERENCE SPEED OF ACTUAL OPERATION

Model	Inference Time (*Window/ms*)
3-layer CNN (Baseline)	73.39∼79.73
3-layer CNN + C3	**80.47∼95.49**
6-layer CNN	145.81∼168.3

to Table III, we could clearly observe that there is only a slight increase in computational overhead. The baseline deep network with six-layer takes around 145.81–168.3 ms to process one window. All in all, the C3 method is able to strike a better trade-off between accuracy and computational cost.

V. CONCLUSION

In this article, we first introduce an effective and efficient network block in sensor based HAR scenario, called as cross-channel communication, i.e, C3. Most existing CNNs process sensor input by extracting channel-wise features, and the information from each channel can only be hierarchically propagated, which overlooks information exchange among all channels. In C3, all channels at the same layer are able to have a more comprehensive interaction by graph neural network to learn activity features, which improves the representation ability of the network. We show the advantages of C3 on a large variety of HAR tasks, which enables shallower CNNs to aggregate more useful information and surpasses baseline deep networks and other competitive methods. We hope that the analyses of C3 may encourage further advances in deep HAR research.

REFERENCES

[1] T. Tuncer, F. Ertam, S. Dogan, and A. Subasi, "An automated daily sports activities and gender recognition method based on novel multikernel local diamond pattern using sensor signals," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9441–9448, Dec. 2020.

[2] Z. Chen, C. Jiang, S. Xiang, J. Ding, M. Wu, and X. Li, "Smartphone sensor-based human activity recognition using feature fusion and maximum full a posteriori," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 3992–4001, Jul. 2020.

[3] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surveys*, vol. 46, no. 3, pp. 1–33, Jan. 2014.

[4] P. Rashidi and D. J. Cook, "Keeping the resident in the loop: Adapting the smart home to the user," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 5, pp. 949–959, Sep. 2009.

[5] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1192–1209, 3rd Quart., 2013.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[7] M. Khanafer and S. Shirmohammadi, "Applied AI in instrumentation and measurement: The deep learning revolution," *IEEE Instrum. Meas. Mag.*, vol. 23, no. 6, pp. 10–17, Sep. 2020.

[8] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, Mar. 2019.

[9] M. Zhang and A. A. Sawchuk, "Human daily activity recognition with sparse representation using wearable sensors," *IEEE J. Biomed. Health Informat.*, vol. 17, no. 3, pp. 553–560, May 2013.

[10] S. Guo, X. Zhang, Y. Du, Y. Zheng, and Z. Cao, "Path planning of coastal ships based on optimized DQN reward function," *J. Mar. Sci. Eng.*, vol. 9, no. 2, p. 210, Feb. 2021.

[11] T. Jin, X. Yang, H. Xia, and H. Ding, "Reliability index and option pricing formulas of the first-hitting time model based on the uncertain fractional-order differential equation with Caputo type," *Fractals*, vol. 29, no. 1, pp. 2150012–2150183, 2021.

[12] T. Jin, H. Ding, H. Xia, and J. Bao, "Reliability index and Asian barrier option pricing formulas of the uncertain fractional first-hitting time model with Caputo type," *Chaos, Solitons Fractals*, vol. 142, Jan. 2021, Art. no. 110409.

[13] F. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, Jan. 2016.

[14] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 1307–1310.

[15] K. Wang, J. He, and L. Zhang, "Attention-based convolutional neural network for weakly labeled human activities' recognition with wearable sensors," *IEEE Sensors J.*, vol. 19, no. 17, pp. 7598–7604, Sep. 2019.

[16] M. Zeng *et al.*, "Understanding and improving recurrent networks for human activity recognition by continuous attention," in *Proc. ACM Int. Symp. Wearable Comput.*, Oct. 2018, pp. 56–63.

[17] H. Ma, W. Li, X. Zhang, S. Gao, and S. Lu, "AttnSense: Multi-level attention mechanism for multimodal human activity recognition," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3109–3115.

[18] M. Zeng *et al.*, "Convolutional neural networks for human activity recognition using mobile sensors," in *Proc. 6th Int. Conf. Mobile Comput., Appl. Services*, 2014, pp. 197–205.

[19] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2736–2744.

[20] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*. [Online]. Available: http://arxiv.org/abs/1510.00149

[21] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1389–1397.

[22] X. Gao, Y. Zhao, Ł. Dudziak, R. Mullins, and C. Xu, "Dynamic channel pruning: Feature boosting and suppression," 2018, *arXiv:1810.05331*. [Online]. Available: http://arxiv.org/abs/1810.05331

[23] J. Jeong and J. Shin, "Training CNNs with selective allocation of channels," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3080–3090.

[24] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.

[25] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[26] L. Chen *et al.*, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5659–5667.

[27] J. Dai *et al.*, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[28] Y. Wu and K. He, "Group normalization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[29] J. Yang, Z. Ren, C. Gan, H. Zhu, and D. Parikh, "Cross-channel communication networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1295–1304.

[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907

[31] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*. [Online]. Available: http://arxiv.org/abs/1710.10903
[32] Y. Tang, Q. Teng, L. Zhang, F. Min, and J. He, "Layer-wise training convolutional neural networks with smaller filters for human activity recognition using wearable sensors," *IEEE Sensors J.*, vol. 21, no. 1, pp. 581–592, Jan. 2021.
[33] Q. Teng, K. Wang, L. Zhang, and J. He, "The layer-wise training convolutional neural networks using local loss for sensor-based human activity recognition," *IEEE Sensors J.*, vol. 20, no. 13, pp. 7265–7274, Jul. 2020.
[34] F. Li, K. Shirahama, M. Nisar, L. Köping, and M. Grzegorzek, "Comparison of feature learning methods for human activity recognition using wearable sensors," *Sensors*, vol. 18, no. 3, p. 679, Feb. 2018.
[35] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Syst. Appl.*, vol. 59, pp. 235–244, Oct. 2016.
[36] Y. Guan and T. Plötz, "Ensembles of deep LSTM learners for activity recognition using wearables," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 2, pp. 1–28, Jun. 2017.
[37] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Proc. Int. Workshop Ambient Assist. Living*. Springer, 2012, pp. 216–223.
[38] D. Roggen *et al.*, "Collecting complex activity datasets in highly rich networked sensor environments," in *Proc. 7th Int. Conf. Netw. Sens. Syst. (INSS)*, Jun. 2010, pp. 233–240.
[39] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput.*, Jun. 2012, pp. 108–109.
[40] D. Micucci, M. Mobilio, and P. Napoletano, "UniMiB SHAR: A dataset for human activity recognition using acceleration data from smartphones," *Appl. Sci.*, vol. 7, no. 10, p. 1101, Oct. 2017.
[41] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2014, pp. 818–833.
[42] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

Wenbo Huang received the B.S. degree from Nanjing Tech University, Nanjing, China, in 2019. He is currently pursuing the M.S. degree with Nanjing Normal University, Nanjing.

His research interests include activity recognition, computer vision, and machine learning.

Lei Zhang received the B.Sc. degree in computer science from Zhengzhou University, Zhengzhou, China, the M.S. degree in pattern recognition and intelligent system from the Chinese Academy of Sciences, Beijing, China, and the Ph.D. degree from Southeast University, Nanjing, China, in 2011.

He was a Research Fellow with IPAM, UCLA, in 2008. He is currently an Associate Professor with the School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing. His research interests include machine learning, human activity recognition, and computer vision.

Wenbin Gao received the B.S. degree from the Changzhou Institute of Technology, Changzhou, China, in 2019. He is currently pursuing the M.S. degree with Nanjing Normal University, Nanjing, China.

His research interests include activity recognition, computer vision, and machine learning.

Fuhong Min received the master's degree from the School of Communication and Control Engineering, Jiangnan University, Wuxi, China, in 2003, and the Ph.D. degree from the School of Automation, Nanjing University of Science and Technology, Nanjing, China, in 2007.

From 2009 to 2010, she was a Post-Doctoral Fellow with the School of Mechanical Engineering, University of Southern Illinois, Carbondale, IL, USA. She is currently a Professor with the School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing. Her research interests include circuits and signal processing.

Jun He (Member, IEEE) received the Ph.D. degree from Southeast University, Nanjing, China, in 2009.

He was a Research Fellow with IPAM, UCLA, in 2008. From 2010 to 2011, he was a Post-Doctoral Research Associate with the Chinese University of Hong Kong, Hong Kong. He is currently an Associate Professor with the School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing. His main research is in the areas of machine learning, computer vision, and optimization methods. In particular, he is interested in the applications of weakly supervised learning via reinforcement learning methods.