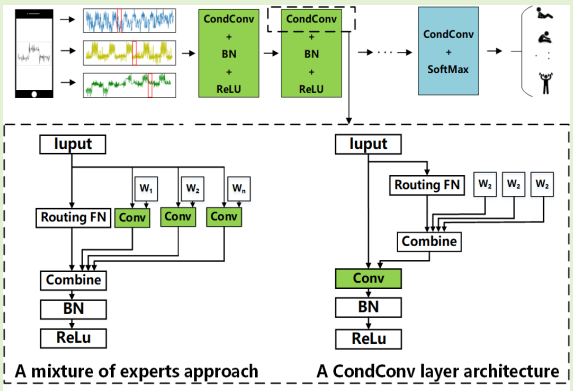
Real-time Human Activity Recognition Using Conditionally Parametrized Convolutions on Mobile and Wearable Devices

Xin Cheng, Lei Zhang, Yin Tang, Yue Liu, Hao Wu and Jun He , *Member, IEEE*

***Abstract*— Recently, deep learning has represented an important research trend in human activity recognition (HAR). In particular, deep convolutional neural networks (CNNs) have achieved state-of-the-art performance on various HAR datasets. For deep learning, improvements in performance have to heavily rely on increasing model size or capacity to scale to larger and larger datasets, which inevitably leads to the increase of operations. A high number of operations in deep leaning increases computational cost and is not suitable for real-time HAR using mobile and wearable sensors. Though shallow learning techniques often are lightweight, they could not achieve good performance. Therefore, deep learning methods that can balance the trade-off between accuracy and computation cost is highly needed, which to our knowledge has seldom been researched. In this paper, we for the first time propose a computation efficient CNN using conditionally parametrized convolution for real-time HAR on mobile and wearable devices. We evaluate the proposed method on four public benchmark HAR datasets consisting of WISDM dataset, PAMAP2 dataset, UNIMIB-SHAR dataset, and OPPORTUNITY dataset, achieving state-of-the-art accuracy without compromising computation cost. Various ablation experiments are performed to show how such a network with large capacity is clearly preferable to baseline while requiring a similar amount of operations. The method can be used as a drop-in replacement for the existing deep HAR architectures and easily deployed onto mobile and wearable devices for real-time HAR applications.**



***Index Terms*— Human activity recognition, deep learning, convolutional neural networks, conditionally parametrized convolution, wearable devices, mobile phone.**

I. INTRODUCTION

HUMAN activity recognition (HAR) has become an important research area in ubiquitous computing and human computer interaction, which has a variety of applications including health care, sports, interactive gaming, and monitoring systems for general purposes. With the rapid technical advancement of mobile phones and other wearable devices, various motion sensors have been placed at different body positions in order to collect data and infer human activity details [1]. Unlike video or wireless signals based method, mobile phone and wearable devices are more popular, which are not location dependent, easy to deploy and have no any health hazard caused by radiation. As we have known, mobile phones have become an important part of human's daily life and can be carried around almost every day. Therefore, the use of data generated by mobile phones and other wearable sensors has dominated the research landscape in HAR, which provides obvious advantages over other sensor modalities [2]. On the whole, mobile and wearable sensor based methods provide a better alternative to real-time implementation of HAR applications [3].

On the other hand, sensor based HAR mainly lies in the assumption that specific body movement can be translated into characteristic sensor signal pattern, which may be further classified using machine learning technique [4]. Recently, deep learning technique outperformed many conventional machine learning methods, which has represented an important research trend in HAR [5]. In particular, deep convolutional neural networks (CNNs) have achieved state-of-the-art performance on various HAR tasks [6]. For deep

The work was supported in part by the National Nature Science Foundation of China under Grant 61962061 and 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.(Corresponding author: Lei Zhang)

Xin Cheng, Lei Zhang, Yin Tang and Yue Liu are with School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing, 210023, China (e-mail: leizhang@njnu.edu.cn).

Hao Wu is with School of Information Science and Engineering, Yunnan University, Kunming 650091, China.

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

learning, improvements in performance have to heavily rely on increasing model size or capacity to scale to larger and larger datasets [7]. However, increasing model size or capacity inevitably leads to the increase of operations or computation cost. Building larger CNN may result in higher performance, but lead to the need for more resources such as computational power that is expensive for mobile and wearable devices. Therefore, deploying optimal deep models for mobile and wearable HAR applications are often impractical, which limits their wide use for real-time HAR applications with strict latency constraints.

At present, there have been hundreds of brands of mobile phones, which have drastically different inference time even for the same network architecture (Yu *et al.* [8]), as illustrated in Table I. Given the same response time, there is often a higher accuracy for high-end mobile phones that can implement larger models. Instead, low-end mobile phones that are hard to implement larger models have to sacrifice accuracy in order to maintain the same latency. For even the same device, the computational power still may vary due to potential consumption caused by background APPs that tend to reduce the available computing budget (e.g., a mobile phone that works in a power-saving mode). Therefore, it deserves further research to develop computation efficient CNN to perform real-time HAR using mobile and wearable sensors.

TABLE I: Runtime of MobileNet v1 for image classification on different devices.

	Google Pixel	**LG Nexus 5**	**Samsung Galaxy S3**
RunTime	116ms	332ms	553ms

Without loss of generality, there are two ways to design computation efficient CNN for mobile and wearable HAR applications. For the first case, using fewer convolutional layers or decreasing the size of existing convolutions may lead to the decrease of computation cost. Thus, current computationally efficient models often are smaller, which have suboptimal performance with fewer parameters on mobile deployment [9]. For the second case, decreasing the size of the input to convolution also can proportionally decrease computation cost. Actually, HAR using mobile phones and wearable sensors can be seen as a classic multivariate time series classification problem, which makes use of sliding window [10] to segment time series sensor signals and extracts discriminative features from them to be able to recognize activities by utilizing a classifier. Intuitively, using smaller sliding window can yield faster inference. However, in this case it often is hard to obtain most suitable size for feature extraction of HAR, which make CNNs be not able to offer best results. Therefore, as indicated in both cases, current computationally efficient models often are suboptimal for HAR. Recently, there has been rising research interest in conditional computation [11], [12], whose goal is to increase model capacity or performance without a proportional increase in computation cost. In particular, Yang *et al.* [13] proposed an idea of conditionally parameterized convolution (CondConv), which can easily be optimized by gradient descent. According to our research motivation, replacing conventional convolutions with CondConv could be one feasible step to realize efficient inference for mobile and wearable HAR applications without compromising computation cost.

In this paper, we propose a new CNN using the idea of CondConv for HAR applications with strict latency constraints, which aims to increase model capacity or performance while maintaining efficient inference to better serve these real-time HAR applications on mobile and wearable devices. To the best of our knowledge, how to build an accurate CNN for HAR without sacrificing computational cost has been rarely explored, and this paper is the first try to develop conditional-computation CNN for real-time HAR in ubiquitous and wearable computing area. To be specific, we replace the standard convolution $W * x$ with CondConv which is a linear combination of n experts $(\alpha_1 \cdot W_1 + \alpha_2 \cdot W_2 + ... + \alpha_n \cdot W_n) * X$, where α_1,...,α_n are weight functions of the input learned through gradient descent. The standard convolution $W * x$ requires expensive computation cost as it needs to be computed at many different positions within the input. In comparison with standard convolution, increasing the number of experts in the CondConv is able to greatly improve the representing ability of CNN without compromising computation cost, as all the experts are combined only once per input. Our main contributions are summarized as follows:

- Firstly, we for the first time present a novel CNN using the idea of CondConv for HAR applications with strict latency constraint, which may increase model capacity or performance while maintaining efficient inference speed in order to better serve these real-time HAR applications in mobile and wearable devices.
- Secondly, compared with ordinary convolution, the proposed method is able to effectively improve activity recognition accuracy via increasing the number of experts. The experimental results show that there is a significant performance gain on four benchmark HAR datasets consisting of WISDM dataset [14], PAMAP2 dataset [15], UNIMIB-SHAR dataset [16], and OPPORTUNITY dataset [17] at almost the same computation cost.
- Finally, various ablation experiments are performed to analyze the effect of several important hyper-parameters such as the number of experts. We visually show the distribution of routing weights activated by activity examples in the convolutional layer. The actual inference speed is measured on a smartphone with an Android platform, which indicates its advantage with regard to typical challenges for real-time HAR in ubiquitous and wearable computing scenario.

The rest of this paper is organized as follows. Section II presents the related works in activity recognition and conditional computation. Section III details the proposed framework for HAR. In Section IV, we first describe the HAR dataset used and experimental setup, and then present the experimental result comparison and analysis from several aspects. The last section concludes this study with a brief summary and points out future research work.

II. RELATED WORKS

In recent years, deep learning has become popular in mobile and wearable sensors based HAR, due to their superior performance. In particular, CNN is one of the most researched deep learning techniques which can automatically extract features and identify the hidden or unknown activity patterns from raw time series sensor data. A number of CNN architectures for the use of HAR have been developed by researchers. For example, Zeng *et al.* [18] firstly proposed a shallow CNN based approach to recognize activities, which has achieved state-of-the-art performance in three public HAR datasets. Yang *et al.* [6] developed a new architecture of CNN, in which the convolution filters are applied along the temporal dimension for each sensor and all feature maps for different sensors are unified as an input for a classifier. CNNs that combine other fusion techniques were also proposed. Ordóñez *et al.* [19] proposed an architecture of DeepConvLSTM, which replaced the fully connected layer of CNN with Long Short Term Memory (LSTM) to capture temporal relationship contained in time series sensor data. Wang *et al.* [20] proposed an attention-based CNN which is able to enhance interesting activity in the weakly supervised learning scenarios. Ignatov *et al.* [21] proposed a CNN which combines local feature extraction with simple statistical features that preserve global information about the time series sensor data. Teng *et al.* [22] developed a layer wise training CNN for HAR with local loss, which is able to achieve remarkable performance with less parameters on various HAR application domains. Guo *et al.* [23], [24] proposed an idea of dual-ensemble class imbalance learning, where two ensemble models are nested each other to handle an imbalance classification problem in HAR scenario. On the whole, deep CNNs have yielded excellent results in terms of recognition accuracy, but often need a lot of computation cost, which is infeasible for mobile and wearable HAR applications that have strict latency constraints.

In addition, we introduced some related literatures such as model compression to enrich the related works. In order to accelerate inference while maintaining satisfactory accuracy, one main research direction is to prune network connections [25], or channels [26], in a pre-trained network, which could reduce redundant connections and meanwhile preserving classification performance. Network quantization, [27], [28], or factorization [29] are also introduced in many literatures in order to speed up inference, which can reduce redundant calculations. Knowledge distilling methods [30] are able to transfer knowledge from larger networks into smaller ones, which enables smaller networks to perform inference while preserving comparable accuracy. Overall, above these methods inevitably need a large pre-trained network. Comparing with video data that is easier to be understood and annotated by humans, it is much harder to accurately segment and label an interesting activity from a long sensor sequence. Therefore, they are very infeasible for activity recognition tasks, due to the scarcity of sensor data.

Due to the growing number of hyper-parameters, designing computation efficient CNN for HAR applications becomes increasingly difficult. In another line of research, recent research effort on visual recognition or natural language processing has been shifting to conditional computation, which aims to increase model capacity or performance without a proportional increase in computation cost. For example, Wu *et al.* [31] proposed BlockDrop method which uses reinforcement learning to dynamically learn discrete routing functions, in order to best reduce computation cost without decreasing model accuracy. Mullapudi *et al.* [32] developed HydraNet model which uses unsupervised clustering method to choose proper subset of the entire network architecture to run most efficient inference on a given input. Shazeer *et al.* [33] proposed a trainable gating network by introducing a sparsely-gated mixture-of-experts layer, which is able to determine a sparse combination of different experts to use for each example. However, these aforementioned approaches in conditional computation often require to learn discrete routing decisions of different experts across every example, which is hard to train using gradient descent and not suitable for CNN based HAR applications. Recently, Yang *et al.* [13] proposed CondConv to challenge the fundamental assumption that the same convolutional kernels should be shared for each example, which enables different expert convolution kernels to focus on their specialized examples. In particular, the CondConv can easily be trained with gradient descent without requiring access to discrete routing of each example. Despite the success of conditional computation, their primary use mainly lies in imagery or natural language processing tasks, which has never been used to perform HAR. The increasing demands for running efficient deep neural networks for HAR on mobile and wearable devices encourage our current study. In the next section, we will describe the CondConv and then present the entire architecture of deep HAR applications using CondConv.

III. MODEL

In this section, we will discuss our new CNN architecture using CondConv to handle the unique challenges existed in mobile and wearable HAR applications. An overview figure of the proposed HAR system is presented in *abstract*. For sensor based HAR, we have to firstly deal with multiple channels of time series sensor signals, in which the convolution need to be applied along temporal dimension and then be shared or unified among multiple different sensors. Due to implementational simplicity and no need of preprocessing, the sliding window technique is ideally suitable for real-time HAR applications, which has been widely used to segment time series sensor signal into a collection of smaller data pieces as an input for CNN. Hence an instance handled by CNN typically corresponds to a two-dimensional matrix with r raw samples representing the number of samples per window, in which each sample contains multiple sensor attributes recorded at time t. Though in any case the sensor signal stream must be segmented into data windows, they can be of a continuous nature. Thus, an overlap between adjacent windows is tolerated to preserve the continuity of activities. Intuitively, decreasing the size of sliding window leads to a faster activity inference, as well as a reduced need for computation cost. To make fair comparison, we still select the same window size that

is preferably used in previous state-of-the-art works.

Our main research motivation is to realize computation efficient CNN using CondConv for the practical use of HAR on mobile and wearable devices. Without loss of generality, the baseline CNN is typically comprised of four units: (i) a convolution layer with a set of learned kernels that convolve the input along temporal dimension or the previous layer's output; (ii) a ReLU layer with activation function max(x,0) that maps the previous layer's output; (iii) a max pooling layer that subsamples via finding the maximum feature map across a range of local temporal neighborhood; (iv) a Batch Normalization(BN) [34] layer used to normalize the values of different feature maps from the previous layer. Without loss of generality, we denote a standard convolution as:

$$W \in R^{C \times C^{'} \times k_h \times k_w} \tag{1}$$

with the input feature map $X \in R^{C \times h \times w}$ and the output feature map $Y \in R^{C^{'} \times h^{'} \times w^{'}}$, where (h, w), $(h^{'}, w^{'})$ and (k_h, k_w) denote the heights and the widths for the input, output, and convolutional filter respectively. Following the settings of Yang *et al.* [13], we replace the standard convolution kernels used in convolutional layers with a linear combinations of n experts:

$$Output = \sigma\left((\alpha_1 \cdot W_1 + ... + \alpha_n \cdot W_n) * X\right) \tag{2}$$

in which σ is ReLU activation function and n is the number of experts. The dimension of each kernel W_i is still the same to that in original convolution. Obviously, if the scalar weight a is constant for all examples, a CondConv layer has almost the same capacity with a standard convolutional layer. To avoid the case, the weight α_i can be computed using a routing function r_i(x):

$$r_i(x) = S\left(GlobalAveragePool\left(x\right)R\right) \tag{3}$$

Here S is Sigmoid activation function and GlobalAverage-Pool is global average pooling layer. As a result, α_i is identical to r_i(x), i.e.,α_i=r_i(x). R is a dense layer that maps the pooled inputs to n expert weights with the parameters learned across lots of training examples. To be more specific, the Eq.3 can be expressed as:

$$\alpha_i = S\left(W_{fc1} \times \frac{1}{hw} \sum_{i \in h, j \in w} X_{c,i,j}\right) \tag{4}$$

Therefore, the weights of n experts are example-dependent, which enable different experts to specialize in their interesting examples. That is to say, the weights of n experts are different across all examples, in which each individual example can be processed with different weights.

From the perspective of matrix theory, a CondConv layer can be equally expressed as:

$$Output = \sigma\left(\alpha_1 \cdot (W_1 * x) + ... + \alpha_n \cdot (W_n * x)\right) \tag{5}$$

which is more computationally expensive. As a comparison, the CondConv for each example can be computed as a linear combination of n experts, and then only one expensive convolution needs to be computed. To be specific, each additional expert requires only one additional multiply-add operation, which suggests that we can increase model capacity or performance via increasing the number of experts, with only a very small increase in computation cost. Though increasing the number of experts inevitably requires more memory resource, it is often affordable due to the rapid technical advancement of mobile phones and other wearable devices. Hence the CondConv is able to achieve higher inference performance without compromising computation cost, which provides a better alternative to serve mobile and wearable HAR that has strict latency constrains. With the increase in the number of experts, the CondConv is able to increase model capacity, which is also prone to overfitting. To avoid overfitting, we additionally introduce data augmentation via improving the overlapping rate of sliding windows, as well as randomly dropping out to ensure sufficient regularization.

IV. Experiment

We evaluate the proposed method on four public benchmark HAR datasets consisting of WISDM dataset [14], PAMAP2 dataset [15], UNIMIB-SHAR dataset [16], and OPPORTUNITY dataset [17], which are recorded with different sampling rates, number of sensors and kinds of activities. In terms of accuracy and FLOPs, we compare our method against the baseline CNN, as well as other state-of-the-art techniques that have been widely used in the HAR tasks. To make fair comparison, we restrain the baseline CNN with the same hyperparameters and regularization methods as the CondConv model. For each baseline architecture, we replace standard convolution layer with CondConv Layer to evaluate CondConv via increasing the number of experts per layer. To be specific, model performance is evaluated via varying the number of experts in the CondConv layer from 1, 2, 4, 8, 16. To fully exert the effect of CondConv, we additionally replace the fully connected layer with a 1x1 CondConv layer in some cases. For each CondConv layer, the BN layer is inserted right after a convolutional layer, but before feeding into ReLU activation [35]. We introduce the detailed parameter settings such as the number of convolution layers and kernel size in Table II. To determine the routing weight functions, we experiment with different activation functions including Tanh, Sigmoid, Softmax, LReLU, ELU and ReLU, in which the results suggest that Sigmoid significantly outperforms other activation functions. Various ablation experiments are performed to further analyze the effect of CondConv layer across different examples at different depths in the network.

Models are trained in a supervised way, and the model parameters are optimized by minimizing the cross-entropy loss function with mini-batch gradient descent using an Adam optimizer. Training is done for at least 400 epochs. The epoch that achieves the best performance is selected and the corresponding model is applied to test set. For the CondConv, increasing the number of experts will inevitably lead to the increase of parameter count, which requires enough examples to train the model. The data augmentation and dropout technique are used for the CondConv model with large capacity, which aims to ensure sufficient regularization.

TABLE II: SIMPLE DESCRIPTION OF NEURAL NETWORK PARAMETER

Layers \ Dataset	WISDM			PAMAP2			UniMib-SHAR			OPPORTUNITY		
	conv	filters	stride	conv	filters	stride	conv	filters	stride	conv	filters	stride
Layers 1	(6,1)	64	(2,1)	(6,2)	64	(3,1)	(6,1)	128	(2,1)	(5,7)	64	(1,2)
Layers 2	(6,1)	128	(2,1)	(6,2)	128	(3,1)	(6,1)	256	(2,1)	(5,7)	64	(1,2)
Layers 3	(6,1)	384	(2,1)	(6,2)	256	(3,1)	(6,2)	384	(2,1)	(5,7)	128	(1,2)
Layers 4	(1,1)	6	(1,1)	-	-	-	(1,1)	17	(1,1)	(5,7)	128	(2,3)
Layers 5	-	-	-	-	-	-	-	-	-	(5,7)	256	(2,3)

First, data augmentation technique is added via improving the overlapping rate of sliding time windows. We use smaller sliding step length to segment time series sensor signal, which is able to generate more training examples. The proposed CondConv model has the same experiment settting with that of the baseline. Second, dropout technique is applied to avoid overfitting during the training stage. However, normal combination of dropout and BN technique often lead to worse results unless some conditioning is done to prevent the risk of variance shifts. As suggested by Li *et al.* [36], the worse performance caused by the variance shift only happens when there exists a dropout layer before a BN layer. Thus, we insert only one dropout layer right before the final Softmax layer. All the experiment in this paper are implemented in Python using TensorFlow backend on a machine with an Intel i7-6850K CPU, 64GB RAM and NVIDIA RTX 2080 Ti GPU. In addition, we test the actual inference speed on a smartphone with an Android platform.

A. Experiment Results and Performance Comparison

1) ***The WISDM dataset*** *[14] :* The WISDM dataset used for the experiment is provided by the Wireless Sensor Data Mining(WISDM) Lab, which contains various human activities with 6 attributes: user, activity, timestamp, x-acceleration, y-acceleration, z-acceleration. The smartphones were placed in a front leg pocket of each dominant, in which one triaxial accelerometer embedded in smartphones with an Android platform was used to generate time series data at a constant sampling rate of 20Hz. The activities were collected from 29 subjects and each subject performed 6 distinctive human activities consisting of walking, jogging, walking upstairs, walking downstairs, sitting and standing .

In the experiment, the sliding window technique is utilized to segment the time series accelerometer signals. The size of sliding time window is set to 10s and a 95% overlapping rate is used, which equals to 0.5s of the sliding step length. The whole WISDM dataset is partitioned into two parts, in which 70% is randomly selected to generate training examples and the rest test examples. The shorthand description of the baseline CNN architecture is C(64)-C(128)-C(384)-FC-Sm, which consists of three convolutional layers and one fully connected layer. To be specific, each convolution begins with Conv-BN-ReLU and then another one. We use a 1x1 CondConv layer to replace the fully connected classification layer. The model will be trained using mini-batches with a size of 210. Adam is used for optimization. The initial learning rate is set as 0.0001, which will be reduced by a factor of 0.1 after each 50 epochs.

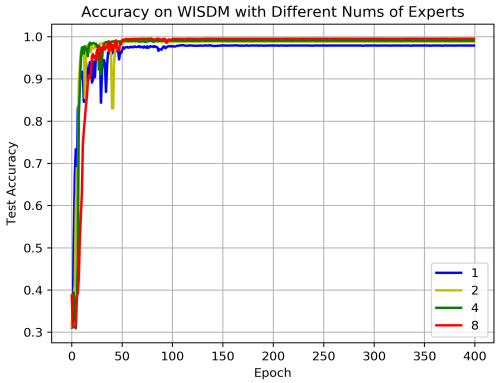


Fig. 1: Accuracy on **WISDM** dataset with different nums of experts

In Fig.1, we evaluate our model performance using CondConv via varying the number of experts. As can be seen in the figure, there is a steady increase in accuracy on test data with increasing the number of experts. During training stage, increasing the number of experts tends to make the model converge faster. In terms of accuracy and FLOPs, Table III demonstrates the classification results compared with the baseline and state-of-the-arts. The number of experts that obtains the best accuracy on test set are n=1 (98.12%), n=2 (98.94%), n=4 (99.12%) and n=8 (99.60%). From the results, we can see that the models with CondConv ($n > 1$) consistently perform better than their counterparts without CondConv (n=1). As a reference, the baseline has an accuracy of 98.12% at the cost of 30.01 MFLOPs. The CondConv model has 99.60% classification accuracy with a computation complexity of 31.69 MFLOPs. There is an improvement of 1.48% in accuracy with a very small increase in FLOPs. To the best of our knowledge, the best performance on the dataset was 98.97% using a federated learning system (Xiao *et al.* [37]). The second best result was 98.82% using a CNN with local loss (Teng *et al.* [22]). Our result with CondConv is best reported, which surpasses recent state-of-the-art results. The results indicate that the proposed model demonstrates state-of-the-art performance at almost the same computational cost.

2) ***The PAMAP2 dataset*** *[15]:* The physical activity monitoring dataset is an open source dataset available at UCI repository, which contains extensive physical activities: both everyday household and sports performed by 9 participants

TABLE III: Performance on **WISDM** Dataset with Different nums of Experts

Model	Test Acc	FLOPs
CondConv(with n=1)	98.12%	30.01M
CondConv(with n=2)	98.94%	30.25M
CondConv(with n=4)	99.12%	30.73M
CondConv(with n=8)	**99.60**%	31.69M
Teng *et al.*2020 [22]	98.82%	-
Xiao *et al.*2020 [37]	98.97%	-
Noori *et al.*2020 [38]	98.7%	-
Ravi *et al.*2016 [39]	98.20%	-

wearing 3 inertial measurement units (IMUs) and a heart rate monitor. The IMU sensors were placed over the chest, wrist and side's ankle on the dominant. The participants were asked to perform 12 protocol activities such as stand, sit, ascend stairs, descend stairs, rope jumping and run. In addition, some of them performed 6 optional activities such as watching TV, car driving, house cleaning and playing soccer. The sampling rate of heart rate monitor is 9Hz, and the sampling rate of IMUs is 100Hz; i.e. data is recorded 100 times per second. For the use of HAR, we subsample the IMU signals from 100Hz to 33.3Hz.

As indicated, HAR is typically computed over a sliding window. The sliding window length is usually fixed. Different window lengths are selected by authors in various studies. To compare the result with other works, we selecte window size of 512 (5.12 seconds) to slide one instance at a time, which leads to a 78% overlap with around 473k samples. All samples are normalized into zero mean and unit variance. We randomly select 70% of the data in each class for training, the rest for test. The shorthand description of the baseline CNN is described as C(64)-C(128)-C(256)-FC-Sm, which consists of three convolutional layers and one fully connected layer. BN is applied before ReLU activation. The batch size is set to 204 and Adam optimization [40] is used for training. The learning rate is set to 0.001, 0.0005 and 0.00001 during 12.5%, 25% and 62.5% of the total training time.

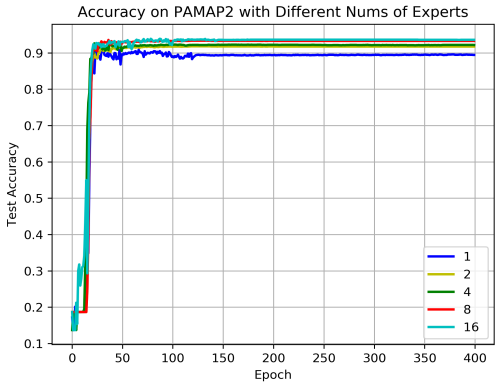


Fig. 2: Accuracy on **Pamap2** dataset with different nums of experts

Keeping all hyper-parameters except the number of experts the same, we train the model using CondConv to see if it could further improve the results. Fig.2 shows the effect of increasing number of experts on test accuracy using the CondConv architectures with n=1,n=2,n=4,n=8 and n=16. It can be seen that the model performance consistently increases when the number of experts is greater than 1. Under a variety of n, we compare classification accuracy and FLOPs with the baseline of n=1, as well as the state-of-the-arts on this dataset. Results from Table IV , it can be seen that the number of experts that achieves the best results on test set are n=1(89.97%), n=2(91.8%), n=4(92.7%), n=8(93.79%) and n=16(94.01%) respectively. Our method using CondConv with n=16 surpasses the baseline by 4.04%, accompanied by a very small increase in computational cost. As can be seen in Table IV, the best published result on this dataset using CNN is to our knowledge 91.4% (Yang *et al.* [41]). The second best result was 91% designing a smartphone inertial accelerometer-based architecture (Wan *et al.* [42]). The proposed method surpasses the state-of-the-art result by a large margin. This result implies that we can exploit this CondConv as a drop-in replacement of standard convolution to produce better results with only a small increase in computational cost.

TABLE IV: Performance on **PAMAP2** Dataset with Different nums of Experts

Model	Test Acc	FLOPs
CondConv(with n=1)	89.97%	212.57M
CondConv(with n=2)	91.80%	213.35M
CondConv(with n=4)	92.70%	215.69M
CondConv(with n=8)	93.79%	218.76M
CondConv(with n=16)	**94.01**%	224.86M
Yang *et al.*2018 [41]	91.40%	-
Wan *et al.* 2020 [42]	91.00%	-
Zeng *et al.*2018 [43]	89.96%	-
Chen *et al.*2019 [44]	90.33%	-

*3) **The UNIMIB-SHAR dataset** [16]:* UNIMiB-SHAR is a new dataset including 11771 samples designed for the use of HAR and fall detection. In a supervised condition, the 30 subjects of ages ranging from 18 to 60 years wearing a Samsung Galaxy Nexus I9250 smartphone were instructed to perform activities. Each activity was performed 2 or 6 times. The half of all participants placed the smartphone in their left pocket, and the other half in their right pocket. An embedded Bosh BMA220 3D accelerometer was used to generate examples. The whole dataset consists of 17 fine grained classes which is further grouped into two coarse grained classes: one containing samples of 9 types of activities of daily living(ADLs) and the other containing samples of 8 types of falls.

For fair comparison, the sliding window with a fixed length T=151 is selected, which equals to approximately 3s. Since the accelerometer signals are recorded at a constant sampling rate of 50 Hz, for each activity, the accelerometer signal is comprised of 3 vectors of 151 values, one for each acceleration

direction. Thus, the whole dataset contains 11,771 windows of size 151*3 in total, which describes both ADLs (7759) and falls (4192) unequally distributed across activity types. The architecture of the baseline CNN is C(128)-C(256)-C(384)-FC-Sm, which contains three convolutional layers and one fully connected layer. At the last, we use a 1x1 CondConv layer to replace the final fully connected classification layer. The samples are split into 70% training and 30% test set. Adam optimizer [40] is used to train with batch size of 203. The learning rate is set to 0.0004, 0.00001 and 0.000001 during 12.5%, 25% and 62.5% of the total training time.

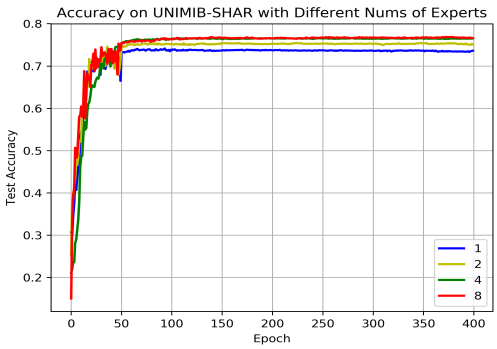


Fig. 3: Accuracy on **UNIMIB-SHAR** dataset with different nums of experts.

We evaluate the performance of our proposed method with various number of experts on this dataset. Fig.3 shows that the test accuracy will increase as the number of experts grows, which is consistent with our motivation. The CNN that utilizes CondConv always performs better than its counterpart without CondConv. Table V demonstrates the performance of our model compared with the baseline and other state-of-the-arts in terms of accuracy and FLOPs. It can be seen that our method achieves 1.32%, 2.74% and 3.16% performance gain over baseline with n=2, n=4 and n=8 respectively. There is only a small increase in computation cost. In addition, our model using CondConv outperforms other state-of-the-arts. When compared to the best result obtained by Li *et al.* [45] using CNN, our method with n=8 achieves 2.34% improvement. Our CondConv also surpasses the Long *et al's* method [46] by 1.27%, which uses dual residual networks. Under same parameter configurations, by increasing the number of experts, the CondConv with sufficient regulation is able to improve the representation ability of CNN by a large margin.

4) ***The OPPORTUNITY dataset*** *[17]:* The OPPORTUNITY dataset is publicly available on the UCI Machine Learning repository, which comprises both static/periodic and sporadic activities collected with sensors of different modalities integrated into the environment and on the subjects, in a daily living scenario. The samples were recorded from four subjects performing morning activities, in which each subject was asked to perform one ADL session and one drill session. During the ADL session, without any strict restriction, subjects performed a session five times with activities such as preparing and drinking a coffee, preparing and eating a sandwich, cleaning up, and so on. During the drill session, subjects were instructed to perform 20 repetitions of a predefined sorted set of 17 activities.

TABLE V: Performance of Different Experts for **UNIMIB-SHAR** dataset

Model \ Data	Magnitude		Raw
	Test Acc	FLOPs	Test Acc
CondConv(with n=1)	74.15%	31.53M	86.92%
CondConv(with n=2)	75.47%	31.86M	87.57%
CondConv(with n=4)	76.89%	32.46M	88.25%
CondConv(with n=8)	77.31%	33.57M	88.63%
Li *et al.*2018 [45]	74.97%	-	-
Long *et al.*2019 [46]	76.04%	-	-
Micucci *et al.*2020 [16]	65.96%	-	-
Falco *et al.*2020 [47]	-	-	86.00%

The dataset has been used in numerous activity recognition challenges. In this paper, we evaluate our method on the same subset employed in previous OPPORTUNITY challenge, which contains the samples collected from 4 subjects with only on-body sensors. The sensor signals are recorded at a sampling rate of 30Hz from 12 locations on the dominant and annotated with 18 mid-level gesture annotations. ADL1, ADL2 and ADL3 from subject 1, 2 and 3 are used as training set. ADL4 and ADL5 from subject 4 and 5 are used as test set. The size of sliding time window and sliding step length are set to 64 and 8 respectively, which generates approximately 650k samples. The baseline model is a deep CNN, whose shorthand description is presented as C(64)→C(128)→C(256)→C(256)→C(256)→FC→Sm, that contains five convolutional layers and one fully connected layer. For this experiment, The initial learning rate is set as 0.0001, which will be reduced by a factor of 0.1 after each 50 epochs using Adam with default parameters. Initial batch size is set to 204.

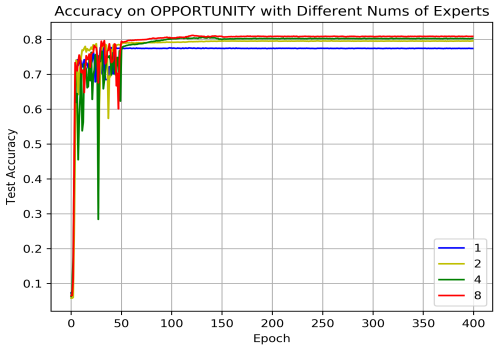


Fig. 4: Accuracy on **OPPORTUNITY** dataset with different nums of experts.

We characterize the effect of the number of experts employed to increase model capacity or performance. Fig.4 shows that increasing the number of experts tends to improve model performance with sufficient regulation. The results of the proposed method are shown in Table VI, which also includes a comprehensive list of past published deep learning techniques employed on this dataset. It can be seen that the CondConv method systematically performs best among deep architectures. As can be seen in VI, the best result was Alia

et al. [48] using Random Forest. The CondConv with the number of experts larger than 1 consistently outperforms our baseline, with a very small increase in computational cost. In the paper, we aim to improve CNN via using more experts. As a result, our method (n=8) slightly outperforms Alia *et al* [48] by 4.09%. The results indicate that more experts (n>1) can obtain better results at a negligible computational burden.

TABLE VI: Performance of Different Experts for **OPPORTUNITY** dataset

Model	Test Acc	FLOPs
CondConv(with n=1)	77.5%	126.23M
CondConv(with n=2)	78.7%	127.06M
CondConv(with n=4)	80.9%	128.84M
CondConv(with n=8)	**81.18**%	132.65M
Zeng *et al.*2014 [18]	76.83%	-
Alia *et al.*2020 [48]	77.09%	-
Hammerla *et al.*2016 [49]	74.50%	-
Guan *et al.*2017 [50]	72.60%	-

As shown in Table III-VI, the CondConv algorithm is able to significantly improve the performance of CNN at a negligible computation cost. For WISDM, PAMAP2, UNIMIB-SHAR and OPPORTUNITY datasets, our algorithm outperforms baselines (n=1) by a 1.48%, 4.04%, 3.16% and 3.68% respectively, which is accompanied by only an increase of 5.6%,5.78%,6.47% and 5.08% in terms of FLOPs.

On the whole, compared with image datasets such as ImageNet, the public HAR datasets are much smaller. In order to prevent overfitting, we propose data augmentation technique via using smaller sliding step to segment sensor signal for improving overlapping rate, which is able to generate more activity examples. In the case where there are no more than 8 experts, it can be seen that the test accuracy improves as the number of experts increases. If we further increase the experts from 8->16->32, it is more prone to overfitting, which often results in accuracy drop. Actually, the overfitting takes place because there are no more training examples to train more experts.

B. The Ablation studies

To better understand model design with the CondConv block, we conduct several ablation studies to further explore why the CondConv with larger model capacity is able to improve accuracy while maintaining efficient inference. Our ablation experiments are performed on the UNIMIB-SHAR dataset, and all hyper-parameters are exactly the same as used above. Finally, we also evaluate the actual inference time of our model on an Android smartphone.

First, we study the influence of routing weights across different classes of activities at three different depths in the network. As mentioned above, if all the experts have the same routing weight for each example, the CondConv will degenerate into standard convolutions. Thus, all the experts

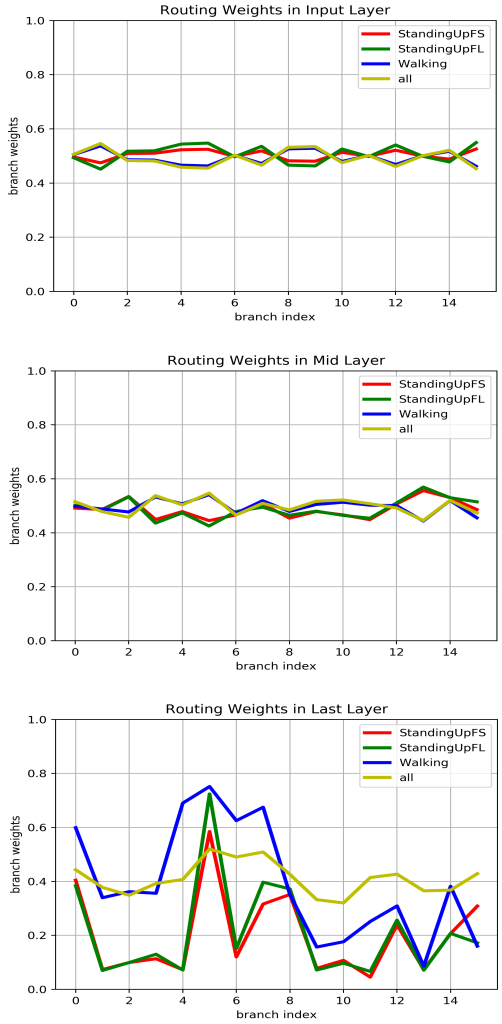


Fig. 5: Mean routing weights for three classes across the **UNIMIB-SHAR** dataset at three different depths

are example-dependent, and each individual example can yield different activation weights. We apply the CondConv in all convolutional layers as well as the final fully connected classification layer. Results are shown in Fig.5. It can be seen that the value discrepancy is increased layer by layer. For shallow layers, the distributions of routing weights of different experts are very close across classes, while in deep layers they are diverse. That is to say, the experts are more class specific or sensitive to high-level features, which suggests that there is no significant performance improvement if the CondConv layer is applied near the input of the network. In particular, we also find that the examples from the similar activities such as StandingUpFL and StandingUpFS tend to follow very close distribution.

Next, to demonstrate the superiority of our method, we use the CondConv to compute the confusion matrices on the UNIMIB-SHAR dataset. As can be seen in Fig.6, for the similar activities such as StandingUpFL and StandingUpFS, the baseline CNN made 31 errors, while the CondConv in case of $n = 8$ misclassified only 17 activities. Though the experts activated by the similar activities follow almost the same distribution, their combination is still able to offer better results, which indicates that multiple experts are often

more useful than one. The CondConv is able to enhance the expression ability of CNN by a large margin via increasing the number of experts.

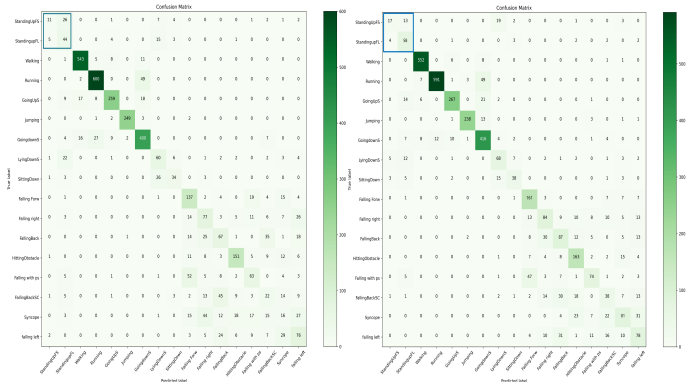


Fig. 6: Confusion matrix for UNIMIB-SHAR dataset using the CondConv with n=1 and n=8 from left to right.

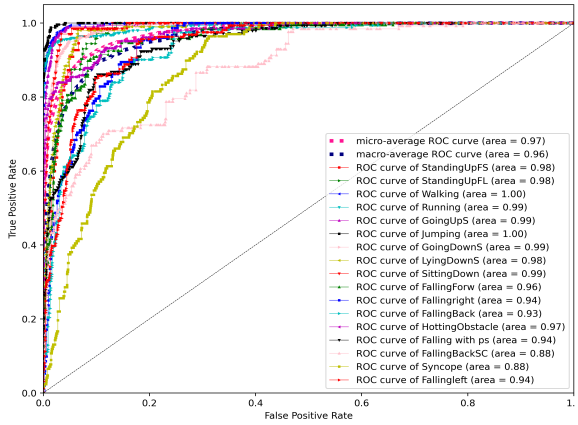


Fig. 7: Roc curves on UNIMIB-SHAR Dataset

As shown in Fig.7, the ROC curve is utilized to evaluate the performance for various activities on the UNIMIB-SHAR dataset, which consists of 9 different types of ADLs and 8 different types of falls. The lines with different colors and symbols represent the area under curve (AUC) values. We also summarize the macro-average and micro-average performance of the proposed method among all the activities. The average AUC value is 0.96 and 0.97 respectively. These ROC curves suggest that distinguishing among falls is very complicated. For example, the most misclassified falls are Syncope, Falling backward-sitting-chair, Falling with protection strategies, Falling rightward and Falling leftward, etc. Among all activities, recognition based on no fall activities obtains better performance, while recognition based on various fall activities has the worst performance. This is mainly because the no fall activities may result in more distinctive features, while there is a fewer number of explicit features between various fall activities to be distinguished.

After that, we further evaluate the performance of the proposed CondConv method via using a 10-fold cross-validation protocol . In 10-fold cross-validation, the original 70% training set is divided randomly into 10 parts, where each part is held out in turn and the training is performed on the remaining

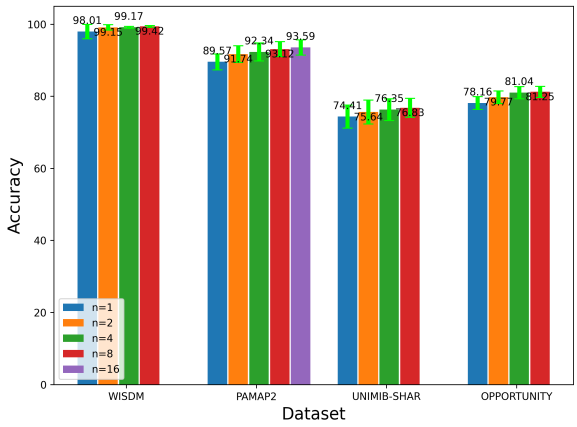


Fig. 8: 10-fold cross validation result

nine-tenths. The hyperparameters are tuned according to the accuracy calculated on the holdout set. Thus, the learning procedure is executed a total of 10 times on different training sets. As a result, an overall accuracy estimate can be achieved via averaging the 10 accuracy estimates. Fig.8 shows the mean accuracy and standard deviation of the methods when evaluated on different datasets. It can be observed that more experts can further improve the baseline CNN due to the increase of model capacity. The CondConv method with n=8 reports the highest mean accuracy when compared to the other smaller expert number.

Keeping other hyperparameters as the fixed settings, we evaluate the sensitivity of the performance by changing the number of convolution layers and kernel size. In the following, the effect of the varying number of convolution layers is evaluated on OPPORTUNITY dataset. Results are shown in Fig.9 (left). It can be seen that the classification results first increase and then decrease as the number of layers increases. The best accuracy is obtained when setting the number of layers to 5. In this case, the classification accuracy is increased from 90.58% to 92.76% on OPPORTUNITY dataset. We also evaluate the sensitivity of kernel size with the values 5*1, 5*3, 5*5, 5*7. In Fig.9 (right), the obvious trend shows that the classification results still first evolve and then decreases as the kernel size continues increasing. It shows that this small kernel is more beneficial for the convolutional network to learn.

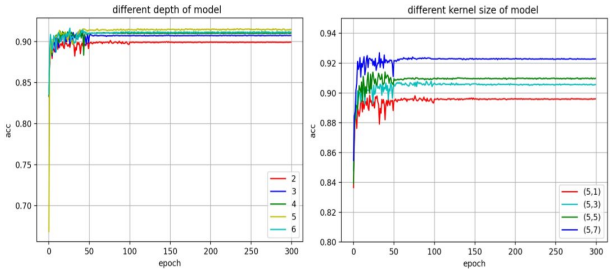


Fig. 9: The sensitivity of the performance by change the number of convolution layers(left) and kernel size(right).

In order to evaluate the superiority of the proposed method, we reproduce the state-of-the-art DeepConvLSTM accchitecture using our parameter setting (code is available from https://github.com/STRCWearlab/DeepConvLSTM). The standard convolution in the DeepConvLSTM is replaced with the

proposed CondConv module. In the paper by Ordóñez *et al.* [19], the final results are reported in terms of F1 score, where the window length is set to 500 ms, with a step length of 250 ms. The window length in [19] is smaller, which is more suitable for LSTM to capture temporal relationship between adjacent windows. For fair comparisons, we perform the DeepConvLSTM on the preprocessed dataset. As a result, the test accuracy obtained by the baseline DeepConvLSTM is 91.28%. When the number of experts n is set to 8, the test accuracy is increased from 91.28% to 92.96%, with a small increase in computational cost. The experimental results show a good generality ability of the proposed method.

TABLE VII: Performance of Different Experts for **OPPORTUNITY** dataset

Model	Test Acc	FLOPs
DeepCondConvLSTM(with n=1)	91.28%	4.68M
DeepCondConvLSTM(with n=2)	91.72%	4.87M
DeepCondConvLSTM(with n=4)	92.63%	5.16M
DeepCondConvLSTM(with n=8)	**92.96**%	6.03M

Next, we evaluate the distribution of routing weights activated by all the examples in the UNIMIB-SHAR test set in the final CondConv layer. Actually, we do not need to normalize all the weights via a softmax. Instead, the Sigmoid function is used to compute the routing function. As a result, Sigmoid functions most often outputs a value in the range 0 to 1. The distribution of the weights can be seen in Fig.10, which shows that all the weights change between 0 and 1. Comparing Sigmoid and Softmax, we find that the former significantly outperforms the latter, which agrees well with Yang *et al's.* results [13]. The main purpose of this evaluation is to disentangle the influence of different experts at deeper layers. Fig.10 shows the routing weights follow a bi-modal distribution, and most of them approximately equal to 0 or 1. Without using any L_1 regularization technique, most experts are sparsely activated. That is to say, for each individual example, only a small portion of the entire network is activated, which suggests an explanation why the CondConv is able to realize efficient inference with larger model capacity.

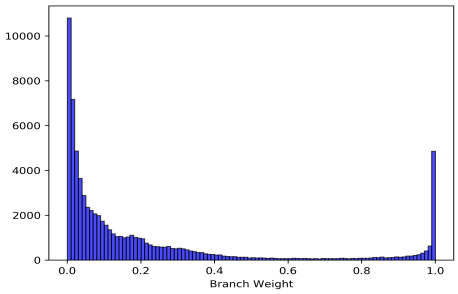


Fig. 10: Distribution of routing weights in the final CondConv layer. The y-axis represents the number of the routing weights for a given value between 0 and 1, which is generated by Sigmoid function. The distribution is evaluated on all activity examples in the UNIMIB-SHAR test set when there are 16 experts. All routing weights follow a bi-modal distribution.

We then study the variation of routing weights within one class in the final CondConv layer. Results are shown in Fig.11. We find that even within one class the routing weights between examples show much higher variance. In addition, to gain a better understanding of experts in the final CondConv layer, we visualize several typical activity examples of top 4 classes with highest activated values on 8 difference experts, as shown in Fig.12.

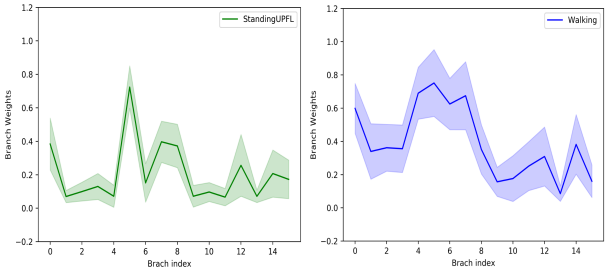


Fig. 11: Routing weights in the final CondConv layer in our model for 2 classes averaged across UNIMIB-SHAR test set. Error bars indicate one standard deviation.

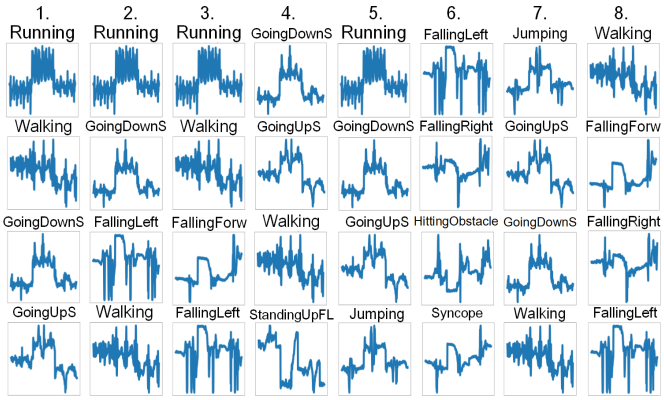


Fig. 12: Each subgraph represents one time-series activity example. The x-axis represents time, and the y-axis represents the average value of three acceleration signals collected in the UNIMIB-SHAR dataset. In this figure, from top to bottom, we could see that most experts are activatied by Running due to the imbalanced dataset in which Running accounts for a large fraction of all 17 categories. The fourth expert is more specific to GoingUp and GoingDown and the sixth expert is most activated by FallingLeft and FallingRight.

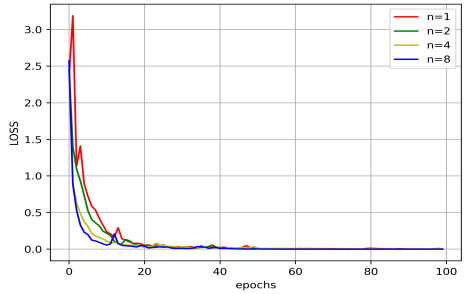


Fig. 13: Training loss on WISDM dataset

From the results in Fig.13, it can be seen that the training time does not increase linearly. The training process converges

faster, and there a faster drop in training loss if there are more experts.

Finally, we evaluate the actual inference time of the CondConv models on a smartphone. On the one hand, the Eq.2 and Eq.5 are mathematically equivalent, which can be formulated as:

$$(\alpha_1 \cdot W_1 + .. + \alpha_n \cdot W_n) * X = \alpha_1 \cdot (W_1 * x) + .. + \alpha_n \cdot (W_n * x) \tag{6}$$

When there are multiple experts, the latter is more computationally expensive. As a comparison, the former (i.e., the CondConv method) only requires one expensive convolution operation, which has lower computational complexity during inference. On the other hand, as shown in Fig.10, routing weights follow a bi-modal distribution. Most routing weights stay close to 0 or 1. That is to say, most experts are sparsely activated even without any regularization, which is able to make the inference process faster. The open source APP introduced in [51] is directly utilized for the evaluation, which is a smartphone-based application for mobile HAR. A screenshot of the APP's user interface is shown in Fig.14. The CondConv models with $n = 1$ and $n = 8$ are trained on the WISDM dataset. We convert the models into .pb file, which are deployed to build an Android application. Our experiment is implemented on a Huawei Mate 30 device with the Android OS(10.0.0). Without loss of generality, we evaluate the actual implementation over WISDM dataset, where a 10-second window with an 95% overlap rate is slide over real sensor readings to generate one sample. As a consequence, the sliding length is identical to 500ms, and the recognition system will wait for 500ms to read and predict next sample. In other words, the system is triggered by scheduled interruptions every 500ms. As shown in Table VIII, it can be seen that the CondConv with $n = 8$ has almost the same inference speed with baseline in the actual implementation, which is far below 500ms. Thus, the proposed method can meet real-time requirement in the actual implementation.

Fig. 14: Screenshot of the APP's user interface

TABLE VIII: Inference time between Conv and CondConv

Model	Inference Time(ms/window)
CNN(Baseline)	228-272ms
CondConv(n=8)	241-292ms

V. Conclusion

Recently, deep CNNs have achieved state-of-the-art performance on various mobile and wearable HAR tasks. However, this technique is severely hampered by the computation power in current mobile and wearable devices. A high number of computations in deep leaning increases computational time and is not suitable for real-time HAR on mobile and wearable devices. Shallow and conventional machine learning methods could not achieve good performance. Therefore, deep learning methods that can balance the trade-off between accuracy and computation cost is highly needed. In this paper, we have presented an efficient solution for HAR on mobile and wearable devices via replacing conventional convolutions with CondConv. The proposed CondConv method is evaluated in on four public HAR benchmark datasets, WISDM dataset, PAMAP2 dataset, UNIMIB-SHAR dataset, and OPPORTUNITY dataset, achieving state-of-the-art accuracy without compromising inference speed. We have also performed various ablation experiments to show how such a larger network is clearly preferable to the baseline while requiring a similar amount of operations. On the whole, with efficient regulation, the proposed method can greatly improve recognition accuracy of the existing HAR using CNN without compromising computation cost, which is very suitable for HAR that has strict latency constrains. By combining the efficient architecture design with any existing CNN based HAR method, we are able to perform real-time HAR tasks on mobile and wearable devices.

References

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

[2] F. Demrozi, G. Pravadelli, A. Bihorac, and P. Rashidi, "Human activity recognition using inertial, physiological and environmental sensors: a comprehensive survey," *arXiv preprint arXiv:2004.08821*, 2020.

[3] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[4] 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 *International workshop on ambient assisted living*. Springer, 2012, pp. 216–223.

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

[6] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[8] J. Yu, L. Yang, N. Xu, J. Yang, and T. Huang, "Slimmable neural networks," *arXiv preprint arXiv:1812.08928*, 2018.

[9] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[10] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, pp. 6474–6499, 2014.

[11] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[12] K. Cho and Y. Bengio, "Exponentially increasing the capacity-to-computation ratio for conditional computation in deep learning," *arXiv preprint arXiv:1406.7362*, 2014.

[13] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "Condconv: Conditionally parameterized convolutions for efficient inference," in *Advances in Neural Information Processing Systems*, 2019, pp. 1305–1316.

[14] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[15] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th International Symposium on Wearable Computers*. IEEE, 2012, pp. 108–109.

[16] D. Micucci, M. Mobilio, and P. Napoletano, "Unimib shar: A dataset for human activity recognition using acceleration data from smartphones," *Applied Sciences*, vol. 7, no. 10, p. 1101, 2017.

[17] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. d. R. Millán, and D. Roggen, "The opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 2033–2042, 2013.

[18] 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 *6th International Conference on Mobile Computing, Applications and Services*. IEEE, 2014, pp. 197–205.

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

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

[21] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Applied Soft Computing*, vol. 62, pp. 915–922, 2018.

[22] 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 Journal*, vol. PP, pp. 1–1, 03 2020.

[23] Y. Guo, Y. Chu, B. Jiao, J. Cheng, Z. Yu, N. Cui, and L. Ma, "Evolutionary dual-ensemble class imbalance learning for human activity recognition," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2021.

[24] J. Cheng, L. Ye, Y. Guo, J. Zhang, and H. An, "Ground crack recognition based on fully convolutional network with multi-scale input," *IEEE Access*, vol. 8, pp. 53 034–53 048, 2020.

[25] S. Anwar, K. Hwang, and W. Sung, "Structured pruning of deep convolutional neural networks," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 13, no. 3, pp. 1–18, 2017.

[26] C. Zhao, B. Ni, J. Zhang, Q. Zhao, W. Zhang, and Q. Tian, "Variational convolutional neural network pruning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2780–2789.

[27] Q. Jin, L. Yang, and Z. Liao, "Towards efficient training for neural network quantization," *arXiv preprint arXiv:1912.10207*, 2019.

[28] J. Yang, X. Shen, J. Xing, X. Tian, H. Li, B. Deng, J. Huang, and X.-s. Hua, "Quantization networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7308–7316.

[29] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6655–6659.

[30] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[31] Z. Wu, T. Nagarajan, A. Kumar, S. Rennie, L. S. Davis, K. Grauman, and R. Feris, "Blockdrop: Dynamic inference paths in residual networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[32] R. Teja Mullapudi, W. R. Mark, N. Shazeer, and K. Fatahalian, "Hydranets: Specialized dynamic architectures for efficient inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8080–8089.

[33] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.

[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[35] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315–323.

[36] X. Li, S. Chen, X. Hu, and J. Yang, "Understanding the disharmony between dropout and batch normalization by variance shift," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2682–2690.

[37] Z. Xiao, X. Xu, H. Xing, F. Song, X. Wang, and B. Zhao, "A federated learning system with enhanced feature extraction for human activity recognition," *Knowledge-Based Systems*, vol. 229, p. 107338, 2021.

[38] F. M. Noori, M. Riegler, M. Z. Uddin, and J. Torresen, "Human activity recognition from multiple sensors data using multi-fusion representations and cnns," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 2, pp. 1–19, 2020.

[39] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," in *2016 IEEE 13th international conference on wearable and implantable body sensor networks (BSN)*. IEEE, 2016, pp. 71–76.

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

[41] Z. Yang, O. I. Raymond, C. Zhang, Y. Wan, and J. Long, "Dfternet: Towards 2-bit dynamic fusion networks for accurate human activity recognition," *IEEE Access*, vol. 6, pp. 56 750–56 764, 2018.

[42] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, "Deep learning models for real-time human activity recognition with smartphones," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 743–755, 2020.

[43] M. Zeng, H. Gao, T. Yu, O. J. Mengshoel, H. Langseth, I. Lane, and X. Liu, "Understanding and improving recurrent networks for human activity recognition by continuous attention," in *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, 2018, pp. 56–63.

[44] K. Chen, L. Yao, D. Zhang, B. Guo, and Z. Yu, "Multi-agent attentional activity recognition," *arXiv preprint arXiv:1905.08948*, 2019.

[45] F. Li, K. Shirahama, M. A. Nisar, L. Köping, and M. Grzegorzek, "Comparison of feature learning methods for human activity recognition using wearable sensors," *Sensors*, vol. 18, no. 2, p. 679, 2018.

[46] J. Long, W. Sun, Z. Yang, O. I. Raymond, and B. Li, "Dual residual network for accurate human activity recognition," *arXiv preprint arXiv:1903.05359*, 2019.

[47] I. De Falco, G. De Pietro, and G. Sannino, "Evaluation of artificial intelligence techniques for the classification of different activities of daily living and falls," *Neural Computing and Applications*, vol. 32, no. 3, pp. 747–758, 2020.

[48] S. S. Alia, P. Lago, and S. Inoue, "Mcomat: a new performance metric for imbalanced multi-layer activity recognition dataset," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, 2020, pp. 232–237.

[49] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *arXiv preprint arXiv:1604.08880*, 2016.

[50] Y. Guan and T. Plötz, "Ensembles of deep lstm learners for activity recognition using wearables," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1–28, 2017.

[51] D. Singh, E. Merdivan, I. Psychoula, J. Kropf, S. Hanke, M. Geist, and A. Holzinger, "Human activity recognition using recurrent neural networks," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, 2017, pp. 267–274.

Xin Cheng received the B.S. degree from YanTai University of measurement and control technology and instruments, YanTai, China, in 2019. He is currently pursuing the M.S. degree with Nanjing Normal University. His research interests include activity recognition, computer vision, and machine learning.

Lei Zhang received the B.S. degree in computer science from Zhengzhou Universitiy, China, and the M.S. degree in pattern recognition and intelligent system from Chinese Academy of Sciences, China, received the Ph.D. degree from Southeast University,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. His research interests include machine learning, human activity recognition and computer vision.

Yin Tang received the B.S. degree from Hunan University of Engineering, Xiangtan, China, in 2018. He is currently pursuing the M.S. degree with Nanjing Normal University. His research interests include activity recognition, computer vision, and machine learning.

Yue Liu is currently pursuing the B.S. degree with Nanjing Normal University. Her research interests include activity recognition, computer vision, and machine learning.

Hao Wu received the Ph.D. degree in computer science from Huazhong University of Science and Technology, Wuhan, China, in 2007. Now, he is an associate professor at School of Information Science and Engineering, Yunnan University, China. He has published more than 50 papers in peer-reviewed international journals and conferences. He has also served as reviewers and PC members for many venues. His research interests include natural language processing, recommender systems and service computing.

Jun He received the Ph.D. degree from Southeast University, Nanjing, China, in 2009. He was a Research Fellow with IPAM, UCLA, in 2008 and a Post-Doctoral Research Associate with the Chinese University of HongKong from 2010 to 2011. He is currently an associate professor with the school of Electronic and Information Engineering, Nanjing University of Information Science and Technology. His research interests include machine learning, computer vision and optimization methods.