PatchHAR: A MLP-like architecture for efficient activity recognition using wearables

Shuoyuan Wang, Lei Zhang, Xing Wang, Wenbo Huang, Hao Wu, and Aiguo Song, *Senior Member, IEEE*

Abstract—To date, convolutional neural networks have played a dominant role in sensor-based human activity recognition (HAR) scenarios. In 2021, researchers from four institutions almost simultaneously released their newest work to arXiv.org, where each of them independently presented new network architectures mainly consisting of linear layers. This arouses a heated debate whether the current research hotspot in deep learning architectures is returning to MLPs. Inspired by the recent success achieved by MLPs, in this paper, we first propose a lightweight network architecture called all-MLP for HAR, which is entirely built on MLP layers with a gating unit. By dividing multi-channel sensor time series into nonoverlapping patches, all linear layers directly process sensor patches to automatically extract local features, which is able to effectively reduce computational cost. Compared with convolutional architectures, it takes fewer FLOPs and parameters but achieves comparable classification score on WISDM, OPPORTUNITY, PAMAP2 and USC-HAD HAR benchmarks. The additional benefit is that all involved computations are matrix multiplication, which can be readily optimized with popular deep learning libraries. This advantage can promote practical HAR deployment in wearable devices. Finally, we evaluate the actual operation of all-MLP model on a Raspberry Pi platform for real-world human activity recognition simulation. We conclude that the new architecture is not a simple reuse of traditional MLPs in HAR scenario, but is a significant advance over them.

Index Terms—Human Activity Recognition, Deep Learning, all-MLP, Wearable Sensors

1 Introduction

Over the past few decades, with the vast proliferation of Internet of Things (IoT) technology, sensor-based Human Activity Recognition (HAR) has drawn favorable attention and become an active research area. Human activity signals can be collected and analyzed from different sensor modalities (e.g., inertial sensors), which can provide a smart decision on embedded devices [1]. Due to exceptional advantages in sensing devices such as lower cost, higher accuracy, and smaller size, HAR has a wide range of applications in Ambient Assisted Living (AAL), motion tracking, elderly health assessment, and Human–Machine Interaction (HMI) [2], [3], which have greatly improved the quality of life and the healthcare of the elderly or other dependent people.

Traditional machine learning (ML) algorithms such as Random Forest, naive Bayes, and Support Vector Machine (SVM) have achieved competitive results in HAR area. However, they are usually constrained by complex feature engineering involving specific expert experience or domain knowledge, where designed statistical features have to be manually extracted from raw sensor data. Thus, conventional ML algorithms using handcrafted features are not very suitable for HAR since complex human activities usually contain highly abstracted semantics. Recently, the emergence of deep learning and increased computational power have provided an effective solution in HAR scenario [4]. Artificial intelligence methods with automatic feature extraction have been adopted for classifying complex human activities [5]. Through automatically capturing discriminative feature representations, Convolutional Neural Networks (CNNs) have gained strong results on various standard HAR benchmarks and turned to the current state-of-the-art, which is widely utilized for activity recognition [6], [7].

In another line of research, almost at the same time, four research groups located in different institutions (e.g., Google, Facebook, Tsinghua University, and Oxford University) released their newest work in the first week of May 2021 [8], [9], [10], [11]. All of them present novel network architectures mainly consisting of linear layers, which perform comparably well or even better than convolutional-based architectures. As is universally known, static-parameterized multi-layer perceptron (MLP) with a huge function space can fit arbitrary functions in theory [12]. Current mainstream deep learning architectures in computer vision (CV) and natural language processing (NLP) areas are returning to MLPs. This research progress immediately sparks related debate and discussion: Whether pure MLPs are sufficient? However, on ImageNet, their current accuracies still maintain 5–10% lower than those reported by the best CNNs or Transformer networks. Thus, it still needs deeper research to explore their potential to the greatest extent.

Actually, highly miniaturized wearable devices often have a very limited computational budget. Real HAR applications typically pursue the best accuracy under a resource-

- *Shuoyuan Wang (E-mail: claytonwang0205@gmail.com), Lei Zhang (E-mail: leizhang@njnu.edu.cn) and Xing Wang (E-mail:chaunceywx@gmail.com) are with the School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing, 210023, China.*
- *Wenbo Huang (E-mail: wenbohuang1002@outlook.com) is with the School of Computer Science and Engineering, Southeast University, Nanjing, 210096, China.*
- *Hao Wu (E-mail: haowu@ynu.edu.cn) is with the School of Information Science and Engineering, Yunnan University, Kunming, 650500, China.*
- *Aiguo Song (E-mail: a.g.song@seu.edu.cn) is with the School of Instrument Science and Engineering, Southeast University, Nanjing, 210096, China.*
- *Corresponding author: Lei Zhang*

constrained platform, where an accuracy/speed trade-off should be preferably considered. A natural idea arises: in order to develop a lightweight HAR model, can we solely exploit MLP architectures? In this paper, taking inspiration from recent MLP research, we investigate the necessity of convolution in key HAR applications and propose a novel MLP-based alternative to CNN. The overview of the proposed all-MLP model is shown in Fig.1. The all-MLP model is stacked by a set of MLP layers with a simple linear gate unit, which implements channel projection and spatial projection respectively. To the best of our knowledge, the all-MLP is the first HAR model relying entirely on MLPs without any convolutional architecture. Moreover, the all-MLP architecture is entirely based on matrix multiplication, which is similar to plain convolution and can be easily optimized using popular deep learning libraries [13]. Despite the radically new design, all-MLP obtains strong results on several activity recognition benchmarks, including WISDM, OPPORTUNITY, PAMAP2 and USC-HAD. Through linear transformation and gating operation, it can significantly surpass CNNs meanwhile greatly reducing memory and computational overhead. The main contributions of this paper are summarized as follows:

• Without using convolution, we propose a radically new architecture entirely relying on MLP layers with a simple linear gating for HAR task. Due to the simplicity of these MLP architectures, our model can easily handle time series sensor data.

• Extensive experiments are conducted on several public HAR datasets, which indicate that our all-MLP model can achieve competitive results at smaller memory and computational cost. Several key hyper-parameters are analyzed in detail.

• All computations only involve in matrix multiplication, which can be easily optimized with mainstream deep learning libraries. This advantage can promote practical HAR deployment in wearable devices. We also examine actual speedup on a Raspberry Pi platform with an ARM-based computing core. The all-MLP model achieves $\sim 4\times$ actual speedup over CNNs meanwhile maintaining comparable accuracy.

The remainder of this paper is organized as follows: In Section 2, we review several representative convolutional architectures in HAR research and recent MLP backbones. We introduce our proposed MLP-based model in detail in Section 3. The benchmark datasets, experiment setups, and our main results are introduced in Section 4. We further evaluate several key hyperparameters by conducting ablation experiments in Section 5. Finally, we conclude the paper in Section 6.

2 Related works

Deep learning for HAR: As an alternative to shallow ML approaches, CNNs have become a mainstream deep learning technique for classifying human activities from raw sensor data, which have provided state-of-the-art performance over a large variety of HAR tasks [14]. [15] first adopted CNNs in HAR scenario to automatically capture local invariant features of time series from the accelerometer. [16] further presented the unique advantage of CNNs via learning hierarchical feature representation from inertial sensors under supervised HAR scenario. [17] proposed a deep framework by combining automatic features extracted by CNN and shallow handcrafted features, which is applied for real-time execution on wearable devices. [18] further improved the performance of CNNs by replacing cross-entropy loss with local supervised loss, which is very beneficial for memory reuse. [19] investigated the influence of hyperparameters for different HAR tasks, which provides detailed guidelines for the reseachers who aim to adopt deep learning in their problem setting. [20] presented a novel network architecture called DeepConvLSTM by inserting LSTM layers into normal CNN, which shows obvious advantages in extracting temporal features from raw sensor time series. [21] introduced a new framework called AttenSense, which combines attention module with a CNN and a Gated Recurrent Unit to learn the dependencies of sensor signals across both spatial and temporal domains. However, current deep HAR research mainly focuses on convolutional structure. In this paper, our main research motivative is to explore the potentiality of MLPs in HAR scenarios to the greatest extent.

MLP backbones: Current mainstream deep learning researches are returning to pure MLPs. Taking an idea of data pre-processing in Vision Transformer (ViT), [8] first released a new MLP backbone called MLP-Mixer, which can provide interaction between channels (channel-mixing) and patches (token-mixing). [9] presented a similar network architecture, which produces competitive results compared with state-of-the-art in CV area. [10] utilized a gating unit to reinforce patch communications, which is superior to MLP-mixer. [11] proposed an External-Attention block, which is built via solely using two MLP layers. [22] introduced a circulant channel-specific structure to generate a larger reception field, which is consistently able to provide higher recognition accuracy with fewer parameters. [13] reviewed recent MLP works and summarized their advantages, which indicates that pure MLPs can be readily deployed on resource-constrained hardware and reduce energy consumption. To the best of our knowledge, pure MLP architectures have been rarely exploited in HAR scenario. In this paper, we first propose a lightweight network architecture which is entirely built on MLP layers for HAR tasks.

3 Model

3.1 Architecture Overview

In this section, we introduce our MLP-based activity recognition approach. As we know, convolutional neural networks (CNNs) have demonstrated a competent ability to simultaneously capture local dependencies over both different time stamps and sensor modalities across multimodal sensor data, where all the modalities are considered in each time stamp, such that the translating invariance introduced by local filters leads to accurate recognition. In our work, following the same setting [23], [24], we still vertically stacked all axes of sensor signals to form 2D matrices. Using a sliding window with a fixed length and a specific overlap percentage, we first divide the raw sensor time series into continuous samples. As illustrated in Fig.1, the overall network architecture is entirely built on MLP layers. The

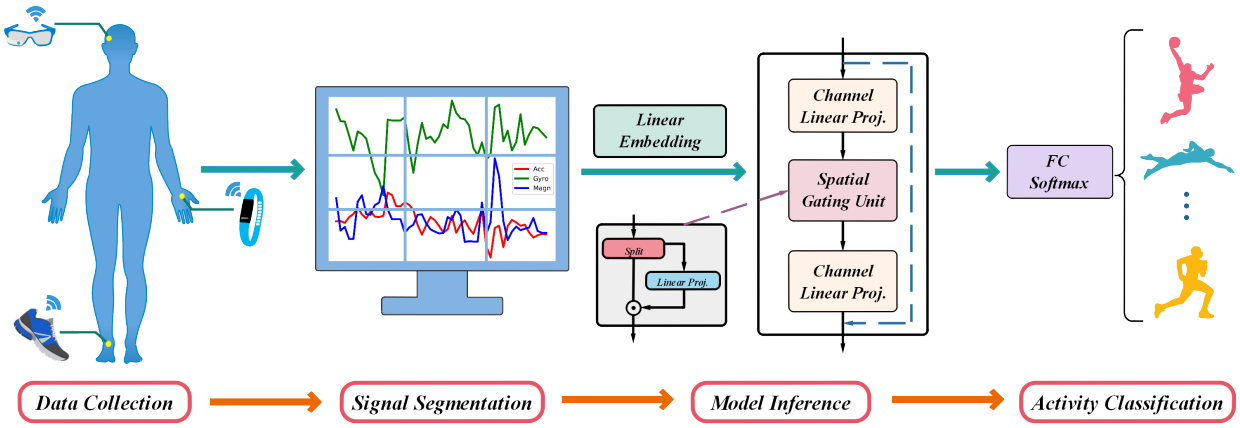


Fig. 1. Overview of the MLP-based model for HAR. The curves visualized on the computer screen are raw multi-channel sensor time series that are further split into patches.

proposed network is mainly composed of a stack of linear layers including channel-projection and spatial gating unit. Different from CNN, the all-MLP model receives a sequence of linearly projected patches which is split from the above multi-channel sensor samples. For brevity, we omit the activation and normalization. Finally, the final classification results can be obtained through a fully-connected layer. In the next subsection, we will present the details of our all-MLP model.

3.2 Patch Embedding

Our data pre-processing stage is different from normal CNNs which allows an overlap between adjacent windows to preserve the continuity of activities. As shown in Figure 1 and Figure 2, similar to recent mainstream MLP backbones, our all-MLP takes an input of N non-overlapping patches which is split from these activity windows. Specifically, the activity windows are constructed like image data, containing the two dimensions corresponding to time-steps and heterogeneous sensor modalities. It should be noted that the patches will be sent to the model in turn, which is different from traditional stacked MLPs. The activity label of each window will be assigned according to the majority voting in the corresponding samples that constitute the window. To utilize MLP architectures, we assume that $X \in R^{C\times H\times W}$ is an input tensor corresponding to an activity window, where H and W represent the number of heterogeneous sensor modalities and time-steps respectively. Here C denotes the number of channels in the input tensor. As a result, we crop the raw input into non-overlapping patches $P = \{p_0, p_1 \dots p_n\}$. Here n is equal to $\frac{HW}{h_p w_p}$, where h_p and w_p represent the heigh and width of every patch respectively. Each patch will be further unfolded into a one-dimension feature vector, whose length is equal to $C \times h_p \times w_p$. Finally, all the patches will be further patch-wisely mapped to a predefined embedding dimension via a shared-weight fully-connected layer:

$$X_i \leftarrow W_p P_i + b_p, \tag{1}$$

where $W_p \in R^{d\times s}$ and $b_p \in R^d$ are the weights and bias of the linear patch embedding layer. Figure 2 illustrates the linear embedding module in detail. All the numbers and variables will be detailed in the following experiment part.

3.3 all-MLP Block

The proposed network consists of a set of MLP layers, which also uses standard components such as residual connection [25] and layer normalization [26]. The input can be represented as $X \in R^{n\times d}$, where n is the number of patches and d is the embedded dimension. The block can be formulated as follows:

$$U = Norm(\sigma(f_1(X))), \tag{2}$$

$$Z{=}s(U), \tag{3}$$

$$Y = X + Norm(f_2(Z)), \tag{4}$$

where σ refers to Gaussian error linear units (GELUs) [27] activation and $Norm(\cdot)$ represents layer normalization operation. f_1 and f_2 denote channel-projection which follows nomenclature in MLP-mixer [8] and can be treated as 1×1 convolution operation for channel expansion. A critical component in the block is $s(\cdot)$ [10], which is a gating unit to provide communication between patches. s is inspired from Gated linear units (GLU) [28] and can achieve long-range spatial interactions than token-mixing [8] meanwhile reducing computational cost.

3.4 Spatial Gating Unit

Similar to convolutional filters in CNNs, to capture information over spatial dimension, the most practical idea in the all-MLP architecture is to use plain linear projection [8], [9]. The schematic diagram of the spatial projection layer is shown in Figure 3. Assuming the same input $X \in R^{n\times d}$ as indicated above, we can formulate the operation as:

$$f_T(X) = WX^T + b, \tag{5}$$

where $W \in R^{n\times n}$ and $b \in R^n$ are the parameters of linear projection. Spatial Gating Unit (SGU) [10] is introduced in

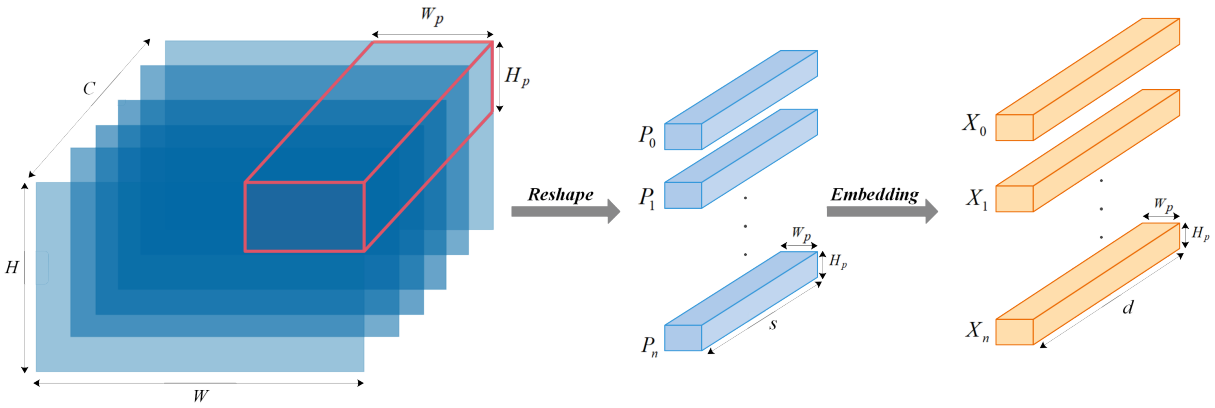


Fig. 2. Visualization of patch embedding process.

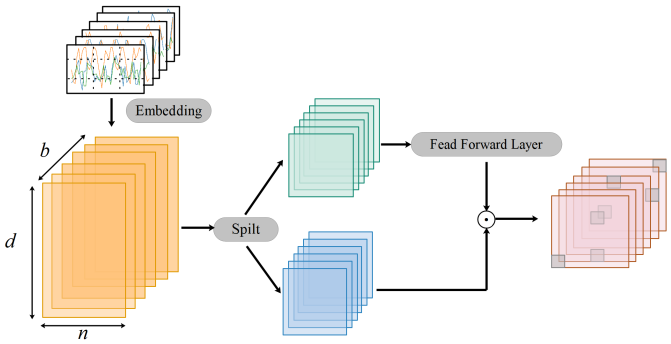


Fig. 3. The schematic diagram of the proposed spatial gate unit that introduces efficient spatial interactions across patches. Here b, d, n represent the batch size, channel dimension, and the number of patches respectively. The interactions in time-series and sensor modalities across patches will be captured SGU automatically by SGU. The gray part of the output represents more important patches.

order to better extract spatial features. Specifically, we regulate the multiplication between linear-transformed input and its initial input as the output of SGU. The procedure of SGU can be expressed as follows:

$$s(X) = X \otimes f(Norm(X)), \tag{6}$$

where $\otimes$ is Hadamard product operator and f denotes linear layers. In practice, Spatial Gating Unit (SGU), i.e., $s(.)$ is a key ingredient in merging modality-specific features across patches, which adopts such element-wise multiplication to regulate the level of contribution of the patch. Specifically, to ensure training stability, we set the initial values of W and b to be close to zero and one respectively, which indicates that $f(x) \approx 1$ and thus $s(x) \approx x$ in the aforementioned formulation, i.e., Eq. 6. Such an initialization makes the branch of SGU initially to be an identity mapping, which enables SGU to gradually capture modality-specific information across patches. That is to say, $s(.)$ can be viewed as an identity mapping at an early stage of training, where all individual patches will be processed independently without any cross-patch communication. Our all-MLP model can behave as a standard feed-forward network at the first few epochs of training [10], [29]. $s(.)$only gradually injects modality-specific information across patches during learning. For the projection branch in SGU, each row in W can be seen as a query for each patch, which is in charge of learning global features via projection branch $f_T(X) = X^T + b$. After that, the element-wise multiplication is performed between the output of $f_T(X)$ and original X to capture cross-patch dependencies across multi-modal sensor data. Overall, SGU provides an alternative mechanism other than self-attention, which is similar to Squeeze-and-Excite (SE) block [29] in terms of element-wise multiplication. However, different from SE, SGU is computed based on a projection over the spatial (cross-patch) dimension rather than the channel dimension.

The computational complexity of SGU without split is equivalent to n^2d, which is computationally heavy. Due to the limited computing power in wearable devices, we continue to divide it into two independent parts evenly along channel dimensions. As the channel dimension is further decreased to half of the original input, the computational cost can be significantly reduced. The modified SGU can be formulated as:

$$s(X) = X_1 \otimes f(Norm(X_2)). \tag{7}$$

Here both X_1 and X_2 are two parts of the X and their relation can be represented as $X = X_1 \oplus X_2$, where $\oplus$ is referred to as matrices merging operation. As the channel dimension decreases to half of the input, the computational cost can be significantly reduced.

4 Experiment

In order to evaluate the effectiveness and efficiency of all-MLP models against representative convolutional networks, we conduct our experiments on four HAR datasets. The experiments consist of four parts. In part one, the four public HAR datasets used in our evaluation are introduced. In part two, the experiment setup and data pre-processing procedure are presented. In part three, detailed quantitative comparisons are performed across different HAR datasets. Finally, ablation studies are conducted and several key factors are analyzed.

4.1 Datasets Description

We selected four popular HAR datasets including WISDM, OPPORTUNITY, USC-HAD, and PAMAP2 as our benchmarks for evaluation. These datasets contain diverse types of human activities and sensor modalities, which are involved in a large variety of application scenarios. On the whole, they are capable of simulating practical HAR applications for our evaluation.

WISDM dataset [30]: The Wireless Sensor Data Mining (WISDM) Lab built this dataset. Under a supervised condition, 29 volunteers were instructed to perform six diverse types of daily human activities. The smartphone embedded with an accelerometer was placed on each volunteer's front leg for data collection. The sampling frequency is set to 20Hz. The whole dataset contains 10,981 samples and each sample corresponds to 10-second sensor data.

OPPORTUNITY dataset [31]: Under a sensor-rich environment, the dataset was collected by Daniel et al. in a breakfast preparation scenario. Without loss of generality, we utilize the subset from the OPPORTUNITY challenge consisting of unsegmented sensor recordings from 4 subjects. The whole sensor system consists of 72 sensors with 10 different modalities, which are attached to 12 different parts of the human body. Each participant was asked to perform 17 types of breakfast-related activities such as 'Preparing and drinking coffee', 'Preparing and eating sandwich', and 'Cleaning table' for 20 repetitions. The sampling rate is set to 30Hz.

PAMAP2 dataset [32]: Physical Activity Monitoring for Aging People 2 (PAMAP2) dataset consists 18 different types of activities of daily living (ADL) such as 'Nordic walking', 'Rope jumping', 'Vacuum cleaning' etc from 9 volunteers. Each volunteer wore 3 IMUs and a heart-rate-monitor. The sampling rate of all 3 IMUs is 100Hz, which is down-sampled into 33Hz for fair comparison. The heart rate monitor is set to 9Hz for estimating motion intensity. The PAMAP2 dataset has been made publicly available.

USC-HAD dataset [33]: The University of Southern California Human Activity Dataset (USC-HAD) was designed for diverse ML algorithm evaluations, especially for elder care and health monitoring. 14 participants joined in the data collection process. Attaching sensing devices to their front right hip, each participant performed 12 well-defined low-level daily activities including 'Sleeping', 'Walking left', 'Walking right', 'Running forward', 'Elevating up', and 'Elevating down', etc. The sampling rate is set to 100Hz.

4.2 Experimental setup

Because the four selected benchmark HAR datasets contain various human activities in different contexts, the activities under evaluation may be diverse according to duration and complexity. Thus, we introduce several key hyper-parameters such as window length, sampling rate, and the height and width of each patch, which have a potential effect on final activity recognition performance. Moreover, we describe the network architectures, implementation details, and evaluation procedure used in the following experiments.

4.2.1 *Dataset Prepossessing*

Several important properties of these datasets are summarized in Table 1. Time series signals collected from different sensor modalities have to be first segmented into continuous samples by sliding window technique, which are then normalized into zero mean and unit variance via subtracting the mean and dividing by standard deviation. An overlap is allowed between consecutive windows. Actually, the window size and overlap have an important effect on classification performance, which could be variable and rely on specific activity recognition tasks. Because there is no consensus on what are the optimal window size and overlap, we utilize the previous successful segmenting strategy [18], [34], which transforms multi-channel sensor time series into desired input tensors. Referring to recent literature [21], [35], [36], we implement a Fast Fourier Transform to generate windows of physical activities for PAMAP2. First, a linear interpolation is used to handle data loss of raw sensory data. Second, a 5.12-second window is used to divide the sensory data. Third, the Fast Fourier Transform (FFT) is computed, which is in charge of transforming time window from time domain to the frequency domain. As a result, we can have one 256-point representation of frequency spectrum between the 0–50 Hz frequency range. It is well known that the energy of physical activities often concentrates on low-frequency parts. Without loss of generality, the first 120 points of the spectrum is selected, which approximately cover the 0–23 Hz frequency range. That is to say, the window size of frequency is set to 120*1 for PAMAP2 in the frequency domain. In other cases, e.g., WISDM, USC-HAD, OPPORTUNITY, the activity windows still contain the two dimensions that correspond to time-steps and heterogeneous sensor modalities respectively. Moreover, the padding technique is used to evenly divide each window into patches for further use in MLP models.

We detail how the training and test data are partitioned. Actually, splitting the training, validation, and test sets has a huge impact on the final results. It is important to choose the proper training and test sets from the same distribution and it must be taken randomly from all the data. Here we use the *train-test-split()* method from the *sklearn* library to split the data into train, validation and test sets. For fair comparisons, we strictly follow the same split strategy available in recent research literature [19], [20], [36], [37]. For example, the same split strategy [20] in the OPPORTUNITY challenge has been utilized to train and test our models. Specifically, the data of all ADL and drill sessions from subject 1, as well as ADL1, ADL2 and drill sessions from Subjects 2 and 3 is used to train our models. The data of ADL3 from Subjects 2 and 3 is left for validation. The data of ADL4 and ADL5 from Subjects 2 and 3 is held out for the final test. For PAMAP2 dataset, as the studies in [19], [38] have suggested, the recordings from participants 5 and 6 are used as validation and test sets respectively, while the rest data is held out for training. For WISDM dataset, as suggested by [36], [39], 70% data is used to create the training set, and the validation and test sets are produced with the remaining

30% of the data. The reason for leaving out the validation set is to tune the hyper-parameters, which enables the model to perform the best on unseen data. Finally, USC-HAD dataset has a version [37], [40] that is already split into training, validation and test sets as a ratio of 7:1:2 that contain data from different participants, therefore the user-independent strategy is dominant in this case.

4.2.2 *Networks Setup*

In this subsection, the network architecture and training details are introduced in detail. Referring to recent MLP research, we adopt common components in our non-convolutional architecture: Layer Normalization [26], element-wise activation function (GeLU [27]) and skip-connection [25]. In the case of WISDM, PAMAP2 and USC-HAD, the hidden dimension and channel-expanded dimension are set to 256 and 768 respectively in the all-MLP model. For OPPORTUNITY, the channel-expanded dimension is set to 1024, since there is high dimension for the input sensor channel in this dataset. Following current HAR research literature, we build a 3-layer CNN as our baseline, which can provide very close results compared to current representative state-of-the-art CNNs. The 3-layer CNN can be readily deployed in wearable-based HAR applications with limited computing power. The shorthand description of baseline CNN can be presented as $C\,(64) \rightarrow C\,(128) \rightarrow C(256) \rightarrow FC$ or $C\,(128) \rightarrow C\,(256) \rightarrow C(384) \rightarrow FC$ according to different datasets, where the $C(n)$ denotes channel dimension and FC represents full-connected layer for final classification. When aiming for state-of-the-art results for CNNs, researchers often prefer stochastic gradient descent (SGD) with momentum because models trained with Adam have been observed to not generalize as well. However, our MLP architecture can be seen as an evolution from the transformer. it is hard to directly adopt CNN's training recipes to train the MLP, which is due to the incompatibility between SGD and Transformers, as discussed in [41], [42]. Compared with the SGD optimizer, AdamW [43] enables more successful training of transformers, which has been widely applied for MLP architectures. Thus, we choose AdamW in all-MLP and SGD in CNN as our optimizers respectively. The other training hyperparameters are set to be the same. We use smooth cross-entropy [44] as loss function. A piecewise-decay strategy is adopted, where the initial learning rate is set to 5e-4, which decays to 10% every 50 epochs. The number of training epochs is set to 200 and the batch size is 256. All the experiments are implemented with the PyTorch deep learning framework on a server (GPU: 24 GB GeForce RTX 3090; CPU: 6th Gen Intel i7-6850K; RAM: 64 GB).

4.3 Quantitative Comparison

In this part, we conduct quantitative comparisons in terms of accuracy, FLOPs, and the number of parameters. Our all-MLP models are compared with the 3-layer baseline CNN and other representative convolutional networks. The hyper-parameters have been carefully chosen to ensure training stability and prevent overfitting, which will be discussed in further ablation studies. Following recent literature [45], [46], we choose the best epoch result as the final result and compute the average result over ten runs for our evaluation, which proves that the results are stable.

Recognition Performance: Since sensor data for specific activities such as falls of elderly people is particularly hard to collect, it leads to a main challenge called class imbalance for HAR. To provide a more comprehensive evaluation, we exploit both accuracy and weighted F1-score as our performance metrics, which can be formulated as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{8}$$

$$F_{1-Weighted} = \sum_{i=1}^{n_c} 2 * w_i \frac{precision_i \cdot recall_i}{precision_i + recall_i}, \tag{9}$$

where TP, TN, FP, and FN represent true positives, true negatives, and false positives respectively. For a specific activity class i, $precision_i$ is equivalent to $\frac{TP_i}{TP_i+FP_i}$ and $recall_i$ is equivalent to $\frac{TP_i}{TP_i+FN_i}$. w_i denotes the corresponding sample proportion. n_c is referred to the number of activity classes.

The results are shown in Table 2. Test accuracy curves of our all-MLP models and baseline CNNs are illustrated in Figure 4. It can be clearly observed that our all-MLP models can consistently outperform the 3-layer CNNs over all four HAR benchmarks. According to the results of USC-HAD, the proposed model significantly exceeds baseline CNN by 5.04%. In the case of WISDM, OPPORTUNITY and PAMAP2, the all-MLP models also provide an accuracy gain of 1.26%, 0.59% and 0.87% over CNN respectively. We also incorporate the quantitative results in terms of the weighted F1-score in Table 2. In addition, Table 2 compares our model with a few popular models. As can be evidently seen, our all-MLP models perform comparably well or even better than these representative convolutional architectures and their variants.

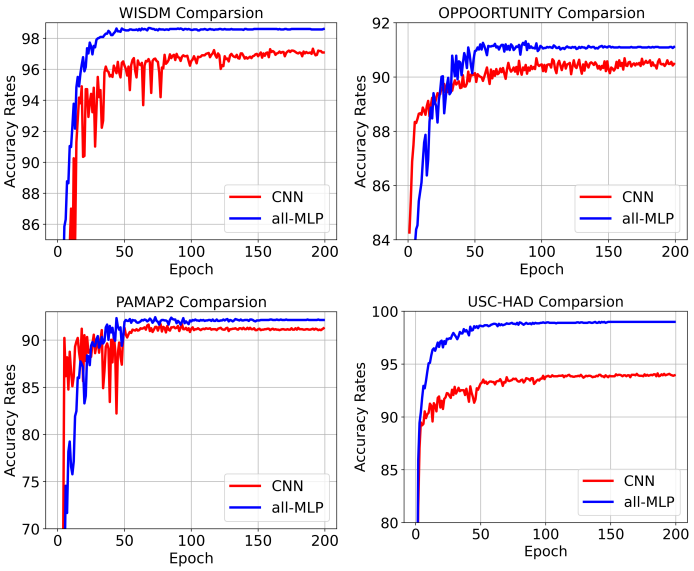


Fig. 4. Test accuracy curves of all-MLP and CNN on four HAR benchmarks.

Computation cost: Due to limited computing power in wearable devices, the inference speed is another key

TABLE 1
Data prepossessing.

Dataset\Attribute	Categories	Sampling Rate	Channel	Window Size	Overlap Rates	Patch Size
WISDM	6	20Hz	1	200x3	50%	10x3
PAMAP2	12	33Hz	86	120x1	78%	15x1
USC-HAD	12	100Hz	1	512x6	50%	64x3
OPPORTUNITY	18	30Hz	1	30x114	50%	6x19

TABLE 2
Performance comparison on public benchmarks.

Benchmarks		WISDM	OPPORTUNITY	PAMAP2	USC-HAD
standard CNNs	Accuracy(%)	97.35±0.38	90.52±0.13	91.26±0.28	93.95±0.14
	F_1-score	97.16±0.42	90.41±0.46	90.81±0.37	93.85±0.23
	Parameters(M)	1.5	0.8	0.9	0.4
	FLOPs(M)	78	40	7.4	78
all-MLP(Ours)	Accuracy(%)	98.61±0.32(↑1.26)	91.11±0.08(↑0.59)	92.13±0.46(↑0.86)	98.99±0.21(↑5.04)
	F_1-score(%)	98.75±0.28(↑1.59)	91.75±0.83(↑1.34)	91.99±0.20(↑1.91)	98.95±0.11(↑5.10)
	Parameters(M)	0.3	0.4	0.6	0.3
	FLOPs(M)	6.3	13	6.9	5.6
Relate Researches		97.2 † [47] 97.51 ⋄ [50] 98.23 † [36] 98.32 ⋄ [54]	88.6 † [48] 91.5 † [20] 91.57 ⋄ [52] 92.06 † [55]	89.96 † [24] 90.33 † [51] 91.0 † [53] 91.66 ⋄ [53]	90.88 † [49] 97.8 † [37] 98.44 † [40] 98.93 ⋄ [23]

†: Test Accuracy. ⋄: Test Weight F_1-score.

evaluation metric in ubiquitous HAR computing scenarios. Table 2 compares FLOPs and the number of parameters on the four HAR datasets. Results show that with comparable classification score, our all-MLP models is much more efficient than convolutional architectures. For WISDM, OPPORTUNITY, PAMAP2, and USC-HAD, our all-MLP models are theoretically $\sim 12.4\times$, $\sim 1.5\times$, $\sim 3.1\times$, $\sim 13.9\times$ faster than CNNs at much smaller memory overhead. We will evaluate the actual runtime in Sec 4.4.5. It is worthwhile to note that the lightweight architecture design makes it suitable for equipping pure MLPs with HAR applications.

4.4 Ablation Study

We conduct extensive ablation studies to independently analyze the impact of each component in our all-MLP model. To be specific, we investigate the influence of several key hyperparameters such as patch size, channel dimension and SGU variants on recognition accuracy and computational complexity. We also visually compare classification performance by computing confusion matrices. Finally, we evaluate actual runtime on an embedded Raspberry Pi 3B+ platform.

4.4.1 ***Model Scale***

While not strictly comparable, this motivates us to further extrapolate the effect of different scales. We compare various configurations of CNN to our MLPs on the four benchmark datasets. Figure 5 summarizes the main results, where we include 1-layer, 2-layer, 3-layer, and 4-layer CNN models with a different number of kernels, e.g., 16, 32, 64, 128,256 and 512. Similarly, we change our all-MLP model size by changing the hidden dimension and channel dimension, whose numbers range from 64, 512 to 256, and 2048 respectively. As the number of layers grows, both MLP's and CNN's performance steadily improves. Although the 1-layer and 2-layer CNNs attain a reasonable accuracy, it tends to overfit as the number of kernels increases. The performance of both 3-layer and 4-layer is obviously superior to that of 1-layer and 2-layer. The figure also reveals that both 3-layer and 4-layer models achieve very similar values of classification accuracy. The performance gap between 3-layer and 4-layer shrinks with model scale, and the relative improvement is almost negligible. In addition, 3-layer CNNs runs much faster than 4-layer ones, which is in good line with recent literature [15], [16], [56]. Considering both accuracy and computational efficiency, we set the number of layers to 3 by default. To differentiate, the all-MLP model results are shown below using a purple dotted line. It can be seen that the classification score firstly increases then decreases, and non-monotonically evolves as the hidden dimension and channel dimension increase. If there is no extra data augmentation, the classification score is prone to saturate or even drop, which is an obvious overfitting phenomenon. In order to better trade off accuracy and speed, we set the hidden dimension and channel dimension to be 256 and 768 respectively on WISDM, PAMAP2, USC-HAD. For OPPORTUNITY, we select 256/1024 for better classification results. More importantly,our all-MLP model can easily outperform CNN-based models in a relatively fair way. Overall, the results support our main claim that in terms of the accuracy-speed trade-off MLP-like architectures are more competitive than conventional convolution-like

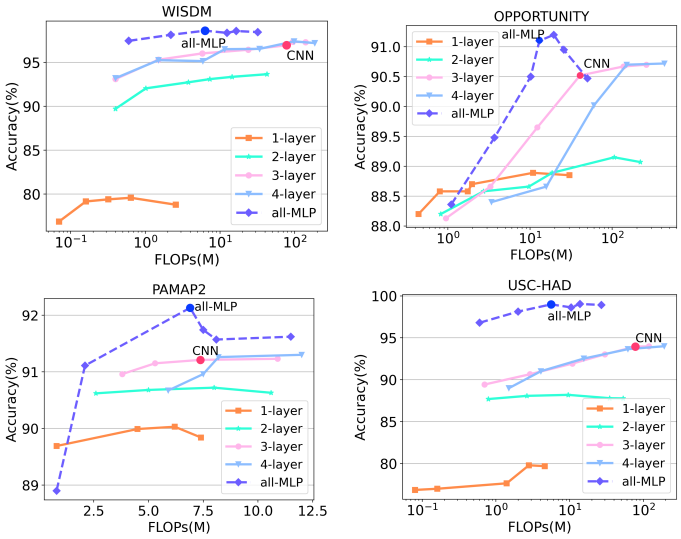


Fig. 5. The influence of model scale.

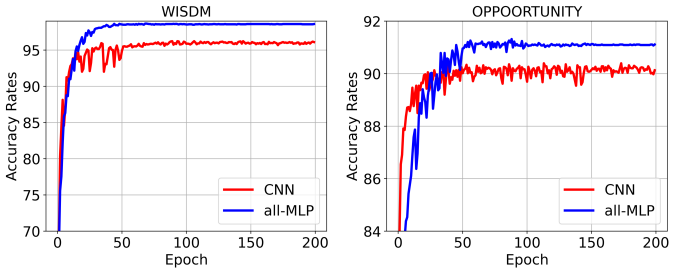


Fig. 6. Convergence speed comparison.

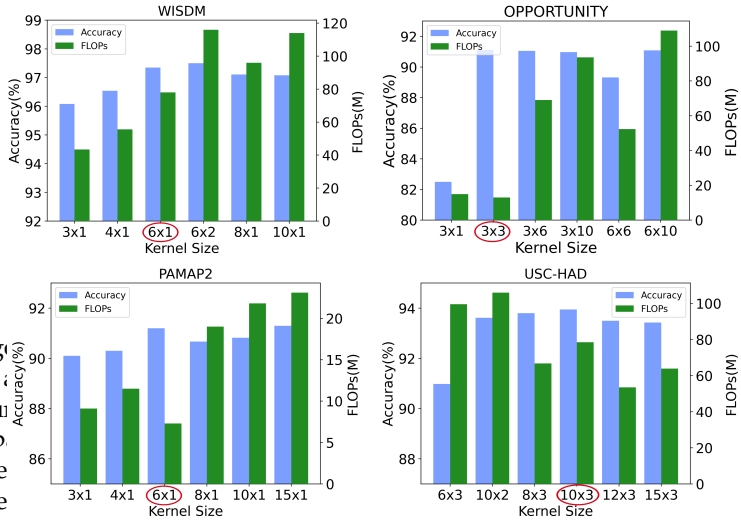


Fig. 7. The influence of kernel size on accuracy and FLOPs in CNNs.

ones on challenging activity recognition tasks.

In addition, it is interesting to note that MLP converg faster than CNN on WISDM and USC-HAD datasets, illustrated in Figure 6. In practice, there is a huge amou of things such as optimizer, learning rate, the number of p rameters, other architectural details, which potentially affe the convergence speed of the model. We intuitively suspe that this case is due to that MLP has fewer parameters so it may converge faster than CNN at this time. Intuitively, for the same architecture, smaller model size leads to lower network capacity and thus convergences faster; vice versa, a large model size requires a relatively long training time while keeping other training hyperparameters fixed. In our work, it should be pointed out that MLP with SGU might converge slower than CNN. To validate this, keeping the same model size to ensure fair comparisons, we further evaluate the convergence speed of CNN and MLP. As shown in Figure 6. it is evident that our MLP model converges slower than the CNN in terms of epochs, which we could attribute to the effect of SGU because it does not work initially and only tends to gradually inject cross-patch information during learning, thus leading to slower convergence. The results agree well with our expectations.

4.4.2 *Kernel Size in CNN*

Similarly, we show the comparison with classic CNN-based models with different kernel sizes. Because it could be an unfair comparison due to different model sizes, we adjust the kernel size of CNNs to ensure comparable model performance with the MLP counterparts. In Figure 7, fixing the number of layers and kennels, we show the influence of kernel size. It can be seen that our MLP models with much smaller model complexity still receive comparable or better results. Overall, the results support our main claim that in terms of the accuracy-speed trade-off MLP-like architectures are more competitive than conventional convolution-like ones on challenging activity recognition tasks.

4.4.3 *Patch Size in all-MLP*

In this part, we explore the influence of patch size on our all-MLP model. We select USC-HAD dataset for our evaluation. Without loss of generality, we select all four benchmark datasets for our evaluation. Results are summarized in Figure 8. Interestingly, we find an empirical relationship between patch size and classification score. For example, in the case of USC-HAD, it can be seen that FLOPs rapidly decay from 16.5M to 5.7M (3×) when patch size is increased from 64×1 to 64×3, which indicates that patch size has an important impact on computational overhead. As shown in Fig.7, in an extreme case e.g., 16×1 and 128×6, the classification accuracy has a sudden fall (e.g., 5.41% and 2.14% respectively) away from the peak. We intend to provide a reference design as accurate as possible. Although we find that a larger patch size might significantly decrease computational complexity, we select the patch size of 64×3 to ensure the overall classification score is roughly unchanged. As a comparison, it is worthwhile noting that patch size only causes a very small change in the number of parameters. Overall, the smaller patch size tends to generate more patches but inevitably raises computational costs. On the contrary, the larger patch size enjoys higher computation efficiency, but it is hard to capture finer-level details. Thus, in contrast, a medium-size scale is more beneficial for capturing fine-grained details, which can still maintain higher recognition accuracy.

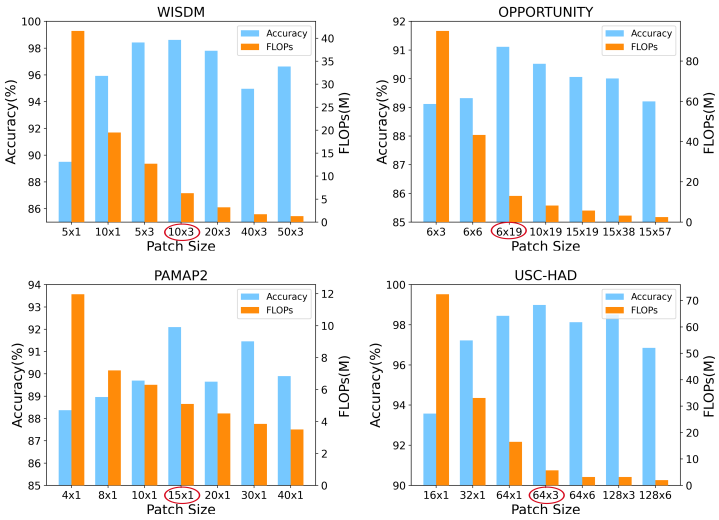


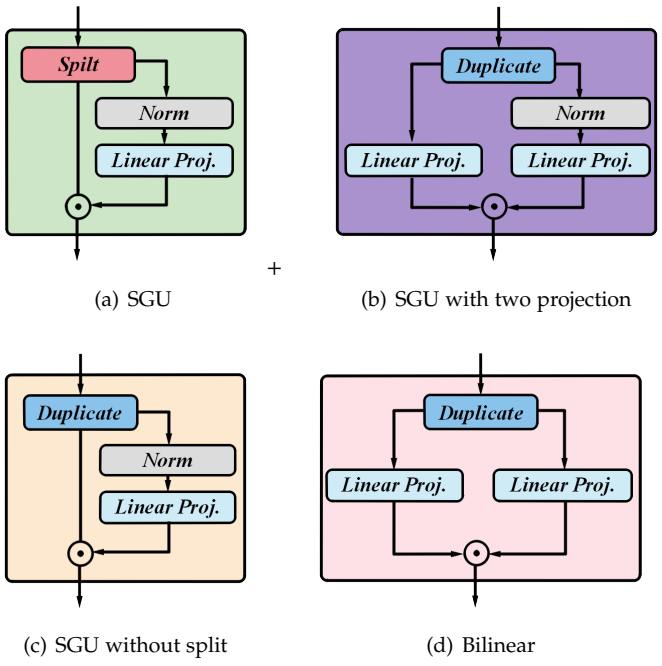
Fig. 8. The influence of patch size on accuracy and FLOPs in all-MLP.

4.4.4 *SGU Variants*

Unlike convolutional architecture with a fixed receptive field, the gating unit, i.e., SGU can handle long-range interactions between patches from sensor signals, which enables suitable and irregular receptive fields. The OPPORTUNITY challenge is used in the evaluation, which is very typical for HAR application scenarios due to its imbalanced class distributions. From a macro perspective of the overall structure (also see Figure 1), our proposed model primarily includes both channel projections (MLP) and Spatial Gating Unit (SGU), which accepts a sequence of linearly projected patches shaped as a "patches × channels" as an input. To isolate the effect of SGU, we further provide an ablation study by removing it from our model, where only channel projections are stacked through MLPs. The comparison result denoted as "Without SGU" is presented in Table 3. It can be seen that such pure MLP achieves an accuracy of 90.14%, which is behind the state-of-the-art CNN model, as well as our MLP with SGU. We conclude that the presence of this extra SGU module is related to the performance gap, because it is capable in allowing communication across patches other than linear projections that only allow communication across channels. These two types of operations, i.e., SGU and linear projection are interleaved to enable interaction of different input dimensions. The results support our main claim that our MLP model is competitive with mainstream conventional neural network architectures in terms of the accuracy-cost trade-off, which are also consistent with previous observations.

In order to adequately quantified the effectiveness of SGU, we replace SGU with its three variants and the details of their architectures are illustrated in Figure 9. We compare the performance of these SGU variants. In (a), SGU is the gating unit used in the above main experiments. As a comparison, (c) is a larger SGU without spilt, which can be represented as $s(X) = X \otimes f(\tilde{X})$. SGU also has a close connection with GLU [28]. (b) is similar to GLU and their main difference lies in that we substitute the sigmoid activation and normalize only one branch, where the output can be represented as $s(X) = f_1(X) \otimes f_2(\tilde{X})$. If we further remove the normalization in (b), it will degenerate into (d), which can be called "bilinear" in [57]. Results are illustrated Table 3. In particular, we can find that it would be more efficient to split the X in the aforementioned formulation (i.e., Eq. 6) into two independent parts (X_1, X_2) along the channel dimension, and then perform the element-wise multiplication in the gating function: $s(X) = X_1 \otimes f(Norm(X_2))$. It can be seen that SGU with split performs on par with the one without split in terms of accuracy, where the performance gap between them is almost negligible. Therefore, to strike a better accuracy-cost trade-off, it is always recommended to adopt SGU with split in this paper since it can perform comparably well but at a smaller computational cost.

Moverover, we empirically find that it is critical to initialize W as near-zero values and b as ones in $(f(\cdot))$ to ensure training stability. Such initialization can force the model to behave like a regular FFN at the beginning of learning, in which each patch can be independently processed, and then gradually inject spatial information across patches during the stage of training. As is shown in Table 4, SGU without weight initialization will have worse performance and instability. Specifically, the models without SGU initialization have worse classification accuracy across all four HAR datasets. Moreover, the training will be unstable since the results show higher variance over several independent runs.



(a) SGU (b) SGU with two projection

(c) SGU without split (d) Bilinear

Fig. 9. The architectures of SGU variants.

4.4.5 *Leave-one-out Cross Validation*

To evaluate the effectiveness of our MLP architectures, we conduct leave-one-subject-out cross-validation. It is a special case of k-fold cross-validation, where one person is treated as one-fold. Hence, the number of persons decides the number of folds. Using leave-one-subject-out cross-validation for performance evaluation of CNN and MLP requires rather expensive computation. Without loss of generality, we select

TABLE 3
Performance comparisons of SGU variants.

Benchmark	OPPORTUNITY		
Module	Accuracy(%)	Para.	FLOPs
Without SGU	90.14±0.17	0.56M	16.7M
SGU	91.11±0.08	**0.43M**	**13.3M**
SGU with two projection	90.16±0.05	0.56M	18.7M
SGU without split	**91.16±0.11**	0.56M	17.8M
Bilinear	89.70±0.07	0.56M	18.5M

TABLE 4
Performance comparisons of SGU initialization.

Dataset	WISDM	PAMAP2	OPPO	USC-HAD
w/o initialization	97.75±0.24	91.02±0.33	90.80±0.28	97.85±0.29
w/ initialization	**98.21±0.24**	**92.15±0.24**	**91.13±0.13**	**98.98±0.08**

PAMAP2 dataset for our evaluation. Because nine persons take part in this data collection, the number of folds will be set to 9. Thus, we perform 9-fold cross-validation, where the network will be trained on eight subjects and tested on one "left out" subject at each time. That is to say, every individual subject will be treated as a "test" set in turn. The average macro F1-scores with 95% confidence intervals are utilized as a metric to evaluate the robustness of the model. Table 5 summarizes our results. It can be clearly observed that MLP could produce a stable performance gain, which surpasses the standard CNN by 1.2% in terms of the averaged accuracy in a computation-saving scenario.

TABLE 5
Average macro F1-scores with 95% confidence intervals obtained in Leave-one-out Cross Validation on PAMAP2.

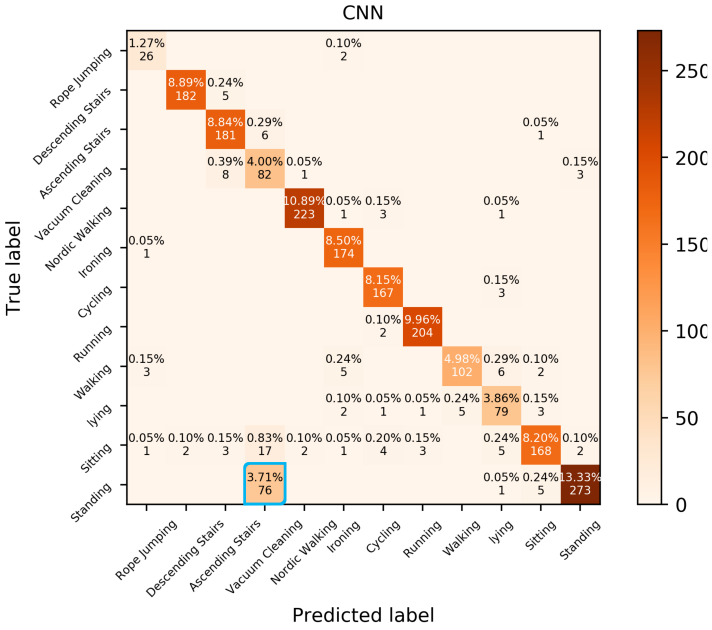
Subject\Model	CNNs	all-MLP
Subject 1	90.20±0.93	**90.58±1.15**
Subject 2	74.22±0.17	**74.76±0.31**
Subject 3	82.48±0.73	**83.45±0.67**
Subject 4	**94.47±0.34**	93.10±0.29
Subject 5	**97.39±0.45**	96.89±0.32
Subject 6	90.80±0.41	**91.92±0.72**
Subject 7	95.60±0.06	**95.60±0.02**
Subject 8	90.28±2.78	**98.61±1.39**
Subject 9	91.92±0.07	**92.51±0.18**
overall	89.60±0.66	**90.82±0.56**

4.4.6 *Confusion Matrices*

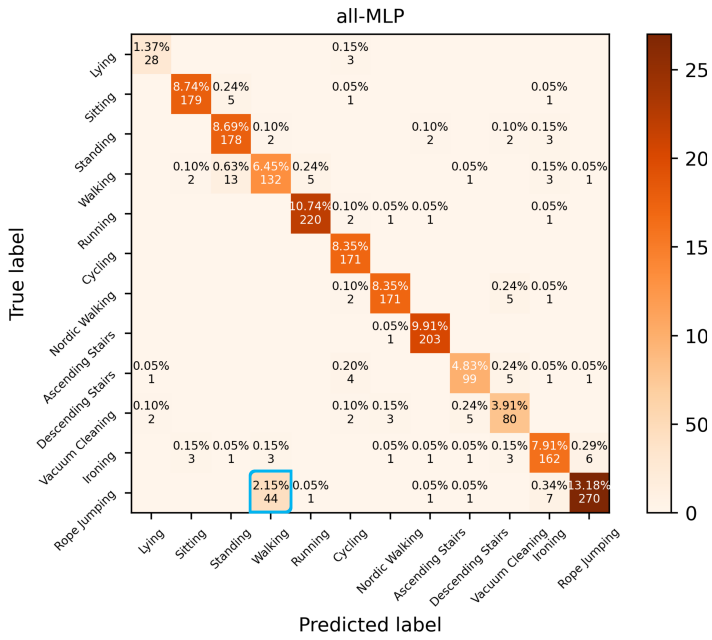
We visually compare the confusion matrix of all-MLP with CNNs. Datasets including PAMAP2 and OPPORTUNITY are selected for our evaluation, which is universally recognized as a realistic and challenging activity recognition task. Results are shown in Figure 10. The x-axes represent the predicted activity labels and the y-axes represent the real activity labels, while the diagonal values illustrate the number of samples to be recalled. The figure shows that the 3-layer normal CNN makes lots of misclassification (e.g.,76) when distinguishing the two types of activities 'Walking' and 'Rope Jumping', which can be attributed to the fact that two similar activities can produce very close waveform signals. It is apparent that our all-MLP model is better at discriminating very similar activities than convolutional architecture. In particular, the misclassified samples are significantly reduced from 76 to 48, which proves the robustness and effectiveness of our all-MLP model in the HAR scenario. Actually, the OPPORTUNITY dataset is extremely imbalanced, as the Null class represents more than 75% of all examples. Thus, it is relatively hard to illustrate performance gain by the confusion matrices with different colors. Hence, excluding the Null class, we further compute the confusion matrices on OPPORTUNITY dataset. Note that confusion between 'Open Drawer 1' and 'Open Drawer 2' is high. It can be observed from Figure 11 that many of the misclassifications occur between the two activities, which is due to that they are relatively similar. The results show the superiority of our MLP architectures. In essence, CNN tends to extract local features extremely well, but it cannot capture long-range interactions. Thus, the proposed all-MLP architecture can handle long-range interactions, which provides a better alternative to sense the whole waveform of sensor signals.

4.4.7 *Test on Mobile Platform*

Besides indirect metric like FLOPs, the direct metric such as inference speed should be considered in network architecture design. Thus, we evaluate the actual inference time of our all-MLP model on a resource-constrained embedded device. Since the Raspberry Pi system has a good combability with the PyTorch deep learning library, we chose the Raspberry Pi 3 Model B with ARM Cortex-A53 and SDRAM as our test platform. In order to evaluate the efficiency of the proposed model, we compare our all-MLP model with CNN in terms of inference speed. The patch size is set to 10×3. Both models are trained with WISDM dataset and then deployed on the Raspberry Pi platform. A Raspberry Pi-based application is developed for real-time activity recognition. The graphic user interface is written in Python and its three screenshots is illustrated in Figure 12. It presents the probability of the predicted activities and inference time. We average the inference time of 500 random samples from the test set and the results are shown in Figure 13. The averaged inference time by CNN takes around 119ms per window, while our all-MLP model takes only 30ms per window, which can significantly speedup the inference process. Specifically, under a higher classification score, it can be seen that every $\sim 12.4\times$ theoretical speedup often leads to $\sim 4\times$ actual speedup in our implementation due to memory access and other overheads. This actual speedup is partly attributed to all-MLP's simple design of architecture. In summary, our all-MLP model involved in only matrix multiplication can provide an actual speedup for activity inference at a much smaller memory footprint, which is very suitable for practical HAR deployment in wearable devices.

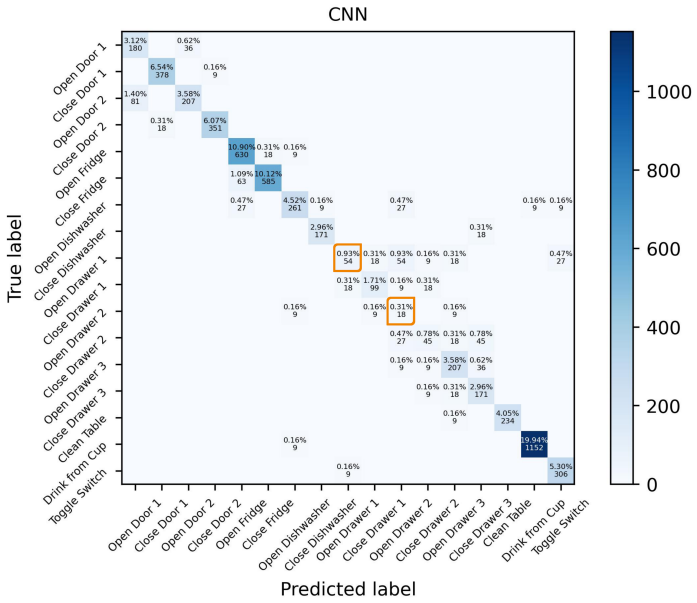


(a) standard CNN

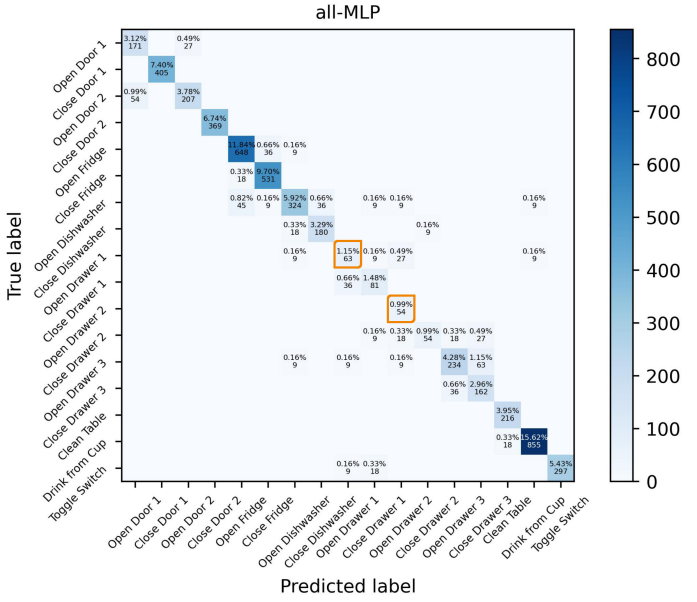


(b) all-MLP

Fig. 10. Performance comparsion on PAMAP2 dataset using Confusion Matrices.



(a) standard CNN



(b) all-MLP

Fig. 11. Confusion Matrices of OPPORTUNITY dataset.

5 Conclusion

Current research hotspot in learning architectures is returning to MLPs. In this paper, we first present a novel all-MLP architecture for HAR tasks based on wearable devices. The proposed architecture is entirely built by a set of MLP layers with a gating unit, without relying on any convolutional architecture. By dividing multi-channel sensor time series into nonoverlapping patches, all linear layers directly process sensor patches to extract local features, thereby decreasing computational overhead. The new architecture is simpler than CNN, which can easily handle sensor data structures. Through experiments conducted on four HAR benchmarks, we compare our all-MLP model with standard CNNs and other representative convolutional architectures. The experimental results show the proposed model can provide compelling results for activity recognition meanwhile greatly reducing computational complexity. It takes fewer FLOPs and parameters but achieves a comparable or higher classification score. We also conduct extensive ablation studies to prove the advantage of all-MLP model. Moreover, the computations of all-MLP only involve matrix multiplications, which can be readily optimized via using popular deep learning libraries, e.g., TensorFlow and PyTorch. This additional benefit indicates that our all-MLP model is hardware-friendly and can promote practical deployment in

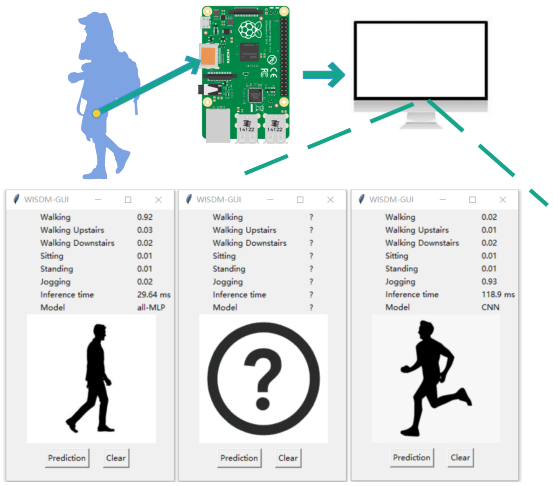


Fig. 12. The graphic user interface of the HAR application on Raspberry Pi.

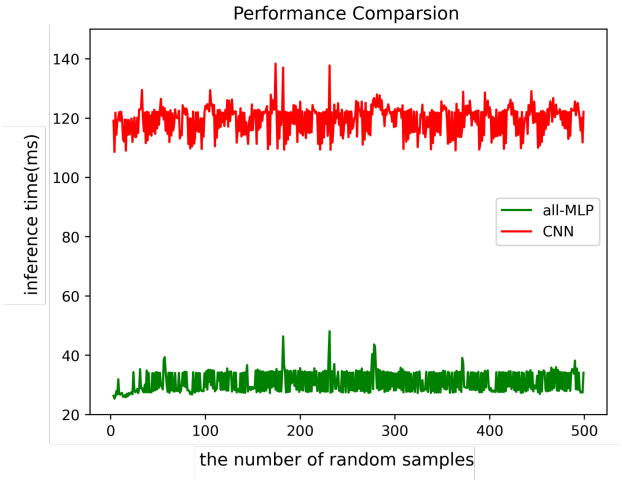


Fig. 13. Actual test on Raspberry Pi 3 B+.

resource-constrained wearable devices. We further evaluate actual inference speed on a Raspberry Pi platform, which shows that it can achieve $\sim 4\times$ actual speedup with higher accuracy over standard CNN. All in all, the proposed HAR model has a simple network structure and fast inference speed throughput. We conclude the new architecture is not a simple reuse of traditional MLPs, but is a significant evolution over them. Therefore, it deserves deeper investigation to explore the potential of such architectures to the greatest extent. We hope this work could encourage future work of network architecture design for wearable-based HAR with MLPs, which is platform-friendly and more practical.

ACKNOWLEDGMENTS

The work was supported in part by the National Nature Science Foundation of China under Grant 62373194 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. Lei Zhang is the corresponding author.

REFERENCES

[1] B. Khaertdinov, S. Asteriadis, and E. Ghaleb, "Dynamic temperature scaling in contrastive self-supervised learning for sensor-based human activity recognition," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2022.

[2] 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.

[3] Z. Qin, Y. Zhang, S. Meng, Z. Qin, and K.-K. R. Choo, "Imaging and fusing time series for wearable sensor-based human activity recognition," *Information Fusion*, vol. 53, pp. 80–87, 2020.

[4] L. M. Dang, K. Min, H. Wang, M. J. Piran, C. H. Lee, and H. Moon, "Sensor-based and vision-based human activity recognition: A comprehensive survey," *Pattern Recognition*, vol. 108, p. 107561, 2020.

[5] B. Khaertdinov, E. Ghaleb, and S. Asteriadis, "Deep triplet networks with attention for sensor-based human activity recognition," in *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2021, pp. 1–10.

[6] S. A. Rokni, M. Nourollahi, and H. Ghasemzadeh, "Personalized human activity recognition using convolutional neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[7] K. Wang, J. He, and L. Zhang, "Sequential weakly labeled multi-activity localization and recognition on wearable sensors using recurrent attention networks," *IEEE Transactions on Human-Machine Systems*, 2021.

[8] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 261–24 272, 2021.

[9] L. Melas-Kyriazi, "Do you even need attention? a stack of feed-forward layers does surprisingly well on imagenet," *arXiv preprint arXiv:2105.02723*, 2021.

[10] H. Liu, Z. Dai, D. So, and Q. V. Le, "Pay attention to mlps," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9204–9215, 2021.

[11] M.-H. Guo, Z.-N. Liu, T.-J. Mu, and S.-M. Hu, "Beyond self-attention: External attention using two linear layers for visual tasks," *arXiv preprint arXiv:2105.02358*, 2021.

[12] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[13] M.-H. Guo, Z.-N. Liu, T.-J. Mu, D. Liang, R. R. Martin, and S.-M. Hu, "Can attention enable mlps to catch up with cnns?" *Computational Visual Media*, vol. 7, no. 3, pp. 283–288, 2021.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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. 20, no. 13, pp. 7265–7274, 2020.

[19] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, 2016, pp. 1533–1540.

[20] 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.

[21] H. Ma, W. Li, X. Zhang, S. Gao, and S. Lu, "Attnsense: Multi-level attention mechanism for multimodal human activity recognition." in *IJCAI*, 2019, pp. 3109–3115.

[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[23] W. Huang, L. Zhang, H. Wu, F. Min, and A. Song, "Channel-equalization-har: A light-weight convolutional neural network for wearable sensor based human activity recognition," *IEEE Transactions on Mobile Computing*, 2022.

[24] 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.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[26] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[27] D. Hendrycks and K. Gimpel, "Bridging nonlinearities and stochastic regularizers with gaussian error linear units," *arXiv preprint arXiv:1606.08415*, 2016.

[28] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*. PMLR, 2017, pp. 933–941.

[29] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[30] 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.

[31] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha *et al.*, "Collecting complex activity datasets in highly rich networked sensor environments," in *2010 Seventh international conference on networked sensing systems (INSS)*. IEEE, 2010, pp. 233–240.

[32] 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.

[33] M. Zhang and A. A. Sawchuk, "Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors," in *Proceedings of the 2012 ACM conference on ubiquitous computing*, 2012, pp. 1036–1043.

[34] W. Huang, L. Zhang, W. Gao, F. Min, and J. He, "Shallow convolutional neural networks for human activity recognition using wearable sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.

[35] S. Mahmud, M. Tonmoy, K. K. Bhaumik, A. Rahman, M. A. Amin, M. Shoyaib, M. A. H. Khan, and A. A. Ali, "Human activity recognition from wearable sensor data using self-attention," *24th European Conference on Artificial Intelligence(ECAI)*, 2020.

[36] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan, "Deep activity recognition models with triaxial accelerometers," in *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[37] A. Murad and J.-Y. Pyun, "Deep recurrent neural networks for human activity recognition," *Sensors*, vol. 17, no. 11, p. 2556, 2017.

[38] F. Moya Rueda, R. Grzeszick, G. A. Fink, S. Feldhorst, and M. Ten Hompel, "Convolutional neural networks for human activity recognition using body-worn sensors," in *Informatics*, vol. 5, no. 2. Multidisciplinary Digital Publishing Institute, 2018, p. 26.

[39] S. W. Pienaar and R. Malekian, "Human activity recognition using lstm-rnn deep neural network architecture," in *2019 IEEE 2nd wireless africa conference (WAC)*. IEEE, 2019, pp. 1–5.

[40] L. Wang, J. Sun, T. Pan, Y. Ye, W. He, and K. Yang, "Personalized human activity recognition using hypergraph learning with fusion features," in *2021 IEEE 4th International Conference on Electronics Technology (ICET)*. IEEE, 2021, pp. 1251–1255.

[41] Y. Bai, J. Mei, A. L. Yuille, and C. Xie, "Are transformers more robust than cnns?" *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[42] L. Liu, X. Liu, J. Gao, W. Chen, and J. Han, "Understanding the difficulty of training transformers," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Association for Computational Linguistics, 2020, pp. 5747–5763.

[43] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR)*, 2019.

[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[45] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Device-free human activity recognition using commercial wifi devices," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1118–1131, 2017.

[46] L. Battle, P. Duan, Z. Miranda, D. Mukusheva, R. Chang, and M. Stonebraker, "Beagle: Automated extraction and interpretation of visualizations from the web," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–8.

[47] Z. N. Khan and J. Ahmad, "Attention induced multi-head convolutional neural network for human activity recognition," *Applied Soft Computing*, vol. 110, p. 107671, 2021.

[48] Y. Zhang, T. Gu, and X. Zhang, "Mdldroid: a chainsgd-reduce approach to mobile deep learning for personal mobile sensing," in *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2020, pp. 73–84.

[49] S. P. Singh, M. K. Sharma, A. Lay-Ekuakille, D. Gangwar, and S. Gupta, "Deep convlstm with self-attention for human activity decoding using wearable sensors," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 8575–8582, 2020.

[50] 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 Journal*, vol. 21, no. 1, pp. 581–592, 2020.

[51] K. Chen, L. Yao, D. Zhang, B. Guo, and Z. Yu, "Multi-agent attentional activity recognition," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 2019, pp. 1344–1350.

[52] N. Rashid, B. U. Demirel, and M. A. Al Faruque, "Ahar: Adaptive cnn for energy-efficient human activity recognition in low-power edge devices," *IEEE Internet of Things Journal*, 2022.

[53] 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.

[54] M. Abdel-Basset, H. Hawash, R. K. Chakrabortty, M. Ryan, M. Elhoseny, and H. Song, "St-deephar: Deep learning model for human activity recognition in ioht applications," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4969–4979, 2020.

[55] D. Kong, Y. Bao, and W. Chen, "Collaborative learning based on centroid-distance-vector for wearable devices," *Knowledge-Based Systems*, vol. 194, p. 105569, 2020.

[56] X. Wang, L. Zhang, W. Huang, S. Wang, H. Wu, J. He, and A. Song, "Deep convolutional networks with tunable speed-accuracy tradeoff for human activity recognition using wearables," *IEEE Transactions on Instrumentation and Measurement*, 2021.

[57] A. Mnih and G. Hinton, "Three new graphical models for statistical language modelling," in *Proceedings of the 24th international conference on Machine learning (ICML)*, 2007, pp. 641–648.

Shuoyuan Wang is currently pursuing the B.S. degree with Nanjing Normal University. His research interests include activity recognition, pattern recognition, and machine learning.

Lei Zhang received the B.Sc. 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.

Xing Wang received the B.S. degree from Huaiyin Institute of Technology, Huaian, China, in 2020. He is currently pursuing the M.S.degree with Nanjing Normal University. His research interests include activity recognition, computer vision, and machine learning.

Wenbo Huang received the B.S. degree (2019) from Nanjing Tech University, Nanjing, China. He received the M.S. degree (2022) from Nanjing Normal University, Nanjing, China. Now, he is pursuing the Ph.D. degree in the School of computer science and engineering, Southeast University, Nanjing, China. His research interests include Ubiquitous Computing 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.

Aiguo Song (Senior Member, IEEE) received the Ph.D. degree in measurement and control from Southeast University, Nanjing, China, in 1996. He is currently a Professor with the School of Instrument Science and Engineering, Southeast University. His current research interests include teleoperation, haptic display, the Internet Telerobot-ics, distributed measurement systems, and machine learning. Dr. Song is also the Chair of the China Chapter of the IEEE Robotics and Automation Society.