



Research paper

HT-AggNet: Hierarchical temporal aggregation network with near-zero-cost layer stacking for human activity recognition

Jaegyun Park^a, Dae-Won Kim^{a,*}, Jaesung Lee^b^a School of Computer Science and Engineering, Chung-Ang University, 221, Heukseok-Dong, Dongjak-Gu, Seoul, 06974, Republic of Korea^b Department of Artificial Intelligence, Chung-Ang University, 221, Heukseok-Dong, Dongjak-Gu, Seoul, 06974, Republic of Korea

ARTICLE INFO

Keywords:

Human activity recognition

Wearable sensors

Neural networks

Real-time systems

ABSTRACT

With the steady growth of sensor technology and wearable devices in pervasive computing applications, sensor-based human activity recognition has gained attention in fields such as healthcare monitoring and fitness tracking. This has resulted in an increased need for accurate and real-time systems. Recent studies to satisfy the real-time conditions have attempted to design lightweight neural networks by mainly restricting the number of layers shallowly, which has decreased both inference time and accuracy. To recover the loss of accuracy, we propose an innovative hierarchical temporal aggregation network (HT-AggNet) that allows the network architecture to be deeper, leading to an accuracy gain with only a near-zero increase in computational cost. Furthermore, a temporal glance convolution is presented to model the global context information of the signal patterns. Consequently, the HT-AggNet hierarchically extracts the local and global temporal information and then merges them based on hierarchical temporal aggregation. In our experiments, the HT-AggNet outperformed existing methods on seven publicly available datasets and achieved state-of-the-art performance. The source code for the HT-AggNet is publicly available at <https://github.com/jgpark92/HT-AggNet>.

1. Introduction

Human activity recognition (HAR) aims to identify the activity tracks or motion details of users from the signal patterns captured by sensors (Dang et al., 2020). With the popularity of smart wearable devices, wearable-sensor-based HAR has become essential in widespread applications such as healthcare (Wang et al., 2019a; Lentzas and Vrakas, 2020), fitness tracking (Bulling et al., 2014; Ignatov, 2018), and surveillance (Prati et al., 2019; Hussain et al., 2022). These applications require real-time responses, as immediate action is needed for urgent events (Bianchi et al., 2019; Qiu et al., 2022; Park et al., 2023). In real-time HAR, neural networks (NNs) have been widely used to achieve higher accuracy without manually processing the features (Wang et al., 2019b; Chen et al., 2021). Neural networks (NNs) are typically trained on resource-rich computing devices and then deployed on resource-limited wearable devices where inference is performed (Zebin et al., 2019; Ramanujam et al., 2021; Zhang et al., 2022).

Recent real-time HAR studies have focused on lightweight NNs with shallow architectures owing to their low computational cost (Zebin et al., 2019; Nweke et al., 2018; Wan et al., 2020; Huang et al., 2021; Tong et al., 2022; Gao et al., 2021; Tang et al., 2022; Park et al., 2022; Teng et al., 2023). These shallow NNs show competitive performance in analyzing simple activities such as walking

and sitting, usually characterized by repetitive movements with lower-level semantics. However, their performance is low when monitoring complicated activities such as vacuuming and folding laundry, which consist of various permutations of continuous basic movements with higher-level semantics (Chen et al., 2021; Hussein et al., 2019; Xu et al., 2019a). NNs with deeper architectures have achieved remarkable success in computer vision because of their ability to model higher-level semantics (Bianchini and Scarselli, 2014; Sze et al., 2017; Li et al., 2021). Similarly, deeper NNs can improve the accuracy of complicated activities, as shown in Fig. 1. However, the unavoidable additional computations ($\sim 5\times$ floating point operations (FLOPs)) increase for the T-ResNet (Wang et al., 2017; Fawaz et al., 2019), hindering real-time response.

To overcome this limitation, this study aimed to design a deeper NN architecture while maintaining the efficiency of shallow architectures. Achieving this goal is challenging, as deeper NNs can easily violate real-time constraints owing to the inherent trade-off between depth and computational cost. Our research questions are, therefore, as follows: (1) *How can deeper NNs be developed without significantly increasing the computational cost (i.e., can near-zero-cost NNs be achieved for this purpose)?* (2) *To what extent does increasing the network depth improve performance?*

* Corresponding authors.

E-mail addresses: dwkim@cau.ac.kr (D.-W. Kim), curseor@cau.ac.kr (J. Lee).<https://doi.org/10.1016/j.engappai.2025.110465>

Received 10 June 2024; Received in revised form 17 October 2024; Accepted 27 February 2025

Available online 11 March 2025

0952-1976/© 2025 Elsevier Ltd. All rights reserved, including those for text and data mining, AI training, and similar technologies.

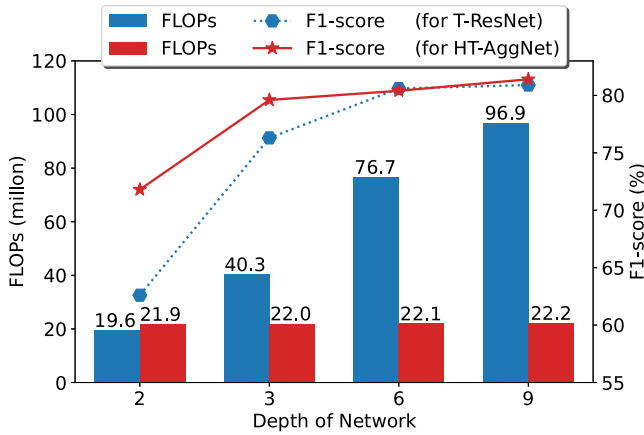


Fig. 1. Influence of network depth in T-ResNet and HT-AggNet on HAR performance. The experiment was performed on the OPPORTUNITY dataset (with complicated activities) (Chavarriaga et al., 2013). Both models achieved better F1 scores as the network became deeper. Compared to a T-ResNet, our HT-AggNet can stack layers with a negligible increase in FLOPs.

To answer these questions, we propose a hierarchical temporal aggregation network (HT-AggNet) with near-zero-cost layer stacking. As shown in Fig. 1, our HT-AggNet allows the deepening of the network architecture with a negligible increase in computational cost, resulting in performance gains. Given sensor signal patterns, a sequence of convolutional layers extracts intermediate features by capturing local temporal correlations. Subsequently, the HT-AggNet aggregates these hierarchical features using a new temporal glance convolution (TG-Conv) and a hierarchical temporal aggregation strategy, enabling near-zero-cost layer stacking. Finally, the aggregated features are fed into the final SoftMax layer to classify the activities. The most important contributions of this paper are as follows:

- HT-AggNet, a NN with *near-zero-cost layer stacking* (capable of stacking layers with a negligible increase in computational costs), is proposed, which results in a performance gain without significantly increasing the actual response time.
- A TGConv is introduced, which allows the network to encode global temporal information while maintaining the efficiency of the convolutions, resulting in the replacement of the recurrent layer.
- The superiority of the HT-AggNet compared to eight state-of-the-art methods is demonstrated on seven well-known public benchmark datasets.

The remainder of this paper is organized as follows: Section 2 describes the existing research on real-time HAR. Section 3 describes the HT-AggNet and near-zero-cost layer stacking. In Section 4, the details of the experimental setup are described, and the results are analyzed. Finally, Section 5 concludes the paper and presents future research directions.

2. Related work

The HAR literature has highlighted the importance of real-time applications and NN-based models in many comprehensive reviews (Dang et al., 2020; Wang et al., 2019b; Chen et al., 2021; Qiu et al., 2022; Ramanujam et al., 2021; Zhang et al., 2022; Nweke et al., 2018; Qi et al., 2018). Early real-time HAR studies transformed sensor signal patterns into 2D spectral images based on discrete or short-time Fourier transforms and fed them into 2D convolutional NNs (CNNs) with shallow architectures (Jiang and Yin, 2015; Chen and Xue, 2015; Ravi et al., 2016; Nutter et al., 2018). To avoid such complex data

preprocessing, Ignatov (Ignatov, 2018) used a 1D CNN with simple statistical features, including the magnitudes and histograms of the input signals. Although this approach demonstrated the effectiveness of the 1D CNN, it still required several preprocessing steps for the input signals.

More recent real-time HAR studies have focused on 1D CNNs without preprocessing the input signals. For example, Zebin et al. (2019) used a CNN consisting of four convolutional layers and parameter quantization. Wan et al. (2020) used a CNN with three convolutional layers and compared its performance on simple HAR datasets with that of recurrent NN variants. Huang et al. (2021) introduced cross-channel communication, where information is exchanged between the channels within the same layer. Although these models achieved real-time HAR, it was difficult to identify complicated activities due to their shallow architectures.

Several strategies have been proposed to solve this problem. The first strategy integrates recurrent layers into CNNs. Ordóñez and Roggen (2016) proposed DeepConvLSTM, which includes four convolutional layers and two recurrent layers, to capture the temporal dynamics. Furthermore, Xu et al. (2019a) developed the InnoHAR, which combines multibranch convolutional and recurrent layers. Luwe et al. (2022) integrated bidirectional long short-term memory (Bi-LSTM) into CNNs. Tong et al. (2022) attempted a real-time HAR by integrating gated recurrent units into CNNs. However, these approaches for real-time HAR are challenging due to the additional computations and the lack of hardware accelerators for the recurrent layers (Zebin et al., 2019; Gehring et al., 2017).

The second strategy considers multiple time scales for each convolutional layer. For example, Gao et al. (2021) developed a selective kernel module to learn multiscale temporal features to improve HAR accuracy. In another study, Tang et al. (2022) proposed a hierarchical split block to improve multiscale temporal features. Similarly, Park et al. (2022) introduced a multi-temporal sampling module that allows CNNs to consider multiple sampling frequencies. Furthermore, Teng et al. (2023) trained a multibranch CNN and re-parameterized it into a simple CNN before deployment. Although CNNs considering multiple time scales improve HAR accuracy, their shallow architectures make it difficult to capture higher-level semantics. Park et al. (2023) introduced a grouped temporal shift module to derive more efficient NNs while preserving the accuracy of existing NNs.

To summarize, existing real-time HAR studies use shallow NNs to fulfill real-time constraints. However, they only focused on improving the temporal modeling capabilities of each layer while maintaining a shallow depth of NNs, resulting in a limited improvement in HAR accuracy. In contrast, our study focused on developing a deeper CNN architecture while maintaining the efficiency of shallow NNs. Therefore, existing approaches can be easily integrated into HT-AggNets, resulting in better HAR performance.

3. Proposed method

In this section, the objective is to design a CNN architecture that can stack convolutional layers while maintaining the efficiency of shallow CNNs. The remainder of this section is organized as follows: Section 3.1 introduces the temporal glance convolution to extract the global context information of the signal patterns. Section 3.2 describes the hierarchical temporal aggregation strategy based on temporal glance convolution. Section 3.3 presents an efficient layer-stacking strategy. Finally, Section 3.4 describes the benefits of HT-AggNets in terms of model training.

Table 1

Notations.

Symbol	Meanings
L	Number of layers across a network
$\mathcal{K}^{(l)}$	l th layer
$\alpha(\cdot)$	Cost of calculating each layer
T	Temporal resolution of input for each layer
M	Number of input features for each layer
N	Number of output features for each layer
X	Input for each layer, $X \in \mathbb{R}^{T,M}$
Y	Output for each layer, $Y \in \mathbb{R}^{T,N}$
D_k	Kernel size in a convolution layer
V	Number of activities

3.1. Temporal glance convolution

Human activities consist of different permutations of continuous basic movements that exhibit considerable variations in temporal duration and order (Hussein et al., 2019). To capture these variations, some HAR studies have integrated recurrent layers into CNNs such that the network can extract the global contextual information of the signal patterns (Ordóñez and Roggen, 2016; Lee et al., 2021; Tong et al., 2022). However, these recurrent layers are challenging for real-time HAR because they require additional computation, and hardware accelerators for edge implementation are lacking. Therefore, in this section, the design of a new convolution operator for extracting global temporal features such that they can be integrated into CNNs while maintaining the time complexity of the standard convolution is explained. Table 1 lists the notation used in this study.

Let $\mathcal{K}_s$ and $\mathcal{K}_g$ be the convolutional layers used to extract the local and global temporal features, respectively, where $\mathcal{K}_s$ is the standard convolutional layer. In the convolutional layer, given an input $X \in \mathbb{R}^{T,M}$ and a kernel $K \in \mathbb{R}^{D_k,M,N}$ with kernel size D_k , the n th channel of $Y \in \mathbb{R}^{T,N}$ at time i is calculated as follows:

$$Y_{i,n} = \sum_{i,m} K_{i,m,n} X_{i+i,m}, \quad (1)$$

where $\hat{i} = i - \lfloor D_k/2 \rfloor$ is the re-centered time index and its time complexity is $\mathcal{O}(TD_kMN)$.

We note that D_k determines the time scale at which the convolutional layer extracts the information. It is well-known that the convolution can model local temporal information when $D_k \ll T$. If $D_k = T$, the convolutional layer can extract the global context information because each filter is connected with whole time stamps. As a result, its time complexity is $\mathcal{O}(T^2MN)$. However, using it as $\mathcal{K}_g$ is inappropriate because the time complexity of the CNNs is likely to increase according to the term T^2 .

To solve this problem, a temporal glance convolution (TGConv) was introduced as $\mathcal{K}_g$. Given an input $X \in \mathbb{R}^{T,M}$ and a kernel $K \in \mathbb{R}^{T,M,N}$, the n th channel of $Y^g \in \mathbb{R}^N$ is calculated as follows:

$$Y_n^g = \sum_{i,m} K_{i,m,n} X_{i,m}, \quad (2)$$

where Y_n^g has only one scalar value. This reduces the time complexity of the TGConv to $\mathcal{O}(TMN)$, which is smaller than that of the standard convolution, that is, $\mathcal{O}(TD_kMN)$.

3.2. Hierarchical temporal aggregation network

In this section, we describe how our TGConv improves the CNNs, resulting in a hierarchical temporal aggregation network (HT-AggNet). The TGConv extracts global temporal features with a loss of temporal resolution. Therefore, the HT-AggNet, with a hierarchical temporal aggregation strategy, is introduced to recognize activities based on the scalar outputs of the TGConv.

Fig. 2 shows the network architecture with hierarchical temporal aggregation. Here, hierarchical features are extracted by standard

convolution by computing short temporal correlations between layers. Then, the features extracted from each layer are fed into the TGConv to learn the global temporal relationships and are hierarchically aggregated. These hierarchical features provide more contextual and semantic information, resulting in increased performance (Zhang et al., 2021). More specifically, the earlier layers focus on identifying basic movements by capturing the transitions that cause substantial changes within each signal pattern. The features of the latter layers contain higher-level semantic information related to the contexts between the sequences of basic movements, which is beneficial for recognizing complicated activities. Finally, the SoftMax layer classifies the activities based on the aggregated features.

The time complexity of the HT-AggNet is analyzed to investigate whether it can maintain the efficiency of conventional CNNs. Let $\mathcal{K}^{(l)}$ and $\alpha(\mathcal{K}^{(l)})$ be the l th layer of a NN and its computational cost, respectively. For a NN with L layers, the input $X^{(l)}$ is fed into the l th layer with trainable parameters $\theta^{(l)}$ to calculate the output $X^{(l+1)}$. As the layers are calculated sequentially and independently, the computational cost of the NN is described as follows:

$$Cost = \sum_{l=1}^L \alpha(\mathcal{K}^{(l)}(X^{(l)}; \theta^{(l)})). \quad (3)$$

For the HT-AggNet, $\mathcal{K}^{(l)}$ is calculated using $\mathcal{K}_s^{(l)}$ and $\mathcal{K}_g^{(l)}$. Therefore, the time complexity of each layer is $\mathcal{O}(TD_kMN) + \mathcal{O}(TMN) = \mathcal{O}(TD_kMN)$, which is the same as the time complexity of each layer in conventional CNNs. Consequently, the HT-AggNet can extract global temporal features while maintaining the efficiency of a CNN. Moreover, hierarchical temporal aggregation with TGConvs can replace the recurrent layers.

3.3. Near-zero-cost layer stacking

In this section, a near-zero-cost layer-stacking strategy is presented that can deepen the network architecture without significantly increasing the computational cost. This strategy can fully exploit hierarchical temporal aggregation by deeply stacking layers for a given computational budget, improving the tradeoff between computation and accuracy.

It is assumed that T , D_k , M , and N are set to similar values in all layers. Under this assumption, Eq. (3) simplifies to

$$Cost \approx \mathcal{O}(TD_kMNL). \quad (4)$$

where $\mathcal{O}(TD_kMN)$ is the time complexity required to calculate the standard convolution and TGConv for each layer.

To reduce the time complexity of stacking standard convolutions, we use a grouped convolution with channel shuffling (Zhang et al., 2018). Given M input channels, they are evenly divided into L channel groups and then fed into the standard convolution to generate $\lfloor N/L \rfloor$ output channels for each channel group. Consequently, the time complexity of the stacked grouped convolutions is reduced to $\mathcal{O}(TD_kMN)$. This allows the HT-AggNet to stack the convolutional layers without increasing the time complexity.

Furthermore, each layer contains a channel shuffle operation to relate the channels between the groups. To fully relate the whole channel, it should satisfy $L \times (\text{number of channels within each group}) \geq (\text{number of channel groups})$ (Zhang et al., 2018). In general, as $L \times \lfloor N/L \rfloor \geq L$, the HT-AggNet, with its deeper architecture, can generate the hierarchical temporal features without loss of information for the channel correlations.

To reduce the time complexity of stacking TGConvs, the focus is on the number of scalar values extracted from each TGConv. The number of features a TGConv generates increases proportionally with L , and they are fed into the SoftMax layer. As a large number of input features in the SoftMax layer can lead to overfitting (Xu et al., 2019b), the number of features is reduced by dividing the number of filters in each

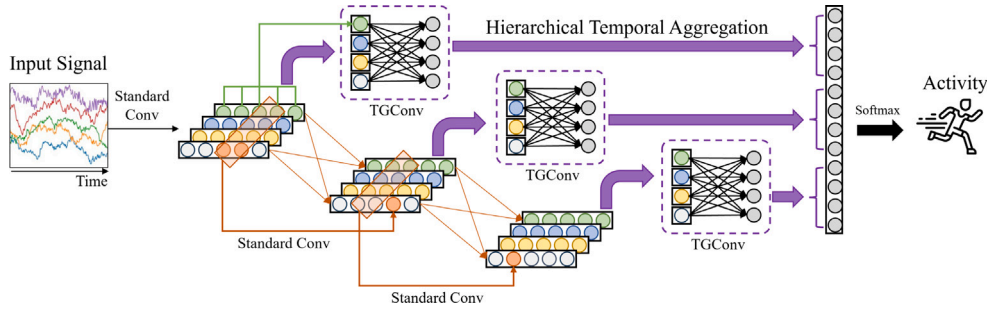


Fig. 2. Network architecture of HT-AggNet. Standard convolutions and the TGConv extract the local and global temporal features, respectively. The temporal features are hierarchically aggregated and fed into the SoftMax layer to predict activities.

Algorithm 1 Inference of HT-AggNet

```

1: Input:  $X^{(0)} \in \mathbb{R}^{T,M}$ ; ▷ Input signal data
2: Output:  $\hat{Y} \in \mathbb{R}^V$ ; ▷ The one-hot activity prediction
3:  $X^{(1)} \leftarrow \mathcal{K}_l^{(0)}(X^{(0)}; \theta_l^{(0)})$ ; ▷ generate the feature signals
4: for  $l = 1$  to  $L$  do
5:    $X^{(l+1)} \leftarrow \mathcal{K}_l^{(l)}(X^{(l)}; \theta_l^{(l)}, L)$ ; ▷ use Eq. (1) with  $L$  channel groups
6:    $X^{(l+1)} \leftarrow \text{ReLU}(\text{BN}(X^{(l+1)}))$ ; ▷ batch normalization BN
7:    $X^{(l+1)} \leftarrow \text{Shuffle}(X^{(l+1)})$ ; ▷ shuffle them at channel axis
8:    $H^{(l)} \leftarrow \mathcal{K}_g^{(l)}(X^{(l+1)}; \theta_g^{(l)})$ ; ▷ compute TGConv (Eq. (2))
9: end for
10:  $\tilde{H} \leftarrow \text{Concat}(H)$ ; ▷ concatenate them at channel axis
11:  $\hat{Y} \leftarrow \text{Softmax}(\tilde{H})$ ; ▷  $\tilde{H} \in \mathbb{R}^N$ 

```

TGConv by L . Consequently, the time complexity of stacking TGConvs is reduced to $\mathcal{O}(TMN)$.

Finally, an HT-AggNet with near-zero-cost layer stacking is introduced by omitting the factor L from its time complexity. As a result, our HT-AggNet has an efficiency comparable to that of shallow CNNs. Algorithm 1 shows the pseudocode for the forward pass of the HT-AggNet. In line 5, the intermediate feature signals are fed into the grouped convolution with L channel groups to extract local temporal information. All the convolutional layers are equipped with batch normalization and rectified linear units. In line 8, the TGConv extracts the global temporal information across all layers and generates the hierarchical temporal features $H^{(l)} \in \mathbb{R}^{\lfloor N/L \rfloor}$. Finally, the SoftMax layer predicts the activities based on H in lines 10–11.

3.4. Training

In addition to efficient inference, the HT-AggNet has another advantage with respect to the training parameters of the network. Specifically, let $\mathcal{L}$ be the loss function. Given the l th convolutional layer $\mathcal{K}_l^{(l)}$, the gradient of $\mathcal{L}$ with respect to $\theta_l^{(l)}$ is given as follows:

$$\nabla_{\theta_l^{(l)}} \mathcal{L} = \frac{\partial \theta_g^{(l)}}{\partial \theta_l^{(l)}} \cdot \nabla_{\theta_g^{(l)}} \mathcal{L} + \frac{\partial \theta_l^{(l+1)}}{\partial \theta_l^{(l)}} \cdot \nabla_{\theta_l^{(l+1)}} \mathcal{L} \quad (5)$$

where $\theta_g^{(l)}$ is a trainable parameter of the l th TGConv.

As shown in Eq. (5), the TGConv allows the trainable parameters to be trained using a shorter backward pass. This characteristic facilitates the training of deeper network architectures by improving the flow of information and gradients through the network (He et al., 2016; Lee et al., 2015; Huang et al., 2017).

4. Experimental results

In this section, we evaluate the performance of the HT-AggNet. Section 4.1 describes the benchmark datasets, the baseline networks, and the evaluation metrics in detail. Section 4.3 describes the results

of our HT-AggNet and other networks on the seven HAR datasets. In Section 4.4, an in-depth performance analysis of the ablation study is conducted. Finally, in Section 4.2, the actual response time of the HT-AggNet is measured.

4.1. Experimental settings

Seven well-known public benchmark datasets with various activities, segment sizes, and sampling rates were used in this study. The details are given in Table 2. We used the signals collected from wearable sensors such as accelerometers, gyroscopes, and magnetometers. In Table 2, OPPORTUNITY, PAMAP2, and REALDISP include more complicated activities (e.g., watching TV and folding laundry). The segment size and the ratio of cross-validation were set according to the authors' recommendations. On the OPPORTUNITY dataset, according to Mim et al. (2023), ADL 5 data were used for subject 1, drill data for subject 2, ADL 1 and ADL 4 data for subject 3, and ADL 4 data for subject 4 as the test set. The remaining dataset was used as the training dataset. Because the PAMAP2 dataset was split based on the subject's identification (ID), we split the data for the activities performed by only one subject into training and test sets at a ratio of 7:3, ensuring a balanced number of each activity in training and test sets.

The HT-AggNet was compared with eight baseline networks, as follows:

- **CNNs for real-time HAR.** Three state-of-the-art real-time HAR models were used. Shallow ConvNet (Ignatov, 2018) consists of one convolutional layer and two fully connected layers, where the outputs of the last convolutional layer are concatenated with basic statistical features. RepHAR (Teng et al., 2023) consists of three multibranch convolution layers with reparameterization and a SoftMax layer. GTSNet (Park et al., 2023) consists of nine grouped temporal shift module layers and one SoftMax layer. MTS-ConvNet (Park et al., 2022) comprises five multi-temporal sampling module layers and a Softmax layer.
- **CNNs with recurrent layers.** To verify the effectiveness of the TGConvs, two well-known CNNs with recurrent layers were used. Deep-ConvLSTM (Ordóñez and Roggen, 2016) consists of four convolutional layers and two recurrent LSTM layers with a SoftMax classifier. Bi-GRU-I (Tong et al., 2022) comprises two bidirectional GRU layers, three inception layers, and a SoftMax layer.
- **Time-Series Classification (TSC) models.** To demonstrate the superiority of the HT-AggNet on HAR datasets, two CNNs that are widely used in TSC and have a more general purpose than HAR were used. T-ResNet (Wang et al., 2017; Fawaz et al., 2019) consists of three residual blocks and a SoftMax layer. Each residual block contained three convolutional layers and a residual connection. T-FCN (Wang et al., 2017; Fawaz et al., 2019) is composed of three convolution layers (with more output channels than real-time models) and one SoftMax layer.

Table 2
Description of benchmark datasets.

	Context	# Subjects	Segment size (Frequency)	# Activities
UCI-HAR (Anguita et al., 2013)	Daily living	30	128 (50 Hz)	6
UniMiB-SHAR (Micucci et al., 2017)	Daily living & fall detection	30	151 (50 Hz)	9 ADL & 8 Fall
DSADS (Barshan and Yüsek, 2014)	Daily living & sports	8	125 (25 Hz)	19
KU-HAR (Sikder and Nahid, 2021)	Daily living	90	300 (100 Hz)	18
OPPORTUNITY (Chavarriaga et al., 2013)	Daily living	4	90 (30 Hz)	17 (Complicated)
PAMAP2 (Reiss and Stricker, 2012)	Daily living	9	512 (100 Hz)	18 (6 of them is Complicated)
REALDISP (Baños et al., 2012)	Fitness	17	250 (50 Hz)	33 (Complicated)

Table 3

Comparison of F1-scores and FLOPs of various models for two datasets. ▼/▲ indicates that the corresponding model is significantly worse/better than the proposed network according to a paired *t*-test at a 95% significance level.

Model	UCI-HAR		UniMiB-SHAR	
	F1-score	FLOPs	F1-score	FLOPs
HT-AggNet (Ours)	0.956	14.092M	0.773	16.149M
+ GTSCov (Ours)	0.937 ▼	2.459M	0.747 ▼	2.770M
+ GTSCov ($\times 1.5N$) (Ours)	0.939 ▼	4.342M	0.761	4.996M
+ MTS module (Ours)	0.929 ▼	1.816M	0.738 ▼	2.008M
Shallow ConvNet	0.925 ▼	17.900M	0.722 ▼	18.230M
RepHAR	0.951	31.755M	0.716 ▼	37.292M
GTSNet	0.945 ▼	6.383M	0.772	7.513M
MTS-ConvNet	0.940 ▼	2.178M	0.693 ▼	2.750M
DeepConvLSTM	0.914 ▼	67.204M	0.716 ▼	80.404M
Bi-GRU-I	0.946 ▼	46.073M	0.752 ▼	54.007M
T-ResNet	0.953	123.242M	0.765	145.506M
T-FCN	0.958	68.880M	0.769	80.602M

Table 4

Comparison of F1-scores and FLOPs of various models for two datasets. ▼/▲ indicates that the corresponding model is significantly worse/better than the proposed network according to a paired *t*-test at a 95% significance level.

Model	DSADS		KU-HAR	
	F1-score	FLOPs	F1-score	FLOPs
HT-AggNet (Ours)	0.873	20.107M	0.957	33.029M
+ GTSCov (Ours)	0.848 ▼	3.983M	0.938 ▼	5.721M
+ GTSCov ($\times 1.5N$) (Ours)	0.900 ▲	5.838M	0.942 ▼	10.059M
+ MTS module (Ours)	0.874	3.347M	0.918 ▼	4.239M
Shallow ConvNet	0.856 ▼	48.454M	0.779 ▼	41.552M
RepHAR	0.855 ▼	32.890M	0.934 ▼	74.428M
GTSNet	0.844 ▼	7.280M	0.945 ▼	15.331M
MTS-ConvNet	0.866	2.972M	0.884 ▼	5.588M
DeepConvLSTM	0.855 ▼	68.272M	0.935 ▼	169.143M
Bi-GRU-I	0.856 ▼	48.741M	0.949 ▼	107.985M
T-ResNet	0.873	125.829M	0.950	290.001M
T-FCN	0.867	76.133M	0.925 ▼	161.745M

4.2. Real-time activity prediction

Two widely used metrics were used for performance evaluation: F1-score and FLOPs. They measure the effectiveness and efficiency of the model, respectively. For fairness, we re-implemented all baselines in PyTorch (Paszke et al., 2019) and excluded sophisticated tricks, including gradient clipping (Mikolov et al., 2012) and learning rate warmup (He et al., 2019). All networks were trained for 500 epochs with a batch size of 128 using a 2080Ti graphics processing unit. The Adam optimizer (Kingma and Ba, 2015) was used with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$, where the learning rate and weight decay were set to 0.0005. For the HT-AggNet, D_k , N , and L were set to 5, 128, and 8, respectively. Details on all network architectures and their time complexity are provided in Appendix.

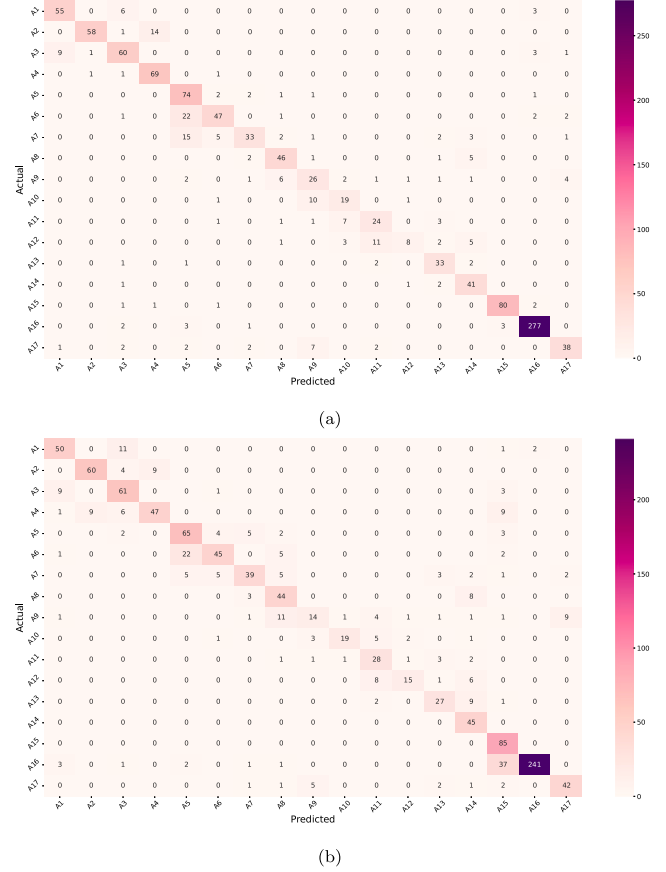


Fig. 3. Confusion matrix of (a) HT-AggNet ($L=8$) with 22.238M FLOPs and (b) HT-ConvNet ($L=2$) with 27.860M FLOPs on the OPPORTUNITY dataset. A1, open-door1; A2, open-door2; A3, close-door1; A4, close-door2; A5, open-fridge; A6, close-fridge; A7, open-dishwasher; A8, close-dishwasher; A9, open-drawer1; A10, close-drawer1; A11, open-drawer2; A12, close-drawer2; A13, open-drawer3; A14, close-drawer3; A15, clean-table; A16, drink-from-cup; A17, toggle-switch.

4.3. Comparative results

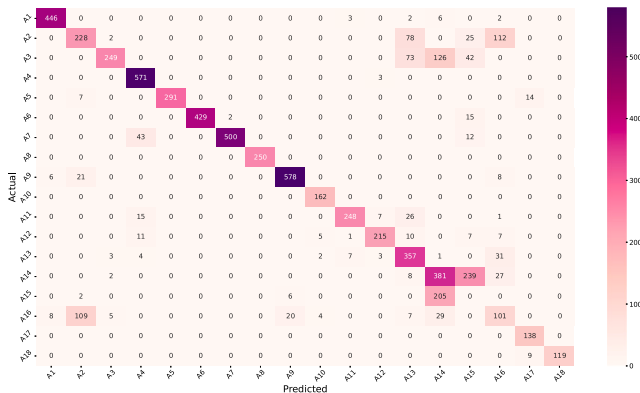
Tables 3–5 list the comparison results between the HT-AggNet and the baseline networks. The experiments were performed ten times, and the average values were recorded for all datasets. In addition, a paired *t*-test was performed at the 95% significance level for each dataset. As shown in Tables 3–5, ▼/▲ indicate whether the results, based on the paired *t*-test, of the comparison model, were worse or better than those of the proposed.

In Tables 3–5, the HT-AggNet has the best F1-scores on the seven datasets compared to Shallow ConvNet and RepHAR while maintaining lower FLOPs than them. HT-AggNet, in particular, achieved higher F1-scores than T-ResNet (with similar depth) on the seven datasets despite its significantly low FLOPs. These results support our claim that our HT-AggNet with near-zero-cost layer stacking can improve accuracy while maintaining the efficiency of shallow architectures. Meanwhile,

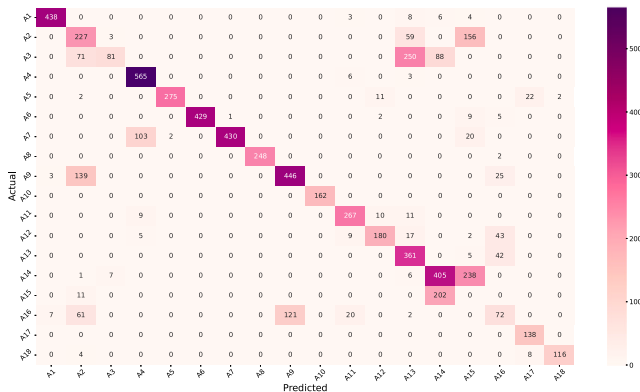
Table 5

Comparison of F1-scores and FLOPs of various models for three datasets with *complicated* activities. ▼/▲ indicates that the corresponding model is significantly worse/better than the proposed network according to a paired *t*-test at a 95% significance level.

Model	OPPORTUNITY		PAMAP2		REALDISP	
	F1-score	FLOPs	F1-score	FLOPs	F1-score	FLOPs
HT-AggNet (Ours)	0.814	22.238M	0.786	72.750M	0.975	63.048M
+ GTSCConv (Ours)	0.791 ▼	4.824M	0.743 ▼	13.833M	0.955 ▼	13.657M
+ GTSCConv (×1.5N) (Ours)	0.805	6.165M	0.769 ▼	21.197M	0.986 ▲	17.284M
+ MTS module (Ours)	0.761 ▼	4.364M	0.742 ▼	11.320M	0.930 ▼	12.417M
Shallow ConvNet	0.715 ▼	74.344M	0.674 ▼	151.812M	0.959 ▼	209.882M
RepHAR	0.800 ▼	26.032M	0.730 ▼	131.933M	0.947 ▼	72.689M
GTSNet	0.760 ▼	6.383M	0.762 ▼	28.616M	0.949 ▼	18.042M
MTS-ConvNet	0.728 ▼	3.415M	0.754 ▼	10.348M	0.878 ▼	8.240M
DeepConvLSTM	0.620 ▼	50.362M	0.773 ▼	303.892M	0.917 ▼	156.939M
Bi-GRU-I	0.772 ▼	39.794M	0.710 ▼	194.122M	0.961 ▼	111.304M
T-ResNet	0.809	96.865M	0.714 ▼	506.074M	0.960 ▼	270.088M
T-FCN	0.762 ▼	65.784M	0.725 ▼	298.456M	0.959 ▼	184.520M



(a)



(b)

Fig. 4. Confusion matrix of (a) HT-AggNet ($L = 8$) with 72.750M FLOPs and (b) HT-ConvNet ($L = 2$) with 104.731M FLOPs on the PAMAP2 dataset. A1, lying; A2, sitting; A3, standing; A4, walking; A5, running; A6, cycling; A7, Nordic-walking; A8, watching-TV; A9, computer-work; A10, car-driving; A11, ascending-stairs; A12, descending-stairs; A13, vacuum-cleaning; A14, ironing; A15, folding-laundry; A16, house-cleaning; A17, playing-soccer; A18, rope-jumping.

the GTSNet and MTS-ConvNet, with efficient variants of convolutional layers, had significantly lower FLOPs. The lower bound of calculating HT-AggNet inference depends on the computational cost of calculating each convolutional layer. Therefore, these variants can reduce the FLOPs of our HT-AggNet further if HAR should be run on tiny devices like microcontrollers, although efficient convolutions may lead to a slight decrease in accuracy. For instance, GTSCConv or MTS module can be seamlessly integrated into HT-AggNet by simply replacing the standard convolutions with them. As shown in Tables 3–5, the FLOPs

Table 6

Ablation study of HT-AggNet on four datasets with *simple* activities; NLS: Near-zero-cost layer stacking, HTA: Hierarchical temporal aggregation.

UCI-HAR ($T = 128, 50$ Hz)	L	F1-score	FLOPs
HT-AggNet	8	0.956	14.092M
HT-AggNet without NLS	2	0.936 ▼	26.283M
HT-ConvNet (without HTA & NLS)	2	0.918 ▼	22.087M
Improvement		3.8%	1.57x
UniMiB-SHAR ($T = 151, 50$ Hz)	L	F1-score	FLOPs
HT-AggNet	8	0.773	16.149M
HT-AggNet without NLS	2	0.728 ▼	30.629M
HT-ConvNet (without HTA & NLS)	2	0.715 ▼	25.644M
Improvement		5.8%	1.59x
DSADS ($T = 125, 25$ Hz)	L	F1-score	FLOPs
HT-AggNet	8	0.873	20.107M
HT-AggNet without NLS	2	0.849 ▼	32.111M
HT-ConvNet (without HTA & NLS)	2	0.860	27.978M
Improvement		1.3%	1.39x
KU-HAR ($T = 300, 100$ Hz)	L	F1-score	FLOPs
HT-AggNet	8	0.957	33.029M
HT-AggNet without NLS	2	0.877 ▼	61.603M
HT-ConvNet (without HTA & NLS)	2	0.876 ▼	51.768M
Improvement		8.1%	1.57x

Table 7

Ablation study of HT-AggNet on three datasets with *complicated* activities; NLS: Near-zero-cost layer stacking, HTA: Hierarchical temporal aggregation.

OPPORTUNITY ($T = 90, 30$ Hz)		L	F1-score	FLOPs
HT-AggNet	8		0.814	22.238M
HT-AggNet without NLS	2		0.801 ▼	30.813M
HT-ConvNet (without HTA & NLS)	2		0.767 ▼	27.860M
Improvement			4.7%	1.25×
PAMAP2 ($T = 512, 100$ Hz)		L	F1-score	FLOPs
HT-AggNet	8		0.786	72.750M
HT-AggNet without NLS	2		0.763 ▼	121.513M
HT-ConvNet (without HTA & NLS)	2		0.724 ▼	104.731M
Improvement			6.2%	1.44×
REALDISP ($T = 250, 50$ Hz)		L	F1-score	FLOPs
HT-AggNet	8		0.975	63.048M
HT-AggNet without NLS	2		0.948 ▼	86.865M
HT-ConvNet (without HTA & NLS)	2		0.945 ▼	78.664M
Improvement			3.0%	1.25×

of the HT-AggNet were significantly reduced by using the GTSCConv and MTS modules. In addition, we conducted an experiment for HT-AggNet + GTSCConv with 1.5 times channels to compare its accuracy with that

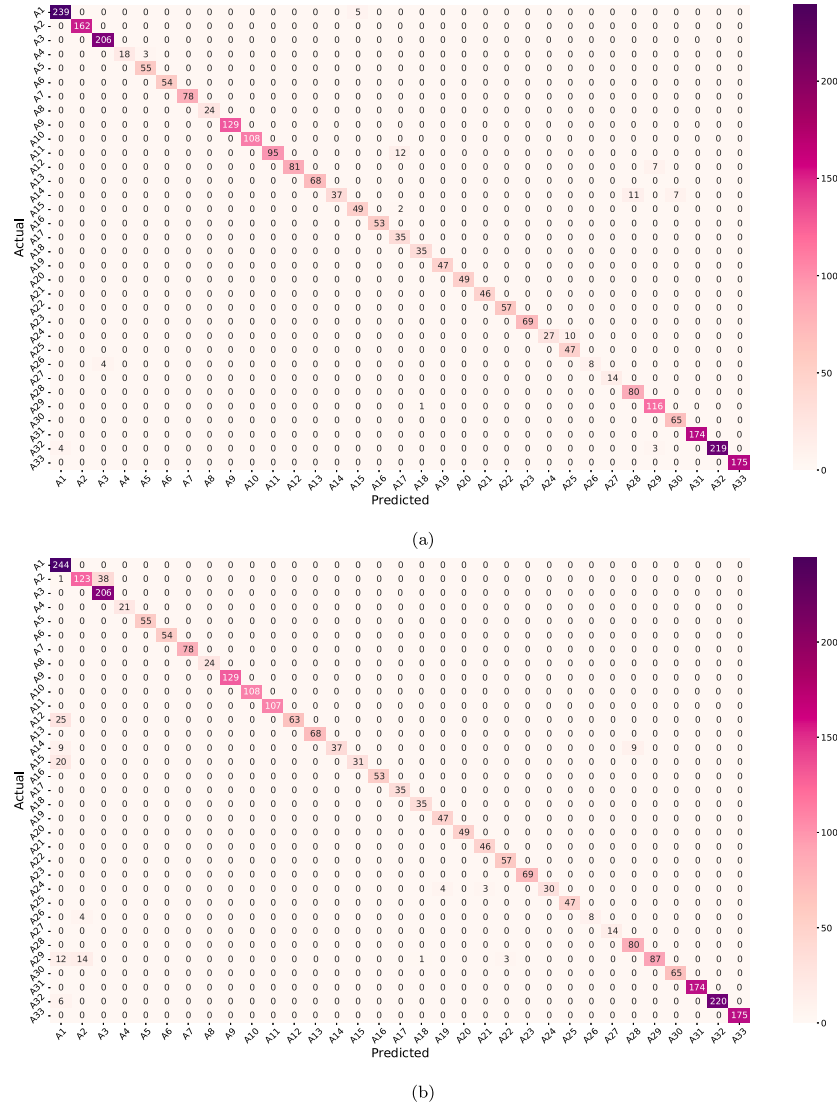


Fig. 5. Confusion matrix of (a) HT-AggNet ($L = 8$) with 63.048M FLOPs and (b) HT-ConvNet ($L = 2$) with 78.664M FLOPs on the REALDISP dataset. A1, walking; A2, jogging; A3, running; A4, jump-up; A5, jump-front-back; A6, jump-sideways; A7, jump-leg/arms-open/closed; A8, jump-rope; A9, trunk-twist-arms; A10, trunk-twist-elbows; A11, waist-bends-forward; A12, waist-rotation; A13, waist-bends; A14, reach-heels-backwards; A15, lateral-bend; A16, lateral-bend-arm-up; A17, repetitive-forward-stretching; A18, upper-trunk-and-lower-body-opposite-twist; A19, arms-lateral-elevation; A20, arms-frontal-elevation; A21, frontal-hand-claps; A22, arms-frontal-crossing; A23, shoulders-high-amplitude-rotation; A24, shoulders-low-amplitude-rotation; A25, arms-inner-rotation; A26, knees-alternatively-breast; A27, heels-alternatively-backside; A28, knees-bending-crouching; A29, knees-alternatively-bend-forward; A30, rotation-on-the-knees; A31, rowing; A32, elliptical-bike; A33, cycling.

of GTSNet in similar FLOPs. As a result, HT-AggNets with GTSCov and MTS module exhibited higher F1-scores than GTSNet and MTS-ConvNet on four and five datasets among seven datasets.

As shown in Tables 3–4, the HT-AggNet has the best F1-scores for the UniMiB-SHAR, DSADS, and KU-HAR datasets. For the UCI-HAR dataset, the F1-score of the HT-AggNet was statistically equivalent to that of the T-FCN (which had the best performance). Our HT-AggNet showed F1-scores equivalent to a T-ResNet on four simple HAR datasets, although it had significantly lower FLOPs. In particular, the HT-AggNet outperformed the T-FCN (with more channels) on the KU-HAR dataset. These results demonstrate that HT-AggNets can successfully predict simple activities with fewer computations. Compared with Shallow ConvNet, while F1-score of HT-AggNet showed a 1.7% improvement on DSADS dataset (with the lowest sampling frequency of 25 Hz), F1-score of HT-AggNet achieved a 17.8% improvement on KU-HAR dataset (with the highest frequency of 100 Hz). It implies that the more information the input has, the more critical the impact of network depth may become.

Table 5 shows the superiority of the HT-AggNet in terms of complicated activities. In particular, the HT-AggNet exhibited the best

F1-scores on three complicated HAR datasets compared to the state-of-the-art models. More importantly, the HT-AggNet results were statistically superior to those of other networks on the PAMAP2 and REALDISP datasets (which had longer segment sizes than the OPPORTUNITY dataset). This indicates that the HT-AggNet can extract meaningful features over long-term timescales. Moreover, the HT-AggNet was statistically superior to the other networks, except for T-ResNet, on the OPPORTUNITY dataset. These results show that CNNs can capture complicated activities as their depth increases. In addition, among the baseline networks, T-ResNet, DeepConvLSTM, and Bi-GRU-I, our HT-AggNet achieved the best F1 scores on the OPPORTUNITY, PAMAP2, and REALDISP datasets. The HT-AggNet achieved a 0.5% higher F1-score and $4.4 \times$ lower FLOPs on the OPPORTUNITY dataset, 1.3% higher F1-score and $4.2 \times$ lower FLOPs on the PAMAP2 dataset, and 1.4% higher F1-score and $1.8 \times$ lower FLOPs on the REALDISP dataset compared with the T-ResNet, DeepConvLSTM, and Bi-GRU-I. These results indicate that the HT-AggNet can achieve a better trade-off between accuracy and computational complexity.

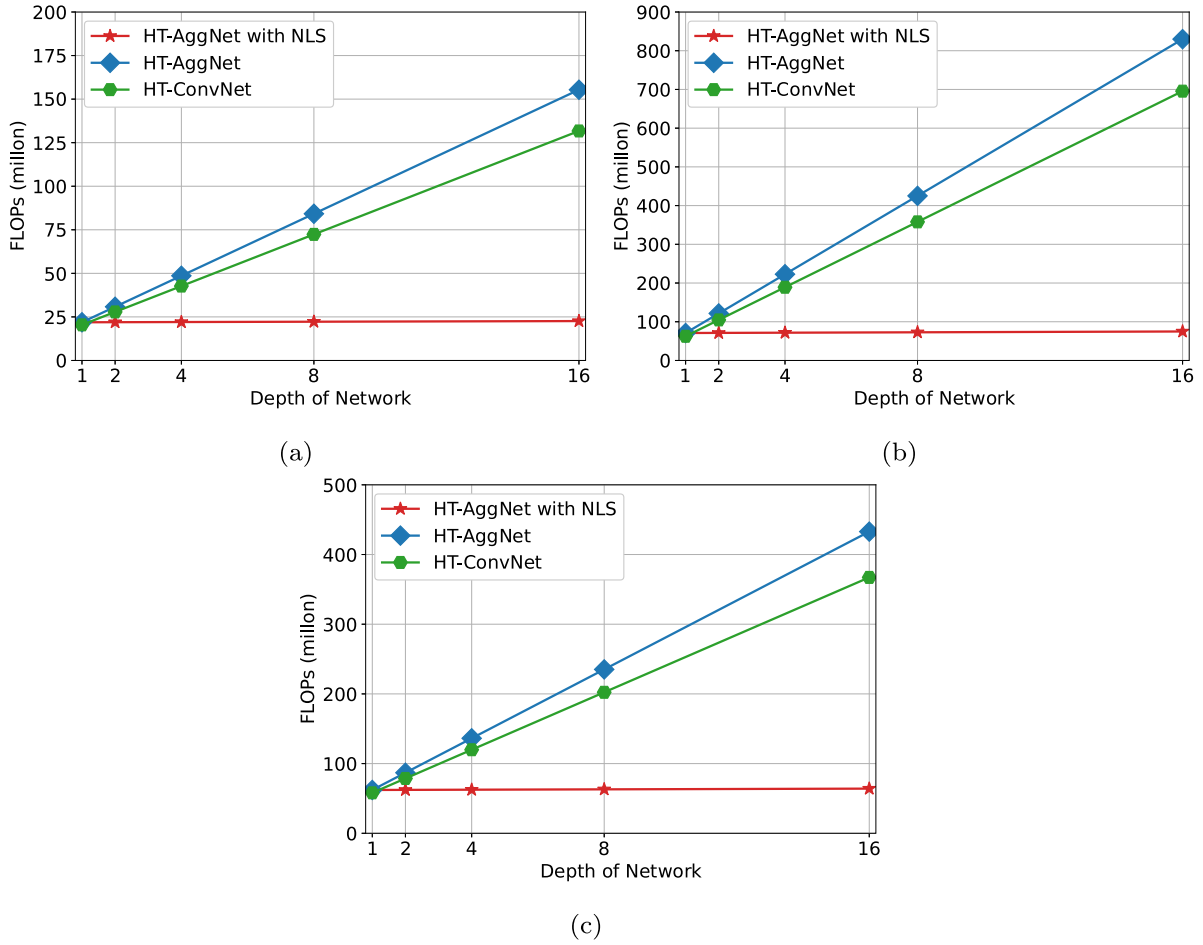


Fig. 6. Influence of L on the FLOPs for three networks. (a) OPPORTUNITY. (b) PAMAP2. (c) REALDISP.

4.4. Ablation study

The breakdown effect of HT-AggNet. An ablation study was conducted to investigate the effectiveness and efficiency of the HT-AggNet. The key components of the HT-AggNet are hierarchical temporal aggregation and near-zero-cost layer stacking. Therefore, the performance of the HT-AggNet was compared with that of its two variants, which were obtained by individually removing its components. As shown in Tables 6–7, the deeper HT-AggNet (with $L = 8$) outperformed the shallow one (with $L = 2$) while maintaining low FLOPs. In addition, Tables 6–7 show that hierarchical temporal aggregation using TGConvs improved the F1-score of the network on all datasets except for the DSADS dataset. However, on the DSADS dataset, hierarchical temporal aggregation using TGConvs reduced the F1-score, and its performance was improved by deepening the network architecture. These results suggest that our hierarchical temporal aggregation strategy can improve HAR accuracy by learning more contextual information and that, for a given computational budget, the hierarchical features can be improved by deep stacking layers. In addition, the higher the segment size or sampling frequency, the higher the improvement in the F1-score tended to be exhibited, as shown in Tables 6–7. These results suggest that our HT-AggNet can gain more benefits from using the pre-processing setup with a high segment size or sampling frequency. On the other hand, the near-zero-cost layer stacking can maintain the FLOPs regardless of the segment size.

Comparative analysis on complicated activities. Figs. 3–5 show confusion matrices for three datasets with complicated activities, where the left and right sides represent the HT-AggNet and HT-ConvNet (with only two standard convolutions). Their diagonal elements represent

Table 8

F1-score of HT-AggNet for different depths (L).

L	1	2	4	8	16
UCI-HAR	0.925	0.929	0.946	0.956	0.951
UniMiB-SHAR	0.661	0.719	0.752	0.773	0.763
DSADS	0.847	0.849	0.862	0.873	0.882
KU-HAR	0.876	0.875	0.924	0.957	0.970
OPPORTUNITY	0.718	0.796	0.804	0.814	0.807
PAMAP2	0.759	0.759	0.775	0.786	0.775
REALDISP	0.940	0.939	0.975	0.975	0.979

Table 9

Inference times of real-time HAR models.

Model	Inference time (ms/window)		
	Min	Mean	Max
HT-AggNet ($L = 8$)	1.81 ms	2.31 ms	4.78 ms
HT-AggNet ($L = 2$)	1.67 ms	2.21 ms	4.41 ms
Shallow ConvNet	3.36 ms	3.98 ms	5.74 ms

the number of instances that have been correctly classified as related activities, with the color becoming darker as the number increases. In Fig. 3, HT-ConvNet misclassified “drink from a cup” as “clean the table” 37 times, while our HT-AggNet misclassified these activities only three times on the OPPORTUNITY dataset. Specifically, both “drink from a cup” and “clean the table” involve a basic movement of picking up some objects. Therefore, it is essential to understand the context of the adjacent movements to classify them. The HT-AggNet can capture contextual information through hierarchical temporal aggregation. Similarly, the HT-ConvNet tends not to classify analogous activities,

such as “computer work” and “sitting” for the PAMAP2 dataset. As shown in Fig. 4, the HT-AggNet significantly improves the accuracy of classifying these activities. As shown in Fig. 5, the HT-ConvNet tends to misclassify other fitness activities, such as “walking”, while the HT-AggNet only misclassified them four times. These results show that our HT-AggNet with a deep-free architecture can improve classification accuracy for complicated activities.

F1-score vs. FLOPs. Finally, we investigated the trade-off between the F1-score and FLOPs. First, the F1-score of the HT-AggNet was compared as a function of the network depth L . As shown in Table 8, the F1-score increased for the four datasets until L reached eight. For the other three datasets, the HT-AggNet exhibited the best F1-score when L was set to the highest value. As mentioned in Section 1, we investigated *how much increasing the network depth improves the accuracy*. Although the relationship between network depth and accuracy varies depending on the sampling frequency, activity type, sensor type, and subject type of the dataset, increasing the network depth tends to improve the model’s accuracy, as shown in Table 8. Second, the growth rate of the FLOPs with increasing network depth L was compared for three networks: HT-AggNet (with and without near-zero-cost layer stacking) and HT-ConvNet. As shown in Fig. 6, the HT-AggNet with near-zero-cost layer stacking showed a negligible increase in FLOPs compared with the other networks as L increased. When $L = 16$, the HT-AggNet with near-zero-cost layer stacking achieved $\sim 10\times$ improvement in FLOPs compared to the HT-ConvNet. These results demonstrate that the HT-AggNet can improve HAR accuracy by increasing the network depth while maintaining low FLOPs.

An AMD Ryzen 7 5800X 8-Core Processor was used to estimate the response time of our HT-AggNet. The measurements were repeated 1000 times, and the minimum, maximum, and mean values were recorded. Table 9 lists the inference times of the two networks with the window length set to 512 (5.12 s), sliding one instance at a time. The HT-AggNet exhibited a faster response time than the Shallow ConvNet. In addition, the response times of HT-AggNets with different network depths were almost the same, although the network depth was varied by four times. Thus, these measurements show that deeper networks can fulfill the real-time requirements with improved accuracy.

Specifically, an example that HT-AggNet conducts an inference for each segment is shown in Fig. A.7 of Appendix. Given a segment $X \in \mathbb{R}^{512,31}$ from the same setup that collects the PAMAP2 dataset ($T = 512, M = 31$), HT-AggNet extracts the first features $H \in \mathbb{R}^{256,128}$ via a convolutional and pooling layer, resulting in $512 \times 5 \times 31 \times 128 \approx 10.2M$ multiplications. After that, the hierarchical temporal aggregation generates the last features $F \in \mathbb{R}^{128}$, resulting in $256 \times 5 \times (128/8) \times (128/8) \times 8 + 256 \times 128 \times (128/8) \times 8 \approx 6.8M$ multiplications. As a result, the inference of HT-AggNet costs $\approx 17M$ multiplications. On the other hand, Shallow ConvNet costs $512 \times 16 \times 31 \times 196 = 49.8M$ multiplications only to extract the first features. Consequently, our HT-AggNet has the benefit of maintaining low computational costs regardless of the number of layers because most of the computations in CNNs run when extracting the first features.

5. Conclusion

In this study, an effective neural network called HT-AggNet is proposed for real-time HAR from wearable sensors. In particular, our HT-AggNet has a near-zero-cost layer stacking strategy, in which layers can be stacked with only a negligible increase in computational costs. In addition, a temporal glance convolution is introduced, which enables the network to encode global temporal information without recurrent layers. Our experimental results show that HT-AggNets can achieve state-of-the-art performance on low-latency HAR datasets.

Future studies should be conducted to overcome the limitations of the proposed method. For example, HT-AggNets may improve the search for hardware-aware neural architectures by determining the appropriate network depth based on an optimal trade-off. In addition,

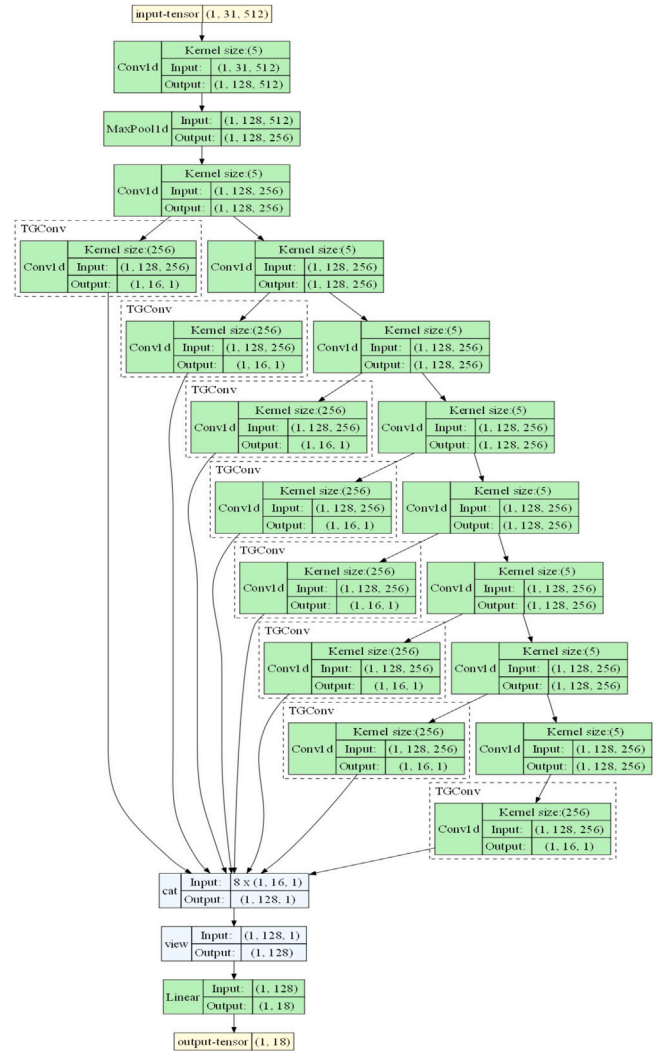


Fig. A.7. Network architectures of HT-AggNet.

HT-AggNet can be used for other applications. The near-zero-cost layer stacking may be useful in applications that should be run on an independent devices of which computational capability is fixed and hard to change.

CRedit authorship contribution statement

Jaegyun Park: Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Conceptualization. **Dae-Won Kim:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Jaesung Lee:** Writing – review & editing, Methodology, Investigation, Formal analysis.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jaesung Lee reports financial support was provided by Institute for Information Communication Technology Planning and Evaluation. Dae-Won Kim reports financial support was provided by National Research Foundation of Korea.

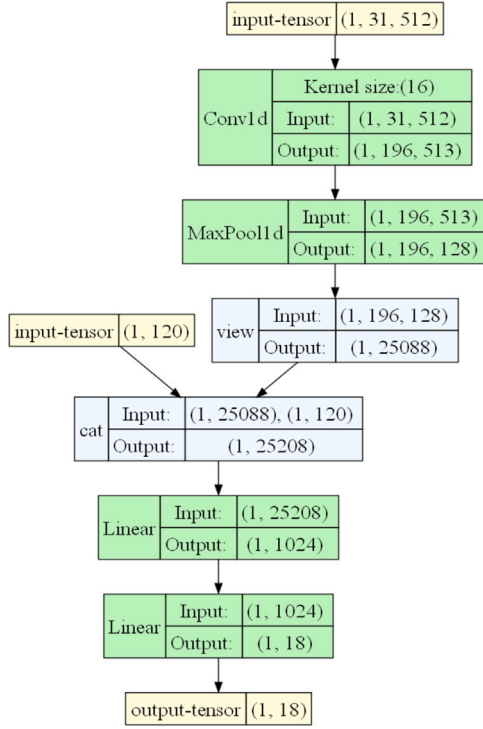


Fig. A.8. Network architectures of Shallow ConvNet.

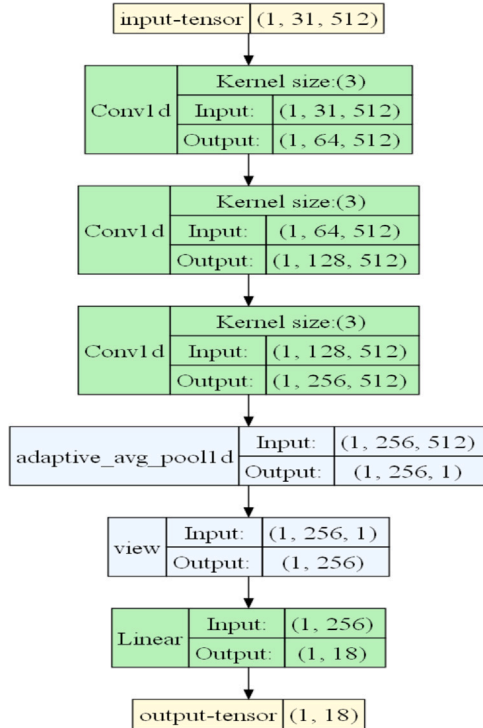


Fig. A.9. Network architectures of RepHAR.

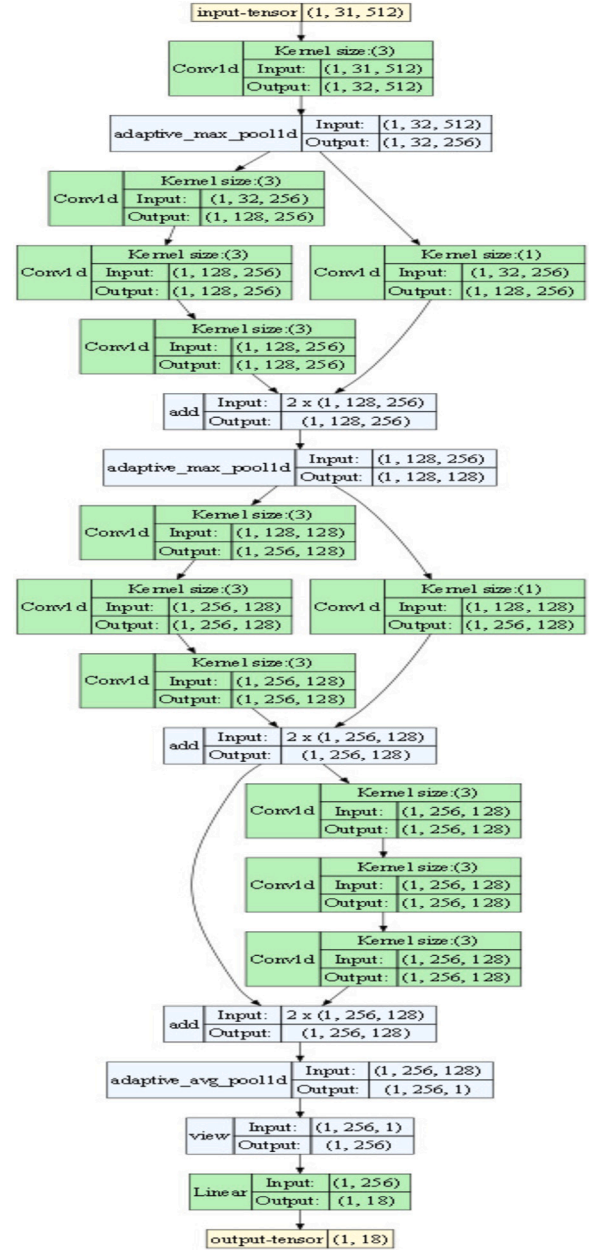


Fig. A.10. Network architectures of GTSNet.

Acknowledgments

This research was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2021-0-01341, Artificial Intelligence Graduate School Program (Chung-Ang University)) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2023R1A2C1006745).

Appendix. Details on network architectures

Figs. A.7–A.14 describe the network architectures used in our experiments. Hyperparameters for all baseline networks were set based on original papers. In addition, we roughly describe the time complexity for the baseline networks.

From Eq. (4), HT-AggNet achieved the time complexity of $\mathcal{O}(TD_kMN)$ via a grouped convolution and fixing the number of

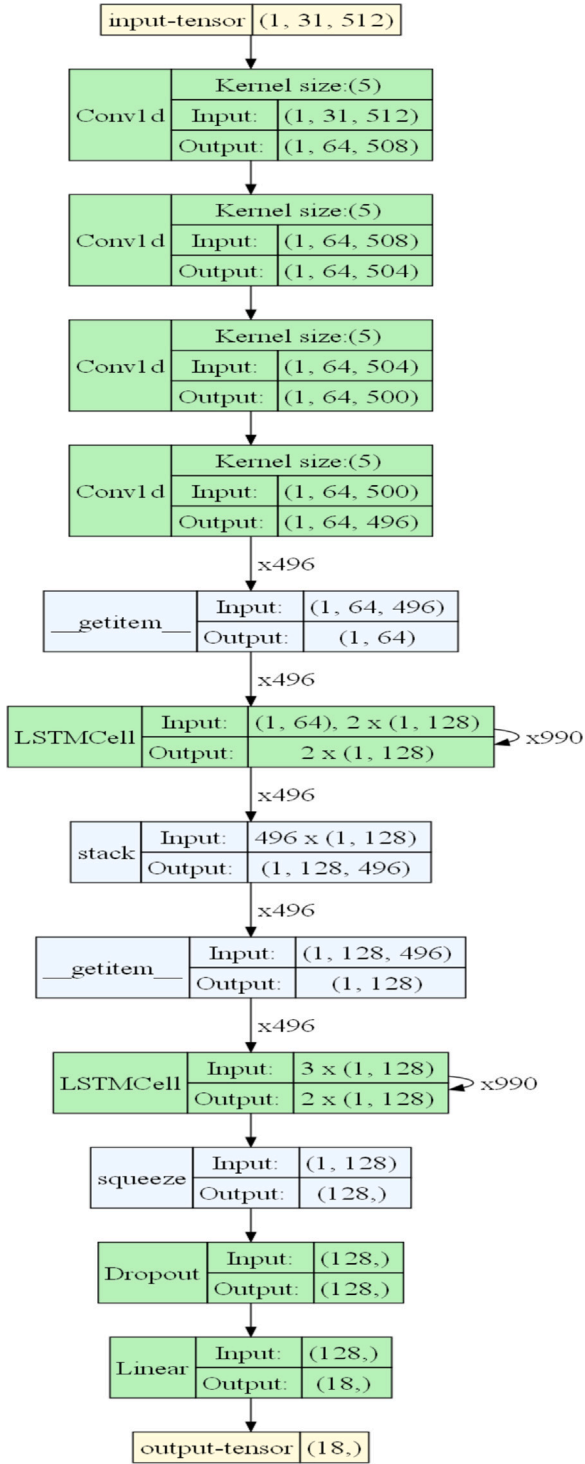


Fig. A.11. Network architectures of DeepConvLSTM.

last features. Shallow ConvNet includes only one convolutional layer. Therefore, its time complexity is $\mathcal{O}(TD_kMN)$. Furthermore, its FLOPs depend heavily on the shape of input signal patterns because of the flattening and statistical features. Meanwhile, RepHAR is re-parameterized at the inference step, as shown in Fig. A.9. Consistent with Eq. (4), its time complexity $\mathcal{O}(TD_kMNL)$. Similarly, MTS-ConvNet and GTSNet

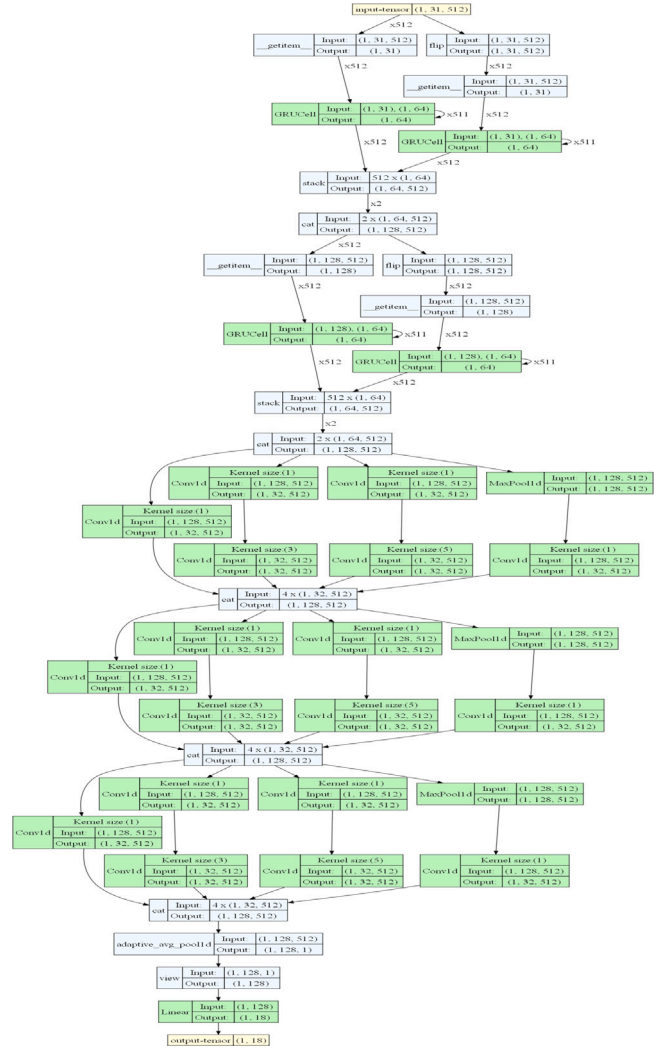


Fig. A.12. Network architectures of Bi-GRU-I.

have the time complexity of $\mathcal{O}(TNL(D_k + M/D_k))$ and $\mathcal{O}(TLM \log(N))$ by replacing standard convolutions with their efficient variants, respectively. Consequently, these models, except for HT-AggNet and Shallow ConvNet, exhibit higher FLOPs as the network depth increases.

As shown in Figs. A.11 and A.12, DeepConvNet and Bi-GRU-I consist of convolution and recurrent layers. The time complexity of a recurrent layer is determined by the number of input features, hidden units, and gates. In HAR, since each feature map has one dimension, we can write the number of input features as the term M and the number of hidden units as the term N without loss of generality. Therefore, the time complexity of calculating an output of each gate is $\mathcal{O}(TN(M + N))$. Given the number of gates G , the time complexity of $(L - l)$ recurrent layers is formalized as $\mathcal{O}(TGN(M + N)(L - l))$. As a result, these models have the time complexity of $\mathcal{O}(TD_kMNL + TGN(M + N)(L - l))$. Meanwhile, the time complexity of T-ResNet and T-FCN are $\mathcal{O}(TD_kMNL)$. Their architectures have more layers and channels, respectively, resulting in higher FLOPs. Unlike these networks, our HT-AggNet allows its network architecture to deepen with a negligible increase in FLOPs.

Data availability

The data that has been used is confidential.

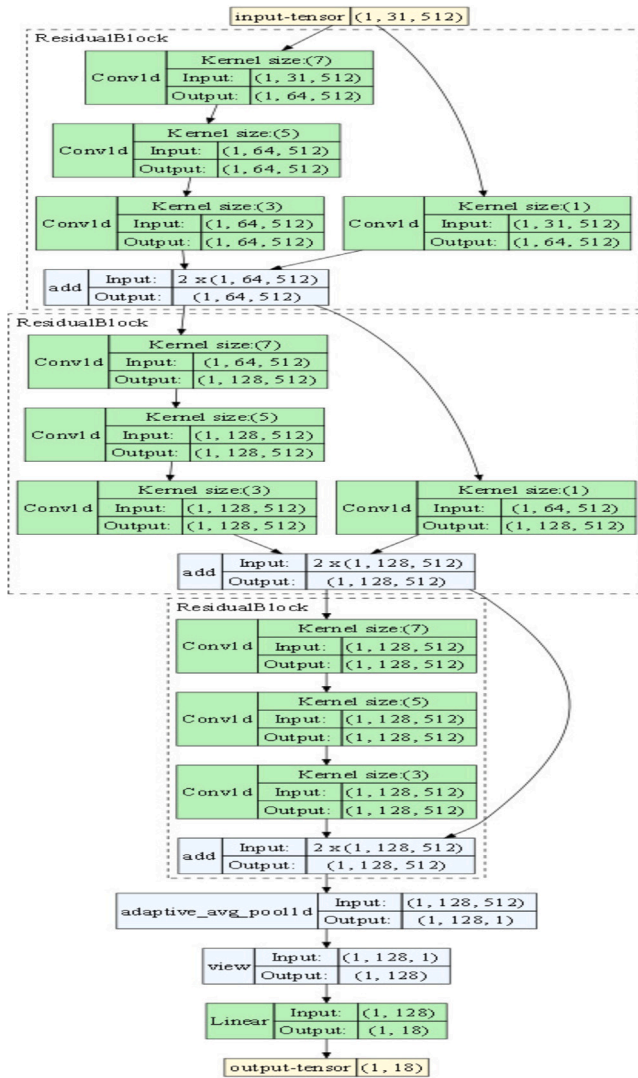


Fig. A.13. Network architectures of T-ResNet.

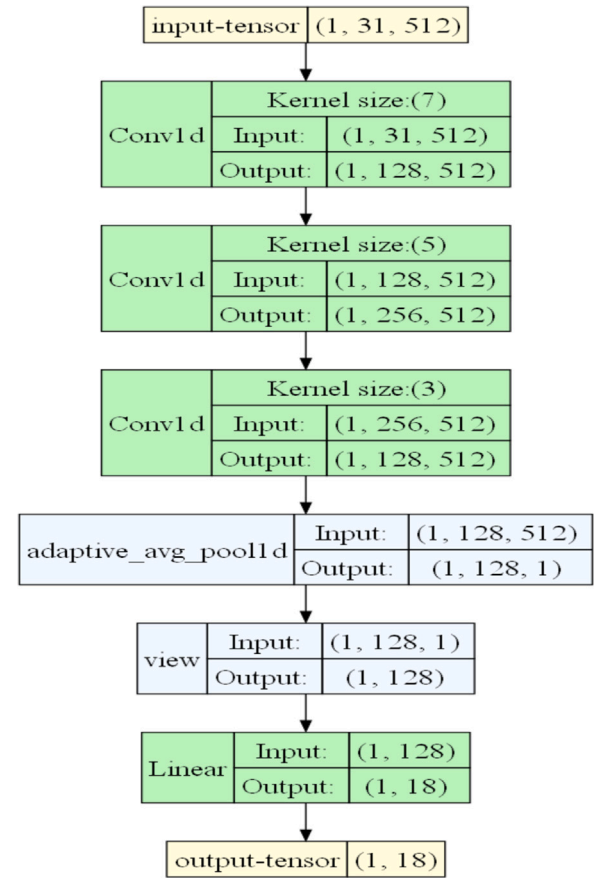


Fig. A.14. Network architectures of T-FCN.

References

- Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J., 2013. A public domain dataset for human activity recognition using smartphones. In: Proc. 21st Int. Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn.. ESANN, pp. 437–442.
- Baños, O., Damas, M., Pomares, H., Rojas, I., Tóth, M.A., Amft, O., 2012. A benchmark dataset to evaluate sensor displacement in activity recognition. In: Proc. ACM Conf. Ubiquitous Comput.. pp. 1026–1035.
- Barshan, B., Yüsek, M.C., 2014. Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units. Comput. J. 57 (11), 1649–1667.
- Bianchi, V., Bassoli, M., Lombardo, G., Fornacciari, P., Mordonini, M., De Munari, I., 2019. IoT wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment. IEEE Internet Things J. 6 (5), 8553–8562.
- Bianchini, M., Scarselli, F., 2014. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. IEEE Trans. Neural Netw. Learn. Syst. 25 (8), 1553–1565.
- Bulling, A., Blanke, U., Schiele, B., 2014. A tutorial on human activity recognition using body-worn inertial sensors. ACM Comput. Surv. (CSUR) 46 (3), 1–33.
- Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S.T., Tröster, G., Millán, J.d.R., Roggen, D., 2013. The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. Pattern Recognit. 34 (15), 2033–2042.
- Chen, Y., Xue, Y., 2015. A deep learning approach to human activity recognition based on single accelerometer. In: Proc. IEEE Int. Conf. Syst., Man, Cybern.. IEEE, pp. 1488–1492.

- Chen, K., Zhang, D., Yao, L., Guo, B., Yu, Z., Liu, Y., 2021. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. ACM Comput. Surv. (CSUR) 54 (4), 1–40.
- Dang, L.M., Min, K., Wang, H., Piran, M.J., Lee, C.H., Moon, H., 2020. Sensor-based and vision-based human activity recognition: A comprehensive survey. Pattern Recognit. 108, 107561.
- Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.-A., 2019. Deep learning for time series classification: a review. Data Min. Knowl. Discov. 33 (4), 917–963.
- Gao, W., Zhang, L., Huang, W., Min, F., He, J., Song, A., 2021. Deep neural networks for sensor-based human activity recognition using selective kernel convolution. IEEE Trans. Instrum. Meas. 70, 1–13.
- Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N., 2017. Convolutional sequence to sequence learning. In: International Conference on Machine Learning. PMLR, pp. 1243–1252.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. CVPR, pp. 770–778.
- He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., Li, M., 2019. Bag of tricks for image classification with convolutional neural networks. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.. CVPR, pp. 558–567.
- Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. CVPR, pp. 4700–4708.
- Huang, W., Zhang, L., Gao, W., Min, F., He, J., 2021. Shallow convolutional neural networks for human activity recognition using wearable sensors. IEEE Trans. Instrum. Meas. 70, 1–11.
- Hussain, A., Hussain, T., Ullah, W., Baik, S.W., 2022. Vision transformer and deep sequence learning for human activity recognition in surveillance videos. Comput. Intell. Neurosci. 2022, 3454167.
- Hussein, N., Gavves, E., Smeulders, A.W., 2019. Timeception for complex action recognition. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.. CVPR, pp. 254–263.
- Ignatov, A., 2018. Real-time human activity recognition from accelerometer data using convolutional neural networks. Appl. Soft Comput. 62, 915–922.
- Jiang, W., Yin, Z., 2015. Human activity recognition using wearable sensors by deep convolutional neural networks. In: Proc. 23rd ACM Int. Conf. Multimedia. pp. 1307–1310.

- Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. In: Proc. Int. Conf. Learn. Repr.. ICLR, pp. 1–15.
- Lee, D., Lee, S., Yu, H., 2021. Learnable dynamic temporal pooling for time series classification. In: Proc. AAAI Conf. Artif. Intell.. Vol. 35, pp. 8288–8296.
- Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z., 2015. Deeply-supervised nets. In: Proc. Artif. Intell. Statist.. PMLR, pp. 562–570.
- Lentzas, A., Vrakas, D., 2020. Non-intrusive human activity recognition and abnormal behavior detection on elderly people: A review. *Artif. Intell. Rev.* 53 (3), 1975–2021.
- Li, S., Sun, Y., Yen, G.G., Zhang, M., 2021. Automatic design of convolutional neural network architectures under resource constraints. *IEEE Trans. Neural Netw. Learn. Syst.*
- Luwe, Y.J., Lee, C.P., Lim, K.M., 2022. Wearable sensor-based human activity recognition with hybrid deep learning model. *Informatics* 9 (3), 56.
- Micucci, D., Mobilio, M., Napolitano, P., 2017. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Appl. Sci.* 7 (10), 1101.
- Mikolov, T., et al., 2012. Statistical Language Models Based on Neural Networks. Vol. 80, Presented at Google, Mountain View, 2nd Apr. (26).
- Mim, T.R., Amatullah, M., Afreen, S., Yousuf, M.A., Uddin, S., Alyami, S.A., Hasan, K.F., Moni, M.A., 2023. GRU-INC: An inception-attention based approach using GRU for human activity recognition. *Expert Syst. Appl.* 216, 119419.
- Nutter, M., Crawford, C.H., Ortiz, J., 2018. Design of novel deep learning models for real-time human activity recognition with mobile phones. In: Proc. IEEE Int. Joint Conf. Neural Netw.. IJCNN, IEEE, pp. 1–8.
- Nweke, H.F., Teh, Y.W., Al-Garadi, M.A., Alo, U.R., 2018. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Syst. Appl.* 105, 233–261.
- Ordóñez, F.J., Roggen, D., 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 16 (1), 115.
- Park, J., Lim, W.-S., Kim, D.-W., Lee, J., 2022. Multitemporal sampling module for real-time human activity recognition. *IEEE Access* 10, 54507–54515.
- Park, J., Lim, W.-S., Kim, D.-W., Lee, J., 2023. GTSNet: Flexible architecture under budget constraint for real-time human activity recognition from wearable sensor. *Eng. Appl. Artif. Intell.* 124, 106543.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. Pytorch: An imperative style, high-performance deep learning library. In: Proc. Adv. Neural Inf. Proces. Syst.. NIPS, Vol. 32, pp. 8026–8037.
- Prati, A., Shan, C., Wang, K.I.-K., 2019. Sensors, vision and networks: From video surveillance to activity recognition and health monitoring. *J. Ambient. Intell. Smart Env.* 11 (1), 5–22.
- Qi, J., Yang, P., Waraich, A., Deng, Z., Zhao, Y., Yang, Y., 2018. Examining sensor-based physical activity recognition and monitoring for healthcare using internet of things: A systematic review. *J. Biomed. Inform.* 87, 138–153.
- Qiu, S., Zhao, H., Jiang, N., Wang, Z., Liu, L., An, Y., Zhao, H., Miao, X., Liu, R., Fortino, G., 2022. Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges. *Inf. Fusion* 80, 241–265.
- Ramanujam, E., Perumal, T., Padmavathi, S., 2021. Human activity recognition with smartphone and wearable sensors using deep learning techniques: A review. *IEEE Sens. J.* 21 (12), 13029–13040.
- Ravi, D., Wong, C., Lo, B., Yang, G.-Z., 2016. A deep learning approach to on-node sensor data analytics for mobile or wearable devices. *IEEE J. Biomed. Heal. Inform.* 21 (1), 56–64.
- Reiss, A., Stricker, D., 2012. Introducing a new benchmarked dataset for activity monitoring. In: Proc. IEEE 16th Int. Symp. Wearable Comput.. ISWC, IEEE, pp. 108–109.
- Sikder, N., Nahid, A.-A., 2021. KU-HAR: An open dataset for heterogeneous human activity recognition. *Pattern Recognit.* 146, 46–54.
- Sze, V., Chen, Y.-H., Yang, T.-J., Emer, J.S., 2017. Efficient processing of deep neural networks: A tutorial and survey. *Proc. IEEE* 105 (12), 2295–2329.
- Tang, Y., Zhang, L., Min, F., He, J., 2022. Multiscale deep feature learning for human activity recognition using wearable sensors. *IEEE Trans. Ind. Electron.* 70 (2), 2106–2116.
- Teng, Q., Tang, Y., Hu, G., 2023. RepHAR: Decoupling networks with accuracy-speed tradeoff for sensor-based human activity recognition. *IEEE Trans. Instrum. Meas.* 72, 1–11.
- Tong, L., Ma, H., Lin, Q., He, J., Peng, L., 2022. A novel deep learning bi-gru-i model for real-time human activity recognition using inertial sensors. *IEEE Sens. J.* 22 (6), 6164–6174.
- Wan, S., Qi, L., Xu, X., Tong, C., Gu, Z., 2020. Deep learning models for real-time human activity recognition with smartphones. *Mob. Netw. Appl.* 25 (2), 743–755.
- Wang, Y., Cang, S., Yu, H., 2019a. A survey on wearable sensor modality centred human activity recognition in health care. *Expert Syst. Appl.* 137, 167–190.
- Wang, J., Chen, Y., Hao, S., Peng, X., Hu, L., 2019b. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognit.* 119, 3–11.
- Wang, Z., Yan, W., Oates, T., 2017. Time series classification from scratch with deep neural networks: A strong baseline. In: Proc. IEEE Int. Joint Conf. Neural Netw.. IJCNN, IEEE, pp. 1578–1585.
- Xu, C., Chai, D., He, J., Zhang, X., Duan, S., 2019a. InnoHAR: A deep neural network for complex human activity recognition. *IEEE Access* 7, 9893–9902.
- Xu, Q., Zhang, M., Gu, Z., Pan, G., 2019b. Overfitting remedy by sparsifying regularization on fully-connected layers of CNNs. *Neurocomputing* 328, 69–74.
- Zebin, T., Scully, P.J., Peek, N., Casson, A.J., Ozanyan, K.B., 2019. Design and implementation of a convolutional neural network on an edge computing smartphone for human activity recognition. *IEEE Access* 7, 133509–133520.
- Zhang, S., Li, Y., Zhang, S., Shahabi, F., Xia, S., Deng, Y., Alshurafa, N., 2022. Deep learning in human activity recognition with wearable sensors: A review on advances. *Sensors* 22 (4), 1476.
- Zhang, X., Wang, J., Wang, T., Jiang, R., 2021. Hierarchical feature fusion with mixed convolution attention for single image dehazing. *IEEE Trans. Circuits Syst. Video Technol.* 32 (2), 510–522.
- Zhang, X., Zhou, X., Lin, M., Sun, J., 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.. CVPR, pp. 6848–6856.