IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 5, 2022, accepted May 17, 2022, date of publication May 20, 2022, date of current version May 26, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3176606

Multitemporal Sampling Module for Real-Time Human Activity Recognition

JAEGYUN PARK[1], WON-SEON LIM[1], DAE-WON KIM[1], (Member, IEEE), AND JAESUNG LEE[2]

[1]School of Computer Science and Engineering, Chung-Ang University, Dongjak-Gu, Seoul 06974, Republic of Korea
[2]Department of Artificial Intelligence, Chung-Ang University, Dongjak-Gu, Seoul 06974, Republic of Korea

Corresponding authors: Dae-Won Kim (dwkim@cau.ac.kr) and Jaesung Lee (jslee.cau@gmail.com)

This work was supported in part by Chung-Ang University Research Grant, in 2021; and in part by the National Research Foundation of Korea (NRF) Grant through the Korea Government (MSIT) under Grant 2020R1A2C101357511.

ABSTRACT Human activity recognition, which recognizes human activities from time-series signals collected by sensors, is an important task in human-centered intelligent systems such as in healthcare and smart vehicles. In these applications, rapid response of the system is necessary because critical events such as an elderly person falling or drowsy driving require immediate action. A straightforward approach to achieving this requirement is to reduce the amount of information the model must handle. To this end, traditional studies have attempted to abstract the original signal by sampling it with a pre-defined interval. However, it is difficult to achieve the best efficiency because the ideal sampling interval is unknown in advance. In this study, we propose a multi-temporal sampling module that allows the neural networks to consider multiple sampling intervals simultaneously. Experiments on four benchmark datasets showed that the proposed model achieved the best F1 score over seven conventional models under the computation budget of 10M multiply-accumulate operations. Especially, an experiment on PAMAP2 dataset demonstrated that the proposed model can achieve the best trade-off between efficiency and accuracy when the input signal is oversampled at a high sampling frequency. In addition, the proposed model achieved $\sim 1,000\times$ improvement with respect to model size compared to the conventional methods.

INDEX TERMS Activity recognition, real-time systems, neural networks, signal sampling.

I. INTRODUCTION

Human Activity Recognition (HAR) aims to identify the daily activities of humans from time-series signals collected by sensors [1]. In most HAR studies, machine learning approaches have reported high performance [2]–[4] but require sophisticated feature engineering that incurs a lot of expert knowledge and time. For this reason, recent studies have increasingly considered Deep Learning (DL) models because they yield a superior learning performance based on how they automatically engineer features [5], [6]. As most DL models have high latency due to their complex architectures, 1D Convolutional Neural Networks (CNNs), which are supported by various hardware accelerators [7], are often considered to achieve real-time HAR [1], [6], [8].

In traditional HAR [9], [10], the original signal is sampled according to a pre-defined interval to improve efficiency, because the signal is typically oversampled [11]. Conventional CNNs for HAR, alternatively, can abstract the signal or feature map by pooling layers [12], [13]. However, it is difficult to achieve the best efficiency because the ideal sampling interval is unknown in advance. A straightforward solution to solve this issue is to allow CNNs to consider multiple sampling intervals simultaneously, yet the stride of pooling is fixed and the pooling is applied after the vanilla convolutions have already consumed a significant computational cost [14], [15].

To address this issue, we design a novel Multi-Temporal Sampling (MTS) module. Specifically, the module contains a new convolution unit, namely Sparse Sampling Convolution (S2Conv), that takes the random permuted form of a diagonal matrix to abstract the information delivered from the prior layer based on the sampling. Multiple S2Conv units that work with different sampling intervals are integrated into our MTS module to consider multiple sampling intervals simultaneously. By stacking MTS modules, the information

The associate editor coordinating the review of this manuscript and approving it for publication was Felix Albu.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

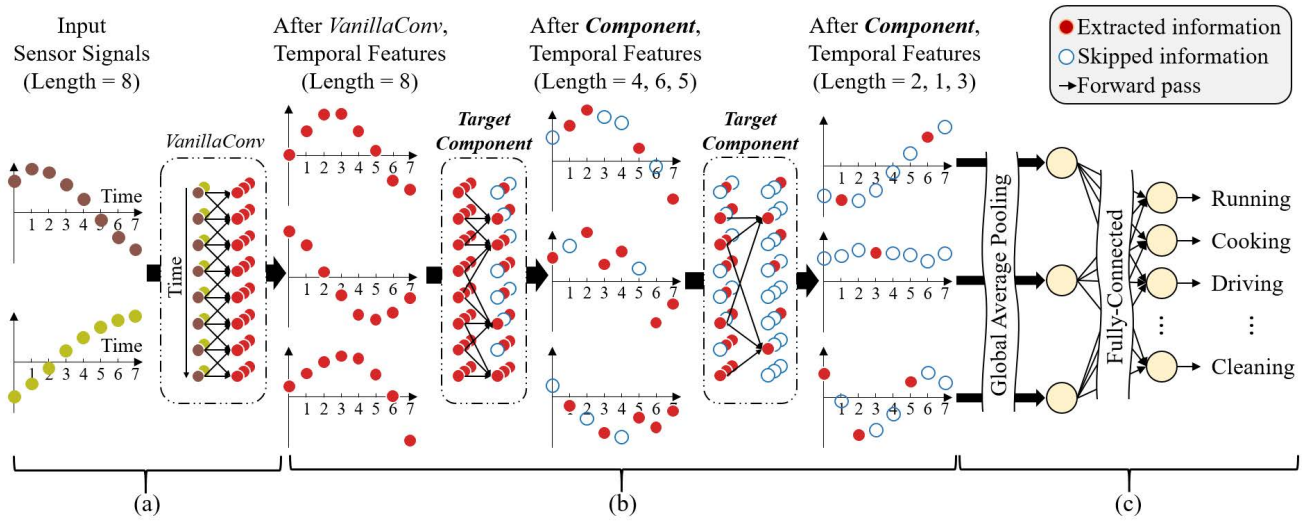


FIGURE 1. Design goal of a new component to replace a vanilla convolution (VanillaConv) unit. As temporal features pass through components, the amount of information to be handled is exponentially reduced, resulting in a reduction of computational cost and model size. (a): new feature maps are generated through VanillaConv from the original signals; (b): our components efficiently extract higher-level features by abstracting the information delivered from prior layers; (c): Based on the final features extracted by the components, an activity on each pattern is predicted. All types of convolutions come with rectified linear units.

to be handled can be reduced significantly as it passes through each layer. Our experiments demonstrate that our MTS-ConvNet achieves ∼1, 000× and ∼15× improvements with respect to model size and Multiply-Accumulate (MAC), respectively, compared to the existing methods without incurring a significant degradation of accuracy.

II. RELATED WORK

In recent years, many HAR studies have proposed large DL models to solve various challenges derived from real-world applications. Some studies proposed DL models for complex activity recognition, including composite activities [16], [17], concurrent activities [18], [19]. To address multi-modal sensory datasets, some other studies integrated an attention mechanism into the DL models [20]–[24]. These models, including vanilla convolutions and recurrent units, demonstrated their effectiveness, but they are insufficient to ensure the real-time response when the available computation budget is reduced due to low-cost devices or background APPs [25]. Especially, the recurrent units often require infeasible computational costs at edge implementation [7], [26], thus making the real-time HAR difficult.

In HAR literature, many comprehensive surveys have emphasized the importance of real-time applications [1], [6], [8], [27], [28]. Most prior studies have focused on sampling the signals and experimentally demonstrating that the ideal sampling interval depends on the type of activity [9], [11], [29]. Furthermore, Cheng *et al.* attempted to predict an instance-wise sampling interval [10]. Similarly, Yang *et al.* proposed an instance-wise dynamic sensor selection method [30]. However, these approaches, which focus on data acquisition, may result in irretrievable information loss.

Some early studies proposed DL models for real-time HAR, including pre-processing that transforms the input data from the time domain to the frequency domain [31], [32]. However, this complex pre-processing should be conducted without stopping, resulting in an increase in overhead. Ignatov first attempted to design a CNN architecture for real-time HAR without any transformation of the data [12]. Wan *et al.* proposed a CNN architecture including three convolutional layers for real-time HAR using accelerometers and gyroscopes [13]. However, these studies have not examined the effects of the sampling interval on performance for HAR.

III. MULTI-TEMPORAL SAMPLING MODULE

Our goal is to design a novel component to replace a vanilla convolution (VanillaConv) unit for real-time HAR. The component abstracts the information delivered from the prior layer, resulting in an improvement of efficiency. Figure 1 shows the forward pass of temporal information in the proposed neural network. Given original signals, a VanillaConv unit generates new temporal feature maps by learning intra-modality information or inter-modality correlations [8]. As the components are stacked along with the layer, the information to be handled is exponentially reduced. Because the ideal sampling interval may also vary when the information is abstracted, the component must consider multiple sampling intervals at each layer. To this end, we will design an integrated module as the basic component. Finally, a fully-connected (FC) layer predicts a human activity using extracted features.

A. VANILLA CONVOLUTION AND NOTATIONS

Let $I \in \mathbb{Z}^{T,M}$ and $O \in \mathbb{Z}^{T,N}$ be an input and output for VanillaConv, respectively, where T is the temporal resolution, and M and N are the numbers of feature maps. Given an input and a kernel $K \in \mathbb{Z}^{D_k,M,N}$ with kernel size D_k, the n-th

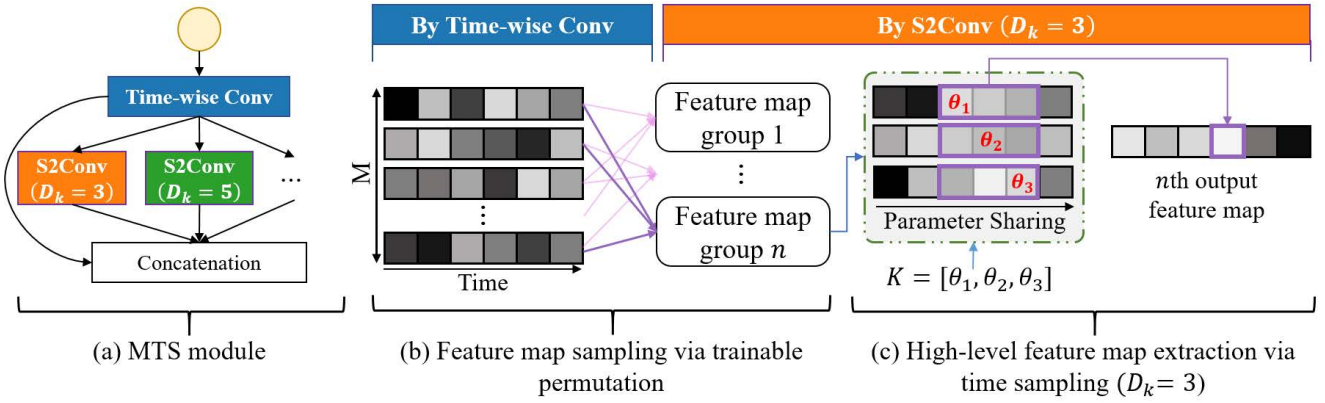


FIGURE 2. Illustration of key elements of our Multi-Temporal Sampling (MTS) module. (a): The MTS module includes Sparse Sampling Convolution (S2Conv) units with multiple sampling intervals, where a kernel size D_k is considered as a sampling interval; (b): some input feature maps are sampled and used to generate each output feature map, wherein a time-wise convolution learns about the group mapping; (c): our S2Conv efficiently extracts an output feature map per group via time sampling.

feature map of O at time t is calculated as follows [33]:

$$O_{t,n} = \sum_{i,m} K_{i,m,n} I_{t+\hat{i},m} \tag{1}$$

where $\hat{i} = i - \lfloor D_k/2 \rfloor$ is the re-centered time index and the computational cost is $T \times D_k \times M \times N$.

In the field of HAR, the purpose of convolution is to emphasize local temporal patterns closely related to the activities from input feature maps [8]. In this regard, VanillaConv can be divided into feature map-wise convolution and time-wise convolution, similarly to [34]. Given the kernel $K^1 \in \mathbb{Z}^{D_k,M}$ of feature map-wise convolution, the m-th feature map at time t is calculated as

$$\hat{O}_{t,m} = \sum_{i} K^1_{i,m} I_{t+\hat{i},m} \tag{2}$$

Next, for a given kernel $K^2 \in \mathbb{Z}^{M,N}$ of time-wise convolution, the n-th feature map is calculated as

$$O_{t,n} = \sum_{m} K^2_{m,n} \hat{O}_{t,m} \tag{3}$$

where the computational cost is $T \times M \times (D_k + N)$. Based on this decomposition, we integrate two sampling processes into VanillaConv, i.e., time sampling and feature map sampling, as shown in Figure 2.

B. SPARSE SAMPLING CONVOLUTION

We design an efficient convolution unit that extracts higher-level features at a specific sampling interval within MTS modules. In Eq. (3), the feature map sampling is applied to $\hat{O}$. Let $G^n \in \mathbb{Z}^{T,M_s}$ be the sampled feature maps used to generate the n-th output feature map $O_{:,n}$, where $M_s \ll M$ is the number of the sampled feature maps. As a result, the size of the kernel K^2 will be reduced, i.e., $K^2 \in \mathbb{Z}^{M_s,N}$. Meanwhile, the time sampling is applied to K^1 in Eq. (2). We consider the kernel size D_k as the sampling interval and thus parameters of K^1 are sampled with D_k. The sampled kernel $\tilde{K}^1$ can be written as

$$\tilde{K}^1_{i,m} = \begin{cases} \theta, & \text{if } i = i_m \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where θ is a trainable parameter and i_m is feature map-dependent index that has only one value in $\tilde{K}^1_{:,m} \in \mathbb{Z}^{D_k}$.

To extract temporal information, $O_{t,n}$ should be computed by interacting with features of the adjacent times. To this end, the time sampling can be conducted by spreading i_m of Eq. (4) across adjacent times as shown in Figure 2(c), while satisfying the following condition:

$$\forall i : \sum_{m}^{M} |\tilde{K}^1_{i,m}| \neq 0 \tag{5}$$

Furthermore, the feature map sampling can be directly applied to I because the order of input feature maps is unchanged by $\tilde{K}^1$. Additionally, the parameters of $\tilde{K}^1$ can be replaced with one by the associative law of multiplication between them and the parameters of K^2 of Eq. (3). As a result, two kernels can be integrated into $\tilde{K} \in \mathbb{Z}^{M_s,N}$ without additional computation.

Meanwhile, it is difficult to find the optimal M_s with respect to a trade-off between accuracy and efficiency. Therefore, we simply set M_s to D_k, which is the minimum number of parameters within $\tilde{K}_{:,n}$ that satisfy Eq. (5). Consequently, we propose a Sparse Sampling Convolution (S2Conv) of which the computational cost is $T \times D_k \times N$. It extracts the n-th output feature map at time t as follow:

$$O_{t,n} = \sum_{i}^{D_k} \tilde{K}_{i,n} G^n_{t+\hat{i},i} \tag{6}$$

C. PERMUTATION OF FEATURE MAPS

To avoid loss of information resulting from the random sampling of feature maps, the following issues should be considered. First, mapping M input feature maps to N sample groups generates a large search space, resulting in ${}_M C_{D_k}$

Algorithm 1 Multi-Temporal Sampling Module

1: **Input:** $I \in \mathbb{Z}^{T,M}$, S^k; ▷ the set of the kernel sizes S^k
2: **Output:** $O \in \mathbb{Z}^{T,N}$;
3: $N_b \leftarrow \lfloor \frac{N}{|S^k|+1} \rfloor$ ▷ # of output channels for branches
4: **for** $n = 1$ to N_b **do**
5: **for** $t = 1$ to T **do**
6: $\tilde{I}_{t,n} \leftarrow \sum_m^M K_{m,n} I_{t,m}$ ▷ TWConv
7: **end for**
8: **end for**
9: $\tilde{I} \leftarrow \text{ReLU}(\text{BN}(\tilde{I}))$ ▷ the batch normalization BN
10: $b \leftarrow 0$
11: **for** $D_k \in S^k$ **do** ▷ the user-defined parameter
12: $b \leftarrow b + 1$
13: $\tilde{O}^b \leftarrow$ calculate **S2Conv**$(\tilde{I}, D_k)$ ▷ use Eq. (6)
14: $\tilde{O}^b \leftarrow \text{ReLU}(\text{BN}(\tilde{O}^b))$ ▷ $\tilde{O}^b \in \mathbb{Z}^{T,N_b}$
15: **end for**
16: $O \leftarrow \text{Concat}(\tilde{I}, \tilde{O})$ ▷ concatenate them at channel axis

possible cases per group. Second, the order of sampled feature maps within G^n should be considered because their dependency can vary across time. To handle these issues, we permute the input feature maps and map them to each sample group in order. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be permutation functions for grouping and an arrangement of feature maps within each group, respectively. Based on this, Eq. (6) can be rewritten as follows:

$$\begin{aligned} O &= (\mathcal{K} \circ \mathcal{P}_2 \circ \mathcal{P}_1)(I) \\ &= \mathcal{K}(IP) \end{aligned} \tag{7}$$

where $\circ$ is a function composition and $\mathcal{K}$ is computed by Eq. (6). Because $\mathcal{P}_1$ and $\mathcal{P}_2$ can be computed through the permutation matrix, the composition of them can be conducted by multiplying a permutation matrix $P \in \mathbb{Z}^{M,M}$ to $I \in \mathbb{Z}^{T,M}$.

However, it is difficult for P to be optimized by stochastic gradient descent because it has discrete values. This issue can be resolved by training a permuted time-wise convolution directly [35]. Specifically, the input feature maps to each module are permuted by time-wise convolution. After that, feature maps can be simply sampled in sequence. Given a permuted input $\tilde{I} \in \mathbb{Z}^{T,M}$, when $M = N$, each feature map group G^n is sampled as follows:

$$G^n = \tilde{I}_{:,n-\lfloor D_k/2 \rfloor : n+\lfloor D_k/2 \rfloor} \tag{8}$$

The number of feature maps for the permuted input is controlled by the time-wise convolution within the MTS module. As a result, according to Eq. (6) and Eq. (8), our S2Conv unit substantially reduces the computational cost and model size compared with the VanillaConv via both the time sampling and feature map sampling.

D. CONSTRUCTING MODULE AND ARCHITECTURE

Built on the S2Conv units with multiple sampling intervals or kernel sizes D_k, we propose a Multi-Temporal Sampling (MTS) module. Algorithm 1 represents the pseudocode

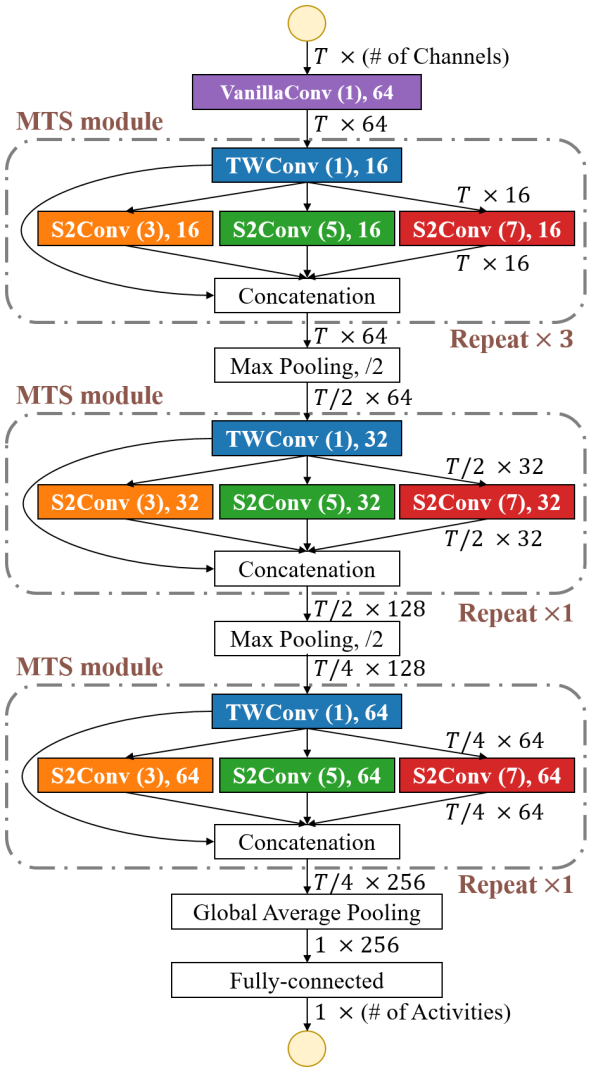


FIGURE 3. The neural architecture of our MTS-ConvNet. All types of convolutions are represented by the form of "convolution (kernel size), the number of output feature maps."

for the forward pass of our MTS module. Given the input feature maps and the set of the kernel sizes, its branches extract the same number of output feature maps (Line 3), where the set of the kernel sizes is the user-defined parameter. In Lines 4–7, the input feature maps are then permuted by the Time-Wise Convolution (TWConv). The permuted feature maps are fed into branches of MTS module with different D_k (Line 11–15). In Line 16, their outputs and the permuted input are concatenated. All convolutions come with a Batch Normalization (BN) and Rectified Linear Unit (ReLU); herein, the BN layers include biases.

In the MTS module, the time complexity is determined by computations of TWConv and S2Conv. By assuming that elements of S^k are an arithmetic sequence of which the first term and the common difference are three and two, respectively, their computations cost can be computed as follows:

$$(M + |S^k|(|S^k| + 2)) \frac{N}{|S^k| + 1} T \tag{9}$$

TABLE 1. Comparison results from four datasets. ▼/△ indicates that the corresponding model is significantly worse/better than our MTS-ConvNet based on paired *t*-test at 95% significance level on three metrics.

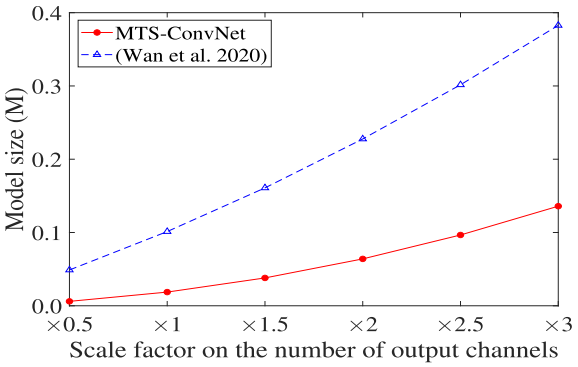
Model	Objective	UCI-HAR MACs	UCI-HAR Model size	UCI-HAR F1 score	WISDM MACs	WISDM Model size	WISDM F1 score
MTS-ConvNet (**Ours**)	Real-time HAR	1.089M	0.019M	0.940	0.500M	0.018M	0.864
Ignatov [12]	Real-time HAR	8.950M▼	6.490M▼	0.925▼	3.658M▼	3.068M▼	0.840▼
Wan et al. [13]		2.199M▼	0.228M▼	0.939	0.968M▼	0.161M▼	0.862
ResNet	Time-series classification	61.769M▼	0.483M▼	0.952△	28.862M▼	0.481M▼	0.893△
FCN		34.440M▼	0.269M▼	0.958△	15.983M▼	0.267M▼	0.880△
ResNext	Efficient design for image classification	27.466M▼	22.036M▼	0.941	12.878M▼	22.035M▼	0.869
MobileNetV2		11.997M▼	2.189M▼	0.925▼	5.972M▼	2.188M▼	0.861
SqueezeNet		5.338M▼	0.360M▼	0.920▼	2.146M▼	0.358M▼	0.845▼
Model	Objective	OPPORTUNITY MACs	OPPORTUNITY Model size	OPPORTUNITY F1 score	PAMAP2 MACs	PAMAP2 Model size	PAMAP2 F1 score
MTS-ConvNet (**Ours**)	Real-time HAR	2.301M	0.029M	0.849	5.174M	0.023M	0.754
Ignatov [12]	Real-time HAR	31.507M▼	8.292M▼	0.871△	75.906M▼	25.930M▼	0.674▼
Wan et al. [13]		9.749M▼	0.315M▼	0.857	14.375M▼	0.704M▼	0.718▼
ResNet	Time-series classification	80.431M▼	0.538M▼	0.879△	253.037M▼	0.496M▼	0.697▼
FCN		54.742M▼	0.367M▼	0.876△	149.228M▼	0.293M▼	0.725▼
ResNext	Efficient design for image classification	32.461M▼	22.069M▼	0.877△	109.857M▼	22.053M▼	0.772△
MobileNetV2		15.694M▼	2.214M▼	0.872△	48.596M▼	2.206M▼	0.759
SqueezeNet		11.694M▼	0.438M▼	0.832▼	27.011M▼	0.383M▼	0.710▼

Therefore, the time complexity of the MTS module is $O((|S^k| + \frac{M}{|S^k|})NT)$. Suppose that VanillaConv's D_k is a mean of S^k. Then, its time complexity can be computed as $O(|S^k|MNT)$. Because $M \gg |S^k| \geq 2$ generally, $|S^k|M > |S^k| + \frac{M}{|S^k|}$. Thus, our MTS module has lower time complexity than does VanillaConv.

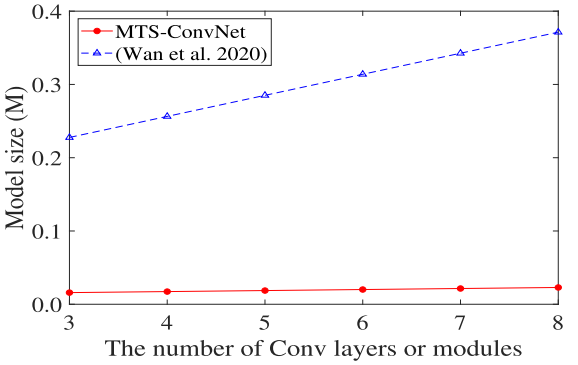
Built on the MTS module, we constructed our MTS-ConvNet architecture as shown in Figure 3. For brevity, we represent all type of convolution with a specific kernel size as "convolution (kernel size)". We used VanillaConv (1) to examine the temporal modeling capability of the MTS module and to improve the efficiency of MTS-ConvNet. Because a delayed reduction of resolution may lead to higher classification accuracy [36], the pooling layers were concentrated toward the end of the network. MTS-ConvNet for all datasets are trained in PyTorch [37] by Adam optimizer [38] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. Both the learning rate and weight decay are set as 0.0005. We train MTS-ConvNet for 500 epochs with a batch size of 128 on a 2080Ti Graphics Processing Unit (GPU).

IV. EXPERIMENTAL RESULTS

We conducted experiments using four benchmark datasets. The **UCI-HAR** [39] and **WISDM** [40] datasets include six activities such as "jogging" and "walking". They were collected at 50 and 20 Hz frequencies, and their segment lengths are 128 and 60, respectively. Also, the UCI-HAR dataset was collected from accelerometer and gyroscope, while the WISDM dataset was collected from the only accelerometer. The **OPPORTUNITY** [41] and **PAMAP2** [42] datasets contain 18 classes, including complex activities such as "playing soccer". For real-time HAR, we only considered the on-body sensors. They were collected at 30 and 100 Hz, and their segment lengths are 150 and 512, respectively.



(a) The cost of scaling the number of output channels.



(b) The cost of stacking VanillaConv layers or MTS modules.

FIGURE 4. Comparison between our MTS module and VanillaConv with respect to model size.

To evaluate the performance of our MTS-ConvNet for HAR, we used three metrics: Multiply-Accumulate (MAC) operation, model size, and F1 score. The MAC operation is the basic computation of neural networks, which takes the form $w \times x + b$. Therefore, we used MACs to measure the computations of the neural networks. Model size is typically

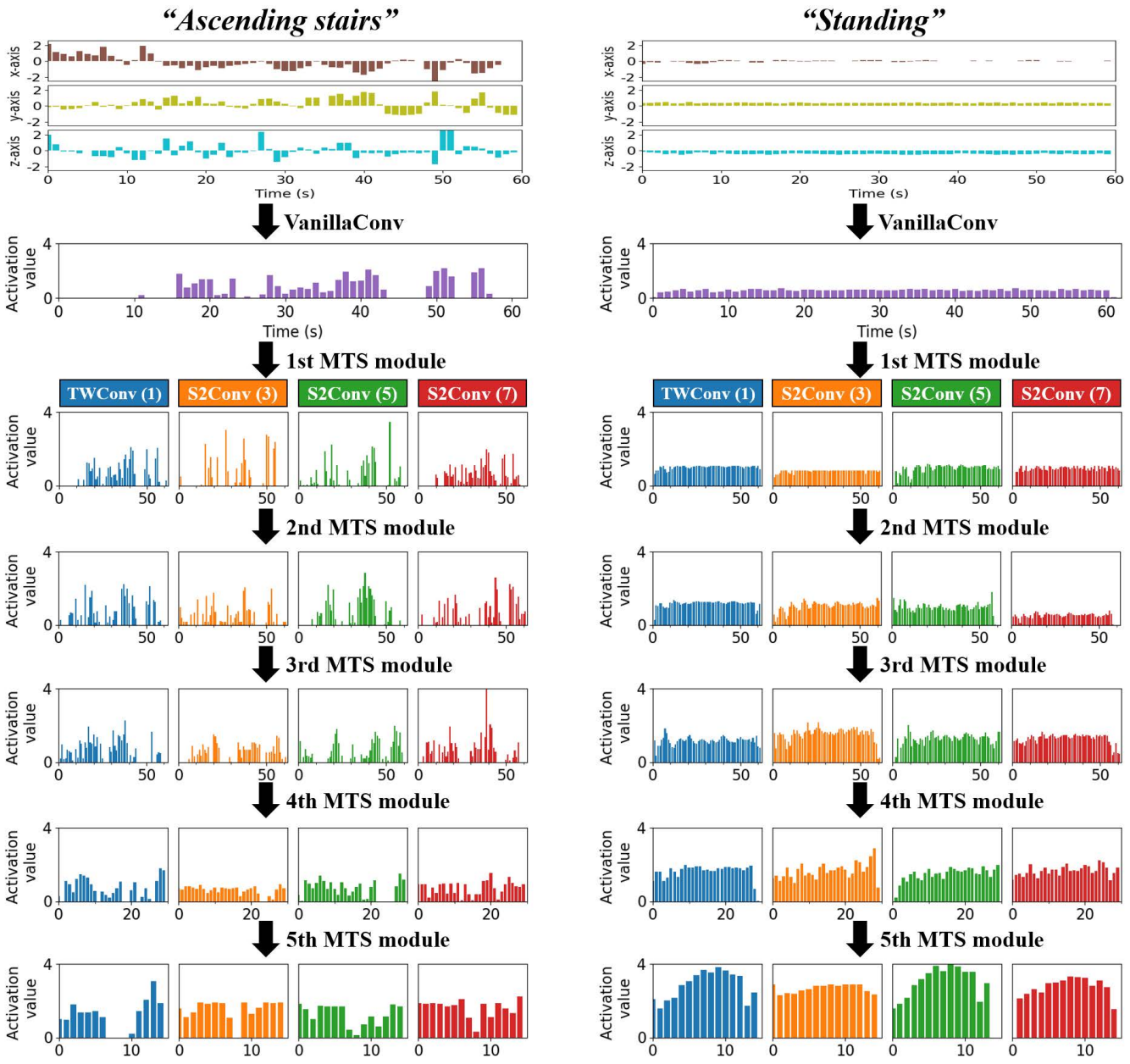


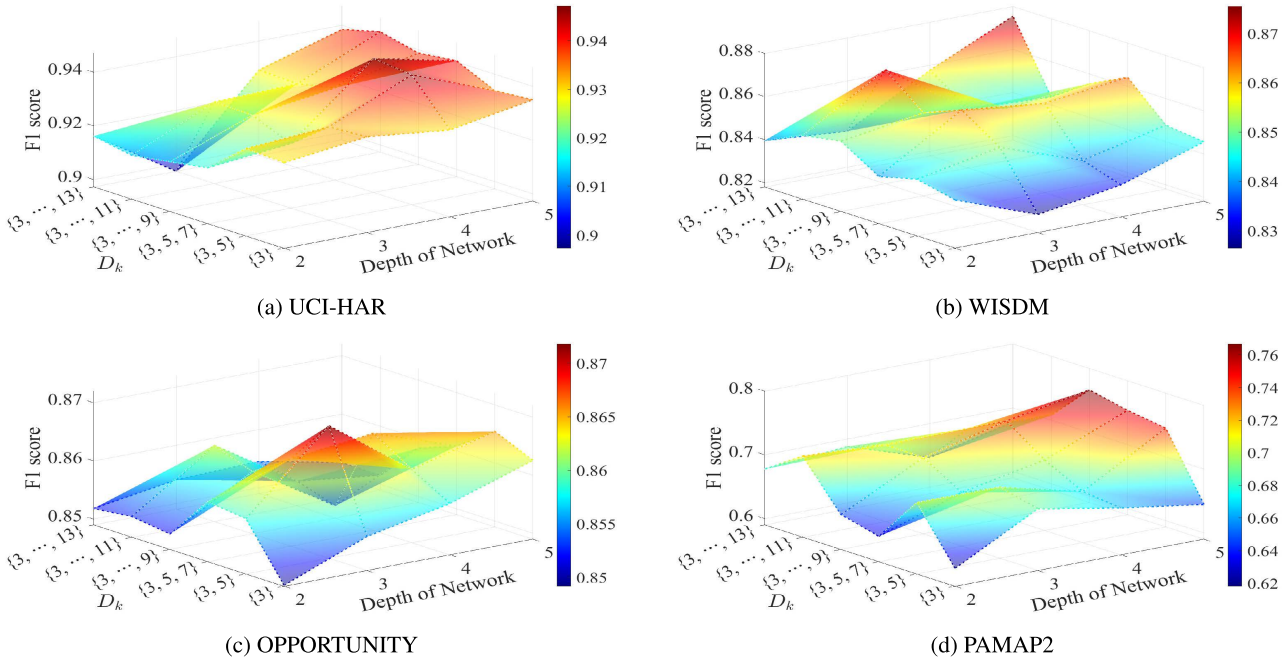
FIGURE 5. Comparison of feature maps extracted at different kernel sizes D_k for two activities, "Ascending stairs" and "Standing." Among the outputs of each operation, a feature map with the highest mean values across the time axis are shown.

used to measure the number of parameters in neural networks. Lastly, because HAR datasets often involve class imbalance issues [8], the F1 score is commonly used as an alternative to accuracy.

We adopted seven baseline networks, as follows.

- **Real-time HAR models.** Ignatov [12] introduced a CNN for real-time HAR that consists of a single convolutional and FC layer. In addition, Wan *et al.* [13] proposed a CNN that consists of three convolutional and two FC layers. Their experiments support baseline performance for real-time HAR models.
- **Time-Series Classification (TSC) models.** We adopted two CNNs that have reported significant success for the TSC problem. The 1D Residual Neural Network (ResNet) [43] and Fully Convolutional Network (FCN) [44] include eleven and three convolutional layers, respectively.
- **Efficient CNNs.** We also use three CNNs: MobileNetV2 [45], SqueezeNet [36] and ResNext [46]. These were designed to produce a computationally efficient model for image classification. To conduct experiments on HAR datasets, we replaced 2D operations with 1D operations.

Table 1 shows the overall comparison results of our MTS-ConveNet along with all the baseline networks. We obtained the average value by repeating the experiment 10 times.



(a) UCI-HAR (b) WISDM

(c) OPPORTUNITY (d) PAMAP2

FIGURE 6. Performance comparison according to the number of kernel sizes $|D_k|$ and the depth of the neural network on MTS-ConvNet. The depth of the neural network indicates the number of MTS modules which the network includes.

Our MTS-ConvNet exhibited comparable F1 scores while attaining the lowest MACs and model size. Specifically, MTS-ConvNet achieved higher F1 scores than existing real-time HAR models on the UCI-HAR, WISDM, and PAMAP2 datasets. Furthermore, MTS-ConvNets trained on the OPPORTUNITY and PAMAP2 datasets had smaller model sizes than real-time HAR models trained on the UCI-HAR and WISDM datasets, which makes real-time HAR on complex activities possible.

Meanwhile, ResNet and FCN with the highest MACs achieved the best F1-score on UCI-HAR, WISDM, and OPPORTUNITY datasets. These results indicate that a deeper network and more output channels can produce better F1 scores. However, ResNet and FCN, which require at least 23× MACs compared to MTS-ConvNet, can be insufficient to ensure the real-time response on wearable devices with various hardware specifications. On the other hand, our MTS-ConvNet outperformed ResNet and FCN in terms of the F1 score on PAMAP2 collected at 100 Hz even though 49× and 29× improvements with respect to MACs, respectively. This result indicates that the proposed method can achieve the best trade-off between efficiency and accuracy when the input signal is oversampled at a high sampling frequency.

V. ANALYSIS ON MODEL SIZE AND SAMPLING INTERVALS

Our MTS-ConvNet demonstrated substantial efficiency with respect to model size. Figure 4 shows a comparison result of the MTS-ConvNet and [13], which is a real-time HAR model built on VanillaConv using the UCI-HAR dataset. The two horizontal axes indicate a scale factor to control the number of output channels and the depth of VanillaConv units or MTS modules in each model, respectively. As shown in Figure 4, MTS-ConvNet has an efficient rate of increase in model size as the network becomes wider or deeper.

TABLE 2. Improvement of F1 score via hyper-parameter optimization of the MTS module.

	UCI-HAR	WISDM
D_k, Depth	{3, 5, 7}, 4	{3,5,7,9,11,13}, 5
MACs	1.089M ⟶ **0.917M**	0.500M ⟶ **0.427M**
Model size	0.019M ⟶ **0.017M**	0.018M ⟶ **0.015M**
F1 score	0.940 ⟶ **0.947**	0.864 ⟶ **0.876**
	OPPORTUNITY	**PAMAP2**
D_k, Depth	{3, 5}, 3	{3, 5, 7, 9}, 5
MACs	2.301M ⟶ 2.560M	5.174M ⟶ **5.063M**
Model size	0.029M ⟶ 0.034M	0.023M ⟶ **0.022M**
F1 score	0.849 ⟶ **0.872**	0.754 ⟶ **0.767**

To demonstrate the benefit of considering multiple sampling intervals, we compared feature maps extracted at different kernel size D_k as information for two activities passes through our MTS modules. For better visibility, we used the WISDM dataset, which was collected from only an accelerometer. As shown in Figure 5, the activity "Ascending stairs" has higher variations in input signals while "Standing" hardly has variations. For "Standing," the temporal features tended to be emphasized from all S2Conv units, regardless of the specific D_k. Alternatively, for "Ascending stairs," the feature maps extracted by S2Conv units with different D_k tended to have different waveforms (i.e., different abstracted information).

When a VanillaConv is replaced with an S2Conv, the amount of skipped information increases as D_k becomes

higher, resulting in better efficiency while accuracy may be degraded. If the MTS module includes only a single S2Conv (3), the efficiency gain is reduced, while if it includes only a single S2Conv (7), the accuracy can be degraded. However, the ideal D_k is unknown in advance as shown in Figure 5. Consequently, our MTS module can achieve a better trade-off between the efficiency gain and the accuracy by using multiple S2Conv units with diverse D_k.

Furthermore, we examined the impact of the number of kernel sizes $|D_k|$ along with a depth of network on the F1 score. As shown in Figure 6, MTS-ConvNet, which either uses more multiple kernel sizes or is deeper, tends to achieve a higher F1 score. Meanwhile, we found that the small number of filters in the first TWConv led to a poor F1 score on the OPPORTUNITY dataset, which was collected in a sensor-rich environment. This may be because 113 input channels are immediately compressed to 16 channels, resulting in a loss of information. Therefore, we modified the number of output channels of the first MTS module to 32 only for the OPPORTUNITY dataset. Finally, Table 2 shows the improved F1 scores.

VI. REAL-TIME ACTIVITY RECOGNITION

To estimate the actual response time of our MTS-ConvNet, we used a Samsung Galaxy S10 smartphone having an octa-core processor (2×2.7 GHz $+ 2 \times 2.3$GHz $+ 4 \times 1.9$ GHz) and 8GB RAM; herein, our implementation excluded the use of graphics processing units because the real-time HAR should be conducted without stopping as background APPs. Specifically, input signals are acquired in real-time from the smartphone's built-in accelerometer and gyroscope at 50 Hz. After that, our MTS-ConvNet, which was pretrained on the UCI-HAR dataset and embedded into the smartphone, continuously runs the activity recognition whenever the previous prediction is finished. As a result, a short YouTube demo video is available at https://youtu.be/Ie47soUp6Bs. The inference time of our MTS-ConvNet was measured between 20 and 45ms. Because the length of recognition intervals for the UCI-HAR dataset is 128 (2.56s segments), our model is sufficient to meet the real-time requirement.

VII. CONCLUSION

In this paper, we present a novel approach to integrating a traditional sampling process into a neural network for real-time human activity recognition. To this end, we introduce a sparse sampling convolution unit that allows neural networks to abstract the information delivered from the prior layer based on the sampling, resulting in improved efficiency. Furthermore, we propose a novel multi-temporal sampling module that contains the proposed convolution units with multiple kernel sizes, resulting in a better trade-off between the efficiency gain and the accuracy. The proposed module enables a sophisticated architecture that depends on various resource-limited sensor devices. Therefore, a promising study in the future would be a neural architecture search. For example, the trade-off between accuracy and efficiency can be optimized by directly measuring the latency and memory of sensor devices and reflecting them in the search process.

REFERENCES

[1] 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 Recognit.*, vol. 108, Dec. 2020, Art. no. 107561.
[2] X. Yang, S. A. Shah, A. Ren, N. Zhao, Z. Zhang, D. Fan, J. Zhao, W. Wang, and M. Ur-Rehman, "Freezing of gait detection considering leaky wave cable," *IEEE Trans. Antennas Propag.*, vol. 67, no. 1, pp. 554–561, Jan. 2019.
[3] X. Yang, L. Guan, Y. Li, W. Wang, Q. Zhang, M. U. Rehman, and Q. H. Abbasi, "Contactless finger tapping detection at C-band," *IEEE Sensors J.*, vol. 21, no. 4, pp. 5249–5258, Feb. 2021.
[4] H. Bi, M. Perello-Nieto, R. Santos-Rodriguez, and P. Flach, "Human activity recognition based on dynamic active learning," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 4, pp. 922–934, Apr. 2021.
[5] J. Huang, S. Lin, N. Wang, G. Dai, Y. Xie, and J. Zhou, "TSE-CNN: A two-stage end-to-end CNN for human activity recognition," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 1, pp. 292–299, Jan. 2020.
[6] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, Mar. 2019.
[7] T. Zebin, P. J. Scully, N. Peek, A. J. Casson, and K. B. Ozanyan, "Design and implementation of a convolutional neural network on an edge computing smartphone for human activity recognition," *IEEE Access*, vol. 7, pp. 133509–133520, 2019.
[8] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–40, May 2022.
[9] A. Khan, N. Hammerla, S. Mellor, and T. Plötz, "Optimising sampling rates for accelerometer-based human activity recognition," *Pattern Recognit. Lett.*, vol. 73, pp. 33–40, Apr. 2016.
[10] W. Cheng, S. Erfani, R. Zhang, and R. Kotagiri, "Learning datum-wise sampling frequency for energy-efficient human activity recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–8.
[11] L. Zheng, D. Wu, X. Ruan, S. Weng, A. Peng, B. Tang, H. Lu, H. Shi, and H. Zheng, "A novel energy-efficient approach for human activity recognition," *Sensors*, vol. 17, no. 9, p. 2064, 2017.
[12] I. Andrey, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Appl. Soft Comput.*, vol. 62, pp. 915–922, Jan. 2017.
[13] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, "Deep learning models for real-time human activity recognition with smartphones," *Mobile Netw. Appl.*, vol. 25, pp. 743–755, Dec. 2019.
[14] M. Xu, F. Qian, M. Zhu, F. Huang, S. Pushp, and X. Liu, "DeepWear: Adaptive local offloading for on-wearable deep learning," *IEEE Trans. Mobile Comput.*, vol. 19, no. 2, pp. 314–330, Feb. 2020.
[15] W. Chen, D. Xie, Y. Zhang, and S. Pu, "All you need is a few shifts: Designing efficient convolutional neural networks for image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7241–7250.
[16] L. Peng, L. Chen, Z. Ye, and Y. Zhang, "AROMA: A deep multi-task learning based simple and complex human activity recognition method using wearable sensors," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 2, pp. 1–16, 2018.
[17] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan, "InnoHAR: A deep neural network for complex human activity recognition," *IEEE Access*, vol. 7, pp. 9893–9902, 2019.
[18] T. Okita and S. Inoue, "Recognition of multiple overlapping activities using compositional CNN-LSTM model," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput., ACM Int. Symp. Wearable Comput.*, Sep. 2017, pp. 165–168.
[19] Y. Zhang, X. Li, J. Zhang, S. Chen, M. Zhou, R. A. Farneth, I. Marsic, and R. S. Burd, "CAR—A deep learning structure for concurrent activity recognition," in *Proc. 16th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2017, pp. 299–300.
[20] V. S. Murahari and T. Plötz, "On attention models for human activity recognition," in *Proc. ACM Int. Symp. Wearable Comput.*, Oct. 2018, pp. 100–103.

[21] K. Chen, L. Yao, X. Wang, D. Zhang, T. Gu, Z. Yu, and Z. Yang, "Interpretable parallel recurrent neural networks with convolutional attentions for multi-modality activity modeling," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.
[22] H. Ma, W. Li, X. Zhang, S. Gao, and S. Lu, "AttnSense: Multi-level attention mechanism for multimodal human activity recognition," in *Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI)*, Aug. 2019, pp. 3109–3115.
[23] K. Chen, L. Yao, D. Zhang, X. Wang, X. Chang, and F. Nie, "A semisupervised recurrent convolutional attention model for human activity recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1747–1756, May 2020.
[24] K. Chen, L. Yao, D. Zhang, B. Guo, and Z. Yu, "Multi-agent attentional activity recognition," in *Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI)*, Aug. 2019, pp. 1344–1350.
[25] X. Cheng, L. Zhang, Y. Tang, Y. Liu, H. Wu, and J. He, "Real-time human activity recognition using conditionally parametrized convolutions on mobile and wearable devices," *IEEE Sensors J.*, vol. 22, no. 6, pp. 5889–5901, Mar. 2022.
[26] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 1243–1252.
[27] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "A survey of online activity recognition using mobile phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, 2015.
[28] J. Qi, P. Yang, A. Waraich, Z. Deng, Y. Zhao, and Y. Yang, "Examining sensor-based physical activity recognition and monitoring for healthcare using Internet of Things: A systematic review," *J. Biomed. Inform.*, vol. 87, pp. 138–153, Nov. 2018.
[29] J. Lee and J. Kim, "Energy-efficient real-time human activity recognition on smart mobile devices," *Mobile Inf. Syst.*, vol. 2016, pp. 1–12, Jan. 2016.
[30] X. Yang, Y. Chen, H. Yu, Y. Zhang, W. Lu, and R. Sun, "Instance-wise dynamic sensor selection for human activity recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 1104–1111.
[31] D. Ravì, C. Wong, B. Lo, and G.-Z. Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 1, pp. 56–64, Jan. 2017.
[32] G. Bhat, R. Deb, V. V. Chaurasia, H. Shill, and U. Y. Ogras, "Online human activity recognition using low-power wearable devices," in *Proc. Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2018, pp. 1–8.
[33] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and Cooperation in Neural Nets*. Berlin, Germany: Springer, 1982, pp. 267–285.
[34] L. Sifre, "Rigid-motion scattering for image classification," Ph.D. dissertation, Ecole Polytechn., CMAP, Paris, France, 2014.
[35] B. Wu, A. Wan, X. Yue, P. Jin, S. Zhao, N. Golmant, A. Gholaminejad, J. Gonzalez, and K. Keutzer, "Shift: A zero FLOP, zero parameter alternative to spatial convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9127–9135.
[36] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5MB model size," 2016, *arXiv:1602.07360*.
[37] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8026–8037.
[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[39] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. 21st Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn. (ESANN)*, 2013, pp. 437–442.
[40] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explor. Newslett.*, vol. 12, no. 2, pp. 74–82, May 2011.
[41] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. D. R. Millán, and D. Roggen, "The opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognit. Lett.*, vol. 34, no. 15, pp. 2033–2042, 2013.
[42] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput.*, Jun. 2012, pp. 108–109.
[43] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Deep learning for time series classification: A review," *Data Mining Knowl. Discovery*, vol. 33, no. 4, pp. 917–963, Mar. 2019.
[44] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1578–1585.
[45] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[46] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.

JAEGYUN PARK received the B.S. degree from Eulji University, Seongnam, and the M.S. degree from Chung-Ang University, Seoul, South Korea, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering. His research interests include continual learning, sensor-based activity recognition, and feature selection.

WON-SEON LIM received the B.S. and M.S. degrees from Chung-Ang University, Seoul, South Korea, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering. His research interests include continual learning, neural architecture search, and on-device AI.

DAE-WON KIM (Member, IEEE) received the B.S. degree from Kyungpook National University, Daegu, South Korea, and the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology. He was a Postdoctoral Researcher at the Korea Advanced Institute of Science and Technology. He is currently a Professor with the School of Computer Science and Engineering, Chung-Ang University, Seoul, South Korea. His research interests include advanced data mining algorithms with innovative applications to bioinformatics, music emotion recognition, educational data mining, affective computing, and robot interaction.

JAESUNG LEE received the B.S., M.S., and Ph.D. degrees in computer science from Chung-Ang University, Seoul, Republic of Korea, in 2007, 2009, and 2013, respectively. In theoretical domain, he also studies classification, feature selection, and especially multilabel learning with information theory. He is currently an Associate Professor with the Department of Artificial Intelligence, Chung-Ang University. His research interests include machine learning, multilabel learning, model selection, and neural architecture search.