IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 23, 2022, accepted February 9, 2022, date of publication February 17, 2022, date of current version February 28, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3152530

A Comparative Study: Toward an Effective Convolutional Neural Network Architecture for Sensor-Based Human Activity Recognition

ZHAO ZHONGKAI[1], **SATOSHI KOBAYASHI**[1], **KAZUMA KONDO**[1], **TATSUHITO HASEGAWA**[1], **(Member, IEEE), AND MAKOTO KOSHINO**[2]

[1]Graduate School of Engineering, University of Fukui, Fukui 910-8507, Japan
[2]National Institute of Technology, Ishikawa College, Tsubata 929-0392, Japan

Corresponding author: Zhao Zhongkai (zzd21005@u-fukui.ac.jp)

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI Grant-in-Aid for Early-Career Scientists under Grant 19K20420.

ABSTRACT The feature extraction of human activity recognition (HAR) based on sensor data has been studied as a hand-crafted method. The significant feature extraction ability is a key factor in improving the accuracy of HAR. Recently, deep learning methods have been employed for feature extraction. In this paper, we review previous studies on deep learning methods in HAR and discuss suitable models for feature extraction. First, we applied various convolutional neural networks to clarify the effective architecture for HAR. Afterward, we developed advanced models by embedding submodules, such as self-attention and recurrent neural networks, often adopted in recent studies. Comparative experiments on HASC, UCI, and WISDM public datasets showed that Inception-V3, which used cross-channel multi-size convolution transformation, outperformed other backbones. Through comparative experiments after embedding submodules, submodules do not always have a positive effect on accuracy. Compared with other submodules, SENet has a positive effect. We conclude that it is essential to select an appropriate backbone model before applying the submodules, and submodules are unnecessary in some cases.

INDEX TERMS Human activity recognition, convolutional neural network, CNN architecture, submodules.

I. INTRODUCTION

Sensors of wearable devices and smartphones have enabled easy data collection. Sensor-based human activity recognition (HAR) requires high-level features of human activities from waveform data. Currently, deep learning (DL)-based methods have enabled automatic feature extraction with high accuracy in HAR. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) commonly use DL-based HAR methods. The architecture of these methods enables automatic extraction of multiple feature levels or time dependence. Wang *et al.* [1] proposed an attention-based HAR method to process weakly labeled activity data. The architecture consists of basic CNN layers and attention submodules by computing the compatibility between global and local features to generate weighted feature maps. Abdel-Basset *et al.* [2] constructed a CNN and RNN dual-path model and enhanced the learning ability of the model on spatial and temporal representations by adding attention modules to each path. Finally, the feature representations generated by the two paths are concatenated for activity classification. The implementation of DL-based models focuses on the recognition accuracy performance of different classifiers and pays less attention to the impact of model architecture on feature extraction.

HAR has been actively studied, but an effective feature learning approach is yet to be thoroughly investigated. Most studies have only compared conventional and simple CNN architecture. According to our survey, the most effective CNN architecture has not been verified. Because a suitable CNN architecture for sensor-based HAR is unknown and no architecture is universal, it is difficult to determine an appropriate architecture for a HAR service. In this respect, this study performs an effective experimental analysis of the existing HAR technology. The contributions of this study are as follows.

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

- Based on our extensive survey, we employed the well-known CNN architecture as the backbone model with submodules.
- Experiments to compare the activity recognition performance of many CNNs proposed in the field of image recognition showed that the number of parameters and layers have a lower impact on the HAR accuracy than model architecture. Choosing the appropriate backbone architecture and multidimensional scaling of architectures are crucial for sensor-based HAR.
- We embedded submodules into different architecture. For the impact of submodules on HAR, we found that submodules do not always work well, and selecting an appropriate backbone is significant.

II. RELATED STUDIES

A. CONVENTIONAL AND DL METHODS

Although current HAR research has made significant progress, it is still challenging. General HAR includes the following steps from original data to final activity classification: preprocessing, segmentation, feature extraction, and classification. Feature extraction is a key step in HAR because it can capture relevant information to distinguish various activities. The accuracy of a HAR method largely depends on features extracted from original signals. HAR methods are broadly categorized into two: hand-crafted feature extraction (conventional method) and automatic feature extraction (DL-based methods).

The conventional method requires strong manual intervention and high-level experience. In particular, a developer needs to combine specific background knowledge to extract features from raw data and make a classifier by machine learning. Therefore, while observing various acceleration waveforms, It is not easy for humans to recognize activity from sensor waveforms.

Fig. 1 shows that time series data are first preprocessed; then, the processed data are divided into multiple instances according to the window size, and feature extraction for each instance is performed. Basic statistics, such as representative values, are used as features. Extracted features are selected based on prior knowledge or a feature selection algorithm. Finally, the selected features are transformed using a trained classifier. In detecting periodic actions, such as walking and jogging, a fast Fourier transform is employed to obtain the peak frequency and its power spectrum. The representative model-based feature selection method is random forest [3].

Researchers are attracted to DL-based methods for their capacity to extract features automatically. DL-based methods do not require domain-specific knowledge [4]. Therefore, DL provides a standardized method to complete the feature extraction step. The extraction of time features by neural networks is conducive to building an end-to-end DL-based model, thereby facilitating the feature learning and recognition process. The feature extraction and model-building processes of DL-based methods are usually performed simultaneously, and the features can be learned automatically. Deep neural networks obtain deep representations from low-level data, suitable for complex HAR tasks. Various DL-based methods have been applied to time series feature extraction, including RNN, CNN, and hybrid architecture. The advantage of introducing DL into HAR is automatic feature extraction.

Fig. 2 shows the DL-based method in HAR. The significant difference between DL-based and conventional method is that the feature representation itself is learned from the given training data. Because manual intervention is not required, the automatic feature extraction program is significant for future developments. The disadvantage is that it requires large-scale training data and high computational cost. Because the computational complexity of DL-based methods is generally much greater than that of conventional classification methods. The overall model performance needs to be weighed when applied to smartphones or wearable devices.

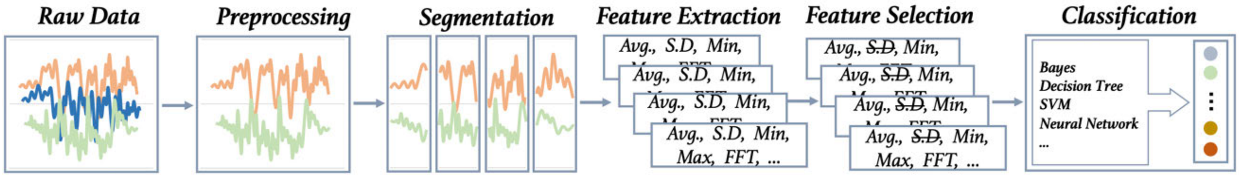


FIGURE 1. Conventional method in HAR using feature learning.

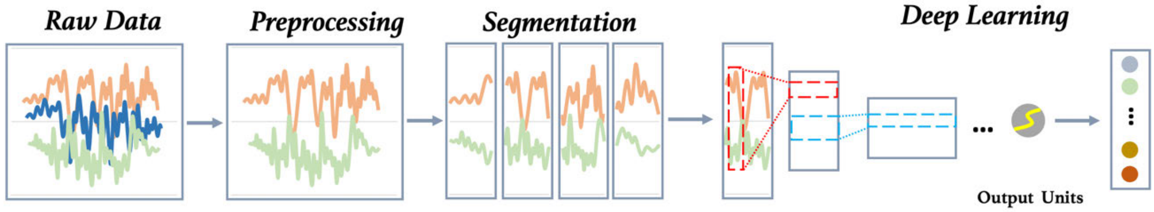


FIGURE 2. Deep learning method in HAR.

TABLE 1. Backbone model and datasets.

Ref.	backbone model	dataset
[5]	Conv+Pool+FC	Skoda[25], MHEALTH[26]
[6]	Conv+Pool+FC×2	Opportunity[27][28], Skoda, Actitracker[29]
[7]	(Conv+Pool)×3+FC	OPPORTUNITY, UniMiB SHAR[30]
[8]	(Conv+Pool)×3	Original
[9]	(Conv+Pool) ×3+FC×2	Original, OPPORTUNITY
[10]	(Conv+Pool) ×3+FC	Original
[11]	(Conv+Pool)×3+Conv+FC	OPPORTUNITY, Hand Gesture
[12]	(Conv+Pool) ×3+FC	OPPORTUNITY, UniMiB SHAR PAMAP2[31][32]

B. FEATURE LEARNING IN HAR

Table 1 summarizes the analysis of the CNN model architecture in sensor-based HAR. The summary is based on (1) the model architecture and (2) datasets.

A simple layer or several convolutional with pooling layers is typically used and connected to a fully connected layer to form a model. As shown in Table 1, reference [5] and [6] used a single-convolutional-and-pooling-layer architecture, and references [7]–[12] used a multiple-convolution-and-pooling-layer composite architecture. The architecture of these models is similar, but the model depths differ.

C. ADVANCED BACKBONE MODELS IN HAR

In advanced models in HAR, compared with the feature learning described above, more complex and deeper model architecture improves accuracy. These models use CNN to extract features automatically. In the field of object detection, a CNN feature extractor is usually called the "backbone" because the model architecture of the feature extractor and the overall model structure are considered separately. This study employs a CNN-based feature extractor as the "backbone" based on the formulation we describe in the next section and use self-attention and RNN as the submodules.

1) STUDY ON THE ADVANCED BACKBONE MODEL

Some researchers proposed advanced backbone models instead of simple ones (Table 2). Dong *et al.* [13] proposed an inception module combined with HCF. Long *et al.* [14] proposed methods to learn large- and small-scale networks separately and connected them. The core of the method is the introduction of two sizes of residual blocks. Turker *et al.* [15] proposed multiple ResNet architecture with different layers as feature extractors and cascaded extracted features to form the backbone. Ronald *et al.* [16] proposed the iSPLInception backbone based on Inception-ResNet, which uses a multichannel-residual composite architecture for HAR study. Mehmood *et al.* [17] used DenseNet as the backbone and dense links for HAR.

TABLE 2. Backbone model and datasets.

Ref.	backbone model	dataset
[13]	Inception+HC	UCI[33]
[14]	Dual Residual	OPPORTUNITY, UniMiB SHAR
[15]	ResNet	Original
[16]	Inception+ResNet	UCI, OPPORTUNITY, DAPHNET[34], PAMAP2
[17]	DenseNet	WISDM[35]

TABLE 3. Embedded submodule.

Ref.	backbone model + embedded submodule	dataset
[18]	Inception + GRU	UCI, OPPORTUNITY, PAMAP2
[19]	Residual + BiLSTM	UCI, OPPORTUNITY
[20]	Residual + LSTM	MHEALTH
[21]	CNN + BiLSTM	UCI, WISDM, PAMAP2
[22]	Dual Attention + CNN	WISDM, UNIMIB, PAMAP2, OPPORTUNITY, Original
[23]	DeepConvLSTM + Attention	OPPORTUNITY, PAMAP2, Skoda
[24]	SENet + CNN	WISDM ,UCI

2) STUDY BY EMBEDDING SUBMODULES

Some studies embedded submodules in different backbone models, e.g., CNN with long short-term memory (LSTM) and CNN with self-attention (Table 3). References [18], [19] used CNN with RNN architecture. Particularly, Xu *et al.* [18] used a kernel-based convolutional layer to extract multidimensional features by the inception module. Then, it incorporated GRU to realize the modeling of time series features and exploit data features to complete the classification task. Zhao [19] used the deep network architecture of residual bidirectional LSTM (BiLSTM) as well as the residual connection between stacked cells as a shortcut to avoid the vanishing gradient problem. Zohair *et al.* [20] used an Inception-like architecture to divide multiple CNN branches; after combining extracted features, the outputs of all branches were concatenated and input into the BiLSTM layer. Challa *et al.* [21] combined convolutional and BiLSTM layers; after combining the two LSTM layers, it was connected to a convolutional layer. In addition, a global average pooling layer was used to replace the fully connected layer after convolution to reduce the model parameters.

References [22]–[24] used convolution layers with self-attention architecture. Gao *et al.* [22] used a dual attention network; the overall architecture of the attention mechanism was similar to that of the convolution block attention module, but the attention mechanism emphasized time information. The model finally performs feature fusion by designing dual channels. Murahari *et al.* [23] proposed a DeepConvLSTM architecture with an attention layer. Attention models learn weights on the input data and leverage them to weigh the temporal contexts considered. The model architecture was first composed of DeepConvLSTM. After dropout and LSTM, the attention layer was embedded behind LSTM to form the main architecture of the model. Khan *et al.* [24] designed a three-channel CNN model block and embedded an attention module into each model block. The above study used many public benchmark datasets [25]–[35] to evaluate generalization performance.

III. STUDY PROCEDURE

A. FORMULATION

Figure 3 shows the framework of our model. We formulate the HAR model as a backbone with submodules based on the abovementioned survey. The input and output of the model remained unchanged. In the middle part, we categorized the model backbone into two. The first (A) is the backbone architecture without submodules. The backbones consisted of CNNs with different architecture. Other categories (B) and (C) are composed of a backbone with each submodule. Among them, the submodules are used in different manners. We classified them into RNN and self-attention submodules. The backbone architecture in categories (B) and (C) was basically implemented as same as the original backbone (A).

B. ARCHITECTURE FOR BACKBONE

DL-based methods, such as CNNs and RNNs, can achieve state-of-the-art (SOTA) results by automatically learning the features of raw sensor data. With the emergence of hybrid networks, the idea of purely deepening neural networks has gradually changed. By constructing various architecture, the hybrid networks enhanced feature representation to improve computing and network performances. This also provides the possibility for developing DL-based methods of the mobile terminal.

We comprehensively investigated different CNN backbone models and employed Baseline [36], VGG16 [37], ResNet18 [38], PyramidNet18 [39], MobileNet [40], MobileNet-V2 [41], MobileNet-V3small [42], MobileNASNet [43], MnasNet [44], DenseNet121 [45], Inception-V3 [46], Xception [47], EfficientNet B0 and EfficientNet lite0 [48] to conduct an experimental comparative study. These models were proposed to solve the problem of image recognition; thus, we rebuild the model architecture for HAR (Fig. 4 and Fig. 5).

We use a CNN with three stacked convolution and pooling layers as the baseline CNN model because this architecture is the basic architecture used in Reference [36] and other related studies.

The main contribution of VGG16 is that increasing the network depth can improve the final network performance to a certain extent, and for the first time, VGG16 replaces the larger convolution kernel using several consecutive 3 × 3 convolution kernels.

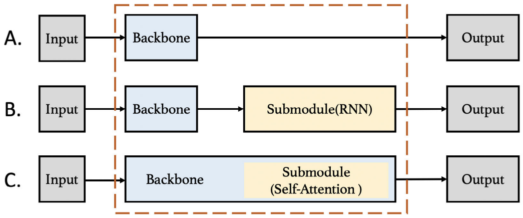


FIGURE 3. Model framework.

The core of ResNet is to alleviate the problem of gradient disappearance. As the conventional CNN architecture deepens, its performance may deteriorate, thereby limiting the number of network layers. ResNet uses the shortcut connections to solve the network deterioration problem.

PyramidNet18 gradually increases the network dimension by improving ResNet using the pyramid architecture. It also uses zero-padding and direct-connected identity mapping to increase the network width and improve recognition accuracy.

The MobileNet series network is a lightweight neural network focused on mobile devices. Using deep separable convolution, the network calculation speed is improved, and new hyperparameters are introduced to adjust the number of output channels to balance the calculation speed and accuracy of the network.

The main contribution of DenseNet is to alleviate the problem of gradient disappearance, strengthen the transfer of features, and use each output feature effectively. By designing dense blocks, the model obtained a narrower network architecture.

The EfficientNet series of networks use neural architecture search (NAS) technology to propose the baseline model and obtain a model architecture with comprehensive performance utilizing an approach that conforms to model expansion.

Inception-V3 uses a multibranch architecture to fuse features from different receptive fields, recognizes data features of different scales, and uses stacked small kernel size (3 × 3) convolution instead of large kernel size (5 × 5) convolution. Fig. 6 shows that the inception module convolves and connects multiple types of receptive fields in parallel as well as deepens the network by stacking these layers in multiple stages.

Similar to MobileNet, Xception adopts deep separable convolution. The convolution sequence is first 3 × 3 depthwise convolution, then 1 × 1 convolution, and the activation function is not connected after the convolution layer.

Mobile NASNet and MnasNet use a technology similar to NAS to search for small and effective network architecture automatically. Therefore, multiple elements are used as optimization indicators to design neural networks efficiently. Compared with the previous algorithms that only consider whether the final result is SOTA, MnasNet improves accuracy, reduces latency on real mobile devices, and so on. NASNet first searches for modules of the neural network architecture on smaller datasets then migrates the network

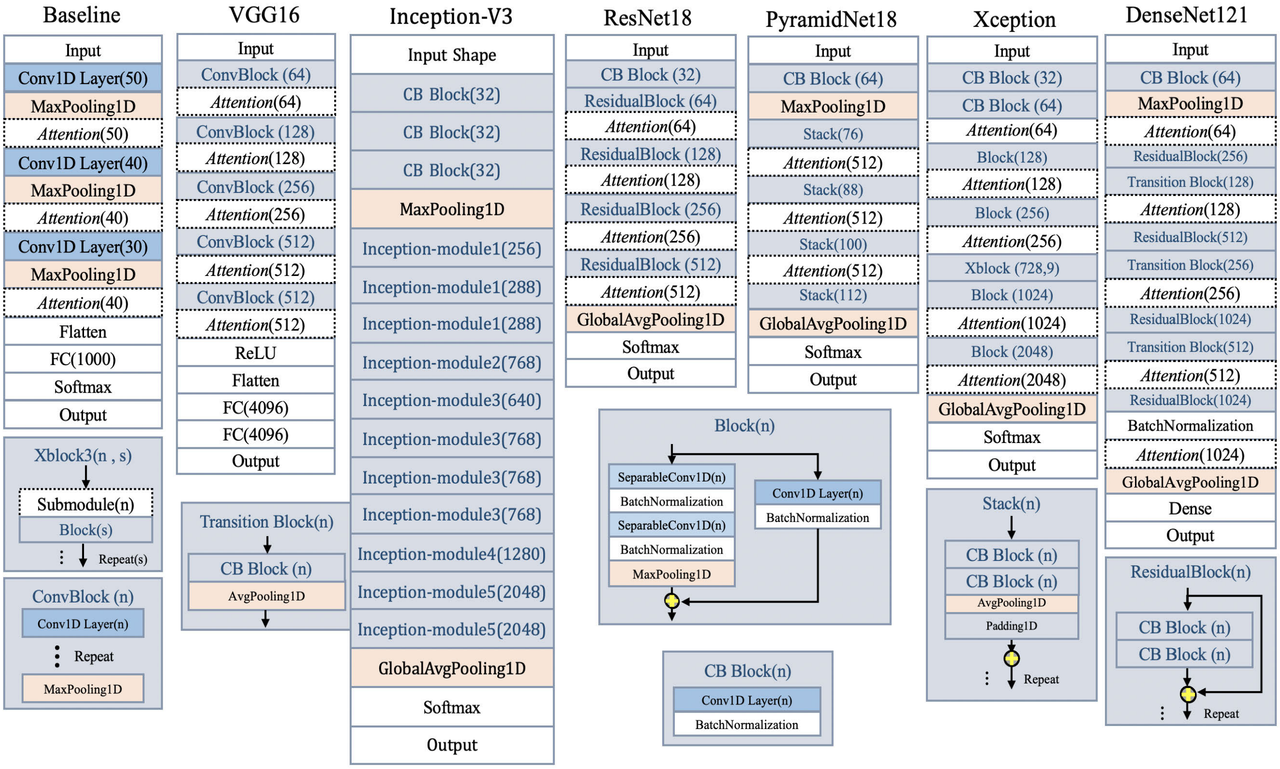


FIGURE 4. Model architecture used in our experiments. (The attention square in the figure is the embedded position of the attention submodule).

architecture to a larger dataset and further proposes the "Scheduled Drop Path" regularization technology, which improves the generation effect of NASNet. The NASNet architecture is more concise and has lower computational complexity than the previous neural network architecture. The overall architecture uses a multichannel convolution architecture and residual architecture.

Through extensive study and investigation, we found that although there are many types of existing CNN model architecture, most models have similar core architecture. We chose different CNN models for implementation based on the core architecture of the model. Large and complex models are challenging to employ In the HAR field. The application field of the sensor requires low latency and fast response speed. Studying small and efficient CNN model architecture in these fields is necessary. To realize a different model architecture, we selected a lightweight model for this study.

C. SUBMODULES

Self-attention model is divided into spatial, channel, and hybrid attention modules. Self-attention model usually transforms information features in data to improve the feature extraction ability. The channel module expresses the correlation between the channel and characteristic information by adding weight to signals on the output characteristic channel. The larger the weight, the higher the correlation. The channel attention mechanism analyzes the relevance of each feature channel by focusing on the correlation between the different channels and feature learning. Finally, different weight coefficients are assigned to each feature result to strengthen the expression of essential features and suppress irrelevant features. The spatial domain mainly enhances the feature expression of key regions to improve its performance and interpretability in classification tasks. Essentially, the spatial information of data features is passed through the spatial conversion module, and the information features are paid attention to using the feature spatial relationship. Generally, the key information on an effective feature descriptor is retained; a weight mask is generated for each position, and the output is weighted to calculate the spatial attention.

Studies have shown that the core of a CNN architecture is the convolutional layer operation. CNN analyzes the information feature space components formed by the feature space and channel information of each layer by calculating the weight of the entire feature relationship, thereby improving the analytical power of the data features.

SENet [49] focused on the channel relationship and proposed a lightweight architecture unit, the "squeeze-and-excitation network" module, to improve data characteristics and extraction ability by establishing the interdependence between channels. The architecture is illustrated in Fig. 7.

SKNet [50] was inspired by the fact that cortical neurons could dynamically adjust their own receptive fields according to different stimuli, and was improved by combining the

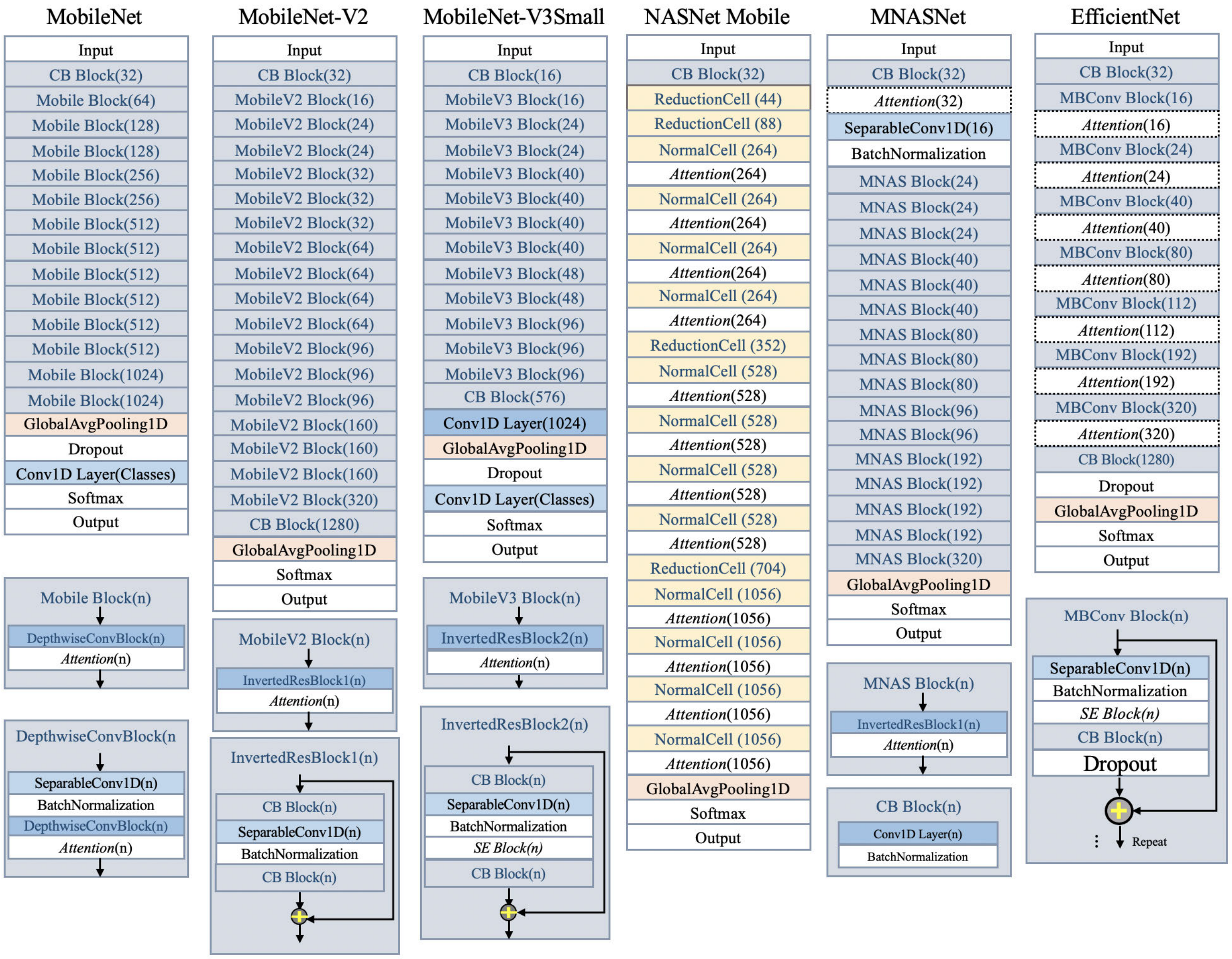


FIGURE 5. Model architecture used in our experiments. (The attention square in the figure is the embedded position of the attention submodule).

idea of SENet. SKNet is relatively simple in design philosophy, i.e., transforming all convolution kernels larger than 1 with selective kernel and using small theoretical parameters brought by group convolution. Therefore, even if the design of multichannel and dynamic selection is added, it will not incur considerable overhead. Fig. 8 shows the architecture.

CBAM [52] is a hybrid attention mechanism model; as a lightweight general-purpose module like SENet and SKNet, it can be integrated into any CNN architecture. The CBAM architecture is divided into channel and spatial attention modules. Regarding the order of using attention mechanism modules in the model, a detailed description of the results was performed through experimental comparisons; using the channel attention mechanism first and then using the spatial attention mechanism was more effective. Fig. 9 shows CBAM architecture. The channel part of CBAM is similar to the attention module of SENet. The main difference is that CBAM adopts global average pooling and global max pooling in the initial stage. The two different global pooling methods can be used to extract rich features of HAR.

IV. EXPERIMENTS

A. EXPERIMENTAL SETTING

In this study, we handle sensor-based HAR as a supervised feature learning task using entirely labeled datasets divided into training set (D_{train}), validation set (D_{valid}), and test set (D_{test}). Labeled D_{train} and D_{valid} were used for parameter tuning during model training, e.g., with or without data augmentation (DA) and learning rate. D_{test} was used to evaluate the performance of each model. In that case, a model was trained on the combined set of D_{train} and D_{valid}. There were two experiments, one to verify backbones performance for each dataset (Experiment A) and the other to verify the impact of submodules (Experiment B).

B. DATASETS

We performed experiments to evaluate the performance of CNN models in sensor-based HAR using three public benchmark datasets: HASC dataset [52], UCI Smartphone dataset, WISDM dataset (hereinafter, HASC, UCI, and WISDM, respectively). The datasets are summarized in Table 4.

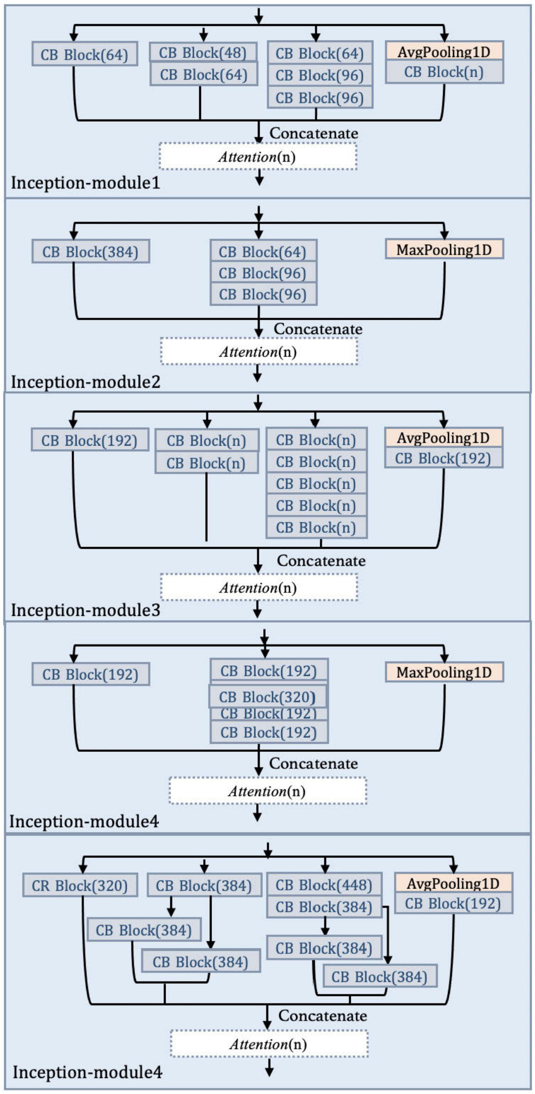


FIGURE 6. Inception-V3 module architecture (The attention square in the figure is the embedded position of the attention submodule).

HASC is a benchmark dataset for basic HAR from acceleration data collected using a smartphone. The estimation targeted six types of basic activities: standing (stay), walking (walk), jogging (jog), skipping (skip), going upstairs (stUp), and going downstairs (stDown). In this study, we used acceleration data of 170 subjects at a sampling frequency of 100 Hz collected using iOS devices. As a preprocessing step, we removed the data 5 s before and after each measurement and divided data into a window width of 256 and a stride width of 256.

UCI is a benchmark dataset for daily living activities using smartphone motion sensors. This dataset collected acceleration and gyroscope data from 30 subjects labeled with six types of activity: walking, going upstairs (walking_upstairs), going downstairs (walking_downstairs), sitting, standing, and laying. The dataset comprises data measured at a sampling frequency of 50 Hz and was divided into 128 samples for each. There are two types of acceleration data in this dataset: acceleration from the accelerometer (total acceleration) and estimated body acceleration. In this study, we used the total acceleration. We also used the gyroscope data. Thus, we used six-axis sensor data as inputs.

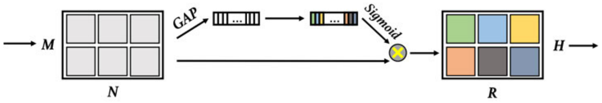


FIGURE 7. SENet network model architecture.

WISDM is a benchmark dataset for HAR using a smartphone-based accelerometer. This dataset collected acceleration data of 36 subjects labeled with six types of basic activity: walking, jogging, going upstairs (upstairs), going downstairs (downstairs), sitting, and standing. The dataset comprises data measured at a sampling frequency of 20 Hz. As a preprocessing, we removed the data 3 s before and after each measurement and divided the data into a window width of 256 and a stride width of 256.

C. TRAINING SETTINGS

We experimented with the training and validation sets for each of three datasets to determine hyperparameters during training, such as the number of epochs and learning rate. As a result of the experiment, for all datasets, we set the batch size to 1024, optimizer to Adam [53], and the learning rate to 0.001. In addition, the number of epochs was set to 1000 and 500 for Experiments A and B, respectively. The number of epochs in Experiment B was set lower than in Experiment A because the number of model architecture to be verified increased due to the combination of model architecture and submodules; also, the time required for the experiment increased. It was confirmed by a pre-experiment that 500 epochs of training generally converged.

In the hyperparameter determination experiment, we also determined whether to apply DA during the training of each dataset. We used two DAs: flipping and channel shuffling. Flipping is a DA that randomly reverses each axis's positive and negative values of sensor data. Channel shuffling is a DA that swaps each axis of the sensor data. These two DAs can simulate various storage orientations of devices that measure sensor data. The experiment results showed that the estimation accuracies of HASC and WISDM were higher with DA, whereas the estimation accuracy of UCI was lower with DA. Therefore, we decided to apply DA only to HASC and WISDM.

The experiments were performed with Intel Core i9-9900X, 64GB RAM, and NVIDIA TITAN RTX. We used TensorFlow to implement and train models.

D. EVALUATION INDICES

We used a subject-based hold-out method to evaluate these models. As the experiment comprised multiple trials, we evaluated these models via the average accuracy of multiple trials. The number of trials differed between Experiments A and B due to the time required for the experiments.

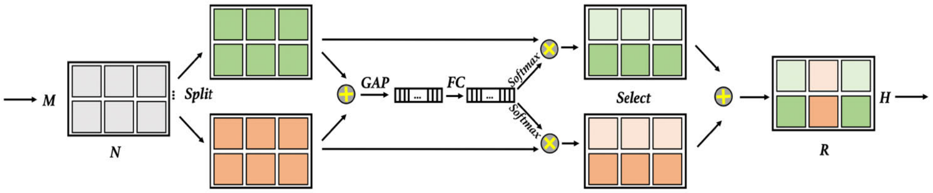


FIGURE 8. SKNet network model architecture.

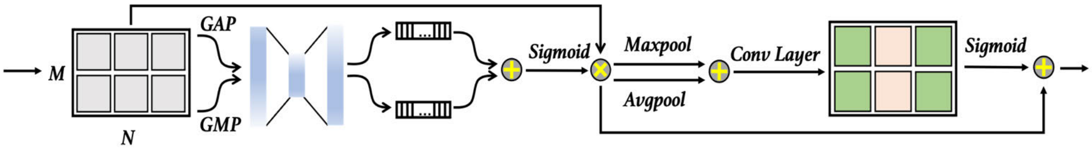


FIGURE 9. CBAM network model architecture.

In Experiment A, there were five trials. In the five trials, the subjects assigned to D_{train}, D_{valid} and D_{test} differed. Considering the time consumption and backbone experimental results, we selected HASC as the experimental data in Experiment B and the number of trials was two. The dataset used in Experiment B was part of the dataset of Experiment A, which was divided into $D_{train} : D_{valid} : D_{test} = 30 : 20 : 20$ from 70 subjects. As in Experiment A, the subjects assigned to D_{train}, D_{valid}, and D_{test} differed for each trial, and the same split dataset was used across the models. For HASC and WISDM, we prepared five sets of subjects randomly split into D_{train}, D_{valid}, and D_{test} for each trial. In UCI, we randomly split the training data prepared by the dataset into D_{train} and D_{valid}, and D_{test} was a set of nine subjects of the test data in the dataset. The numbers of subjects in D_{train}, D_{valid}, and D_{test} for each dataset is listed in Table 4.

V. EXPERIMENTAL RESULTS

A. DEFFECTIVE BACKBONE ARCHITECTURE

In Table 5, Inception-V3 and Xception have similar backbone architecture. The two backbones completely separate channel coupling and spatial correlations using multibranch architecture. Although their parameters and layers were different, they exhibited good accuracy and performance on the three datasets. Therefore, the use of cross-channel multi-size convolutional transform architecture has a positive impact on sensor-based HAR. Xception and MobileNet series use depthwise separable convolution, but the overall accuracy of MobileNet is not as good as that of Xception. Based on the Inception-V3 architecture, Xception replaces conventional convolution with depthwise separable convolution, improving the model's effectiveness without increasing the backbone complexity. In addition, it uses a multichannel multi-size convolution transformation architecture.

For the MobileNet series, although the network deepened, the overall HAR accuracy was poor. Compared with Xception, although depth separable convolution is used, Increasing the network width while ignoring depth hardly improves accuracy. Moreover, the MobileNet series uses model compression to reduce computational costs; however, it reduces accuracy. While deepening the network depth, MobileNet-V3Small reduces the number of parameters, but the overall performance is better than that of the other two mobile series backbones; therefore, This result indicates that many parameters are not always suitable for HAR. Not too large and a suitable parameter setting is essential.

DenseNet121, ResNet18, and PyramidNet18 use residual architecture. However, DenseNet121 has a higher accuracy rate than the other backbone architecture because DenseNet uses the residual structure while strengthening the features in each convolution stage.

TABLE 4. Experimental data details.

Dataset	Number of subjects in D_{train}, D_{valid}, D_{test}	Sampling frequency	Input shape	Output Shape
HASC	100 : 50 : 20	100 Hz	(?, 3, 256)	(?, 6)
UCI	16 : 5 : 9	50 Hz	(?, 6, 128)	(?, 6)
WISDM	25 : 5 : 6	20 Hz	(?, 3, 256)	(?, 6)

TABLE 5. Comparison of the accuracy with the number of params and layers among various backbone models for each dataset.

Backbone Model	Params	Layers	HASC	UCI	WISDM
Baseline	1.3M	10	87.71%	89.98%	89.55%
VGG16	**38.5M**	23	89.54%	90.40%	89.32%
Inception-V3	14.3M	313	91.85%	**94.23%**	**91.54%**
ResNet18	3.9M	67	90.53%	91.44%	87.53%
PyramidNet18	0.4M	74	91.48%	92.56%	85.23%
Xception	20.7M	126	92.31%	92.32%	90.17%
DenseNet121	5.6M	429	**92.55%**	92.52%	88.75%
MobileNet	6.0M	92	91.22%	80.32%	88.82%
MobileNet-V2	6.7M	156	90.62%	81.62%	74.71%
MobileNet-V3Small	2.9M	248	91.45%	91.25%	82.47%
NASNet Mobile	4.1M	**800**	86.46%	54.96%	84.98%
MnasNet	9.4M	147	89.75%	87.43%	84.57%
EfficientNet B0	11.4M	233	92.50%	93.53%	89.11%
EfficientNet lite0	10.8M	153	91.52%	91.20%	85.81%

TABLE 6. The number of params for each combination of backbone model and submodule.

Backbone Model	Params						
	Original	SE-	SK-	CBAM-	GRU-	LSTM-	BI-LSTM-
Baseline	1.3M	1.3M	1.4M	1.3M	0.2M	0.2M	0.6M
VGG16	38.5M	38.8M	39.0M	38.7M	5.3M	5.4M	6.0M
Inception-V3	14.3M	20.8M	17.3M	17.7M	15.2M	15.5M	16.9M
ResNet18	3.9M	4.0M	4.1M	4.0M	4.2M	4.3M	4.9M
PyramidNet18	0.4M	0.4M	0.5M	0.4M	0.6M	0.7M	1.0M
Xception	20.7M	25.8M	23.7M	23.4M	21.6M	21.9M	23.3M
DenseNet121	5.6M	6.3M	6.2M	6.0M	6.1M	6.3M	7.2M
MobileNet	6.0M	7.9M	7.8M	7.0M	6.5M	6.7M	7.6M
MobileNet-V2	6.7M	6.9M	7.2M	6.8M	7.4M	7.6M	8.6M
MobileNet-V3Small	2.9M	2.9M	3.1M	2.9M	3.4M	3.6M	4.5M
NASNet Mobile	4.1M	7.0M	6.2M	5.6M	4.6M	4.8M	5.7M
MnasNet	9.4M	9.5M	9.9M	9.5M	9.6M	9.7M	10.2M
EfficientNet B0	11.4M	13.4M	13.6M	13.4M	12.1M	12.3M	13.3M
EfficientNet lite0	10.8M	10.9M	11.0M	10.9M	11.4M	11.6M	12.6M

Comparing the backbone architecture of EfficientNet B0 and MobileNet-V3Smal. They have similar model architecture, but the EfficientNet B0 structure effectively improves the model accuracy by scaling in multiple dimensions. In the research of sensor-based HAR, scaling the width and depth of a network model to a certain ratio can effectively improve HAR accuracy. MnasNet and NASNet Mobile exhibited poor results in the three datasets. The model architectures are all

TABLE 7. Comparison of the activity recognition accuracy between various backbone CNN architecture with a submodule for HASC dataset.

Backbone Model	Original	SE-	SK-	CBAM-	GRU-	LSTM-	BI-LSTM-
Baseline	85.73%	85.82%	86.35%	86.40%	**89.80%**	89.75%	89.64%
VGG16	85.59%	87.35%	**90.33%**	86.55%	17.10%	17.10%	17.10%
Inception-V3	**90.37%**	90.33%	84.59%	54.01%	90.18%	89.32%	89.01%
ResNet18	87.69%	**89.17%**	86.57%	17.10%	87.43%	86.19%	88.03%
PyramidNet18	90.06%	**90.93%**	89.61%	90.13%	90.53%	90.10%	90.89%
Xception	90.89%	91.31%	90.04%	91.01%	**91.37%**	90.96%	90.38%
DenseNet121	90.17%	90.40%	83.92%	**91.12%**	90.60%	89.97%	89.50%
MobileNet	89.34%	85.06%	89.14%	**89.79%**	89.16%	77.89%	87.88%
MobileNet-V2	**88.92%**	85.50%	55.38%	84.55%	68.11%	81.70%	86.97%
MobileNet-V3Small	**89.04%**	86.18%	83.06%	83.55%	79.76%	82.37%	86.65%
NASNet Mobile	57.26%	**86.83%**	70.22%	80.42%	57.40%	56.70%	59.58%
MnasNet	57.82%	65.46%	71.78%	**89.51%**	76.18%	84.50%	59.36%
EfficientNet B0	**90.82%**	89.00%	88.95%	90.24%	89.38%	90.39%	89.67%
EfficientNet lite0	89.82%	**90.84%**	89.54%	90.09%	87.63%	84.91%	88.83%

based on searching for the best architecture of an image dataset. Therefore, this architecture is not well adapted to sensor-based HAR research. VGG16 had the largest number of parameters, and NASNet Mobile had the largest number of layers. However, compared with the other backbones, both models have poor results. Because VGG16 used two fully connected layers, the nonlinear expression ability of the backbone was enhanced. However, the lack of backbone depth affects the feature expression. NASNet Mobile had the largest number of layers. But ignoring the model width while increasing the model depth affects its results.

Thus, based on the overall results, the number of parameters and layers have a lower impact on the HAR accuracy than model architecture. Choosing the appropriate backbone architecture and multidimensional architecture scaling are crucial for sensor-based HAR.

B. IMPACT OF EMBEDDING SUBMODULES ON ACCURACY

Table 6 summarizes parameters of backbones and embedded submodules (retaining one decimal place). The data show that the number of parameters of Baseline (Simple CNN) and VGG16 after embedding RNN submodules decreased significantly because the fully connected layer was replaced by RNN. For other backbone models, the RNN submodule replaced the global pooling layer; therefore, the number of parameters was increased. The self-attention submodule was directly embedded in backbones. Therefore, the model parameters increased after embedding the self-attention submodule in the backbone.

Li *et al.* proved that RNN submodules are effective for CNNs with a simple backbone architecture. In comparison to the results in Table 7, embedding the self-attention submodule in the simple CNN is also effective, but it is advisable to use the RNN submodule for the simple CNN. When using a more complex model architecture as a backbone, the overall self-attention submodule performs slightly better than the RNN submodule, and compared with the RNN submodule, combining the self-attention submodule and backbone architecture improves the accuracy. Comparing MobileNet-V3Small and MnasNet, shows that the two models use the same module architecture. The difference is that SENet is used in the MobileNet submodule, which also proves that the self-attention submodule has a positive impact on the results.

The SENet submodule performed better with multiple different backbone architectures. However, the same submodule embedded in different backbone architecture does not all work well, and the submodules sometimes negatively affect the convergence of the selected backbone architecture model. For example, the accuracy of SKNet embedding VGG16 increased by 4.74%, but in Inception-V3, it decreased by 5.78%.

Comparing the MobileNet and EfficientNet series model architectures, almost all submodules in the current experiment will negatively impact after being embedded. In addition, compared to MobileNet's architecture, the MobileNet-V3Small architecture incorporates the idea of SENet, which has a negative impact. MnasNet is based on MobileNet-V2 and incorporates the SENet concept.

However, compared to the experimental results, the original accuracy of MnasNet decreased significantly. Therefore, for a lightweight model architecture, it is worth considering whether to embed submodules in sensor-based HAR research. The backbone architecture of the NasNet series can be effectively combined with submodules, but for NASNet Mobile, the self-attention submodule is more effective than the RNN submodule.

Combining Tables 6 and 7, the submodule selection has no direct relationship with the model parameters and layers. In addition, for embedded submodules, no one submodule can be suitable for all backbone architecture. It is crucial to select an appropriate backbone architecture.

VI. CONCLUSION

In the study of sensor-based HAR, most conventional methods have a single model architecture and lack detailed analysis and comparison of different model architecture and various submodules. To address this problem, we first adopted the CNN model of different architecture, initially used in image recognition as the backbone architecture. Second, the impact of the submodules on the backbone architecture was evaluated by embedding different submodules in backbone architecture. We applied different backbone models to three benchmark datasets HASC, UCI, and WISDM to compare and evaluate the effectiveness of the backbone models. The results showed that the coupling between the channel and spatial correlations was completely separated is effective. The accuracy result could improve by simultaneously scaling the depth and width to a specific ratio. The use of submodules is conducive to improving accuracy, but it is necessary to select appropriate submodules according to different backbone architectures. It is worth considering whether the submodules are used in sensor-based HAR research for lightweight models. In future research, we will combine the experimental results to propose a new HAR model that is suitable for sensors.

ACKNOWLEDGMENT

An earlier version of this paper was presented at the 2nd International Conference on Computational Intelligence and Intelligent Systems (CIIS 2019), Bangkok, Thailand, November 2019 [DOI: 10.1145/3372422.3372439].

REFERENCES

[1] K. Wang, J. He, and L. Zhang, "Attention-based convolutional neural network for weakly labeled human activities' recognition with wearable sensors," *IEEE Sensors J.*, vol. 19, no. 7, pp. 7598–7604, Sep. 2019.

[2] M. Abdel-Basset, H. Hawash, R. K. Chakrabortty, M. Ryan, M. Elhoseny, and H. Song, "ST-DeepHAR: Deep learning model for human activity recognition in IoHT applications," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4969–4979, Mar. 2021.

[3] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

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

[5] S. Ha, J.-M. Yun, and S. Choi, "Multi-modal convolutional neural networks for activity recognition," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 3017–3022.

[6] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *Proc. 6th Int. Conf. Mobile Comput., Appl. Services*, 2014, pp. 197–205.

[7] F. Li, K. Shirahama, M. Nisar, L. Köping, and M. Grzegorzek, "Comparison of feature learning methods for human activity recognition using wearable sensors," *Sensors*, vol. 18, no. 3, p. 679, Feb. 2018.

[8] Y. Chen and Y. Xue, "A deep learning approach to human activity recognition based on single accelerometer," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 1488–1492.

[9] H. Gjoreski, J. Bizjak, and M. Gjoreski, "Comparing deep and classical machine learning methods for human activity recognition using wrist accelerometer," in *Proc. IJCAI Workshop Deep Learn. Artif. Intell.*, vol. 10. New York, NY, USA: Intell, 2016, p. 97.

[10] J. Hannink, T. Kautz, C. F. Pasluosta, K.-G. Gaßmann, J. Klucken, and B. M. Eskofier, "Sensor-based gait parameter extraction with deep convolutional neural networks," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 1, pp. 85–93, Jan. 2017.

[11] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 1–7.

[12] Z. Yang, O. I. Raymond, C. Zhang, Y. Wan, and J. Long, "DFTerNet: Towards 2-bit dynamic fusion networks for accurate human activity recognition," *IEEE Access*, vol. 6, pp. 56750–56764, 2018.

[13] M. Dong, J. Han, Y. He, and X. Jing, "HAR-Net: Fusing deep representation and hand-crafted features for human activity recognition," in *Proc. Int. Conf. Signal Inf. Process., Netw. Comput.* Singapore: Springer, 2019, pp. 32–40.

[14] J. Long, W. Sun, Z. Yang, and O. I. Raymond, "Asymmetric residual neural network for accurate human activity recognition," *Information*, vol. 10, no. 6, p. 203, Jun. 2019.

[15] T. Tuncer, F. Ertam, S. Dogan, E. Aydemir, and P. Pławiak, "Ensemble residual network-based gender and activity recognition method with signals," *J. Supercomput.*, vol. 76, no. 3, pp. 2119–2138, Mar. 2020.

[16] M. Ronald, A. Poulose, and D. S. Han, "ISPLInception: An inception-ResNet deep learning architecture for human activity recognition," *IEEE Access*, vol. 9, pp. 68985–69001, 2021.

[17] K. Mehmood, H. A. Imran, and U. Latif, "HARDenseNet: A 1D DenseNet inspired convolutional neural network for human activity recognition with inertial sensors," in *Proc. IEEE 23rd Int. Multitopic Conf. (INMIC)*, Nov. 2020, pp. 1–6.

[18] 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.

[19] Y. Zhao, R. Yang, G. Chevalier, X. Xu, and Z. Zhang, "Deep residual bidir-LSTM for human activity recognition using wearable sensors," *Math. Problems Eng.*, vol. 2018, pp. 1–13, Dec. 2018.

[20] M. Zohair, E. Atlam, G. Dagnew, A. R. Alzighaibi, E. Ghada, and I. Gad, "Bidirectional residual LSTM-based human activity recognition," *Comput. Inf. Sci.*, vol. 13, no. 3, pp. 1–40, 2020.

[21] S. K. Challa, A. Kumar, and V. B. Semwal, "A multibranch CNN-BiLSTM model for human activity recognition using wearable sensor data," *Vis. Comput.*, pp. 1–15, Aug. 2021.

[22] W. Gao, L. Zhang, Q. Teng, J. He, and H. Wu, "DanHAR: Dual attention network for multimodal human activity recognition using wearable sensors," *Appl. Soft Comput.*, vol. 111, Nov. 2021, Art. no. 107728.

[23] 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.

[24] Z. N. Khan and J. Ahmad, "Attention induced multi-head convolutional neural network for human activity recognition," *Appl. Soft Comput.*, vol. 110, Oct. 2021, Art. no. 107671.

[25] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Tröster, "Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection," in *Proc. Eur. Conf. Wireless Sensor Netw.* Berlin, Germany: Springer, 2008, pp. 17–33.

[26] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga, "mHealthDroid: A novel framework for agile development of mobile health applications," in *Proc. Int. Workshop Ambient Assist. Living* Cham, Switzerland: Springer, 2014, pp. 91–98.

[27] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Forster, G. Troster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. D. R. Millan, "Collecting complex activity datasets in highly rich networked sensor environments," in *Proc. 7th Int. Conf. Netw. Sens. Syst. (INSS)*, Jun. 2010, pp. 233–240.

[28] 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, pp. 2033–2042, Nov. 2013.

[29] J. W. Lockhart, G. M. Weiss, J. C. Xue, S. T. Gallagher, A. B. Grosner, and T. T. Pulickal, "Design considerations for the WISDM smart phone-based sensor mining architecture," in *Proc. 15th Int. Workshop Knowl. Discovery Sensor Data*, 2011, pp. 25–33.

[30] D. Micucci, M. Mobilio, and P. Napoletano, "UniMiB SHAR: A dataset for human activity recognition using acceleration data from smartphones," *Appl. Sci.*, vol. 7, no. 10, p. 1101, 2017.

[31] 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.

[32] A. Reiss and D. Stricker, "Creating and benchmarking a new dataset for physical activity monitoring," in *Proc. 5th Int. Conf. Pervasive Technol. Rel. Assistive Environ.*, 2012, pp. 1–8.

[33] D. Anguita, A. Ghio, L. Oneto, X. P. Perez, and J. L. R. Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. 21st Int. Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn. (ICACNI)*, 2013, pp. 437–442.

[34] M. Bächlin, M. Plotnik, D. Roggen, N. Giladi, J. M. Hausdorff, and G. Tröster, "A wearable system to assist walking of Parkinson's disease patients," *Methods Inf. Med.*, vol. 49, no. 1, pp. 88–95, 2010.

[35] 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, 2011.

[36] F. Li, K. Shirahama, M. Nisar, L. Köping, and M. Grzegorzek, "Comparison of feature learning methods for human activity recognition using wearable sensors," *Sensors*, vol. 18, no. 3, p. 679, Feb. 2018.

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

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[39] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5927–5935.

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

[41] 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.

[42] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[43] X. Qin and Z. Wang, "NASNet: A neuron attention stage-by-stage net for single image deraining," 2019, *arXiv:1912.03151*.

[44] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "MnasNet: Platform-aware neural architecture search for mobile," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2820–2828.

[45] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.

[46] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[47] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[48] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2019, pp. 6105–6114.

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

[50] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 510–519.

[51] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

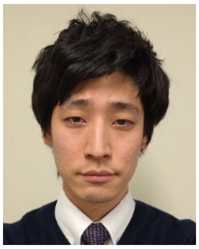
[52] H. Matsuyama, K. Hiroi, K. Kaji, T. Yonezawa, and N. Kawaguchi, "Ballroom dance step type recognition by random forest using video and wearable sensor," in *Proc. Adjunct Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput., ACM Int. Symp. Wearable Comput.*, Sep. 2019, pp. 774–780.

[53] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

ZHAO ZHONGKAI received the M.S. degree in engineering from Lanzhou Jiaotong University, Lanzhou, China, in 2020. He is currently pursuing the Ph.D. degree with the Graduate School of Engineering, University of Fukui. His research interests include human activity recognition and transfer learning.

SATOSHI KOBAYASHI received the B.S. degree in engineering from the University of Fukui, Fukui, Japan, in 2020, where he is currently pursuing the M.S. degree with the Graduate School of Engineering. His research interests include human activity recognition and deep learning application.

KAZUMA KONDO received the B.S. degree in engineering from the University of Fukui, Fukui, Japan, in 2020, where he is currently pursuing the M.S. degree with the Graduate School of Engineering. His research interests include human activity recognition and deep learning application.



TATSUHITO HASEGAWA (Member, IEEE) received the Ph.D. degree in engineering from Kanazawa University, Kanazawa, in 2015. From 2011 to 2013, he was a System Engineer with Fujitsu Hokuriku Systems Ltd. From 2014 to 2017, he was an Assistant with Tokyo Healthcare University. From 2017 to 2020, he was a Senior Lecturer with the Graduate School of Engineering, University of Fukui. He is currently an Associate Professor with the University of Fukui. His research interests include HAR, applying deep learning, and intelligent learning support systems. He is also a member of IPSJ.

MAKOTO KOSHINO received the Ph.D. degree from Kanazawa University, in 2004. He is currently an Associate Professor with the National Institute of Technology, Ishikawa College. His current research interest includes intelligent smartphones. He is a member of IEICE, IPSJ, and JSAI.