

# Deep Ensemble Learning for Human Activity Recognition Using Wearable Sensors via Filter Activation

WENBO HUANG, LEI ZHANG\*, and SHUOYUAN WANG, Nanjing Normal University, China

HAO WU, Yunnan University, China

AIGUO SONG, Southeast University, China

During the past decade, human activity recognition (*HAR*) using wearable sensors has become a new research hot spot due to its extensive use in various application domains such as healthcare, fitness, smart homes and eldercare. Deep neural networks, especially convolutional neural networks (*CNNs*) have gained a lot of attention in *HAR* scenario. Despite exceptional performance, *CNNs* with heavy overhead is not the best option for *HAR* task due to the limitation of computing resource on embedded devices. As far as we know, there are many invalid filters in *CNN* that contribute very little to output. Simply pruning these invalid filters could effectively accelerate *CNNs*, but it inevitably hurt performance. In this paper, we first propose a novel *CNN* for *HAR* that uses filter activation. In comparison with filter pruning that is motivated for efficient consideration, filter activation aims to activate these invalid filters from an accuracy boosting perspective. We perform extensive experiments on several public *HAR* datasets, namely UCI-HAR (*UCI*), OPPORTUNITY (*OPPO*), UniMiB-SHAR (*Uni*), PAMAP2 (*PAM2*), WISDM (*WIS*) and USC-HAD (*USC*) which show the superiority of the proposed method against existing state-of-the-art (*SOTA*) approaches. Ablation studies are conducted to analyze its internal mechanism. Finally, the inference speed and power consumption are evaluated on an embedded *Raspberry Pi Model 3 B plus* platform.

CCS Concepts: • **Human-centered computing** → Ubiquitous and mobile computing; Ubiquitous and mobile devices; Mobile devices.

Additional Key Words and Phrases: sensor, convolutional neural network, human activity recognition, deep learning, filter activation

## ACM Reference Format:

Wenbo Huang, Lei Zhang, Shuoyuan Wang, Hao Wu, and Aiguo Song. 2022. Deep Ensemble Learning for Human Activity Recognition Using Wearable Sensors via Filter Activation. *J. ACM* 1, 1 (August 2022), 24 pages. <https://doi.org/10.1145/3551486>

## 1 INTRODUCTION

### 1.1 Background

During the past decade, with rapid development of sensor and Internet of Thing technology, human activity recognition (*HAR*) [1–3] using wearable sensors such as accelerometer and gyroscope has

---

\*Corresponding Author

Authors' addresses: Wenbo Huang, wenbohuang1002@outlook.com; Lei Zhang, leizhang@njnu.edu.cn; Shuoyuan Wang, 21180403@njnu.edu.cn, Nanjing Normal University, No.2, Xuelin Road, Qixia Street, Qixia District, Nanjing, Jiangsu, China, 210023; Hao Wu, Yunnan University, University Town East Outer Ring South Road, Chenggong District, Kunming, Yunnan, China, 650500, haowu@ynu.edu.cn; Aiguo Song, Southeast University, No.2, Sipailou, Sipailou Street, Xuanwu District, Nanjing, Jiangsu, China, 210096, a.g.song@seu.edu.cn.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2022 Association for Computing Machinery.

0004-5411/2022/8-ART \$15.00

<https://doi.org/10.1145/3551486>

become a new research trend. Due to its obvious advantages (*e.g.*, smaller size, lower price, portability) over other sensor modalities such as camera, such sensors have been extensively leveraged to recognize human activities, which plays a vital role in wide range of application domains [4–9] such as health-care, sport tracking, smart home and game console designing. In principle, sensor-based *HAR* can be treated as a multivariate time series classification problem, which can be handled by conventional machine learning algorithms such as naive Bayes networks and support vector machines in combination with heuristic handcrafted features, *e.g.*, variance, mean value, and other statistics in temporal or frequency domain. However, such heuristic handcrafted features rely on expert experience or domain knowledge, which requires expensive human intervention and generally lacks scalability for a large range of activity recognition tasks.

Recent years have witnessed significant advances in Deep Learning (*DL*) community [10]. It is worthwhile mentioning that convolutional neural networks (*CNNs*) [11–13] have made major breakthroughs in *HAR* area, which become a favorable deep learning architecture due to an obvious advantage of automatic feature extraction. For example, Zeng *et al.* [14], Ronao *et al.* [15] and Yang *et al.* [16] at the earliest time adopted *CNN* over multivariate time series to perform *HAR* tasks, where convolutional filters can be directly applied along multimodal sensor signals to capture local dependencies. Ravi *et al.* [17] and Jiang *et al.* [18] proposed an efficient *HAR* method with the use of *CNN* performed on the spectral domain of sensor data. Kim *et al.* [19] introduced an interpretable *CNN*, which uses group *LASSO* method to produce sparse convolution filters for selecting important sensor signals. Ignatov *et al.* [20] utilized *CNN* for *HAR* to automatically extract discriminative features, which is combined with handcrafted features to preserve global information about sensor time series. Gao *et al.* [21] proposed a three-layer *CNN* to handle smartphone-based inertial sensor data for *HAR*, which performs considerably better than conventional machine learning algorithms on two standard *HAR* benchmarks. Ordóñez *et al.* [22] proposed a hybrid model called *DeepConvLSTM*, which combines *CNNs* and *LSTMs* to simultaneously capture local and global features for *HAR*. Despite exceptional performance, deep *CNNs* with heavy computational overheads are not the best choice for *HAR* task, due to the limitation of computing resource on embedded devices.

## 1.2 Motivation of our research

Actually, deep *CNNs* are severely restrained for *HAR* applications, which is mainly because the computational cost caused by stacked convolutional layers is not affordable for embedded devices. Recently, many researches have been devoted to investigating how to prune or compress convolutional networks. As mentioned above, filter is a core component in modern *CNNs*. [23] shows that there are more or less unimportant filters (invalid filters) in *CNN*. [24] adopts feature boosting and suppression (*FBS*) to skip invalid filters during training stage. Generally speaking, these invalid filters affect and contribute far less to output. So far, most investigations [25, 26] have been confined to detect these invalid filters and prune them for efficient inference, which aims to maintain model performance [27]. However, simply pruning such invalid filters [24] might hurt the generality ability of deep *CNNs*, which is not the best choice for *HAR* task. It still remains unclear whether these invalid filters can be transformed into valid ones for improving final recognition performance.

The similar story also takes place in the ensemble learning arena, in which a weak classifier can be lifted or boosted by the combination with other weak classifiers. Motivated by an idea of ensemble learning like boosting [28], it is conjectured that these invalid filters may be once more useful and contribute much to activity prediction. Although a single invalid filter is week or poor, their mutual enhancement is sometimes able to produce significant information gain. From an accuracy boosting perspective, it deserves deeper investigation whether such weak filters within one network can be lifted by incorporating other weak filters from external networks. That is to

say, it is possible that these invalid filters can be reactivated again by absorbing useful information from external networks, which can at the same time preserve the same network structure as before. As a result, such invalid filters can be turned into valid ones after activating operation, which leads to a considerable performance improvement without incurring extra computational burden, hence being vital for *HAR* according to lightweight property.

### 1.3 Key contributions

This paper aims to explore the potential of reactivating such invalid filters for *HAR*. Taking an inspiration from ensemble learning [28], we for the first time propose a new *CNN* that performs filter activation in ubiquitous *HAR* scenario. Instead of simply pruning invalid filters, we prefer to reactivating them via absorbing useful information from other networks (external), where the information source used to activate invalid filters needs to be selected carefully. Because it is unrealistic to simultaneously deploy multiple deep models on embedded devices for efficient inference, directly adopting ensemble learning is infeasible for *HAR*. Therefore, we propose to train multiple networks in parallel with the same number of layers and filters across every layer, which are initialized through different hyper-parameters for raising diversity. Specifically, an entropy-based measure is used to carefully select meaningful filters from external networks to replace such invalid filters of internal network. An adaptive weighting strategy is used to share information (weights) among multiple networks. In particular, the filter activation is performed in layer level rather than filter level. The main advantages of this paper are summarized as follows: Instead of filter pruning, we develop a new *CNN* for *HAR* that aims to reactive such invalid filters by absorbing useful information from outside networks, hence leading to a significant performance gain. In comparison with ensemble learning, this filter activation approach requires only one network instead of multiple networks to be deployed on resource-restrained embedded devices, which is more suitable for activity inference in terms of lightweight property. At negligible computational burden, the filter activation approach can be flexibly plugged into current *CNNs* that without altering network structure.

In addition, we highlight the novelty of the work by comparing filter activation against three popular machine learning approaches. We thoroughly investigate the differences between them, which are summarized in Table 1: 1) The proposed approach is very similar to transfer learning, which also aims to transfer information between networks. However, transfer learning is a ‘two-step’ process, which need a retraining process with supervision for fine-tuning. The difference between filter activation and transfer learning lies in that filter activation is a ‘single-stage’ process without the retraining process; 2) The research motivation of filter activation is opposite to pruning. The difference between filter activation and filter pruning lies in that filter activation involves training multiple networks, which does not need to modify network structure. Filter activation can yield a significant performance gain by absorbing useful information from outside networks and meanwhile maintaining almost the same inference speed; 3) The filter activation inherits an idea ensemble learning, and both of them involve training multiple networks in parallel. While for filter activation, inference is completed with only one network rather than multiple networks, which is more efficient. It utilizes an entropy-based measure to select meaningful filters and transfer information between multiple networks by calculating the weighting coefficient adaptively.

The rest of this paper is organized as follows. In Section 2, we review related works. Section 3 introduces an overview of the proposed method. In Section 4, the details of experiment setting are presented, and we then conduct extensive ablation studies on various benchmark datasets. Main results are discussed. Finally, we draw a conclusion in Section 5.

Table 1. Difference between several methods

Methods	Without modifying structure	One stage	Without supervision	Inference using one network
Filter pruning	No	Yes	Yes	Yes
Transfer learning	Yes	No	No	Yes
Ensemble learning	Yes	Yes	No	No
Filter activation	Yes	Yes	Yes	Yes

## 2 RELATED WORKS

**Filter Pruning and Activation.** Especially in computer vision field, many researchers have put their sight on filter pruning for model compression. In order to speed up network's inference, [23] prunes invalid filters by  $l_1$ -norm criterion. [24] apply pruning technique during training stage to slim network structure. [29] proposes to use spectral clustering to detect invalid filters. [25] reduces the convolutional layer's redundancy via subspace clustering technique. [30] discovers that the invalid filters can be reactivated to produce more powerful feature representations in visual recognition tasks. In wearable HAR scenario, [26] designs a multi-sensor fusion with ensemble pruning system (*MSF-EP*). This system transforms the problem of multiple sensors configuration into the problem of multi-classifier ensemble pruning. The shortcoming of filter pruning is that it simply removes the unimportant (invalid) filters, which inevitably hurts the generality ability of deep CNNs. On the contrary, filter activation aims to re-activate such invalid filters instead of simply pruning them, which can further lift model performance. In addition, compared to pruning, filter activation does not need to modify network structure.

**Attention mechanism.** Attention mechanism has been widely used to pay more attention to these valid and meaningful filters. In order to investigate related temporal features, [31] discusses attention mechanism in HAR research and treat it as a data-driven method. [32] locates and classifies weakly labeled sensor data via applying attention on CNN. [14] uses continuous attention to improve recurrent network's performance in HAR tasks. [33] presents a framework based on a bran-new combination of CNN attention mechanism and Gated Recurrent Unit (*GRU*) network. This framework can capture both spatial and temporal features, in which attention mechanism can enlarge the influence of these valid filters. The difference between filter activation and attention mechanism is that attention mechanism treats these invalid filters as useless information, which ignores these invalid filters' potential effect. Instead, filter activation make such invalid filters become once more useful by absorbing useful information from outside networks, which can fully exploit their potential.

**Distillation, Mutual and Ensemble learning.** In order to absorb useful information from outside, filter activation need to train multiple networks at the same time, which is very similar to distillation, mutual [34] and ensemble learning [28] at this point. However, distillation generally requires two stages to train a teacher model and a student model respectively, which then enables the teacher model to guide the training process of the student model. While for filter activation, it only needs to perform one stage to train multiple networks in parallel. The obvious difference between mutual learning and filter activation is that mutual learning need an additional mutual loss in the learning process in order to supervise multiple networks to learn from each other. While for filter activation, it does not require this supervised loss. In addition, it is worth mentioning that filter activation is performed at every epoch instead of iteration steps, which can decrease communication cost among networks compared to mutual learning. [28] proposes a novel ensemble extreme learning machine technique for HAR using smartphones. Filter activation has obvious

advantages over [28], which is more lightweight and only needs to deploy a single network on embedded systems. The shortcoming of ensemble learning lies in that it has to deploy multiple networks during inference stage, which is computationally expensive for embedded devices. While for filter activation, inference requires only one network to be deployed, which performs better than ensemble learning in light of lightweight property.

### 3 MODEL

In this section, we introduce our model in details. In 3.1, we present an overview of the proposed method in *HAR* scenario with filter activation. In 3.2, we first explore how to select more useful information source to reactivate invalid filters. In 3.3, we discuss two criteria in order to measure the filter's information. In 3.4, we study how to perform filter activation in two-network case. In 3.5, the activation algorithm is further extended to multi-network case.

#### 3.1 Overview

Activity recognition can be seen as a typical multivariate time series classification problem, where data preprocessing is one crucial step that might have an essential impact on classification performance. There are several main preprocessing procedures that are commonly used to transform multimodal sensor data before feeding them into one classifier, including filtering noise, handling missing value, standardization, and segmentation. We will detail each of them as follows. **Denoising:** A 3<sup>rd</sup> order Butterworth low-pass filter with the cutoff frequency, *i.e.*, 20 Hz is often adopted to remove the impact of noise on raw sensor data. Besides, another Butterworth low-pass filter with the cutoff frequency, *i.e.*, 0.3 Hz is applied over accelerometer time series to further separate the gravitational and body motion components. **Handling missing value:** Difference sensors placed over different body positions can generate multi-dimensional time series data at a constant frequency, which is used to monitor human activities. Such sensor readings are easily missing because of sensor malfunction. A simple method is to directly discard these samples with missing values, which will compromise classification accuracy. An alternative is to use interpolation method that assumes the missing readings are close to the nearest measurements. More methods can be acquired in the literature [35], which rely on the sensor modalities used and the activities of interest. **Standardization:** Due to heterogeneous sensor modalities, it usually needs to scale raw sensor data into a certain range, which enables deep models to converges faster. Standardization is a mainstream rescaling method to normalize input samples by subtracting the mean and dividing the standard deviation, which may effectively avoid influence of outliers. It is worth noting that the preprocessing should only be performed on the training set, which is then conducted over the validation or test set. **Segmentation:** Unlike image data, a single sensor reading is hard to represent the characteristics of a specific activity. According to the continuity of sensor time series, the sliding window technique with a fixed window length has been widely used to segment the steaming sensor data into sequences, where an overlap rate between adjourning windows are generally required to maintain the continuity of sensor data. All produced sequences are then fed into deep models for final classification. We present an entire overview of the proposed method, as shown in Fig.1.

#### 3.2 Activation Information source

How to select more useful information source is one key step. In this part, we introduce three ways.

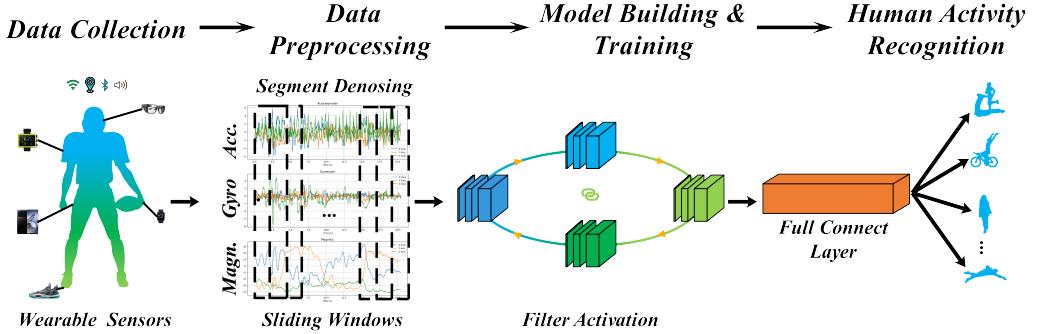


Fig. 1. The overview of the proposed HAR method with filter activation. The curves are sensor time series.

**3.2.1 Gaussian Noise Activation.** Because Gaussian noise has been extensively leveraged to boost the performance of CNN during weight initialization [10], we may select Gaussian noise  $N(0, \sigma_t)$  as the information source to replace invalid filters. These invalid filters contribute less to the output, and their  $l_1$ -norm is often smaller than that of valid filters. Fig.2 is the sketch map of selecting Gaussian noise as information source. After Gaussian noise activation, these invalid filters can acquire a larger  $l_1$ -norm and produce more contribution to output. If there is too much noise, the model might have trouble in convergence. Thus, we have to decrease  $\sigma_t$  along with time.

$$\sigma_t = \alpha^t \quad (0 < \alpha < 1) \quad (1)$$

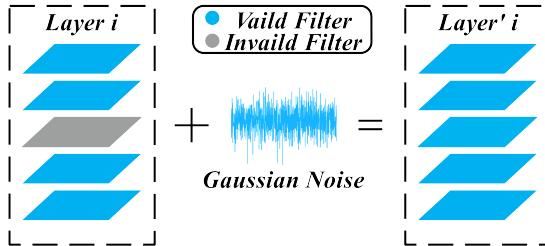


Fig. 2. Gaussian Noise Activation

**3.2.2 Internal Activation.** This method performs filter activation within a single network, in which these valid filters that have larger  $l_1$ -norm are selected as information source. We use the valid filters to activate the invalid filters (have smaller  $l_1$ -norm). To be specific, all filters are first sorted according to  $l_1$ -norm. To perform the internal activation, we set a threshold  $\gamma$ . The filters are treated as invalid filters if their  $l_1$ -norm is smaller than  $\gamma$ , while other filters are seen as valid ones. Then we replace the  $i$ -th smallest filter's weights with the  $i$ -th biggest filter's ones. Fig.3 is the sketch map of internal activation. Although these invalid filters have been activated and have more effect to output, the internal activation could not provide any new information gain [25, 30, 31] to network since their information source is internal filters.

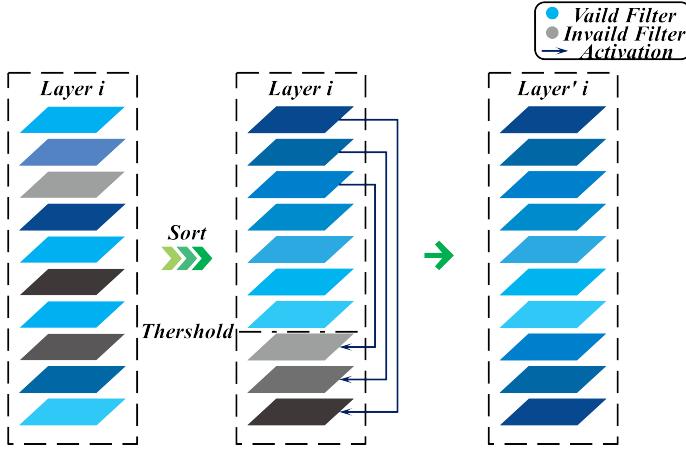


Fig. 3. Internal Activation

**3.2.3 External Activation.** In order to overcome the shortcomings of Gaussian noise and internal activation, we adopt external filters as information source. For simplicity, we first discuss two-network case, where the network  $M_1$  and  $M_2$  are trained in parallel. During training stage, the external activation method utilizes  $M_1$ 's valid filters as information source to replace  $M_2$ 's invalid filters. Compared with internal activation, external activation has two adjustments:

- Although two networks with the same structure are trained in parallel, the locations of the filters across the same layer are possibly different. Because there is no one-to-one mapping between filters, it is hard to align the different filters within a layer in the filter activation process. Since the two networks have different initialized weights and locations of invalid filters [28, 30, 36], the simple activation operation in filter level may hurt the layer's consistency. Thus we conduct external activation in a layer level. All of  $M_1$ 's meaningful filters in a certain layer are used to activate  $M_2$ 's invalid filters at the same layer (also  $M_1$  to  $M_2$ ). After external activation, the mutual information can be learned from each other by  $M_1$  and  $M_2$ .

- We weight the internal and external information along with the external activation:

$$W_i^{M'_2} = \alpha W_i^{M_2} + (1 - \alpha) W_i^{M_1} \quad (0 < \alpha < 1) \quad (2)$$

where  $W_i^{M_2}$  is the weights of  $M_2$  in the  $i$ -th layer and  $W_i^{M_1}$  is the weights of  $M_1$  in the  $i$ -th layer. Assuming that  $W_i^{M_1}$  is less informative than  $W_i^{M_2}$ ,  $\alpha$  should be larger than 0.5 and vice versa. External activation between two networks is shown in Fig.4 and Eq.2. Fig.4 helps to explain filter activation's two critical problems: 1) how to measure  $W_i^{M_1}$  and  $W_i^{M_2}$ 's information; 2) how to choose  $\alpha$  (weighting coefficient). These problems will be discussed in the following Section 3.

In order to raise two network's diversity, we deliberately initialize two networks differently and set their hyper-parameters (e.g., optimizer, learning rate...) to be different from each other [36], which make the two networks have different weights. In the two-network case, it is worthwhile to mention that after activation operation the two networks' weights will be the same if the activation is only implemented at each epoch. At other iteration steps, their weights are still different from each other due to the diversity. In 3.5, the phenomenon will disappear when it is extended to multi-network case.

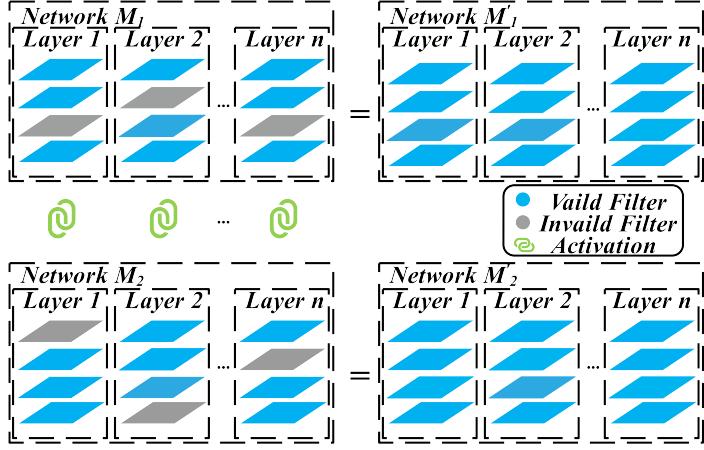


Fig. 4. External Activation between two networks

### 3.3 Filter or layer Information's Calculating Criteria

In this part, we discuss two information calculation criteria in the level of filter and layer respectively.

**3.3.1 \$l\_1\$-norm.** \$l\_1\$-norm is used to measure filter's information in previous sections. \$W\_{i,j} \in \mathbb{R}^{N\_i \times K \times K}\$ represents \$j\$-th filter's weight at the \$i\$-th layer. \$N\_i\$ is the number of filters (in \$i\$-th layer). The \$W\_{i,j}\$'s \$l\_1\$-norm is expressed as follows:

$$\|W_{i,j}\|_1 = \sum_{n=1}^{N_i} \sum_{k_1=1}^K \sum_{k_2=1}^K |W_{i,j}(n, k_1, k_2)|. \quad (3)$$

\$l\_1\$-norm is widely used as layer or filter information's calculating criteria. However, it is not always true that smaller norm has less importance [24, 29]. For example, all 1 filters sometimes perform worse than the filters arranged from 0 to 1 regularly. [23] shows that the \$l\_1\$-norm's use must meet certain pre-requisite. That is to say, filter activation according to \$l\_1\$-norm may deteriorate valid filters.

**3.3.2 Entropy.** The above disadvantages can be attributed to the reason that \$l\_1\$-norm only pay attention to the absolute value of filter's weight and ignores its variation. Thus, we need to direct our attention to the weight's variation. Assuming \$W\_{i,j}(n, k\_1, k\_2) = a\$ for \$n \in \{1, \dots, N\_i\}, k\_1, k\_2 \in \{1, \dots, K\}\$, all single values in \$W\_{i,j}\$ are identical. That is to say, every part of an input has an equal contribution to output under normal convolution operation. Regardless of absolute value of \$a\$, the filter loses its ability to find more important part of an input. Thus, we decide to measure weight's variation instead of its absolute value. We transform discrete distribution from continuous distribution. The range of value is divided into \$m\$ bins. Referring to related literature [26, 30], we can calculate each layers' weight \$W\_i \in \mathbb{R}^{N\_i \times N\_{i+1} \times K \times K}\$ as a whole to protect the consistency among layers:

$$H(W_{i,j}) = - \sum_{k=1}^B p_k \log p_k \quad (4)$$

in which \$B\$ and \$p\_k\$ denote the amount of bins and the probability of bin \$k\$ respectively.

### 3.4 Activation's Adaptive Weighting

In this part, two networks' weights from Eq.2 are weighted by an adaptive weighting strategy.  $W_i^{M_1}$  is the weight of layer  $i$  in network  $M_1$ .  $H(W_i^{M_1})$  denotes the information of the layer in network  $M_1$  and we can calculate  $H(W_i^{M_1})$  according to Eq.4. Calculating the coefficient  $\alpha$  needs to meet two conditions:

- If  $H(W_i^{M_1}) = H(W_i^{M_2})$ , Eq.2's  $\alpha$  should be 0.5. If  $H(W_i^{M_1}) < H(W_i^{M_2})$ ,  $\alpha$  needs to larger than 0.5.
- Either  $H(W_i^{M_1}) < H(W_i^{M_2})$  or  $H(W_i^{M_1}) > H(W_i^{M_2})$ , every network must contribute to their own inherent information.

An adaptive weighting coefficient [23, 30] is applied to meet the above two requirements:

$$\alpha = A \times \left( \arctan \left( c \times \left( H(W_i^{M_2}) - H(W_i^{M_1}) \right) \right) \right) + 0.5 \quad (5)$$

where  $A$  and  $c$  in Eq.5 are a pair of fixed hyper-parameters to calculate weighted coefficient. In Fig.5, we visualize the function that successfully meets two above requirements.

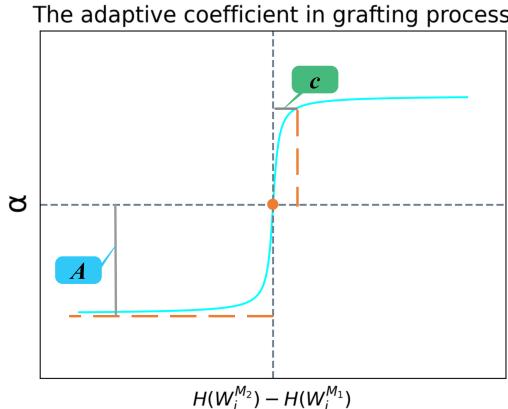


Fig. 5. Adaptive Weighting Coefficient

### 3.5 Multiple Networks' Activation

In this part, this two-network case can be easily extended to multiple-network case. We train multiple networks in parallel. During each training epoch, all these networks can exchange information from each other in a circular way, *i.e.*,  $M_1 \rightarrow M_2 \rightarrow \dots \rightarrow M_{k-1} \rightarrow M_k \rightarrow M_1$ , as illustrated in Fig.6. For example, the network  $M_{k-1}$  gives out its information to  $M_k$  by using  $W_l^{M_{k-1}}$  to update  $W_l^{M_k}$ , where the above adaptive weighting strategy can be used to calculate the weighting coefficient between them. Every network is able to fully absorb useful information from other outside networks after a specific number of training epochs. Though different training hyper-parameters are used to raise diversity, the performance of each network still tends to be very close to each other under external activation. Without loss of generality, the first network is always chosen to be deployed in the multi-network setting. Overall, we develop a new filter activation algorithm for activity recognition where we propose an entropy-based criterion and adaptive weighting strategy to perform filter activation in layer level rather than filter level, as presented in Algorithm 1.

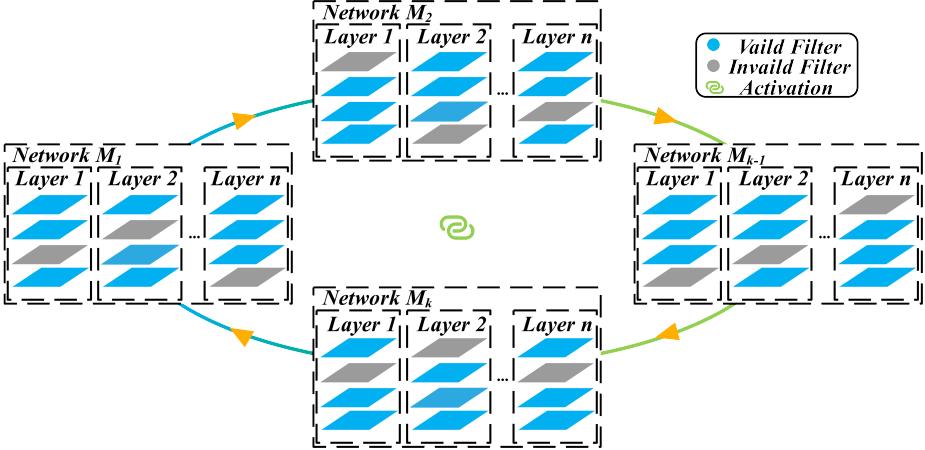


Fig. 6. Multiple Networks' Activation

**Algorithm 1** External Activation**Input:**

- 1:  $K$ : Number of Networks;  $M_1$  to  $M_k$ : The  $1^{st}$  to  $k^{th}$  networks;
- 2:  $L$ : Number of Layers;  $N_T$ : Each epoch's iteration number;
- 3:  $\mathcal{N} \in \{1, \dots, N_{\max}\}$ : Iterations of training;
- 4:  $W_l^{M_k}$ : Initial Weight of  $L$  layers in  $K$  networks;
- 5:  $\lambda_k$ : Hyper-parameters of  $k$  networks;  $\mathcal{D}$ : Training Set
- 6: **for**  $\mathcal{N}$  in range  $(1, N_{\max})$  **do**
- 7:   **for**  $k$  in range  $(1, K)$ ,  $l$  in range  $(1, L)$  **do**
- 8:     On the base of  $\mathcal{D}$  with  $\lambda_k$ , update  $W_l^{M_k}$ ;
- 9:     **if**  $\mathcal{N} // N_T = 0$  **then**
- 10:       Calculate weight coefficient  $\alpha$  according to Eq.5;
- 11:        $W_l^{M_k} = \alpha W_l^{M_k} + (1 - \alpha) W_l^{M_{k-1}}$
- 12:     **end if**
- 13:   **end for**
- 14: **end for**

**4 EXPERIMENT**

In this section, we introduce our experiments in details. 4.1.1 presents our experimental setup. In 4.1.2 and 4.1.3, several benchmark HAR datasets and evaluation metrics we use are introduced. We discuss the quantitative comparisons in 4.1.4. In 4.2, we conduct further ablation study.

**4.1 Experimental setup**

**4.1.1 Network's setup.** After taking the balance of classification accuracy and computational burden into consideration, we choose several 3-layer CNNs as our backbones. In order to proceed a fair comparison, we set each method's hyper-parameters to be the same. We select Adam as optimizer. For the baseline model, the learning rate is  $1e^{-4}$  and it decays to 95% at every 10 epochs. The number of epoch is 200 because we find that too few epoch makes models hard to converge and the test accuracy tends to be stable after 150 epochs. The batch size is different according to various

datasets. We select a Rectified Linear Unit (*ReLU*) as activation function. After 3 convolutional layers are Full Connect (*FC*) layer and *Softmax*. We choose PyTorch as deep learning framework. All the experiments are conducted on a server for deep learning (*OS: Ubuntu 20.04; GPU: 24 GB GeForce RTX 3090; CPU: 6th Gen Intel i7-6850K; RAM: 64 GB*). More details of our networks are illustrated in Table 2, where  $C(L_s)$  means the convolutional layer has  $L_s$  feature maps.

Table 2. Network's details

Dateset	1 <sup>st</sup> Layer	2 <sup>nd</sup> Layer	3 <sup>rd</sup> Layer	<i>FC, Softmax</i>
<i>UCI</i>	$C(64)$	$C(128)$	$C(256)$	✓
<i>OPPO</i>	$C(64)$	$C(256)$	$C(384)$	✓
<i>Uni</i>	$C(64)$	$C(256)$	$C(384)$	✓
<i>PAM2</i>	$C(128)$	$C(256)$	$C(384)$	✓
<i>WIS</i>	$C(64)$	$C(128)$	$C(256)$	✓
<i>USC</i>	$C(64)$	$C(128)$	$C(256)$	✓

**4.1.2 Datasets.** During recent years, the researchers in embedded systems, persuasive and ubiquitous computing, and human-computer interaction have built various human activity datasets to benchmark machine learning algorithms for activity recognition research. In this paper, we select several mainstream public *HAR* datasets to analyze and recognize various types of human activities. Before feeding these sensor signals into networks, we use sliding window to divide sensor signals into tensors with different shapes. UCI-HAR, UniMiB-SHAR, WISDM and USC-HAD datasets are divided into training set, validation set and test set at a ratio of 7:1:2 while OPPORTUNITY and PAMAP2 datasets are subject-wisely divided into three parts. The details of data preprocessing are summarized in Table 3.

**UCI-HAR dataset (*UCI*)** [37]. In order to provide a benchmark for comparing various machine learning algorithms, several researchers from the *University of California Irvine* conduct this data collection. In a supervised scenario, 30 volunteers (19-48 years old) who join the data collection were asked to wear a *Samsung Galaxy S2* smartphone on their waists and perform 6 different types of activities of daily living (*ADLs*) including *walking downstairs* and *upstairs*, *sitting*, *standing*, *lying*, *walking*. The sensor signals are sampled (sampling rate: 50 Hz) by triaxial angular velocity and acceleration sensors.

**OPPORTUNITY dataset (*OPPO*)** [38]. The project was conducted by Daniel *et al.* in *University of Sussex*. They built a rich sensor environment that consists of 15 wireless and wired networked sensor systems. There are 72 sensors of 10 modalities within the sensor system. The sampling rate is set to 30 Hz. All in all, 17 types of activities in a breakfast scenario were recorded from four subjects. On each subject was equipped with rich wearable sensor nodes for the inference of human activities.

**UniMiB-SHAR dataset (*Uni*)** [39]. Daniela *et al.* in *University of Milano-Bicocca* created this new acceleration dataset. The samples were acquired by a smartphone with *Android OS*. The sampling rate is 50 Hz. The whole dataset was designed for monitoring human activity and detecting falls. 30 volunteers ranging from 18 to 60 years contributed to all 11771 samples.

**PAMAP2 dataset (*PAM2*)** [40]. This dataset was collected by *Department of Augmented Vision German Research Center of Artificial Intelligence*. Within the PAMAP (*Physical Activity Monitoring for Aging People*) project, the researchers recorded 18 types of activities consisting of *walking*, *cycling*, *rope jumping etc*, which are collected from 9 subjects. Each subject wears 3 Inertial Measurement

Units (*IMUs*) and a heart-rate-monitor attached to arm, ankle and chest respectively. The sampling rate is 100 Hz. The PAMAP2 dataset were created and made publicly available.

**WISDM dataset (WIS)** [41]. To collect enough sensor data for benchmarking supervised activity recognition task, the researchers from *Fordham University* enlisted 29 subjects who carry an *Android* smartphone to perform certain daily activities. Specifically, placing the smartphone in their front pants pocket, all the subjects were asked to do a set of activities including *sitting*, *standing*, *jogging*, *walking*, *ascending* and *descending stairs*. The acceleration signals are collected every 50ms. That is to say, there are 20 samples within every second. To ensure the quality of sensor data, one of the WISDM research team strictly supervise the whole data collection process.

**USC-HAD dataset (USC)** [42]. Zhang and A.Sawchuk from the *University of Southern California* created this dataset. The activity data is recorded by a high-performance sensing device called *MotionNode*, which is a 6-DOF inertial measurement unit (*IMU*) integrating a triaxial accelerometer, a triaxial gyroscope, and a triaxial magnetometer. There is total 14 volunteer subjects including 7 males and 7 females who take part in the data collection. During data collection process, the *MotionNode* were attached to each subject's right front hip and then connected to a laptop held on one subject's right hand with *USB* connection. 12 different kinds of activities such as *jumping*, *sitting* and *sleeping etc.* are collected at a constant sampling rate of 100 Hz.

Table 3. Data pre-processing

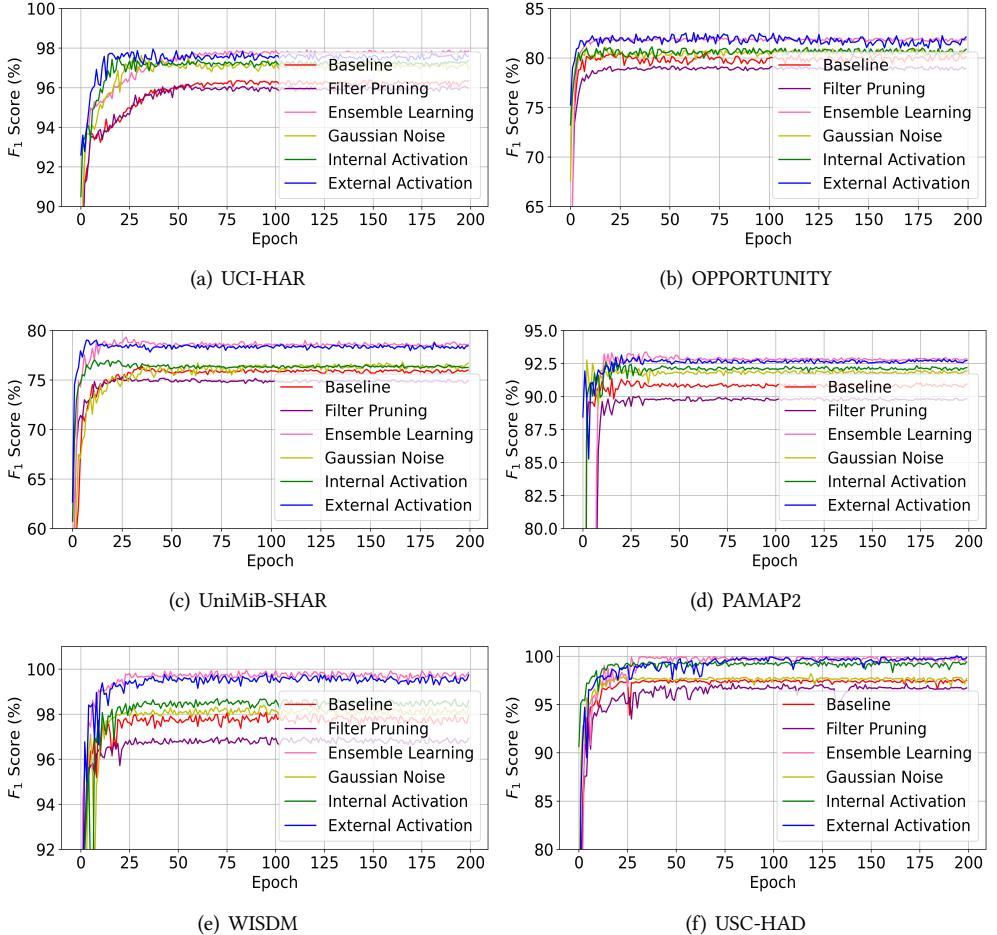
Dataset	Categories	Window Size	Overlap Rates
UCI	6	128×9	50%
OPPO	17	107×64	30%
Uni	17	151×3	50%
PAM2	12	120×86	50%
WIS	6	200×3	78%
USC	12	512×6	50%

**4.1.3 Evaluation Metrics.** For a more comprehensive evaluation, we select four diverse evaluation metrics including *Accuracy*, *F1-measure*, *Informedness* and *Markedness*, which can be mathematically formulated as follows:

$$\begin{aligned}
 TPR (\text{Recall}) &= \frac{TP}{TP + TN}, TNR = \frac{TN}{TN + FP}, \\
 PPV (\text{Precision}) &= \frac{TP}{TP + FP}, NPV = \frac{TN}{TN + FN}, \\
 F1 - \text{measure} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \\
 \text{Accuracy} &= \frac{TP + TN}{TP + FP + FN + TN}, \\
 BM (\text{Informedness}) &= TPR + TNR - 1, \\
 MK (\text{Markedness}) &= PPV + NPV - 1.
 \end{aligned} \tag{6}$$

In Eq.6, *TP*, *FN*, *FP*, and *TN* denote true positives, false negatives, false positives, and true negatives respectively.

**4.1.4 Quantitative Comparison.** After conducting experiments 5 times and calculating the mean value, we present our experiment results in this part. The filter activation is compared with

Fig. 7. Validation  $F_1$  curves on different datasets

the 3-layer baseline *CNNs* and other recent *SOTA* algorithms. The results are summarized in Table 4. Setting  $A$  to be 0.4 and  $c$  to be 100, we select 6 networks to evaluate the filter activation. The best number of networks are further explored in the following ablation experiments. As shown in Table 4, maintaining the number of parameters at a relatively low level, the accuracy improvement caused by external activation is the highest due to richer information from multiple networks that are able to provide more meaningful filters. According to the validation results from UCI-HAR (Fig.7(a)) and OPPORTUNITY (Fig.7(b)), the external activation outperforms their corresponding baselines 1.29% and 2.39% respectively. In comparison with baselines from UniMiB-SHAR (Fig.7(c)) and PAMAP2 (Fig.7(d)), the external activation produces 1.93% and 2.1% accuracy improvement. On WISDM (Fig.7(e)) and USC-HAD (Fig.7(f)), the accuracy rates are improved by external activation at 1.94% and 2.47% respectively when compared with their baselines. We extend the experimental results by adding more baselines including filter pruning and ensemble learning. We first compare filter activation with the competing pruning approaches. For fair comparisons, the baseline network structures for filter activation and pruning are the same, in which there are

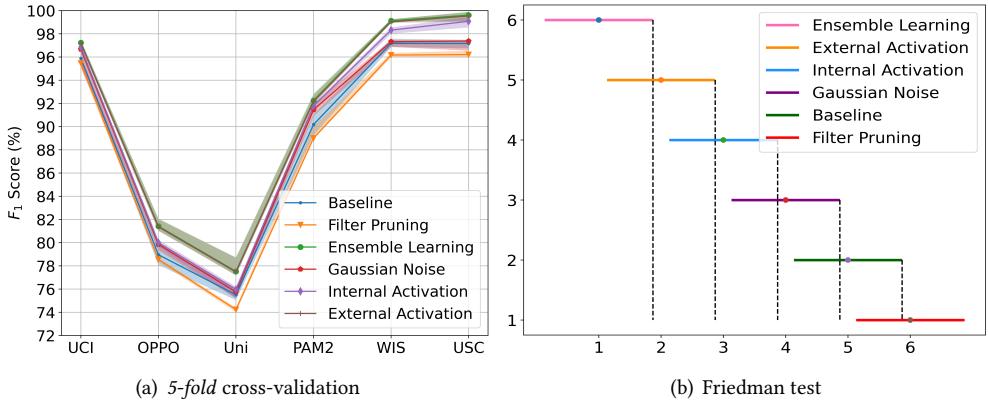


Fig. 8. Cross-validation &amp; Friedman test

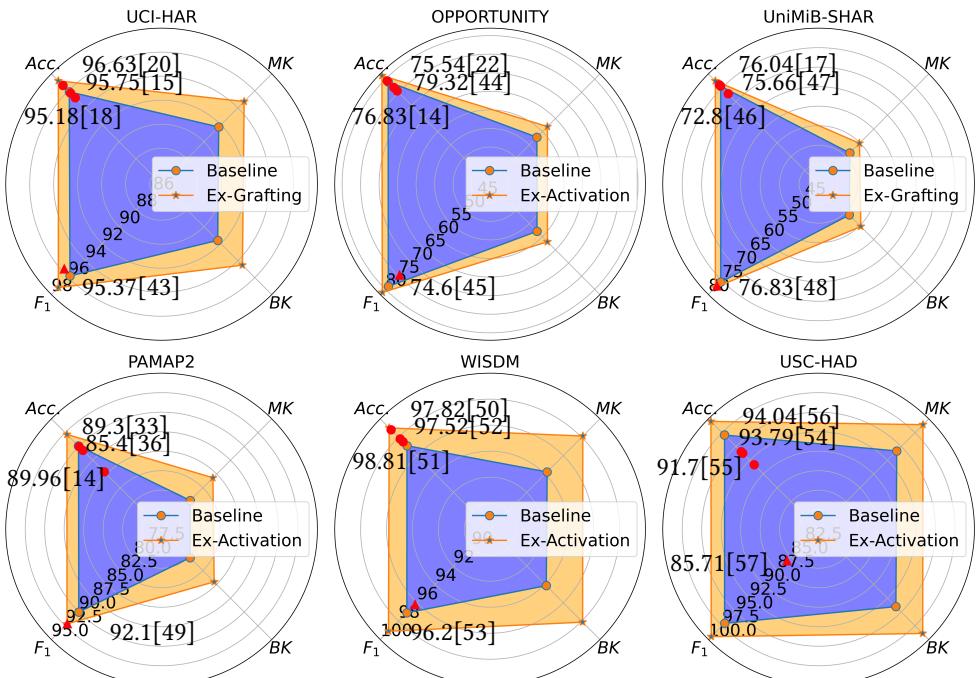


Fig. 9. Radar chart in terms of four evaluation metrics (%), see Table 4)

about 50% filters identified as invalid to be removed from each layer in the baseline network. The other training hyper-parameters are consistent with those in the first network used to perform filter activation. Results from Table 4, it can be seen that filter pruning performs even worse than the baseline network. We may find that filter activation can easily beat filter pruning on every dataset setting, which confirms once again that filter activation can improve the potential of convolutional networks by encouraging filters to learn complementary information from outside

networks. Then we experimentally explore the difference between ensemble learning and filter activation that involve multiple networks. For fair comparisons, we ensemble six networks in this experiment, where their hyperparameters' setting is consistent with the corresponding setting for filter activation. The model accuracy with parameter usage are summarized in Table 4. We can find that six-network ensembles perform slightly better than six-network activation. There exists small gap caused by filter activation because such valid filters might also be influenced by invalid ones, hence leading to negligible information loss. But filter activation only maintains one network instead of six networks for inference, which is significantly superior to ensemble learning according to lightweight property. The results are consistent with our analyses.

Besides, we also conduct *5-fold* cross-validation (Fig.8(a)) to prove the improvement of our method is reliable. On the basis of cross-validation, we conduct further Friedman test (Fig.8(b)) on 6 datasets and 4 models to show that the improvement of external activation is conspicuous when compared with the baseline. These results demonstrate that the accuracy boost is robust and usual phenomenon after adopting external activation.

The external activation is also compared with recent SOTA algorithms. The comparison results are summarized in Table 4. In particular, a radar chart is used to illustrate relative performance gain according to several comprehensive evaluation metrics including  $F_1$  score, accuracy, etc. The results from Fig.9 show that filter activation could achieve the best performance among all the learning algorithms. For example, on UCI-HAR dataset, the external activation outperforms [15], [18], [20] and [43] by 1.43%, 2.0%, 0.55% and 1.81% respectively. In the case of OPPORTUNITY dataset, the external activation is superiors to [22], [14], [44] and [45] by 5.82%, 4.53%, 2.04% and 6.61% respectively. When it comes to UniMiB-SHAR dataset, there are 1.43%, 4.67%, 1.81% and 0.63% performance improvements caused by external activation compared with [17], [46], [47] and [48]. In the case of PAMAP2 dataset, the external activation surpasses [33], [14], [36] and [49] by 2.88%, 2.22%, 6.78% and 0.07% respectively. On WISDM dataset, our external activation outperforms [50], [51], [52] and [53] by 1.2%, 0.21%, 1.5% and 2.9% respectively. In terms of USC-HAD dataset, there are 5.75%, 7.84%, 5.5% and 13.19% accuracy improvements caused by external activation compared with [54], [55], [56] and [57].

## 4.2 Ablation Study

All ablation experiments are composed of 6 parts, which are conducted on UCI-HAR dataset and OPPORTUNITY dataset respectively. In 4.2.1, we investigate how to select useful activating information source. In 4.2.2, we explore the optimal number of networks in external filter activation. In 4.2.3, we visually analyze confusion matrices of external filter activation. In 4.2.4, the effectiveness of the proposed method is evaluated via counting the number of invalid filters. In 4.2.5, we analyze the influence of training diversities according to several key hyper-parameters such as learning rate and sample order. Finally, for efficient consideration, we perform real-time prediction on an embedded platform (*Raspberry Pi Model 3 B Plus*) in 4.2.6.

**4.2.1 How to select useful information source.** We perform extensive experiments to examine the impact of three different activating information sources. For fair comparisons, the same network structure and hyper-parameters are used as indicated in Section 4.1.1. Six networks are trained in parallel for external filter activation. In external activation case, all networks have almost the same performance after training process. Without loss of generality, we select the first network for our evaluation. As shown in Table 4, it can be seen that internal activation has very similar recognition results with ‘noise’ source, which suggests that reactivating internal filters within a single network

Table 4. Test results (%) and parameters (M)

		UCI	OPPO	Uni	PAM2	WIS	USC
Baseline	Acc.	95.89	78.97	75.54	90.08	97.08	97.07
	$F_1$	95.81	78.92	75.51	90.03	97.09	97.06
	BM	91.62	57.94	51.15	80.16	94.14	94.03
	MK	91.74	57.89	51.32	80.2	94.23	94.18
	Para.	0.34	1.67	0.34	0.34	1.48	0.4
Gaussian noise	Acc.	96.67	79.85	75.75	91.45	97.32	97.38
	$F_1$	96.57	79.81	75.74	91.35	91.32	97.39
	BM	93.29	59.68	51.4	82.76	94.75	94.76
	MK	93.33	59.58	51.56	82.77	94.69	94.72
	Para.	0.34	1.67	0.34	0.34	1.48	0.4
Filter pruning	Acc.	95.47	78.55	74.23	89.01	96.17	96.21
	$F_1$	95.43	78.54	74.27	88.98	96.17	96.22
	BM	90.87	57.09	48.45	77.8	92.41	92.4
	MK	90.86	57.06	48.57	78.2	92.35	92.31
	Para.	0.21	0.96	0.22	0.2	0.94	0.19
Ensemble Learning	Acc.	97.23	81.42	77.52	92.25	99.13	99.61
	$F_1$	97.21	81.24	77.48	92.23	99.09	99.58
	BM	94.58	61.97	55.41	84.56	98.14	99.01
	MK	94.49	61.96	55.39	84.48	98.12	98.95
	Para.	2.07	10.09	2.08	2.08	8.92	2.6
Internal activation	Acc.	96.84	80.0	75.96	91.79	98.32	99.08
	$F_1$	96.83	80.01	75.99	91.78	98.26	99.12
	BM	93.58	60.15	51.96	83.75	96.56	98.3
	MK	93.62	60.09	52.01	83.56	96.58	98.32
	Para.	0.35	1.69	0.36	0.36	1.5	0.41
External activation	Acc.	97.18	81.36	77.47	92.18	99.02	99.54
	$F_1$	97.18	81.21	77.46	92.17	99.1	99.48
	BM	94.54	61.94	55.33	84.54	98.11	98.9
	MK	94.47	61.93	54.84	84.32	98.13	98.95
	Para.	0.35	1.69	0.36	0.36	1.5	0.41
Results of other researches	95.75 [15]	75.54 [22]	76.04 [17]	89.3 [33]	97.82 [50]	93.79 [54]	
	95.18 [18]	76.83 [14]	72.8 [46]	89.96 [14]	98.81 [51]	91.7 [55]	
	96.63 [20]	79.32 [44]	75.66 [47]	85.4 [36]	97.52 [52]	94.04 [56]	
	95.37* [43]	74.6* [45]	76.83* [48]	92.1* [49]	96.2* [53]	85.71* [57]	

\*: Test  $F_1$  Score.

is not able to produce extra information gain. The external filters from multiple networks perform the best among all three information sources. Therefore, we consider external filter activation in the remaining experiments.

**4.2.2 The best number of networks in external filter activation.** For external filter activation, the number of networks is an important hyper-parameter. We analyze the influence of this parameter on recognition accuracy for UCI-HAR and OPPORTUNITY. The recognition accuracy for external filter activation with the number of networks ranging from 1 to 8 are summarized

in Table 5. One could clearly observe that the accuracy of external filter activation first gradually increases with more networks, and then decreases when the number of networks is larger than a certain number. It is worthwhile to note that 6 networks activation performs best among all the models. As we continue to raise the number of networks, the recognition accuracy slightly becomes worse. The filter activation method could help filters to learn more useful information from external networks, which is able to greatly enhance the network's representation ability. Unlike traditional ensemble learning, the external activation ensembles more useful filters into only a single network for final evaluation, which does not increase any memory and computational overhead.

Table 5. Test Accuracy(%) under different numbers of networks involved in filter activation

	<b>UCI</b>	<b>OPPO</b>
Baseline	95.89	79.14
2 networks activation	96.57	79.4
3 networks activation	96.6	79.55
4 networks activation	96.64	80.08
6 networks activation	<b>97.18</b>	<b>81.36</b>
8 networks activation	96.84	80.33

**4.2.3 Confusion matrices.** Fig.10 visually shows the performance of the proposed method through confusion matrices on PAMAP2 dataset. The figure presents one typical misclassification example, in which there are 89 ‘Standing’ examples to be misclassified as ‘Vacuum Cleaning’ by the baseline model. Our method can greatly reduce the number of misclassified activity samples down to 50. All in all, the filter activation can provide much lower errors, which confirms its superiority in ubiquitous activity recognition tasks.

**4.2.4 The effectiveness of filter activation.** We further evaluate the effectiveness of the proposed method by counting the number of invalid filters after training. The *3-layer CNN* without activation is selected as our baseline. We consider the same network structure with activation trained on OPPORTUNITY and UCI-HAR for our evaluation. The effectiveness is analyzed by changing different threshold  $\gamma$ , and the comparison results are shown in Fig.11. We first discuss the case on OPPORTUNITY dataset (Fig.11(a)). When the threshold is large, *i.e.*,  $\gamma=1$ , it can be observed that the invalid filters accounts for the large proportion (50% and 61.1% respectively). We continue to decrease the threshold  $\gamma$ . When  $\gamma$  is equal to  $1e^{-3}$ , the activation can maintain the number of invalid filters at a much lower ratio, *i.e.*, about 5.4%, while there are nearly a 37.2% ratio of invalid filters for the baseline model, which indicates that the filter activation is very beneficial for reducing the number of invalid filters. One also could observe a similar phenomenon on UCI-HAR dataset (Fig.11(b)), which verifies that filter activation do improve the representation ability of the network. However, as one continues to reduce  $\gamma$  to  $1e^{-3}$  or lower, the reduction of invalid filter indicates a saturating trend.

In order to prove that filter activation does affect model performance, we show the information sharing between multiple networks parallelly trained by calculating the number of invalid filters and information gain during the training process. The results are shown in Fig.12, in which the  $x$  axis represents the number of training epochs, *e.g.*, 0, 25, 50 75, 100. Specifically, the network's information is denoted as the sum of all the layers' entropy within the first network. On one hand, it can be seen that there is a large proportion of filters counted as ‘invalid’ before training and

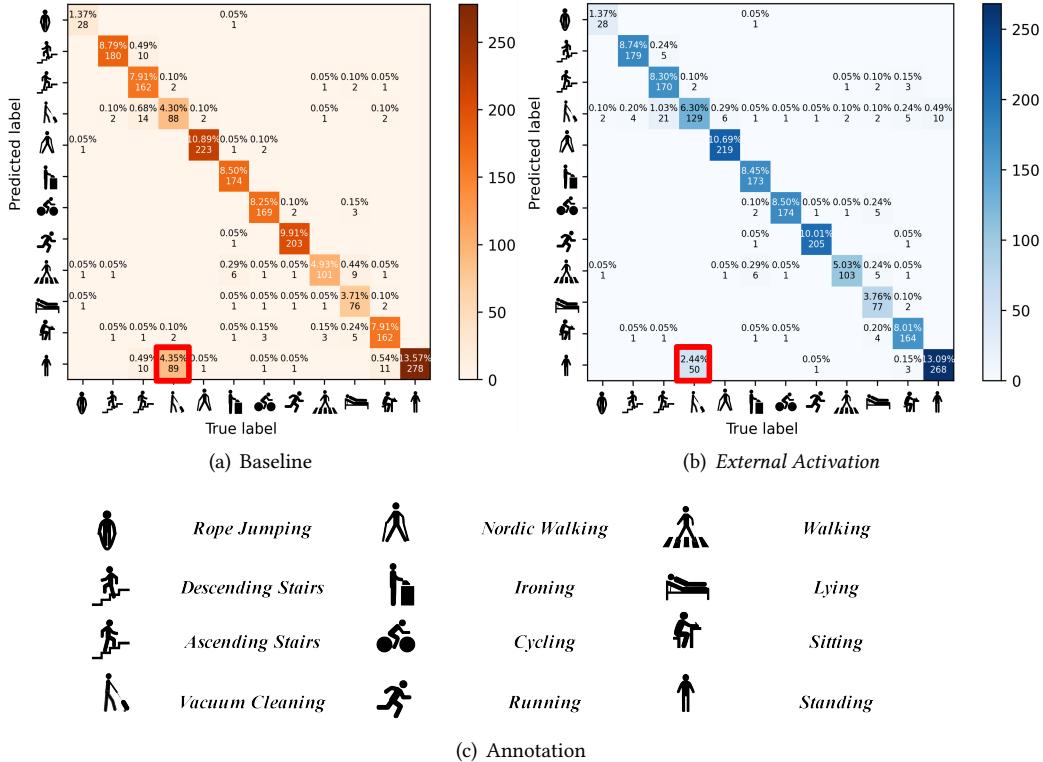


Fig. 10. Confusion Matrices of PAMAP2 dataset

filter activation does gradually reduce the number of invalid filters. On the other hand, one can clearly observe that the network aggregates more information through our activation approach as the training process goes on, which proves that filter activation does improve the potential of the network.

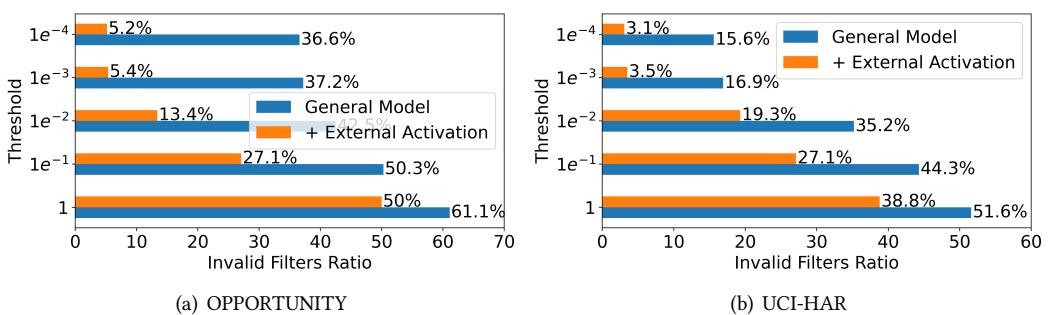


Fig. 11. The effectiveness of filter activation

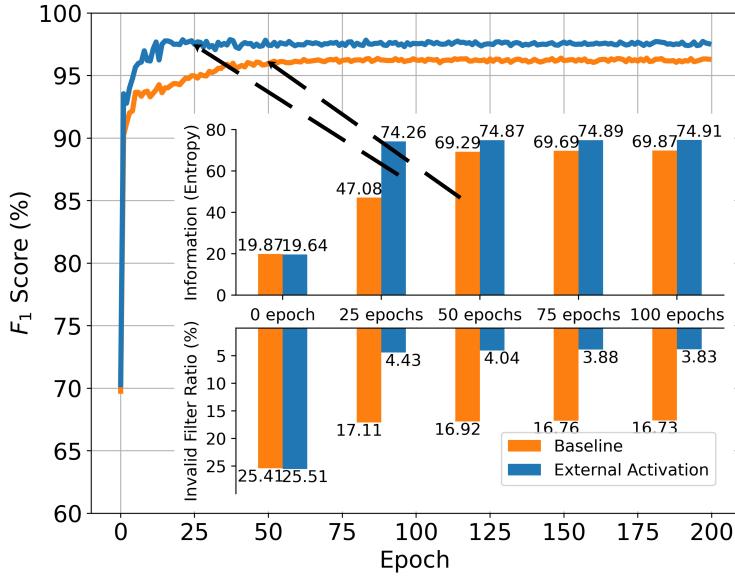


Fig. 12. The Information (Entropy) and Ratio of Invalid Filter (%; Threshold =  $1e^{-3}$ ) for the Baseline network and Ex-activated network on UCI-HAR during Training Stage

**4.2.5 Training Diversity in Filter Re-activation.** Ensemble diversity, *i.e.*, the difference among all individual learner, plays a fundamental role in filter activation, which is crucial for final performance. More variations in external activation can produce a better learning effect. We set different learning strategies and different sample orders of training set to raise ensemble diversity. Six heterogeneous networks ( $M_1$ - $M_6$ ) are used in this part. For simplicity, the initial learning rate of  $M_1$ ,  $M_2$ ,  $M_3$ ,  $M_4$ ,  $M_5$ , and  $M_6$  are  $5e^{-3}$ ,  $5e^{-4}$ ,  $2e^{-3}$ ,  $2e^{-4}$ ,  $1e^{-3}$ , and  $1e^{-4}$  which are decayed to 95% every 10, 20, 25, 30, 40, and 50 epochs, respectively. As shown in Table 6, the various learning strategies and different sample orders can significantly enhance training diversity, which leads to better classification performance. We also encourage further study about how to improve the performance of filter activation via promoting ensemble diversity.

Table 6. Test Accuracy (%) with Training Diversity

Different LS	Different Order	UCI	OPPO
✗	✗	95.91	79.33
✗	✓	96.77	79.78
✓	✗	96.87	80.23
✓	✓	<b>97.18</b>	<b>81.36</b>

**4.2.6 Prediction on real-time platform (Raspberry Pi Model 3 B Plus).** Regardless of the effectiveness, we continue to evaluate actual operation in real-time embedded systems for efficient consideration. Due to the limitation of computing resource, there are two main steps to deploy the embedded HAR systems: 1) train the network with filter re-activation on collected training dataset; 2) import this trained network into embedded system and run it to read real-time data

and output prediction. Since the *PyTorch* library can be easily installed into *Raspbian OS*, we select the *Raspberry Pi Model 3 B plus* with *ARM Cortex-A53* and *1GB SDRAM* as our test platform. We develop a *Raspbian-based* application software for real-time activity recognition, and its user interface is shown in Fig.13(a). This application is built to test our activation algorithms using UCI-HAR dataset. The timing is done after the network is loaded and starts to output a prediction. The inference time with two network structures is compared in Table 7. The results show that it takes about 116.56~127.88 ms to predict one window for the baseline model. The inference speed reaches 112.21~129.17 ms per each window for the same network structure with filter activation. The inference time curves are plotted in Fig.13(b). Both the models have almost the same inference time, and the activation does not lead to any extra computational burden.

In order to monitor real-time electricity usage and cost, we perform an experimental analysis of power consumption by plugging the *Raspberry Pi* into a *TC66* meter. As shown in Fig.13(c), this device supports *USB* communication with an external connected laptop, which allows power measurements to be programmatically sampled at a frequency of *1 Hz*. The measurement process lasts two minutes long. Table 7 reports the statistic of power consumption for each model. The power consumption of baseline network is 4.16~5.17W, which is very close to that of filter activation, i.e., 4.15~5.21W. The results verify that filter activation may compensate invalid filters to achieve an accuracy improvement at similar energy efficiency.

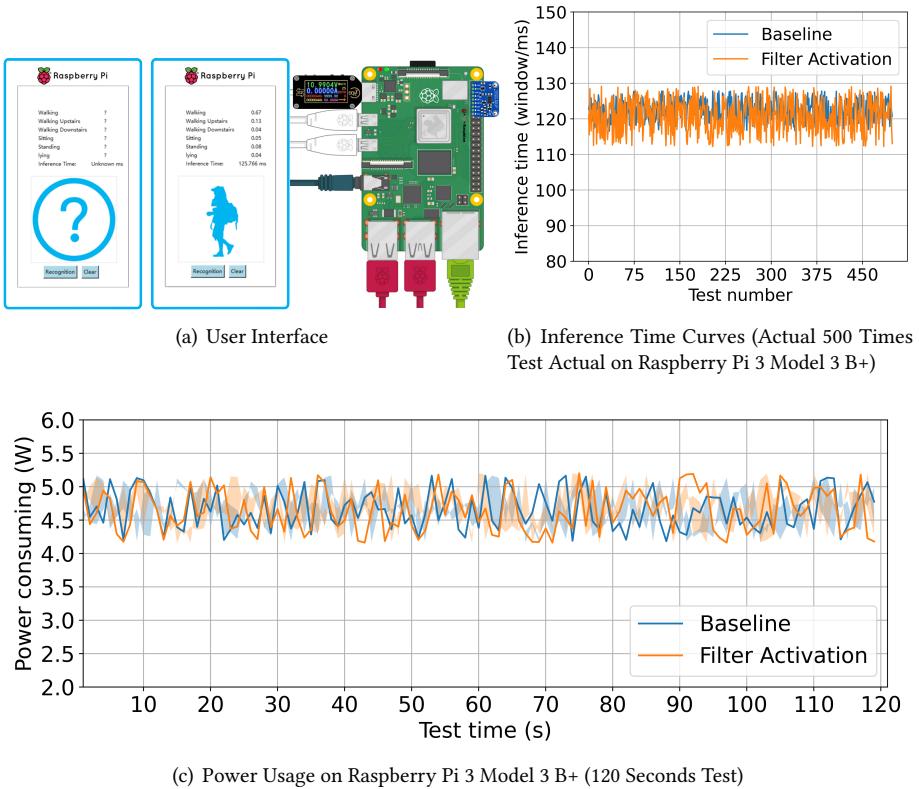


Fig. 13. Actual Testing on *Raspberry Pi Model 3 B Plus*

Table 7. Actual Testing

	Number	Baseline	Filter activation
Inference time (windows/ms)	1 <sup>st</sup> test	125.61	114.61
	2 <sup>nd</sup> test	125.94	125.31
	3 <sup>rd</sup> test	117.21	124.06
	4 <sup>th</sup> test	118.01	124.21
	5 <sup>th</sup> test	127.75	125.19
	...		
	Average	<b>122.05</b>	<b>122.61</b>
	Std	3.82	4.37
	Power Usage (W)	Average	<b>4.66</b>
		Std	0.29
		Range	4.16~5.17
			4.15~5.21

## 5 CONCLUSION

In this paper, inheriting an idea of ensemble learning, we develop a novel learning paradigm that uses filter activation to reform convolutional architecture for activity recognition. The invalid filters that have small impact on output performance potentially waste computing resources on embedded devices. In order to re-activate invalid filters, we consider the following three strategies: **1)** Gaussian noise activation; **2)** Internal activation; **3)** External activation. Instead of simply pruning invalid filters, our approach aims to re-activate such invalid filters by absorbing useful information from outside networks, which requires multiple networks to be trained in parallel, hence leading to two main challenges for effective filter activation: **1)** How to determine which filters will be used to perform activation in *CNNs*. **2)** How to share information (weights) between multiple networks. To resolve the two issues, we first propose an entropy-based measure to select these meaningful filters, and then use an adaptive weighting strategy to share weights in layer level rather than filter level for avoiding breaking layer consistency. Compared with filter pruning, our approach is simpler as it does not need to modify network architecture. It is also more efficient than ensemble learning in light of lightweight property, as inference is performed using only one network instead of multiple networks. There are two possible directions for future research: **1)** How to perform filter activation among networks using heterogeneous structures instead of the same ones; **2)** How to share weights between networks in filter level instead of layer level. We hope that this paper may motivate other researchers to develop new criteria to further improve the filter activation performance.

## 6 ACKNOWLEDGMENTS

The work was supported in part by the Natural Science Foundation of Jiangsu Province under grant BK20191371 and the National Nature Science Foundation of China under Grant 61962061, and in part by the Industry Academia Cooperation Innovation Fund Projection of Jiangsu Province under Grant BY2016001-02. Lei Zhang is the corresponding author.

## REFERENCES

- [1] Ganapati Bhat, Yigit Tuncel, Sizhe An, Hyung Gyu Lee, and Umit Y Ogras. An ultra-low energy human activity recognition accelerator for wearable health applications. *ACM Transactions on Embedded Computing Systems (TECS)*, 18(5s):1–22, 2019.

- [2] Bing Li, Wei Cui, Wei Wang, Le Zhang, Zhenghua Chen, and Min Wu. Two-stream convolution augmented transformer for human activity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 286–293, 2021.
- [3] Yi Zhang, Zheng Yang, Guidong Zhang, Chenshu Wu, and Li Zhang. Xgest: Enabling cross-label gesture recognition with rf signals. *ACM Transactions on Sensor Networks (TOSN)*, 17(4):1–23, 2021.
- [4] Ali Akbari, Jonathan Martinez, and Roozbeh Jafari. Facilitating human activity data annotation via context-aware change detection on smartwatches. *ACM Transactions on Embedded Computing Systems (TECS)*, 20(2):1–20, 2021.
- [5] Federico Concone, Giuseppe Lo Re, and Marco Morana. A fog-based application for human activity recognition using personal smart devices. *ACM Transactions on Internet Technology (TOIT)*, 19(2):1–20, 2019.
- [6] Wenbo Huang, Lei Zhang, Hao Wu, Fuhong Min, and Aiguo Song. Channel-equalization-har: A light-weight convolutional neural network for wearable sensor based human activity recognition. *IEEE Transactions on Mobile Computing*, 2022.
- [7] Xin Cheng, Lei Zhang, Yin Tang, Yue Liu, Hao Wu, and Jun He. Real-time human activity recognition using conditionally parametrized convolutions on mobile and wearable devices. *IEEE Sensors Journal*, 22(6):5889–5901, 2022.
- [8] Xinyu Li, Yuan He, Francesco Fioranelli, and Xiaojun Jing. Semisupervised human activity recognition with radar micro-doppler signatures. *IEEE Transactions on Geoscience and Remote Sensing*, 2021.
- [9] Wenbo Huang, Lei Zhang, Wenbin Gao, Fuhong Min, and Jun He. Shallow convolutional neural networks for human activity recognition using wearable sensors. *IEEE Transactions on Instrumentation and Measurement*, 70:1–11, 2021.
- [10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [11] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [12] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Liangyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.
- [13] Yonglong Tian, Guang-He Lee, Hao He, Chen-Yu Hsu, and Dina Katabi. Rf-based fall monitoring using convolutional neural networks. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3):1–24, 2018.
- [14] Ming Zeng, Haoxiang Gao, Tong Yu, Ole J Mengshoel, Helge Langseth, Ian Lane, and Xiaobing Liu. Understanding and improving recurrent networks for human activity recognition by continuous attention. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, pages 56–63, 2018.
- [15] Charissa Ann Ronao and Sung-Bae Cho. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert systems with applications*, 59:235–244, 2016.
- [16] Jianbo Yang, Minh Nhut Nguyen, Phyotho San, Xiao Li Li, and Shonali Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.
- [17] Daniele Ravi, Clarence Wong, Benny Lo, and Guang-Zhong Yang. Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In *2016 IEEE 13th international conference on wearable and implantable body sensor networks (BSN)*, pages 71–76. IEEE, 2016.
- [18] Wenchao Jiang and Zhaozheng Yin. Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1307–1310, 2015.
- [19] Eunji Kim. Interpretable and accurate convolutional neural networks for human activity recognition. *IEEE Transactions on Industrial Informatics*, 16(11):7190–7198, 2020.
- [20] Andrey Ignatov. Real-time human activity recognition from accelerometer data using convolutional neural networks. *Applied Soft Computing*, 62:915–922, 2018.
- [21] Zan Gao, Hai-Zhen Xuan, Hua Zhang, Shaohua Wan, and Kim-Kwang Raymond Choo. Adaptive fusion and category-level dictionary learning model for multiview human action recognition. *IEEE Internet of Things Journal*, 6(6):9280–9293, 2019.
- [22] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016.
- [23] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.
- [24] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.
- [25] Dong Wang, Lei Zhou, Xueni Zhang, Xiao Bai, and Jun Zhou. Exploring linear relationship in feature map subspace for convnets compression. *arXiv preprint arXiv:1803.05729*, 2018.
- [26] Jingjing Cao, Wenfeng Li, Congcong Ma, and Zhiwen Tao. Optimizing multi-sensor deployment via ensemble pruning for wearable activity recognition. *Information Fusion*, 41:68–79, 2018.
- [27] Seul-Ki Yeom, Philipp Seegerer, Sebastian Lapuschkin, Alexander Binder, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Pruning by explaining: A novel criterion for deep neural network pruning. *Pattern Recognition*,

115:107899, 2021.

- [28] Zhenghua Chen, Chaoyang Jiang, and Lihua Xie. A novel ensemble elm for human activity recognition using smartphone sensors. *IEEE Transactions on Industrial Informatics*, 15(5):2691–2699, 2018.
- [29] Huiyuan Zhuo, Xuelin Qian, Yanwei Fu, Heng Yang, and Xiangyang Xue. Scsp: Spectral clustering filter pruning with soft self-adaption manners. *arXiv preprint arXiv:1806.05320*, 2018.
- [30] Fanxu Meng, Hao Cheng, Ke Li, Zhixin Xu, Rongrong Ji, Xing Sun, and Guangming Lu. Filter grafting for deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6599–6607, 2020.
- [31] Vishvak S Murahari and Thomas Plötz. On attention models for human activity recognition. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, pages 100–103, 2018.
- [32] Kun Wang, Jun He, and Lei Zhang. Attention-based convolutional neural network for weakly labeled human activities’ recognition with wearable sensors. *IEEE Sensors Journal*, 19(17):7598–7604, 2019.
- [33] Haojie Ma, Wenzhong Li, Xiao Zhang, Songcheng Gao, and Sanglu Lu. Attnsense: Multi-level attention mechanism for multimodal human activity recognition. In *IJCAI*, pages 3109–3115, 2019.
- [34] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018.
- [35] Marcin Straczkiewicz, Peter James, and Jukka-Pekka Onnela. A systematic review of smartphone-based human activity recognition methods for health research. *Nature Partner Journal: Digital Medicine*, 4(1):1–15, 2021.
- [36] Yu Guan and Thomas Plötz. Ensembles of deep lstm learners for activity recognition using wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):1–28, 2017.
- [37] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *International workshop on ambient assisted living*, pages 216–223. Springer, 2012.
- [38] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirk, Alois Ferscha, et al. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh international conference on networked sensing systems (INSS)*, pages 233–240. IEEE, 2010.
- [39] Daniela Micucci, Marco Mobilio, and Paolo Napoletano. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, 7(10):1101, 2017.
- [40] Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th International Symposium on Wearable Computers*, pages 108–109. IEEE, 2012.
- [41] Jeffrey W Lockhart, Gary M Weiss, Jack C Xue, Shaun T Gallagher, Andrew B Grosner, and Tony T Pulickal. Design considerations for the wisdm smart phone-based sensor mining architecture. In *Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data*, pages 25–33, 2011.
- [42] Mi Zhang and Alexander A Sawchuk. Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM conference on ubiquitous computing*, pages 1036–1043, 2012.
- [43] Zanobya N Khan and Jamil Ahmad. Attention induced multi-head convolutional neural network for human activity recognition. *Applied Soft Computing*, 110:107671, 2021.
- [44] Qi Teng, Kun Wang, Lei Zhang, and Jun He. The layer-wise training convolutional neural networks using local loss for sensor-based human activity recognition. *IEEE Sensors Journal*, 20(13):7265–7274, 2020.
- [45] Alireza Abedin, Mahsa Ehsanpour, Qinfeng Shi, Hamid Rezatofighi, and Damith C Ranasinghe. Attend and discriminate: beyond the state-of-the-art for human activity recognition using wearable sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(1):1–22, 2021.
- [46] Kun Wang, Jun He, and Lei Zhang. Sequential weakly labeled multiactivity localization and recognition on wearable sensors using recurrent attention networks. *IEEE Transactions on Human-Machine Systems*, 2021.
- [47] Hangwei Qian, Sinno Jialin Pan, and Chunyan Miao. Latent independent excitation for generalizable sensor-based cross-person activity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11921–11929. AAAI, 2021.
- [48] Mohammed AA Al-qaness, Abdelghani Dahou, Mohamed Abd Elaziz, and AM Helmi. Multi-resatt: Multilevel residual network with attention for human activity recognition using wearable sensors. *IEEE Transactions on Industrial Informatics*, 2022.
- [49] Songpengcheng Xia, Lei Chu, Ling Pei, Zixuan Zhang, Wenxian Yu, and Robert C Qiu. Learning disentangled representation for mixed-reality human activity recognition with a single imu sensor. *IEEE Transactions on Instrumentation and Measurement*, 70:1–14, 2021.
- [50] Wesllen Sousa Lima, Hendrio LS Braga, and Eduardo JP Souto. Nohar-novelty discrete data stream for human activity recognition based on smartphones with inertial sensors. *Expert Systems with Applications*, 166:114093, 2021.
- [51] Luay Alawneh, Belal Mohsen, Mohammad Al-Zinati, Ahmed Shatnawi, and Mahmoud Al-Ayyoub. A comparison of unidirectional and bidirectional lstm networks for human activity recognition. In *2020 IEEE International Conference*

- on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 1–6. IEEE, 2020.
- [52] Min Zhang. Gait activity authentication using lstm neural networks with smartphone sensors. In *2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, pages 456–461. IEEE, 2019.
  - [53] Xiaojie Sun, Hongji Xu, Zheng Dong, Leixin Shi, Qiang Liu, Juan Li, Tiankuo Li, Shidi Fan, and Yuhao Wang. Capsganet: Deep neural network based on capsule and gru for human activity recognition. *IEEE Systems Journal*, 2022.
  - [54] Chenglin Li, Di Niu, Bei Jiang, Xiao Zuo, and Jianming Yang. Meta-har: Federated representation learning for human activity recognition. In *Proceedings of the International World Wide Web Conferences (WWW) 2021*, pages 912–922. ACM, 2021.
  - [55] Haixia Bi, Miquel Perello-Nieto, Raul Santos-Rodriguez, and Peter Flach. Human activity recognition based on dynamic active learning. *IEEE Journal of Biomedical and Health Informatics*, 25(4):922–934, 2020.
  - [56] Satya P Singh, Madan Kumar Sharma, Aimé Lay-Ekuakille, Deepak Gangwar, and Sukrit Gupta. Deep convlstm with self-attention for human activity decoding using wearable sensors. *IEEE Sensors Journal*, 21(6):8575–8582, 2020.
  - [57] Shohreh Deldari, Daniel V Smith, Hao Xue, and Flora D Salim. Time series change point detection with self-supervised contrastive predictive coding. In *Proceedings of the Web Conference 2021*, pages 3124–3135. ACM, 2021.