This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3132088, IEEE Transactions on Instrumentation and Measurement

Deep Convolutional Networks with Tunable Speed-Accuracy Trade-off for Human Activity Recognition Using Wearables

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

***Abstract*—Activity recognition plays a critical role in various applications such as medical monitoring and rehabilitation. Deep learning has recently made great development in wearable based human activity recognition (HAR) area. However, real HAR applications should be adaptive and flexible to available computational budget. So far, this problem has rarely been explored. In contrast to existing deep HAR researches focusing on static networks, this paper aims to investigate adaptive networks, which can adjust their structure conditioned on available computing resource to trade off between accuracy and speed. We for the first time present an adaptive convolutional neural network by dynamically modifying network width. Specifically, first, instead of normal convolution, the network is stacked by lower-triangular convolutional layers in order to remove the impact of activation statistics caused by varying widths. Second, instead of fixed sampling, we perform random sampling over width, which can provide smooth control for trade-off between accuracy and speed. As a consequence, the networks with different widths are simultaneously trained as subnetworks by accumulating their losses during each iteration. On multiple HAR datasets such as UCI-HAR, PAMAP2 and OPPORTUNITY, extensive experiments verify that the proposed approach can consistently provide further improved efficiency on top of state-of-the-art CNNs for HAR. Finally, evaluations are conducted on a Raspberry Pi platform to demonstrate its usefulness and practicality.**

***Index Terms*—Sensor, convolutional neural networks, activity recognition, deep learning, wearable device**

I. INTRODUCTION

DURING the past decade, there is a rapid development in Internet of Things (IoT) and sensing technology, which enables various motion sensors such as accelerometer and gyroscope embedded in smartphones or other wearable devices to record physical activities for automatic recognition task. Due to small size and low cost, sensor based human activity recognition (HAR)[1], [2], [3] has played an indispensable role in human-computer interaction and ubiquitous computing, which has gained a lot of attention in various application scenarios such as wellbeing, smart home, sports monitoring, medical monitoring and rehabilitation[4], [5]. In essence, HAR can be treated as a typical pattern recognition problem[6], [7]. Traditional recognition algorithms such as decision tree, support vector machine and naive Bayes have been devoted to the design of shallow handcrafted features[8], which heavily rely on specific domain knowledge or human experience. However, such handcrafted features such as the mean, variance, frequency and amplitude of Fourier transform are hard to infer complex human activities, which leads to a lower chance to build a successful HAR system.

Deep learning, especially convolutional neural network (CNN)[9] has recently provided an alternative to overcome above limitations. In deep learning, the feature extraction and inference model often can be performed simultaneously. The high-level features can be extracted automatically through stacking deeper layers rather than being manually designed, which makes it very suitable for inferring complex activities. Although CNN has significantly improved the accuracy of state-of-the-art HAR algorithms[10], [11], [12], the low-latency or realtime feedback may be very critical. Besides accuracy, computational complexity is another key factor to be considered. Actually, highly miniaturized wearable devices often have a very limited computational budget. Real HAR applications typically pursue best accuracy under a resource-constrained platform, where an accuracy/speed trade-off should be preferably considered.

On the other hand, in order to deploy HAR applications under realistic conditions, one has to maintain model flexibility to adapt to varying computational budgets (e.g., varying hardware platforms)[13], [14]. Thus, fast inference alone is not always sufficient. For example, as far as we know, there are over 20 brands (e.g., iPhone, Samsung and Google) of phones, which have a limited yet variable computing power. It will lead to drastically different inference times, even for the same model. For satisfactory inference speed, low-end phones have to run smaller models with lower accuracy due to limited computing power, while high-end phones can run larger models to produce higher accuracy. As a result, the inference time constraint has to be conditionally dependent for specific hardware platforms. Even for the same phone, the computing power could potentially vary due to resource consumption caused by other running apps. On the

The work was supported in part by the National Science Foundation of China under Grant 61962061 and the Industry-Academia Cooperation Innovation Fund Projection of Jiangsu Province under Grant BY2016001-02, and in part by the Natural Science Foundation of Jiangsu Province under grant BK20191371. *(Corresponding author: Lei Zhang.(e-mail: leizhang@njnu.edu.cn))*

Xing Wang, Lei Zhang, Wenbo Huang and Shuoyuan Wang are with School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing, 210023, China.

Hao Wu is with School of Information Science and Engineering, Yunnan University, Yunnan, 650500, China

Jun He is with School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing, 210044, China.

Aiguo Song is with School of Instrument Science and Engineering, Southeast University, Nanjing, 210096, China.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3132088, IEEE Transactions on Instrumentation and Measurement

whole, real HAR algorithms must be not only simply fast, but also flexible to varying computational budgets, which have been rarely explored in HAR area.

Our core research motivative is to design an adaptive deep network in HAR scenario, which can adapt to varying computational budgets. A common idea is to modify width (the number of active channels), instead of depth (the number of layers). Specially, the simplest strategy is to train N networks at different widths, which can dynamically perform switch between them for trade-off between accuracy and speed. However, there are obvious drawbacks for the N-Width strategy. Due to varying widths, the activation statistics in each sub-network is variable. Thus, N networks have to be separately trained. If N is small, one only can provide coarse-grained control for accuracy-speed trade-off. Instead, if N is increased to a fine-grained level, it will lead to a great increase in training burden. As a consequence, it needs to store multiple offline trained versions, which will inevitably lead to larger memory footprint. Moreover, it takes necessary communication cost (e.g., time) to download new models for switch.

In this paper, we propose an adaptive deep network in HAR scenario, which can maintain model flexibility to adapt to varying computational budgets at runtime, without the need of downloading new models for switch on wearable devices. We address above challenges from two technical aspects. First, instead of N fixed sampling, we perform random sampling over width, where the networks at arbitrary width can be simultaneously trained as subnetworks by accumulating their losses during each iteration. In other words, we only need to train a single network executable at any width, which can provide a smooth control over accuracy/speed trade-off. Second, to avoid potential performance degradation caused by varying activation statistics at any width, the network is built using lower-triangular convolutional layers rather than normal convolution. Evaluations are performed on multiple public HAR datasets, namely UCI-HAR, PAMAP2, UniMib-SHAR, OPPORTUNITY and WISDM. In contrast to static models, the proposed model can provide a smooth control over accuracy/speed trade-off, which can dynamically adapt deep HAR to available computing resource. The contributions of this paper are three-fold:

First, we for the first time propose an adaptive deep network executable at any width for HAR, which can provide a smooth control over accuracy/speed trade-off via random sampling approach.

Second, when switching at arbitrary width, we avoid potential performance degradation caused by varying activation statistics via replacing normal convolution with lower-triangular convolution.

Third, without need of downloading new models, we perform actual evaluations at test time on a Raspberry Pi platform, which verify the practibility and usefulness of the proposed model.

The rest of this paper is structured as follows. Section II reviews the related works. Section III presents the structure of adjustable CNN in HAR scenario. Section IV details experimental setup, performance comparison and analysis, which indicates the advantage of our proposed model. Finally, conclusions are made in Section V.

II. Related works

Recent developments in deep learning have significantly improved the accuracy of state-of-the-art HAR algorithms, which can greatly alleviate the burden of handcrafted feature designing procedures. In HAR scenario, Zeng et al.[9] at the earliest time exploited CNN to automatically extract the local dependency and scale invariant features from raw acceleration time series. Yang et al.[15] presented a novel CNN constructed by multiple iterations of convolutional and pooling layers, where convolution and pooling operation are combined for feature extraction of HAR. Hammerla et al.[16] evaluated various deep, convolutional and recurrent approaches on several benchmark datasets, which shows how they surpass traditional shallow machine algorithms for a large variety of HAR tasks. Huang et al.[17] introduced a shallow CNN that uses cross-channel communication in HAR scenario, where graph neural network is used to realize a comprehensive interaction between different channels for capturing discriminative features of raw sensor signals. In order to capture temporal dynamics for HAR, Ordóñez et al.[18] proposed a novel network architecture called DeepConvLSTM consisting of convolutional and recurrent units, which significantly outperforms CNN alone. On the whole, those above network architectures are often static. However, most wearable devices only have a limited, yet dynamic computing power due to their varying hardware platforms. Thus, HAR algorithms should be adaptive or flexible to available computational budgets. It deserves deeper research into adaptive network architectures for HAR, which so far has been rarely explored.

In another line of research, deep learning has made major breakthroughs in machine vision, in which the trade-off between speed and accuracy is at the forefront of this research. Many previous works have been devoted to designing lightweight networks. For example, Howard et al.[19] and Sandler et al.[20] presented a series of lightweight networks called MobileNet, which can adaptively scale model size for different vision tasks by adjusting width and resolution multipliers. Zhang et al.[21] and Ma et al.[22] proposed a family of small networks called ShuffleNet, which can effectively decrease computational overhead by exploiting channel shuffle and pointwise group convolution. There are also other model compression techniques known as weight or channel pruning, decomposition, and knowledge distillation[23]. Despite their success, these above methods make the trade-off decision during network design or training stage. Instead, we aim to trade off accuracy and speed at inference time. Several prior approaches have been proposed for inference time control by modifying network depth. One mainstream idea is to perform early-exits or early-stopping inference according to available computing resource[24]. In order to reduce computation, other research efforts focus on layer skipping or dropping[25], [26]. These depth-modulation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3132088, IEEE Transactions on Instrumentation and Measurement

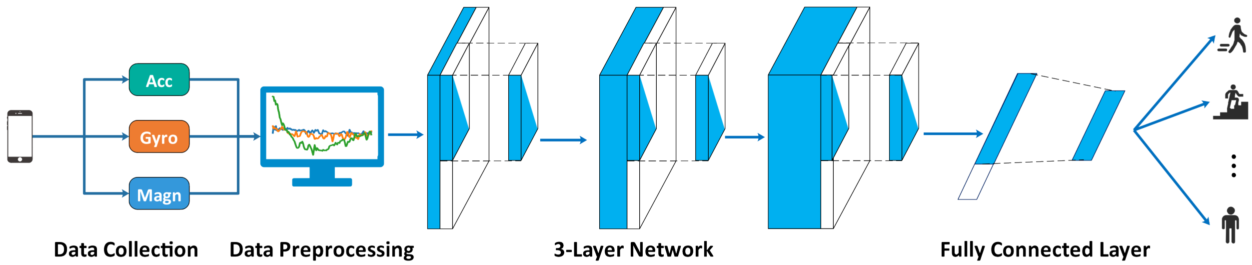

Fig. 1. The overview of tunable-width HAR model using lower-triangular convolution

techniques are orthogonal to resource-constrained control.

Recently, the idea of width-modulation has been exploited to control the trade-off between accuracy and speed. Using pruning technique, Kim et al.[27] trained one network in terms of a specific number of sparsity ratios, which accordingly provides a fixed set of internal networks. All these networks are jointly optimized to form a final N-in-1 nested network, which enables multiple hierarchical classification tasks through a single network. Similarly, Yu et al.[28] trained one network with multiple versions of a specific number of predefined N-widths, which can be switched during inference time. As a result, they only can provide coarse-fined control due to a limited number of switchable networks, where multiple copies of the trained model need to be stored. From the viewpoint of width, Lu et al. first investigated how it affects the performance of deep networks. Moreover, they presented a universal theorem for the width-bounded ReLU networks[29]. Yu et al.[30] and Huang et al.[31] exploited two-step knowledge distilling technique to train an Universally Slimmable Network. However, most width-modulation techniques have been devoted to various vision applications, which has rarely been exploited in ubiquitous HAR scenario. Actually, there have always been increasing demands to deploy HAR system on resource-constrained wearable devices. While many prior researches have focused on designing a static deep network for HAR, it is very infeasible to deploy a single neural network across different wearable devices with variable computing resource. Because a new type of device usually requires a new network architecture, one has to build and train a new model from scratch. In this paper, we for the first time tackle this challenge in HAR scenario by designing an adaptive network that can provide assured performance on available computing resource.

III. Model

As indicated above, our research motivative is to train a single network for HAR task, which can run at arbitrary width. Without loss of generality, the basic form of feature aggregation within a specific convolution layer can be formulated as[23], [32]:

$$y = \sum_{i=1}^{k} \omega_i x_i \tag{1}$$

where n is the number of channels (width)x_i, and w_i, $i \in \{1, 2, \cdots, k\}$ correspond to the input feature vector and learnable coefficient respectively, and y is an output. The network width plays an important part in trading off accuracy and speed: the larger number of channels often provides better accuracy, but sacrifices inference speed. In principle, the performance of wider networks should be no worse than that of its slim version, which has been proved by the theory of channel-wise residual learning[31], [32], [33]:

$$0 \leq \delta_{k+1} \leq \delta_k \leq \delta_{k_0},\ \delta_k = \left| y^n - y^k \right|, \delta_{k+1} = \left| y^n - y^{k+1} \right| \tag{2}$$

where y^k and y^{k+1} is the sum of the first k and $k+1$ channels respectively, i.e., $y^k = \sum_{i=1}^{k} \omega_i x_i$ and $y^{k+1} = \sum_{i=1}^{k+1} \omega_i x_i$. k_0 denotes the minimum width. The above inequality 2 indicates that there is an upper bound δ_{k_0} for residual errors between fully aggregated features y^n and partially aggregated features y^k, which continuously decreases as the network width k increases. As the upper bound exists, a single network could run at arbitrary width by varying k from k_0 to k_n. To train a single network with any width, an intuitive idea is to accumulate losses randomly sampled from all sub-networks at arbitrary widths. In order to access all widths, fixing lower bound (e.g., smallest width 0.125x) and upper bound (e.g., largest width 1.0x), we randomly sample $n-2$ widths over the open interval (0.125 ,1.0). At each training iteration, besides the lower and upper bound, the network is trained with $n-2$ randomly sampled widths by performing gradient back-propagation from accumulated losses with all widths, where ground truth is used as training label for HAR.

Actually, the random sampling method can be seen as a kind of knowledge distillation, which transfers knowledge learned from full-width network to subnetworks at smaller widths in each iteration. Specifically, we first train a large network at full width, then its learned knowledge is transferred to a small network, which is then trained with predicted soft labels. Using this rule, we can train the tunable-width network at largest width, smallest width and other randomly sampled widths all together during each training iteration. That is to say, this training idea is naturally supported by knowledge distillation: the ground truth is directly used to train the network at largest width, while the predicted label by the network at the largest width is used as the training label for the networks at other widths. The experimental results verifies that the classification accuracies do not decay rapidly as the network width decreases. This training scheme can implement well without extra computational and memory overhead[23], [31].

On the other hand, if the network is switched in a multi-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3132088, IEEE Transactions on Instrumentation and Measurement

width scenario, the issue of varying activation statistics at different widths has to be considered. For example, if k=1, the Eq.1 can be explicitly written as[34]:

$$[y] = [\omega_{11}][x_1] = [\omega_{11}x_1] \tag{3}$$

and if k=2 and k=3, it is:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \omega_{11} & \omega_{12} \\ \omega_{21} & \omega_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \omega_{11}x_1 & \omega_{12}x_2 \\ \omega_{21}x_1 & \omega_{22}x_2 \end{bmatrix} \tag{4}$$

$$\begin{aligned} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} &= \begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{21} & \omega_{22} & \omega_{23} \\ \omega_{31} & \omega_{32} & \omega_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\ &= \begin{bmatrix} \omega_{11}x_1 + \omega_{12}x_2 + \omega_{13}x_3 \\ \omega_{21}x_1 + \omega_{22}x_2 + \omega_{23}x_3 \\ \omega_{31}x_1 + \omega_{32}x_2 + \omega_{33}x_3 \end{bmatrix} \end{aligned} \tag{5}$$

For simplicity and without loss generality, y_1 can be formulated as:

$$y_1 = \begin{cases} \omega_{11}x_1 & if\ k=1 \\ \omega_{11}x_1 + \omega_{12}x_2 & if\ k=2 \\ \omega_{11}x_1 + \omega_{12}x_2 + \omega_{13}x_3 & if\ k=3 \end{cases} \tag{6}$$

The expected value of y_1 may vary at different widths:

$$\begin{aligned} \mathbb{E}\left[y_1^{\{1\}}\right] &= \mathbb{E}[\omega_{11}x_1] = \mu_1^{\{1\}} & k=1 \\ \mathbb{E}\left[y_1^{\{2\}}\right] &= \mathbb{E}[\omega_{11}x_1 + \omega_{12}x_2] \\ &= \mathbb{E}[\omega_{11}x_1] + \mathbb{E}[\omega_{12}x_2] \\ &= \mu_1^{\{1\}} + \mathbb{E}[\omega_{12}x_2] & k=2 \\ \mathbb{E}\left[y_1^{\{3\}}\right] &= \mathbb{E}[\omega_{11}x_1 + \omega_{12}x_2 + \omega_{13}x_3] \\ &= \mathbb{E}[\omega_{11}x_1] + \mathbb{E}[\omega_{12}x_2] + \mathbb{E}[\omega_{13}x_3] \\ &= \mu_1^{\{1\}} + \mathbb{E}[\omega_{12}x_2] + \mathbb{E}[\omega_{13}x_3] & k=3 \end{aligned} \tag{7}$$

Clearly, three cases (k=1, k=2 and k=3) could not follow the same distribution. In other words, the switch between multi-width will lead to varying statistics of y_1 in batch normalization[34], which may deteriorate classification performance. In order to guarantee the same distribution in two-width or three-width case, one has to ensure that $\mathbb{E}[\omega_{12}x_2]$ = 0 or $\mathbb{E}[\omega_{12}x_2] = \mathbb{E}[\omega_{13}x_3]$=0. In a similar way, the expected value of y_2 may vary at different widths:

$$\begin{aligned} \mathbb{E}\left[y_2^{\{2\}}\right] &= \mathbb{E}[\omega_{21}x_1 + \omega_{22}x_2] \\ &= \mathbb{E}[\omega_{21}x_1] + \mathbb{E}[\omega_{22}x_2] \\ &= \mathbb{E}[\omega_{21}x_1] + \mu_2^{\{2\}} & k=2 \\ \mathbb{E}\left[y_2^{\{3\}}\right] &= \mathbb{E}[\omega_{21}x_1 + \omega_{22}x_2 + +\omega_{23}x_3] \\ &= \mathbb{E}[\omega_{21}x_1] + \mathbb{E}[\omega_{22}x_2] + \mathbb{E}[\omega_{23}x_3] \\ &= \mathbb{E}[\omega_{21}x_1] + \mu_2^{\{2\}} + \mathbb{E}[\omega_{23}x_3] & k=3 \end{aligned} \tag{8}$$

To ensure $\mathbb{E}\left[y_2^{\{2\}}\right]$=$\mathbb{E}\left[y_2^{\{3\}}\right]$, one has to set $\mathbb{E}[\omega_{23}x_3]$=0. As a result, the weights in convolutional layers can be formulated as:

$$\begin{bmatrix} \omega_{11} & 0 & 0 \\ \omega_{21} & \omega_{22} & 0 \\ \omega_{31} & \omega_{32} & \omega_{33} \end{bmatrix} \tag{9}$$

Without loss generality, when above two-width or three-width case is extended to multi-width case, one feasible solution is to constrain the weight matrices in convolutional layers to be lower-triangular[32], [34], [35], which can address this problem well. The overview of the design is illustrated in Fig.1. Thus, we aim to design a convolutional network consisting of triangular layers (Fig.2), which can run at arbitrary width to perform activity recognition tasks. The pseudo code in Algorithm 1 summarizes the overall procedure of model training.

Algorithm 1 Convolutional Network with Tunable Speed-Accuracy Trade-off

Input: **A**, Convolutional Network
Input: E_{iters}, number of training iterations
Input: N, number of randomly sampled width
Input: $L_{\max}$, sampled width upper bound, $L_{\min}$, sampled width lower bound
Input: x, training data processed by sliding window, y^*, labels processed by sliding window

```
Initialize network A
for i=1 to E_iters do
    Load x, y*
    Sample N-2 widths between the interval [L_min, L_max]
    S = {L_min, a_1..., a_{N-2}, L_max}
    Clearing gradients
    for j in S do
        Set network's width as j
        y = A(x)
        Loss = criterion(y*, y)
        Loss backward
    end for
    Network optimize
end for
```

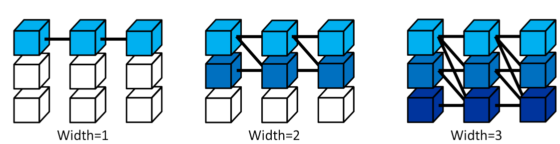

Fig. 2. Channels connection of lower-triangular convolution at different widths

IV. Experiment

In this section, three types of experiments are conducted. In part one, we aim to analyze how HAR accuracy is affected as network width is varied. In order to do this, we measure the effect of triangular convolutional layer at different widths. In part two, we perform further ablation studies with regard to the statistics of batch normalization[34]. We also analyze

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3132088, IEEE Transactions on Instrumentation and Measurement

TABLE I
SIMPLE DESCRIPTION OF DATASET

Attribute \ Dataset	UCI-HAR	WISDM	PAMAP2	UniMib-SHAR	OPPORTUNITY
Categories	6	6	12	17	18
Windows Size	128×9	200×3	171×40	151×3	40×113
Overlap rate	50%	95%	50%	-	50%
Sampling rate	50Hz	20Hz	100Hz	50Hz	30Hz

TABLE II
SIMPLE DESCRIPTION OF MODEL'S STRUCTURE

Dataset \ Layers	Input Size	Layer1		Layer2		Layer3	
		Conv1	BN+ReLU	Conv2	BN+ReLU+Maxpool	Conv3	BN+ReLU+Maxpool
UCI-HAR	(none,1,128,9)	(none,128,42,9)	(none,128,42,9)	(none,256,13,9)	(none,256,14,9)	(none,512,4,9)	(none,512,5,9)
WISDM	(none,1,200,3)	(none,128,100,2)	(none,128,100,2)	(none,256,50,1)	(none,256,51,1)	(none,512,26,1)	(none,512,27,1)
PAMAP2	(none,1,171,40)	(none,128,56,40)	(none,128,56,40)	(none,256,18,40)	(none,256,19,40)	(none,512,6,40)	(none,512,7,40)
UniMib-SHAR	(none,1,151,3)	(none,128,74,3)	(none,128,74,3)	(none,256,36,3)	(none,256,37,3)	(none,512,17,3)	(none,512,18,3)
OPPORTUNITY	(none,1,40,113)	(none,128,20,57)	(none,128,20,57)	(none,256,10,29)	(none,256,11,29)	(none,512,6,15)	(none,512,7,15)

TABLE III
SIMPLE DESCRIPTION OF NEURAL NETWORK PARAMETER

Layers \ Dataset	UCI-HAR			WISDM			PAMAP2			UniMib-SHAR			OPPORTUNITY		
	conv	padding	stride	conv	padding	stride	conv	padding	stride	conv	padding	stride	conv	padding	stride
layer1	(6,1)	(1,0)	(3,1)	(3,3)	(1,1)	(2,2)	(6,1)	(1,0)	(3,1)	(6,1)	(1,0)	(2,1)	(3,3)	(1,1)	(2,2)
layer2	(6,1)	(1,0)	(3,1)	(3,3)	(1,1)	(2,2)	(6,1)	(1,0)	(3,1)	(6,1)	(1,0)	(2,1)	(3,3)	(1,1)	(2,2)
layer3	(6,1)	(1,0)	(3,1)	(3,3)	(1,1)	(2,2)	(6,1)	(1,0)	(3,1)	(6,1)	(1,0)	(2,1)	(3,3)	(1,1)	(2,2)

TABLE IV
TEST ACCURACY(%)

Width Mutiplier \ Dataset	UCI-HAR	WISDM	PAMAP2	UniMib-SHAR	OPPORTUNITY
baseline	96.28	97.32	91.35	74.61	93.48
1.0	96.63	98.04	92.22	75.21	93.56
0.875	96.44	98.27	92.51	75.25	93.59
0.75	96.64	98.13	92.51	75.33	93.49
0.625	96.38	98.04	91.93	75.33	93.45
0.5	96.54	97.86	92.15	75.03	93.25
0.375	96.33	97.45	91.78	74.17	93.12
0.25	96.49	97.40	91.49	74.17	92.82
0.125	95.32	96.36	91.49	73.32	92.68
Other Researches	95.75[36] 95.18[40] 96.37[37]	93.32[37] 96.90[41] 97.50[39]	89.30[38] 89.96[9] 91.40[41]	74.46[39] 74.66[42] 74.97[44]	91.50[18] 89.15[43] 92.7[16]

TABLE V
MAC OF DIFFERENT DATASET

Width Mutiplier \ Dataset	UCI-HAR	WISDM	PAMAP2	UniMib-SHAR	OPPORTUNITY
0.125	0.855M	0.748M	5.591M	1.038M	3.280M
0.25	3.314M	2.916M	21.504M	3.993M	12.537M
0.375	7.376M	6.504M	47.738M	8.866M	27.812M
0.5	13.042M	11.510M	84.295M	15.655M	49.063M
0.625	20.312M	17.936M	131.174M	24.361M	76.305M
0.75	29.185M	25.781M	188.375M	34.985M	109.537M
0.875	39.662M	35.046M	255.897M	47.525M	148.760M
1	51.743M	45.729M	333.742M	61.982M	193.973M

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3132088, IEEE Transactions on Instrumentation and Measurement

the influence of several key factors such as the number of randomly sampled width (N), width lower bound and width sampling rule. Finally, the actual operation of adaptive width network is evaluated on a resource-constrained Raspberry Pi platform.

A. Datasets

1) UCI-HAR[45]: Wearing a Samsung Galaxy S2 smartphone on the waist, a group of 30 volunteers between 19 and 48 years old joined in the data collection process. Each volunteer was instructed to perform a set of predefined activities consisting of Standing, Lying, Sitting, Walking, Walking upstairs and Walking downstairs. At a sampling frequency of 50Hz, the triaxial acceleration and angular velocity signals are recorded by the embedded accelerometer and gyroscope in the smartphone.

2) WISDM[46]: The WISDM research team members from Fordham University created the dataset. Wearing an Android based smartphone in their trouser pants pocket, 29 subjects were supervised to perform six kinds of activities including Walking, Jogging, Ascending stairs, Descending stairs, Sitting, and Standing. The accelerometer signals are collected every 50ms, i.e., 20 samples/second.

3) PAMAP2[47]: The dataset is composed of sensor recordings collected from 9 volunteers, who were instructed to perform 18 physical activities, consisting of 12 specific activities (Walking, Cycling, Rope jumping, etc.) and a few optional activities (Watching TV, Playing soccer, Car driving, etc.). Each volunteer wore three Colibri wireless inertial measurement units (IMU), which were placed over the dominants chest, hand and ankle respectively. The sampling rate of 100Hz is down-sampled into 33.3Hz for further analysis. A heart rate monitor with sampling rate 9Hz is used to estimate sport intensity.

4) UniMib-SHAR[48]: The dataset is composed of 11771 samples collected from 30 subjects whose ages range between 18 and 60 years. Wearing a Samsung Galaxy Nexus I9250 phone with an accelerometer BMA220 in the front trouser pocket, each subject performed 8 types of activities of daily living such as Walking, Standing, Sitting and 9 types of falls such as Falling leftward, Falling rightward, Falling back and Syncope. All samples were collected at a frequency of 50 Hz.

5) OPPORTUNITY[49]: The dataset was collected from various hybrid sensor modalities such as accelerometer, gyroscope, magnetometer, and video camera. In a breakfast scenario, 12 volunteers are instructed to perform a specific set of daily morning activities such as Preparing and Drinking coffee, Preparing and Eating sandwich, and Cleaning table. In this paper, we utilize the subset from the OPPORTUNITY challenge consisting of unsegmented sensor recordings from 4 subjects. Data is collected at a sampling rate of 30Hz by the body-worn sensors placed on 12 different locations of human body.

B. Training details

We perform experiments on the five benchmark HAR datasets: UCI-HAR, WISDM, PAMAP2, UniMib-SHAR, and OPPORTUNITY. Data preprocessing is a crucial step in this activity recognition process. Sensor signals are often involved in various human activities in different contexts, which have been recorded via hybrid sensor modalities. The heterogeneous sensor values have to be normalized into zero mean and unit variance via subtracting the mean and dividing by the standard deviation. Traditional machine learning algorithms could not directly handle raw sensor input, which need to be first segmented via a fixed-length sliding window. To be specific, fixing an overlap rate, one can slide window over continuous sensor reading to produce continuous samples, where each window may be assigned a specific activity label. As a result, data are divided into windows of a fixed size and with no inter-window gaps, and an overlap between adjoining windows is tolerated to preserve the continuity of sensor signals. Although sliding window has been normally used to perform segmentation, there is still no clear consensus on how to select an optimal window size. Actually, the window size has an important effect on activity recognition performance[50]. According to our intuition, reducing the window length will be more beneficial for a faster activity recognition, as well as reduced computational cost and energy consumption. Instead, increasing window length are usually used for the recognition of complex activities that last a longer time. Actually, most designs normally rely on figures used in previous works, but with no strict researches that support them. For fair comparisons, we still select the same values used in previous HAR literatures. During data preprocessing stage, several important properties such as window size, overlap rate and sampling rate are summarized in Table I. All datasets are divided into three parts consisting of a training set (70%), a validation set (10%) and a test set (20%).

All detailed training parameters are provided for reproducibility. On each dataset, the 3-layer tunable width network equipped with triangular convolutional layers is compared with normal baseline CNN. The detailed descriptions of network architectures are illustrated in Table II. Each convolutional (Conv) layer is followed by a batch-normalization (BN) layer andăactivation function (ReLU). Batch normalization plays a role in standardizing the inputs to a layer for each mini-batch, which has an effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep networks. In particular, three lower-triangular convolutional layers are stacked in order to remove the impact of activation statistics caused by varying widths. The max-pooling operation pools a feature map by taking its maximum values. Two max-pooling operations are inserted after the second and third convolutional layers. In the case of activity recognitionătask, the final output may be further fed into a Softmax layer to produce a class

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3132088, IEEE Transactions on Instrumentation and Measurement

probability distribution. We introduce the detailed parameter settings of low-triangular convolution, such as kernel size, step length and padding size in Table III. The source code

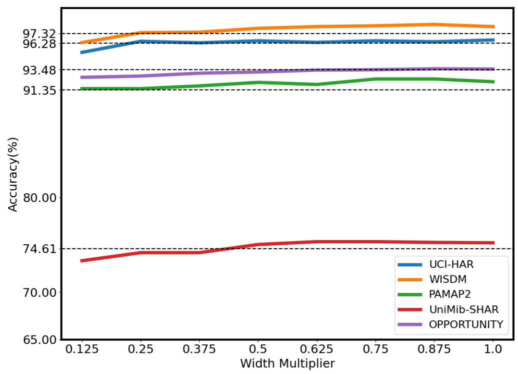

Fig. 3. Accuracy curves at varying widths

will be available at the website https://github.com/Chauncey-Wang/Tunable-Speed-Accuracy-Trade-off-for-HAR. During each training iteration, all networks are trained at eight random sampling widths. That is to say, we train all eight widths as sub-networks simultaneously by accumulated their losses. We use the random sampling strategy as indicated above in Section III. All networks are trained for 200 epochs by using an Adam optimizer with batch size of 512. The initial learning rate is set to 5e-4, which is decayed by a factor of 0.1 every 40 epochs. All experiments are implemented by deep learning library PyTorch on a workstation equipped with Intel I7-6850K CPU, NVIDIA GeForce RTX3090 GPU and 64G RAM. Main experiments results are presented in Table IV at eight width factors of 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1.0. It can be seen that our tunable width networks

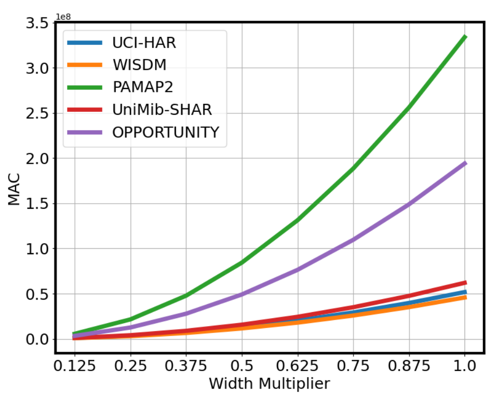

Fig. 4. MAC of different datasets at varying widths

perform as well or better than its normal CNN counterparts. In terms of accuracy, our models outperform their normal baselines by 0.35%, 0.72%, 0.87%, 0.60% and 0.08% respectively at the same width of 1. In an extreme case where the network width is reduced to 0.125x, there are only 0.96%, 0.96%, 1.29%, 0.80% and 0.08% performance degradation when compared with 1.0x(Fig.3). As an acknowledged

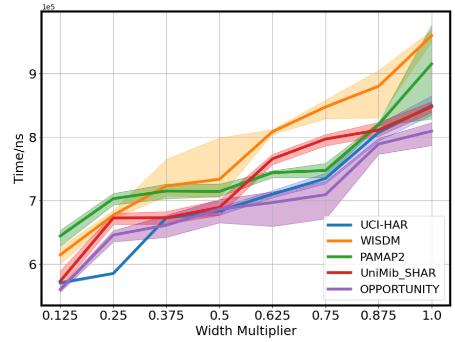

Fig. 5. Inference time of different datasets at varying widths

limitation caused by the trade-off between accuracy and speed, the accuracy drops are an expected behavior, which is totally acceptable for multi-width HAR applications. From an aspect of computational overhead, it is worthwhile to mention that the triangular convolutional layer itself can roughly halve the number of total parameters, where accuracy only slightly decays as network width decreases. In addition, we also measure the number of multiply-add operations (MAC) and inference time(Table V). Due to the lower-triangular design constrain, the computational complexity is significantly reduced. As illustrated in Fig.4 and Fig.5, if one decreases the width factor, the inference time can rapidly decay while the precisions of prediction still can maintain parity.

C. Albation experiment

In order to explore why the lower-triangular convolution can perform better in multi-width HAR scenario, we implement two network architectures: a normal CNN, a tunable width CNN constructed by lower-triangular convolutional layers. For simplicity, we train both models on UCI-HAR dataset at four different widths of 0.25, 0.5, 0.75, 1.0. During training, the means of 1st-layer and variances of 3rd-layer are accumulated from batch normalization layer. At different widths, we compare the means and variances of selected channel. Fig.6 and Fig.7 shows that there is a noticeable deviation of activation statistics within normal CNN if width is varied, which is consistent with our expectation. In other words, different widths produce different activation statistics, which leads to worse performance. Replacing normal convolution with lower-triangular convolution, we could clearly see that lower-triangular weight constrain significantly converges better, which can help to stable activation statistics. It will be more beneficial for providing higher and more consistent validation accuracy when one switches network between different widths.

We continue to investigate how the number of randomly sampled widths (i.e., N) affects classification performance, because too large N will lead to unnecessary training burden. During each iteration, we train our models on PAMAP2 and WISDM datasets when N is set to 1, 2, 4, 8 and 16 respectively. Fig.8 shows that the classification performance will decay rapidly as network width decreases if we perform only one or two sampling. It can be seen that the models

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3132088, IEEE Transactions on Instrumentation and Measurement

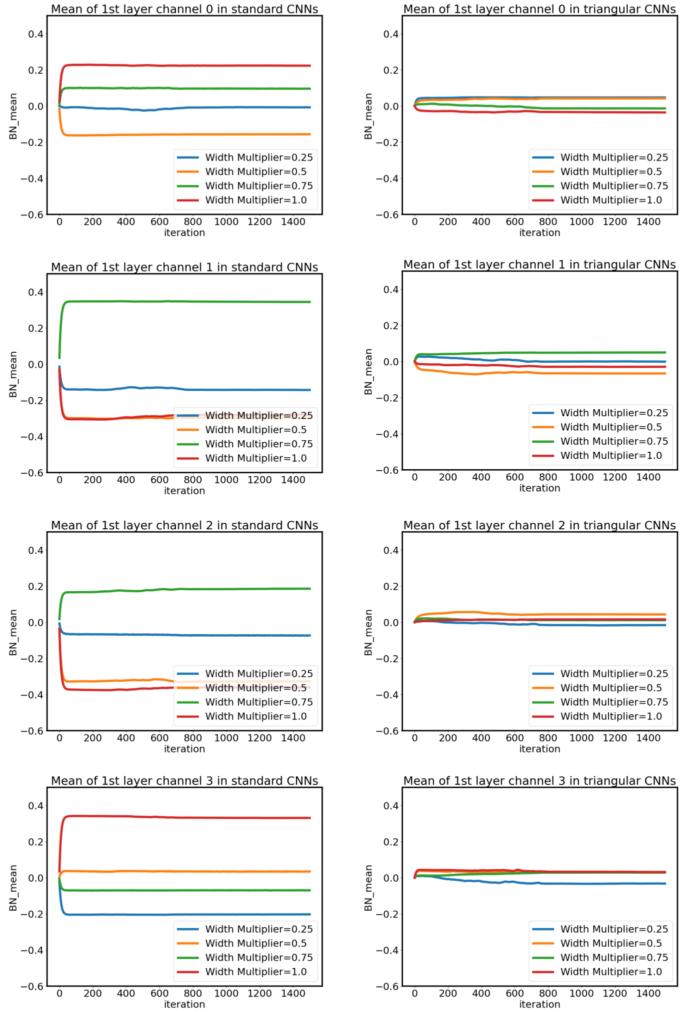

Fig. 6. Mean of standard CNN and triangular CNN

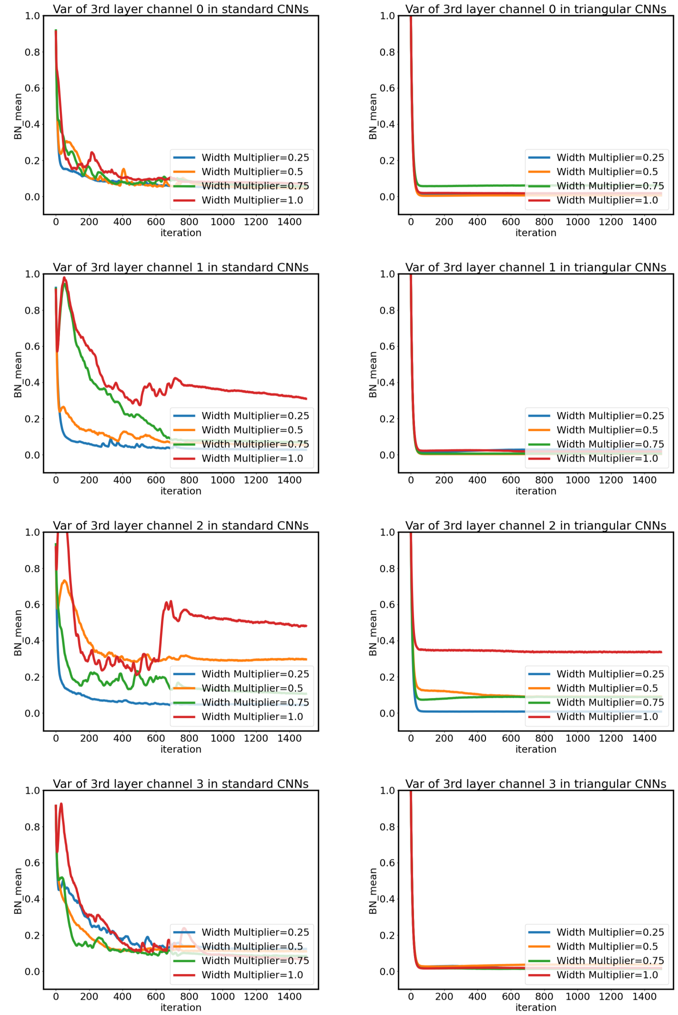

Fig. 7. Variances of standard CNN and triangular CNN

trained with N= 4, 8 and 16 significantly outperform those with N = 1 and 2. The classification accuracy starts to saturate or even drop if N become relatively large (e.g., N = 16). N = 8 is the default through all our experiments.

As indicated in above Eq.2, there is a lower bound k_0, which plays an important role in controlling width. In order to examine how the lower bound influences final classification performance, we perform the evaluation on UCI-HAR dataset, under four different width lower bounds as 0.0625, 0.125, 0.25, and 0.375. As illustrated in Fig.9 and Fig.10, it can be observed that k_0=0.125 has obviously better performance. Compared with k_0=0.0625, a smaller sampling interval is usually enough to produce satisfactory accuracy. The results indicate that the classification accuracy depends on the width lower bound, it is totally feasible to sample width between the interval [0.125, 1].

We further investigate the effectiveness of random width sampling strategy. Four cases are evaluated: N random widths without a minimum width and a maximum width; N-1 random widths without a minimum width; N-1 random widths without a maximum width; N-2 random widths with a minimum width and a maximum width. As we can see, the total number of sampling widths is set to N. In the first case, the total N random widths need to be sampled because both a minimum width and a maximum width are not fixed. In the second or third case, only N-1 random widths need to be sampled because either a minimum width or a maximum width is already fixed. In the fourth case, only N-2 random widths need to be sampled because two widths, i.e., a minimum width and a maximum width are already fixed. The evaluations are performed on UCI-HAR dataset. Fig.11 and Fig.12 emonstrate that the models trained with a minimum width and a maximum width perform better than those without a minimum width and a maximum width. In addition, the model trained with lower bound has higher accuracy than that with upper bound, which indicates the importance of lower bound k_0. Thus, we train all models by fixing lower bound and upper bound.

To verify the effectiveness of the proposed model on a resource-constrained edge device, we deploy the flexible HAR system into a Raspberry Pi 3B+ equipped with an official supported Raspberry Pi operating system. It has a good compatibility with the deep learning library PyTorch 1.7 used in our experiments. The model is trained on WISDM dataset. We run it on the embedded platform to read one sensor sample and perform an online prediction. A Python program

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3132088, IEEE Transactions on Instrumentation and Measurement

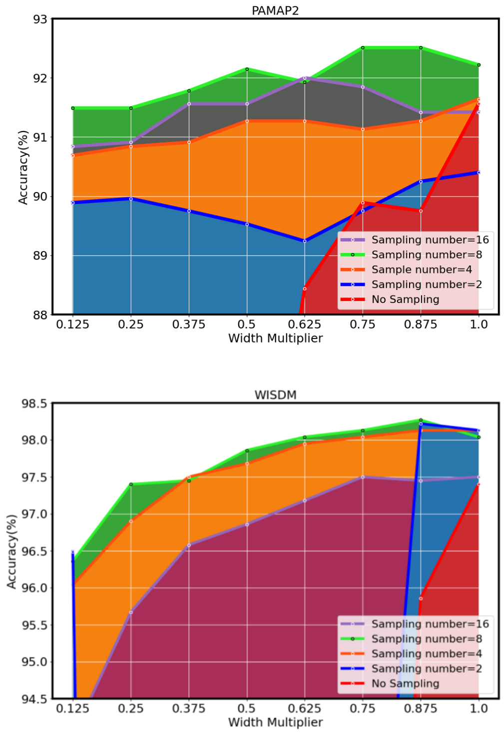

Fig. 8. The speed-accuracy trade-off curves at different sampling number

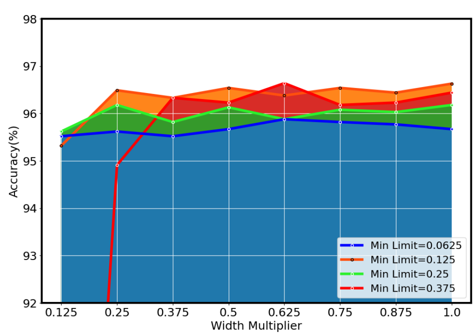

Fig. 9. The speed-accuracy trade-off curves at different width lower bounds

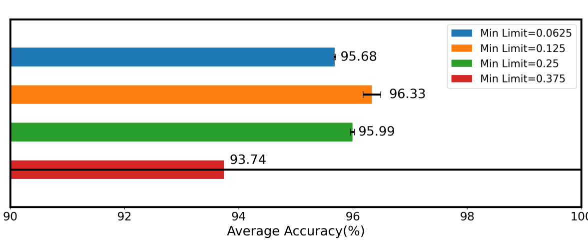

Fig. 10. Performance of different width lower bounds

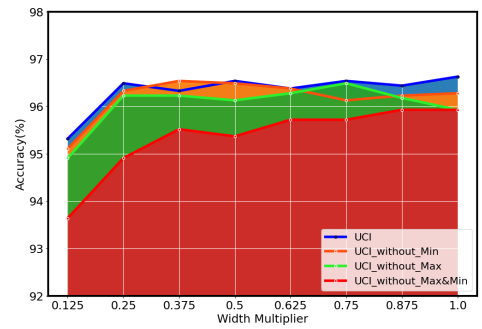

Fig. 11. The speed-accuracy trade-off curves at different sampling rules

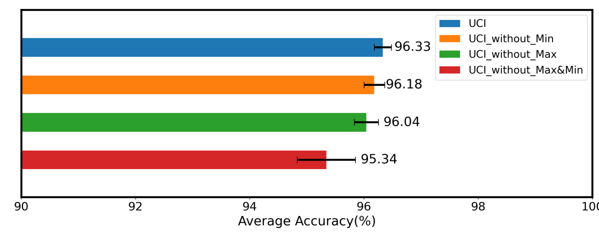

Fig. 12. Performance of different sampling rules

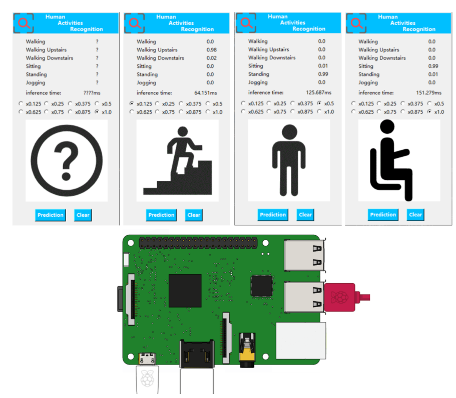

Fig. 13. GUI of the HAR application on Raspberry Pi

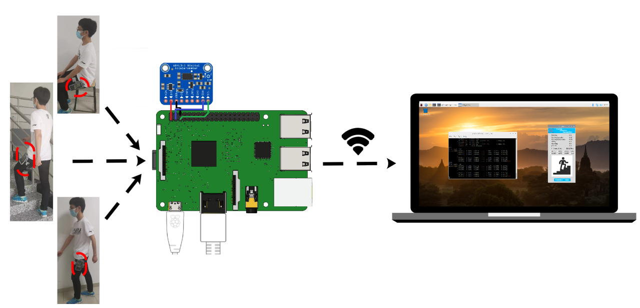

Fig. 14. Experimental demonstration of application on Raspberry Pi 3B+

is developed for the HAR application. Fig.13 illustrates the graphical user interface (GUI) of the application, which is able to tune network width, show prediction probability of each activity and calculate inference time. We compare the inference time at different widths e.g., 0.125, 0.5, and 1.0, and results are shown in Table VI. As network width increases, it takes around 52.901∼68.591ms, 101.434∼125.847ms and 147.261∼167.773ms respectively to predict one ten-second window.

We further add the results of real experiments based on

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3132088, IEEE Transactions on Instrumentation and Measurement

the proposed HAR system. For practical implementation, we choose a 3-axis accelerometer(adxl345) as IMU and communicate it with Raspberry Pi 3B+ via Inter-Integrated Circuit(I2C). We configure Wi-Fi on the Raspberry Pi so as to access it remotely via a laptop computer. As shown in Fig. 14, following the experiment settings in WISDM dataset, the IMU node is attached to the subjects front leg pocket. The main functions are composed of two independent threads process: ProcessSignals and OnlinePrediction. The former is charge of communicating with Raspberry Pi and periodically reads sensor signals, and then data normalized is done by using the mean and standard deviation of training data. The latter is charge of predicting activities. For the WISDM case, a ten-second window with an 95% overlap rate is slide over real sensor readings recorded from the IMU node to generate one sample. Thus, the step length is equal to 500ms, and the recognition system will wait for 500ms to read and predict next sample. That is to say, OnlinePrediction is triggered by scheduled interruptions every 500ms that correspond to 5% of window length. We set three widths to implement activity inference for 500 runs. The inference times with different widths are shown in Fig.15. Overall, the experimental results verify that the proposed method can obtain an efficient speedup for activity inference at much smaller memory footprint.

TABLE VI
RASPBERRY PI'S INFERENCE TIME

Width Multiplier	Time/ms
0.125	52.901∼68.591
0.50	101.434∼125.847
1.0	147.261∼167.773

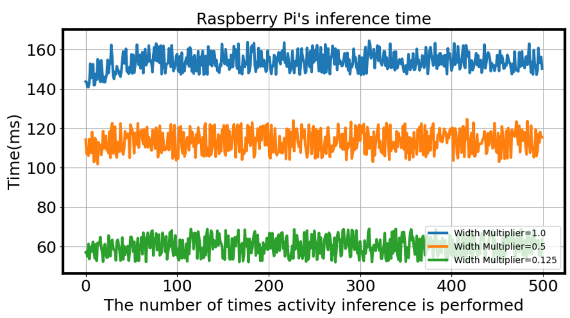

Fig. 15. Inference time

TABLE VII
TEST SET COLLECTED BY RASPBERRY PI

activity	label	number	second
Downstairs	0	315	167
Jogging	1	634	327
Sitting	2	208	114
Standing	3	132	76
Upstairs	4	378	199
Walking	5	615	317

The measured values are collected from two subjects, and the collection process is illustrated in Fig.14. Each subject performs six types of activities: Walking, Jogging, Upstairs, Downstairs, Sitting, and Standing. The embedded measuring system still maintains a sampling rate of 20Hz, where a fixed-length sliding window of 10 seconds and an 95% overlap rate are used to segment raw sensor readings (i.e., 200 raw accelerometer readings per sample). The measured sensor values need to be standardized into zero mean and unit variance. Overall, the test set is composed of 2,282 samples, whose statistics is shown in Table VII. To verify the effectiveness of our model, we summarize our results of the embedded measuring system in Fig.16. In order to show the predictive accuracy associated with each of the activities done by the system, we compute the confusion matrices that contains information about actual and predicted activity classifications. For example, in the case of the activity Jogging, the CNN with width factor 1 makes 5 errors, while the CNNs with width factor 0.5 and 0.125 make 6 and 9 errors respectively. Similarly, for the activity Upstairs, the CNN with width factor 1 makes 45 errors, while the CNNs with width factor 0.5 and 0.125 misclassifies 58 and 77 samples respectively. The results indicate that the classification performance only decays slightly as the width factor decreases, which is in line with our prior results.

V. CONCLUSION

Decent recent years, deep learning has achieved state-of-the-art performance in sensor based HAR area. However, real HAR applications are often deployed across different wearable devices or hardware versions. Thus, an important issue is how to maintain model flexibility for adapting quickly to new computing platform, which has rarely been investigated so far. In this paper, we propose an tunable width convolutional neural network to perform activity recognition on resource-constrained wearable devices. Specially, the network is stacked by lower-triangular convolutional layers rather than normal convolution to avoid the influence of varying batch statistics caused by the switch between multiple widths. The lower-triangular constrain is very suitable for tunable width networks. To acquire smooth control over width for the trade-off between speed and accuracy, we perform random sampling rather than fixed sampling over width, where the network at different widths can be trained simultaneously as sub-networks by accumulating their losses. Evaluation results validate the practicability and effectiveness of the proposed HAR model, compared with normal CNN.

At present, there are at least 24,000 Android devices, which have drastically different computing resources. Even for the same device, the inference speed also always varies due to excessive consumption caused by background apps that decreases the available computational budget. In addition, it will take extra time and data for downloading and offloading models when switching to a larger or smaller model. However, at runtime prior networks need to be re-configured for adapting across different devices for a given response time. This paper attempts to handle this issue: For a given computational

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3132088, IEEE Transactions on Instrumentation and Measurement

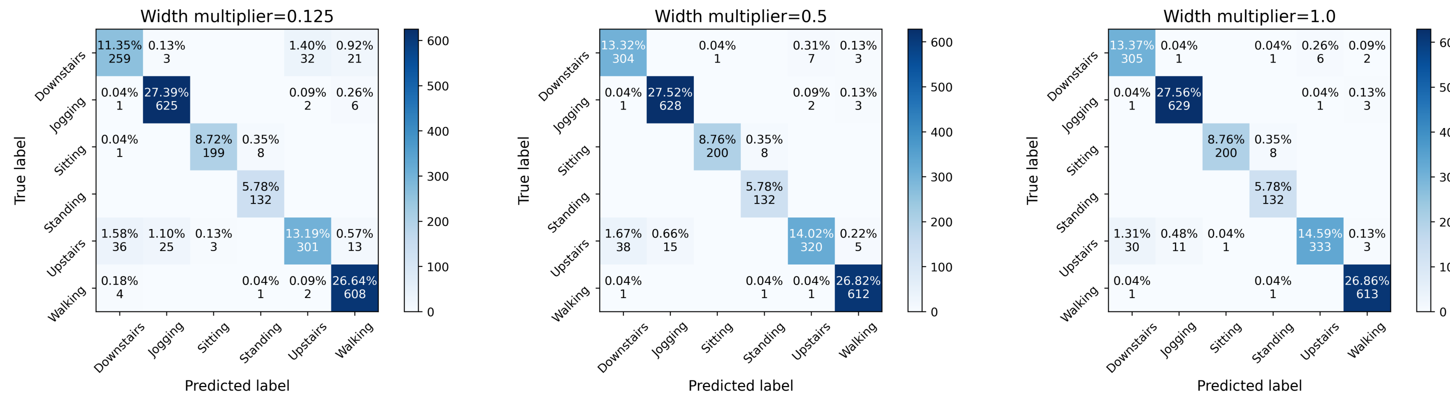

Fig. 16. Test performance on Raspberry Pi

budget, how to adaptively tradeoff accuracy and speed for activity inference at runtime? We propose a tunable-width neural network as a potential solution, which can be executed at different widths. For brevity, the network may be easily varied for a switch by tuning its width, i.e., the number of channels alone. That is to say, we only need to change network width without the need of redownloading or offloading new models. Comparing to prior static networks, our method has several advantages: (1) For different computational budget, a single HAR model is trained. (2) A near-optimal tradeoff between accuracy and speed can be flexibly deployed on a target device by adjusting network width accordingly. (3) The solution is generally applicable to popular mainstream building blocks of neural networks such as convolutions or fully-connected layers, etc. Overall, it is easier to be deployed on wearable or mobile devices with existing deep learning libraries, which could be a better strategy to perform activity inference without extra computational and memory cost. Our research provides a new research direction for building flexible models for activity inference on resource-constrained wearable devices.

References

[1] Z. Chen, C. Jiang, S. Xiang, J. Ding, M. Wu, and X. Li, "Smartphone sensor-based human activity recognition using feature fusion and maximum full a posteriori," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 7, pp. 3992–4001, 2019.

[2] T. Tuncer, F. Ertam, S. Dogan, and A. Subasi, "An automated daily sports activities and gender recognition method based on novel multikernel local diamond pattern using sensor signals," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 12, pp. 9441–9448, 2020.

[3] F. Luo, S. Khan, Y. Huang, and K. Wu, "Binarized neural network for edge intelligence of sensor-based human activity recognition," *IEEE Transactions on Mobile Computing*, 2021.

[4] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[5] S. Xia, L. Chu, L. Pei, Z. Zhang, W. Yu, and R. C. Qiu, "Learning disentangled representation for mixed-reality human activity recognition with a single imu sensor," *IEEE Transactions on Instrumentation and Measurement*, 2021.

[6] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[7] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, no. 3, Jan. 2014.

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

[9] M. Zeng, H. Gao, T. Yu, O. J. Mengshoel, H. Langseth, I. Lane, and X. Liu, "Understanding and improving recurrent networks for human activity recognition by continuous attention," in *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, 2018, pp. 56–63.

[10] Y. Zhang, G. Tian, S. Zhang, and C. Li, "A knowledge-based approach for multiagent collaboration in smart home: From activity recognition to guidance service," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 2, pp. 317–329, 2019.

[11] G. Panahandeh, N. Mohammadiha, A. Leijon, and P. Händel, "Continuous hidden markov model for pedestrian activity classification and gait analysis," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 5, pp. 1073–1083, 2013.

[12] W. Gao, L. Zhang, W. Huang, F. Min, J. He, and A. Song, "Deep neural networks for sensor-based human activity recognition using selective kernel convolution," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.

[13] M. Abbas and R. L. B. Jeannès, "Exploiting local temporal characteristics via multinomial decomposition algorithm for real-time activity recognition," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.

[14] S. Qiu, Z. Wang, H. Zhao, and H. Hu, "Using distributed wearable sensors to measure and evaluate human lower limb motions," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 4, pp. 939–950, 2016.

[15] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *IJCAI*, 2015, pp. 3995–4001.

[16] N. Y. Hammerla, S. Halloran, and T. Ploetz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," 2016.

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

[18] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[19] 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.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[21] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.

[22] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 116–131.

[23] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," 2020.

[24] T. Bolukbasi, J. Wang, O. Dekel, and V. Saligrama, "Adaptive neural networks for efficient inference," in *International Conference on Machine Learning*. PMLR, 2017, pp. 527–536.

[25] Z. Wu, T. Nagarajan, A. Kumar, S. Rennie, L. S. Davis, K. Grauman, and R. Feris, "Blockdrop: Dynamic inference paths in residual networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8817–8826.

[26] A. Veit and S. Belongie, "Convolutional networks with adaptive inference graphs," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–18.

[27] E. Kim, C. Ahn, and S. Oh, "Nestednet: Learning nested sparse structures in deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8669–8678.

[28] J. Yu, L. Yang, N. Xu, J. Yang, and T. Huang, "Slimmable neural networks," in *International Conference on Learning Representations*, 2019.

[29] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6232–6240.

[30] J. Yu and T. Huang, "Autoslim: Towards one-shot architecture search for channel numbers," 2019.

[31] J. Yu and T. S. Huang, "Universally slimmable networks and improved training techniques," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1803–1811.

[32] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 448–456.

[35] T. Vu, M. Eder, T. Price, and J.-M. Frahm, "Any-width networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 704–705.

[36] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert systems with applications*, vol. 59, pp. 235–244, 2016.

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

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

[39] Q. Teng, K. Wang, L. Zhang, and J. He, "The layer-wise training convolutional neural networks using local loss for sensor-based human activity recognition," *IEEE Sensors Journal*, vol. 20, no. 13, pp. 7265–7274, 2020.

[40] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 1307–1310.

[41] Y. Tang, Q. Teng, L. Zhang, F. Min, and J. He, "Layer-wise training convolutional neural networks with smaller filters for human activity recognition using wearable sensors," *IEEE Sensors Journal*, vol. 21, no. 1, pp. 581–592, 2020.

[42] F. Li, K. Shirahama, M. A. Nisar, L. Köping, and M. Grzegorzek, "Comparison of feature learning methods for human activity recognition using wearable sensors," *Sensors*, vol. 18, no. 2, p. 679, 2018.

[43] C. Hu, Y. Chen, L. Hu, and X. Peng, "A novel random forests based class incremental learning method for activity recognition," *Pattern Recognition*, vol. 78, pp. 277–290, 2018.

[44] H. Cho and S. Yoon, "Applying singular value decomposition on accelerometer data for 1d convolutional neural network based fall detection," *Electronics Letters*, vol. 55, no. 6, pp. 320–322, 2019.

[45] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International workshop on ambient assisted living*. Springer, 2012, pp. 216–223.

[46] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[47] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th International Symposium on Wearable Computers*. IEEE, 2012, pp. 108–109.

[48] D. Micucci, M. Mobilio, and P. Napoletano, "Unimib shar: A dataset for human activity recognition using acceleration data from smartphones," *Applied Sciences*, vol. 7, no. 10, p. 1101, 2017.

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

[50] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–40, 2021.

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

Lei Zhang received the B.Sc. degree in computer science from Zhengzhou Universitiy, China, and the M.S. degree in pattern recognition and intelligent system from Chinese Academy of Sciences, China, received the Ph.D. degree from Southeast University,China, in 2011. He was a Research Fellow with IPAM, UCLA, in 2008. He is currently an Associate Professor with the School of Electrical and Automation Engineering, Nanjing Normal University. His research interests include machine learning, human activity recognition and computer vision.

Wenbo Huang received the B.S. degree from Nanjing University of Technology, Nanjing, China, in 2019. He is currently pursuing the M.S. degree with Nanjing Normal University. His research interests include activity recognition, computer vision, and machine learning.

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

Hao Wu received the Ph.D. degree in computer science from Huazhong University of Science and Technology, Wuhan, China, in 2007. Now, he is an associate professor at School of Information Science and Engineering, Yunnan University, China. He has published more than 50 papers in peer-reviewed international journals and conferences. He has also served as reviewers and PC members for many venues. His research interests include natural language processing, recommender systems and service computing.

Jun He received the Ph.D. degree from Southeast University, Nanjing, China, in 2009. He was a Research Fellow with IPAM, UCLA, in 2008. From 2010 to 2011, he was a Post-Doctoral Research Associate with the Chinese University of Hong Kong. He is currently an Associate Professor with the School of Electronic and Information Engineering, Nanjing University of Information Science and Technology. His main research is in the areas of machine learning, computer vision, and optimization methods. In particular, he is interested in the applications of weakly supervised learning via reinforcement learning methods.

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