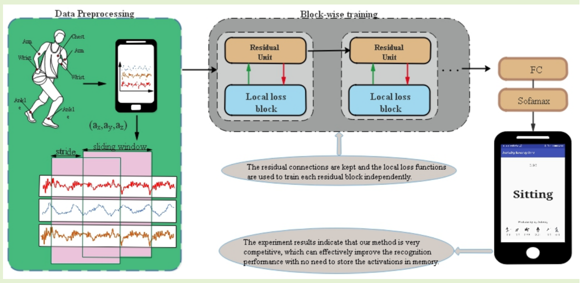
Block-wise training residual networks on multi-channel time series for human activity recognition

Qi Teng, Lei Zhang, Yin Tang, Shuai Song, Xing Wang, Jun He , *Member, IEEE*

***Abstract*— Recently, human activity recognition (HAR) has become an active research area in wearable computing scenario. On the other hand, residual nets have continued to push the state-of-the-art of computer vision and natural language processing. However, residual nets have rarely been considered in the HAR field. As residual nets grow deeper, memory footprint limit its wide use for a variety of HAR tasks. In this paper, we present a novel block-wise training residual nets that use local loss functions for HAR applications. Instead of global backprop, the local cross-entropy loss together with a supervised local similarity matching loss is utilized to train each residual block independently, in which gradient need not to be propagated down the network. As a result, the gradient and activations do not have to be kept in memory any more, which alleviates the memory requirements and is more beneficial for wearable HAR computing. We demonstrate the effectiveness of block-wise training residual nets on OPPORTUNITY, WISDM, UNIMIB SHAR and PAMAP2 datasets, which establishes obvious better classification accuracy compared to equally-sized residual nets, even though memory requirement is much smaller.**


Data Preprocessing
Chest
Arm
Wrist
Arm
Wrist
Ankle
Ankle
(a_x, a_y, a_z)
stride
sliding window
Block-wise training
Residual Unit
Local loss block
Residual Unit
Local loss block
FC
Sofamax
Sitting
The residual connections are kept and the local loss functions are used to train each residual block independently.
The experiment results indicate that our method is very competitive, which can effectively improve the recognition performance with no need to store the activations in memory.

***Index Terms*—Human activity recognition, convolutional neural networks, wearable sensors, residual network, local loss**

SEnsor based human activity recognition (HAR) [1] [2] is a hotspot where various human activities are analyzed via embedded inertial sensors such as accelerometer and gyroscope. HAR can be treated as a typical multichannel time series pattern recognition problem [3], which utilizes sliding window technique to segment sensor data and extract features for classifying activities.

Conventional machine learning approaches such as multi-layer perception (MLP) [4], decision tree [5], support vector machine (SVM) [6] have made considerable progress on HAR. However, the traditional approaches usually depend on heuristic handcrafted feature extraction requiring human domain knowledge, which limit their wide applications for most daily HAR tasks [7]. Recent advances have seen the success of deep learning, that is able to greatly relieve the effort on designing features [8] and achieve unparalleled performance in many research areas such as computer vision and natural language processing (NLP). Not only computer vision or NLP task is able to benefit from such a automatic feature extraction mechanism, HAR is also a perfect match as it often need to handle translation invariance [9] of sensor signals, as well as hierarchical structure of activities. Many researchers have tried to apply deep neural networks (DNNs), especially convolutional neural networks (CNNs) to HAR systems, which are able to extract automatically features without human domain knowledge.

The development of deep learning makes sensor based HAR undergo a transition from feature design to network design. Research in computer vision and NLP field has always been at the forefront of this work. For a wide range of computer vision tasks, the newest network architectures have been growing deeper. Deep residual networks (ResNets) [10] put forward latest state-of-the-art across multiple visual processing tasks. The residual block is an essential architecture innovation behind ResNets, which allows information to be passed directly through, making the backpropagated gradient error signals less prone to exploding or vanishing. Though ResNet models achieved remarkable performance in NLP and computer vision, it has not been fully exploited in multichannel time series for HAR. The vastly increased depth can bring significant performance gains, but it inevitably requires large amount of memory to store the network's activations. As networks grow deeper, the increasing memory burden inevitably prevents the wide use of ResNets in sensor based HAR deployed on mobile

The work was supported in part by the National Science Foundation of China under Grant 61203237 and in part by the Natural Science Foundation of Jiangsu Province under grant BK20191371. *(Corresponding author: Lei Zhang.)*

Qi Teng, Lei Zhang Yin Tang and Xing Wang are with School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing, 210023, China (e-mail: teqi159@gmail.com; leizhang@njnu.edu.cn).

Jun He is with the School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China.(e-mail: jhe@nuist.edu.cn).

Shuai Song is with the School of Department of Computer Science and Technology,Tsinghua University, Beijing 100000, China.

phones or other wearable devices.

Actually, ResNets, similar to most modern neural networks, are trained with gradient backpropagation, in which the hidden activations have to be kept until the forward and backward propagation has completed [11]. The phenomenon is termed as the backward locking, which prevent the reuse of memory resources used to store hidden activations. Moreover, the gradient and the activations have to be stored, and as a result memory consumption becomes a bottleneck for embedded HAR applications with limited computational resources. Due to biological plausibility, some neuroscientists and biologists have criticized the gradient backprop of global loss [12] [13]. To tackle the above challenges, the Reversible ResNet (RevNet) [14], a variant of ResNets [15], is proposed. In the RevNet, as one can exactly reconstruct each layer's activations from the next layer's, the activations for most layers need not to be kept, which provides considerable reduction in memory footprint at little or no cost to performance. Another way to avoid the backward locking problem is to train each hidden layer independently with local loss functions, instead of global backpropagation. Recently, several layer-wise training methods with local loss functions have been developed [16] [17], which could be more biologically plausible compared to global loss because gradient needs not be propagated back along the whole network. However, due to the existence of skip connection, the above layer-wise training ideas are not suitable for residual architecture.

In this work, we presented a new block-wise training ResNet with local loss functions for HAR in ubiquitous computing scenarios. Instead of globally back-propagating errors, each residual block is trained independently by local learning signals, which need not to be back-propagated down the network. The local learning signals are generated by two separate single-layer sub-networks, where one is trained with a supervised similarity matching loss, and the other with a conventional cross-entropy loss. The local error signals are only backpropagated within each residual block. After each residual block, the computational graph is detached to prevent backward gradient flow from the output loss. The normal cross-entropy loss is utilized to train the final output layer. We evaluate our block-wise training ResNet on four public benchmark HAR datasets, namely WISDM [18], PAMAP2 [19], UNIMIB SHAR [20], and OPPORTUNITY [21]. Extensive experiments are performed to compare the block-wise training ResNet with baselines, as well as other state-of-the-art algorithms. The experiment results indicate that our method is very competitive, which can effectively improve the recognition performance with no need to store the activations in memory. In summary, our main contributions are three-fold:

- A novel block-wise training ResNet is proposed for HAR in multimodal HAR scenarios, where the local learning signals provided by two separate single-layer sub-networks are used to train each residual block independently for avoiding the influence of skip connection, that need not to be backpropagated to previous blocks.
- The RevNet, constraining the architecture to be reversible, only can achieve nearly identical classification accuracy to equally-sized ResNets. On the contrary, the presented methods can outperform RevNet, as well as ResNet, by a large margin even though memory requirements are much smaller. That is to say, the local cross-entropy combined with supervised similarity matching loss can outperform global backprop.
- Several ablation experiments are performed, which demonstrate the effectiveness and efficiency of the block-wise training ResNet. The impact of some hyper-parameters such as activation function and layer number are studied.

In this paper, we demonstrate that residual nets can be trained with locally generated errors, which is generated by two different supervised loss functions, i.e., a cross-entropy loss and a similarity matching loss. It does not depend on a globally generated error, and the gradient is not backpropagated down the whole network. That is to say, the hidden weights can be updated during the forward pass, which is different from global backprop. When the weights for each residual block have been updated, the gradient and the activations do not have to be kept in memory any more, which greatly alleviates the memory requirements when training residual net. Although we train the whole network at the same time, locally generated errors also enables us to greedily training residual blocks one-at-a-time, which can reduce the memory footprint. On the whole, compared with global backprop, we demonstrate that the proposed algorithm that uses local backprop is more accurate and memory efficient.

I. RELATED WORKS

Over the past five years, due to development of the computational capabilities, deep learning has made promising results on multichannel time series for HAR, which outperformed state-of-the-art shallow algorithms requiring cumbersome hand-crafting feature extraction. [22], [23] firstly developed a method based on CNN to automatically extract discriminative features from sensor signals for activity recognition. In [24], CNNs are utilized to extract local feature, which is combined with simple shallow statistical features to preserve information about the global form of time series sensor signals. [25] proposed a deepConvLSTM framework for activity recognition via combining CNN and Long Short-Term Memory (LSTM) units, that can better capture temporal dependencies of time series signals. [26] proposed a novel attention-based CNN for multimodal HAR which can learn the dependencies of sensor signals across both spatial and temporal dimension. Although ResNet models have continued to push the state-of-the-art in computer vision areas, it has not been fully exploited in multichannel time series for sensor based HAR.

Recently, [14] presented the RevNet, which can achieve nearly identical classification performance to equally-sized ResNets. As the activations of each layer can be exactly computed from the next layer's, the hidden parameters for most layers do not have to be stored in memory any more. Another interesting direction is approximating the activations of previous layers with local loss functions. In certain cases, independently of the global loss, local loss functions have been used to pre-train hidden layers, which can further improve the performance after fine-tuning using global back-propagation [27]. Without fine-tuning, [28] directly used supervised layer-wise loss functions with ensembling, approaching the accuracy of global loss on the CIFAR-10. Specially, [16] proposed to combine local cross-entropy loss and supervised similarity matching loss to train each hidden layer in VGG-like networks independently, which can match the equally-sized ResNets with global backprop in terms of test error. However, due to the occurrence of skip connection, the above layer-wise ideas are not suitable for deep ResNets. Our approach differs from the above methods because we don't try to train the networks layer by layer. Instead, we utilize the target vector of each block to create an error signal independently of the blocks above.

II. MODEL

In this part, we introduce our block-wise training ResNet framework with local loss functions. The proposed model mainly consists of several residual units, dense layers and softmax layers, which will be explained below. Recently, Nøkland et al used local loss functions to train the VGG-like network architectures layer by layer, which can achieve almost the same performance with equally-sized residual architectures trained with global back-propagation. Different from the [16]'s works, in the paper, the residual connections are kept and the local loss functions are used to train each residual block independently. We aim to improve the results acquired by residual models further. For this purpose, two single-layer sub-networks are used to generate two different local learning signals to update and optimize the hidden layer parameters within each residual block, and the block-wise training ResNet is shown in Fig.1. The computational graph is detached after each residual block to inhibit gradient back-progapatioon, and the hidden activation weights can be updated along the forward pass. As a result, instead of globally backpropagating errors, the weights of all residual blocks are trained simultaneously or orderly one-by-one via local learning signals, which can avoid the above back-locking problem, and meanwhile reducing the requirement for computing resource.

A. Residual Unit

Actually, residual nets perform significantly better than VGG nets for visual recognition tasks. However, residual learning has rarely exploited in HAR scenario. During recent years, VGG-like CNNs have achieved state-of-the-art accuracy across various HAR datasets. In the paper, we firstly propose a block-wise training ResNet in HAR scenario. In comparison with ImageNet dataset, several public HAR datasets are much smaller, which is not enough to train deep networks with many layers. In order to prevent overfitting, we build a light-weight residual net with 3 building blocks as our backbone, where each block contains 2 stacked layers with skip connection, which can perform well in small HAR datasets. In addition, we use local loss functions to train the residual network block by block, and the results shows that our model can obtain state-of-the-art performance in wearable HAR scenario. In detail, we perform activation Rectified Linear Unit (ReLU) before addition and therefore the stacked layers orders is Conv-BN-Relu-Conv-BN-ReLU, where BN and ReLU denote batch normalization and Rectified Linear Unit respectively. As can be seen from Fig.2, adopting residual learning to every few stacked layers [10], the building block can be defined as:

$$y^{l+1} = \mathcal{I}\left(y^l\right) + \mathcal{F}\left(y^l, \boldsymbol{w}^l\right) \tag{1}$$

where $\mathcal{F}$, a function called the residual function, is the learned residual mapping from a shallow neural net. $\boldsymbol{w}^l = \left\{w_{jk}^l \,|1 \leq j \leq n, 1 \leq k \leq m\right\}$ is a set of two-layer weights associated with l-th residual block, and w_{jk}^l is the weight from the k-th neuron in the first layer to the jth neuron in the second layer. The function $\mathcal{I}$ denotes an indentity mapping. More specially, as can be shown in Fig.2, each block takes input y^l and produces output y^{l+1} alone with the forward flow:

$$c\left(y^l\right) = Conv\left(y^l\right) \tag{2}$$

$$a\left(y^l\right) = \text{Re}LU\left(BN\left(c\left(y^l\right)\right)\right) \tag{3}$$

A basic residual unit can be reconstructed as follwing computation:

$$y^{l+1} = h\left(y^l\right) + a_2\left(a_1\left(y^l\right)\right) \tag{4}$$

where a_i, $i \in \{1, 2\}$, denote the output of stacked layers.

B. Similarity Matching Loss

We consider a similarity matching loss to compute L2 distance between matrices [16] [29], in which the elements pose pairwise similarities between examples in a mini-batch. Formally, in this paper the similarity matching loss can be expressed as following form:

$$\mathcal{L}_s = \left\|S\left(c\left(y^{l+1}\right)\right) - S\left(Y\right)\right\|_2 \tag{5}$$

where Y = $(y_1,\, y_2,\, \ldots y_n)$ is a set of one-hot encoded label, and $c(.)$ is a convolutional layer in similarity matching loss.

A standard deviation operation is implemented on the feature maps c(.), which aims to reduce the dimension of the output to 2D. $S(H)$ is the cosine similarity matrix operation, in which H denotes a mini-batch of hidden layer activations. The symmetric S_{ij}, an element in the $S(\mathcal{H})$, is formulated as follows:

$$S_{ij} = S_{ji} = \frac{\left\langle \tilde{\mathcal{H}}_i, \tilde{\mathcal{H}}_j \right\rangle}{\left\|\tilde{\mathcal{H}}_i\right\|_2 \left\|\tilde{\mathcal{H}}_j\right\|_2} \tag{6}$$

where the notation $\tilde{\mathcal{H}}_i$ is the mean-centered vector $S(\mathcal{H})$. For simplicity, the similarity matching loss, a supervised clustering loss, abbreviated as sim loss or $\mathcal{L}_s$ in this paper.

C. Prediction Loss

As for prediction loss, we employ the cross-entropy function between the target and a prediction of local linear classifier as

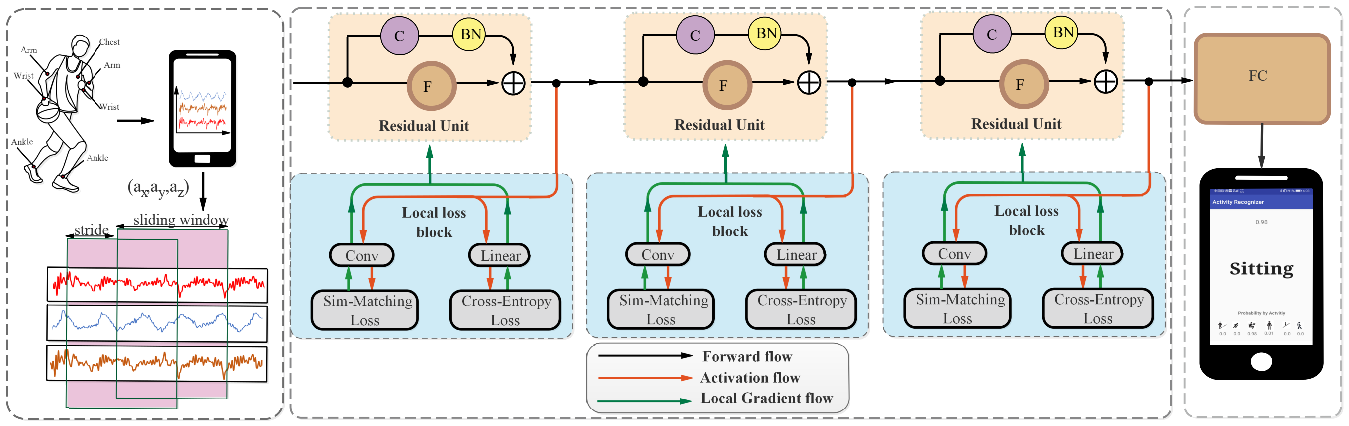


Fig. 1: The overview of the predsim ResNet.

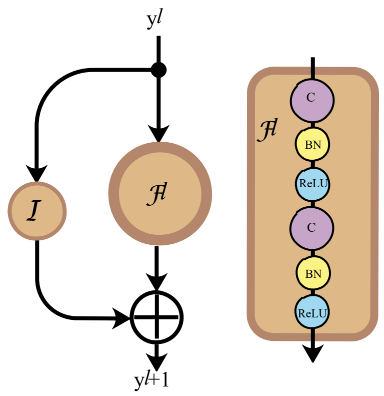


Fig. 2: The residual unit: ReLU before addition

the other local loss signal.

$$\mathcal{L}_p = CE\left(Y,\ \boldsymbol{w}_l^T (y^{l+1})^\dagger\right) \tag{7}$$

where $\boldsymbol{w}_l^T$ denotes a weight matrix of linear classifier. Through applying flatted operation we obtain $(y^{l+1})^\dagger$ from the original y^{l+1}. For simplicity, the prediction loss is denoted as pred loss or $\mathcal{L}_p$ in this paper.

D. Local Loss

In order to generate local error signals for the each residual block, we proceed by implementing a weighted combination between the sim loss $\mathcal{L}_s$ and the pred loss $\mathcal{L}_p$.

$$\mathcal{L}_{sp} = \alpha\mathcal{L}_s + (1-\alpha)\,\mathcal{L}_p \tag{8}$$

where α is a hyper-parameter to be tuned. The combination is denoted as predsim loss or $\mathcal{L}_{sp}$ in this paper.

E. Backprop Within Residual Unit

In a plain ResNet, the gradient can be decomposed into two additive parts: one part that propagates information along skip connection directly without concerning any weight layers, and another part that propagates through the weight layers. In contrast to plain ResNet, the local loss is applied within each residual unit independently, which need not to be propagated between different units, either forward or backward. Thus, instead of an identity mapping, we only focus on using the local loss functions to update the weight layers. Through the chain rule of backpropagation, we can obtain:

$$\frac{\partial \mathcal{L}_{sp}}{\partial w_{jk}^l} = \delta^l \cdot \frac{\partial \mathcal{F}^l}{\partial w_{jk}^l} \tag{9}$$

where the output error δ^l can be written as :

$$\delta^l = \frac{\partial \mathcal{L}_{sp}}{\partial y^{l+1}} \cdot \frac{\partial y^{l+1}}{\partial \mathcal{F}^l} \tag{10}$$

The block-wise training algorithm is shown in Alg.1. As a result, the forward and backward propagation along the whole network is transform into one-at-a-time local block-wise training, which avoids large amount of memory requirement without any extral cost. Note that the final output layer trained by conventional cross entropy function is detached to prevent the gradient flow from the previous blocks. The source code will be available.

III. Experiment

In the paper, we do conduct extensive experiments on OPPORTUNITY, WISDM, UNIMIB SHAR and PAMAP2 datasets to test the proposed method. The four public datasets have been widely adopted to evaluate various HAR approaches using machine learning technique. In the section, we detail setup and the numerical results.

A. Experimental Setup

The experiments are implemented in Pytorch framework on a server with NVIDIA 2080ti GPU with 11GB video memory, 64 GB memory. Referring to Zeng et al's work [22], we set the batch size to 200. In Zeng et al' work, stochastic gradient descent with the learning rate of 0.05 is used. For simplicity and without loss of generality, adam optimizer with dynamic learning rate is used to make training process faster. The initial learning rate is set to 0.001. The parameters in optimizers are initialized by the default setting. The weighting factor α is set to 0.99 for all the experiments.

Algorithm 1: Block-wise training ResNet

```
Input: $\mathcal{L}_g$:Global loss; $\boldsymbol{CE}$: Cross-Entrory;
  $\mathcal{O}$: The output of the final FC layer;
  L:The number of all residual blocks;
  $l$: The index of the residual block;
  $(w^L)^\dagger$:The weights of the $l$-th block;
  $y^l$:The output of each residual block;
  Y:The label matrix of one-hot encoded target output;
  $w^l_{jk}$: The weight of $l$-th block from the $k$-th neuron in
  the ($i$-1)-th layer to the $j$-th neuron in the $i$-th layer;
1 Gradient back-propagation is detached from
    computation graph.
2 Initializing all weight parameters $W$
3 while not Stop-Criterion do
4     if $l \leq L$ then
5         $\mathcal{L}_{sp} \leftarrow \alpha\mathcal{L}_s + (1-\alpha)\mathcal{L}_p$;
6         $\boldsymbol{\delta}^l \leftarrow \frac{\partial\mathcal{L}_{sp}}{\partial y^{l+1}} \cdot \frac{\partial y^{l+1}}{\partial\mathcal{F}^l}$;
7         $\frac{\partial l_{sp}}{\partial w^l} \leftarrow \delta^l_i \cdot \frac{\partial y^l}{\partial w^l_{jk}}$;
8         $(w^l)^\dagger \leftarrow w^l$;
9     else
10        $f \quad \leftarrow \boldsymbol{w}^{L+1} \cdot \mathbf{y}^L$;
11        $\boldsymbol{z} \quad \leftarrow \sigma(f)$;
12        $l_g \quad \leftarrow \boldsymbol{CE}(\mathcal{O}, Y)$;
13        $\frac{\partial l_g}{\partial w^{L+1}} \leftarrow \nabla_{\boldsymbol{z}} l_g \odot \sigma'(\boldsymbol{y}^L) \cdot (\boldsymbol{y}^L)$;
14        $(w^{L+1})^\dagger \leftarrow w^{L+1}$;
```

Due to highly imbalanced class in naturalistic human activity datasets, we use the mean F1 score in the evaluation, which weights classes according to their sample proportion:

$$F_1 = \sum_i 2 \cdot \beta \frac{precision_i * recall_i}{precision_i + recall_i} \tag{11}$$

where i denotes class index and β_i denotes the sample proportion of calss i. For a given class i, $recall_i = TP_i/TP_i + FN_i$, $recall_i = TP_i/TP_i + FN_i$. TP_i,FP_i represents the number of true and false positive respectively and FN_i corresponds to the number of false negatives.

B. Datasets

The proposed method is performed on four public HAR datasets, which are recorded in different contexts by either worn or embedded into smart scenarios in which one subject performed activities. Sliding windowing techniques have been extensively leveraged to segment sensor time series, which has an important effect on the accuracy of the algorithm. So far, there is no clear consensus about which window size should be preferably employed.It has been rarely studied. Actually, to our knowledge, most existing window selections normally depends on figures used in previous successful works, in which there is no strict reseaches to support them. According to our intuition, larger window sizes are normally utilized for the recognition of complex activities, which takes place for a longer time period. For larger window size, the recognition system has to "wait" the longer for a new window to output a prediction, which requires more computing resources. Instead, the smaller window size is more beneficial for a faster activity detection, e.g., detecting falls, as well as reduced computing resource consumption. In this paper, in order to proceed fair comparison, we use the same sizes used in previous classical literatures [18] [19] [20] [21] [30] for each dataset. The datasets are used as follows.

- **OPPORTUNITY** It includes various naturalistic human activities recorded from four subjects in a very rich sensor environment [21].The body-worn sensors include 7 inertial measurement units and 12 3D acceleration sensors. The inertial measurement units provide readings of: 3D acceleration, 3D rate of turn, 3D magnetic field, and orientation of the sensor with respect to a world coordinate system in quaternions. Five sensors are on the upper body and two are mounted on the user's shoes. The acceleration sensors provide 3D acceleration. They are mounted on the upper body, hip and leg. Four tags for an ultra-wideband localization system are placed on the left/right front/back side of the shoulder. At a sampling rate of 30Hz, sensor signals are recorded from 12 different locations on the subject, which is annotated with 18 mid-level gesture annotations. Five different runs are recoded for each volunteer. The sliding time window and sliding step length are set to 64 and 8 respectively, which generates approximately 650k samples.ADL1,ADL2 and ADL3 from subject 1, 2 and 3 are used as training set and ADL4 and ADL5 from subject 4 and 5 are left for testing.
- **WISDM** The dataset is collected using triaxial accelerometer sensor embedded on Android smartphone [31]. Each volunteer placed their smartphone in their front leg pocket and performed 5 asked activities consisting of walking, jogging, going upstairs, going downstairs, sitting and standing. The sensor data are recorded at a frequency of 20HZ. The sliding window is set to 10s and a 95% overlapping rate is used, which corresponds to 0.5 second's sliding step length. We preprocess the dataset as described and use 30% of the data in each class for testing, and the rest 70% data for training.
- **UNIMIB SHAR** The dataset is a set of fall detection dataset collected from 30 subjects. It consists of 17 types of activities: 9 types of ADLs and 8 types of falls. Each volunteer carrying a Nexus I9250 smartphone was instructed to perform activities. The half of all subjects placed the smartphone in their right pocket, and the other half in their left pocket. The accelerometer signals are recorded with an embedded Bosh BMA220 at a constant sampling rate of 50 Hz. The window is set to T=151, which equals to approximately 3s. The samples are randomly split into 70% training and 30% test set.

- **PAMAP2** The dataset is collected from 9 participants who wear 3 inertial measurement units (IMUs) [19]. Each IMU is composed of an accelerometer, gyroscope, magnetometer, temperature and heart rate sensor. Each participant was asked to perform 12 protocol activities (“walking”, “lying down”, “standing”, etc) and 6 optional activities (“watching TV”, “folding laundry”, etc). The sampling rate of IMUs is 100Hz, which is downsampled to 33.3HZ. The window with the size 512 (5.12 seconds) is used to slide one instance at a time, which produces around 473k samples with a 78% overlap rate. The PAMAP2 dataset are randomly partitioned into 2 parts, in which 80% for training and 20% for testing.

C. Comparing Algorithms

The proposed model is compared with the following baseline algorithms:

- **Plain CNN**: An equally-sized CNN with standard global backprop.
- **Plain ResNet**: An equally-sized ResNet with the global loss. This baseline is set to investigate the efficacy of standard global backprop. The baseline has almost the same architecture as the above plain CNN, expect that a shortcut connection is added.
- **Predsim CNN**: An equally-sized CNN with predsim loss.

To verify the contributions of different local loss functions, we consider three variants of our model as follows:

- **Pred ResNet**:The block-wise training residual model with the pred loss. This baseline is set to investigate the efficacy of the pred loss on residual architecture.
- **Sim ResNet**: The block-wise training residual model with the sim loss. This baseline is set to investigate the efficacy of the sim loss on residual architecture.
- **Predsim ResNet**: The block-wise training residual model with the combination of pred loss and sim loss, i.e., predsim loss. This baseline is set to investigate the efficacy of the predsim loss on residual architecture.

In addition, extensive experiments are conducted to compare our method with other state-of-the-arts respectively on OPPORTUNITY, WISDM, UNIMIB SHAR and PAMAP2.

IV. Discussion

The proposed model is compared with baselines on four public datasets, as shown in Table I. The test classification errors on different baselines are illustrated in Fig.3. As shown in the table, the residual model outperforms the equally-sized. CNN by a large margin, due to its greater feature extraction capability. Compared with baseline CNNs, the proposed algorithm is able to achieve 0.97%,3.39%,2.2% and 4.05% relative accuracy gain on WISDM, UNIMIB SHAR, OPPORTUNITY, and PAMAP2 datasets respectively, which has a great improvement. The proposed method that uses local loss functions is considerably better than the plain counterparts, even though the memory requirements are much smaller. The figure shows that the block-wise training ResNet consistently performs best among all baseline algorithms, which achieves significant performance improvement compared to the plain CNN and ResNet, as well as the layer-wise training CNN.

We also compare the block-wise training residual ResNet and its two variants with the state-of-the-art methods, which are from those literatures such as [32] [26] [24] [33], accordingly (Table I). The test classification errors with different local loss functions are illustrated in Fig.4. As can be seen from Table I, the predsim loss exceeds its two variants. The results indicate that training with a local cross-entropy (pred) or supervised similarity matching (sim) loss alone cannot provide a remarkable good training signal for updating hidden layers. The performance can improve significantly when both loss functions are combined (predsim). The proposed model with predsim loss performs better than the state-of-the-art methods by a large margin on those datasets. Fig.4 shows that the predsim loss consistently performs best, which achieves significant performance improvement compared to its two variants in terms of test error. In most cases, the test error for sim loss is lower than that for pred loss, which indicates that the supervised sim loss is more suitable for creating training signals to update weights.

On the whole, the local loss is able to achieve lower test errors than global backprop. Actually, as mentioned above, the local similarity matching loss (sim) alone is able to provide a remarkable good training signal for each residual block. Due to the use of the cosine similarity, the local loss encourages examples from distinct classes to have distinct representations, which can be deemed as a kind of supervised clustering technique (also see equation 5). This supervised learning objective is sufficient to create a hidden representation that is suitable for classification, independently of the blocks above. When both loss functions, i.e., pred loss and sim loss, are combined (predsim), the test errors decrease significantly. That is to say, the local similarity matching loss plays a key role in enhancing the accuracy and mean F1 score.

The normalized confusion matrices on PAMAP2 are compared. Fig.5 shows that many of the misclassifications come from the confusion between two kinds of very similar activities such as ’Walking’ and ’Rope juming’. This can be attributed to their similar vibrations in signal waveforms. From the results, the plain CNN and the predsim CNN makes 48 and 42 errors respectively, while the plain ResNet and predsim ResNet misclassifies only 45 and 32 activities, which demonstrates the advantage of our model in recognition accuracy.

Next we study the influence of the layer number on UNIMIB SHAR via varying the number of blocks in ResNets from 2, 3, 4, 5. We construct 4-layer, 6-layer, 8-layer and 10-layer ResNets via utilizing 2-layer residual blocks. In addition, each 2-layer block is replaced with one 3-layer bottleneck block, which results in the 6-layer, 9-layer, 12-layer and 15-layer ResNets. As illustrated in Table II, the 2-layer block performs better than the 3-layer counterpart. The ResNet-8 performs best among all the architectures.

He et al have found that there is significant difference in recognition performance when changing the usage of activation functions in residual blocks. To further investigate the performance of the proposed model with respect to different usages of activations, we have examined three com-

TABLE I: The overall comparison results on the four datasets (%). Note that the results of the-state-of-arts are directly copied from the corresponding articles.

Model \ Dataset	WISDM			UNIMIB			PAMAP2		OPPOTUNITY		
	Acc(%)	F1(%)	Par	Acc(%)	F1(%)	Par	Acc(%)	Par	Acc(%)	F1(%)	Par
CNN	98.03	98.02	2.09M	76.94	76.31	1.48M	91.75	2.29M	79.01	78.55	1.48M
Predsim CNN	98.73	98.73	0.10M	78.85	78.11	0.11M	92.57	0.88M	81.18	80.79	0.09M
ResNet	97.54	95.73	3.39M	75.20	74.66	3.04M	92.58	3.04M	79.29	79.14	2.28M
Pred ResNet	97.33	97.32		75.77	75.52		90.43		80.67	80.11	
Sim ResNet	98.85	98.84	0.10M	79.42	79.25	0.11M	91.36	0.88M	82.42	81.99	0.09M
Predsim ResNet	**99.00**	**99.10**		**80.33**	**79.89**		**93.95**		**83.06**	**82.67**	
-	[31]	98.20		[29]	78.07		[32]	93.70	[33]	72.68	
-	[34]	98.23		[35]	74.66		[26]	89.30	[24]	78.90	
-	[25]	93.32		-			[36]	93.38	[22]	76.83	

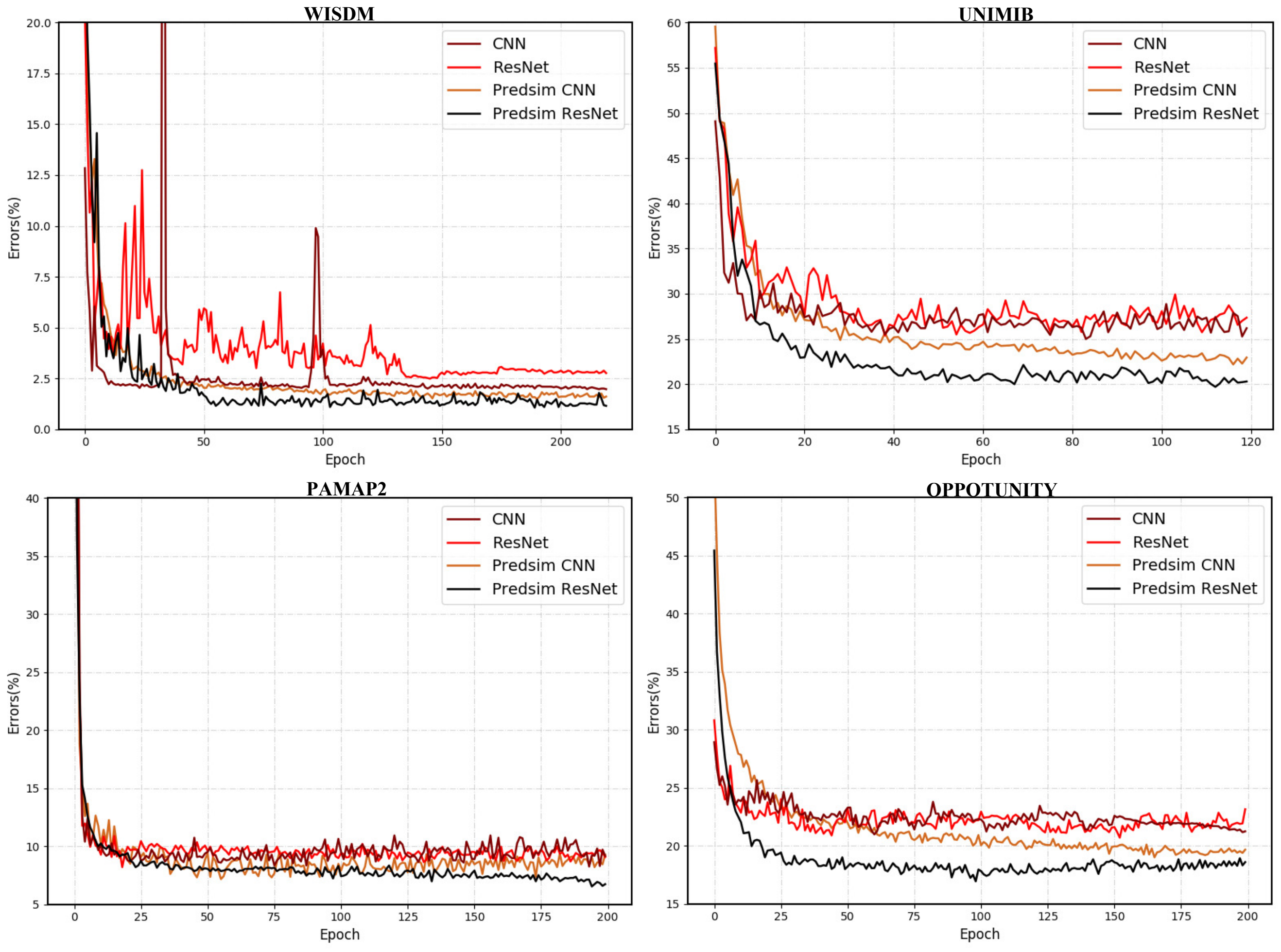


Fig. 3: Test errors of different baselines on four datasets

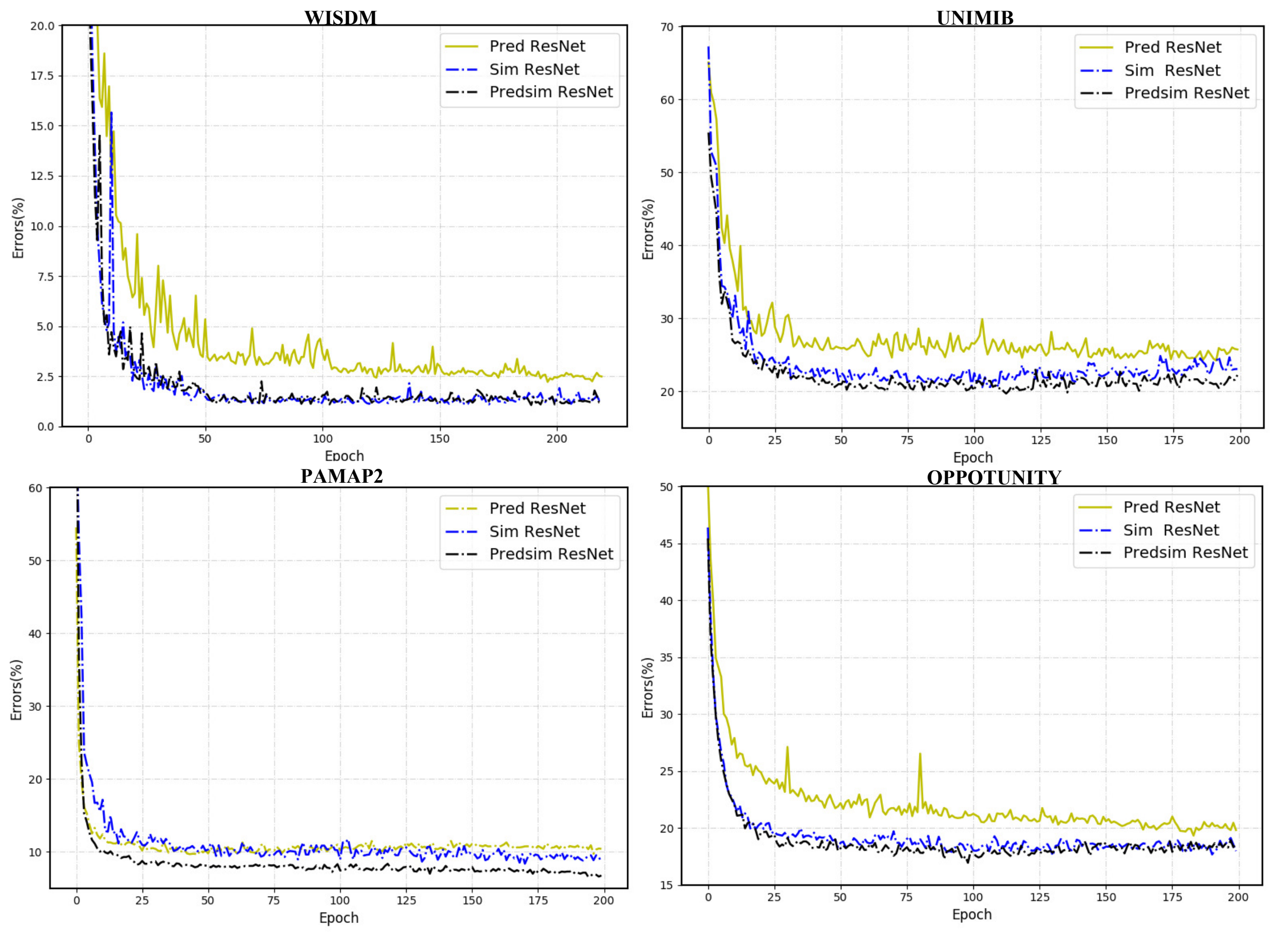


Fig. 4: Test errors with different local loss functions.

TABLE II: Classification performance on UNIMIB dataset with different stacked structure.

	ResNet-2-4	ResNet-2-6	ResNet-2-8	ResNet-2-10
ACC	77.04%	80.33%	**80.90%**	80.87%
F1	76.82%	79.89%	**80.66%**	79.69%
Par	0.1494M	0.0942M	0.0332M	0.0083M
	ResNet-3-6	ResNet-3-9	ResNet-3-12	ResNet-3-15
ACC	74.24%	76.79%	77.56%	78.03%
F1	72.72%	76.62%	77.71%	78.31%
Par	0.2241M	0.1411M	0.0581M	0.0166M

bination that are most commonly used in image recognition: ReLU before addition, ReLU-only pre-activation, and full pre-activation. According to Table III that the scheme of ReLU before addition performs best among different residual architecture, achieving 0.91% as well as 8.87% relative accuracy gain compared to the full pre-activation and ReLU-only pre-activation scheme. Therefore, it is better to employ the ReLU before addition scheme for the block-wise training ResNet.

The proposed algorithm can be performed to analyze the transition between activities via using public dataset, i.e., Smartphone-Based Recognition of Human Activities and Postural Transitions Dataset Version [30]. The dataset was collected via recruiting 30 subjects whose ages range from 19 to 48 years. All the subjects performed 12 activities: three static postures (standing, lying, sitting), three dynamic activities (walking, walking upstairs, and walking downstairs), and postural transitions occurring between the static postures which consists of: stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, and lie-to-stand. All the subjects were carrying a smartphone (Samsung Galaxy S II) on the waist. The 3-axial linear acceleration and 3-axial angular velocity were recorded at a sampling rate of 50Hz. The experiments were video-recorded to label the data manually. Because these signals are segmented with fixed-length windows of 2.56 seconds and 50% overlap, the tranistions from one activity toward another activity share the same window length and overlap with the other postures (static and dynamic). Thus, they have the same inference speed. For fair comparison, one subject was selected for test and the remaining subjects were used for training. The test errors and ROC curves are shown in Fig. 6. The results indicate that our residual model significantly improves the classification of TFilt Discrete approach (about 3.22% test error) by reducing the error down to 1.09% on the dataset by experiment creators in [30]. This is a noticeable advantage against the previous implementation.

During recent years, some researchers have demonstrated that various sensor modalities such as accelerometer and

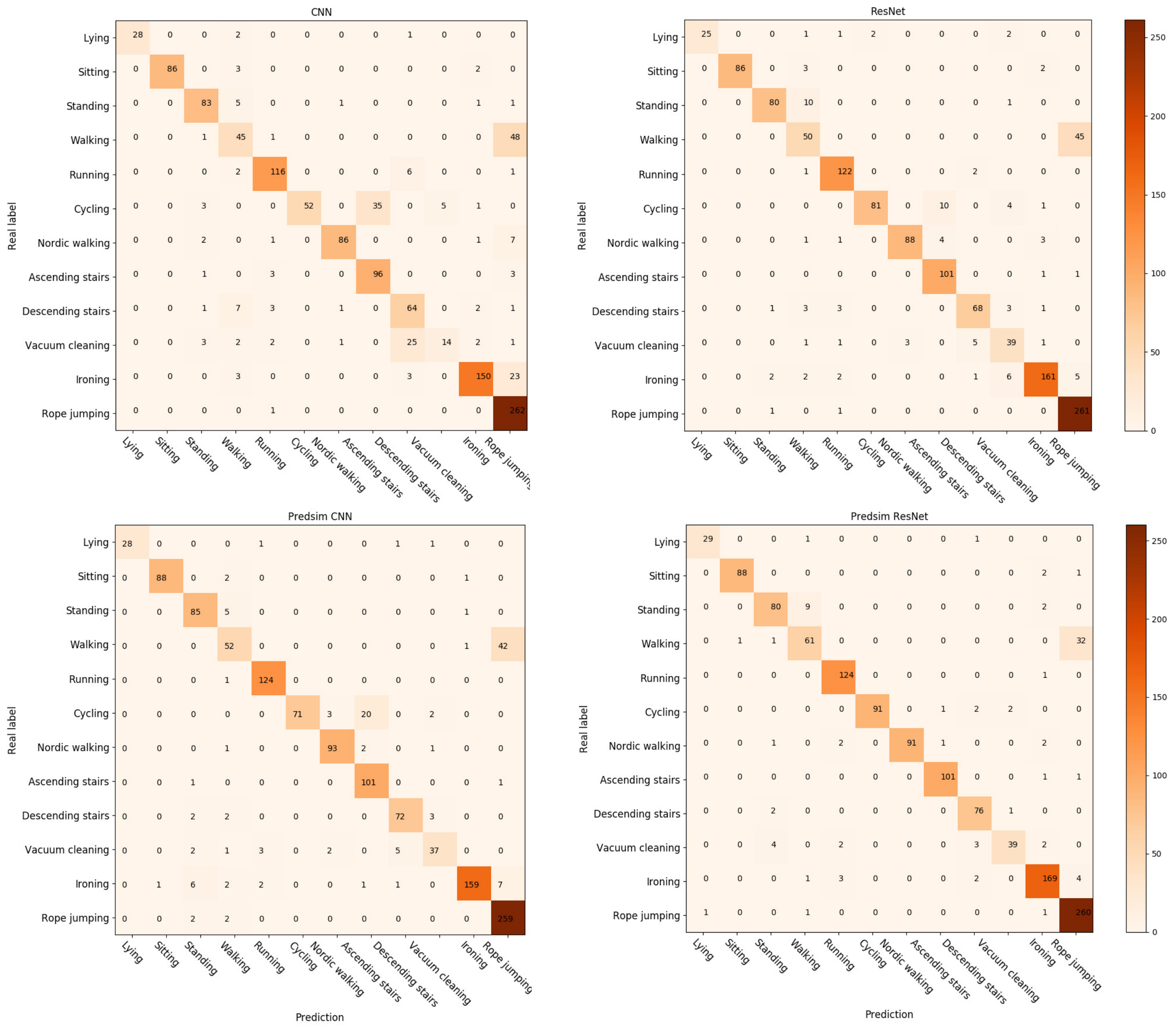


Fig. 5: (a)-(d): Confusion matrix of different models on PAMAP2 dataset

gyroscope at different human body parts play an important role in recognizing activities. For example, it is infeasible to recognize activities which involve hand gestures such as eating and smoking via using pocket position alone. We further evaluate the effect of different human body parts, i.e., which sensor position is more important for recognizing one specific activity. Without loss of generality, the PAMAP2 dataset is used in the experiment. As shown in Fig.7, 3 IMUs with different sensor modalities were placed on the arm, chest and side's ankle of each subject. It can be seen that the chest sensor (Acc1, Gyro) and the arm sensor (Acc1) that can complement each other lead to better recognition results for Rope jumping, which is interpretable according to people's common intuition. As can be seen in Fig.7, the accelerometer features may be more significant than gyroscope features. In this case. the accelerometer performs better than the gyroscope.

We perform two kinds of Jitter experiments, where Gaussian noise is added to the signals to make predictions more resistant to additive sensor noise that is common in wearable sensors. The hyper-parameters used are the same as those in training the network on WISDM dataset. In the first case, data is jittered by adding Gaussian noise of zero mean value and 6 different values of standard deviations ranging from 2ρ to 14ρ, where ρ denotes the standard deviation of original data. As shown in Fig. 8, it can be observed that adding Gaussian noise does not contribute to the improvement of classification performance; instead, it leads to slightly worse result than our "original" case. Second, in order to make the network more robust, different levels of noise (Gaussian) are added to sensor signal as data augmentation. Fig.9 shows the performance of our method, where 2x, 4x and 6x indicate that 2, 4 and 6 times more data are created by Gaussian noise augmentation. The results indicate that by adding noisy samples to training set, the performance does not increase significantly. Since a

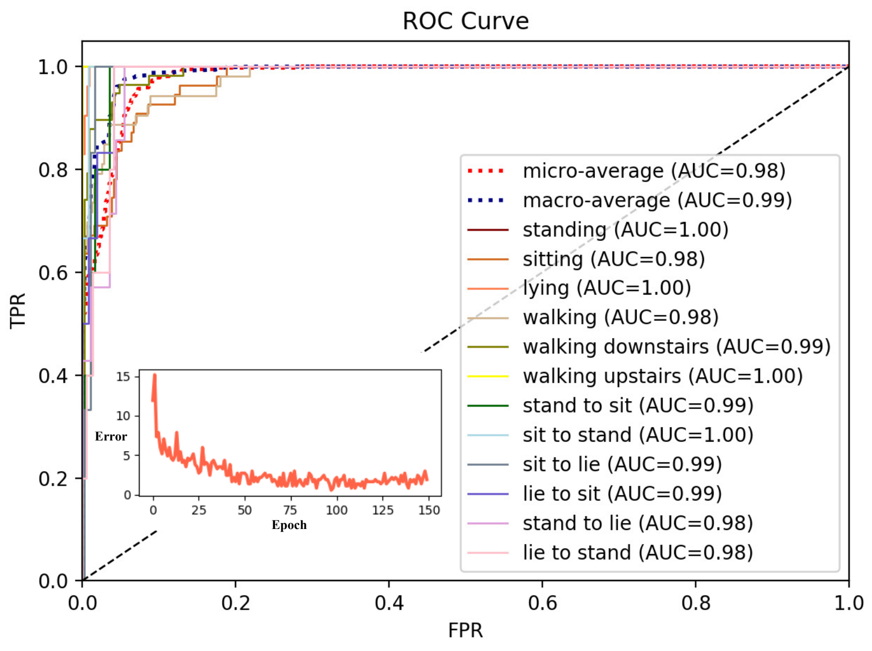


Fig. 6: : ROC curves for different kinds of activities where the subplot represents test error curve

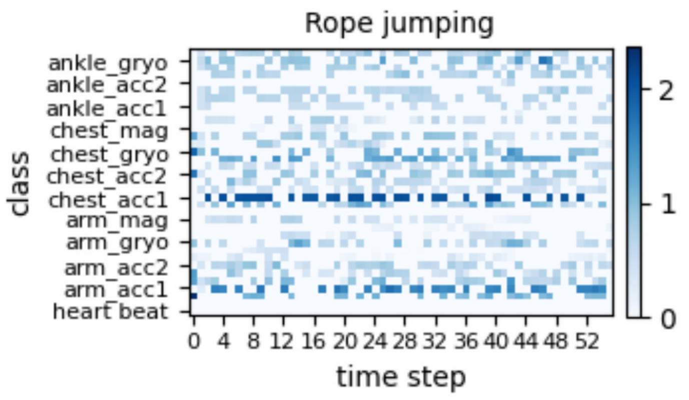


Fig. 7: Different sensors position on human body

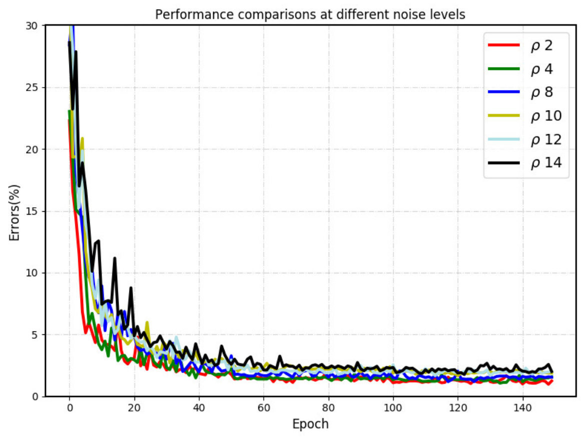


Fig. 8: Test error with different noise levels

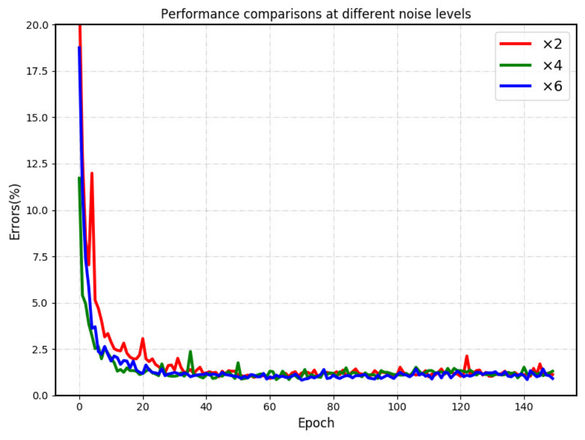


Fig. 9: Test errors with different data augmentation by adding Gaussian noise

random jitter in sensor's sample still has to preserve the coarse structure of the original data, their noise could not produce any new patterns across each class. The augmentation by this Jittering provides only slightly better results than "original case" in average, but the improvement does not have any statistical significance.

In order to analyze the computation complexity, the number of float-point operations, i.e., FLOPs have been widely used as a mainstream metric. In terms of FLOPs, the proposed algorithm is compared with baseline, i.e., ResNet. In order to evaluate the superiority of our method, we also perform the residual models with three blocks on Raspberry Pi 3 Model B+ (Fig. 10). The performance comparisons on WISDM dataset are shown in Table I. It can be seen that the computation complexity of the proposed algorithm is equal to that of baseline, which yields an almost the same inference speed on mobile devices. The advantage of our method lies in that it can obtain state-of-the-art performance at much smaller memory budget.

TABLE III: Classfication performance about the different usages of activations on UNIMIB dataset.

Activation orders	Acc	F1
Relu before addition	**80.23%**	**79.67%**
Full pre-activation	79.32%	79.11%
ReLU-only pre-activation	71.36%	71.80%

TABLE IV: Model inference time and flops for

Model	Flops	Inference Time(ms/window)
ResNet	229.91M	404ms
Predsim ResNet	228.97M	396ms

V. Conclusion

Reducing the memory consumption of storing activations would significantly improve the efficacy of the existing ResNets. Recently, the reversible RevNets have been proposed to compute each layer's activations from the next reversible layer's activations, which can provide considerable reduction in activation storage requirements at little or no cost to performance. Another approach to reducing memory is to replace backprop itself. In the paper, a new block-wise training residual model is proposed. Instead of global backprop, we use local loss functions to train each residual block independently, where the hidden activations for each layer do not need to be kept in memory anymore. Extensive experiments with analysis are conducted to compare with baselines and other state-of-the-art methods on several benchmark HAR datasets HAR datasets including OPPOTUNITY, WISDM, UNIMIB SHAR and PAMAP2. The residual model with local loss functions outperforms the equally-sized counterparts by a large margin, even though the activation storage requirements are much smaller. The experiment results demonstrate the superiority of our model. We envision our block-wise training residual block as a module which will soon enable training more powerful deep HAR networks with limited computational resources.

References

[1] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–33, 2014.

[2] 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.

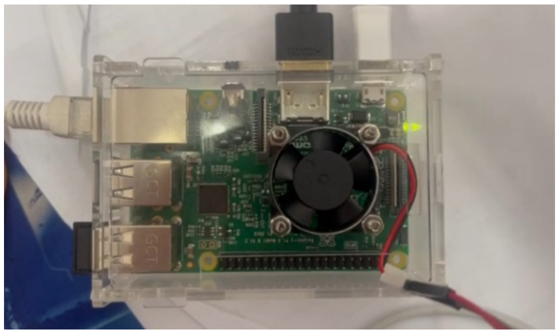


Fig. 10: Raspberry Pi 3 Model B+

[3] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, pp. 6474–6499, 2014.

[4] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 790–808, 2012.

[5] M. Prossegger and A. Bouchachia, "Multi-resident activity recognition using incremental decision trees," in *International Conference on Adaptive and Intelligent Systems*. Springer, 2014, pp. 182–191.

[6] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[7] S. C. Mukhopadhyay, "Wearable sensors for human activity monitoring: A review," *IEEE sensors journal*, vol. 15, no. 3, pp. 1321–1330, 2014.

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

[9] 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.

[10] 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.

[11] M. Jaderberg, W. M. Czarnecki, S. Osindero, O. Vinyals, A. Graves, D. Silver, and K. Kavukcuoglu, "Decoupled neural interfaces using synthetic gradients," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1627–1635.

[12] Y. Bengio, D.-H. Lee, J. Bornschein, T. Mesnard, and Z. Lin, "Towards biologically plausible deep learning," *arXiv preprint arXiv:1502.04156*, 2015.

[13] X. Xie and H. S. Seung, "Equivalence of backpropagation and contrastive hebbian learning in a layered network," *Neural computation*, vol. 15, no. 2, pp. 441–454, 2003.

[14] A. N. Gomez, M. Ren, R. Urtasun, and R. B. Grosse, "The reversible residual network: Backpropagation without storing activations," in *Advances in neural information processing systems*, 2017, pp. 2214–2224.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[16] A. Nøkland and L. H. Eidnes, "Training neural networks with local error signals," *arXiv preprint arXiv:1901.06656*, 2019.

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[18] 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*, 2020.

[19] 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.

[20] 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.

[21] 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.

[22] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *6th International Conference on Mobile Computing, Applications and Services*. IEEE, 2014, pp. 197–205.

[23] 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*, vol. 15. Citeseer, 2015, pp. 3995–4001.

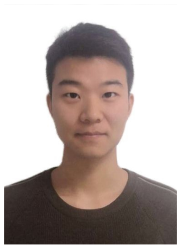
[24] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

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

[26] 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.

[27] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artificial intelligence and statistics*, 2015, pp. 562–570.

[28] E. Belilovsky, M. Eickenberg, and E. Oyallon, "Greedy layerwise learning can scale to imagenet," in *International Conference on Machine Learning*. PMLR, 2019, pp. 583–593.
[29] 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.
[30] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, 2016.
[31] D. Ravi, C. Wong, B. Lo, and G.-Z. 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)*. IEEE, 2016, pp. 71–76.
[32] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *arXiv preprint arXiv:1604.08880*, 2016.
[33] Y. Guan and T. Plötz, "Ensembles of deep lstm learners for activity recognition using wearables," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1–28, 2017.
[34] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan, "Deep activity recognition models with triaxial accelerometers," in *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
[35] 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.
[36] H. Qian, S. J. Pan, B. Da, and C. Miao, "A novel distribution-embedded neural network for sensor-based activity recognition," in *The Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

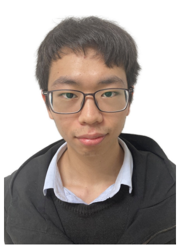


Qi Teng received the B.S. degree from Henan University of Engineering, Zhengzhou, China, in 2018. He is currently pursuing the M.S. degree with Nanjing Normal University. His research interests include activity recognition, computer vision, and machine learning.

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.

Yin Tang received the B.S. degree from Hunan University of Engineering, Xiangtan, China, in 2018. He is currently pursuing the M.S. degree with Nanjing Normal University. His research interests include activity recognition, computer vision, and machine learning.

Shuai Song is currently pursuing the B.S. degree with Department of Computer Science and Technology, Tsinghua University. His research interests include activity recognition, computer vision, and machine learning.



Xing Wang received the B.S. degree from Huaiyin Institute of Technology, Jiangsu, China, in 2020. He is currently pursuing the M.S. degree with Nanjing Normal University. His research interests include artificial intelligence, machine learning, embedded systems.

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.