Deep Dilated Convolution on Multimodality Time Series For Human Activity Recognition

Rui Xi*, Mengshu Hou*, Mingsheng Fu*, Hong Qu* and Daibo Liu*
*School of Computer Science and Engineering
University of Electronic Science and Technology of China,
Chengdu, China 611731
Email: ruix.ryan@gmail.com, mshou@uestc.edu.cn,
uestc_fumingsheng@126.com, hongqu@uestc.edu.cn, dbliu.sky@gmail.com

***Abstract*—Since Convolutional Neural Networks (CNNs) is capable of automatically learning feature representations, CNN-based recognition algorithm has been an alternative method for human activity recognition. Even though general convolution operation followed by pooling could expand the receptive fields for extracting features, it will bring about information loss in feature representation. Due to that dilated convolutions not only could expand receptive field exponentially without changing the size of field map or pooling, but it also will not cause information loss, hence, we propose D^2CL, a novel deep learning framework for human activity recognition using multi-model wearable sensors. This framework consists of dilated convolutional neural networks and recurrent neural networks. At first, learning from previous works, we add a general convolutional layer to map inputs into a hidden space for improving the capability of nonlinear representations. Subsequently, a stacked dilated convolutional networks automatically learn feature representations for inter-sensors and intra-sensors from hidden space. Then, given these learned features, two RNNs are applied to model their latent temporal dependencies. Finally, a softmax classifier at the topmost layer is utilized to recognize activities. To evaluate the performance of D^2CL on activity recognition, we select two open datasets OPPORTUNITY and PAMAP2 for training and testing. Results show that our proposed model achieves a higher classification performance than the state-of-the-art DeepConvLSTM.**

***Index Terms*—Convolutional neural networks, Dilated Convolution, Human Activity Recognition, Multimodality Time Series, Recurrent neural networks;**

I. Introduction

Human activity recognition (HAR) plays an important role in peoples daily life and has become a key problem in human-computer interaction (HCI) [1], human behavior analysis [2], mobile and ubiquitous computing [3] [4]. Thereinto, using real-time signals from multiple on-body sensors to recognize human activities is at the core of recognition problems, this is due to the following factors [5]: i) less limitations of environment constraints; ii) more accurate signal measurements; iii) without interference information from other nontarget subjects.

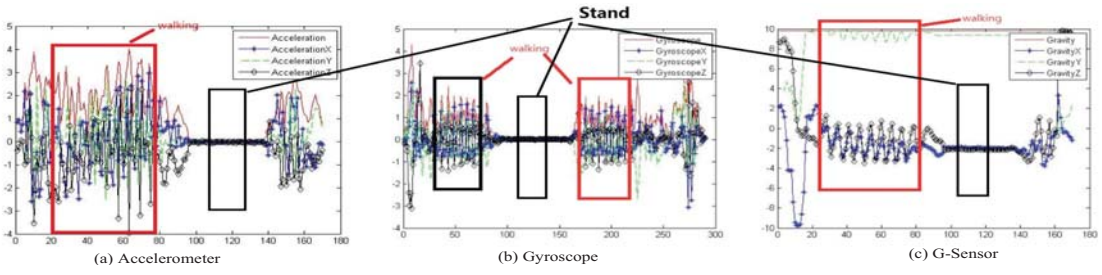
Considerable research efforts have been made to find an effective feature representation of time-series signals, but it is still an open problem of high-performance HAR system. In previous works, K-nearest neighbor (KNN) [6], support vector machines (SVM) [7], decision trees [8] and other shallow supervised machine learning algorithms were widely used to recognize human activities by building various activity models with a set of features. However, these approaches are heuristic and task-dependent. The feature extraction is totally handcrafted based on experience that involves laborious human intervention. Besides, they ignore the characteristics of HAR task: intra-class variability and similarity, and complexness of physical activities [9]. Generally speaking, some implicit features are required to recognize activities. Unfortunately, it is extremely difficult to manually extract implicit features from a mass of sensor samples.

Due to the powerful ability of learning feature representation, Convolutional Neural Network(*CNN*) has been employed for sensor-based activity recognition [10] [11] to extract latent features. In comparison to conventional shallow machine learning, *CNN* could automatically extract latent high-level features from high-dimensional data, which is scarcely possible for a handcraft method. In addition, it also provide more discriminative power under the supervision of output labels. Observing from Fig. 1, we can find that: i) for intra-sensors, these sensor datas are time-series and have inherent translation and hierarchical characteristics, ii) for local inter-sensors, these realtime signals perform some similarity and synchronicity. Unfortunately, the existed methods only consider feature representations along time-series data of individual channel in time dimension without characteristics between sensors. Even though conventional CNN can expand its receptive field by pooling to extract inter-sensor features, a following deconvolution or upsampling is required to enlarge the feature map size, this series of reduction and expansion operation will result in important feature representations loss.

In this paper, we propose D^2CL, a **D**eep **D**ilated **C**onvolutional networks[1] and **L**ong Short-Term Memory (LSTM) recurrent networks model for HAR with multimodality wearable sensors. Since adding hidden layer could significantly improve the ability of nonlinear expression [12], we add a general convolutional layer to cast its inputs into a hidden space. Then, based on the fact that dilated convolutions can exponentially expand the receptive field with no loss of resolution or coverage (see Section III-A), we

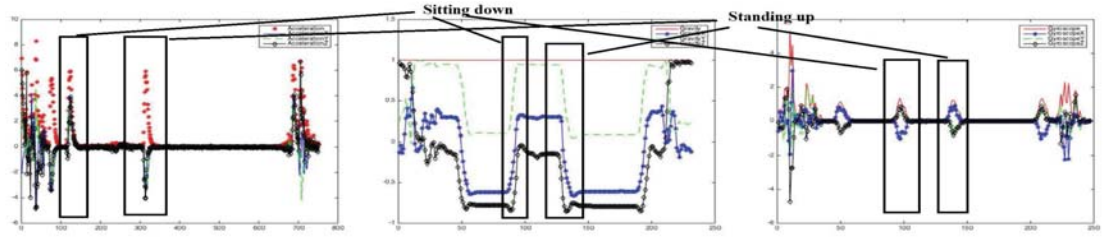
[1]Dilated convolutional network refers to a convolutional network that the convolution operation is using a dilated filter, and as explained in Section III-A, its implementation does not involve construction of dilated filter

978-1-5090-6014-6/18/$31.00 ©2018 IEEE



(a) Accelerometer (b) Gyroscope (c) G-Sensor

(a) walking



(a) Accelerometer (b) G-Sensor (c) Gyroscope

(b) Sitting down & standing up

Fig. 1: Time-series data sampled from accelerometer, gyroscope and G-sensor for some human activities, $e.g.$, walking, sitting down and standing up.

adopt stacked dilated convolutional neural network for feature representation extraction and multi-scale context aggregation. In this paper, we stack three dilated convolutional neural networks with different dilation factors following the general convolutional layer. Subsequently, two LSTM layers are applied for modeling intrinsic time dependencies of the learned features by dilated convolutional layers. Finally, a dense layer with softmax function will yield a class probability distribution. Through minimal preprocessing like normalization and engineering bias minimization, the raw measurements are split into a series of data segments with the same interval. This step is also referred to as data segmentation. For data segmentation, the true label of a matrix instance is determined by the most-frequently happened label for r raw records. To evaluate D^2CL's performance, we conduct extensive evaluations on two open human activity datasets, OPPORTUNITY and PAMAP2. Results show that D^2CL performs a very competitive efficiency of HAR, it preforms better than the current state-of-the-art DeepConvLSTM by 0.5% weighted F_1 score. Furthermore, in terms of time costs and computational workloads, this proposed model is approximately 2 times less than DeepConvLSTM.

The contributions of this paper are the following:

- For HAR, based on characteristics of inter-sensors and intra-sensors, we propose a new method to automatically extract feature representations using dilated convolution, exponentially expands receptive fields without losing resolution or coverage and aggregates multi-scale context.
- We design D^2CL, a deep framework of recognizing human activities. It consists two main modules: dilated convolutional networks automatically extract high-level features from multimodality time series, and LSTM recurrent networks model temporal dependencies between extracted features.
- We show that D^2CL outperforms other existed methods on the OPPORTUNITY challenge, including a deep scheme achieved the state-of-the-art results.

The rest of this paper is organized as follows. Section II gives a brief overview of related works on HAR. Before giving a detailed description of our proposed framework in Section III, we firstly introduce dilated convolutional network. In Section IV, we briefly describe two performance metrics and introduce some experimental settings. Then we make a comparative analysis of D^2CL and some state-of-the-art HAR systems in Section V. In the end, we conclude our work in Section VI.

II. Related Works for HAR

Majority of traditional sensor-based HAR systems manually extract a set of features from time-series sensor signals, then map them to various human activities. Subsequently, a shallow supervised machine learning algorithm is applied to recognize activities, and the most popular learning algorithms include decision tree [8], KNN [6] [13], SVM [7] [14]. For example, Anguita et al. [14] extract 561 features from accelerometer and gyroscope, and apply a multi-class SVM to classify six different activities. However, a common drawback of these systems is that the features are handcrafted and empirical.

In contrast to shallow supervised machine learning algorithm, deep learning could decompose a large and complex problem into some small and simple ones. Moreover, what the most important is that it provides a powerful ability of latent

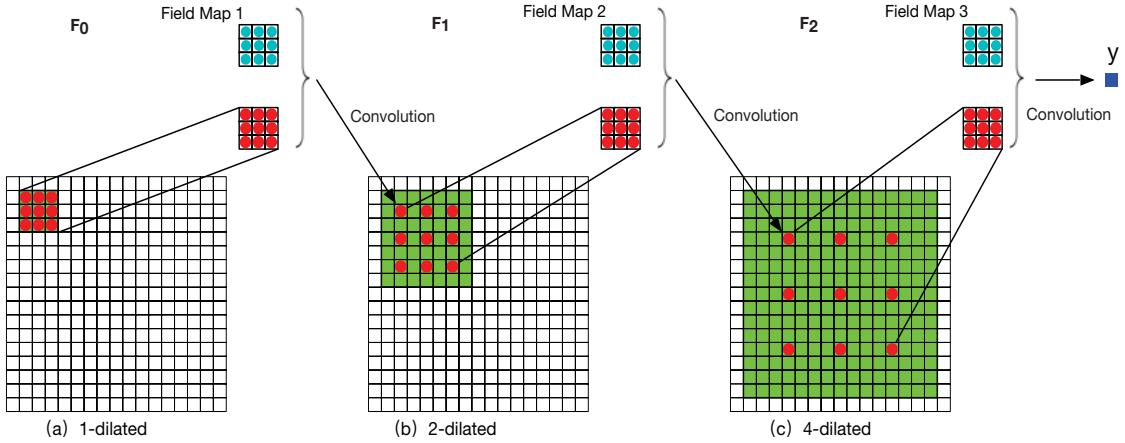


Fig. 2: Illustration of the receptive field in a dilated convolutional neural networks. Here, it consists 3 dilated convolutional neural network layers F_0, F_1, F_2, and the dilation factors are 1, 2, 4, respectively. As figure shows, the input of F_0 is a 17×17 matrix, and the field maps ($e.g.$ Field Map 1, Field Map 2, Field Map 3 in the graph) have the same size, a 3×3 matrix. (a) Conducting convolution operation on the input while the dilation factor is 1, thus each element of F_0 has a receptive field of 1×1; (b) The output of F_0 is fed into its following layer F_1, and each element is a result of a 3×3 receptive field and a 3×3 field map (Field Map 1). (c) Each element of F_2 is produced from its previous layer by a 2-dilated convolution, and the receptive size of individual element is 7×7. Increasing the dilation factor to 4, then a convolution result y has a receptive field of 15×15. Generally, the receptive field grows exponentially as the dilated factor linearly grows.

feature representation, such as specific variance of signals at different scales that reflects the salient pattern of signals. Inspired by [15], we classify deep learning approaches for HAR into three categories: generative architecture, discriminative architecture and hybrid architecture.

Generative deep architecture aims to build a model by characterizing joint distributions from the visible data and classes. The popular ones employed for HAR tasks include autoencoder and recurrent neural network. reference [17] adopts stacked autoencoder (SAE) for HAR, and firstly adopt the greedy layer-wise pre-training. However, SAE depends too much its layers and activation functions results in hard searching the optimal. For RNN, reference [18] investigates several model parameters and proposes a relative good model, which can perform HAR with high throughput.

Discriminative deep architecture offers a discriminative power to patterns via characterizing the posterior distributions of classes conditioned on visible data. It includes deep full-connected network(DFN) and convolutional neural network. For example, [16] feeds hand-engineered features from the sensors into a DFN model to recognize activities. But, DFN only serve as a classification model without automatic feature extraction, hence it cannot generalize well. In [10], CNN is firstly applied to recognize human activities. Each axis of accelerometer is seemed as one channel, then convolution and pooling are performed on individual channels. Additionally, [5] further proposed to unify and share weights in multi-sensor by using 1D convolution.

Hybrid deep architecture is consisting of discriminative models and generative models. In [19], it reveals that a combination of CNN and RNN can achieve a better performance, this is due to the reason that CNN can capture spatial relationship while RNN models temporal relationship.

III. Framework

A. Dilated Convolutional Network

Before beginning the introduction of dilated convolutional network, for a clear understanding, it is noteworthy that the notation "dilated convolution" just represents a convolution operation with a dilated filter. Usually, the dilated convolution was applied in the field of wavelet decomposition [20] [21]. Because the dilated convolution operator only use a same filter at different scales with different dilation factors, thus its implementation does not involve construction of dilated filter. Not only that, the dilated convolutional network could expand receptive field size only rely on increasing dilation factor rather than enlarging the size of field map of the network.

Objective to describe the dilated convolution mathematically, we define symbol $F : \mathbb{Z}^2 \to \mathbb{R}$ as a discrete function, and symbol $k : \Omega_r \to \mathbb{R}$ for a discrete filter whose size is $(2r+1) \times (2r+1)$. Here, we let $\Omega_r = [-r, r]^2 \cap \mathbb{Z}^2$. The discrete convolution operator $\bigotimes$ can be expressed as follows:

$$(F \bigotimes k)(x, y) = \sum_{m=-r}^{r} \sum_{n=-r}^{r} F(x-m, y-n)k(m, n). \quad (1)$$

If we denote l as a dilation factor, the l-dilated convolution operator $\bigotimes_l$ can be generalized as (2).

$$(F \bigotimes_l k)(x, y) = \sum_{m=-r}^{r} \sum_{n=-r}^{r} F(x-lm, y-ln)k(m, n). \quad (2)$$

Hence, the convolution $\bigotimes$ is also named as the 1-dilated convolution.

Now, we assume $F_0, F_1, \ldots, F_{n-1} : \mathbb{Z}^2 \to \mathbb{R}$ are discrete functions, and $k_0, k_1, \ldots, k_{n-2} : \Omega_1 \to \mathbb{R}$ are discrete 3×3 filters. In addition, we apply the filters with exponentially increasing dilation factors, such as 2^0, 2^1, $\ldots$, 2^{n-2}. Then, the discrete function F_{i+1} can expressed as (3),

$$F_{i+1} = F_i \bigotimes_{2^i} k_i \ \ for\ i = 0, 1, \ldots, n-2 \tag{3}$$

According to the definition of receptive field, the receptive size of each element in F_{i+1} is $(2^{i+2}-1) \times (2^{i+2}-1)$, which can be obtained using bottom-up deduction method. Hence, the receptive field is a square of exponentially increasing size. See from Fig. 2, the size of field map remains unchanged, we can obtain a 15×15 receptive field by successively applying dilated convolution with the dilation factor 1, 2, and 4, respectively. However, a stack of three general CNN can only achieve a 7×7 receptive field whose size performs a linear correlation with the number of CNN layer.

As we statement above, the implementation of a dilated convolutional network does not involve construction of dilated filter, hence, its convolution operation is similar to a general convolutional network. In convolutional layers, the previous layer's feature maps are convolved with several convolutional kernels, namely field map. Then, results of individual layer added by a bias are fed into an activation function to form a feature map. Assuming $v_{ij}^{x,d}$ is a value at the xth row for channel d in jth feature map of the ith layer, the value $v_{ij}^{x,d}$ can be obtained according to (4),

$$v_{ij}^{x,d} = tanh\big(b_{ij} + \sum_{m} \sum_{p=1}^{P_i - 1} \omega_{ijm}^{p} v_{(i-1)m}^{x+p,d}\big), \forall d = 1, \ldots, D \tag{4}$$

here, $tanh(\cdot)$ is a hyperbolic tangent function, namely activation function. b_{ij} is the bias for this feature map, m indexes current feature map connected to the $(i-1)$th layer, and ω_{ijm}^{p} represents a value at position p in convolutional kernel whose size is defined as P_i.

Due to dilated convolution could provide a greater receptive field under a same computational condition, when a network layer needs a larger receptive field with limited computational ability, it is possible to consider hollow convolution without increasing the size of convolutional kernels.

B. Architecture

As we mentioned in above section, dilated convolutions could exponentially expand the receptive field without loss of resolution or coverage. Based on it, we present a deep learning framework for HAR using multi-modality wearable sensors by combining dilated convolutional neural networks and LSTM neural networks. In this paper, we refer to it as D^2CL. As Fig. 3 illustrates, it consists of three building blocks: convolutional layers, recurrent layers and output layer, as detailed below.

For the first block, each convolutional layer is constituted by (i) a convolution layer, convolves its inputs with a set of kernels to be learned in the training phase; (ii) a rectified linear unit (ReLU) layer, maps convolved results by the function $relu(v) = max(v, 0)$; (iii) a normalization layer, normalizes values of different feature maps in the previous layer $v_{ij} = v_{(i-1)j}(\kappa + \alpha \sum_{t \in G(j)} v_{(i-1)t}^2)^{-\beta}$, where κ, α, β are hyper-parameters and $G(j)$ is a set of feature maps involved in the normalization. Moreover, inspired by Network In Network [12], we choose a conventional convolutional layer with 1×1 filter size as the first layer of our proposed model, so that this convolution operator could cast input into hidden space for a better capability of nonlinear representation. The following three layers are dilated convolutional layers with different dilated factors, for instance, in this paper, we successively choose 1, 2, 4.

For the second block, on the basis of previous experience [24] that a depth of at least two recurrent layers is beneficial to process sequential data, in this paper, we employ a two-layer stacked LSTM. In addition, ReLU is denoted as the activation function, we apply dropout layer to the input of LSTM layer for regularization. In addition, recurrent batch normalization to reduce internal covariance shift among time steps.

The third block is a full-connected network layer. This layer is same as a standard multilayer perceptron neural network that maps the latent features into the output classes. In this layer, through a softmax function $v_{ij} = \frac{exp(v_{(i-1)j})}{\sum_{j=1}^{C} exp(v_{(i-1)j})}$ (here, C is the number of output classes), a posterior probability of the classification results is obtained. Then, an entropy cost function is constituted based on this probabilistic results and the true labels of training instances. During the training phrase, all parameters are modulated to search the minimum cost.

However, the shorthand description of this architecture can be expressed as: $C(1) - C(64) - C(64) - C(64) - R(64) - R(64) - Sm$, where $C(F_l)$ means a convolutional layer l convolved with F_l feature maps, $R(n_l)$ is a LSTM layer contains n_l cells and Sm represents a softmax classifier. For regularization and avoiding overfitting, we apply dropout in the two later blocks of D^2CL, two LSTM layers and a dense layer with softmax classifier, in our implementation, we set the dropout probability p_{drop} to be 0.5 for all layers.

Furthermore, a sliding window strategy is employed to segment the time series signal into a collection of short pieces of signals. Specially, an instance used by the CNN is a two-dimensional matrix containing r raw samples (each sample with D attributes). Here, r is chosen to be as the sampling rate or a fixed duration, and the step size of sliding a window is chosen to be 50% overlap between adjacent windows. Therefore, the smaller step size is, the larger amount of instances we have incurring higher computation workloads. Additionally, the short piece of signals is usually labelled as the most-frequently label.

IV. Evaluation

In what follows, for eliminating influence of imbalanced class, we firstly bring in two appropriate measures to assess

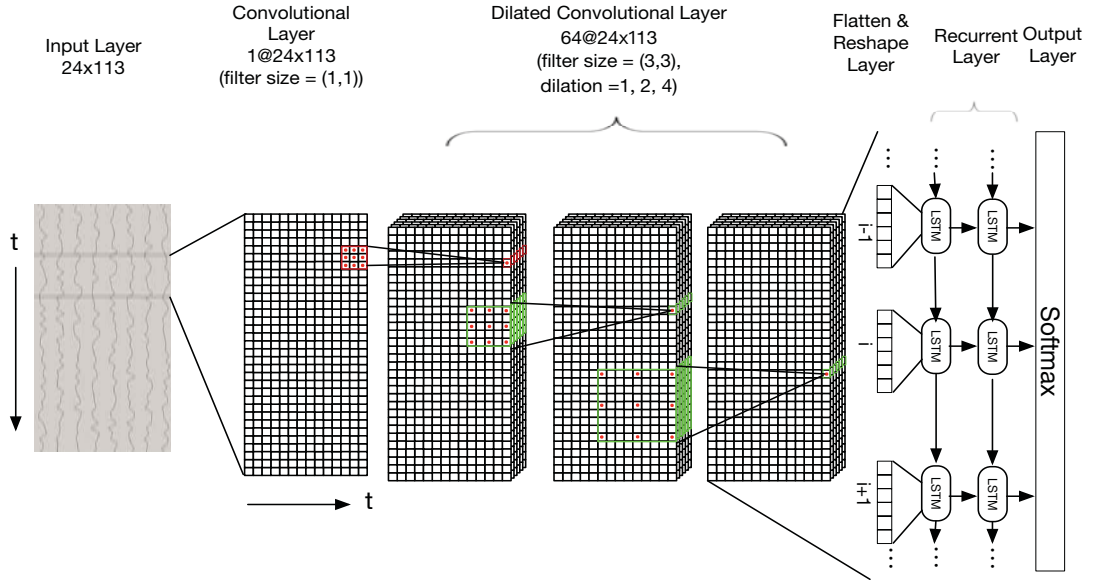


Fig. 3: Illustration of our proposed architecture for HAR on OPPORTUNITY dataset. The numbers before and after "@" refer to the number of feature maps and the dimension of a feature map in this layer. Note that we successively set dilation to 1, 2, 4 in three dilated convolutional layers. For simplicity, ReLU and normalization layer are not mentioned.

performance of complex HAR systems. Moreover, we state the parameters of our proposed model in the training.

A. Performance Metrics

Due to highly imbalanced datasets exist in continuous recording, an appropriate performance metric is needed and crucial to assess a complex activity recognition system. For example, the OPPORTUNITY dataset is extremely imbalanced and more than 75% of recording data represent $NULL$. Therefore, the overall classification accuracy could not be used to measure performance, since the majority class will achieve very high accuracy while the minority performs badly.

In our evaluation, we assess models using the weighted F_1 score, which takes into account the $precision$ and $recall$ for individual activity and performs better than accuracy, namely correct predicted / number of samples. Here, $precision$, $recall$ seperately refer as $\frac{TP}{TP+FP}$ and $\frac{TP}{TP+FN}$. Moreover, to counter the class imbalance, we bring in the sample proportion w to compute the weighted F_1 value as follows,

$$F_w = \sum_c 2 * w_c \frac{precision_c \times recall_c}{precision_c + recall_c} \tag{5}$$

where c, w_c are the class index and the proportion of samples of class c. Furthermore, $w_c = \frac{n_c}{N}$ where n_c is the amount of class c and N represent the total number of samples.

Besides, we also compute another performance metric, the mean F_1 score. It is independent of the class distribution and could be computed using the following equation,

$$F_m = \frac{2}{|c|} \sum_c \frac{precision_c \times recall_c}{precision_c + recall_c} \tag{6}$$

B. Model Training

In our work, we use python with lasagne library to implement D^2CL. Then, we train this model in a fully-supervised way, and back-propagate gradients from the softmax layer. For a higher efficiency, the collected sensor signals are segmented into mini-batches with a size of 100 during training and testing. Based this configuration, an accumulated gradient for the parameters is computed after each mini-batch. Meanwhile, we use $categorical\ cross-entropy$ function to compute the loss between predictions and targets and add it with additional $l2$ penalty terms. In our training, we set $l2$ penalty to 0.001. In the initial phase of training, all weights and biases are randomly orthogonally initialized, and are updated using RMSProp update rule with a initial learning rate of $10e^{-4}$ and a decay factor rho = 0.9. In addition, we iteratively train this model until no increase in performance for 10 epochs successively. We follow the rules of thumb shown in [27] to choose other parameters, but it is still an open issue to find the optimal parameters.

V. Results

A. Classification Performance

In contrast, we summarize the quantitative comparison on two public datasets in terms of the weighted F_1 score (F_w) and the mean F_1 score (F_m). Results of D^2CL and baselines mentioned above are shown in TABLE I, and for each performance metric, we highlight the best score in bold. From the results, we could intuitively see that the peak performance of individual method varies hugely that there is a more than 10% F_m score difference between the best and the worst

TABLE I: Best results obtained for each model and dataset along with some baselines for comparison. D^2C is a modified D^2CL which only consists of 3 dilated convolutional layers without any LSTM recurrent network.

	PAMAP2	OPPORTUNITY	
Performance	F_m	F_m	F_w
CNN [22]	**0.937**	0.591	0.894
LSTM-F [22]	0.929	0.672	0.908
DeepConvLSTM [19]	-	0.704	0.915
D^2C	0.9271	0.6604	0.9076
D^2CL	0.932	**0.7107**	**0.9197**

methods on OPPORTUNITY. However, our proposed model outperforms other recognition models with 71.07% mean f1-score and by a considerable margin of 0.5% weighted F_1 score with 91.97%. On PAMAP2, a three-layered CNN architecture achieves the best F_m score performance by 93.7%, while our model D^2CL performs a little bit less at 93.2%.

It is noteworthy that [22] proposed a deep learning architecture using bidirectional recurrent network, and it achieves the best performance on Opportunity dataset by 92.7% weighted F_1 score. But, for bi-directional LSTM, at any time-step t, both the past and future context are required to interpret the input for activity recognition, it is not suitable for online analysis. Therefore, we don't make a comparative analysis between b-LSTM-S and D^2CL in this paper.

Besides, we compute the weighted F1 score for each gesture in order to reveal the relationship between recognition performance and magnitude of training dataset. As the dotted line plots in Fig. 4, we can see that the imbalance problem of training data is very serious, for training dataset, $NULL$ activity is almost 70% while other classes are rarely more than 2% (only class 17 reaches at the proportion as few as 6.9%). This result is in accordance with above mentioned fact that most of signals are not corresponding to interesting activities in HAR. However, due to this imbalance problem, there is a significant difference in the weighted F1 score between these gesture classes. In Fig. 4, as the solid line plots, the best performance is above 95%, achieved by class 1(namely $NULL$ activity which has the largest proportion), while the worst performance is below 40%. As we all know, the more data we train, the better feature presentation our proposed model can learn, and the higher classification performance we can achieve. But surprisedly, even though the training datasets of class 3 and class 5 are very small (both under 2%), they still achieve a considerable accuracy performance by 85% above, which is far better than other classes which also have little proportion. This phenomenon proves to us that deep learning method for human activity recognition is feasible even without a large amount of dataset for training.

B. Efficiency

Due to collected time-series samples from body-worn sensors are always multi-variate and have relatively high spatial and temporal resolution ($e.g.$ 20Hz - 200Hz), hence, it is a time-critical problem to recognize human activity . For real-time recognition, we make a analysis of recognition efficiency in comparison with DeepConvLSTM from two aspects: the size of parameters and recognition time per activity.

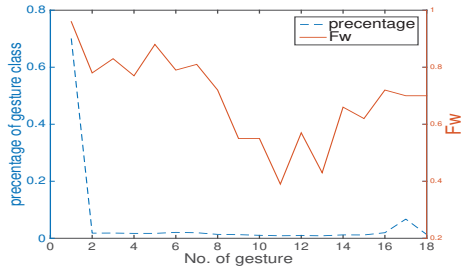


Fig. 4: Performance of D^2CL on different gesture class on OPPORTUNITY. The solid line represents the weighted F1 score for every class, and the dotted line represents the percentage of each class in dataset OPPORTUNITY. Among, the horizontal axis represents No. of gesture. the left vertical axis represents ratio of the amount of per gesture class to the amount of all gesture classes in training dataset. The right vertical axis represents the weighted F_1 score performance.

At first, as TABLE II lists, we present the number and size per parameter and layer for DeepConvLSTM and D^2CL. Here, all these two architectures have 8 layers, including one input layer, four convolutional layers, two recurrent layers and a single output layer. The shorthand description of overall architecture is $I-C(f_1)-C(f_2)-C(f_3)-C(f_4)-R(c_1)-R(c_2)-Sm$, where f_1, f_2, f_3, f_4 represent the number of filter maps in individual convolutional layer, and c_1, c_2 are the number of hidden units in the layer. Due to there is no parameter in Input layer, thus, we count the number of parameters from 2nd layer in TABLE II. See from last row of table, the total number of parameters in D^2CL is $365,376 + (64 * n_c)$ that is approximately 2 times less than DeepConvLSTM. Thus, in comparison to DeepConvLSTM, our proposed model has a much lower computational complexity.

Moreover, we are going to look forward to analyzing the time efficiency for recognizing human activities. Reference from [22], for both two models in the evaluation, we select ADL4 and ADL5 from subject 2 and 3 in OPPORTUNITY dataset as our testing set, which has approximate 10k frames that each frame corresponds to one activity sample. Results reveal that D^2CL achieves the best performance that its recognition time for per activity is only $1.355ms$, approximately as 2 times less as DeepConvLSTM. Therefore, it validates that D^2CL has a good time efficiency.

C. Impact of sliding window length

In previous statement, we present that the default sliding window length is 800ms (containing 24 samples per frame). But, If those activities whose duration is much longer, it is hard to notice time dependences from this sequence of signals. If some activities are too short, there will be more than one activity in a data segment. Therefore, in order to evaluate performance of gestures recognition with different

TABLE II: Comparison of number and size of parameters for DeepConvLSTM architecture and our proposed architecture. As [19] presents, the architecture of DeepConvLSTM can be described as $C(64)-C(64)-C(64)-C(64)-R(128)-R(128)-Sm$. The final number of parameters depends on the number of classes in classification task, denoted as n_c.

Layer	DeepConvLSTM		Our proposed model	
	Size Per Parameter	Size Per Layer	Size Per Parameter	Size Per Layer
2	K: 64*5 b: 64	384	K: 1*1*1 b: 1	2
3-5	K: 64*64*5 b: 64	20,544	K: 64*64*3*3 b: 64	36,928
6	$W_{ai}, W_{af}, W_{ac}, W_{ao}$: 4928*128 $W_{hi}, W_{hf}, W_{hc}, W_{ho}$:128*128 b_i, b_f, b_c, b_o:128 W_{ci}, W_{cf}, W_{co}: 128 c: 128 h: 128	647,680	$W_{ai}, W_{af}, W_{ac}, W_{ao}$: 4928*64 $W_{hi}, W_{hf}, W_{hc}, W_{ho}$:64*64 b_i, b_f, b_c, b_o:64 W_{ci}, W_{cf}, W_{co}: 64 c: 64 h: 64	319,872
7	$W_{ai}, W_{af}, W_{ac}, W_{ao}$: 128*128 $W_{hi}, W_{hf}, W_{hc}, W_{ho}$:128*128 b_i, b_f, b_c, b_o:128 W_{ci}, W_{cf}, W_{co}: 128 c: 128 h: 128	33,280	$W_{ai}, W_{af}, W_{ac}, W_{ao}$: 64*64 $W_{hi}, W_{hf}, W_{hc}, W_{ho}$: 64*64 b_i, b_f, b_c, b_o: 64 W_{ci}, W_{cf}, W_{co}: 64 c: 64 h: 64	8,448
8	W: 128*n_c b: nc	(128*n_c) + n_c	W: 64*n_c b: nc	(64*n_c) + n_c
Total	**996,800** + (128*n_c) + n_c		**365,250** + (64* n_c) + n_c	

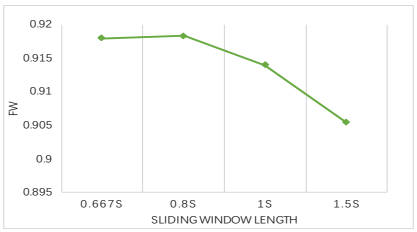


Fig. 5: Results of D^2CL on OPPORTUNITY with different sliding window length. each point represents the weighted F1 score for sliding window length of 667 ms, 800 ms, 1000 ms and 1500 ms, repectively.

duration, namely sliding windows time, in particular while the gestures are significantly longer or shorter than the sequence length, besides 800ms, we carried out experiments with data sequences segments at a duration of 667ms, 1000ms, 1500ms.

Fig. 5 plots the best performance of D^2CL on OPPORTUNITY dataset with different siding window length. From this figure, we can observe that D^2CL achieves the best F_w score performance of 91.97% when a data sequences segment has a duration of 800ms. As the length increases to 1000ms and 1500ms, the F_w performance is getting much worse, especially, in the case of 1500ms, F_w score severely decrease to below 91%, at score of 90.54%. This phenomenon reveals that the sliding window length has great impact on recognition performance of HAR method. If the length is too large, more than one short activity or gesture will be included in a data segment. Too small makes a data segment only cover part of signals of a longer activity or gesture.

In Fig. 6, we display D^2CL's performance (F_w) of recognizing individual gestures in the dataset OPPORTUNITY as the sliding window length varies. As illustrated in this graph, for majority of gestures, there is no significantly changes in recognition performance while we change the length from 667ms to 1500ms. But, for the shorter gestures, such as "Open Drawer 1", "Close Drawer 1", even though their duration is shorter than the siding window length and can be completely covered in a data sequence segment, there is still a steep change in recognition performance. Taking "Close Drawer 2" for instance, when we increase the length to 1500ms, its F_1 score drops to 30% by a decrease of 20%. For some gestures that occupy more than one sequence that D^2CL could not extract all features only with a partial view of gesture, even so, results show that D^2CL can still obtain good performance.

VI. Conclusion

In this paper, we present a new deep learning framework for human activity recognition, D^2CL. Firstly, we apply a general convolutional layer to cast inputs to a hidden space for a better capability of nonlinear representation. Then, in order to automatically extract more features of inter-sensors and intra-sensors, a stacked of three dilated convolutional layers is applied in the hidden space for the reason that dilated convolutions support exponential expansion of the receptive field without loss of resolution or coverage. Subsequently, given these extracted features, two LSTMs are able to capture latent time dependencies among features. At last, on the top of D^2CL, we use a dense layer with softmax function for classifying human activities. We evaluate the recognition performance of D^2CL in two published datasets: OPPORTUNITY and PAMAP2. For OPPORTUNITY dataset, our proposed model achieves 91.97% weighted F_1 score which outperforms than the state-of-the-art method, DeepConvLSTM. Moreover, in terms of time efficiency and computational workload, this model is approximately 2 times less than DeepConvLSTM. However, massive amounts of data for training a deep model is a major drawback and has a great influence on accuracy, performance and versatility. Unfortunately, acquiring high-volume data is

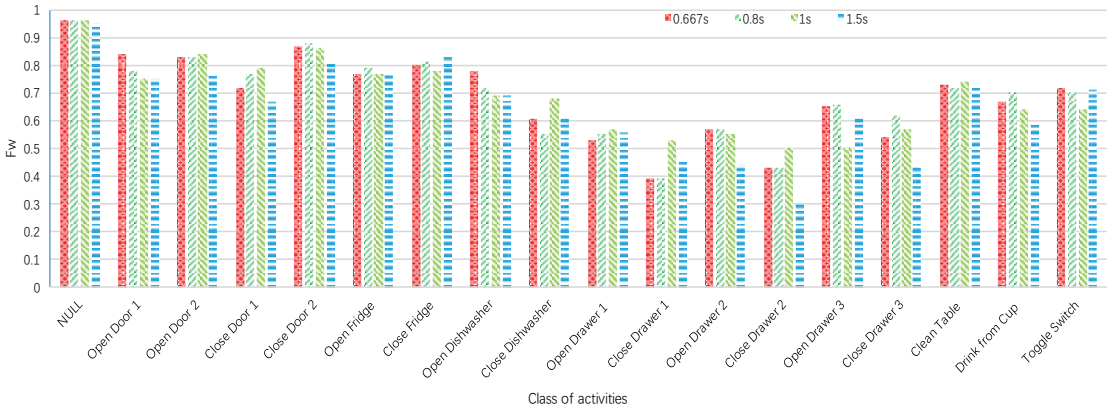


Fig. 6: Results of D^2CL on individual gestures for different sliding window length, OPPORTUNITY is used as training dataset and testing dataset. Experiments carried out with sequence length of 667 ms, 800 ms, 1000 ms and 1500 ms.

a complex and expensive process and often results in data imbalance. As we knows, for a serious imbalanced dataset, model will achieve a good coverage in the majority examples. Contrarily, for the minorities, model will probably have a higher misclassification.

ACKNOWLEDGMENT

The research leading to these results has received funding from NSF China Projects No. 61472067. We are grateful to the anonymous shepherd and reviewers whose comments helped bring the paper to its final form.

REFERENCES

[1] A. Jaimes, N. Sebe, "Multimodal humancomputer interaction: A survey". In *Computer vision and image understanding*, vol. 108, no. 1, 2007.

[2] M. Pantic, P. Alex, N. Anton, and T. S. Huang. "Human computing and machine understanding of human behavior: A survey." In *Artifical Intelligence for Human Computing*, 2007.

[3] D. Anguita, et al., "A public domain dataset for human activity recognition using smartphones", In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2013.

[4] M. Zhang and A. Sawchuk, "Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors", In *ACM Conference on Ubiquitous Computing*, 2012.

[5] J. Yang, M. Nguyen, P. San, et al. "Deep Convolutional Neural Networks On Multichannel Time Series For Human Activity Recognition", In *The 24th International Joint Conference on Artificial Intelligence*, 2015

[6] K. Kunze and P. Lukowicz. "Dealing with sensor displacement in motion-based onbody activity recognition systems". In *Proceedings of UbiComp*. 2008.

[7] A. Bulling and D. Roggen. "Recognition of visualmemory recall processes using eyemovement analysis". In *Proceedings of the 13th International Conference on Ubiquitous Computing*, 2011.

[8] T. van Kasteren, A. Noulas, G. Englebienne, et al. "Accurate activity recognition in a home setting". In *Proceedings of UbiComp*, 2008.

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

[10] M. Zeng, L. Nguyen, B. Yu, et al. Convolutional neural networks for human activity recognition using mobile sensors. In *International Conference on Mobile Computing, Applications, and Services*, 2014.

[11] W. Jiang, Z. Jin. "Human Activity Recognition using Wearable Sensors by Deep Convolutional Neural Networks" In *MM*, 2015

[12] M. Lin, Q. Chen, and S. Yan. "Network In Network." In *Proceedings of the 2nd international conference on learning representations*, 2014.

[13] S. Hasan, M. Masnad, H. Mahmud, and et al. "Human Activity Recognition using Smartphone Sensors with Context Filtering". In *Proceedings of the Ninth International Conference on Advances in Computer-Human Interactions*, 2016

[14] D. Anguita, A. Ghio, L. Oneto, et al. "A Public Domain Dataset for Human Activity Recognition using Smartphones." In *Proceedings of the 21th annual European Symposium on Artificial Neural Networks*, 2013.

[15] J. Wang, Y. Chen, S. Hao, et al. "Deep Learning for Sensor-based Activity Recognition: A Survey". In *Pattern Recognition Letters*, 2017.

[16] P. Vepakomma, D. De, S.K. Das, et al. "A-wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities". In *12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2015

[17] B. Almaslukh, J. AlMuhtadi, A. Artoli, "An effective deep autoencoder approach for online smartphone-based human activity recognition". In *International Journal of Computer Science and Network Security*, 2017.

[18] M. Inoue, S. Inoue, T. Nishida, "Deep recurrent neural network for mobile human activity recognition with high throughput". *arXiv:1611.03607*, 2016.

[19] F. J. Ordonez, & D. Roggen. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. In *Sensors*, vol. 16, no. 1, 2016.

[20] M. Holschneider, R. Kronland-Martinet, J. Morlet, et al. "A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform", In *Wavelet: Time-Frequency Methods and Phase Space, Proceedings of the International Conference*, 1987.

[21] M. J. Shensa. "The discrete wavelet transform: wedding the à trous and Mallat algorithms." In *IEEE Transactions on Signal Processing*, vol. 40, no.10, 1992

[22] N Y Hammerla, S Halloran, T Plotz, et al. Deep, convolutional, and recurrent models for human activity recognition using wearables. In *international joint conference on artificial intelligence*, 2016.

[23] F. Yu and V. Koltun. "Multi-scale context aggregation by dilated convolutions." *arXiv:1511.07122*, 2016.

[24] A. Karpathy, J. Johnson, F. F. Li. "Visualizing and understanding recurrent network", *arXiv:1506.02078*, 2015,

[25] D. Roggen, A. Calatroni, M. Rossi, et al. "Collecting complex activity data sets in highly rich networked sensor environments." In *Proceedings of the 27th IEEE International Conference on Networked Sensing Systems(INSS)*, 2010.

[26] A. Reiss, D. Stricker. "Introducing a New Benchmarked Dataset for Activity Monitoring." In *Proceedings of the 16th International Symposium on Wearable Computer*. 2012

[27] Y. LeCun, L. Bottou, G. Orr, and et al, "Efficient backprop" In *Neural Networks: Tricks of the Trade*, 1998.