2020 7th NAFOSTED Conference on Information and Computer Science (NICS)

Systematic Evaluation of Deep Learning Models for Human Activity Recognition Using Accelerometer

Thu-Hien Le*, Quang-Huy Tran †, Thi-Lan Le‡
* School of Electrical Engineering
Hanoi University of Science and Technology, Hanoi, Vietnam
Email: hien.lt173846@sis.hust.edu.vn
† VNPT Hanoi, Vietnam.
‡School of Electronics and Telecommunications,
Hanoi University of Science and Technology, Hanoi, Vietnam
Email: lan.lethi1@hust.edu.vn

2020 7th NAFOSTED Conference on Information and Computer Science (NICS) | 978-0-7381-0553-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/NICS51282.2020.9335853

***Abstract*—Human Activity Recognition (HAR) based on data from wearable sensors has become an attractive research topic thanks to its applications in different fields such as healthcare and smart environments. Recently, the advancement of deep learning with capability to perform automatically high-level feature extraction has achieved promising results. However, the performance of the deep learning models depends deeply on the characteristics of the datasets such as the number of classes, the inter-similarity and intra-variation. Therefore, directly comparing these models has become difficult since a wide variety of experimental protocols, evaluation metrics, and datasets are employed. In this paper, for the first time, a systematic evaluation of several deep learning models for HAR from wearable sensors is provided. In particular, three models named Convolutional Neural Network (CNN) [1], DeepConvLSTM - a combination of CNN and Long Short Term Memory (LSTM) [2], and SensCapsNet - a Capsule Neural Network for wearable sensor-based HAR [3] were implemented and evaluated on three benchmark datasets that are 19NonSens, CMDFall, and UCI-HAR dataset. Moreover, to have an intuitive explanation of deep learning models, a visualization of features learnt from these models is given. The evaluation codebase and results will be made publicly available for community use.**

***Index Terms*—Human Activity Recognition; Capsule Network; 19NonSens Dataset**

I. INTRODUCTION

Human Activity Recognition (HAR) is a research area which concentrates on automatically detecting/assessing what a particular human user is doing based on related sensor data. Recognizing activities of the user provides valuable contextual information to help user-centered applications to have better adaptation for the user demands in many different areas. In fact, HAR has been widely applied to several areas such as sport training, remote health monitoring, health self-management, military applications, gaming, home behavior analysis, gait analysis and gesture recognition [4], [5]. Based on the granularity of the activity being recognized, activities could be divided into movements or gestures or grouped together into sequences of activities (complex or composed activities). Moreover, in term of used sensors, wearable sensors have been utilized more prevalent in daily life, especially HAR field. These mentioned sensors are attached to objects which are handled by users in the environment (such as smart-phones, smart-watches and smart-shoes). By collecting the signals from embedded sensors, the human activities would be able to be segmented, analyzed and recognized by learning signal patterns. Different machine learning based algorithms have been used over the past decades to address the HAR problems [4], [6]–[8]. However, traditional machine learning methods using one-dimensional temporal sequences usually cope with common challenges (i.e. noise, fixed-length of sliding windows, temporal correlations between the collected signals). Hence, exploiting a robust classification method that can overcome these challenges is essentially required.

In recent years, deep neural networks have made of great advance in a large number of classification tasks. Ones have indicated their feasibility for automatically extracting and representing features in a hierarchy from low-level to high-level abstractions. Deep neural networks avoid heuristic parameters of conventional hand-designed features as well as scale better for more complicated behavior-recognition tasks. Recently, the different deep learning models have been proposed for HAR from wearable data [4]. However, the performance of the deep learning models depends deeply on the characteristics of the datasets such as the number of classes, the inter-similarity and intra-variation. Moreover, directly comparing these models has become difficult since a wide variety of experimental protocols, evaluation metrics and datasets are employed.

In this paper, a systematic evaluation of different deep learning models for HAR from wearable sensors was introduced. Precisely, three models that are Convolutional Neural Network (CNN) [1], DeepConvLSTM - a combination of CNN and Long Short Term Memory (LSTM) [2] and SensCapsNet - a Capsule Neural Network for wearable sensor-based HAR [3] were implemented and evaluated on three benchmark datasets that are 19NonSens, CMDFall, and UCI-HAR dataset.

The remainder of this paper is organized as follows: Section II focuses on the deep learning models for HAR. Section III presents the experimental results. The final section is dedicated to the conclusions and future works of the paper.

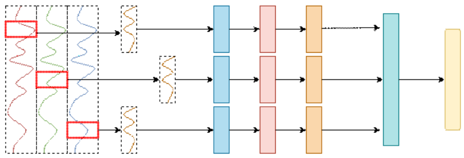

Fig. 1. Architecture of the CNN for HAR

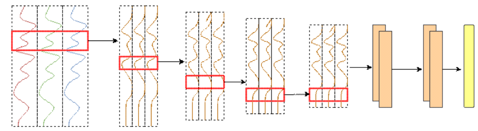

Fig. 2. Architecture of the DeepConvLSTM

II. Deep Leanring Models for Human Activity Recognition

A. Convolutional Neural Network (CNN)

Recently, neural network has been doing good with processing raw sensor signals, utilizing them to features extraction from the raw sensor signals normally shows higher performance [9]. Discovering adequate features expects considerable knowledge, which necessarily limits a systematic exploration of the feature space [10]. Convolutional neural networks (CNNs) have been proposed to address this issue [11]. A CNN with a single layer extracts features from the input signal through a convolution operation of the signal with a filter (or kernel). In a CNN, the activation of a unit represents the result of the convolution of the kernel with the input signal. By computing the activation of a unit on different regions of the same input (using a convolutional operation), it is possible to detect patterns captured by the kernels, regardless of where the pattern occurs.

Toward to a one-dimensional temporal sequence (e.g., a sensor signal), normally a 1D kernel is utilized in a temporal convolution [12]. In the 1D domain, a kernel might be considered as a filter, capable of removing outliers, filtering the data or operating as a feature detector, defined to respond maximally to specific temporal sequences within the timespan of the kernel. Generally, extracting a feature map using a one-dimensional convolution operation is given by:

$$a_i^{l,c} = b^c + \sum_{v=1}^{D} \sum_{u=1}^{k} w_{uv}^{l,c} a_{i-\frac{k}{2}+u,v}^{l-1} \forall c = 1,..,C \tag{1}$$

where b^c is the bias term of the $c-th$ output feature in the set of C output features. k is the size of kernel which slices along the times axis, $w^{l,c}$ is the weight matrix at layer l regarding the c-th output feature.

In this paper, the CNN model introduced in [1] is used.This model consists of three stages. The first stages include three modules, each of them works on the stream of a sensor. A module is a stack of four sets of four layers: a convolution layer, a rectified linear unit (ReLU) layer, a max pooling layer, and a normalization layer. The second stage unites the outcome of the three above streams by using a fully connected layer that creates a parametric-concatenation. The final stage is a fully connected layer that maps the information into classes. Fig. 1 illustrates the architecture of the CNN model for human activity recognition.

B. Deep Convolutional Long Short Term Memory (DeepConvLSTM)

The long-term, short-term memory (LSTM) method was introduced to overcome vanishing or exploding gradient [13]. In this paper, a variant of RNNs (Long Short-Term Memory) is approached and explained comprehensively. In LSTM, at each time t, i_t, f_t, o_t and $\widetilde{a}_t$ are input gate, forget gate, output gate and candidate value [14], which can be described as the following equations:

$$i_t = \sigma(W_{i,x}x_t + W_{i,h}h_{t-1} + b_i) \tag{2}$$

$$f_t = \sigma(W_{f,x}x_t + W_{f,h}h_{t-1} + b_f) \tag{3}$$

$$o_t = \sigma(W_{o,x}x_t + W_{o,h}h_{t-1} + b_o) \tag{4}$$

$$\widetilde{a}_t = \tanh(W_{\widetilde{a},x}x_t + W_{\widetilde{a},h}h_{t-1} + b_{\widetilde{a}}) \tag{5}$$

where $W_{i,x}$, $W_{i,h}$, $W_{f,x}$, $W_{f,h}$,$W_{o,x}$, $W_{o,h}$, $W_{\widetilde{a},x}$ and $W_{\widetilde{a},h}$ are weight matrices, b_i, b_f, b_o and $b_{\widetilde{a}}$ are bias vectors, x_t is the current input, h_{t-1} is the output of the LSTM at the previous time $t-1$, and σ is the Sigmoid activation function. The forget gate determines how much of prior memory value should be removed from the cell state. Similarly, the input gate specifies new input to the cell state. Then, the cell state a_t is computed as:

$$a_t = f_t \circ a_{t-1} + i_t \circ \widetilde{a}_t \tag{6}$$

where $\circ$ denotes the Hadamard product [15]. The output h_t of the LSTM at the time t is derived as:

$$h_t = o_t \circ \tanh(a_t) \tag{7}$$

Finally, we project the output h_t to the predicted output $\widehat{z}_t$ as:

$$\widehat{z}_t = W_y h_t \tag{8}$$

where W_y is a projection matrix to reduce the dimension of h_t.

The LSTM cell at current state receives a feedback h_{t-1} from the previous LSTM cell to capture the time dependencies, therefore, the information from the previous cell has been stored.

In this study, the DeepConvLSTM model presented in [2] is used. This model has four convolutional layers stacked on top of the raw sensor channel. Those layers with convolution operations extract features for stacked LSTM layers. Following the study in [2], two LSTM layers are stacked to enable the ability of modeling high level of abstraction.

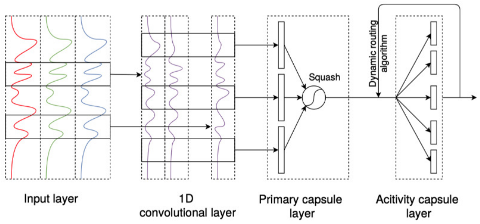

Fig. 3. Architecture of the proposed SensCapsNet for HAR [3]

C. Capsule Network (SensCapsNet)

As aforementioned deep learning models employ scalar outputs and pooling, they can get the invariance but not the equivariance. The capsule networks (CapsNet) with the vector output and routing by agreement is able to capture the equivariance. In [3], Pham et al. have proposed a method for recognizing human activity from wearable sensors based on a capsule network named SensCapsNet.

SensCapsNet a three-stage architecture containing a convolutional stage, a primary capsule stage and an activity capsule stage. The architecture of the network is presented in Fig. 3. The convolutional stage contains multiple 1D convolutional layers as presented in equation 1 and projection layers with ReLU activation function. This stage extracts abstract features for primary capsules from sensing data features. The primary capsule layer containing a large number of capsules. Each capsule encodes information into 8-dimensional vectors using a 1D convolution with a novel squash activation function [16]. The activity capsule layer contains as many numbers of capsules as the number of activities. The capsules in this stage connect densely to the capsules in primary capsule stage.

III. Experimental Results and Discussion

A. Datasets and Evaluation Metrics

Three datasets that are 19NonSens [3], CMDFall [17] and UCI-HAR [18].

19NonSense is a dataset of 19 activities (18 activities plus Null activity) collected 13 subjects using accelerometer sensor attached on e-Shoes and Samsung Gear S2 smart-watch. The dataset employs different evaluation protocol. In this paper, we follow the protocol: samples of twelve subjects are used for training and samples of the remaining subject are used for testing.

CMDFall dataset was built with 20 activities including 8 falls of different styles and 12 daily activities [17] from 50 subjects using accelerometer sensors WAX3 worn on left wrist and waist and 7 Kinect sensors. The evaluation protocol of CMDFall is as follows: training set including 25 subjects with even ID, validation set including 5 random subjects among 25 subjects with odd ID, and the remaining are utilized for testing set. The evaluation metrics used in this study are Precision, Recall and F1 score.

UCI-HAR dataset [19] is a dataset collected from 30 volunteers within an age bracket of 19-48 years. Each person performed six activities wearing a smartphone (Samsung Galaxy S II) on the waist.

B. Experimental Results

The overall results obtained when using three deep learning models for three datasets are given in Tab.I. DeepConvLSTM outperforms both CNN and SensCapsNet with F1-score of 0.89, 0.4, and 0.91 for 19NonSens, CMDFall and UCI-HAR datasets. SensCapsNet model that exploited the spatial-temporal correlation by using vectors as inputs achieved results are quite high. The F1-score of this model are 0.84, 0.38 and 0.91. It is quite interesting to see that when working with a dataset containing few activities and high number of input channels such as UCI-HAR, performance of all models are similar. This means that DeepConvLSTM and SensCapsNet with the spatial-temporal learning capacity can distinguish activities with high inter-class similarity and intra-class variation.

Table II shows the accuracy obtained for each class of 19NonSens dataset. Although DeepConvLSTM outperforms CNN and SensCapsNet on three datasets. The detailed results of each class have shown that SensCapsNet performs quite good for ”Down-stair”, ”Up-stair”, ”Hand-washing”, ”Wiping” and ”Null” activities thanks to its equivariance characteristic. Using DeepConvLSTM allows to obtaind the performance over 80% for all activity classes except ”Null”. In fact, recognizing ”Null” activity is extremely difficult since they could be any activity of the human.

Table III shows the detailled accuracy of each activity in CMDFall dataset. The recognition results of three models are very low since the activities on CMD Fall dataset are relatively similar such as ”Left fall” and ”Right fall”. ”Run slowly” and ”Jump in place” yield the highest recognition results because they are explicitly different. ”Jump in place” has F1 Score on three models are 0.66, 0.77, and 0.74 while that of ”Run slowly” are 0.71, 0.85, and 0.80 respectively with three models. When looking closely the behavior of DeepConvLSTM and SensCapsNet for recognizing different fall types, we can see that SensCapsNet can detect correctly fall activity (Precision of SensCapsNet is greater than that of DeepConvLSTM) . However, it misses more fall activity than DeepConvLSTM does (Recall of SensCapsNet is lower than that of DeepConvLSTM).

Obtained results for UCI-HAR dataset are shown in Tab. IV. Three proposed models perform good on UCI-HAR dataset and have a similar performance. In particular, three evaluation metrics of each model are approximately the same and is 0.91. Almost the results for each activity are quite high but the obtained results for ”Sitting”, ”Standing” are a little low than the others. Because these activities are partly similar to others activities. Therefore, they are easily misclassified.

C. Data visualization

To have an intuitive explanation of deep learning models, we employ t-SNE method [20] that allows visualizing high-dimensional data by giving each datapoint a location in a

TABLE I
PERFORMANCE ON 19NONSENS, CMDFALL AND UCI-HAR DATASETS

Model	19NonSens dataset			CMDFall dataset			UCI-HAR dataset		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
CNN	0.7	0.65	0.67	0.2	0.3	0.29	0.91	0.91	0.91
DeepConvLSTM	**0.89**	**0.9**	**0.89**	**0.4**	**0.42**	**0.4**	0.91	0.91	0.91
SensCapsNet	0.84	0.83	0.84	0.41	0.37	0.38	0.91	0.91	0.91

TABLE II
DETAILED PERFORMANCE FOR EACH ACTIVITY ON 19NONSENS DATASET

Activities	CNN			DeepConvLSTM			SensCapsNet		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Turning knee	0.93	0.78	0.85	0.97	0.94	0.92	0.84	0.83	0.83
Brushing	0.51	0.57	0.54	0.88	0.86	0.87	0.78	0.69	0.73
Cycling	0.84	0.88	0.86	0.94	0.93	0.93	0.88	0.87	0.88
Down-stair	0.76	0.51	0.61	0.87	0.85	0.86	0.86	0.88	0.87
Turning haunch	0.82	0.58	0.68	0.91	0.94	0.92	0.74	0.55	0.63
Kicking	0.65	0.59	0.62	0.88	0.86	0.87	0.77	0.84	0.8
Turning ankle	0.71	0.65	0.68	0.9	0.87	0.89	0.75	0.79	0.77
Mixing	0.53	0.44	0.48	0.87	0.9	0.89	0.92	0.82	0.87
Peeling	0.69	0.77	0.73	0.92	0.9	0.91	0.82	0.53	0.64
Running	0.89	0.91	0.9	0.96	0.97	0.97	0.87	0.9	0.88
Turning shoulder	0.73	0.59	0.65	0.92	0.95	0.94	0.76	0.91	0.83
Slicing	0.78	0.83	0.8	0.95	0.92	0.93	0.86	0.87	0.86
Sweeping floor	0.57	0.62	0.59	0.87	0.94	0.9	0.83	0.94	0.88
Up-stair	0.7	0.6	0.65	0.85	0.82	0.84	0.86	0.92	0.89
Walking	0.82	0.78	0.8	0.86	0.95	0.9	0.89	0.86	0.87
Hand washing	0.53	0.45	0.49	0.79	0.82	0.81	0.89	0.95	0.92
Wiping	0.47	0.4	0.43	0.75	0.89	0.82	0.89	0.98	0.94
Turning wrist	0.73	0.67	0.7	0.92	0.93	0.92	0.87	0.96	0.91
Null	0.48	0.61	0.54	0.82	0.76	0.79	0.91	0.9	0.91

TABLE III
DETAILED PERFORMANCE FOR EACH ACTIVITY ON CMDFALL DATASET

Activities	CNN			DeepConvLSTM			SensCapsNet		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Forward fall	0.09	0.10	0.10	0.21	0.33	0.26	0.28	0.21	0.24
Back fall	0.15	0.10	0.12	0.26	0.15	0.19	0.27	0.15	0.19
Left fall	0.17	0.19	0.18	0.26	0.24	0.25	0.28	0.21	0.24
Right fall	0.20	0.28	0.24	0.25	0.35	0.29	0.33	0.34	0.34
Left fall while sitting on chair	0.05	0.05	0.05	0.25	0.25	0.25	0.31	0.13	0.18
Right fall while sitting on chair	0.08	0.11	0.09	0.17	0.26	0.21	0.24	0.07	0.11
Walking in all direction	0.45	0.48	0.46	0.46	0.66	0.54	0.37	0.75	0.49
Run slowly	0.75	0.68	0.71	0.85	0.85	0.85	0.82	0.79	0.80
Stagger	0.28	0.40	0.33	0.39	0.50	0.44	0.43	0.38	0.41
Crawl	0.38	0.30	0.33	0.57	0.62	0.59	0.61	0.54	0.57
Move the chair	0.36	0.45	0.4	0.39	0.38	0.39	0.27	0.44	0.34
Move hands and knees	0.47	0.39	0.42	0.58	0.43	0.49	0.58	0.49	0.53
Crouch down to pick up things by left hand	0.32	0.47	0.38	0.47	0.43	0.45	0.69	0.48	0.57
Crouch down to pick up things by right hand	0.22	0.48	0.31	0.42	0.42	0.42	0.41	0.45	0.43
Jump in place	0.63	0.69	0.66	0.79	0.76	0.77	0.81	0.69	0.74
Sit down on a chair then stand up	0.06	0.06	0.06	0.16	0.17	0.16	0.11	0.04	0.06
Left fall while lying on a bed	0.10	0.09	0.09	0.31	0.36	0.33	0.26	0.24	0.25
Right fall while lying on a bed	0.49	0.32	0.39	0.54	0.38	0.45	0.47	0.45	0.46
Sit down on a bed then stand up	0.16	0.19	0.17	0.19	0.42	0.26	0.25	0.33	0.28
Lie on a bed then stand up	0.21	0.25	0.23	0.34	0.35	0.35	0.28	0.29	0.28

TABLE IV
DETAILED PERFORMANCE FOR EACH ACTIVITY ON UCI-HAR DATASET

Activities	CNN			DeepConvLSTM			SensCapsNet		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Walking	0.99	0.92	0.96	0.98	0.94	0.96	0.99	0.95	0.97
Walking upstairs	0.89	0.99	0.93	0.90	0.94	0.92	0.90	0.94	0.92
Walking upstairs	0.88	0.96	0.92	0.96	0.96	0.96	0.91	0.99	0.95
Sitting	0.85	0.79	0.82	0.79	0.85	0.82	0.86	0.76	0.81
Standing	0.86	0.86	0.86	0.85	0.80	0.82	0.83	0.88	0.85
Laying	1.00	0.95	0.98	1.00	1.00	1.00	1.00	0.95	0.97

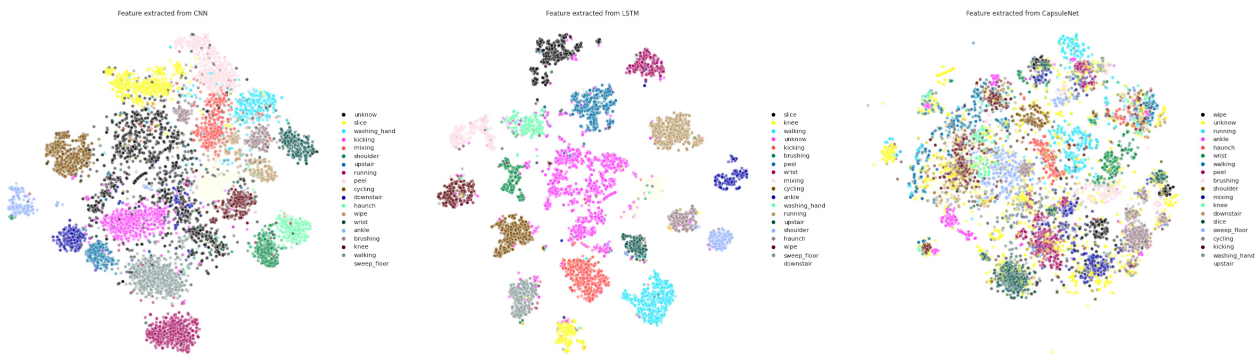

Fig. 4. Visualizations of the 19NonSens dataset for CNN, DeepConvLSTM and SensCapsNet

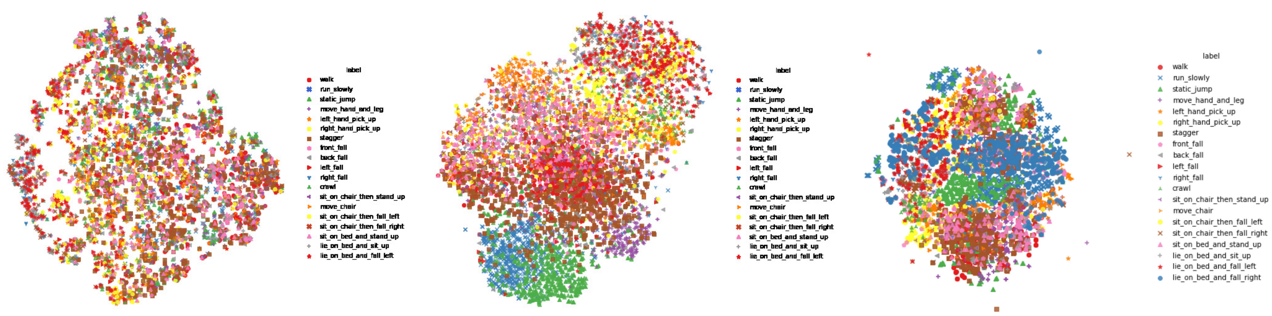

Fig. 5. Visualizations of the CMDFall dataset for CNN, DeepConvLSTM and SensCapsNet

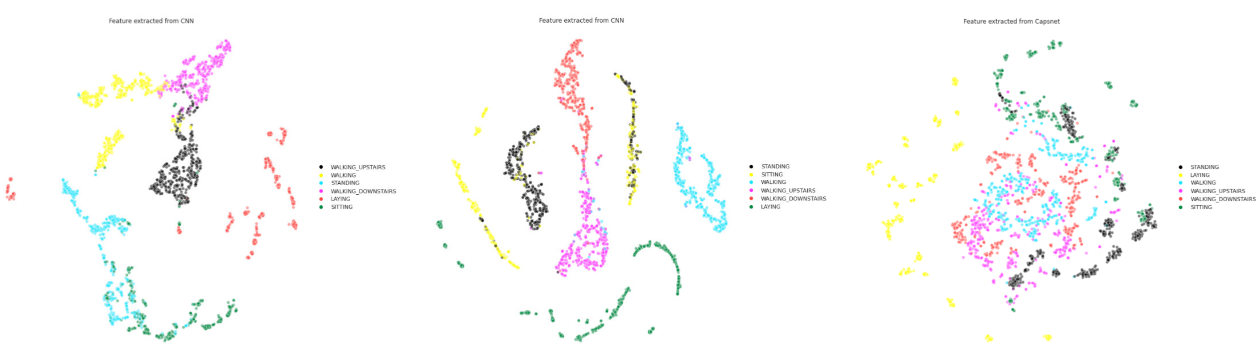

Fig. 6. Visualizations of the UCI-HAR dataset for CNN, DeepConvLSTM and SensCapsNet

two or three-dimensional map. Figures 4, 5 and 6 show the data visualization of 19NonSens, CMDFall and UCI-HAR datasets when using CNN, DeepConvLSTM and SensCapNet respectively.

We can observe that for 19NonSens dataset, DeepConvLSTM model allows clearly separating classes in the map. This explains the best performance that this model obtains for 19NonSense dataset. When working with more challenging dataset CMDFall, the visualization results of these models show that it is difficult to get a clear separation of different activity classes as there exist a large inter-class similarity and intra-class variation. We can observe an interesting phenomenon with UCI-HAR dataset. For this dataset CNN and DeepConvLSTM bring good performances and the activities are separated quite clearly. Nevertheless, some examples of ”Sitting” and ”Standing” activities are still be stacked. This might be proven since the signals obtained from the subjects are relatively similar. Visualization result with SensCapNet model is not good as others, all activities is not explicitly separated but thanks to routing algorithm SensCapNet model still achieves a good result.

IV. Conclusion

In this paper, for the first time, an systematic evaluation of deep learning model for wearable-based activity recognition is provided. The experimental results obtained with three well-known deep learning models have shown the potential of deep learning for recognizing certain types of activities from wearable sensors. However, when working with more challenging dataset such as CMDFall, the lack of context information of wearable sensors results poor recognition performance. In order to recognize the full spectrum of human activities, different modality such as images should be exploitd and combined with wearable datas as suggested in [21].

V. Acknowlegements

This research was funded by Vietnam Ministry of Science and Technology under grant number DTDLCN-16/18 ”Automated Respiration Symptoms monitoring and Abnormal Human Activity Detection Using the Internet of Things”

References

[1] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, “Deep convolutional neural networks on multichannel time series for human activity recognition,” in *IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 2015, pp. 3995–4001.

[2] F. J. O. Morales and D. Roggen, “Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition,” *Sensors*, vol. 16, no. 1, p. 115, 2016.

[3] C. Pham, S. Nguyen-Thai, H. Tran-Quang, S. Tran, H. Vu, T.-H. Tran, and T.-L. Le, “Senscapsnet: Deep neural network for non-obtrusive sensing based human activity recognition,” *IEEE Access*, vol. 8, pp. 86 934–86 946, 2020.

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

[5] A. Murad and J.-Y. Pyun, “Deep recurrent neural networks for human activity recognition,” *Sensors*, vol. 17, no. 11, p. 2556, 2017.

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

[7] 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.

[8] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. Alo, “Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges,” *Expert Systems with Applications*, vol. 105, pp. 233–261, 2018.

[9] D. Palaz, R. Collobert *et al.*, “Analysis of cnn-based speech recognition system using raw speech as input,” Idiap, Tech. Rep., 2015.

[10] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. Cardoso, “Preprocessing techniques for context recognition from accelerometer data,” *Personal and Ubiquitous Computing*, vol. 14, no. 7, pp. 645–662, 2010.

[11] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, “Deep convolutional neural networks on multichannel time series for human activity recognition,” in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[12] 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.

[13] A. Alzahrani, P. Shamsi, C. Dagli, and M. Ferdowsi, “Solar irradiance forecasting using deep neural networks,” *Procedia Computer Science*, vol. 114, pp. 304–313, 2017.

[14] T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.

[15] R. A. Horn and Z. Yang, “Rank of a hadamard product,” *Linear Algebra and its Applications*, vol. 591, pp. 87–98, 2020.

[16] S. Sabour, N. F., and G. E. Hinton, “Dynamic routing between capsules,” in *NIPS 2017, 4-9 December 2017, Long Beach, CA, USA*, 2017, pp. 3859–3869.

[17] T.-H. Tran, T.-L. Le, D.-T. Pham, V.-N. Hoang, V.-M. Khong, Q.-T. Tran, T.-S. Nguyen, and C. Pham, “A multi-modal multi-view dataset for human fall analysis and preliminary investigation on modality,” in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 1947–1952.

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

[19] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes-Ortiz, “A public domain dataset for human activity recognition using smartphones,” 01 2013.

[20] L. van der Maaten and G. Hinton, “Visualizing high-dimensional data using t-sne,” *Journal of Machine Learning Research*, vol. 9, 2008.

[21] C. Tong, S. A. Tailor, and N. D. Lane, “Are accelerometers for activity recognition a dead-end?” in *Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications*, ser. HotMobile ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 39–44. [Online]. Available: https://doi.org/10.1145/3376897.3377867