Activity Recognition using Deep Denoising Autoencoder

Mohd Halim Mohd Noor
School of Computer Sciences
Universiti Sains Malaysia
Pulau Pinang, Malaysia
halimnoor@usm.my

Mohd Anuaruddin Ahmadon
Graduate School of Sciences and Technology for Innovation
Yamaguchi University
Japan
anuar@yamaguchi-u.ac.jp

Muhammad Khusairi Osman
Fakulti Kejuruteraan Elektrik
Universiti Teknologi MARA
Pulau Pinang, Malaysia
khusairi@ppinang.uitm.edu.my

***Abstract*—Existing feature extraction method for activity recognition is time consuming and laborious and prone to error. This paper proposes an unsupervised deep learning method for feature learning in activity recognition using tri-axial accelerometer. The proposed method extracts the relevant features automatically, eliminating the needs of feature extraction and selection stages. We evaluate and compared the proposed method with the conventional method in terms of recognition accuracy on a public dataset with wide range of activities. Results have shown that the proposed method achieved a better performance, improving the recognition accuracy by 0.03.**

Keywords—Activity recognition, autoencoder, deep learning, accelerometer

I. INTRODUCTION

Activity recognition is important for large number of applications such as pervasive healthcare. In elderly healthcare, it is used to analyze the daily activities and behaviour of a person (especially elderly) to evaluate his or her general health. Physical inactivity and sedentary behaviour may accelerate biological aging and lead to poor health, including chronic diseases and frailty, which is an indicator of health status of an elderly and associated with dependency, disability and mortality. Physical activity as preventive and therapeutic measures are well-documented. Promotion of physical activity has been incorporated in routine clinical practice [1], [2]. However, broader implementation is facing multiple barriers, which one of them is in ability to accurately assess patient status and progress due to imprecise assessment tools. Currently, physical activity assessment methods mostly rely on questionnaire and self-report measures of physical activity. Such methods may induce inaccuracy and bias in the assessments due to difficulties in recalling and questionnaire's design.

Wearable technology offers a prospective solution to the increasing demand for activity monitoring. Wearable sensor-based activity recognition is a system that continuously gather body motion measurements using inertial sensors such as accelerometer and gyroscope. The collected data is then processed to recognize the physical activities such as walking, standing and sitting. In general, the activity recognition can be divided into two steps which are the feature extraction and classification. The first step is the most crucial because the performance of the classification is heavily dependent on the extracted features. For that reason, numerous studies have been carried out to determine the most salient features for activity recognition.

Features could be extracted directly from the sensor data such as mean, variance (standard deviation) and signal magnitude area [3]. Features could also be extracted from Fourier transformed signal such as spectral energy and entropy [4]. More advanced features have been proposed for activity recognition. In [5], cepstral-based features have been proposed for classifying dynamic activities such as walking, running and bicycling. The proposed methods achieved a recognition accuracy of 0.91. In [6], ensemble empirical mode decomposition has been used to decompose the sensor data before feature extraction is performed for classifying dynamic and static activities. The proposed method achieved a recognition accuracy of 0.81. Spatiotemporal features have been proposed by calculating the Euler angles and quaternions from the sensor data. The results show that the proposed method achieved a recognition accuracy of 0.85. Although the proposed features have been shown to be effective in distinguishing the activities, feature engineering or manual feature extraction is time consuming, laborious and prone to error. Furthermore, the process relies on expert knowledge to extract and select the most discriminative features.

Deep learning methods have been proposed to automatically extract features from sensor data for activity recognition. In [7], a deep learning model consists of one convolutional and one max-pooling layer has been proposed for automatic feature extraction. The proposed method achieved a recognition accuracy of 0.95. Similar work is found in [8]. The proposed method achieved a recognition accuracy of 0.94. A hybrid deep learning model using long short term memory and convolutional neural network has been proposed for activity recognition. The model can automatically extract not only the features, but also model the temporal information in the sensor data for more accurate classification. The results show that the proposed method achieved a recognition accuracy of 0.864. However, the aforementioned works are purely supervised learning methods or involve human annotation and labeling that are time consuming and laborious.

In this paper, we propose a method that utilizes a specialized deep neural network known as autoencoder to automatically extract or learn the relevant features from accelerometer sensor data for activity recognition. The approach uses a variant of autoencoder called denoising autoencoder that is capable of learning features in an unsupervised manner. As a result, the feature engineering step is eliminated, making the feature extraction more accurate and reliable.

The rest of the paper is organized as follows. The architecture of the denoising autoencoder is presented in Section II. Section III presents the experimental setup for activity recognition. Section IV presents results obtained.

Finally, Section V presents the conclusion drawn from the experimental results.

II. Proposed Methodology

The overview of the proposed method is illustrated in Fig. 1. The proposed method is divided into two phases which are unsupervised feature learning and classification. The feature learning pipeline utilizes a denoising autoencoder to automatically extract relevant feature representation of the data (on the left). The autoencoder is then cascaded with a softmax classifier to classify the activities (on the right). The proposed method uses denoising autoencoder which is trained with a corrupted input data to reconstruct the original input data and as a result, the autoencoder can learn more robust features. The input signal must be segmented into a sequence of data windows before they are classified into activities. The size of the data window defines the input size of the denoising autoencoder. An accelerometer produces three measurements along vertical, horizontal and sideway axes. The three windows are concatenated to become a single sequence of sensor readings, $\boldsymbol{x}$ which represents the input data of the denoising autoencoder.

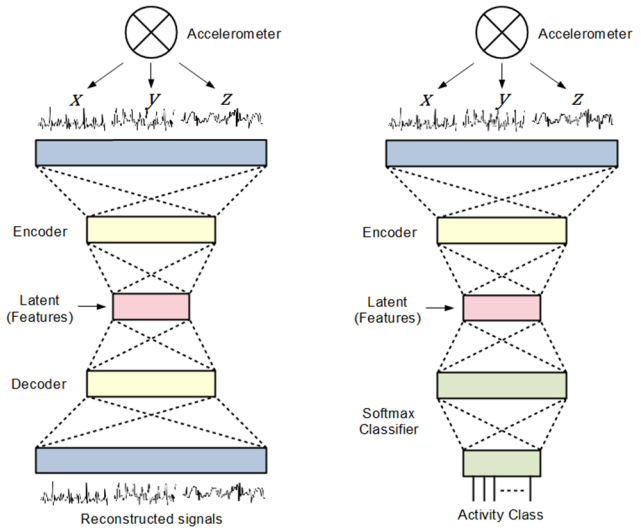


Fig. 1. Overview of the proposed method. The feature learning pipeline (left) and classification pipeline (right).

$$\boldsymbol{x} = [\boldsymbol{s_x}, \boldsymbol{s_y}, \boldsymbol{s_z}] \tag{1}$$

where $\boldsymbol{s_x}, \boldsymbol{s_y}$ and $\boldsymbol{s_z}$ are the window segmentation of acceleration data along vertical axis, A_x, horizontal axis, A_y and sideway axis, A_z. Therefore, the input size equals $3w$ where w is the size of the data window. Typically, the fixed sliding window is used to segment the signals. However, the technique has been found to be not effective due to varying length of transitional activity signals. Therefore, the adaptive sliding window technique is employed to segment the signals [9]. The technique defines an initial size of window segmentation which can be expanded to accommodate transitional activity signals that are longer then the initial window. As a result, a more effective segmentation can be achieved. The linear interpolation is used to resize the transitional activity windows to the size of the initial window. We corrupt the input data by adding uniform random noise, $\boldsymbol{\omega}$ which can be defined as follows.

$$\acute{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\omega} \tag{2}$$

Algorithm 1

Unlabeled training dataset, $D_u = \{\acute{\boldsymbol{x}}_i\}$

Labeled training dataset, $D_l = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}$

1. Set initial window size, overlapping factor and expansion factor
2. Segment the signals using adaptive sliding window
3. Apply the linear interpolation to resize windows of different size to initial window size
4. Concatenate the window segmentation $\boldsymbol{s_x}, \boldsymbol{s_y}$ and $\boldsymbol{s_z}$ to obtain input vector $\boldsymbol{x}$
5. Corrupt the input vector by adding random uniform noise to obtain $\acute{\boldsymbol{x}}$
6. Initialize the weights $\boldsymbol{W}$ and bias $\boldsymbol{b}$ of each layer
7. Train the denoising autoencoder using the unlabeled training dataset D_u
8. Replace the decoder with a softmax classifier
9. Freeze the encoder and code, and train the softmax classifier using the labeled training dataset D_l

TABLE I. The Parameters of the Denoising Autoencoder and Softmax Classifier

Layer	Parameters
Encoder	Number of hidden layers: 3 Number of units: 160, 80, 50 Activation function: ReLU
Code	Number of units: 5, 10, 15, 20, 25, 30 Activation function: ReLU
Decoder	Number of hidden layers: 3 Number of units: 50, 80, 160 Activation function: ReLU
Softmax Classifier	Number of hidden layers: 2 Number of units: 40, 20 Activation function: ReLU

The denoising autoencoder consists of two parts which are encoder and decoder. The encoder accepts the corrupt input data and propagate them through hidden layers to transform the data into code (feature representation) and the decoder uses the code to reconstructs the signals. Given an encoder with a single layer, the encoding can be defined as follows.

$$\boldsymbol{h} = \sigma(\boldsymbol{W_1}\acute{\boldsymbol{x}} + \boldsymbol{b_1}) \tag{3}$$

where h and b are N-dimensional code and bias vector respectively. W_1 is an $N \times M$ weight matrix where N is the number of neurons in the hidden layer and M is the number of input neurons. σ is a non-linear activation function. The decoding is performed using the same operation as follows.

$$\hat{\boldsymbol{x}} = \sigma(\boldsymbol{W_2}\boldsymbol{h} + \boldsymbol{b_2}) \tag{4}$$

where $\hat{x}$ and b_2 are M-dimensional reconstructed vector of $\acute{x}$ and bias vector respectively and W_2 is the $M \times N$ weight matrix.

We train the proposed autoencoder by regressing to the original input data. To do this, the proposed autoencoder is trained to minimize reconstruction error or the squared error between the input data and the reconstructed vector. The loss is defined as follows.

$$L(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \sum_{j=1}^{M} \left(x_j - \hat{x}_j\right)^2 \quad (5)$$

We found that minimizing squared error loss is not producing good reconstructed signals. Therefore, we add an extra penalty term to the cost function to penalize the network when $\hat{x}$ deviates from x. We use Kullback-Leibler (KL) divergence function which is a distance measure between two probability distribution. The cost function is defined as follows.

$$J = L(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \lambda \sum_{j=1}^{M} KL\left(x_j \parallel \hat{x}_j\right) \quad (6)$$

where

$$\sum_{j=1}^{M} KL\left(x_j \parallel \hat{x}_j\right) = \sum_{j=1}^{M} x_j \ln x_j - \sum_{j=1}^{M} x_j \ln \hat{x}_j \quad (7)$$

λ is the KL divergence penalty which is in the range of 0 and 1. The pseudocode of the proposed method is described in Algorithm 1. It starts with the signal segmentation using adaptive sliding window followed by linear interpolation to obtain windows segmentations of equal size. The window segmentations are concatenated to obtain the unlabeled training dataset. Then, the proposed autoencoder is trained to learn useful features using the unlabeled training dataset. Next, the decoder part is replaced with a softmax classifier and the classifier is trained using the labeled training dataset. Table 1 lists the parameters of the denoising autoencoder and softmax classifier.

III. Experimental Setup for Activity Recognition

We performed the experiments using a public dataset [10]. The dataset contains activity signals collected from 30 subjects using a smartphone inertial sensor. The position of the device is the front waist. The dataset includes basic activities and the transitions between body postures as shown in Table II. In the experiment, only the accelerometer data is used for activity recognition. The 3-fold cross validation scheme is used to avoid bias in the results. The initial window size of the adaptive sliding window is set to 100 samples (2 s). Both the overlapping factor and expansion factor is set to 0.5.

TABLE II. Categories of Activity

Dynamic	Static	Transitional
Walking	Standing	Stand-to-Sit
	Sitting	Sit-to-Stand
	Lying down	Sit-to-Lie
		Lie-to-Sit

We implemented the proposed method using TensorFlow. Then, Adam optimizer is used to train the weights of the autoencoder and softmax classifier layers. The learning rate and λ are set to 0.001 and 0.0001 respectively. We compute and tabulate the recall measure, precision measure, f-score and accuracy from the values of true positive (TP), false positive (FP), true negative (TN) and false negative (FN) to evaluate the performance of the proposed method. The evaluation metrics are given by

$$\text{Recall} = \frac{TP}{TP+FN} \quad (8)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (9)$$

$$\text{F-score} = \frac{2 \times \text{Recall} \times \text{Precision}}{(\text{Recall} + \text{Precision})}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (10)$$

IV. Results and Disccusion

We analyze the performance of the proposed method in learning the features for activity recognition. To do so, we performed the experiments by varying the code size (number of units of the code layer) of the denoising autoencoder. The values of the code size are 5, 10, 15, 20, 25 and 30. The comparison of recognition accuracies for different code sizes is given in Fig. 2. The code size determines the number of features for activity recognition. Therefore, the code size has significant influence on the recognition accuracy. As shown in Fig. 2, a relatively low accuracy is obtained when the code size is set to 5, whereby only 0.7634 of the activities are correctly classified. The low recognition accuracy could be attributed to the insufficient features to distinguish the activities. The recognition accuracy is increased significantly when the code size is set to 10, and the accuracy is highest when the code size equals 25. No improvement is observed when the code size is increased to 30. This could be due to overfitting as more features are added to the classification model.

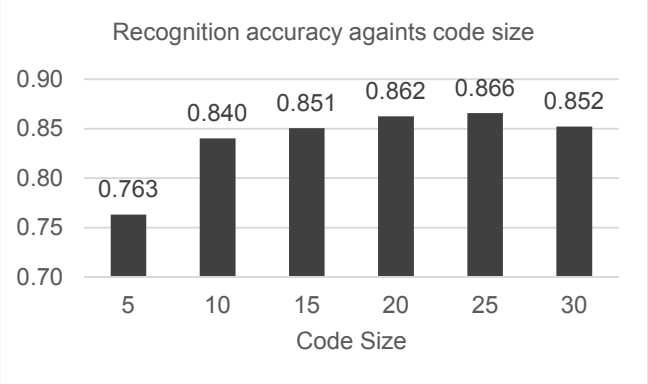


Fig. 2. Comparsion of recognition accuracies for different code sizes.

We analyze the performance of proposed method with code size of 25 for classifying each activity. The activities that have been considered are walking (A1), standing (A2), stand-to-sit (A3), sitting (A4), sit-to-stand (A5), sit-to-lie (A6), lying down (A7) and lie-to-sit (A8). The recall measure, precision measure and f-score of the activity recognition are given in Fig. 3. In general, the proposed method performed well in classifying most of the activities except walking, stand-to-sit and sit-to-stand where the recall measures are less than 0.80. This shows that the proposed method able to learn the relevant features in sensor data to classify the activities. As per the confusion matrix, it is observed that a quarter of walking samples are misclassified in which half of them are classified

as stand-to-sit and sit-to-stand. This is reflected in the low precision measures of stand-to-sit and sit-to-stand. The reason for this is due to the fact that the features of the activities have similar representation which causes misclassification. Table III shows the confusion matrix of activity recognition using the proposed method.

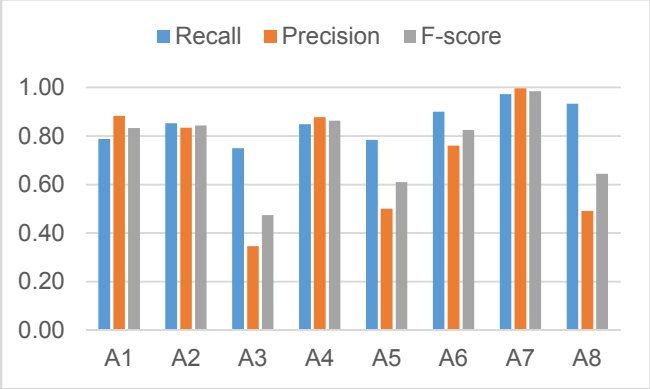


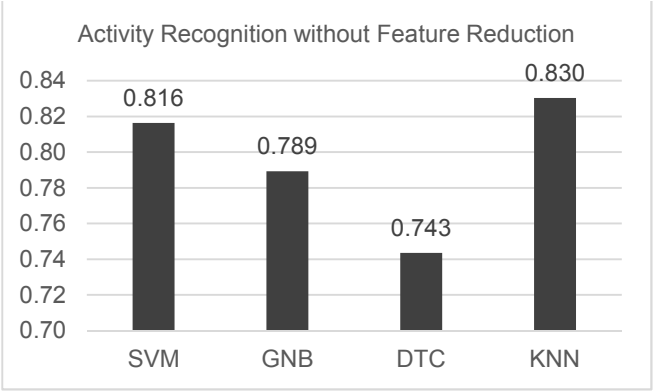
Fig. 3. The recall measure, precision measure and f-score of activity recognition.

TABLE III. CONFUSION MATRIX OF ACTIVITY RECOGNITION (CODE SIZE OF 25)

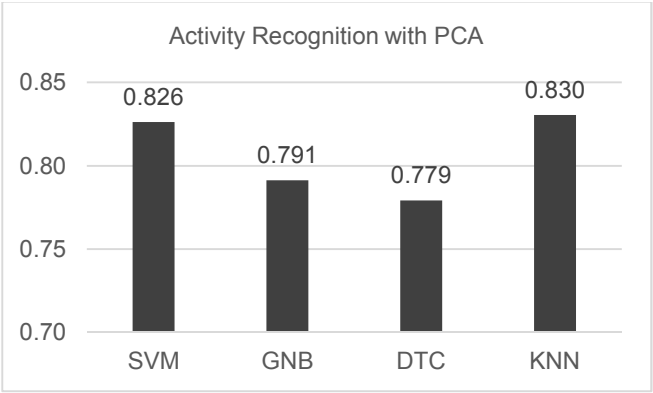
	A1	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**
A1	929	103	79	21	43	5	0	0
A2	87	1230	1	123	1	0	0	0
A3	9	0	45	4	0	2	0	0
A4	10	140	0	1062	1	7	0	32
A5	7	1	3	0	47	0	1	1
A6	1	0	2	0	0	54	3	0
A7	7	0	0	0	1	3	1265	25
A8	2	0	0	0	1	0	1	56

We compare the proposed method with the conventional method whereby the hand-crafted features are extracted and modeled using machine learning techniques such as Support Vector Machine (SVM), Gaussian Naïve Bayes (GNB), Decision Tree (DTC) and K-Nearest Neighbor (KNN). In this work, we extract the commonly used features such as mean, variance, tilt angle, mean crossing rate and signal magnitude area [11]. The features are extracted from the three axis of acceleration data and also from the derived acceleration data that represents the sagittal plane, transverse plane and frontal plane. A total of 30 features have been extracted from the data. We performed the activity recognition using all the extracted features and using Principal Component Analysis (PCA). PCA transforms the features into a new dimensional space to maximize the separation of the classes and as a result improves the classification performance.

The results of the activity recognition using the four classifiers are presented in Fig. 4. Generally, the performance of classifiers has been improved after the application of PCA. We can see that the best performance can be achieved using KNN with a recognition accuracy of 0.830 which is lower than the proposed method. Unlike the proposed method, the performance of conventional method relies on the feature extraction and reduction by human experts. Moreover, the extracted features may not capture the full representation of the data or optimize for activity classification.



(a)



(b)

Fig. 4. Recognition accuracy of the classifiers. (a) without feature reduction, (b) with PCA.

V. CONCLUSION

This paper presents an unsupervised deep learning method for activity recognition using denoising autoencoder and softmax classifier. The proposed method eliminates the need of feature extraction and selection, allowing the features to be learned from the data automatically. As a result, a more discriminative and salient features to be extracted, improving the performance of activity recognition. We demonstrate the performance of the proposed method on a public dataset. The results showed that the proposed method effectively extracts the relevant features resulting in better classification accuracy in a wide range of activities. We compare the proposed method with the conventional method. The proposed method outperforms the conventional method by 0.03.

ACKNOWLEDGMENT

This work was supported in part by the Universiti Sains Malaysia under Short Term Research Grant 304/PKOMP/6315206.

REFERENCES

[1] E. G. Eakin, W. J. Brown, A. L. Marshall, K. Mummery, and E. Larsen, 'Physical activity promotion in primary care', *Am. J. Prev. Med.*, vol. 27, no. 4, pp. 297–303, Nov. 2004.

[2] G. Grandes *et al.*, 'Effectiveness of Physical Activity Advice and Prescription by Physicians in Routine Primary Care: A Cluster Randomized Trial', *Arch. Intern. Med.*, vol. 169, no. 7, pp. 694–701, Apr. 2009.

[3] S. González, J. Sedano, J. R. Villar, E. Corchado, Á. Herrero, and B. Baruque, 'Features and models for human activity recognition', *Neurocomputing*, vol. 167, pp. 52–60, Nov. 2015.

[4] S. Rosati, G. Balestra, and M. Knaflitz, 'Comparison of Different Sets of Features for Human Activity Recognition by Wearable Sensors', *Sensors*, vol. 18, no. 12, p. 4189, Dec. 2018.

[5] S. R. Vanrell, D. H. Milone, and H. L. Rufiner, 'Assessment of Homomorphic Analysis for Human Activity Recognition From Acceleration Signals', *IEEE J. Biomed. Health Inform.*, vol. 22, no. 4, pp. 1001–1010, Jul. 2018.

[6] Z. Wang, D. Wu, J. Chen, A. Ghoneim, and M. A. Hossain, 'A Triaxial Accelerometer-Based Human Activity Recognition via EEMD-Based Features and Game-Theory-Based Feature Selection', *IEEE Sens. J.*, vol. 16, no. 9, pp. 3198–3207, May 2016.

[7] A. Ignatov, 'Real-time human activity recognition from accelerometer data using Convolutional Neural Networks', *Appl. Soft Comput.*, vol. 62, no. Supplement C, pp. 915–922, Jan. 2018.

[8] C. A. Ronao and S.-B. Cho, 'Human activity recognition with smartphone sensors using deep learning neural networks', *Expert Syst. Appl.*, vol. 59, pp. 235–244, Oct. 2016.

[9] M. H. M. Noor, Z. Salcic, and K. I.-K. Wang, 'Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer', *Pervasive Mob. Comput.*, vol. 38, pp. 41–59, Jul. 2017.

[10] 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, Jan. 2016.

[11] L. Gao, A. K. Bourke, and J. Nelson, 'Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems', *Med. Eng. Phys.*, vol. 36, no. 6, pp. 779–785, Jun. 2014.