Learning Deep Features for kNN-Based Human Activity Recognition

Sadiq Sani, Nirmalie Wiratunga, and Stewart Massie

School of Computing Science and Digital Media,
Robert Gordon University,
Aberdeen AB25 1HG, Scotland, UK
{s.sani|n.wiratunga|s.massie}@rgu.ac.uk

Abstract. A CBR approach to Human Activity Recognition (HAR) uses the kNN algorithm to classify sensor data into different activity classes. Different feature representation approaches have been proposed for sensor data for the purpose of HAR. These include shallow features, which can either be hand-crafted from the time and frequency domains, or the coefficients of frequency transformations. Alternatively, deep features can be extracted using deep learning approaches. These different representation approaches have been compared in previous works without a consistent best approach being identified. In this paper, we explore the question of which representation approach is best for kNN. Accordingly, we compare 5 different feature representation approaches (ranging from shallow to deep) on accelerometer data collected from two body locations, wrist and thigh. Results show deep features to produce the best results for kNN, compared to both hand-crafted and frequency transform, by a margin of up to 6.5% on the wrist and over 2.2% on the thigh. In addition, kNN produces very good results with as little as a single epoch of training for the deep features.

Keywords: human activity recognition, feature representation, deep learning

1 Introduction

Human activity recognition (HAR) is the computational discovery of human activity from sensor data and is receiving increasing interest in the areas of health care and fitness [3]. This is mainly driven by the need to find innovative ways to encourage physical activity. An example of a health application of HAR is SELFBACK [1] [1], an EU funded project that is developing a self-management system for patients with Lower Back Pain. The motivation for this work is driven by the need for an effective HAR component for SELFBACK, which is required to accurately measure adherence to physical activity targets.

HAR is generally considered as a classification problem where a classifier is trained to identify user activity from sensor data. A CBR approach to this problem makes use of

[1] http://www.selfback.eu/

a kNN classifier in order to facilitate similarity-based reasoning and explanation. However, the effectiveness of a kNN classifier depends on the quality of the feature representation used. Different feature representation approaches have been proposed for HAR, from shallow hand-crafted features to frequency transform features e.g. Fast Fourier Transforms (FFT) and Discrete Cosine Transforms (DCT) coefficients, and more recently, deep learning approaches. All these approaches have had some degree of success and setbacks in performance [6]. It is our view that none of the previous works provides a clear answer to which feature extraction approach is best. Also, previous works have evaluated these feature representation approaches on combinations of different types of data-sets with different mixes of sensor locations and classifiers. In this work, we focus on the feature representation for the kNN classifier using data from two popular body locations, wrist and thigh.

The main contribution of this work is an empirical evaluation of 5 different feature representation approaches across three different classes of features i.e. shallow hand-crafted features, shallow frequency transformation features and deep CNN derived features, for kNN, using sensor data collected from two common body locations, the wrist and the thigh. Wrist data is more prone to random noise compared to data collected at other body locations (e.g. thigh) due to increased variations in movement and posture possible with the hand while undertaking activities. Our goal in this work is to understand which of these feature representations is better suited for the kNN classifier and to analyse any differences in feature performance that may exist between the wrist and thigh.

The rest of this paper is organised as follows: in Section 2, we highlight important related work on feature representation for HAR. Our dataset is described in Section 3. Evaluation is presented in Section 4 and conclusions in Section 5.

2 Related Work on Feature Representation for HAR

Many different feature extraction approaches have been proposed for accelerometer data for the purpose of activity recognition [3]. We broadly classify these into hand-crafted, frequency-transform and deep features.

2.1 Hand-crafted Features

This is the most common approach to HAR and involves the computation of a number of defined measures on either the raw accelerometer data (time-domain) or the frequency transformation of the data (frequency domain) [5]. These measures are designed to capture the characteristics of the signal that are useful for distinguishing different classes of activities. In the case of both time and frequency domains, the input is a vector of real values $\overrightarrow{v} = v_1, v_2, ...v_n$ for each axis x, y and z. A function θ_i is then applied to each vector to compute a single feature value. Typical time domain features include mean, standard deviation and percentiles [10]; while typical frequency domain features include energy, spectral entropy and dominant frequency [2]. The time-domain and frequency domain features used in this work are presented in Table 1

Time Domain Features	*Frequency Domain Features*
Mean	Dominant frequency
Standard deviation	Spectral centroid
Inter-quartile range	Maximum
Lag-one-autocorrelation	Mean
Percentiles (10,25,50,75,90)	Median
Peak-to-peak amplitude	Standard deviation
Power	
Skewness	
Kurtosis	
Log-energy	
Zero crossings	
Root squared mean	

Table 1. Hand-crafted features for both time and frequency domains.

While hand-crafted features have worked well for HAR, a significant disadvantage is that they are domain specific. A different set of features need to be defined for each different type of input data i.e. accelerometer, gyroscope, time-domain and frequency domain. Hence, some understanding of the characteristics of the data is required. Also, it is not always clear which features are likely to work best [5]. Choice of features is usually made through empirical evaluation of different combinations of features or with the aid of feature selection algorithms [9].

2.2 Frequency Transform Features

Frequency transform features extraction involves applying a single function ϕ on the raw accelerometer data to transform this into the frequency domain, where it is expected that distinctions between different activities are more emphasised. The main difference between frequency transform and hand-crafted features is that the coefficients of the transformation are directly used for feature representation without taking further measurements. Common transformations that have been applied include Fast Fourier Transforms (FFTs) and Discrete Cosine Transforms (DCTs).

FFT is an efficient algorithm optimised for computing the discrete Fourier transform of a digital input. Fourier transforms decompose an input signal into its constituent sine waves. In contrast, DCT, a similar algorithm to FFT, decomposes a given signal into it's constituent cosine waves. Also, DCT returns an ordered sequence of coefficients such that the most significant information is concentrated at the lower indices of the sequence. This means that higher DCT coefficients can be discarded without losing information, making DCT better for compression.

For frequency transform feature extraction, a transformation function (DCT or FFT) ϕ is applied to the time-series accelerometer vector $\overrightarrow{v}$ of each axis. The output of ϕ is a vector of coefficients which describe the sinusoidal wave forms that constitute the original signal. Accordingly the transformed vector representations, $\mathbf{x}' = \phi(\mathbf{x})$, $\mathbf{y}' = \phi(\mathbf{y})$ and $\mathbf{z}' = \phi(\mathbf{z})$, are obtained for each axis of a given instance. Additionally we derive a further magnitude vector, $\mathbf{m} = \{m_{i1}, ..., m_{il}\}$ of the accelerometer data for each instance as a separate axis, where m_{ij} is defined as $m_{ij} = \sqrt{x_{ij}^2 + y_{ij}^2 + z_{ij}^2}$. As with $\mathbf{x}'$, $\mathbf{y}'$ and $\mathbf{z}'$, we also apply ϕ to $\mathbf{m}$ to obtain $\mathbf{m}' = \phi(\mathbf{m})$. The final feature representation is obtained by concatenating the absolute values of the first l coefficients of $\mathbf{x}'$, $\mathbf{y}'$, $\mathbf{z}'$ and $\mathbf{m}'$ to produce a single feature vector of length $4 \times l$. The value $l = 80$ is used in this work, which is determined empirically. Further information on feature representation using DCT and FFT can be found in [7].

2.3 CNN Feature Extraction

Convolutional Neural Networks (CNNs) have been applied for feature extraction in HAR, due to their ability to model local dependencies that may exist between adjacent data points in the accelerometer data [8]. CNNs are a type of Deep Neural Network that is able to extract increasingly more abstract feature representations by passing the input data through a stack of multiple convolutional operators [4], where each layer in the stack takes as input, the output of the previous layer of convolutional operators. An example of a CNN is shown in Figure 1.

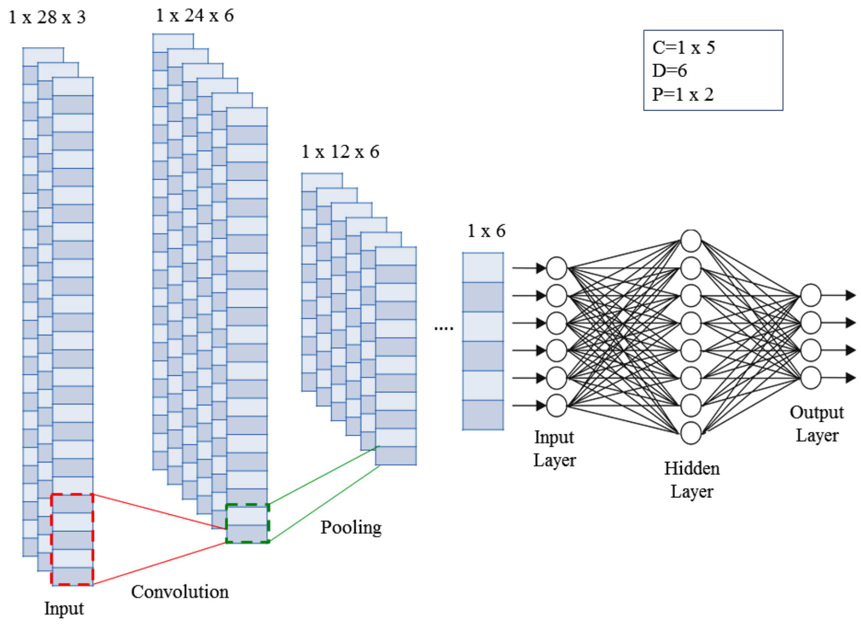

Fig. 1. Illustration of CNN

The input into the CNN in Figure 1 is a 3-dimensional matrix representation with dimensions $1 \times 28 \times 3$ representing the width, length and depth respectively. Tri-axial acceleromter data typically have a width of 1, a length l and a depth of 3 representing the

x, y and z axes. A convolution operation is then applied by passing a convolution filter over the input which exploits local relationships between adjacent data points. This operation is defined by two parameters, D representing the number of convolution filters to apply and C, the dimensions of each filter. For this example, $D = 6$ and $C = 1 \times 5$. The output of the convolution operation is a matrix with dimensions $1 \times 24 \times 6$, these dimensions being determined by the dimension of the input and the parameters of the convolution operation applied. This output is then passed through a Pooling operation which basically performs dimensionality reduction. The parameter P determines the dimensions of the pooling operator which in this example is 1×2, which results in a reduction of the width of its input by half. The output of the pooling layer can be passed through additional Convolution and Pooling layers. The output of the final Pooling layer is then flattened into a 1-dimensional representation and then fed into a fully connected neural network. The entire network (including convolution layers) is trained through back propagation over a number of generations until some convergence criteria is reached. Detailed description of CNNs can be obtained in [4].

Note that once the CNN is fully trained, it can used to provide feature representations for use with other types of classifiers e.g. kNN. This is achieved by cutting off the trained network after the final pooling layer and just before the fully-connected neural network. Each training example is then passed through the convolutional network in order to obtain an abstract representation which is used to train the kNN classifier. A similar operation is performed for each test example to obtain an abstract representation which is passed to kNN for classification.

3 Dataset

A group of 34 volunteer participants was used for data collection. The age range of participants is 18 - 54 years and the gender distribution is 52% Female and 48% Male. Data collection concentrated on the activities provided in Table 2.

Activity	**Description**
Walking	Walking at normal pace
Jogging	Jogging on a treadmill at moderate speed
Up Stairs	Walking up 4 - 6 flights of stairs
Down Stairs	Walking down 4 - 6 a flights of stairs
Standing	Standing relatively still
Sitting	Sitting still with hands on desk or thighs

Table 2. Details of activities classes in our dataset.

This set of activities was chosen because it represents the range of normal daily activities typically performed by most people. Data was collected using the Axivity Ax3 tri-axial accelerometer [2] at a sampling rate of 100Hz. Accelerometers were mounted

[2] http://axivity.com/product/ax3

on the right-hand wrists and right thighs of the participants. Activities are evenly distributed between classes as participants were asked to do each activity for the same period of time (3 minutes).

4 Evaluation

Evaluations are conducted using a leave-one-person-out methodology where each user's data is held out for testing in turns, while the remaining 33 are used for training. In this way, we are testing the general applicability of the system to users whose data is not included in the trained model. Performance is reported using macro-averaged F1 and kNN is used for classification with euclidean distance and the parameter $k = 5$.

The representations included in our comparison are as below:

- *Time*: Time domain hand-crafted features
- *Freq*: Frequency domain hand-crafted features
- *DCT*: DCT frequency features
- *FFT*: FFT frequency features
- *CNN*: CNN deep features with soft-max classifier
- *CNN-kNN*: CNN deep features with kNN classifier

For the CNN, after experimenting with different parameter settings, the final configuration used for thigh data had 3 convolution layers with 150, 100 and 80 convolution filters respectively. The configuration used for wrist data had 5 convolution layers with the same numbers of convolution filters as the thigh data in the first 3 layers and 60 and 40 convolution filters in the fourth and fifth layers respectively. Each convolution layer was followed by a max pooling layer. A convolution filter of size 10 and pooling size of 2 were used on all convolution and pooling layers respectively. The last pooling layer is connected to a fully connected network with 2 hidden layers, where the first layer had 900 units and second layer had 200 units. A dropout probability of 0.5 was used for each hidden layer. The final output layer had 6 units representing the 6 activity classes in our dataset and uses soft-max regression. Loss is computed using cross-entropy and the network is trained using back-propagation for 200 epochs.

4.1 Results

Results of our comparative evaluation are shown in Figure 2. The best results for both thigh and wrist are achieved using deep features (CNN and CNN-kNN). This highlights the fact that kNN, using deep features, can rival the performance of state-of-the-art deep learners, while still providing the ability for similarity-based reasoning and explainability that makes kNN desirable. In general, HAR performance is higher using thigh data compared to wrist by a margin of up to 14.7% (for DCT). This indicates that the thigh is a much better position for HAR compared to the wrist. However, the benefit from deep feature representations is consistent on both wrist and thigh.

Out of the shallow features (Time, Freq, DCT and FFT), the best performance is achieved using DCT. This is consistent with our previous findings [7]. However, in comparison with DCT, CNN-kNN produces 6.5% and 2.2% improvement on the wrist and thigh respectively. Both improvements are statistically significant at 95% using a paired t-test.

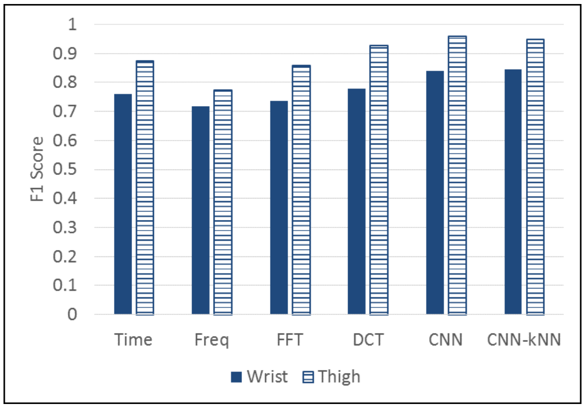

Fig. 2. Results of different representations(Time, Freq, DCT, FFT, CNN, CNN-kNN)

It is known that one of the major bottleneck of applying deep learning is the amount of time required for training. Hence, it is important to understand the effect of training time on the performance of both CNN and CNN-kNN. Particularly, we would like to see the level of performance that can be achieved with minimum training time. Figure 3 presents the results of CNN and CNN-kNN at between 1 to 5 epochs of training for the wrist (left) and thigh (right).

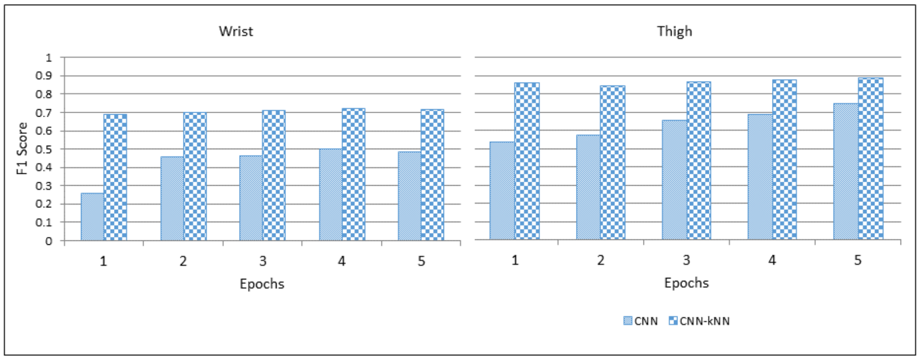

Fig. 3. Results for CNN and CNN-kNN after training for between 1-5 epochs.

Note that CNN-kNN outperforms CNN on both wrist and thigh at all 5 training epochs. Also, the performance of CNN-kNN is good (on par with Freq) even after a single epoch of training. These results are an important finding of this work and demonstrate the robustness of kNN in effectively using deep features, irrespective of the amount of time spent on training.

Finally, we analyse the effect of the depth of our network on the quality of deep features we are able to extract for kNN. Figure 4 shows the performance of CNN-kNN with different numbers of convolution layers between 3 and 5. Note that the best performance for the thigh is achieved using 3 convolution layers (0.949) and performance gradually decreases with the addition of more convolution layer (0.947 for 4 and 0.937 for 5). In contrast, performance on the wrist produces a significant increase (at 95% using a paired t-test) with additional layers from 0.73 for 3 layers to 0.84 for 5 layers. This indicates that deeper layers are required for effective feature extraction on more difficult datasets. However, a relatively shallow architecture seems sufficient for easier datasets.

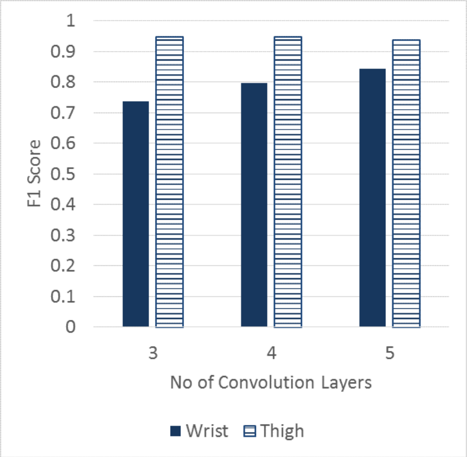

Fig. 4. Results for CNN-kNN at different depths between 3-5 convolution layers.

5 Conclusion

In this paper, we have presented an analysis of different feature representation approaches for the purpose of human activity recognition using kNN. These feature representation approaches can be broadly categorised into three classes: handcrafted, frequency transform and deep features. Evaluation is conducted using accelerometer data collection from two different body locations: wrist and thigh. Results show deep features to significantly out-perform the other representation types on both wrist and thigh by a margin of over 6.5% on the wrist, and 2.2% on the thigh. In addition, our evaluation shows kNN to be very effective at using deep features, even when a minimum amount of time spent in training these deep features.

Future work will investigate the use of RNN for feature extraction due to their ability to model the sequential relationship inherent in the time series accelerometer data.

References

1. K. Bach, T. Szczepanski, A. Aamodt, O. E. Gundersen, and P. J. Mork. Case representation and similarity assessment in the selfback decision support system. In *Proceedings of 24th International Conference on Case-Based Reasoning, ICCBR 2016*, pages 32–46. Springer International Publishing, 2016.
2. D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, 2010.
3. O. D. Lara and M. A. Labrador. A survey on human activity recognition using wearable sensors. *Communications Surveys & Tutorials, IEEE*, 15(3):1192–1209, 2013.
4. Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 255–258. MIT Press, 1998.
5. T. Plötz, N. Y. Hammerla, and P. Olivier. Feature learning for activity recognition in ubiquitous computing. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, IJCAI'11, pages 1729–1734. AAAI Press, 2011.
6. D. Ravi, C. Wong, B. Lo, and G. Z. Yang. A deep learning approach to on-node sensor data analytics for mobile or wearable devices. *IEEE Journal of Biomedical and Health Informatics*, 21(1):56–64, Jan 2017.
7. S. Sani, N. Wiratunga, S. Massie, and K. Cooper. Selfback-activity recognition for self-management of low back pain. In *Research and Development in Intelligent Systems XXXIII: Incorporating Applications and Innovations in Intelligent Systems XXIV*, pages 281–294. Springer, 2016.
8. 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 *Proceedings of 6th International Conference on Mobile Computing, Applications and Services*, pages 197–205, 2014.
9. S. Zhang, P. Mccullagh, and V. Callaghan. An efficient feature selection method for activity classification. In *Proceedings of IEEE International Conference on Intelligent Environments*, pages 16–22, 2014.
10. Y. Zheng, W.-K. Wong, X. Guan, and S. Trost. Physical activity recognition from accelerometer data using a multi-scale ensemble method. In *IAAI*, 2013.