Human Activity Recognition using Wearable Sensors by Deep Convolutional Neural Networks

Wenchao Jiang
Department of Computer Science
Missouri University of Science and Technology
wjm84@mst.edu

Zhaozheng Yin
Department of Computer Science
Missouri University of Science and Technology
yinz@mst.edu

ABSTRACT

Human physical activity recognition based on wearable sensors has applications relevant to our daily life such as healthcare. How to achieve high recognition accuracy with low computational cost is an important issue in the ubiquitous computing. Rather than exploring handcrafted features from time-series sensor signals, we assemble signal sequences of accelerometers and gyroscopes into a novel activity image, which enables Deep Convolutional Neural Networks (DCNN) to automatically learn the optimal features from the activity image for the activity recognition task. Our proposed approach is evaluated on three public datasets and it outperforms state-of-the-arts in terms of recognition accuracy and computational cost.

Categories and Subject Descriptors

I.2 [**Computing Methodologies**]: Artificial Intelligence; F.1.1 [**Theory of Computation**]: Models of Computation—*Self-modifying machines*

Keywords

Wearable Computing, Activity Recognition, Deep Convolutional Neural Networks, Activity Image.

1. INTRODUCTION

Human physical activity is defined by bodily states such as walking and standing, the recognition of which can be applied to many application fields such as human-computer interaction and surveillance [1][2]. Especially, activity recognition based on wearable sensors is directly related to our daily lives such as healthcare and workout monitoring [3][4]. This paper focuses on activity recognition using ubiquitous wearable devices (e.g., smart phones, smart watches and sport bracelets) which embed accelerometers and gyroscopes.

1.1 Related Work

Generally, sensor-based activity recognition methods are evaluated in two aspects: recognition accuracy and computational cost. To improve the accuracy, the pervious work tried to extract effective handcrafted features from tri-axis signals of accelerometers and gyroscopes [1] or explored different classifiers including Support Vector Machine [5], Random Forest [6] and Hidden Markov Model [7]. To decrease the computational cost, researchers performed feature selection [4], sensor selection [8], contextual prediction [9] and sample frequency reduction [10].

However, in most cases, the improvement of accuracy is at the expense of increasing the computational cost [11]. This paper aims to develop an accurate and efficient approach for activity recognition using wearable devices.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
MM'15, October 26–30, 2015, Brisbane, Australia.
© 2015 ACM. ISBN 978-1-4503-3459-4/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2733373.2806333.

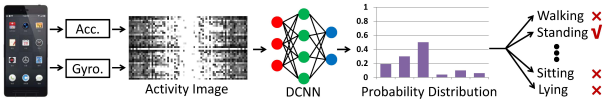

Figure 1: Overview of recognizing query signals.

1.2 Our Proposal

In the previous work, features used for activity recognition are usually extracted independently from multiple time-series sensor signals in a handcrafted way [1]. The correlation among different signals are usually overlooked. We propose to transfer all time-series signals from the accelerometer and gyroscope into a new *activity image* which contains hidden relations between any pair of signals.

As one of the most effective deep learning models, Deep Convolutional Neural Networks (DCNN) achieved the superior performance in speech recognition [12] and image classification [13]. Our novel 2D activity image enables DCNN to automatically learn the best discriminative features suited for activity recognition, rather than defining the features manually. In the DCNN framework, low- and high-level features are extracted from the newly-defined activity image, providing extra contextual information between signals compared with the statistics on the individual time-series signal.

1.3 Algorithm Overview

Fig.1 shows the flowchart to recognize query signals. With the activity image generated from sensor signals as input, DCNN outputs a probability distribution over activity categories, based on which the human activity is determined.

The paper is organized as follows. Section 2 and 3 introduce the activity image and DCNN architecture, respectively. Section 4 presents the complete activity recognition procedure. Experimental results on three public datasets are described in Section 5.

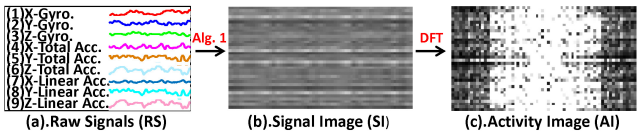

Figure 2: Flowchart to generate an activity image.

2. ACTIVITY IMAGE

The Inertial Measurement Unit in wearable devices includes an accelerometer and a gyroscope, measuring the strength of tri-axis acceleration and angular velocity, respectively. The combination of accelerometer and gyroscope achieves better results than accelerometer only [1]. Furthermore, the total acceleration can be separated into two components: linear acceleration (body motion) and gravity.

We propose a novel *activity image* based on signals of gyroscope, total acceleration and linear acceleration. Firstly, raw signals (Fig.2 (a)) are stacked row-by-row into a signal image (Fig.2 (b)) based on Algorithm 1. In the signal image, every signal sequence has the chance to be adjacent to every other sequence, which enables DCNN to extract hidden correlations between neighboring signals.

Then, 2D Discrete Fourier Transform (DFT) is applied to the signal image and its magnitude is chosen as our activity image (Fig.2 (c)). Fig.3 shows a few activity images corresponding to different activities. The visual difference in these activity images indicates their potential for DCNN to extract discriminative image features for activity recognition.

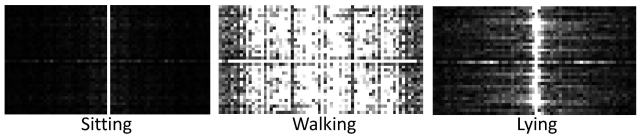

Figure 3: Samples of activity images.

Before introducing the DCNN architecture, we point out some engineering details. Given 9 signal sequences, the size of our signal image is $37 \times L_s$ based on Algorithm 1, where L_s is the time length of signal sequences. Since different devices may have different sampling rates, we interpolate all raw signals in the same time length (e.g., 1 second) with 68 signal samples. $L_s = 68$ will facilitate the design of our DCNN architecture. In addition, we delete the last row of the signal image generated by Algorithm 1, so each signal occurs four times in the revised signal image. The size of our signal image and consequent activity image is finalized as 36×68.

3. DCNN ARCHITECTURE

The architecture of our proposed DCNN is summarized in Fig.4. The first convolutional layer filters the 36×68 input activity image with 5 kernels of size 5×5, followed by 4×4 subsampling. The second convolutional layer takes the output of the first subsampling layer as input and filters it with 10 kernels of size 5×5, followed by 2×2 subsampling. The full connected layer vectorizes the output of the second subsampling layer into a 120-dimensional feature vector.

Each convolutional layer performs 2D convolution on its input maps. The output maps are generated as:

$$y_j = (1 + exp(b_j + \sum_i k_{ij} * x_i))^{-1} \quad (1)$$

Algorithm 1 Raw Signals → Signal Image

Notations:
- Signal Image (SI): a 2D array to store permutated raw signals.
- Signal Index String (SIS): a string to store signal indices, whose length is N_{SIS}.

Intput: N_s signal sequences. *// As shown in Fig.2(a), each signal is label with a sequence number. The number of signal sequences $N_s = 9$.*

Loops:

$i = 1; j = i + 1;$
SI is initialized to be the i-th signal sequence;
SIS is initialized to be $'i'$; N_{SIS}=1;
while $i \neq j$ **do**
  if $j > N_s$ **then**
    $j = 1;$
  else if $'ij' \notin SIS$ && $'ji' \notin SIS$ **then**
    Append the j-th signal sequence to the bottom of SI;
    Append $'j'$ to SIS; $N_{SIS} = N_{SIS} + 1;$
    $i = j; j = i + 1;$
  else
    $j = j + 1;$
  end if
end while

Output: SI.

// The final SIS is '123456789135792468147158259369483726 1' with $N_{SIS} = 37$, when $N_s = 9$.

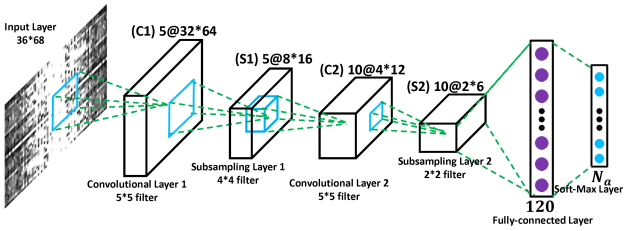

Figure 4: The proposed DCNN architecture.

where $*$ denotes the convolutional operator. k_{ij} is the convolutional kernel on the i-th input map x_i to generate the j-th output map y_j. b_j a bias term.

Mean-pooling is adopted as the subsampling method. The output map y_i is achieved by computing mean values over non-overlapping regions of the input map x_i with a square filter $m \times m$:

$$y_i(r, c) = \frac{\sum_{p=1}^{m} \sum_{q=1}^{m} x_i(r \cdot m + p, c \cdot m + q)}{m^2} \quad (2)$$

where $y_i(r, c)$ denotes the pixel value of y_i at location (r, c).

After fully connecting the output of the second subsampling layer into a 1D feature vector f_i, N_a-way softmax function is applied to obtain a probability distribution over N_a classes, where N_a denotes the number of activities to be recognized:

$$p(s) = \frac{g_s}{\sum_{j=1}^{N_a} g_j}, \text{ where } g_j = max(0, \sum_i f_i \cdot w_{ij} + h_j) \quad (3)$$

where $p(s)$ is the probability of an activity belonging to the s-th class. w_{ij} and h_j are coefficients in the softwmax function.

Fig.5 shows some output maps extracted by DCNN. From the training data, deep learning is able to automatically learn from low-level to high-level features as the layer goes deeper. The parameters such as k_{ij}, b_j, w_{ij} and h_j are updated by stochastic gradient descent where the gradient is obtained by back-propagation.

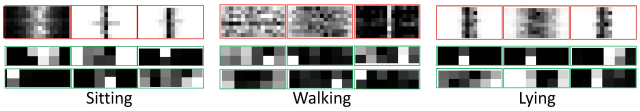

Figure 5: Examples of output maps in each layer. Maps in red boxes are the output of the first subsampling layer and maps in green boxes are the output of the second subsampling layer.

4. ACTIVITY RECOGNITION

With the activity image built from raw sensor signals as input, DCNN outputs a probability distribution over N_a activities. Fig.6 shows two distribution examples over N_a activities. In Fig.6(a), the probability of "Walking Downstairs (W.D.)" is overwhelmingly higher than the others, thus it is confident to determine the user's activity is walking downstairs. However, in Fig.6(b), although the probability of "Walking Downstairs" is still the highest, it's close to "Walking" and "Standing", thus the activity recognition is not confident.

We train pair-wise SVM classifiers to mitigate the uncertainty. For $N_a = 6$, 15 bi-class SVM classifiers are trained based on the statistic features from [1]. Then we compare each two activities in the descending order of their probability values. As shown in Fig.6(a), the result is determined by the activity with the highest probability, when the ratio between the highest probability and the second highest is higher than a threshold T

$$T = \frac{\sum_{i=1}^{N_{train}} \frac{1}{var(p_i(s), s\in[1,N_a])}}{N_{train}}, \quad (4)$$

which is the mean of the variance's reciprocal. N_{train} is the number of training cases. $p_i(s)$ denotes the probability of the i-th training case's activity belonging to the s-th class.

However, in Fig.6(b), the ratios are less than T in the first two iterations, so bi-class SVM is applied to find the optimal activities. In the third iteration, the ratio is larger than T, so we consider the activity with the higher probability as the final result. SVM is incorporated here because the handcrafted features complement the automatically learned features by DCNN and SVM has shown its effectiveness on bi-class classification [14].

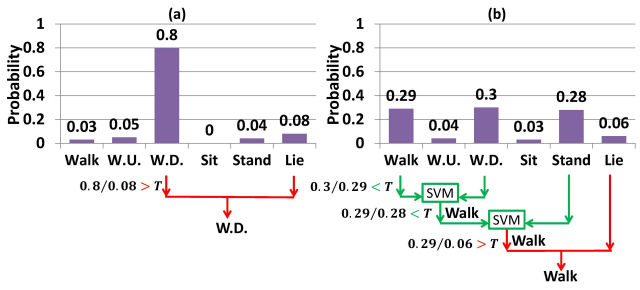

Figure 6: Two probability distribution examples by DCNN.

In summary, our activity recognition procedure consists of three steps: (1) generate an activity image from raw signals; (2) apply DCNN to the activity image and produce the probability distribution over all activities; and (3) if there is a sharp peak in the distribution, the activity categorization is confidently determined. Otherwise, bi-class SVM classifier will be applied to classify uncertain classes.

5. EXPERIMENTAL RESULTS

We validate our method on three public datasets whose information is summarized in Table 1. All our experiments are conducted in Matlab 2014 on an ordinary computer with a 2.6GHZ CPU and 8GB memory. Two quantitative metrics are used to evaluate different methods: (1) **Accuracy**, which is defined as the correctly recognized samples divided by the total number of testing samples and (2) **Computational cost**, which is defined as the average time used to recognize each testing sample.

Table 1: DATASET INFORMATION

Dataset	Number of Activity	Sensor Position	Sampling Rate	Number of Volunteers	Volunteers Age Range	Number of Training Samples	Number of Testing Samples
UCI[1]	6	Waist	50HZ	30	19~48	7352	2947
USC[2]	11	Hip	100HZ	14	21~49	7156	3414
SHO[3]	7	Wrist	50HZ	10	25~30	6826	2904

5.1 Evaluating the Design of Activity Image

Table 2 shows the accuracy of DCNN method with various designs of input activity images. The proposed activity image (Row 3) achieves the highest recognition accuracy. The accuracy decreases when we use the signal image directly (Row 1) or replace the Discrete Fourier Transform with the wavelet transform (Row 2). DFT improves the recognition performance because it increases the visual appearance difference of activity images (Fig.3), making it easier for DCNN to learn discriminative features.

Table 2: DCNN ACCURACY vs. INPUT IMAGES ∗

Input Image Design	DCNN Architecture	DCNN Accuracy (%)		
		UCI[1]	USC[2]	SHO[3]
RS $\xrightarrow{\text{Alg.1}}$ SI	Fig.4	88.87	94.11	96.76
RS $\xrightarrow{\text{Alg.1}}$ SI $\xrightarrow{\text{2D Wavelet}}$ AI	Fig.4	91.38	93.03	95.52
RS $\xrightarrow{\text{Alg.1}}$ SI $\xrightarrow{\text{DFT}}$ AI (Fig.2)	Fig.4	**95.18**	**97.01**	**99.93**

∗Raw Signals (RS); Signal Image (SI); Activity Image (AI).

5.2 Evaluating the Design of DCNN Architecture

Although it is infeasible to evaluate all possible DCNN architectures, we compare four representative DCNN designs with an increasing number of layers (Table 3). For our activity recognition task, two convolutional layers combined with two subsampling layers achieve the best accuracy. If the convolutional neural network is too shallow, high-level features are not be able to be learned. If the network is too deep, useful features may be filtered out during the convolutional and subsampling processes.

Table 3: DCNN ACCURACY vs. ARCHITECTURE ∗

Input Image	DCNN Architecture	DCNN Accuracy (%)		
		UCI[1]	USC[2]	SHO[3]
Fig.2	3C21×21–S8×8	86.70	79.97	96.45
Fig.2	5C5×5–S4×4–10C5×5–S2×2 (Fig.4)	**95.18**	**97.01**	**99.93**
Fig.2	5C5×5–S2×2–8C5×5–S2×2–10C5×5–S2×2	94.77	96.60	99.59
Fig.2	5C3×3–S2×2–7C2×2–S2×2–9C3×3–S2×2–11C2×2–S2×2	90.43	88.61	99.04

∗"C" and "S" denote convolutional layer and subsampling layer, respectively. The architecture is described as "{the number of output maps}C{kernel size}-S{kernel size}".

5.3 Quantitative Comparison

We compare our proposed methods, DCNN and DCNN+ (the DCNN method with the aid from SVM for uncertain cases), with two other state-of-the-arts: (1) SVM method [1], which extracts 561 handcrafted features from signals of accelerometers and gyroscopes and utilizes the SVM classifier; (2) Feature selection method [4], the aim of which is to reduce computational cost by selecting the most useful features. In [4], the feature importance is estimated by random forest which measures the increase of prediction error when a feature is permuted out.

The quantitative comparison on three public datasets in terms of recognition accuracy and computational cost is summarized in Table 4. The threshold T in DCNN+ method is computed by Eq.4, which is 43, 4, 2, respectively for the three datasets. The proposed DCNN+ and DCNN methods achieve the best performance in accuracy and computational cost, respectively. The SVM method [1] gets the second best accuracy but its computational cost is much higher. In contrast, the feature selection method [4] has the second lowest computational cost but its accuracy is the lowest among the four methods. Thus, there is a trade-off between accuracy and computational cost. Compared with the methods proposed in [1] and [4], the DCNN+ and DCNN methods achieve a good balance between accuracy and efficiency (high accuracy and low computational cost).

Table 4: PERFORMANCE COMPARISON

Dataset	Accuracy (%)				Computational Cost (ms)			
	DCNN+	DCNN	SVM[1]	Feature Selection[4]	DCNN+	DCNN	SVM[1]	Feature Selection[4]
UCI[1]	**97.59**	95.18	96.40	91.31	3.85	**1.56**	10.06	1.81
USC[2]	**97.83**	97.01	97.28	96.77	2.66	**1.54**	11.78	1.86
SHO[3]	**99.93**	99.93	99.93	99.58	1.56	**1.53**	9.87	1.56

5.4 Accuracy vs Computational Cost

Fig.7 shows the recognition accuracy and computational cost on the testing datasets if we gradually increase the threshold T in DCNN+ method. The green dots denote the accuracy corresponding to the T set by Eq.4 and the red dots denote the maximal accuracy achievable by DCNN+. They are very close to each other, which means T learnt from the training datasets by Eq.4 is effective on the testing datasets. On the UCI dataset (Fig.7(a)), when T increases, SVM classifier can assist DCNN on some uncertain cases. The accuracy curve increases and then levels off when $T = 50$, which means SVM cannot help solve uncertain cases anymore when $T > 50$. On the USC dataset (Fig.7(b)), the accuracy curve has the peak around $T = 4$, and then decreases a little bit. It indicates that handcrafted features overall can aid automatically learned feature for activity recognition, but there is possibly a small percentage of cases which are correctly recognized by DCNN while falsely classified by SVM if we choose T larger than 4. Accuracy curve on the SHO dataset (Fig.7(c)) is flat since our DCNN method only classifies 2 samples incorrectly out of total 2904 testing samples and the SVM classifier cannot solve these two samples either. Note that for all three datasets, time cost increases when T increases because more bi-class SVM classifications are applied when T increases.

6. CONCLUSION

This paper attacks the problem of accurate and efficient human activity recognition based on wearable sensors. To effectively learn discriminative features automatically, we propose a novel activity image as the input of DCNN. Raw sensor signals from accelerometers and gyroscopes are permutated and stacked in a signal image, the Discrete Fourier Transform of which is the newly-defined activity image. A DCNN architecture is designed to learn low-level to high-level features from the activity image for effective activity recognition. We compared our method with state-of-the-arts on three public datasets. The proposed two methods (DCNN+ and DCNN) achieve superior performance in terms of recognition accuracy and computational cost.

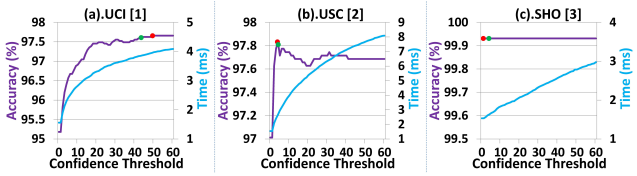

Figure 7: The recognition performance (accuracy and computational cost) regarding to different confidence thresholds. The green dots denote the accuracy corresponding to the T set by Eq.4 and the red dots denote the maximal accuracy achievable by DCNN+.

7. ACKNOWLEDGEMENT

This work was supported by NSF grants CNS-1205695 and IIP-1521289.

8. REFERENCES

[1] Davide 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, ESANN, 2013.

[2] Mi Zhang and Alexander A Sawchuk, "Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors," in ACM Conference on Ubiquitous Computing, 2012.

[3] Muhammad Shoaib, et al., "Fusion of smartphone motion sensors for physical activity recognition," Sensors, 2014.

[4] Pierluigi Casale, et al., "Human activity recognition from accelerometer data using a wearable device," in Pattern Recognition and Image Analysis, 2011.

[5] Davide Anguita, et al., "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in Ambient Assisted Living and Home Care, 2012.

[6] Toma Ât's Peterek, et al., "Comparison of classification algorithms for physical activity recognition," in Innovations in Bio-inspired Computing and Applications, 2014.

[7] Charissa Ann Ronao and Sung-Bae Cho, "Human activity recognition using smartphone sensors with two-stage continuous hidden markov models," in International Conference on Natural Computation, 2014.

[8] Piero Zappi, et al., "Activity recognition ÂÍ from on-body sensors: accuracy-power trade-off by dynamic sensor selection," in Wireless Sensor Networks, 2008.

[9] Dawud Gordon, et al., "Energy-efficient activity recognition using prediction," in International Symposium on Wearable Computers, 2012.

[10] Zhixian Yan, et al., "Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach," in International Symposium on Wearable Computers, 2012.

[11] Jian Cui and Bin Xu, "Cost-effective activity recognition on mobile devices," in International Conference on Body Area Networks, 2013.

[12] Abdel-rahman Mohamed, et al., "Investigation of fullsequence training of deep belief networks for speech recognition.," in INTERSPEECH, 2010.

[13] Alex Krizhevsky, et al., "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems, 2012.

[14] Shigeo Abe, Support vector machines for pattern classification, vol. 53, Springer, 2005.