Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks

- Presented by Ramesh Kumar Sah

Abstract

- Rather than exploring handcrafted features from time-series sensor signals, signal sequences from accelerometers and gyroscopes is assembled 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.
- Proposed approach is evaluated on three public datasets and it outperforms state-of-the-arts in terms of recognition accuracy and computational cost.

Methodology

- Transfer all the time-series signals from the accelerometer and gyroscope into an activity image, and use that to make classification.

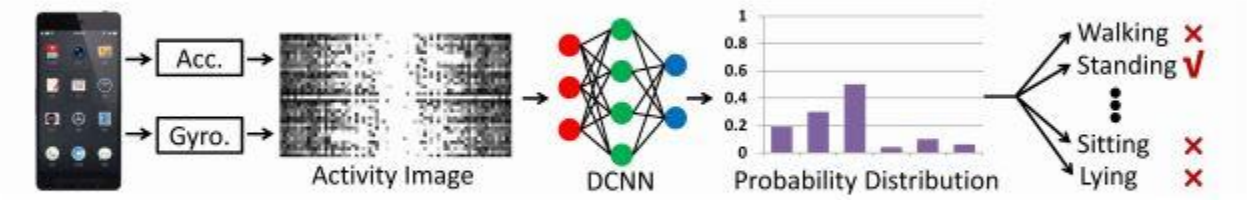


Figure 1: Overview of recognizing query signals.

Activity Image

- Activity image is based on signals of gyroscope, total acceleration, and linear acceleration.
- Raw signals are stacked row-by-row into a signal image. Then 2D Discrete Fourier Transform is applied to the the signal image and its magnitude is chosen as activity image.

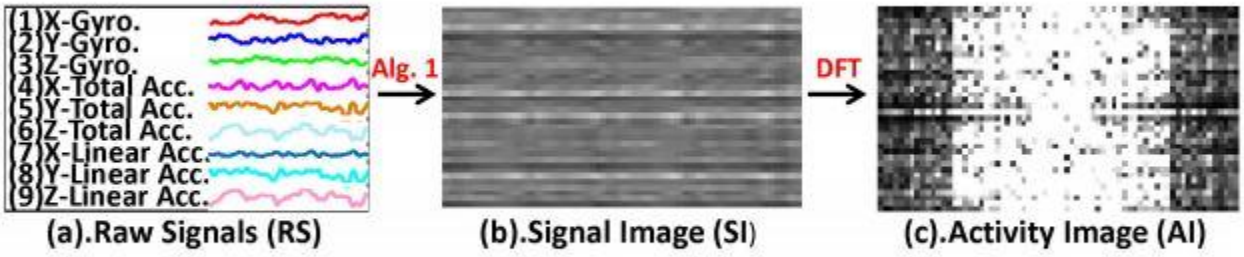


Figure 2: Flowchart to generate an activity image.

- In signal image every signal sequence has the chance to be adjacent to every other sequence. This enables the CNN to extract hidden correlations between neighboring signals.
- The signal length is 68 (1 second observation), and each signal occurs four times, thus the size of the signal image and consequently activity image is 36 * 68.

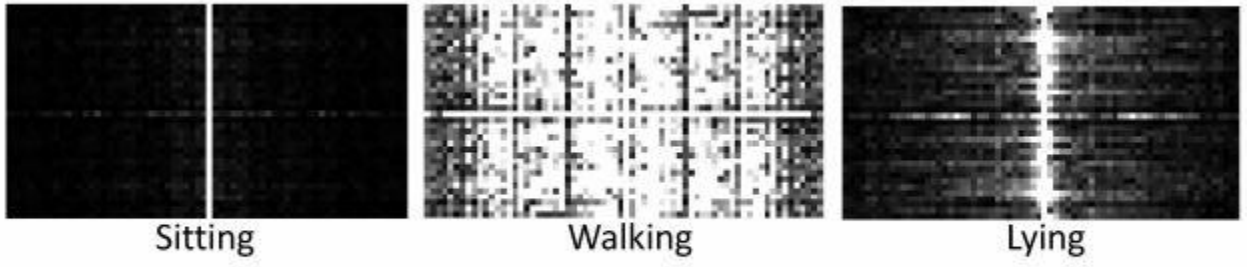


Figure 3: Samples of activity images.

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 $'ij' \not\subset SIS$ && $'ji' \not\subset SIS$ **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 $N_s = 9$.

CNN Architecture

-

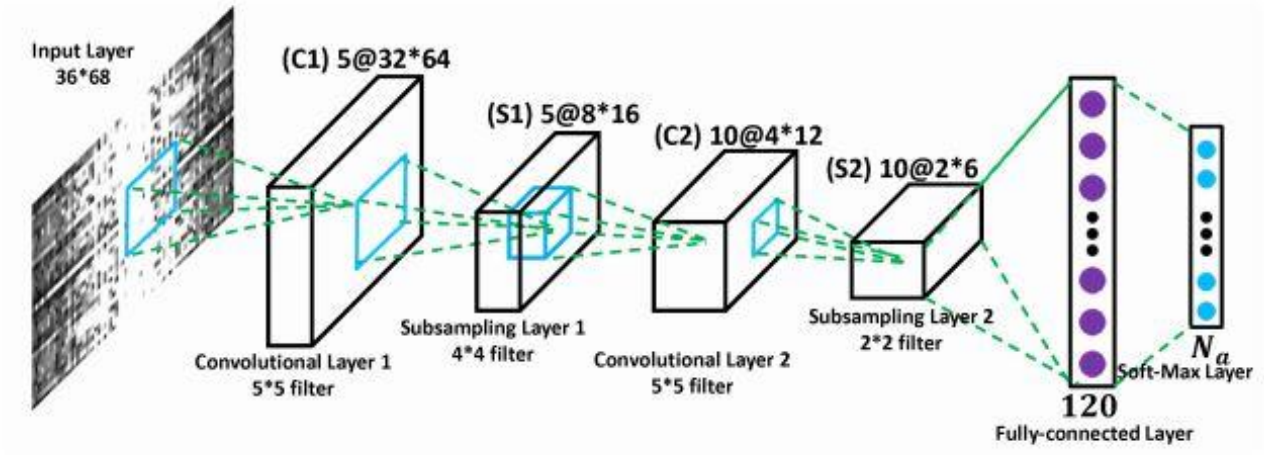


Figure 4: The proposed DCNN architecture.

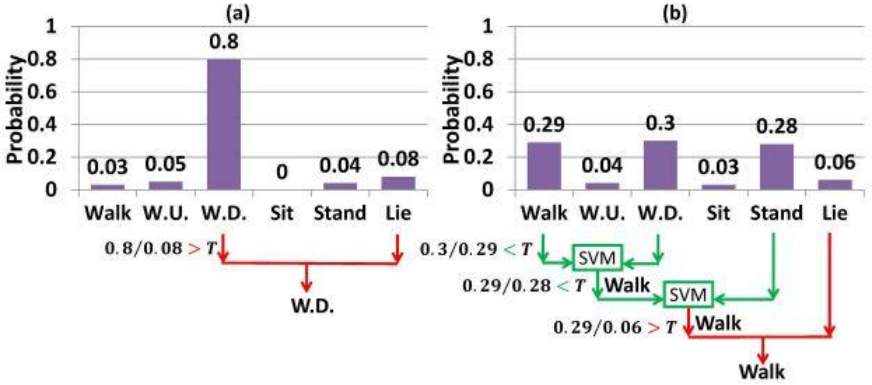
Activity Recognition Process

- Generate an activity image from raw signals
- Apply CNN to the activity image and produce the probability distribution over all activities
- 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 class.

- For 6 activities, 15 bi-class SVM classifiers are trained on the statistic features data.
- Each two activities are compared in the descending order of their probability values. The result is the activity with highest probability, when the ratio between the highest probability and second highest probability is higher than a threshold T, which is the mean of the variance reciprocal.

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

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.



Evaluation Metrics

- Accuracy :: The correctly recognized samples divided by the total number of testing samples
- Computational Cost :: 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

Results

DFT improves the recognition performance because it increases the visual appearance difference of activity images, 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).

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}".

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, which extracts 561 handcrafted features from signals of accelerometers and gyroscopes and utilizes the SVM classifier;

(2) Feature selection method, the aim of which is to reduce computational cost by selecting the most useful features. The feature importance is estimated by random forest which measures the increase of prediction error when a feature is permuted out.

The proposed DCNN+ and DCNN methods achieve the best performance in accuracy and computational cost, respectively.

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

Thank You !