UDC 004.421

HUMAN PHYSICAL ACTIVITY RECOGNITION ALGORITHM BASED ON SMARTPHONE DATA AND CONVOLUTIONAL NEURAL NETWORK

Z. WAN, A.A. BARYSKIEVIC

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus

Received March 18, 2023

Abstract. With a widespread of various sensors embedded in mobile devices, the analysis of human daily activities becomes more common and straightforward. Human activity recognition (HAR) is a prominent application of advanced Machine Learning (ML) and Artificial Intelligence (AI) techniques that utilizes computer vision to understand the semantic meanings of heterogeneous human actions. This paper describes a supervised learning method that can distinguish human actions based on data collected from practical human movements. The primary challenge while working with HAR is to overcome the difficulties that come with the cyclostationary nature of the activity signals. This study proposes a HAR classification model based on a Convolutional Neural Network (CNN) and uses the collected human action signals. The model was tested on the WISDM dataset, which resulted in a 92 % classification accuracy. This approach will help to conduct further researches on the recognition of human activities based on their biomedical signals.

Keywords: human activity recognition, machine learning, convolutional neural network.

Introduction

Humans possess an amazing skill to comprehend information that others pass on through their movements like the gesture of a certain body part or the motion of the entire body. We can differentiate among human postures, track complex human motions, and evaluate human-object interactions to realize what they are doing, and even deduce what they intend to do. Even though these are advanced recognition functionalities performed by the brain based on the images of the surroundings captured by the eyes, the process occurs almost autonomously to us. Machines, on the other hand, are still learning how to apprehend various human activities, and we are teaching them based on our knowledge and understandings of the task. Considering the fact that machines (or computers) were nothing but simple calculators to solve arithmetic problems just sixty years ago, their understanding of complex concepts has come a long way. ML as a part of the AI, has given machines the capacity to interpret various situations in their surroundings and respond accordingly like humans. HAR is being researched since the early 1980s because of its promise in many applied areas. However, the significant breakthroughs in this field have come within the last two decades [1]. The recent developments in microelectronics, sensor technology, and computer systems have made it possible to collect information that is more fundamental from human movements, and the advanced ML techniques have made that information more comprehensible to the machines.

There are several approaches to collect HAR data from the participating subjects; broadly, they fall into one of the two categories – namely camera-based recording or sensor-based recording [2]. In the former approach, one or more video cameras are set up to record the activities of a subject for a certain amount of time, and then the recognition is performed using video analysis and processing techniques. The later one utilizes various types of sensors to track the movements of the subject. This approach can be further classified based on the type of sensors used, whether they involve wearable body sensors or the external ones [1]. External sensors are placed in predetermined points of interest on the subjects' body, whereas wearable sensors require to be attached to the subject while collecting data. Each of these techniques has its advantages, shortcomings, and apposite applications. Some recognition techniques even combine multiple recording techniques to collect data that are more relevant and make the corresponding actions more interpretable to the machines. The applications of HAR

include intelligent surveillance, haptics, human-computer interaction, motion or gesture-controlled devices, automatic health-care monitoring systems, prosthetics, and robotics. Despite many advancements, HAR is still a challenging task because of the articulated nature of human activities, the involvement of external objects in human interactions, and complicated spatiotemporal structures of the action signals [3]. Success in recognizing these activities requires advanced signal and image processing techniques, as well as sophisticated ML algorithms. Since the absolute performance is yet to be achieved, HAR remains a tending field to the researchers.

Datasets

The WISDM dataset contains mobility information that was collected from 30 people of different ages (ranging from 19 to 48 years), genders, heights and weights using a wrist-mounted smartphone. The smartphone has integrated accelerometer and gyroscope. Action data was recorded using these sensors while each of the subjects was performing six predefined tasks, which according to the jargon of ML, represent six different classes. Three-axial linear acceleration and three-axial angular velocity data were acquired at a steady rate of 20 Hz. The collected samples were labeled manually afterward. Before putting in the dataset, the samples were pre-processed using a median filter for noise cancellation and a thirdorder low-pass Butterworth filter having a 20 Hz cutoff frequency.

The proposed physical activity recognition algorithm

This study aims to classify the HAR signals of the WISDM dataset employing a CNN model, as shown in Figure 1. The training stage requires a set of data samples containing various attributes measured from subjects while performing various predefined activities. The supervised learning technique then try to make some "sense" out of the data, find out how the samples that belong to the same class are similar to each other while samples from different classes are diverse, then builds one or more internal models focusing on the crucial attributes that can highlight those contrasting properties to carry out the classification [1]. In the training stage, a preordained portion of the dataset is used to train the machine and build a feasible model, which is then evaluated over the remaining samples.

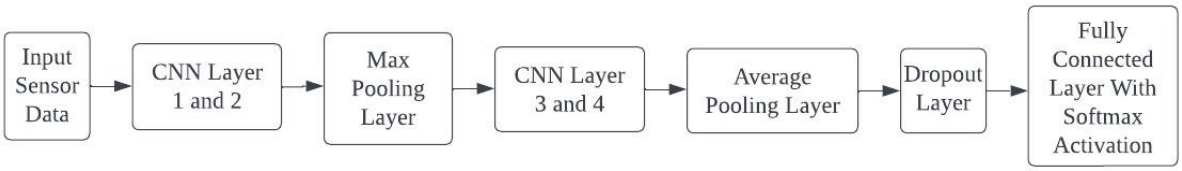

Figure 1. Block diagram of the proposed CNN-based HAR algorithm

The data has been preprocessed in such a way that each data record contains 80 time slices (data was recorded at 20 Hz sampling rate, therefore each time interval covers four seconds of accelerometer reading). Within each time interval, the three accelerometer values for the x axis, y axis and z axis are stored. This results in an 80×3 matrix. The data must be passed into the neural network as a flat vector of length 240. The first layer in the network must reshape it to the original shape, which was 80×3. The model is tested on the human behavior pose dataset WISDM. We selected 80 % of the data in the WISDM dataset for model training and 20 % for model testing. Using Adam as optimizer, batch size equals to 400, epochs equal to 50 (training for 50 rounds).

CNN Layer 1

The first layer defines 100 filters of height 10. This allows us to train 100 different features on the first layer of the network. The output of the first neural network layer is a 71×100 neuron matrix. Each column of the output matrix holds the weights of one single filter. With the defined kernel size and considering the length of the input matrix, each filter will contain 71 weights. The structure and parameters are shown in Table 1.

Table 1. **The input, condition and output of CNN layer 1**

Modules	Value
Input	Two order tensor (1086393,6)
	dtype: float32
Condition	Weight matrix size [10,100]
	Bias vector [10]
	activation function: Relu
Output	Two order tensor (100,71)
	dtype: float32

CNN Layer 2

The result from the first CNN layer will be fed into the second CNN layer. We will again define 100 different filters to be trained on this level. Following the same logic as the first layer, the output matrix will be of size 62×100. The structure and parameters are shown in Table 2.

Table 2. **The input, condition and output of CNN layer 2**

Modules	Value
Input	Two order tensor (100,71)
	dtype: float32
Condition	Weight matrix size [10,100]
	Bias vector [10]
	activation function: Relu
Output	Two order tensor (100,62)
	dtype: float32

Max pooling layer

A pooling layer is often used after a CNN layer in order to reduce the complexity of the output and prevent overfitting of the data. We chose a size of 3, which means that the size of the output matrix of this layer is only a third of the input matrix. The structure and parameters are shown in Table 3.

Table 3. **The input, condition and output of max pooling layer**

Modules	Value
Input	Two order tensor (100,62)
	dtype: float32
Condition	Weight matrix size [3,64]
	Bias vector [64]
Output	Two order tensor (100,20)
	dtype: float32

CNN Layer 3 and 4

Another sequence of 1D CNN layers follows in order to learn higher level features. The output matrix after those two layers is a 2×160 matrix. The structure and parameters are shown in Table 4.

Table 4. **The input, condition and output of CNN layer 3, 4**

Modules	Value
Input	Two order tensor (100,20)
	dtype: float32
Condition	Weight matrix size [10,100]
	Bias vector [10]
	activation function: Relu
Output	Two order tensor (160,2)
	dtype: float32

Average pooling layer

One more pooling layer to further avoid overfitting. This time not the maximum value is taken but instead the average value of two weights within the neural network. The output matrix has a size of 1×160 neurons. Per feature detector there is only one weight remaining in the neural network on this layer. The structure and parameters are shown in Table 5.

Table 5. **The input, condition and output of average pooling layer**

Modules	Value
Input	Two order tensor (160,2)
	dtype: float32
Condition	Weight matrix size [2,1]
Output	Two order tensor (160,1)
	dtype: float32

Dropout layer

The dropout layer will randomly assign 0 weights to the neurons in the network. Since we chose a rate of 0,5, 50 % of the neurons would receive a 0 weight. With this operation, the network becomes less sensitive to react to smaller variations in the data. It should further increase our accuracy on unseen data. The output of this layer is still a 1×160 matrix of neurons. The structure and parameters are shown in Table 6.

Table 6. **The input, condition and output of dropout layer**

Modules	Value
Input	Two order tensor (160,1)
	dtype: float32
Condition	Rate = 0.5
	Bias vector [64]
Output	Two order tensor (160,1)
	dtype: float32

Fully connected layer with Softmax activation

The final layer will reduce the vector of height 160 to a vector of 6 since we have six classes that we want to predict (Jogging, Sitting, Walking, Standing, Upstairs, Downstairs). This reduction is done by another matrix multiplication. Softmax is used as the activation function. It forces all six outputs of the neural network to sum up to one. The output value will therefore represent the probability for each of the six classes. The structure and parameters are shown in Table 7.

Table 7. **The input, condition and output of fully connected layer**

Modules	Value
Input	Two order tensor (160,1)
	dtype: float32
Condition	Weight matrix size [160,6]
	Bias vector [6]
Output	Two order tensor (6,1)
	dtype: float32

Results

Our approach to classifying the samples of six different classes contained in it, as well as the techniques and methods that we have employed in the proposed methodology. We set a classification model where the provided training samples were used to train the two-channel CNN model, and the rest of the samples were used to test it. The result yields a classification accuracy of 92 % on the test samples. Figure 2. presents the classification accuracies on both the training and testing samples at each epoch.

As seen in the figure, the training accuracy gradually increased with each epoch. The performance of the model was slightly unstable throughout the first 20 epochs, but it became pretty stable afterward.

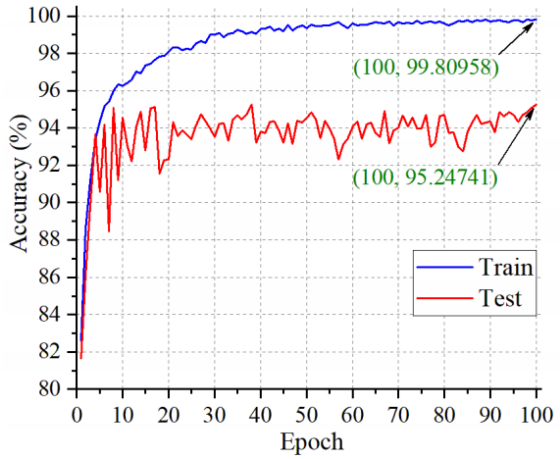

Figure 2. Train and test classification accuracies at each epoch

The confusion matrix provides more details on the output of the classification process. Figure 3. provides the confusion matrix of the epoch of our model for HAR classification. It is apparent that the model works very well while distinguishing six classes (Walking, Upstairs, Downstairs, Sitting, Standing and Jogging) registering over 95 % individual classification accuracies for each class.

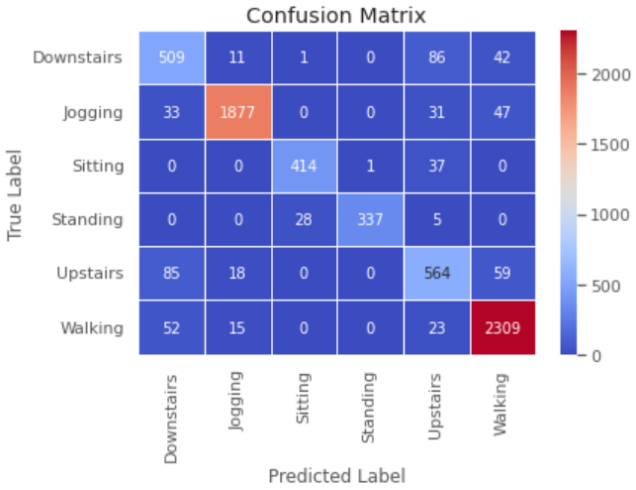

Figure 3. Confusion matrix of the HAR classification

Accuracy: For a given test dataset, the ratio of the number of samples correctly classified by the classifier to the total number of samples is the correct rate for the identified samples:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{7}$$

where TP means True Positives, FP means False Positives, FN means False Negatives, and TN means True Negatives.

The model learns well with accuracy reaching above 92 % and loss hovering at around 0,39.

Conclusion

A CNN-based HAR classification model is proposed in the paper. It is tested on the WISDM dataset. The obtained results yield a 92 % classification accuracy. However, the model can be further modified by tuning specific parameters of CNN and adding more nodes and layers in the CNN architecture. A new set of features can also be extracted and fed in an additional channel of CNN to improve the model’ s performance, which is subjected to future studies.

References

1. Labrador M., Lara Yejas O. // Human activity recognition: using wearable sensors and smartphones. CRC Press, 2013.
2. Fu Y. // Human activity recognition and prediction. Springer, 2016.
3. Wang J., Liu Z. // Human Action Recognition with Depth Cameras. Cham: Springer International Publishing, 2014.