2020 7th International Conference on Electrical and Electronics Engineering

DNN and CNN Approach for Human Activity Recognition

Şeref Recep Keskin
Department of Computer Engineering
Gazi University
Ankara, TURKEY
serefrecepkeskin@gmail.com

Ayşenur Gençdoğmuş
Department of Computer Engineering
Sabahattin Zaim University
Istanbul, TURKEY
aysenur.gencdogmus@izu.edu.tr

Buse Yıldırım
Department of Software Engineering
Sabahattin Zaim University
Istanbul, TURKEY
buse.yildirim@std.izu.edu.tr

Gülüstan Doğan
Department of Computer Science
University of North Carolina Wilmington
North Carolina, NC, USA
dogang@uncw.edu

Yusuf Öztürk
Department of Electrical and Computer Engineering
San Diego University
San Diego, USA
yozturk@sdsu.edu

***Abstract*—One of the common causes of low back pain is postural stress. When sitting or walking, poor posture may result in spinal dysfunction. Increased pressure on the spine can cause tension and spasms in the lumbar muscles and cause low back pain. Monitoring of daily activities becomes more important, especially to help sick and elderly people. Recognition of unstructured daily activities is a more difficult and important task. In this study, we use Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN) to study spinal movement and postural stress through two sensors connected to the pelvis and spine of a healthy subject. Body kinematics data consist of four categories: standing, sitting, walking and other activities. We compared the accuracy of DNN and CNN methods for the identification and labeling of daily activities. We observed the results of deep learning methods with different hyperparameter values and obtained the optimum values.**

Keywords-Human Activity Recognition; Deep Learning; Convolutional Neural Networks; Deep Neural Networks; Signal Processing

I. Introduction

Recognition of human activity has been a widespread research topic. Recording human activities without restricting people's movements is very essential. It is not possible to provide precise information about the frequency of movement of functional activities and the duration of the spine posture in a single session. With all these limitations, we evaluated the patient's motion data. A study at the University of San Diego [1] is performed using two wireless body sensors to monitor the position and movement of the spine.

The study at the University of San Diego has guided our study. Body sensors are located on the pelvis and spine of the subject. It aims to coordinate data from wireless sensors.

Parameters in the dataset; Sequence Number, MAC ID of the sensor, Time Stamp, Position Number, Quaternion X, Quaternion Y, Quaternion Z, Quaternion W. We applied preprocessing operations on the data set (cleaning, filtering, data matching and calculations) as the data captured from the sensors is noisy. Detailed information on these procedures is included in the study [13]. After completing the data preprocessing steps, we applied the CNN method and the DNN method. We used the Python programming language to implement these methods.

We examined the dataset used in the study as four groups of data: standing, sitting, walking and other activities. In this study, we analyzed human activity data obtained from body sensors using CNN and DNN methods. We compared the results of the methods.

In this study, we used DNNs and CNNs to train and recognize the movement type. We tested the batch size and number of epochs used in neural networks for different values. Optimum values were suggested for these hyperparameters. Also, we analyzed and compared the test accuracy, precision, recall and F1 score values of CNN and DNN methods.

II. Related Work

Deep learning methods have recently performed well in many areas with automatic high-level feature extraction. In this study, the current literature in three aspects is summarized, and the sensor method, deep models and applications are investigated. In addition, the latest developments in sensor-based studies based on deep learning in the field o f activity recognition have been considered [1].

New ideas have emerged to address the Human Activity Recognition (HAR) problems that have emerged with the development of deep learning methods. One idea is to propose a deep network architecture that uses bidirectional redundant (incremental) Long Short Term Memory (LSTM) cells [2]. The advantage of this proposed method is that it can combine a two-way connection. These are positive time direction (forward case) and negative time direction (reverse case).

In another study, recurrent neural networks (RNN) with long Short Term Memory (LSTM) are thought to automatically learn and model long-term transient dependencies [3]. The LSTM network proposed in this study is an end-to-end fully connected network. Skeletal-based action is recommended for recognition. The consistency of the proposed model is demonstrated by the results of experiments with three human action recognition data sets.

Other researchers have explored deep, convective and repetitive approaches in three representative datasets. The data sets collected by the wearable sensors contain motion data. They explain how to train repetitive approaches, how to introduce a new regulatory approach, and how the most advanced technology performs better in the state-of-the-art dataset. LSTM networks containing multiple and repetitive unit layers are used for this purpose [4].

A new method has been developed in which information is shared between multiple LSTMs with a new pool layer, with hidden representations of LSTMs corresponding to neighboring trajectories combining in this pool layer [5]. The performance of the method is shown on several general data sets. This model performed 42% better than previous methods.

Researchers investigating deep learning approaches for activity recognition observed the effect of various important hyperparameters on CNN performance, such as core size and number of curved layers [6].

In another study, similar to the two wireless body sensors placed in the pelvis and spine in our study, it is proposed to combine data from two wearable inertial sensors attached to one foot and the waist to assess a person's daily activity [7].

In another study, similar to our study, a DNN algorithm was used for classification [8]. In the approach, subjects were classified into specific activities: walking, standing, sitting and lying down. DNN has been shown to give better results than Decision Tree (DT) and Support Vector Machine (SVM) algorithms. The results also showed that the activities of the participants were classified with an accuracy of more than 98.53% on average.

In a study similar to our study, the use of CNN was proposed to classify human activities [9]. The models in the study used raw data from a series of inertial sensors. By examining the classification potential of single, double and triple sensor systems, the performances of different sensor groups were also compared. Experimental results using five different sensors in the data set of 16 lower extremity activities collected from a group of participants were promising. In our study, a CNN model was used similar to this study. The CNN model we used was applied on body kinematics data obtained from a pair of body sensors.

III. Methodology

This section briefly describes the CNN and DNN model architecture that had been used in our experiments.

A. *CNN Architecture*

CNN is a feedforward neural network that is commonly used for image recognition and object classification [10]. In our study, it is used to recognize the human activity.

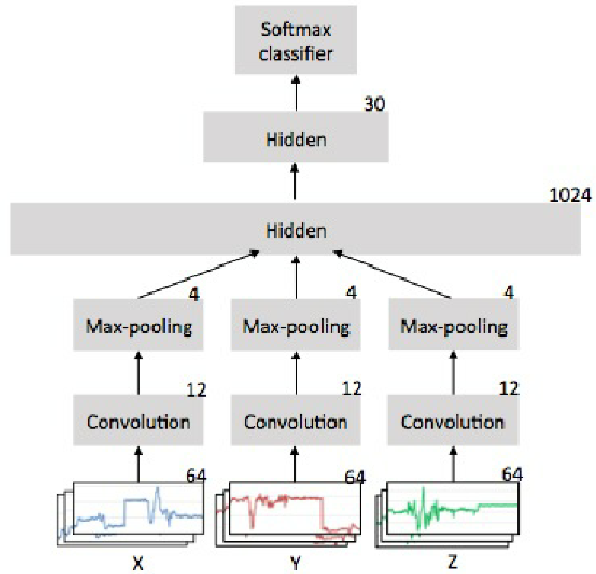

Figure 1. CNN structure for human activity recognition [11]

CNN is a class of artificial neural networks. Therefore, it is composed of so-called neurons that receive a weighted input sum and produce an activity level.

CNN models are mainly composed of three parts: Convolutional Layer, Pooling Layer, and Fully-Connected Layer. It usually takes a 2D (sometimes more dimensions) matrix and gives a result. The CNN processes the data with various layers. Each layer has its function and makes feature inferences and finds hidden patterns. These layers and their purposes can generally be described as follows:

- **Convolutional Layer:** Used to determine properties
- **Non-Linearity Layer:** Introducing non-linearity to the system
- **Pooling Layer :** Reduces the number of weights and controls suitability
- **Flattening Layer:** Prepares data for the classic neural network
- **Fully-Connected Layer:** Standard neural network used in classification

In this study, we use the architecture given in Table I for the design of the convolutional neural network. Also, conv2D, maxpooling2D, dropout, flatten and three dense layers were used for the CNN model.

TABLE I
CNN ARCHITECTURE USED FOR THE EXPERIMENTS

Layer	Description
Conv2D	output dimension: 128, activation: relu
MaxPooling2d	activation: relu
Dropout	rate: 0.2
Flatten	output dimension: 512
Dense	output dimension: 128, activation: relu
Dense	output dimension: 128, activation: relu
Dropout	rate: 0.2
Dense	output dimension:4, activation: softmax

The loss function and optimizer selection are important for successful deep learning models. In this CNN study, we use Categorical Cross Entropy as a loss function and Adam as an optimizer, additionally we selected learning rate as 0.001 and decay select as 1e-6.

B. DNN Architecture

A DNN is an artificial neural network with multiple layers between the input and output layers. DNN finds the correct mathematical methods to convert the input to the output, whether it is a linear relationship or a nonlinear relationship. The network moves between layers that calculate the probability of each output. Each mathematical manipulation is considered a layer.

In addition, a DNN can model complex nonlinear relationships. Extra layers can model complex data with fewer units than a shallow network that has a potentially similar performance.

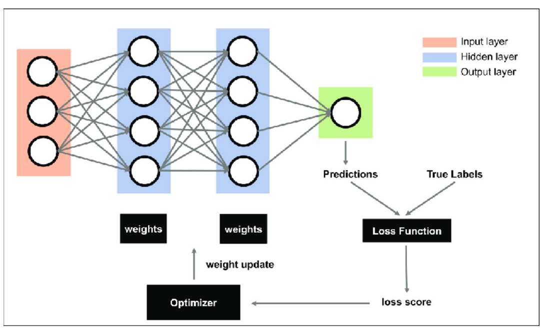

Figure 2. General DNN architecture and workflow [12]

In this study, we used the architecture given in Table II for the design of the DNN. In addition, we used two dropout and three dense layers for the DNN model.

TABLE II
DNN ARCHITECTURE USED FOR THE EXPERIMENTS

Layer	Description
Dense	output dimension: 100, activation: relu
Dense	output dimension: 100, activation: relu
Dropout	rate: 0.2
Dense	output dimension: 100, activation: relu
Flatten	output dimension: 100
Dense	output dimension: 4, activation: softmax

In this DNN study, we used Categorical Cross Entropy as a loss function and Adam as an optimizer.

IV. EXPERIMENTAL RESULTS

In this section, we briefly give the details and results of our work with the two different neural networks. Besides, we describe the data set used in the study.

Experiments were performed on a computer with CentOS operating system with an 8-core CPU and 30 GB RAM.

A. Dataset

In this work, the DNN and CNN are trained to recognize the type of movement (walking, standing, sitting, other activities) based on spinal sensor data. Figure 3 shows the distribution of data by type of activity.

Figure 3. Training examples by activity type

The data to be used for the analytics of body kinematics monitoring data comes from the 9-axis motion processors of the sensors. Sensors with eight differential EMG channels and a 9-axis mothion processor were utitized for studying body kinematics and muscle activity.

One sensor is located in the pelvic region and the other sensor in the cervical spine. The sensors are directly attached to human skin. Data were collected according to people's daily routines. This data is transmitted wirelessly via Bluetooth Low Energy (BLE) to an Android device at a sampling frequency of 100 Hz. Results were stored in an Excel file. The recorded data has noise from both the environment and the sensors. We needed to filter data after mapping and conversion. For this purpose, we edited noisy signals and peak values using a median filter in our previous study [13]. The final dataset consists of user-id, TimeStamp, x-axis, y-axis, z-axis, activity information. Table III shows the features and sample rows stored in the saved Excel file.

TABLE III
SAMPLE VALUES FROM THE OUR HUMAN ACTIVITY RECOGNITION DATASET

User-id	timestamp	x-axis	y-axis	z-axis	Activity
4	1.8	0.2	0.2	-0.1	Walking
8	62.2	9.5	11.5	18.7	Standing
21	3.2	4.3	-0.4	2.0	Sitting
29	37.6	2.3	0.6	-1.1	OtherActivities

In addition, we collected data from different individuals and assigned a unique user-id to each. Figure 4 shows the distribution of data according to the user-id of a person.

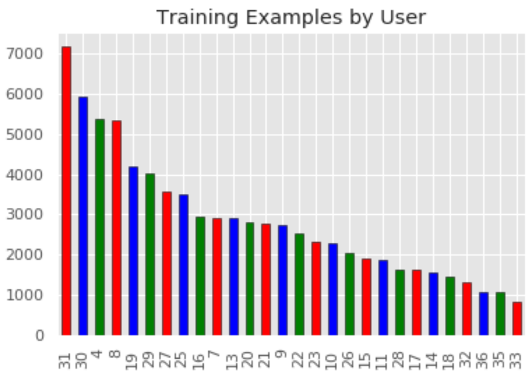

Figure 4. Training examples by user

The Excel file used in the experiments contains 79708 records. These records were divided into a training set, validation set and test set. In this work ratios used are:

- 70% for training set,
- 20% for test set,
- 10% for validation set.

B. Results

Python programming language was used throughout the study. We ran all experiments with the Keras library using a TensorFlow backend.

We have experimented with different batch sizes and number of epochs for DNN and CNN experiments. For batch size, values that fit in RAM such as 16, 32, 64, 128 and 256 were selected. Epoch values of 10, 50, 100, 150 and 200 were used.

We used multiple success metrics for the study. These metrics and their brief descriptions are explained below.

- **Confusion Matrix:** The confusion matrix shows the current state of the data set and the number of true and false estimates of our classification model as a table.
- **Accuracy:** Overall performance of the model.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{1}$$

- **Precision:** How accurate are the positive predictions. It should be as high as possible.

$$Precision = \frac{TP}{FP + TP} \tag{2}$$

- **Recall:** How accurately does the model predict all the positive observations. Must be as high as possible.

$$Recall = \frac{TP}{FN + TP} \tag{3}$$

- **F1 Score:** The F-score helps to measure sensitivity and recall simultaneously. It uses the harmonic mean instead of the arithmetic mean, punishing the extreme values further.

$$F1_{score} = 2 \times (\frac{Recall \times Precision}{Recall + Precision}) \tag{4}$$

Using a DNN architecture, batch sizes of 128 and 256 produced optimal results; using a CNN architecture, batch sizes of 32 and 64 produced optimal results. These results are fully described in the subsection DNN Results and the subsection CNN Results.

*1) **DNN Results**:*

In this subsection, we briefly show the most successful results of the DNN model.

The best results were obtained at 128 and 256 batch size values. Table IV and Table V show the success metrics results with different numbers of epochs.

TABLE IV
DNN MODEL SUCCESS RATES FOR 128 BATCH SIZE

Epoch	Test Accuracy	F1-Score	Recall	Precision
10	77%	75%	73%	78%
50	81%	80%	78%	84%
100	85%	84%	83%	87%
150	86%	85%	84%	87%
200	89%	87%	86%	89%

TABLE V
DNN MODEL SUCCESS RATES FOR 256 BATCH SIZE

Epoch	Test Accuracy	F1-Score	Recall	Precision
10	75%	73%	71%	76%
50	81%	79%	77%	83%
100	83%	82%	80%	85%
150	86%	85%	84%	87%
200	86%	85%	83%	87%

*2) **CNN Results**:*

In this subsection, we briefly show the most successful results of the CNN model.

The CNN model gave the best results at 32 and 64 batch size values. Table VI and Table VII show the success metrics results with different epochs.

TABLE VI
CNN MODEL SUCCESS RATES FOR 32 BATCH SIZE

Epoch	Test Accuracy	F1-Score	Recall	Precision
10	75%	73%	70%	76%
50	81%	80%	78%	82%
100	84%	83%	81%	84%
150	85%	84%	83%	85%
200	87%	86%	85%	87%

TABLE VII
CNN MODEL SUCCESS RATES FOR 64 BATCH SIZE

Epoch	Test Accuracy	F1-Score	Recall	Precision
10	73%	71%	69%	74%
50	81%	80%	78%	82%
100	84%	83%	82%	83%
150	86%	85%	84%	86%
200	86%	85%	84%	86%

Finally, we give details about results of this study. The most accurate result was obtained with DNN architecture by selecting 128 batch size value at 200 epochs. The success rate of 89% is shown in Table IV.

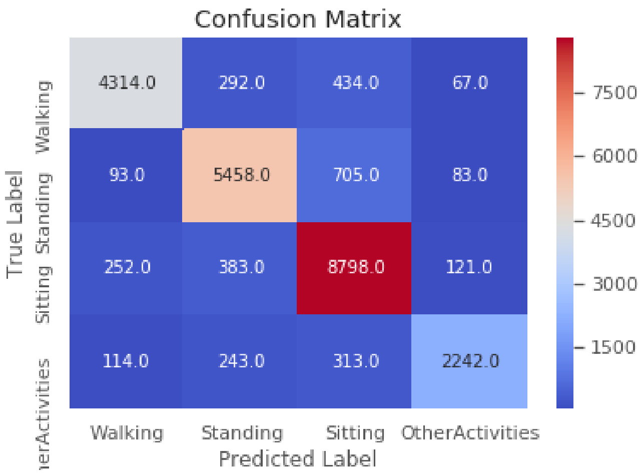

Figure 5. Confusion matrix for the best DNN model

The success of the activity estimation of the DNN model is shown in Figure 5 using the confusion matrix. The model made a successful classification for each activity.

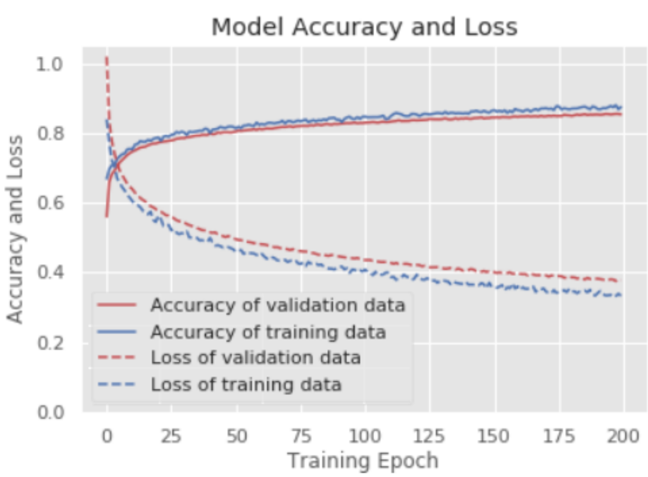

Figure 6. Performance of best DNN model

The behavior of the DNN model during training are shown in Figure 6. The model was not overfitting or underfitting. The training took place at an ideal level. Also, the results of the validation data used to evaluate the success of the model during training are shown in Figure 6.

V. Conclusion

In this study, we used DNN and CNN models for human activity recognition and compared the results. The analyzed data were collected through two wireless sensors attached to a persons's body. Data cleaning, computing Euler angles, matching and filtering procedures have been completed in the previous study [13]. The dataset we used in this study is already processed data. Our dataset consisted of four groups of data: standing, sitting, walking and other activities. Then, we applied DNN and CNN methods on the filtered body kinematics data.

We showed the analysis results of test accuracy, precision, recall and F1-scores of DNN and CNN methods. In conclusion, after comparing the results of DNN and CNN methods, we obtained the highest (89%) success with the DNN method.

We tested the batch size and epoch number of hyperparameters used in DNN and CNN methods with different values. As a result, we showed that the optimum batch size for the DNN method is 128 and the epoch number is 200. In the CNN method, we obtained that the optimal batch size is 32 and the number of epochs is 200.

References

[1] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[2] Y. Zhao, R. Yang, G. Chevalier, X. Xu, and Z. Zhang, "Deep residual bidir-lstm for human activity recognition using wearable sensors," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[3] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, "Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[4] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *arXiv preprint arXiv:1604.08880*, 2016.

[5] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 961–971.

[6] T. Zebin, P. J. Scully, and K. B. Ozanyan, "Human activity recognition with inertial sensors using a deep learning approach," in *2016 IEEE SENSORS*. IEEE, 2016, pp. 1–3.

[7] C. Zhu and W. Sheng, "Human daily activity recognition in robot-assisted living using multi-sensor fusion," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 2154–2159.

[8] H. Amroun, M. Temkit, and M. Ammi, "Dnn-based approach for recognition of human activity raw data in non-controlled environment," in *2017 IEEE International Conference on AI & Mobile Services (AIMS)*. IEEE, 2017, pp. 121–124.

[9] A. Bevilacqua, K. MacDonald, A. Rangarej, V. Widjaya, B. Caulfield, and T. Kechadi, "Human activity recognition with convolutional neural networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 541–552.

[10] R. A. Aral, R. Keskin, M. Kaya, and M. Hacıömeroğlu, "Classification of trashnet dataset based on deep learning models," in *2018 IEEE International Conference on Big Data (Big Data)*, Dec 2018, pp. 2058–2062.

[11] M. Zeng, L. T. Nguyen, B. Yu, O. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," 11 2014.

[12] S. Korkmaz, "Small drug molecule classification using deep neural networks," 03 2019.

[13] A. Gençdoğmuş, R. Keskin, G. Doğan, and Y. Öztürk, "A data-driven approach to kinematic analytics of spinal motion," in *2019 IEEE International Conference on Big Data (Big Data)*, Dec 2019, pp. 2222–2229.