Engineering and Technology Journal
Journal homepage: https://etj.uotechnology.edu.iq

# Tuning the Hyperparameters of the 1D CNN Model to Improve the Performance of Human Activity Recognition

**Rana A. Lateef *, Ayad R. Abbas**

Computer Science Dept., University of Technology-Iraq, Alsina'a street, 10066 Baghdad, Iraq.
*Corresponding author Email: cs.19.77@grad.uotechnology.edu.iq

## HIGHLIGHTS

- How to leverage a 1D single CNN model to produce an excellent performance on the human activity raw data.
- Better results depend on the hyperparameter that has been chosen.
- Tuning the hyperparameter of 1D CNN increased the accuracy of activity recognition.

## ARTICLE INFO

**Handling editor:** Rana F. Ghani

**Keywords:**
Human Activity Recognition.
1D Convolutional Neural Network.
Kernel size.
Hyperparameters
Epochs.
Batch size

## ABSTRACT

The human activity recognition (HAR) field has recently become one of the trendiest research topics due to ready-made sensors such as accelerometers and gyroscopes equipped with smartphones and smartwatches as an embedded devices, decreasing the cost and power consumption. As a result, human activity is considered a time series classification problem. Now a day, deep learning approaches such as Convolutional Neural Network (CNN) have been successful when implemented with HAR to learn automatically higher-order features and, at the same time, work as a classifier. Recently, a one-dimensional Convolutional Neural Network (1D CNN) has been suggested and carried out at the best performance levels in numerous applications, such as the classification of personalized biomedical data and time series classification. This paper studies how to leverage a 1D single CNN model to produce an excellent performance on the human activity raw data. This is done by empirically tuning the values of hyperparameters, such as kernel size, filter maps, number of epochs, batch size, and promoting an advanced multi-headed 1D CNN by employing each convolutional layer with a different kernel size to gain an ensemble–like results. The selected hyper parameter's impact is evaluated on a publicly available dataset named UCI HAR collected from smartphone sensors to perform six activities. A significant determinant of better results depends on the hyperparameter that has been chosen. The results demonstrated that tuning the hyperparameter of 1D CNN increased activity recognition accuracy.

## 1. Introduction

The goals of HAR are to distinguish the behavior done by humans. Recently, the performance of HAR improved with the advent deep learning approach on various benchmark datasets. As smartphone and later smartwatch usage became predominant, research on HAR worked towards taking advantage of the embedded sensors in mobile devices (e.g., accelerometer and gyroscope) to collect signals data instead of on-body sensors [1] [2]. Convolutional Neural Networks (CNNs) are deep neural models created to handle image data. Newly, they have been used for time series prediction [3]. CNNs can be considered feed-forward neural networks (ANNs) with varying convolutional and subsampling layers. The benefits of applying CNNs on HAR compared with other models are the scale-invariant for frequencies and local dependency where the adjacent signals in HAR are correlated. [4]. The tuning or optimization of the hyper parameter is a process to tune a model by changing the parameters to achieve the best results. Hyperparameters or model parameters are considered the important characteristic of a neural network that affects the recognition results.

CNNs now a day are the most public deep neural architecture. This work focuses on the one dimensional Convolutional Neural Network (1D CNN) architecture suitable for implementation with HAR. The hyperparameters in a CNN can compose various components depending on the structure of the neural network. . In this regard, any improvement on the architecture of 1D CNN will increase the performance of the classification. The primary hyperparameters of CNNs are kernel size, number of filters, number of batches, and activation function [5]. This paper investigated the impact of tuning the parameter of the 1D CNN



http://doi.org/10.30684/etj.v40i4.2054
Received 13 March 2021; Accepted 17 June 2021; Available online 25 April 2022
2412-0758/University of Technology-Iraq, Baghdad, Iraq

model applied to the UCI HAR dataset. A simple structure has been selected. One parameter was investigated at a time to easily examine the impact of change in parameters for better improvement of classification. The contribution of this paper is to:

1) Depict the architecture of the 1D CNN model and its hyperparameters.
2) Inspecting the impact of tuning parameters on the model's accuracy to recognize the human activity.

## 2. Related work

[Pasi et al.] Worked on classifying sentences using CNN by various varied parameters such as filer size and patch size impact the CNN performance. They stated that less complex CNN with small adjustments and fine-tuning could produce a considerable performance [6].

[Ronao et al.] Performed CNNs with smartphone sensors data, and they stated that using a large kernel size is useful with single data. In addition, despite the variant of feature complexity level reduced with every additional layer, experiments show that CNN finds more complex and appropriate features with every additional layer [7]. Finally, Koutsoukas et al. compared the performance of Deep Neural Networks (DNNs) by considering a hyperparameter (e.g., dropout regularization, number of hidden layers, activation functions, number of neurons per layer, and learning rate ) with some machine learning methods such as Naïve Bayes, Random Forest, and Support vector machine. The results showed that DNNs outperformed the selected methods [8].

[Nazir et al.] Investigated the effect of parameter tweak on enhancing image classification results by examining one parameter at a time to gain better results. They concentrate on selecting an important hyperparameter for the batch size, momentum, learning rate, and use dropout regularization and batch normalization [9]. Agrawal et al. Studied the impact of CNN parameters (e.g., kernel size and the number of filters) on the recognition accuracy of facial expression. Their work has shown that hyper-parameter selection has a historical impact on network accuracy [10].

## 3. One-dimensional convolutional neural network (1D CNN)

1D CNNs confirmed to have computational advantages to process 1D data. This permits the implementation in real-time and on CPUs to become easier. A 1D CNN model has a hidden convolutional layer. It may succeed by a second convolutional layer with long input sequences and a pooling layer. As shown in eq, the convolutional operation is applied to data vectors by convolving the vector x of length M with the filter w of length L. (1). The result is a one–dimensional output layer C with length [M-L+1] with no zero padding.

$$C(j) = f\left(\sum_{i=0}^{L-1} w(i)\, x(j-1) + b\right), \qquad j = 0,1,\ldots,M-1 \tag{1}$$

Where b is the bias term, and f (.) is a non-linear function called Rectified Linear Unit (ReLU). The Maxpooling layer is followed by each convolutional layer to produce an output vector d by taking the maximum value in a kernel window function u, with size n x 1 and stride s over an input vector C, as in eq.(2)

$$d = \max(\mathrm{u}(\mathrm{n\ x\ 1}, \mathrm{s}))\, C \tag{2}$$

The feature map output from the preceding layer is fed to the fully connected layer and connected to all neuron nodes. The fully connected layer was normalized and scaled using Batch normalization for faster training. The predicted output is specified using the softmax activation function and then used to calculate the cross-entropy loss l, through the network's training, as in eq. (3)

$$l = t(k)\log\big(p(k)\big) \tag{3}$$

Where C is the number of classes, t(k) is the comparison result between the predicted label and the ground truth label [11]. Hence, the HAR problem is considered a time series classification approach. The advantages of 1D CNNs include their compact architecture, cost consumption, capability to work without pre-specified transformations and engineered feature extraction selection, and their ability to use a restricted number of back propagation iterations with a limited set of training data, making them suitable to employ with HAR problem [12]. Figure 1 demonstrates the structure of 1D CNN used to learn data where the data is input as a vector and implements convolution using a 1 x N-type kernel. Consequently, one-dimensional data is extracted and entered into a fully connected layer, whereas the learning is implemented via back-propagation.

## 4. Hyperparameters

Hyperparameter works as knobs that can be tuned during model training. Hyperparameters can be split into two types. The first one specifies the network structure, such as the number of filters, kernel size, which determines the filter size, and the hidden layer between input and output. The other indicates the network training parameters such as the number of epochs (the iterations of the whole dataset that is training to the network), Batch size (the patterns number that appeared in the network before the weights are modified), learning rate (adjusts during weight updated at the end of the batch) [14]. A diagram in figure 2 represented data standardization and the set of hyper parameters used in this paper and applied empirically to the 1D CNN model to improve the classification accuracy.

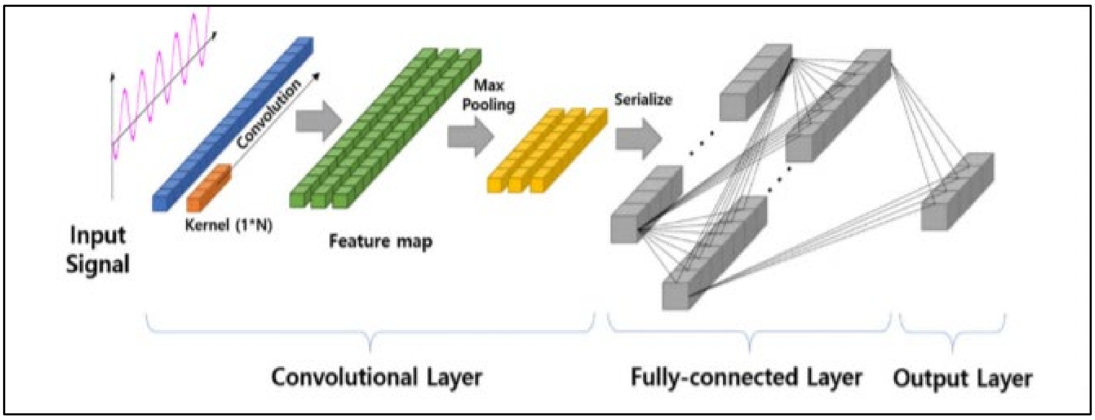


**Figure 1:** The general architecture of 1D CNN. adopted from [13]

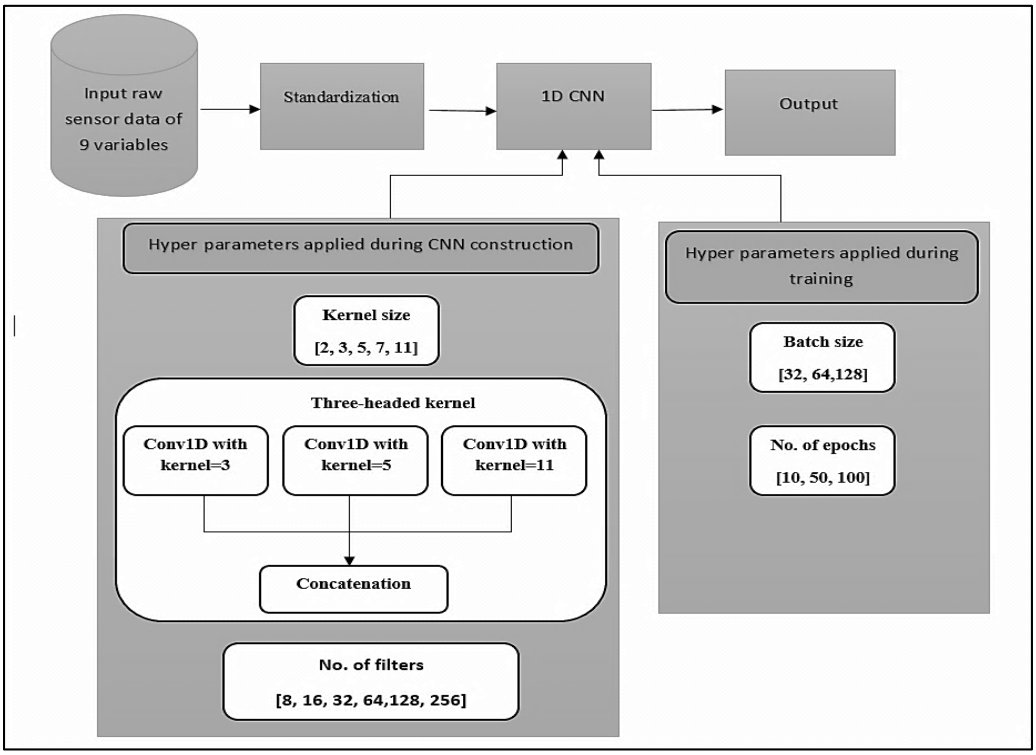


**Figure 2:** Hyperparameters applied with 1D CNN

## 5. Results and Discussion

The experimental process focuses on an arbitrary but sensible selection of hyper parameters configurations to inspect how they affect the performance of the 1D CNN deep neural network. A UCI HAR public available dataset was employed, representing three-dimensional (3D) x and z raw data signals extracted from the accelerometer and gyroscope of a smartphone banded to a waist of a person. The experiments were executed by 30 volunteers aged 19-48 years. Each volunteer performed six activities: Walking, Upstairs, Downstairs, Sitting, Standing, and Laying. The dataset is contained 7,352 train and 2,947 test samples. The signals of sensors have 50 Hz and were preprocessed by applying a noise filter and divided into fixed-width sliding windows of 2.56 seconds (i.e., 128 reading/window). The sensor acceleration signal's gravitational and body motion components were divided into body acceleration and gravity using Butterworth low–pass filter.

The whole dataset consists of three signal types: body acceleration, body gyroscope, and total acceleration. Each has 3 axes represented by x, y, and z. hence, each time step has 9 variables [15]. Simple 1D CNN structures were used, consisting of two successive CNN layers followed by a dropout and max-pooling layer, demonstrated as a model summary in Figure 3. The default value of the number of filters and kernel size was 64 and 3, respectively. The basic set of the three signal data scaled ranged from -1 to 1. The improvement is to standardize the observation before fitting the model, which can be done by shifting the distribution of each variable. Hence, the mean is zero, and the standard deviation is 1. However, the data is divided into the window with a 50% overlap. Therefore, the windowing and overlap were removed to understand the data distribution. Figure 4 shows a plot of 1D CNN as a box and whisker representing a distribution of results without standardization (False) and with data standardization (True). The resultant output from evaluating a CNN with and without data standardization repeated the experiment ten times with two parameters (False for no standardization) and (True for standardization) depicted in Table 1. It shows that standardization leveraged the average accuracy from 90.842% to 91.171%.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, 126, 64)           1792
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 124, 64)           12352
_________________________________________________________________
dropout_1 (Dropout)          (None, 124, 64)           0
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 62, 64)            0
_________________________________________________________________
flatten_1 (Flatten)          (None, 3968)              0
_________________________________________________________________
dense_1 (Dense)              (None, 100)               396900
_________________________________________________________________
dense_2 (Dense)              (None, 6)                 606
=================================================================
Total params: 411,650
Trainable params: 411,650
Non-trainable params: 0
```

**Figure 3:** Model-summary of applying the 1D CNN model

**Table 1:** 1D CNN evaluation accuracy for activity recognition with and without standardization of data

| Iteration no. | Parameter =False (No standardization) | Parameter=True ( standardization |
|---|---|---|
| **#1** | 89.956 | 92.263 |
| **#2** | 90.261 | 91.517 |
| **#3** | 90.159 | 90.668 |
| **#4** | 91.144 | 90.770 |
| **#5** | 91.347 | 89.074 |
| **#6** | 92.365 | 91.958 |
| **#7** | 90.770 | 91.245 |
| **#8** | 90.601 | 90.567 |
| **#9** | 91.720 | 91.890 |
| **#10** | 90.092 | 91.754 |
| | Average= 90.842% (+/-0.752) | Average= 91.171% (+/-0.893) |

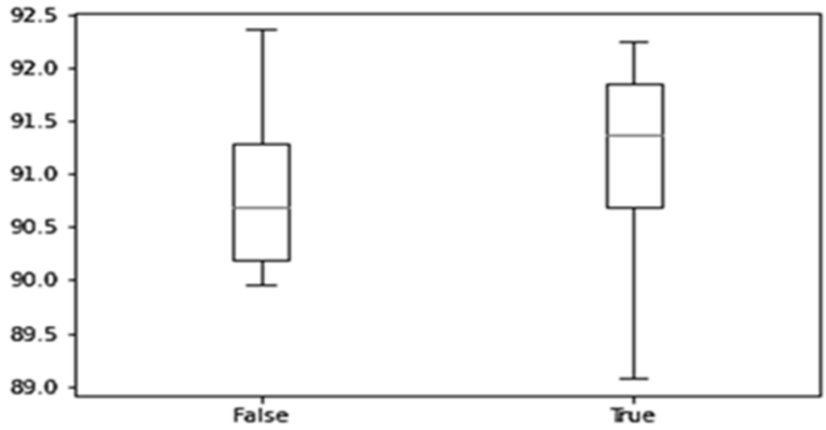


**Figure 4:** Box plot of applying 1D CNN for activity recognition with (parameter=True) and without (parameter=False) standardization

A suite of various kernel sizes was tested with parameters 2, 3, 5, 7, and 11 with ten iterations to evaluate the effect of kernel size used to implement the CNN model. The results stated that a kernel size of 3 might be good with about 90.346% for mean skill, as shown in Table 2. The box and whisker plot of applying the model with different kernel sizes were depicted in Figure 5.

**Table 2:** The evaluation accuracy output of 1D CNN with different sizes of kernel

| Iteration no. | Parameter =2 | Parameter=3 | Parameter =5 | Parameter =7 | Parameter =11 |
|---|---|---|---|---|---|
| #1 | 90.872 | 89.956 | 90.431 | 89.820 | 90.804 |
| #2 | 90.736 | 89.684 | 91.245 | 89.718 | 90.295 |
| #3 | 91.415 | 88.293 | 87.750 | 91.347 | 90.024 |
| #4 | 90.804 | 90.533 | 91.042 | 89.922 | 85.273 |
| #5 | 89.141 | 91.788 | 91.585 | 90.058 | 90.601 |
| #6 | 90.872 | 91.144 | 88.565 | 87.513 | 90.838 |
| #7 | 90.499 | 91.686 | 91.517 | 90.465 | 90.126 |
| #8 | 90.329 | 89.650 | 90.193 | 90.363 | 90.668 |
| #9 | 88.124 | 90.397 | 89.243 | 88.157 | 90.668 |
| #10 | 90.601 | 90.329 | 89.922 | 90.601 | 89.243 |
| Average accuracy | 90.339% (+/-0.924) | 90.346% (+/-0.992) | 90.149% (+/-1.231) | 89.796% (+/-1.087) | 89.854% (+/-1.594) |

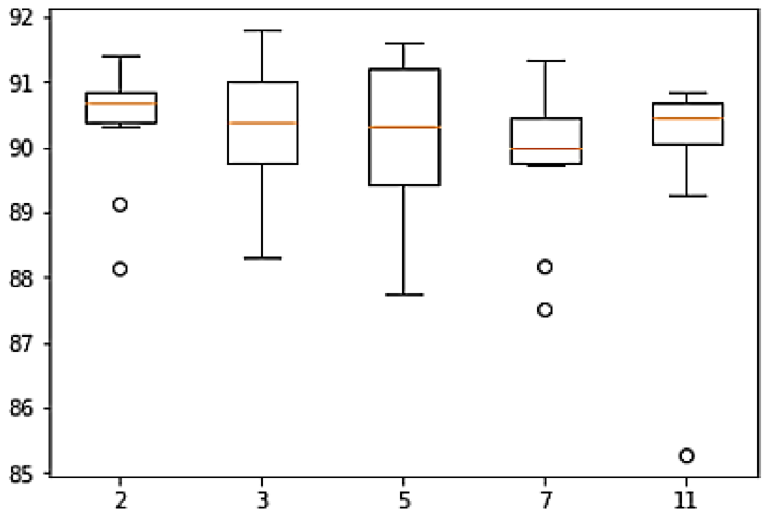


**Figure 5:** Box plot of applying 1D CNN for activity recognition with different kernel size

The kernel size was employed with the multi-headed 1D CNN model, which is done by reading the input time steps using different kernel sizes for each head of the model with the same structure. In this experiment, a three-headed model with three kernel sizes of 3, 5, and 11 that read and interpret a sequence of data at three different resolutions than a concatenation of interpretation from the three-headed model was performed and then interpreted fully – connected layer. The average performance of the model was about 91.4%, with a standard deviation of about 0.8, as shown in Table 3.

Figure 6 represents a plot of the network architecture created with a three-headed where each head of the CNN model stands for one convolutional layer with the same structure, even though the kernel size is varied. This architecture provides a clear idea of how the constructed model fits together.

**Table 3:** The evaluation accuracy output of multi-headed 1D CNN

| Iteration no. | accuracy |
|---|---|
| #1 | 90.499 |
| #2 | 90.499 |
| #3 | 90.363 |
| #4 | 90.736 |
| #5 | 91.653 |
| #6 | 92.297 |
| #7 | 93.078 |
| #8 | 91.110 |
| #9 | 92.365 |
| #10 | 91.449 |
| | Average= 91.405% (+/-0.886) |

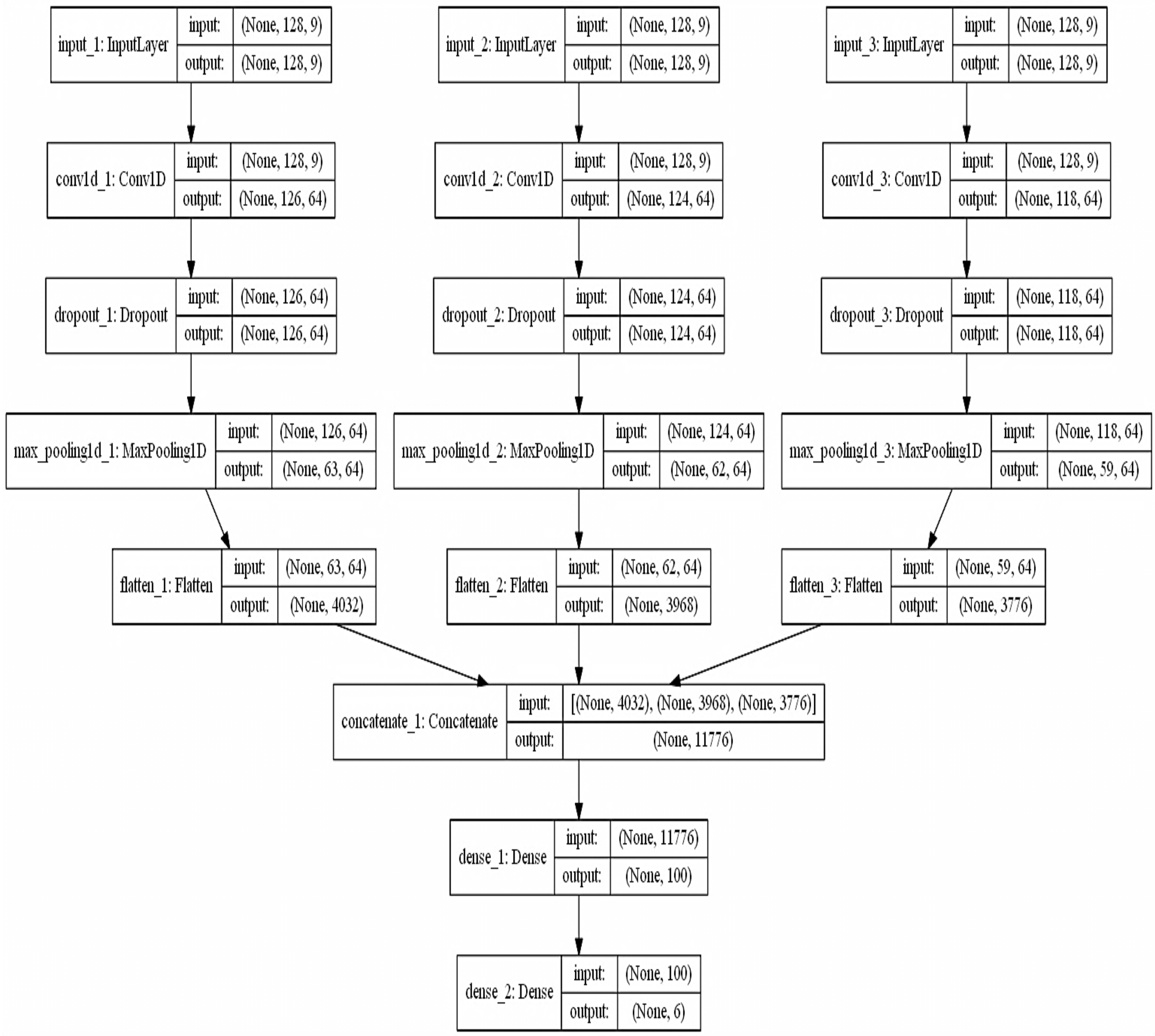


**Figure 6:** Box plot of applying multi-headed 1D CNN for activity recognition

Another hyperparameter that is evaluated is varying the number of filter maps with six various values of 8, 16, 32, 64,128, and 256. The resultant accuracy of each parameter with ten times repetition is depicted in Table 4 with the average accuracy of each filter size. The box plot in Figure 7 shows the average accuracy of each filter size represented by a middle line in a box. The whiskers of the box represent the range of accuracy with each iteration.

**Table 4:** The evaluation accuracy output of 1D CNN with a different filter size

| Iteration no. | Param =8 | Param=16 | Param =32 | Param =64 | Param=128 | Param =256 |
|---|---|---|---|---|---|---|
| #1 | 92.331 | 91.686 | 90.227 | 88.395 | 90.940 | 91.008 |
| #2 | 90.906 | 91.381 | 90.533 | 90.092 | 90.940 | 91.347 |
| #3 | 89.684 | 88.768 | 90.770 | 89.549 | 90.838 | 92.399 |
| #4 | 90.872 | 89.752 | 91.551 | 89.888 | 90.533 | 90.804 |
| #5 | 89.617 | 92.603 | 91.754 | 88.666 | 89.447 | 91.449 |
| #6 | 92.195 | 90.092 | 86.868 | 90.974 | 89.990 | 90.838 |
| #7 | 91.653 | 91.110 | 93.383 | 90.363 | 91.585 | 90.533 |
| #8 | 92.535 | 90.567 | 91.720 | 91.347 | 90.838 | 90.159 |
| #9 | 91.720 | 90.363 | 90.974 | 92.195 | 90.058 | 89.684 |
| #10 | 92.060 | 90.668 | 89.481 | 89.345 | 89.243 | 91.720 |
| Average accuracy | 91.357% (+/-1.001) | 90.699% (+/-1.018) | 90.726% (+/-1.630) | 90.081% (+/-1.125) | 90.441% (+/-0.700) | 90.994% (+/-0.743) |

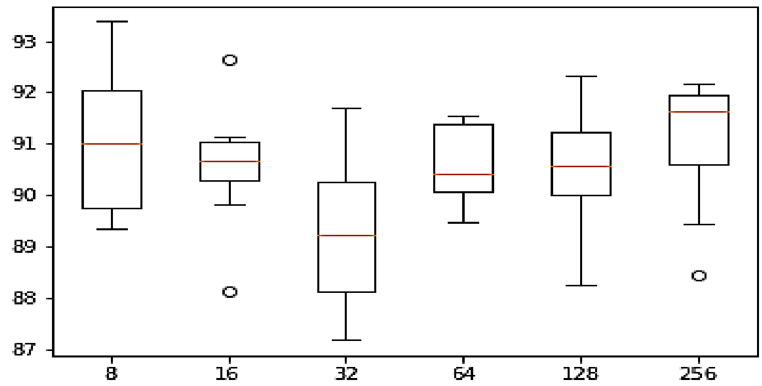


**Figure 7:** Box plot of applying 1D CNN for activity recognition with a different filter size

The effect of changing the number of batches and number of epochs was employed in this experiment. The number of epochs and the batches are tuned to see if a more stable result can be found. The model was trained one time with 10, 50, and 100 with the filter of size 64, the kernel of size 3, batch size of 32, and repetition of 10. On the contrary, the batch size is tuned to 32, 64, and 128 while other parameters remain the same. The average accuracy achieved when tuned epochs to 10, 50, and 100 during model training are 90.550, 91.771, and 92.352, respectively, as shown in Table 5. The effects of using various batches size to train the model on the classification accuracy of HAR are depicted in Table 6.

**Table 5:** The evaluation accuracy output of 1D CNN with various epochs on HAR

| Iteration no. | Epochs =10 | Epochs =50 | Epochs =100 |
|---|---|---|---|
| #1 | 91.992 | 90.668 | 93.451 |
| #2 | 90.906 | 91.822 | 92.026 |
| #3 | 89.243 | 91.754 | 88.836 |
| #4 | 90.702 | 91.347 | 93.078 |
| #5 | 91.042 | 91.822 | 93.451 |
| #6 | 90.940 | 91.754 | 92.094 |
| #7 | 89.006 | 92.297 | 92.535 |
| #8 | 90.838 | 93.587 | 92.840 |
| #9 | 91.245 | 91.076 | 92.433 |
| #10 | 89.583 | 91.585 | 92.772 |
| Average accuracy | 90.550 (+/- 0.906) | 91.771 (+/- 0.741) | 92.352 (+/- 1.262) |

**Table 6:** The evaluation accuracy output of 1D CNN with various batches on HAR

| Iteration no. | Batch =32 | Batch =64 | Batch =128 |
|---|---|---|---|
| #1 | 91.992 | 91.856 | 90.058 |
| #2 | 90.906 | 89.956 | 90.906 |
| #3 | 89.243 | 90.838 | 90.736 |
| #4 | 90.702 | 90.295 | 90.363 |
| #5 | 91.042 | 91.686 | 91.313 |
| #6 | 90.940 | 91.211 | 90.126 |
| #7 | 89.006 | 91.313 | 90.092 |
| #8 | 90.838 | 89.922 | 90.533 |
| #9 | 91.245 | 89.786 | 88.533 |
| #10 | 89.583 | 90.635 | 89.854 |
| Average accuracy | 90.550 (+/- 0.906) | 90.750 (+/- 0.714) | 90.261 (+/-0.688) |

## 6. Conclusion and Future work

The advent of Deep learning improved the HAR concepts and made it the main trend in the research domain. Nowadays, CNN produces promising results. This paper investigated the effect of tuning the hyperparameters of a simple 1D CNN-based HAR applied to the UCI HAR dataset to find the best configuration for these parameters to produce good classification accuracy activities. The best hyperparameters values were derived with 10 experimental repetitions for each parameter. Our study has shown that a prior data standardization lifted the average accuracy from 90.8% to 91.1%. Two procedures were employed with kernel size hyperparameter. These two procedures are either apply different kernel sizes each time or apply a multi-headed CNN model with different kernel sizes for each convolutional layer. The outcome from each head is concatenated. The results show that the applied kernel of size 3 achieved better results with an average accuracy of 90.34%, while utilizing multi-headed leveraged the accuracy to about 91.40%. The filter size of parameter 8 produced optimal results. Increasing the number of epochs plays a significant role in improving the accuracy, and on the contrary, the accuracy decreased with a batch size of more than 64. This work can be further extended by employing another hyper-parameter or evolutionary theory, such as a genetic algorithm or meta-heuristic algorithm for hyperparameter optimization.

## Author contribution

All authors contributed equally to this work.


## Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.


## Data availability statement

The data that support the findings of this study are available on request from the corresponding author.

## Conflicts of interest

The authors declare that there is no conflict of interest.

## References


**[1]** L. Pei , Human behavior cognition using smartphone sensors, Sensors , 13 (2013) 1402–1424. https://doi.org/10.3390/s130201402

**[2]** M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, P. J. M. Havinga, A survey of online activity recognition using mobile phones, Sensors , 15 (2015) 2059–2085. https://doi.org/10.3390/s150102059

**[3]** A. Wang, G. Chen, J. Yang, S. Zhao, C. Y. Chang, A Comparative Study on Human Activity Recognition Using Inertial Sensors in a Smartphone, IEEE Sens. J., 16 (2016) 4566–4578. https://doi.org/10.1109/JSEN.2016.2545708

**[4]** J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, Deep learning for sensor-based activity recognition: A survey, Pattern Recognit. Lett., 119 (2019) 3–11. https://doi.org/10.1016/j.patrec.2018.02.010

**[5]** S. Nazir, S. Patel, D. Patel, Assessing Hyper Parameter Optimization and Speedup for Convolutional Neural Networks, Int. J. Artif. Intell. Mach. Learn., 10 (2020) 17. https://doi.org/10.4018/IJAIML.2020070101

**[6]** K. G. Pasi, S. R. Naik, Effect of parameter variations on accuracy of Convolutional Neural Network, Int.Conf. Computing, Analytics and Security Trends , (2016) 398-403. https://doi.org/10.1109/CAST.2016.7915002

**[7]** C. A. Ronao, S. B. Cho, Human activity recognition with smartphone sensors using deep learning neural networks, Expert. Syst. Appl., 59 (2016) 235-244. https://doi.org/10.1016/j.eswa.2016.04.032

**[8]** A. Koutsoukas, K. J. Monaghan, X. Li, J. Huan, Deep-learning: Investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data, J. Cheminf., 9 (2017) 1–13. https://doi.org/10.1186/s13321-017-0226-y

**[9]** S. Nazir, S. Patel, D. Patel, Hyper Parameters Selection for Image Classification in Convolutional Neural Networks, IEEE Int. Conf. Cogn. Inform. Cognit. Comput. Ber., (2018) 401- 407. https://doi.org/10.1109/ICCI-CC.2018.8482081

**[10]** A. Agrawal, N. Mittal, Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy, Vis. Comput., 36 (2020) 405–412. https://doi.org/10.1007/s00371-019-01630-9

**[11]** I. Mitiche, A. Nesbitt, S. C. Morison, 1D-CNN based real-time fault detection system for power asset diagnostics, IET Gener. Transm. Distrib., 14 (2020) 5766–5773. https://doi.org/10.1049/iet-gtd.2020.0773

**[12]** L. Eren, T. Ince, S. Kiranyaz, A Generic Intelligent Bearing Fault Diagnosis System Using Compact Adaptive 1D CNN Classifier, J. Signal .Process. Syst., 91 (2019) 179–189. https://doi.org/10.1007/s11265-018-1378-3

**[13]** S. H. Kim, Z. W. Geem, G. T. Han, Hyperparameter optimization method based on harmony search algorithm to improve performance of 1D CNN human respiration pattern recognition system, Sensors , 20 (2020) 3697. https://doi.org/10.3390/s20133697

**[14]** S. Gafsi, Convolutional Neural Networks : Hyperparameters tuning and numerical results-A case study , Project Proposal : Convolutional Neural Networks : A case study CS404 / 505 : Convex Optimization for Data Analysis Gafsi Saddam, no. May, (2018).

**[15]** D.Anguita, A . Ghio, L .Oneto, X .Parra, J.L. R. Ortiz, Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine, Lect. Notes. Comput. Sci., 7657 (2012) 216–223. https://doi.org/10.1007/978-3-642-35395-6_30