A Novel Separable Convolution Neural Network for Human Activity Recognition

Ali Boudjema[1] and Faiza Titouna[1]

LaSTIC Laboratory of Computer Sciences, University of Batna2, Algeria
{a.boudjema,f.titouna}@univ-batna2.dz

Abstract. The issue with the time series classification arises in several human applications such as healthcare, industrial monitoring and cybersecurity. Recently, various methods have been developed in order to deal with this matter. In this paper, a novel deep learning-based model for human activity recognition is developed. The proposal examines deeply the training phase in which the acceleration metric is considered by exploring all components of the model. To this end, the architecture of the Convolutional Neural Network (CNN) is studied: a) first, we employ a separable CNN, where we integrate a particular filter model for the depthwise convolution; b) second, we combine the extracted features with the handcrafted features. The proposed classifier is evaluated using a human activity recognition dataset and compared to a set of recent works. The obtained results show that our model outperforms the compared methods under various metrics.

Keywords: Time series Classification · Human activity recognition · Convolutional neural network (CNN) · Separable CNN

1 Introduction

Human activity recognition (HAR) aims to analyze and recognize activities obtained from a sequence of observations. The classification problem based on time series appears in many real applications such as health monitoring, medical care, human-computer interaction, etc. There are two families of time series, video-based, where data are collected using cameras (videos) and sensor-based using smartphones, smartwatches, tablets, MP3 players, or any other digital devices that could detect body movement just by adding specific sensors. The first family requires the recording of body movements with the help of cameras, which presents a significant risk of violating personal data. The quality of the data collected may also be influenced by external conditions (climate, camera quality, lighting, etc.); also, the preprocessing of video requires enormous resources (RAM, CPU, GPU, etc). Meanwhile, sensors are portable , low cost and their data are not influenced by external conditions. Human activities recognition includes four primary applications based on activities [3], covering gestures recognition which aims to recognize hand or face movements. We also cite action recognition that comprises movements and actions of a single person; another

human activity recognition application is interaction recognition that tries to identify actions executed while interacting with an object or another person. The last category regroups the previous classes. It collects data such as wrist-worn accelerometers, gyroscopes and magnetometers. Many other examples of data from real-life applications are represented as a time series, such as biomedical signals (e.g. EEG and ECG), industrial devices (e.g. gas sensors and laser excitation), etc.

In the meantime, exploiting recent methods such as deep learning(DL) has been applied in automatic feature-extraction [28], and achieves a high rate in fields such as computer vision, speech recognition and natural language processing.
The rest of the paper is structured as follows. In the next section, some recent works dealing with the classification of time series are presented. Section 3 covers a wide range of preliminary concepts such as time series, convolutional neural network and feature extraction,In section 4, we describe the proposed architecture based mainly on the separable convolutional neural network model. The experimental results are presented and discussed in section 5. We conclude this work in the last section by relating some perspectives.

2 Related Works

There are significant major categories of time series. The first one is frequency-domain, which includes methods as spectral analysis and wavelet analysis.In contrast, the second is time-domain which contains auto-regression, cross-correlation analysis and auto-correlation methods. Time series classification (TSC) problems are classically solved using model-based, instance-based and feature-based strategies. The first one used algorithms such as the hidden Markov model (HMM) and Auto-regression (AR), in which a model is built for each class by adapting its parameters to this class. The weakness of this approach emerges when it deals with stationary and symbolic non-stationary time series. The second category is based on similarity (dissimilarity) measurement (distance), such as the Euclidean distance-based 1-Nearest Neighbor (1-NN) and Dynamic Time Wrapping (DTW) [21]. This solution is known as computationally expensive. Finally, the feature-based family aims to extract essential features; it includes methods such as the discrete Fourier transform (DFT) [22], the discrete wavelet transform (DWT) [5], singular value decomposition (SVD) [12], and sparse coding [4]. Another family of classification combines a set of classifiers known as ensemble-based. For example, we can We can cite the flat collective of transform-based ensembles (COTE) [17]. These methods need a massive work on preprocessing and feature engineering.

In the last years, CNNs have been exploited to solve the problems of time series classification. Two main approaches are proposed; the first is based on existing (the well-known) CNN architecture [9] that uses 1D time-series signals. Meanwhile, the second approach reshaped 1D time series' signals into 2D matrices then the CNN is applied. The authors in [10] proposed a time-delay neural

network (TDNN) adapted to EEG classification. They used one single hidden layer, which was not able to learn hierarchical features. The convolutional Deep Belief Network(CBDN) was also exploited in [16] to classify audios using the frequency domain. In [30], the authors proposed a multichannel CNN (MCCNN) to deal with multivariate TS. The end-to-end neural network method applies multiple transformations of different scales, sampling rates and frequencies. Then, the authors used convolution operations followed by traditional MLP (Multi layer Perceptron) to classify obtained feature maps. The authors also proposed a pretrained version of MCCNN. This model achieves high accuracy on several real-world data sets. Furthermore, The CNN is also applied to speech recognition within the framework of hybrid NN-HMM mode in [2]. The Multi-scale convolutional neural network for time series classification is presented in [6].
Other papers proposed models such as a Fully Convolutional Neural Network (FCN), a deep multi-layer perceptron network (Dense Neural Network, DNN) and a deep Residual Neural Network on univariate Time series [26].

Recurrent neural networks (LSTM) get involved in human activity recognition and achieves good results. Authors in [27] used bidirectional LSTM by incorporating temporal dependencies.Authors of [29] proposed a deep residual Bidir-LSTM, while later in 2019 [24], another model is created based on LSTM and named it Stacked LSTM network by making a network with two parts. The first one contains a single layer neural network followed by a stack of LSTM cells. In 2020, the authors in [25] evaluate the performance of a set of models (SVM, MLP, CNN, LSTM and BLSTM) and compare the results.

Before describing our proposed model, we first need to give some background to different concepts on time series classification.

3 Preliminaries and Methodology

3.1 Time series

Time series form a sequence of data (measurements) naturally ordered over cycles of time [1]. This kind of data is characterized by high dimensionality and updating continuously. It is divided into two families, univariate and multivariate series. The multivariate time series contain more than one observation. On the other hand, univariate time series is characterized by a single observation. Formally, time series can be represented as :

$$X = \{X_1, X_2, X_3, ..., X_l\}^n \tag{1}$$

Where ℓ is the length of time series and n its rank.
Time-series classification is a learning procedure. This task consists of training a model over a set of samples of times series to each one is associated a label that is the probability distribution over the class values and it is represented as follows:

$$\{X_1, X_2, X_3, ..., X_l\}^n \rightarrow Y^n \tag{2}$$

Where Y^n is the label of time series of rank n.

3.2 Convolutional neural networks

The Convolutional Neural Networks(CNNs) are a powerful family of neural networks. In the simplest neural network known as a multi-layer neural network (MLP), information is propagated through different layers of interconnecting nodes. Through these layers, a non-linear transformation is applied to compute the output of each layer that is expressed in the following equation:

$$y_l = \phi(\sum w_{i_l}.x_{i_l} + b_l) \tag{3}$$

where ϕ corresponds to the non-linearity function applied to the neurons of layer l. The weights and the bias are denoted by w_{i_l} and b_{i_l}. The x_{i_l} is the input time series.
The training process is then performed according to the feed-forward step and the back-propagation step minimizing the global error based on the gradient descent algorithm and adjusting the parameters (randomly initialized weights) of the model. Different loss functions can be used during this process.

In the CNN model, there are two main parts. The first one is considered an extractor of features and it consists of multiple convolutional and pooling layers [15]. The second part is a discriminative layer known as a fully connected layer and defined by a multi-layer neural network.

Feature extraction process In the CNN model, the convolution layer creates a feature map by applying a filter or a kernel to an input. This operation is performed by sliding a filter on the data. Performing several convolutions on the input data leads to different feature maps. Moreover, the padding operation which consists of adding zeros to data, is critical since it avoids shrinking the feature map. [15].

Classification process The latent features issued from each layer are fed into an MLP to perform classification. It takes the feature map of the previous part and mapped it into the output classes. A flatten layer precedes this phase in order to turn the multidimensional feature map into 1D data. All layers usually use the "Relu" activation function (Restricted Linear Unit) defined by $\max(0,\sum w_i.y_i)$ where y_i is the input of each layer. This function allows the model to overcome the vanishing gradient problem and make it learning more efficient [19].

3.3 Separable Convolutional neural networks

Some neural net architectures such as MobileNet [11] use the separable CNN, which allows performing separable convolution spatially or depthwise.

Spatial Separable Convolutions This family of neural networks deals with the spatial dimensions of the input and the kernel by decomposing the latter into two small sub-Kernels. A $N \times N$ kernel will be divided into two kernels of

sizes $N \times 1$ and $1 \times N$, respectively. In other terms, it consists of factorizing the initial matrix defined by the kernel as the product of two rectangular matrices having lower dimensions. More formally, we have:

$$K(i,j) = \sum_i \sum_j K_1(i,1).K_2(1,j) \tag{4}$$

with K_1 and K_2 two smaller kernels which, when multiplied, the original Kernel K is found.
Many works used this strategy such as Flattened networks [13] and Inception models [23].

Depthwise Separable Convolution It is not always obvious to factorize a matrix into two matrices such as defined in equation 4. So, using spatial separable CNN causes troubles during training the model. A Depthwise separable convolution consists of splitting the kernel into two separate kernels known as depthwise and pointwise convolutions. Meanwhile, the conventional CNN applies the kernel on all N channels; the depthwise procedure applies different kernels on each channel individually; as a result, we obtain smaller N outputs with smaller sizes. In the pointwise Convolution, the process consists of applying a kernel of 1×1 and combining the result with the N output obtained by the depthwise phase [14].

3.4 HAR-Model Architecture

In this section, we present our proposed method of learning for time series classification. To categorize activities based on raw data collected using a wearable sensors, we use a separable convolutional network architecture which includes a set of 4 separable convolution layers organized in serial structure such as the output from each previous layer is transmitted as an input to the next layer. A separating layer called max pooling is added between every two convolution layers. Figure 1 illustrates this architecture well.
To achieve our goal, we start the first convolution layer with $1 \times 128 \times 9$ time series sequence and we use 64 filters of size $1 \times 11 \times 9$. A stride of 1 is used . The result in the first layer is $1 \times 128 \times 64$. A Relu as activation function is computed at each separable layer. The performance of our model is improved by incorporating a specific filter of dimension (11,1). This kernel operates at the level of the depthwise convolution layer while the kernel of dimension (1,1) is applied at the level of pointwise convolution, as explained in the previous section.
Compared to the standard convolution operation, the depthwise separable convolution network with a kernel of dimension (11,1) consumes fewer parameters and its computational cost is much lower.
Before introducing the last phase of our architecture, we add a layer block that contains the result of the concatenation operation between the output of the previous layer with handcrafted features. Finally, we use Softmax as an activation function in the last fully connected layer since it is a multiclass classification

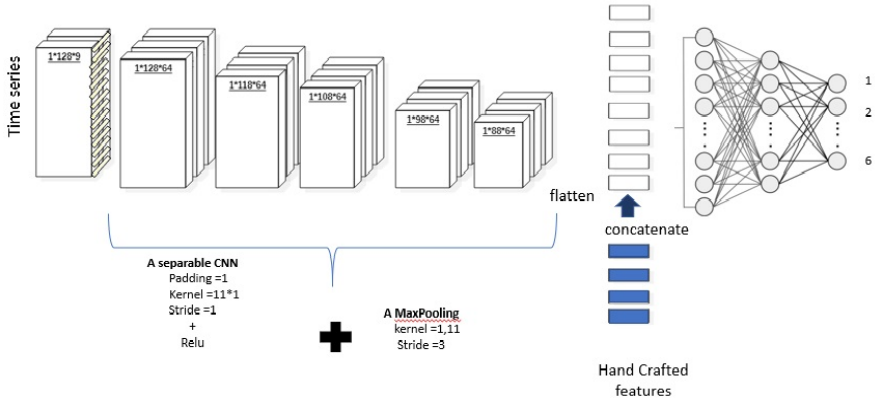

Fig. 1. The proposed architecture

(see equation 5). The output is normalized and corresponds to the probability distribution of learned activity classes.

$$Class = argmax_k(\frac{e^k}{\sum_j^m e^j}) \tag{5}$$

To optimize the error during the learning procedure, we use the loss function defined by categorical cross-entropy and expressed as follows [8]:

$$Loss(x) = -\sum_i^m y_i \log y_i \tag{6}$$

with m is the number of classes given in the dataset.

4 Experimental

4.1 HAR Dataset Description

In our experiments, the data set used is the UCI-HAR provided by [7]; it contains activities performed by volunteers in the age range from 19 to 48, those persons wear a sensor set (Samsung Galaxy S II) on their waist to find out their state (WALKING, WALKING UPSTAIRS, WALKING DOWNSTAIRS, SITTING, STANDING, LAYING). It contains 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz captured using the embedded accelerometer and gyroscope. It includes nine files representing accelerometer and gyroscope signals. The data were labeled manually after being video-recorded and fragmented both part training and testing.

4.2 Experiments Setup

To train this model, we used Keras API [1] and Scikit-learn library [2] on Google Colab environment [3]. The batch size used was 128, the number of epochs was 100, the training dataset contains 7352 experiences for the training part and 2947 for the testing part. As an optimizer, we have used Adam optimizer with its default learning rate equal to 0.001.

For comparison purposes, some evaluation metrics, such as accuracy, precision, recall and F1-score [20], are computed for each model. All these metrics are obtained from the confusion matrix which is a powerful tool for measuring the performance of a machine learning model. Each classification model tries to achieve high performance by correctly predicting the appropriate class for each activity; by testing it, we get four outputs. For example, accuracy counts the ratio of correctly classified data and the error rate that represents the ratio of misclassified data.

5 Results and Discussion

To evaluate the results of our classifier, we used the measurements mentioned in the previous subsection. Indeed, our model aims to classify an activity among six possible activities given in the UCI-HAR dataset. In table 1, we show the results obtained from two models. So, we remark that the presence or the absence of handcrafted features significantly affect the classifiers' performance score. Indeed, the accuracy of the best model reaches a value of 94.77.
The confusion matrix depicted in figure 2 shows clearly the low classification

Table 1. classification report model 1 vs model 2

	Model1(without handcrafted)			Model2(with handcrafted)			
Metrics	Precision	Recall	F1-score	Precision	Recall	F1-score	Support
WALKING	0.74	0.83	0.78	0.92	0.99	0.95	496
WALKING_UPSTAIRS	0.93	0.94	0.82	0.91	0.94	0.92	471
WALKING_DOWNSTAIRS	0.79	0.78	0.78	0.99	0.88	0.93	420
SITTING	0.79	0.87	0.83	0.94	0.93	0.93	491
STANDING	0.80	0.83	0.82	0.94	0.94	0.94	532
LAYING	1	0.95	0.97	1.00	1.00	1.00	537
Accuracy	83.64%			94.77%			2947

error and a high performance for correct class labels.

As can be seen from the figure 3, as the number of epochs increases, the values of both train and test losses decrease. Nevertheless, the accuracy increases

[1] https://keras.io/
[2] https://scikit-learn.org/
[3] https://colab.research.google.com/notebooks/intro.ipynb

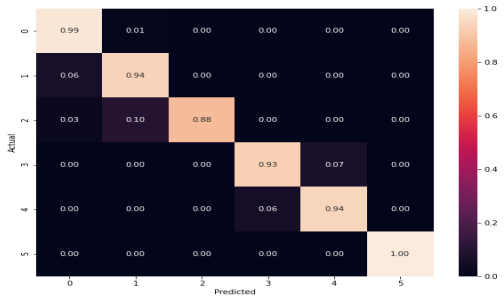

Fig. 2. Confusion matrix

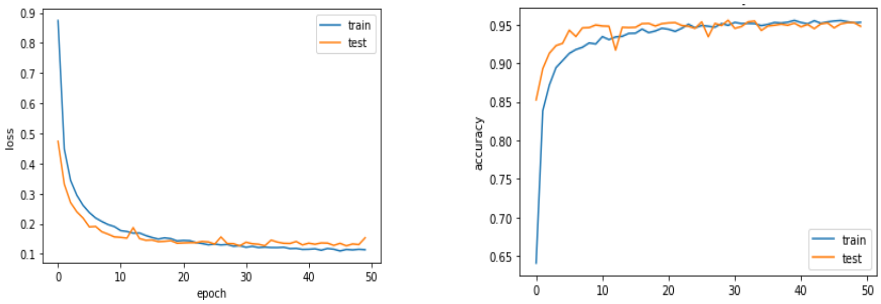

Fig. 3. Loss vs Accuracy curve

and reaches a high performance. In Table 2, we report our results of classification reports for each activity using both models (classical and separable CNN) and we compare a various works in terms of accuracy measure.
Furthermore, the success of classical CNN appears clearly when we handle applications of computer vision, on the other hand, performance is less attractive when we use time series data.
Only one proposed model achieves better performance compared to the other existing models. Indeed, the model1 which used the CNN architecture within a kernel of dimension 11×11, gives an accuracy of about 92.67 %, In contrast, for the second model, which incorporated the kernel of 11×1 in depthwise separable convolution and took in consideration handcrafted features, provides a more exciting result that is 94.77 %. Moreover, we can see clearly the number of reduced parameters in the model2.

6 Conclusion

Time series classification is a challenging problem in particular when we handle activities applications. In this paper, we have proposed a novel architecture of separable convolutional neural networks based on a specific kernel and followed by the handcrafted features concatenation process. Experimental results showed

Table 2. Accuracy comparing of the models for UCI Dataset

Model	Accuracy
CNN [25]	92.71
Stacked Lstm [24]	93.13
Bidir Lstm [27]	93.79
Res Lstm [29]	91.6
Res Bidir Lstm [29]	93.6
Cnn Lstm [18]	92.14

Model	Accuracy	No.Parameters
Model1(cnn modified)	92.67%	6,490,566
Model2(separable)	**94.77** %	6,364,137

that the elaborated classifier outperformed the state-of-art models on the UCI-HAR dataset. The human activity recognition is then achieved with better accuracy. As future work, we first evaluate the model on other HAR datasets then we apply other deep algorithms that can learn features and improve classification performance.

References

1. Bag-of-words representation for biomedical time series classification. Biomedical Signal Processing and Control **8**(6), 634–644 (2013)
2. Abdel-Hamid, O., Mohamed, A.r., Jiang, H., Penn, G.: Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In: 2012 IEEE international conference on Acoustics, speech and signal processing (ICASSP). pp. 4277–4280. IEEE (2012)
3. Aggarwal, J., Ryoo, M.: Human activity analysis: A review (2011)
4. Bahrampour, S., Nasrabad, N.M., Ray, A.: Sparse representation for time-series classification. In: Pattern Recognition And Big Data, pp. 199–215. World Scientific (2017)
5. Chaovalit, P., Gangopadhyay, A., Karabatis, G., Chen, Z.: Discrete wavelet transform-based time series analysis and mining. ACM Computing Surveys (CSUR) **43**(2), 1–37 (2011)
6. Cui, Z., Chen, W., Chen, Y.: Multi-scale convolutional neural networks for time series classification. arXiv preprint arXiv:1603.06995 (2016)
7. Dua, D., Graff, C.: UCI machine learning repository (2017)
8. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data Mining and Knowledge Discovery (2019)
9. Fawaz, H.I., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D.F., Weber, J., Webb, G.I., Idoumghar, L., Muller, P.A., Petitjean, F.: Inceptiontime: Finding alexnet for time series classification. Data Mining and Knowledge Discovery **34**(6), 1936–1962 (2020)
10. Haselsteiner, E., Pfurtscheller, G.: Using time-dependent neural networks for eeg classification. IEEE Transactions on Rehabilitation Engineering **8**(4), 457–463 (2000)
11. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR **abs/1704.04861** (2017)
12. Hui, Z., Tu, B.H., Kawasaki, S.: Wrapper feature extraction for time series classification using singular value decomposition (2005)

13. Jin, J., Dundar, A., Culurciello, E.: Flattened convolutional neural networks for feedforward acceleration (2015)
14. Kaiser, L., Gomez, A.N., Chollet, F.: Depthwise separable convolutions for neural machine translation (2017)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks (2017)
16. Lee, H., Pham, P., Largman, Y., Ng, A.: Unsupervised feature learning for audio classification using convolutional deep belief networks. Advances in neural information processing systems **22**, 1096–1104 (2009)
17. Lines, J., Taylor, S., Bagnall, A.: Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification. In: 2016 IEEE 16th international conference on data mining (ICDM). pp. 1041–1046. IEEE (2016)
18. Mutegeki, R., Han, D.S.: A cnn-lstm approach to human activity recognition. In: 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC). pp. 362–366. IEEE (2020)
19. Nwankpa, C., Ijomah, W., Gachagan, A., Marshall, S.: Activation functions: Comparison of trends in practice and research for deep learning. CoRR **abs/1811.03378** (2018)
20. Powers, D.M.W.: Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation (2020)
21. Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., Keogh, E.: Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping. ACM Transactions on Knowledge Discovery from Data (TKDD) **7**(3), 1–31 (2013)
22. Schäfer, P.: The boss is concerned with time series classification in the presence of noise. Data Mining and Knowledge Discovery **29**(6), 1505–1530 (2015)
23. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning (2016)
24. Ullah, M., Ullah, H., Khan, S.D., Cheikh, F.A.: Stacked lstm network for human activity recognition using smartphone data. In: 2019 8th European Workshop on Visual Information Processing (EUVIP). pp. 175–180. IEEE (2019)
25. Wan, S., Qi, L., Xu, X., Tong, C., Gu, Z.: Deep learning models for real-time human activity recognition with smartphones. Mobile Networks and Applications **25**(2), 743–755 (2020)
26. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: A strong baseline. In: 2017 International joint conference on neural networks (IJCNN). pp. 1578–1585. IEEE (2017)
27. Yu, S., Qin, L.: Human activity recognition with smartphone inertial sensors using bidir-lstm networks. In: 2018 3rd international conference on mechanical, control and computer engineering (icmcce). pp. 219–224. IEEE (2018)
28. Zhao, B., Lu, H., Chen, S., Liu, J., Wu, D.: Convolutional neural networks for time series classification. Journal of Systems Engineering and Electronics **28**(1), 162–169 (2017)
29. Zhao, Y., Yang, R., Chevalier, G., Xu, X., Zhang, Z.: Deep residual bidir-lstm for human activity recognition using wearable sensors. Mathematical Problems in Engineering **2018** (2018)
30. Zheng, Y., Liu, Q., Chen, E., Ge, Y., Zhao, J.L.: Time series classification using multi-channels deep convolutional neural networks. In: International conference on web-age information management. pp. 298–310. Springer (2014)