SRPM–CNN: a combined model based on slide relative position matrix and CNN for time series classification

Taoying Li¹ · Yuqi Zhang¹ · Ting Wang¹

Received: 2 September 2020 / Accepted: 6 February 2021 / Published online: 19 February 2021
© The Author(s) 2021

Abstract

Research on the time series classification is gaining an increased attention in the machine learning and data mining areas due to the existence of the time series data almost everywhere, especially in our daily work and life. Recent studies have shown that the convolutional neural networks (CNN) can extract good features from the images and texts, but it often encounters the problem of low accuracy, when it is directly employed to solve the problem of time series classification. In this pursuit, the present study envisaged a novel combined model based on the slide relative position matrix and CNN for time series. The proposed model first adopted the slide relative position for converting the time series data into 2D images during pre-processing, and then employed CNN to classify these images. This made the best of the temporal sequence characteristic of time series data, thereby utilizing the advantages of CNN in image recognition. Finally, 14 UCR time series datasets were chosen to evaluate the performance of the proposed model, whose results indicate that the accuracy of the proposed model was higher than others.

Keywords Time series classification · Trend extraction · Slide relative position matrix · Convolutional neural networks

Introduction

The objective things develop continuously with the passage of time, and time is an important feature that cannot be ignored in the process of changing things. Information technology is growing rapidly in recent years, and more and more data are being recorded in their respective fields, including astronomy, electronic commerce, government affairs, finance, medicine, transportation, etc. [1]. Most of these data represent the real value of something that changes and grows over time, namely time series data. Therefore, the time series data mining is becoming increasingly prevalent and gaining an importance in the data mining field. Time series classification is a key issue in time series data mining, and is different from the traditional classification due to

the values of the individual feature in sequence. Therefore, the order of data often affects the results of the time series classification [2].

The aim of time series classification is to get a tag for a time series by training a model [3]. Time series classification can assist in obtaining the implicit law hidden behind the data for supporting management decision. For instance, the human behavior can be identified by time series classification model, which can be then adopted to predict target behavior [4, 5]. During decades, a large number of models have been developed for the time series classification [6] and performed good results on the UCR data sets, including the Collective of Transformation Ensembles (COTE) [7, 8], Bag of SFA Symbol (BOSS) [9], INN-DTW [10], etc. Although the 1-Nearest Neighbor (1-NN) algorithm can provide classification outcome and achieves a good performance, it ignores the internal features of the time series. Moreover, the accuracy of these traditional models for time series classification is not good enough, when the dimensions of the time series data are large and the values of data change rapidly. Owing to its greatest strength of extraction of the features from data without any preprocessor, deep learning models, especially the convolutional neural networks (CNN), have become a practical and effective method used widely in many fields, such

✉ Taoying Li
litaoying@dlmu.edu.cn

Yuqi Zhang
zhangyq@dlmu.edu.cn

Ting Wang
wangting1128@dlmu.edu.cn

¹ School of Maritime Economics and Management, Dalian Maritime University, Dalian, China

as image classification, target detection and natural language processing (NLP) [11, 12]. Besides, considering the local and global characteristics of the data, the weights of CNN can be shared to reduce the quantity of the parameters. As a result, the extracted features are closer to the actual situation and can achieve remarkable results in many tasks, which also provides a new way to improve the precision of time series classification. However, in the existing literature on the time series classification with CNN, the original time series data are input directly into the network, which ignores the differences and related information among the different points in the same time series. Meanwhile, only deepening the depth of network helps to increase the consumption of the computing resources, and it does not improve the accuracy of the CNN.

Therefore, a combined model based on the slide relative position matrix and CNN (SRPM-CNN) for time series classification is proposed in the present study. Standard deviation was utilized to extract the trend of the time series data, which not only reduced the dimension of the time series data, but also amplified the fluctuation characteristics. Specifically, the temporal characteristics of the time series data were considered and the original data of time series was converted into images using the slide relative position matrix. This removed some of the redundant features and reduced the dimension of time series data, which minimized the computing time. The converted images were input into the fully convolutional neural network for classification, and the results showed that the accuracy was higher than the relative position matrix with CNN (RPMCNN) [13] in some datasets.

The contributions of this study can be described in three aspects: first, the trend extraction of the time series further enlarged the fluctuation of time series data and obtained the data features that were conducive to the classification results. Second, original time series data were converted into images with the sliding relative position matrix for removing the redundant information and optimizing the relative position matrix. Another important contribution was to construct the CNN, which greatly decreased the number of the network parameters and achieved better results compared with the CNN with the full connection layers.

The research work is organized as follows: In Sect. 2, we have discussed the related work and background. The overall framework of the proposed method has been presented in Sect. 3. Our experimental outcomes are discussed in Sect. 4. In Sect. 5, we have presented the conclusions and future work.

Related work

Due to the values of the time series data that always vary with time, the key issue of the time series classification is to find the changing law of data. Normally, the extraction

of the subsequences from original time series data is done, to obtain their changing rule and then classify these subsequences using the shapelets. Li [14] adopted the standard deviation array, to enlarge the fluctuation of the time series and isolate the optimal shapelets through the key points in the standard deviation array, and then put these optimal shapelets into the SVM (Support Vector Machine) classifier for better classification results. However, the Pruning shapelets with key points ignored the relationship among the different key points, and then extracted and isolated the optimal subsequences that lost some continuity in the information of the original data. Thus, it is essential to pay more attention to the sequence characteristics of the time series data. The traditional classification methods usually put the original data into the classifier after complex preprocessing, but recently, the convolution calculation can identify and extract the key features automatically from the original data and achieve remarkable results for the time series classification.

Since 1995, CNN has attracted the attention of the academic community [15]. Thanks to the advantage of the extracting features, CNN has made a remarkable progress in the speech and image classification, target detection, human behavior recognition and other fields in the last several years. Efforts have been undertaken to apply CNN in the time series classification. Obviously, the structure of CNN is a special structure that is completely different from the artificial neural network (ANN), which comes from the interaction between the neurons in the human brain and can transmit information to each other. The study on time series classification with CNN can be divided into two categories according to the differences on the input data, which are 1D time series data after preprocessing and 2D images transformed from the original time series.

For 1D time series data, the original time series data are input into the Deep Neural Networks (DNN) models for obtaining the results without any preprocessing process. Three DNN models, including the Multilayer Perceptron (MLP) [17], Fully Convolutional Networks (FCN) [16], and Residual Networks (ResNet) [18] are able to classify 1D time series data. Specifically, FCN is remarkable owing to its simple protocol and provides a novel thought for the future research. However, it has a limitation for extracting features only from the time domain and ignoring other factors. To observe the different features of time series data from different perspectives, the multi-scale convolutional neural networks (MCNN) [19] have achieved varied conversion methods of original time series data and classified the one-dimension time series data. The multi-scale features are extracted by one-dimensional convolution, and then combined into the fully convolution network for classification. The long time series data not only lead to low classification accuracy, but also increase some invalid information and the memory consumption.

For 2D images, the original time series data are converted into the 2D images by different transformation methods, and then CNN is employed to classify these images. For instance, Hatami proposed a transformation method based on the Recurrent Plot (RP) [23] to represent the time series data as two-dimensional images. In his study, time series was defined as a number of different periodic behaviors, and the main idea of the RP method was that the trajectory of points appeared already would return to the previous state. But the structure of the CNN in his study consisted two convolution layers and two fully connected layers, which was too simple to affect the accuracy. Wang and Oates paid more attention to the volatility of the time series, and extracted the static and dynamic information. The Gramian Angular Field (GAF) transformed the original time series data into polar coordinates and calculated the transfer probability along the time axis with the Markov Transition Field (MTF). Therefore, they used the GAF-MTF method [20, 21] to convert the time series data into images and made the machine identify it on the vision. In this method, time series data was rescaled through the Min–Max normalization, so that all the values fell into the range of $(-1, 1)$ and then represented the rescaled time series in the polar coordinates before the image conversion phase using the Gramian Angular Field (GAF). The time series was discretized into quantile bins and then its Markov Transition Matrix was calculated for building the Markov Transition Field (MTF) images. The tiled CNN was trained on the compound GAF-MTF images and the results were obtained [22]. The precision of this classification method was not high enough to compete with the other advanced methods. It is for this reason that Chen [13] constructed the relative position matrix for transforming the time series data into grayscale images and input these images into the CNN model based on VGGNet [24]. His method contained five convolution layers and three full connection layers for higher accuracy, but it produced much redundant information and increased the consumption of the computing resources in the process of the transformation matrix. The CNN architecture with a certain complexity improved the accuracy of classification. In recent years,

some classical CNN models, such as VGGNet [24], AlexNet [25], ResNet [18] and GoogLeNet [26] have demonstrated a good performance in the image recognition. The earliest proto structure of CNN was the Lenet-5 proposed by Lecun [27]. These classical networks had five convolutional parts and three full connection layers, and each convolutional part had multiple convolutional layers. Especially, the VGGNet demonstrated better generalization ability for the different tasks.

Therefore, the present work envisaged the development of a combined model based on the slide relative position matrix and CNN for classifying the time series that exhibited a same size and number of convolution kernels as VGGNet and removed the full connection layers for reducing the computational complexity. In the proposed method, standard deviation was adopted to extract the trend features from the time series data, which made the temporal fluctuation characteristics more obvious. Subsequently, the slide relative position matrix was constructed to transform the preprocessed time series data into images. Finally, the CNN without full connection layer was trained on these images and results were analyzed.

Methodology

The overall architecture of the proposed SRPM–CNN model is shown in Fig. 1, which contains three sequential phases, trend extraction phase, image representation phase and convolution network phase.

In the trend extraction phase, a standard deviation window was set to calculate the standard deviation of time stamp, which reduced the dimension of the time series data and extracted the fluctuation trend of data.

In the image representation phase, a slide relative position matrix was employed to convert the preprocessed time series data into 2D image representation.

In the convolution phase, based on the VGGNet without fully connected layer CNN was constructed, which contains

Fig. 1 Overall architecture of the proposed model

five convolution blocks and a softmax layer to get the classification results.

Trend extraction

The standard deviation window is illustrated by Fig. 2. The standard deviation of these points in standard deviation window represents the standard deviation of the center point in window, and the size of window is denoted as w . Assuming the $w=5$, the value of the first point in standard deviation array will be the standard deviation of five points centered on the third point in original array.

A time series is denoted as $T = t_1, t_2, \dots, t_n$, the value of the i th time stamp is denoted as t_i and n is the number of points in this time series. First, an appropriate standard deviation window w should be determined for computing the standard deviation of points in the same window. The first standard deviation array can be defined as Eq. (1).

$$f_i = \sqrt{\frac{(t_{i-w} - \bar{t}_i)^2 + \dots + (t_{i+w} - \bar{t}_i)^2}{2w + 1}}, \quad (1)$$

where $\bar{t}_i$ is the average of points centered on t_i point in the standard deviation window, which is given in Eq. (2)

$$\bar{t}_i = \frac{t_{i-w} + t_{i-w+1} + \dots + t_i + \dots + t_{i+w-1} + t_{i+w}}{2w + 1}. \quad (2)$$

Here, the standard deviation of points in f needs to be computed again, and then the second standard deviation array s is acquired according to Eq. (3):

$$s_i = \sqrt{\frac{(f_{i-w} - \bar{f}_i)^2 + \dots + (f_{i+w} - \bar{f}_i)^2}{2w + 1}}, \quad (3)$$

where $\bar{f}_i$ is the average of points centered on f_i point in standard deviation window, which is given in Eq. (4):

$$\bar{f}_i = \frac{f_{i-w} + f_{i-w+1} + \dots + f_i + \dots + f_{i+w-1} + f_{i+w}}{2w + 1}. \quad (4)$$

The effect of trend extraction with standard deviation window is given in Fig. 3, which indicates the shapes of the raw time series curve, its first standard deviation curve f , and the second standard deviation curve s . Due to the trend extraction of standard deviation, the volatility of original time series data was amplified and the fluctuation trend of time series was highlighted, which makes the curve smoother while it maintains the temporal characteristics. The dimensions of original time series data can reduce from n to m after the task of standard deviation is finished. The relationship between m and n is defined as Eq. (5).

$$m = n - 2w, \quad (5)$$

where m is the length of the second standard deviation array s .

Fig. 2 An example of standard deviation window

Fig. 3 Trend extraction stage

Image representation

It is essential to consider the temporal characteristics, when transforming the second standard deviation array s into image. The correlation between the adjacent data points is stronger than others. Therefore, a $(m/2)*(m/2)$ matrix A is constructed by slide relative position, which represents the position relationship between two points of array s . A is defined as Eq. (6).

$$A = \begin{bmatrix} s_1 - s_1 & s_2 - s_1 & \cdots & s_{(m/2)} - s_1 \\ s_2 - s_2 & s_3 - s_2 & \cdots & s_{(m/2)+1} - s_2 \\ \vdots & \vdots & \cdots & \vdots \\ s_{(m/2)} - s_{(m/2)} & s_{(m/2)+1} - s_{(m/2)} & \cdots & s_m - s_{(m/2)} \end{bmatrix}. \quad (6)$$

Evidently, each point of array s connects to the $m/2$ points before or after itself and this matrix exploits subtraction to denote relative position among them. Each row of A represents the relationship between a particular point and $m/2$ time stamps after it, because they have stronger correlation. Each column of A contains the information of two points with different intervals. Moreover, compared with RPM, matrix A reduces the redundant data and improves the generalization ability, while increasing the data volume, and the computational complexity is reduced to $O(m/2*m/2)$. Subsequently, the Min–Max normalization method is employed to obtain the gray value matrix U defined as Eq. (7).

$$U = \frac{A - \min(A)}{\max(A) - \min(A)} \times 255. \quad (7)$$

To verify the validity of the proposed trend extraction and image representation, some public datasets from the UCR Time Series Classification Archive were chosen as

the samples. Figure 4 shows the images generated from the original time series data of four different datasets with slide relative position matrix, the curves of Fig. 4 are original time series data and the graphs are their corresponding images. Furthermore, these images contain some helpful information of original time series data. For example, the dim part displays the small relative distance between two points and the light part presents the huge relative distance. The frequency of change in color from-bright-to-dim or from-dim-to-bright means the volatility characteristics of the original time series. Specifically, the color changing from dim to bright represents a section of the curve with large fluctuations, and no color change means a smooth phase without drastic fluctuations. Moreover, the overall shape of the image is oblique due to the slide relative position matrix, which is helpful to find the patterns and features embedded in transformed images.

Fourteen datasets were chosen as samples in this study, and the images of different clusters in these datasets are presented in Fig. 5. Two graphs in each sub-graph are graphs converted from time series data in two different clusters of each dataset, which clearly shows the obvious difference between two clusters. For example, the original data of Gun-Point are a smooth time series and the difference between the two clusters is obvious, but the original data of OSULeaf are a serrated time series and have many frequent fluctuations, and two clusters are very similar due to containing more light and shade change in images.

Subsequently, in the convolution phase, a CNN model was adopted to handle the time series classification based on the 2D images representation, which was viewed as an image classification problem.

Fig. 4 Images generated by slide relative position matrix

Fig. 5 Differences between images of different clusters in fourteen datasets

Fig. 6 Overall structure of the proposed model **Table 1** Number of network parameters (Lightning7 dataset)

	CNN with full connection layer	CNN without full connection layer
Parameters	697,844,999	2,714,887

Convolution

The CNN had an outstanding performance in the field of image recognition, and the architecture of classical CNN contained five convolution blocks and three full connection layers. To reduce the complexity of CNN models and consumption of resources, a CNN model containing five convolution blocks and a softmax layer was designed for classifying the images converted from original time series data. The width and height of the input image is related to the length of the time series, which is defined as Eq. (8).

$$D = (n - 2w)/2, \quad (8)$$

where, n is the length of the original sequence, w is the size of the window, D is the size of the input image.

The overall structure of the proposed model in Fig. 6 removed the full connection layer, which reduced the number of parameters. The size of input image changed the flexibly with different length of time series, where the mark “conv” in the parentheses expressed the convolution block and the

numbers after “@” displayed the number of convolution kernel in the present layer, respectively.

To compare the influence of the number of parameters on the networks with or without full connection layer, we performed experiments on the Lightning7 dataset and the results are given in Table 1. Compared with CNN containing two full connection layers with 4096 nodes, the structure of the proposed model reduced the parameters by nearly 99%.

The structure of the proposed model in Fig. 6 was refined into the form of Fig. 7. As shown in Fig. 7, the letter D related to the size of the input images, the CNN model contained five convolution blocks and a softmax layer. Each convolution block had four parts, including the convolution layer, batch normalization (BN) layer, rectification layer and max pooling layer. The convolution layer extracted the features from the input images and obtained the feature maps. Meanwhile, to prevent the edge information of images being lost in the convolution calculation, the input images before convolution layer were padded. The BN layer was used to dispose the gradient vanishing and exploding problem during the back propagation, which avoided the overfitting. The rectification layer adopted a RELU activation function (see Eq. 9) to add the nonlinear factors in this model. The max pool layer reduced the dimension of feature maps produced by the previous layer.

$$\text{RELU} = \max(0, x), \quad (9)$$

where x is the value of the BN layer.

Fig. 7 Details of the CNN structure

After five convolution blocks, the output of the fifth blocks was flattened to a one-dimensional vector, which was then input into the softmax layer that offered the posterior probability of the classification outcomes, in accordance with the true labels of training samples. In this study, the performance of proposed method was evaluated by accuracy (Acc) shown in Eq. (10).

$$\text{Acc} = n/N, \quad (10)$$

where, N is the total number of samples in the test data, and n is the number of samples correctly classified.

Results and discussion

Dataset selection and experiment settings

In the present study, 14 datasets (see Table 2) were downloaded from the UCR Time Series Classification Archive were used to evaluate the proposed SRPM–CNN model. The reasons to choose these datasets are similar to that in reference [28]. The size of training data ranged from 28 to 1000. The length of the training data ranged from 96 to 637. These datasets were standardized to normal distribution in advance.

Table 2 Description of 14 datasets

Dataset	Train size	Test size	Length	Cluster
GunPoint	50	150	150	2
Lightning7	70	73	319	7
Beef	30	30	470	5
OSULeaf	200	242	427	6
OliveOil	30	30	570	4
Coffee	28	28	286	2
CBF	30	900	128	3
Yoga	300	3000	426	2
ECG200	100	100	96	2
Lightning2	60	61	637	2
Fish	175	175	463	7
Trace	100	100	275	4
TwoPatterns	100	4000	128	4
Wafer	1000	6164	152	2

In the proposed SRPM–CNN model, the size of each convolution kernel was set to 3×3 and its stride was fixed to 1 pixel, the size of pooling window was set to 3×3 , and the stride fixed to 2 pixels was exploited to each max pool layer. Furthermore, a 2×2 zero padding method was added before each new convolution layer to prevent the loss of the edge information. The BN layer was after the convolution layer for regulating the processing data, which improved the generalization ability of model and prevented its overfitting. Adam, a stochastic optimization method, was employed to accelerate the training speed. The learning rate was set to 0.000005, which showed a better performance due to the small value of pixel after standardization.

The proposed SRPM–CNN model was implemented on the Tensorflow platform and run on personal computer with Intel Core i5-2000U CPU with 2.2 GHz and 12 GB global memory. The hyperparameters that were input into the model were set to $\{w, 4, 0.000005\}$, which implied the window size of standard deviation, the quantity of batch size and the learning rate during optimizing the weights, respectively. The model was fine-tuned by adjusting the parameters to achieve the better experimental results.

Comparison of models with different standard deviation window

The size of the standard deviation window w had an influence on the final classification results, when it was used to extract the trend of time series. The trend curve of time series data could not be extracted effectively if the small size of window was used for the long time series. On the contrary, some helpful characteristics of the time series data were lost, when the big size of window was used for short time series data. Figure 8 shows the different images with different values of w on the three datasets.

The w was set to a slightly large value for some longer time series data and to a bit small value for some shorter time series data. The length of the time series that extracted trend was within 86–630 (the size of input image D was within 43–315). The influence of standard deviation windows on the loss of different datasets is shown in Fig. 9. Obviously, different w in the model resulted in different decline speeds of the loss for some datasets, including the GunPoint, Beef,

Fig. 8 Effect of standard deviation window on images extracted from different datasets

CBF, Trace, ECG200 and TwoPatterns. However, different w in the model exhibited few effects on loss for some datasets, e.g., the Lightning7, OSULeaf, Coffee, Fish, Lightning2, Yoga, Wafer and OliveOil. As for ECG200 and TwoPatterns, there was a lower loss, when using the original data to generate images.

The influence of different standard deviation windows on the time consumption of different datasets is shown in Fig. 10. Obviously, different w in the model resulted in different consumption of time. For the same dataset, an increase in w resulted in a less time to train the model. For different datasets, the consumption of time was positively correlated with the size and the length of w .

Figure 11 shows the change in the accuracy of different datasets with different values of w in the range of (0, 40). It is obvious that most datasets presented the optimal value of w , which achieved a best performance in this model. But for some datasets, the original data was directly used to construct the SRPM (means $w = 0$) for obtaining better results, since the length of these data is shorter than others.

Comparison of different methods

There are some classical time series classification models, including 1NN-DTW, BOSS, COTE, etc. Meanwhile, the

FCN is baseline model of 1D-CNN methods, which can achieve extraordinary performance through simple proto structure and reduce complexity without any processing in feature selection. The 2D-CNN methods, including GAT-MTF and RPMCNN can transform the 1D time series data into the 2D images. UCR archive has released the accuracies of some models, including TSF, STC and ResNet. The results of the proposed SRPM-CNN model were compared with that of other 13 models in Table 3.

Table 3 shows that the proposed SRPM-CNN model offered better results on most of the datasets. However, its accuracy was lower than that of others in some datasets owing to the different characteristics of the datasets and models. In view of this, some technical findings can be summarized as follows.

- The proposed SRPM-CNN performed better results on datasets with long length of time series data and balanced allocation of training sets and test sets. For example, Lightning7, Beef, OliveOil and Fish had balanced volume of train and test data. Moreover, the length of their original data was longer than that of others. The SRPM could find useful features from a longer time series and highlighted the temporal characteristic. The proposed SRPM-CNN model, not only effectively extracted the

Fig. 9 Effect of standard deviation window on the loss of different datasets

Fig. 10 Effect of standard deviation window on time consumption of different datasets

features, but also paid more attention to the ordering characteristic.

- The accuracy of almost all models on seriously unbalanced dataset was relatively low. For instance, the performances of all models on Yoga were poor, because the numbers of its train data and test data presented a serious imbalance.
- The proposed model performed a slightly low accuracy for datasets with short length of original time series data.

Fig. 11 Effect of standard deviation window on accuracy of different datasets

For example, in CBF and ECG200, some useful information could be lost due to short length of time series, when the proposed model extracted trends with different w and constructed the SRPM.

Conclusions

There are a lot of time series data in our daily life, but the accuracy is often low, when the CNN is applied on them for time series classification. Therefore, a combined SRPM–CNN model for time series classification was proposed in the present study. The proposed SRPM–CNN model consisted of four parts, including the data trend extraction, image representation, convolutional neural networks and performance analysis. Initially, the standard deviation window was adopted to amplify the fluctuation of the temporal characteristics of the time series data. The slide relative position matrix was constructed to code the extracted trend sequences. Subsequently, the corresponding images were transformed from the time series data and input into the full convolutional neural network for image classification. Finally, fourteen datasets were used to verify the performance of the proposed model, and the results showed that the proposed model offered better results than others.

Our future studies will be focused on the data of short time series and the time series classification with multiple features of 2D-CNN.

Table 3 Accuracy of different models on fourteen UCR time series datasets

Dataset	1NN-DTW	BOSS	COTE	FCN	GAF-MTF	RPMCNN	TSF
GunPoint	0.907	1	0.993	1	0.92	1	0.955
Lighting7	0.726	0.658	0.753	0.863	0.74	0.76	0.721
Beef	0.633	0.8	0.867	0.75	0.767	0.87	0.689
OSULeaf	0.91	0.988	0.855	0.988	0.642	0.82	0.643
OliveOil	0.833	0.9	0.9	0.833	0.8	0.9	0.893
Coffee	1	1	1	1	1	1	0.987
CBF	0.997	0.999	0.998	1	1	1	0.972
Yoga	0.836	0.919	0.887	0.845	0.804	0.89	0.866
ECG200	0.768	0.878	0.859	0.985	0.91	1	0.860
Lighting2	0.869	0.852	0.836	0.803	0.886	1	0.764
Fish	0.823	0.989	0.971	0.971	0.886	0.97	0.830
Trace	1	1	0.99	1	1	1	0.992
TwoPatterns	0.904	0.984	1	0.897	0.909	1	0.994
Wafer	0.98	0.999	0.999	0.997	1	1	0.997
Dataset	TS-CHIEF	ROCKET	STC	ResNet	ProximityForest	RISE	SRPM-CNN
GunPoint	1	0.992	0.986	0.991	0.991	0.981	1
Lighting7	0.794	0.798	0.743	0.810	0.792	0.698	0.90
Beef	0.632	0.760	0.736	0.677	0.594	0.742	0.92
OSULeaf	0.974	0.939	0.956	0.975	0.859	0.654	0.88
OliveOil	0.917	0.902	0.879	0.862	0.879	0.893	0.94
Coffee	0.990	1	0.989	0.996	0.992	0.985	1
CBF	0.998	0.996	0.985	0.988	0.994	0.949	0.987
Yoga	0.873	0.914	0.880	0.877	0.887	0.837	0.85
ECG200	0.855	0.899	0.839	0.884	0.873	0.851	0.93
Lighting2	0.769	0.777	0.658	0.801	0.849	0.682	0.81
Fish	0.982	0.974	0.950	0.970	0.934	0.859	0.99
Trace	1	1	1	1	1	0.983	1
TwoPatterns	1	1	0.990	1	1	0.439	1
Wafer	0.999	0.999	1	0.999	0.996	0.995	1

Acknowledgements This research was funded by the National Natural Science Foundation of China (Grant Numbers 51939001 and 61976033), the Liaoning Revitalization Talents Program (Grant Number XLYC1907084), the Natural Science Foundation of Liaoning Province (Grant Number 20180550307), the Key Research and Development Project in Liaoning Province (Grant Number 2020JH2/10100042), and the Fundamental Research Funds for the Central Universities (Grant Numbers 3132019353 and 3132020233).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated

otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Paparrizos J, Gravano L (2017) Fast and accurate time-series clustering. *ACM Trans Database Syst* 42:1–49. <https://doi.org/10.1145/3044711>
2. Bagnall A, Lines J, Bostrom A, Large J, Keogh E (2017) The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min Knowl Discov* 31:606–660. <https://doi.org/10.1007/s10618-016-0483-9>
3. Xi X, Keogh Eamonn, Shelton C, Wei L, Ratanamahatana C (2006) Fast time series classification using numerosity reduction. *Proc. of the 23rd international conference on machine learning*, Pittsburgh, PA, pp 1033–1040

4. Nweke HF, Teh YW, Al-Garadi MA, Alo UR (2018) Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: state of the art and research challenges. *Expert Syst Appl* 105:233–261. <https://doi.org/10.1016/j.eswa.2018.03.056>
5. Wang JD, Chen YQ, Hao SJ, Peng XH, Hu LS (2018) Deep learning for sensor-based activity recognition: a survey. *Pattern Recognit Lett* 119:3–11. <https://doi.org/10.1016/j.patrec.2018.02.010>
6. Baldán FJ, Benítez JM (2019) Distributed FastShapelet Transform: a Big Data time series classification algorithm. *Inf Sci* 496:451–463. <https://doi.org/10.1016/j.ins.2018.10.028>
7. Bagnall A, Lines J, Hills J, Bostrom A (2015) Time-series classification with cote: the collective of transformation-based ensembles. *IEEE Trans Knowl Data Eng* 27:2522–2535. <https://doi.org/10.1109/TKDE.2015.2416723>
8. Lines J, Taylor S, Bagnall A (2018) Time series classification with HIVE-COTE: the hierarchical vote collective of transformation-based ensembles. *ACM Trans Knowl Discov Data* 12:1–35. <https://doi.org/10.1145/3182382>
9. Schäfer P (2015) The boss is concerned with time series classification in the presence of noise. *Data Min Knowl Discov* 29:1505–1530. <https://doi.org/10.1007/s10618-014-0377-7>
10. Kate RJ (2016) Using dynamic time warping distances as features for improved time series classification. *Data Min Knowl Discov* 30:283–312. <https://doi.org/10.1007/s10618-015-0418-x>
11. Matout H, Wang H, Hatori Y, Sato Y, Matsubara K, Wada Y, Tseng CH, Kuriki I, Shioiri S (2019) Convolutional neural networks for humanlike image assessment. *I-Perception* 10:170–170
12. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521:436–444. <https://doi.org/10.1038/nature14539>
13. Chen W, Shi K (2019) A deep learning framework for time series classification using Relative Position Matrix and Convolutional Neural Network. *Neurocomputing* 359:384–394. <https://doi.org/10.1016/j.neucom.2019.06.032>
14. Ling LG, He YW, Da WZ (2019) Discovering shapelets with key points in time series classification. *Expert Syst Appl* 132:6–86. <https://doi.org/10.1016/j.eswa.2019.04.062>
15. LeCun Y, Bengio (1995) Convolutional networks for images, speech, and time series. MIT Press, Publisher, p 3361
16. Wang Z, Yan W, Oates T (2017) Time series classification from scratch with deep neural networks: a strong baseline. In: the 2017 International Joint Conference on Neural Networks, Anchorage, AK, MAY 14–19, pp 1578–1585
17. Rumelhart DE, Hinton GE, Williams RJ (1988) Learning internal representations by error-propagation. *Cogn Sci* 323:399–421
18. Jiang YN, Li Y, Zhang HK (2019) Hyperspectral image classification based on 3-D separable ResNet and transfer learning. *IEEE Geosci Remote Sens* 16:1949–1953. <https://doi.org/10.1109/LGRS.2019.2913011>
19. Cui Z, Chen W, Chen Y (2016) Multi-scale convolutional neural networks for time series classification. <http://arxiv.org/abs/1603.06995>
20. Wang Z, Oates T (2015) Imaging time-series to improve classification and imputation. In: The twenty-fourth international joint conference on artificial intelligence. Buenos Aires. ARGENTINA, JUL 25–31, pp 3939–3945
21. Wang Z, Oates T (2015) Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. *Proc. Of the Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*
22. Le QV, Ngiam J, Chen Z (2010) Tiled convolutional neural networks. *Proc. of the 24th Annual Conference on Neural Information Processing Systems*, 1: 1279–1287
23. Hatami N, Gavet Y, Debayle J (2017) Classification of time-series images using deep convolutional neural networks. In: the 10th International Conference on Machine Vision (ICMV 2017), Vienna, AUSTRIA, pp 226
24. Mateen M, Wen JH, Nasrullah, Song S, Huang ZP (2019) Fundus image classification using VGG-19 architecture with PCA and SVD. *Symmetry-Basel*. <https://doi.org/10.3390/sym11010001>
25. Zhu L, Li ZB, Li C, Wu J, Yue J (2018) High performance vegetable classification from images based on AlexNet deep learning model. *Int J Agric Biol Eng* 11:217–223. <https://doi.org/10.25165/j.ijabe.20181104.2690>
26. Bai J, Jiang HY, Li SQ, Ma XQ (2019) NHL pathological image classification based on hierarchical local information and GoogLeNet-based representations. *Biomed Res Int*. <https://doi.org/10.1155/2019/1065652>
27. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86:2278–2324. <https://doi.org/10.1109/5.726791>
28. Li DY, Bissyandé TF, Klein J, Le Traon, Y (2016) DSCo-NG: a practical language modeling approach for time series classification. *Proc. 15th International Symposium on Intelligent Data Analysis (IDA)*: 9897:1–13. https://doi.org/10.1007/978-3-319-46349-0_1

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.