Pattern Recognition 123 (2022) 108385

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Multi-scale signed recurrence plot based time series classification using inception architectural networks

Ye Zhang, Yi Hou*, Kewei OuYang, Shilin Zhou

College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China

ARTICLE INFO

Article history:
Received 2 December 2019
Revised 12 September 2021
Accepted 20 October 2021
Available online 22 October 2021

Keywords:
Time series classification
Multi-scale
Signed
Recurrence plots
Inception network

ABSTRACT

Inspired by the great success of deep neural networks in image classification, recent works use Recurrence Plots (RP) to encode time series as images for classification. RP provide rich texture information and construct long-term time correlations, which are effective supplements to the networks. However, RP cannot handle the scale and length variability of sequences. Moreover, RP have serious tendency confusion problem. They cannot represent the upward and downward trends of sequences effectively. In addition to the defects of RP, existing time series classification (TSC) networks cannot adapt to the various scales of discriminative regions of time series effectively. To tackle these problems, this paper proposes a method, named MSRP-IFCN. It is composed of two submodules, the Multi-scale Signed RP (MSRP) and the Inception Fully Convolutional Network (IFCN). MSRP are proposed to handle the defects of RP. They comprise three components, namely the multi-scale RP, the asymmetric RP and the signed RP. We first use the multi-scale RP to enrich the scales of images. Then, the asymmetric RP are constructed to represent long sequences. Finally, the signed RP images are obtained by multiplying the designed sign masks to remove the tendency confusion. Besides, IFCN is proposed to enhance the existing TSC networks in multi-scale feature extraction. By introducing the modified Inception modules, IFCN obtains extensive receptive fields and better extracts multi-scale features from the MSRP images. Experimental results on 85 UCR datasets indicate the superior performance of MSRP-IFCN. The visualization results further demonstrate the effectiveness of our method.

1. Introduction

In the era of big data, our daily lives constantly produce a large number of time series data, which should be properly managed [1]. Among the various time series analysis tasks, time series classification (TSC) is likely to be the most fundamental one. This task assigns a certain category to an unlabeled sequence [2,3]. TSC has raised wide attention in extensive academic and industrial fields [1,4], e.g., activity recognition [5], medical diagnosis [6,7], speech signal processing [8] and fault detection [9], etc.

Inspired by the great success of Convolutional Neural Networks (CNNs) in image classification [10–12], some recent works [13,14] encode time series as images, and then use CNNs to classify these images. Among the representative sequence-to-image methods, the Recurrence Plot based methods achieve impressive performance [15,16]. These methods have sparked interest in TSC communities.

Recurrence Plots (RP) are originally used as a visualization tool to reveal the recurrence property of sequences in dynamical systems [17,18]. The main idea is to reveal in which points some trajectories return to a previous state. The formal definition of RP is provided in Section 3.2.1. Recently, due to their graphical nature of exposing hidden patterns and magnifying discriminative regions [19,20], RP are combined with TSC networks to obtain better classification performance. Besides, RP construct long-term time dependencies critical to time series data, handling the defects of CNNs in this respect [19]. In this paper, we incorporate RP and latest TSC networks in one framework. However, there are two major challenges that hinder the implementation, which are summarized as follows.

The first challenge is that, RP cannot handle the variability of discriminative region scales and lengths of sequences, as well as the tendency confusion problem, limiting their further application. To intuitively illustrate these problems, Fig. 1 presents four

* Corresponding author.
E-mail addresses: zhangye18@nudt.edu.cn (Y. Zhang), yihouhowie@gmail.com (Y. Hou).

https://doi.org/10.1016/j.patcog.2021.108385

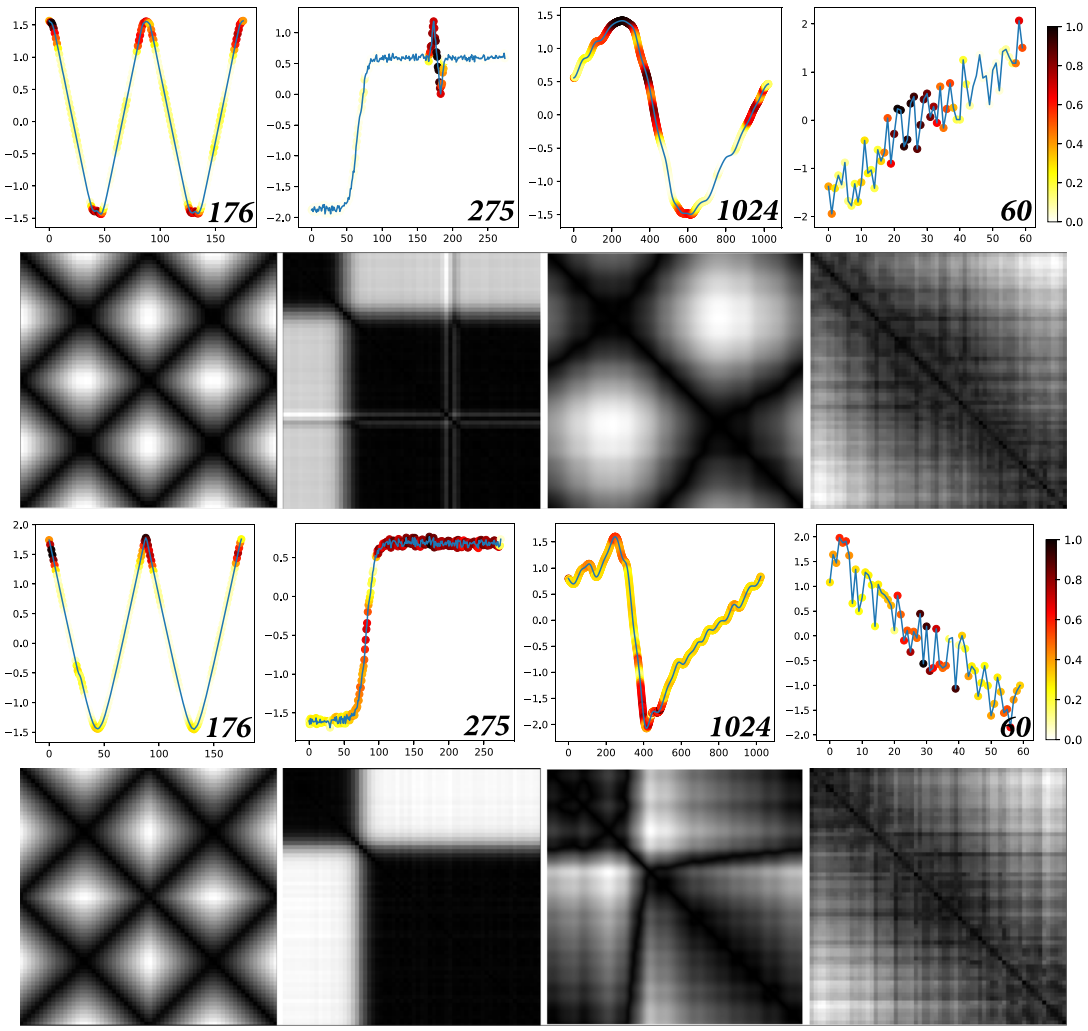


Fig. 1. Four pairs of sequences (first and third row) and their corresponding RP images (second and last row) from the 'Adiac' (left column), 'Trace' (middle left column), 'StarLightCurves' (middle right column) and 'SyntheticControl' (right column) datasets. Each pair corresponds to two different categories. The lengths of these sequences are provided at the bottom right of the figure. The dark colored dots on the curves are the discriminative regions, indicating the contributive areas of their ground-truth category, e.g. the sharp jitter in the second sequence of the first row. It can be observed that these discriminative regions range from tiny waves (first column) to local (second and third column) and global shapes (last column).

pairs of sequences and their corresponding RP images from different datasets [4]. It can be observed that, the lengths of different pairs of sequences vary greatly, and their discriminative regions range from tiny waves to overall shapes. Although the existing methods handle such variability by adjusting the sizes of images [15,16,20,21], the computation costs limit the image sizes in a small range, leaving the problem still intractable. Besides, two sequences with totally opposite tendencies are presented in the last column of Fig. 1. However, their RP images are difficult to distinguish. This is because, RP suffer from serious tendency confusion problem. They cannot represent the upward and downward trends of sequences effectively (see Section 3.2.4 for more details). To handle these issues, this paper comprehensively improves RP to propose multi-scale signed RP (MSRP), which better represent the sequences with limited image sizes. MSRP are composed of three components, namely the multi-scale RP, the asymmetric RP and the signed RP. We first introduce the phase space dimension and time delay embedding of RP to construct multi-scale RP, which are used to enrich the scales of images. Then, asymmetric RP are designed to encode long sequences as images. In these ways, the effects of the scale and length variability of sequences are alleviated effectively. Finally, the signed RP images are obtained by multiplying the designed sign masks in order to remove the tendency confusion.

The second challenge is that, the multi-scale characteristic of time series poses difficulties for the classification networks. Although different lengths of time series are represented as images with a small range of sizes, the multi-scale challenge still exists. As shown in Fig. 1, the discriminative regions of different image pairs are very different in scale, challenging the multi-scale feature extraction abilities of networks. However, existing TSC networks [15,16,22] cannot handle such large scale variabilities effectively due to their relatively fixed receptive fields [23]. To address this problem, this paper first modifies the Inception module of Inception-v4 [12]. Then, a novel network named the Inception Fully Convolutional Network (IFCN) is proposed, which is consisted of the modified Inception modules. Compared with other TSC networks, IFCN better extracts multi-scale features from the MSRP images. Consequently, the scale variabilities of these images are handled effectively.

The contributions of this work are summarized as follows.

1) We find an effective transformation representation for TSC by proposing MSRP. Compared with RP, MSRP better adapt to the variability of discriminative region scales and lengths of sequences, as well as tackle the tendency confusion problem.
2) We handle the multi-scale problem of TSC by proposing an Inception architectural network, IFCN. IFCN outperforms other TSC networks in terms of extracting multi-scale features from MSRP images.
3) Our proposed MSRP-IFCN achieves the state-of-the-art performance on 85 UCR datasets. Each component of MSRP-IFCN is validated through extensive ablation experiments. Moreover, the visualization experiments also demonstrate the effectiveness of our method.

2. Related work

Considering the recent success of deep neural networks (DNNs) in TSC [3], we separate existing TSC methods into traditional methods and DNN-based methods. These methods will be briefly introduced in Sections 2.1 and 2.2, respectively.

2.1. Traditional methods

Traditional methods fall into three popular categories [2,22]: distance-based methods, feature-based methods and ensemble-based methods, respectively.

Distance-based methods utilize the similarity measures between sequences to perform classification. These methods first align the testing samples with training samples through the similarity measures. Then, the similarity distances between these samples are calculated for classification. The most commonly used distance measures for TSC are the dynamic time warping (DTW) measure [24] and its improved versions [25–27]. These DTW-based measures alleviate the effects brought by the noise and warping of sequences. However, they are time consuming and very sensitive to phase shifting, especially for long sequences [2].

Feature-based methods extract hand-crafted features from sequences to perform classification. There have been many feature extraction methods for TSC, e.g. the shapelet-based methods and the dictionary-based methods. The shapelet-based methods [28–30] first generate a set of discriminative subsequences named shapelets. Then, the generated shapelets are used for feature extraction by calculating their similarity distances with the training samples. The dictionary-based methods [31–33] map each time series to the word vectors through a bag of words named dictionary. The extracted word vectors are then used for classification. The feature-based methods reduce the dimensions of sequences effectively. Therefore, it is convenient for the application of some advanced classifiers. However, the representation ability of hand-crafted features is weak, limiting the classification performance of these methods.

Ensemble-based methods [34–36] integrate multiple kinds of features and classifiers in one framework for better performance. Among them, the Collective of Transformation-based Ensembles (COTE) [35] are composed of 35 different classifiers and various kinds of transformation representations. Moreover, the Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COTE) [36] further extend COTE by adding two more classifiers and transformation representations. This method is considered as the state-of-the-art method on the UCR time series archive. Though ensemble-based methods usually achieve very impressive performance, they are complicated, computationally expensive and time-consuming. Consequently, it is difficult to apply these methods in the real word.

2.2. DNN-based methods

Recently, the DNN-based methods have been widely explored in TSC. The DNNs handle the aforementioned weaknesses of traditional methods, due to their advantages in phase shift invariance and feature extraction ability. Therefore, these methods have shown inspiring performance. In this section, existing DNN-based methods are classified according to whether the networks are applied on the raw sequences or the transformation representations.

The methods in Cui et al. [37], Le Guennec et al. [38] are considered as the earliest DNN-based works for TSC. These methods first augmented the training data through the down-sampling, smoothing and slicing operations. Then, a CNN with multiple convolutional, fully connected and pooling layers was trained to perform classification. In [22], a strong baseline was proposed. This method includes three classical DNN, which are the Multilayer Perceptron (MLP), the Fully Convolutional Network (FCN) and the Residual Network (ResNet), respectively. Among them, FCN and ResNet are widely regarded as the baseline networks for TSC [3]. These networks avoid the complicated transformations by directly working on the raw sequences, bringing convenience for their application. However, some discriminative features hidden in the time domain are difficult to extract, limiting the performance of these methods.

Some works map the sequential data to transformation domains, where the hidden features can be extracted more easily. In [39], the learnable wavelets were introduced to decompose time series as subsequences of multiple frequencies. These subsequences were then fed into FCN for feature extraction and classification. However, the features of different frequencies interfere with each other, leading to unsatisfactory performance. Some works transform time series as images for classification. In [40,41], the Gramian Angular Fields (GAF) and Markov Transition Fields (MTF) were used to encode sequences as multi-channel images. Then, a Tiled CNN was trained on these images for classification. In [15,16], time series were transformed as images through RP. Then, a simple multi-layer CNN was applied to classify these images. Similarly, in Chen et al. [13], the signals collected from the manufacturing process of anchor chains were encoded as binary images through threshold RP (TRP). Then, a CNN was trained to classify these images for anomaly detection. In [42] and [14], the signals collected from the drawing process of Parkinson patients were transformed as images through RP and Fuzzy RP, respectively. Then, the AlexNet [43] was applied to classify these images for early Parkinsons disease identification. The aforementioned transformed images bring critical texture information to TSC. However, due to the computation costs, the sizes of these images have to be limited. Therefore, it is difficult for these images to adapt to the scale and length variability of sequences. Besides, the aforementioned TSC networks cannot effectively handle the various discriminative region scales of time series, due to their relatively fixed receptive fields. Consequently, the performance of these networks is limited.

3. Proposed method

3.1. Overview

This subsection briefly introduces our proposed method. The method consists of two stages. In the first part, MSRP are used to encode time series as images. Compared with RP, MSRP handle the tendency confusion issue, and better adapt to the scale variability of time series. In the second part, the MSRP images are fed into IFCN for classification. IFCN extracts multi-scale features effectively utilizing the modified Inception modules. The framework of our proposed method is shown in Fig. 2.

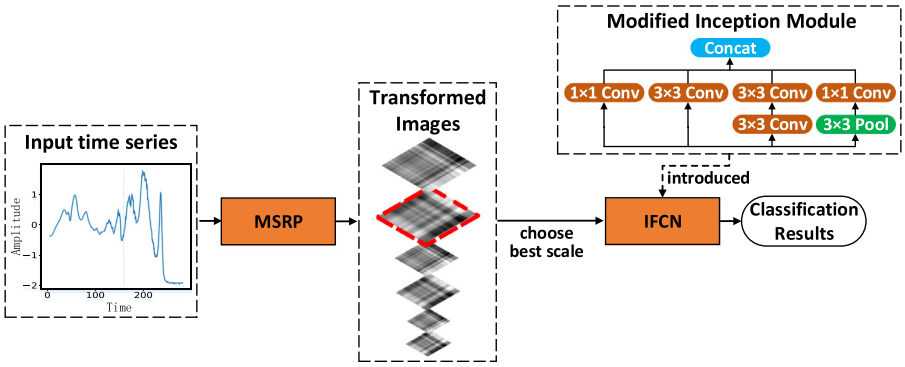


Fig. 2. The framework of our proposed method.

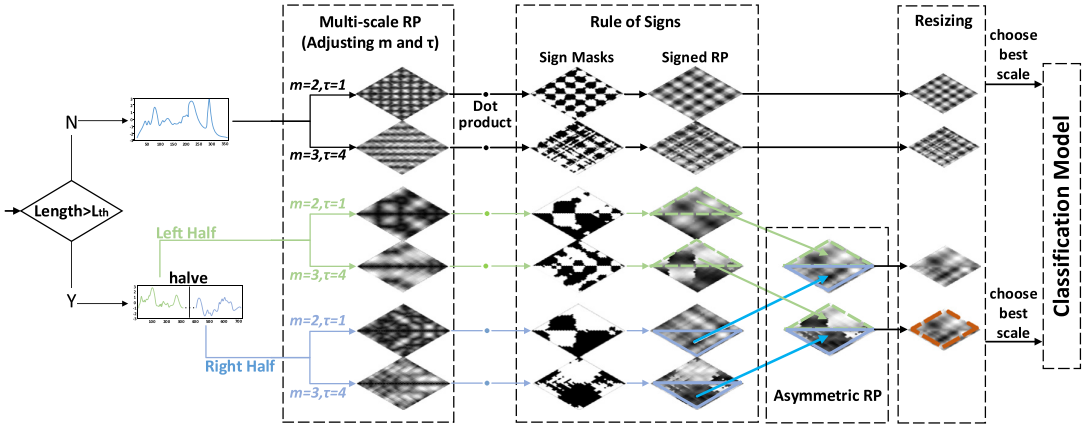


Fig. 3. The flow of MSRP.

3.2. Proposed MSRP for encoding time series

In this subsection, the proposed MSRP are described in detail. First, a review of RP is presented. Then, the components of MSRP, including the multi-scale RP, the asymmetric RP and the signed RP, are described separately. The flow of MSRP is shown in Fig. 3.

3.2.1. Review of RP

RP are originally proposed as a visualization technique to analyze the periodical characters of time series in the complex dynamic systems [18]. They are introduced to TSC to encode time series as images. Concretely, a sequence is first mapped into the multi-dimensional phase space by the sliding subsequences. Each subsequence corresponds to a state in the phase space. Then, a recurrence plot is obtained by calculating the distances of all the states in the phase space. Eq. (1) defines a recurrence plot formally.

$$RP_{i,j}(\epsilon) = \begin{array}{l} \Theta(\epsilon - \|\vec{x}(i) - \vec{x}(j)\|), \\ \vec{x}(\cdot) \in \mathfrak{R}^m, i, j = 1, \ldots, N, \end{array} \tag{1}$$

where $RP_{i,j}$ is a pixel in a recurrence plot, $\vec{x}(i)$ is the i-th state in the phase space, as well as the subsequence sampled at the i-th position of the time series, $\|\cdot\|$ is the norm operation, Θ is the Heaviside function used to binarize the distance matrix through a threshold ϵ, m is the number of points in each state of the phase space, N is the total number of states, which also controls the size of the recurrence plot. One critical parameter not existing in the formula is the embedding time delay τ. It is the interval between two adjacent points in a phase space state $\vec{x}(i)$. Besides, the threshold step in Eq. (1) is usually omitted in TSC to avoid the loss of texture information. Consequently, Eq. (1) can be simplified into Eq. (2):

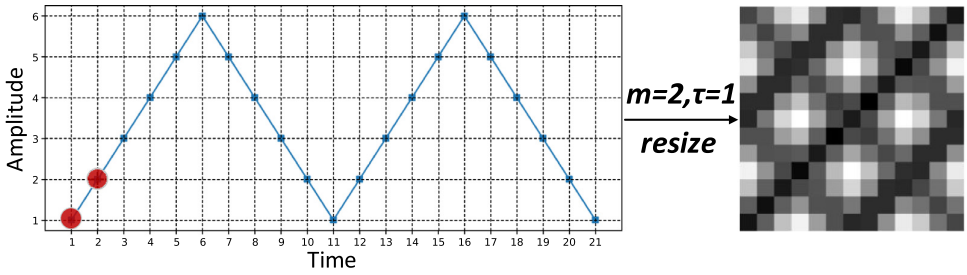
$$RP_{i,j}(\epsilon) = \|\vec{x}(i) - \vec{x}(j)\|, \vec{x}(\cdot) \in \mathfrak{R}^m, i, j = 1, \ldots, N. \tag{2}$$

Several benefits can be obtained through the combination of RP and CNNs. Due to the graphical nature of exposing recurrent patterns and magnifying the discriminative regions of sequences, RP expose features not evident in the time domain [20,21]. Moreover, RP construct long-term correlations for time series. Therefore, they alleviate the defects of CNN-based TSC networks in this respect. Though RP have many advantages and are widely applied in TSC, there still remains several problems to be tackled. First, the scales of discriminative regions of time series range widely. However, RP cannot adapt to this variability due to their limitation in image sizes. Second, it is difficult for RP to encode long sequences effectively. Finally, the important tendency transitions of time series are easily confused by RP. In the following sections, we will handle these challenges separately.

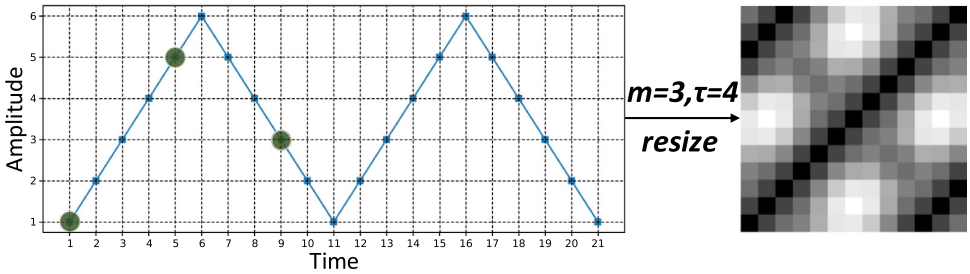
3.2.2. Multi-scale RP for handling scale variability of sequences

As indicated in Section 1, the lengths of sequences vary in a wide range, and the discriminative regions of sequences appear in various scales. Existing methods tackle such variability by adjusting the sizes of images [15,16,20,21,44]. However, the image sizes cannot be too large for the sake of the computation cost. Therefore, a conflict is created between the stronger representation abilities and the limited image sizes.

To tackle this problem, the phase space dimension m and the embedding time delay τ of RP are introduced. The phase space dimension m controls the number of points in each state of the



(a) A triangular periodic sequence and its recurrence plot with $m = 2$, $\tau = 1$



(b) A triangular periodic sequence and its recurrence plot with $m = 3$, $\tau = 4$

Fig. 4. The recurrence plots of a triangular periodic sequence. These images are generated with different m and τ. The red and green dots in (a) and (b) correspond to the first subsequences of the sequence when (m, τ) are (2,1) and (3,4), respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

phase space. Moreover, the embedding time delay τ is the interval between two adjacent points in a phase space state. These two parameters are always ignored and kept fixed in other articles [15,16,20,21,44]. However, in this paper, they are adjusted to enrich the scales of images. In this way, the representation abilities of RP can be enhanced even if the image sizes are restricted within a small range. A simple example illustrates this issue. Fig. 4 shows a triangular periodic sequence and its recurrence plots generated with two different groups of (m, τ), whose values are (2,1) and (3,4), respectively. With the same size, the images of different groups of (m, τ) have an obvious difference in scale.

Actually, the process of RP transformation is similar to the dilated convolution process [45]. Specifically, the sliding subsequences (the states in the phase space) of RP correspond to the dilated convolution kernels, except that the inner product is replaced by the norm calculation. Moreover, the parameters m and τ of a subsequence correspond to the size and the dilatation rate of a dilated convolution kernel, respectively. Different configurations of the dilated convolution kernels vary the scales of feature maps. Similarly, the values of m and τ determine the scales of images.

In summary, the values of m and τ are adjusted together with the image sizes to represent a sequence as multiple scales of images. Then, the image scale and size with better performance are selected.

3.2.3. Asymmetric RP for encoding long time series

Encoding the long sequences is particularly difficult. If the large-size RP images of long sequences are directly reduced to small sizes, it will lead to serious information loss. Even with m and τ being adjusted, RP still cannot represent the long sequences effectively.

To address this problem, we construct asymmetric RP. The original RP are symmetric along the main diagonal, leading to information redundancy. Therefore, we halve a long sequence into two pieces and encode each piece as a recurrence plot. Then, the oblique triangle matrix of each image is extracted. Finally, utilizing the symmetric structure of RP, the two extracted matrices are reassembled into an asymmetric image (see Figs. 3 and 5).

Asymmetric RP mitigate the information loss of the resizing process. In this paper, it is only used for encoding the long sequences.

3.2.4. Signed RP for tackling tendency confusion

The tendency information of certain sequences is critical for their classification. However, RP fail to describe the tendency transitions, weakening its representation ability severely. We use a simple example to illustrate this problem. s_1 and s_2 are two short sequences with opposite tendencies. Their values are [1,2,3] and [3,2,1], respectively. The RP matrices of these two sequences, RP_{s_1} and RP_{s_2}, are calculated by Eq. (2), where $\|\cdot\|$ is the $L_2 - norm$ and (m, τ) is (2,1). Eqs. (3) and (4) provide the calculation results of RP_{s_1} and RP_{s_2}, respectively. It can be observed that, the RP matrices of s_1 and s_2 are identical. Actually, this confusion problem is caused by the norm operation. Though the norm operation measures the distance between states in phase space, it cannot reflect the relative positions of these states.

$$RP_{s_1} = \begin{pmatrix} 0 & \sqrt{2} \\ \sqrt{2} & 0 \end{pmatrix}. \tag{3}$$

$$RP_{s_2} = \begin{pmatrix} 0 & \sqrt{2} \\ \sqrt{2} & 0 \end{pmatrix}. \tag{4}$$

To address this problem, the rule of signs is introduced to describe the tendency transitions. First, after a sequence is mapped to the phase space, the subtraction and L2-norm operations between different phase space states are performed. These two operations are used to calculate the state difference vectors and the recurrence plot, respectively. Second, the difference vector of each state is summed up separately, and a sign mask is extracted from these summed values. Note that the sign mask has a same size with the recurrence plot. Finally, the extracted sign mask is multiplied to the recurrence plot, and the signed recurrence plot is obtained. The

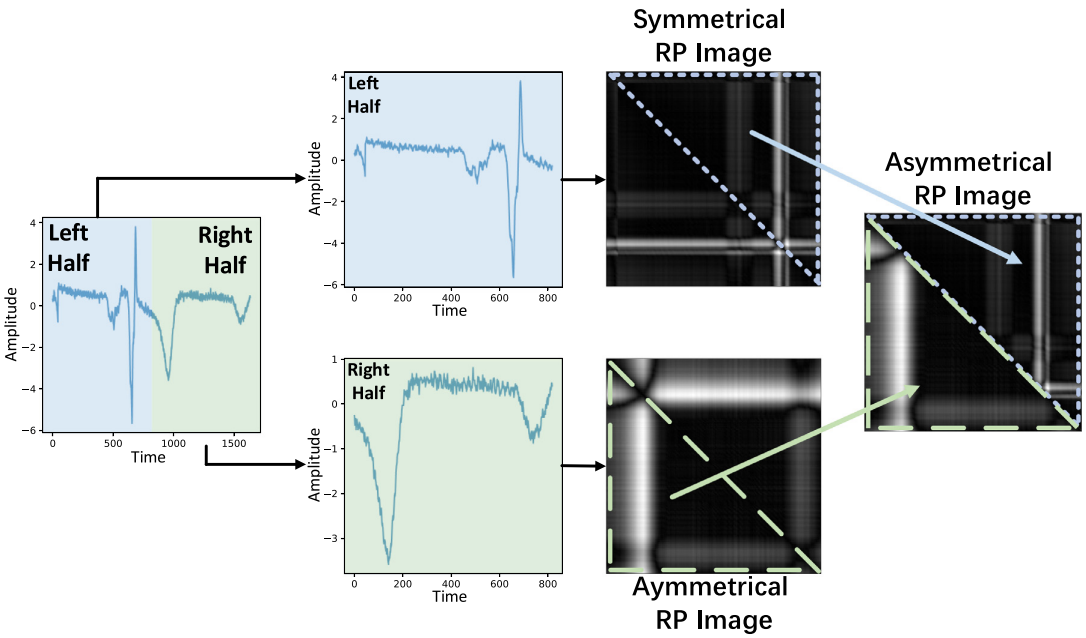


Fig. 5. The process of representing a long sequence (from 'CinCECGTorso' dataset) as an asymmetric recurrence plot.

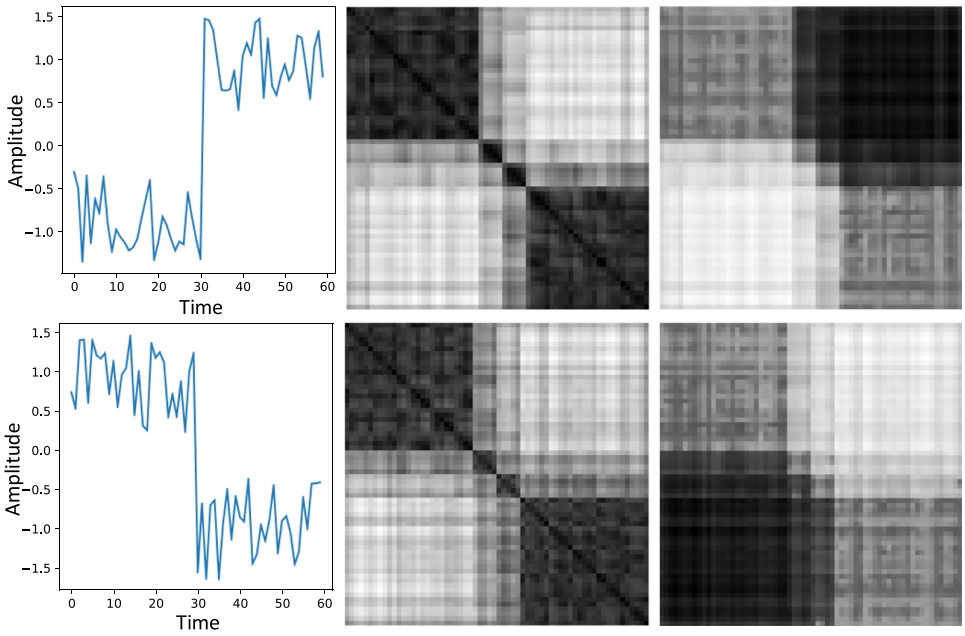


Fig. 6. The RP images of two sequences with opposite tendencies. (left column: two sequences of the 'SyntheticControl' dataset, middle column: the original RP images of these sequences, right column: the signed RP images of these sequences).

whole calculation process can be defined in Eq. (5).

$$RP_{i,j}(\epsilon) = \frac{sum(\vec{x}(i)-\vec{x}(j))\cdot\|\vec{x}(i)-\vec{x}(j)\|}{|sum(\vec{x}(i)-\vec{x}(j))|}, \quad \vec{x}(\cdot) \in \Re^m, i, j = 1, \ldots, N, \tag{5}$$

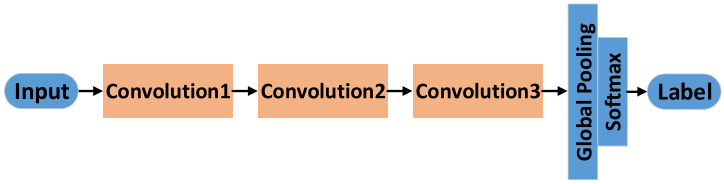
where *sum* is the vector summation function, $\|\cdot\|$ is the $L_2 - norm$ operation, $|\cdot|$ is the absolute value function. A visual comparison between the original RP and the signed RP is shown in Fig. 6. As shown, it is difficult to distinguish the original RP images. However, the signed RP images are significantly different.

3.3. Proposed IFCN for classifying MSRP images

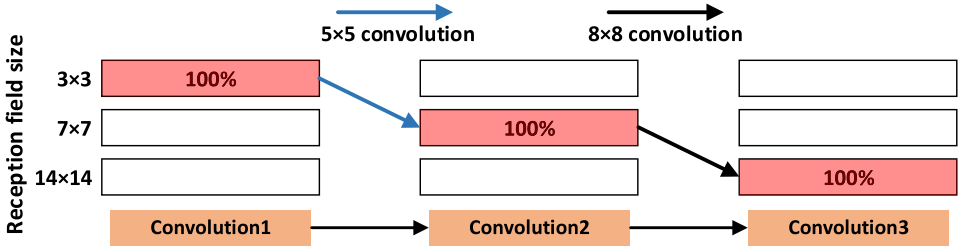
This section introduces our proposed network in detail. In Section 3.3.1, we firstly illustrate the modified Inception module, a network building block used for extracting multi-scale features. Then, the architectures of our proposed network, IFCN, is described in Section 3.3.2.

3.3.1. Modified inception module

As indicated in Section 1, existing TSC networks are not good at handling the multi-scale challenge of time series and their RP im-

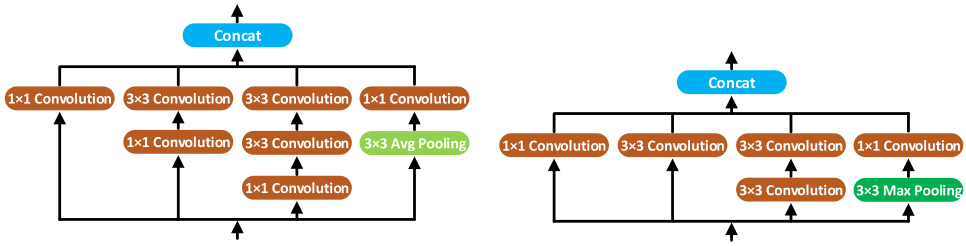


(a) The architecture of FCN



(b) The receptive field sizes of different layers in FCN

Fig. 7. The architecture and receptive field distribution of FCN. The percentage values in the boxes of (b) indicate the channel number proportions of the corresponding receptive fields.



(a) Inception-A module of Inception-v4 network (b) Modified Inception Module of this paper

Fig. 8. The architectures of Inception modules.

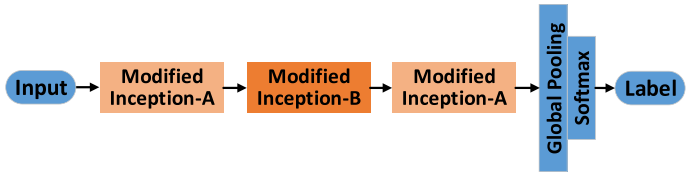
ages. Without loss of generality, we use one of the baseline TSC networks, namely FCN [22], to illustrate this problem. FCN is a three-layer fully convolutional network without the pooling and fully connected layers. Its architecture is shown in Fig. 7(a). The filter sizes of each layer are 3×3, 5×5 and 8×8, respectively. Moreover, the receptive field sizes of different layers in FCN are presented in Fig. 7(b), which are 3×3, 7×7 and 14×14, respectively. Since the features of the last layer are used for classification, the receptive field of FCN is consistent with that of the last layer (14×14). The receptive field can be considered as the view of a network. A network with a lager receptive field can capture longer patterns, while a network with a smaller receptive field tends to focus on extracting fine-grained features. Unfortunately, the discriminative regions of RP images have various scales, which accordingly requires the networks to have plentiful receptive fields. Such a single receptive field of FCN cannot cover this scale variability effectively.

To address this problem, the Inception-A module of Inception-v4 [12] is introduced. The Inception module is a network component clustering multiple convolution layers with different sizes of filters, its architecture is shown in Fig. 8(a). As shown, the introduced module is composed of four branches with different receptive fields, these branches contain layers with multiple sizes of filters (i.e. 1×1 layer, 3×3 layer, average pooling layer). This multi-branch architecture can enrich the receptive fields of a network without increasing too much computational cost [12].

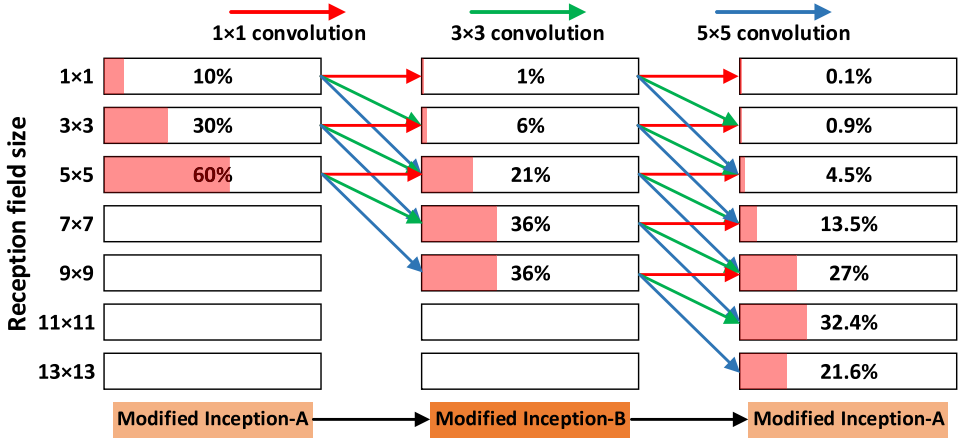
Though the introduced Inception module enriches the receptive fields of FCN, it is a bit redundant for TSC. The 1×1 layers of the original Inception module are used to compress the feature channels for computation cost reduction. However, since the feature channel numbers of TSC networks are relatively small and these networks are quite shallow, limited benefits can be obtained from the 1×1 layers. Moreover, these layers do not contribute to enriching the receptive fields while deepening the network, which may lead to the overfitting risk. Consequently, we remove the 1×1 layers in the second and third branch of the Inception module. Besides, the average pooling layer of the last branch is replaced with max pooling layer for better detecting local shapes. The architecture of the modified Inception module is shown in Fig. 8(b). In this module, the basic convolution blocks comprises a convolutional layer, a followed batch normalization layer [46] and a ReLU activation layer [47]. The modifications to the original Inception module will be demonstrated through the ablation experiments in Section 4.3.2.

3.3.2. Proposed IFCN

Similar to the architecture of FCN, IFCN is also composed of three basic components (see Fig. 9(a)), namely the modified Inception modules (see Fig. 8(b)). According to the difference in the amount of convolutional kernel channels, the modified Inception modules are further divided into Inception-A and Inception-B. Concretely, the kernel channels of the Inception-B module are doubled to alleviate the representational bottleneck of global average pooling. More details can be found in Table 1. Following by the modified Inception modules, a global average pooling layer and a softmax classifier are applied for classification.



(a) The architecture of IFCN



(b) The distribution of receptive field sizes in IFCN

Fig. 9. The architecture and receptive field distribution of proposed IFCN.

Table 1
Parameter configurations of the modified Inception modules.

Inception module	Convolution kernel channel number in different branches			
	1 × 1(left)	3 × 3(mid-left)	3 × 3,3 × 3(mid-right)	1 × 1(right)
Inception-A	16	32	9696	16
Inception-B	32	64	192,192	32

In order to more intuitively indicate the promotion brought by the introduced module, Fig. 9(b) shows the distribution of receptive field sizes of IFCN. As shown, IFCN obtains more diverse receptive field sizes compared with FCN (see Fig. 7(b)). Note that, for facilitating calculation purpose, two 3 × 3 convolution layers are equivalent to a 5 × 5 convolution layer, and a 3 × 3 pooling layer is equivalent to a 3 × 3 convolution layer.

4. Experiments and discussion

4.1. Experimental setup

The UCR archive [4] is a univariate time series repository, containing a large number of datasets collected from various real-world domains. In order to evaluate the performance of different methods, 85 datasets of the UCR time series archive are selected. Besides, we compare MSRP-IFCN with six benchmark methods, which are listed as follows:

- RP-CNN: RP-CNN [15] is the pioneering work to incorporate RP and a CNN in one framework. It is regarded as one of the baseline methods in this paper. The introduced CNN is composed of two convolutional layers, two max-pooling layers and two fully-connected layers.
- RP-AlexNet: In [42], the author uses the multi-channel signals collected from the circle drawing process of volunteers for Parkinsons disease identification. The collected signals are encoded as multi-channel images. These images are then input to the AlexNet for classification. Since the time series data used in this paper is univariate, the RP images input to the AlexNet are the single-channel images.
- FCN and ResNet: These two classifiers are proposed by Wang et al. [22]. They are widely regarded as the strong baselines for the DNN-based TSC methods [3].
- HIVE-COTE: HIVE-COTE is an ensemble-based method composed of various classifiers and multiple time series transformation representations [36]. It is considered as the state-of-the-art method on the UCR archive [3].
- MSRP Inception ResNet (MSRP-IRN): MSRP-IRN is a combination of MSRP and the Inception ResNet (IRN). IRN reuses IFCN by three times through the residual connections [10], and extend the network to a deeper architecture. By constructing this model, we try to explore the balance between powerful feature extraction and harmful overfitting.

The datasets used in this paper have already been z-normalized and split into the training and testing parts. The hyper-parameters of MSRP are phase space dimension m, embedding time delay τ and the image sizes. The values of (m, τ) are selected between (2,1) and (3,4), and the sizes of MSRP images are chosen from (16,48,64,72,80,88,96,112,128), according to different datasets. Suitable MSRP parameters can be obtained according to their performance on the validation sets. These parameters are provided in Tables 2 and 3. The three values in the parentheses of the first column correspond to m, τ and image sizes, respectively. For the UCR datasets, the time series longer than 700 points are regarded

Table 2
Test results Part I: comparison in terms of classification error rates on 85 UCR datasets (Continued in Table 3).

Database	RP-CNN	RP-AlexNet	FCN	ResNet	HIVE-COTE	MSRP-IRN	MSRP-IFCN
Adiac(64,2,1)	0.2800	0.2020	0.1430	0.1740	0.1846	0.1407	**0.1151**
ArrowHead(64,2,1)	0.1429	0.1543	0.1200	0.1830	0.1123	0.1257	**0.0914**
Beef(64,2,1)	0.0800	0.1333	0.2500	0.2330	0.2773	**0.0667**	0.1000
BeetleFly(64,2,1)	0.1000	0.1000	0.0500	**0.0000**	0.0410	**0.0000**	**0.0000**
BirdChicken(128,2,1)	0.1500	0.1000	0.0500	0.1000	**0.0495**	0.1000	0.0500
Car(64,2,1)	0.0667	0.1000	0.0830	0.0670	0.0745	**0.0500**	**0.0500**
CBF(64,2,1)	0.0050	**0.0000**	**0.0000**	0.0060	0.0006	**0.0000**	**0.0000**
ChlorineCon(112,2,1)	**0.1049**	0.2078	0.1570	0.1720	0.2749	0.1729	0.1677
CinCECGTorso(128,3,4)	0.0087	**0.0080**	0.1870	0.2290	0.0120	0.0877	0.0754
Coffee(64,2,1)	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.0018	**0.0000**	**0.0000**
Computers(128,3,4)	0.3680	0.2840	**0.1520**	0.1760	0.1806	0.2440	0.2280
CricketX(64,3,4)	0.2718	0.2436	0.1850	0.1790	0.1696	**0.1564**	0.1718
CricketY(64,3,4)	0.2462	0.2564	0.2080	0.1950	0.1630	0.1513	**0.1487**
CricketZ(64,3,4)	0.2667	0.2282	0.1870	0.1870	**0.1523**	0.1564	0.1538
DiatomSizeR(64,2,1)	0.0098	0.0194	0.0700	0.0690	0.0581	**0.0065**	0.0163
D.PhalanxAgeGroup(72,2,1)	0.2014	0.2158	**0.1650**	0.2020	0.1737	0.2158	0.2086
D.PhalanxCorrect(64,2,1)	0.1957	0.1920	0.1880	0.1800	**0.1751**	0.1848	0.1920
D.PhalanxTW(64,2,1)	0.2878	0.2734	**0.2100**	0.2600	0.3018	0.2590	0.2590
Earthquakes(64,2,1)	0.2014	0.2446	0.1990	0.2140	0.2530	0.2086	**0.1942**
ECG200(64,2,1)	**0.0000**	0.0700	0.1000	0.1300	0.1181	0.0900	0.0800
ECG5000(64,2,1)	0.0562	0.0582	0.0590	0.0690	**0.0527**	0.0551	0.0549
ECGFiveDays(64,3,4)	0.0023	0.0232	0.0150	0.0450	0.0105	**0.0000**	**0.0000**
ElectricDevices(88,2,1)	0.2843	0.2874	0.2770	0.2720	**0.1103**	0.2288	0.2360
FaceAll(96,2,1)	0.1900	0.2107	0.0710	0.1189	**0.0037**	0.0497	0.0373
FaceFour(96,2,1)	**0.0000**	0.0568	0.0680	0.0680	0.0505	0.0455	0.0455
FacesUCR(96,2,1)	0.0483	0.0907	0.0520	0.0420	**0.0164**	0.0541	0.0585
FiftyWords(48,3,4)	0.2600	0.2132	0.3210	0.2730	0.1932	**0.1582**	0.1824
Fish(64,2,1)	0.0850	0.0571	0.0290	0.0110	0.0238	**0.0057**	**0.0057**
FordA(64,2,1)	0.0955	0.0697	0.0940	0.0720	**0.0403**	0.0500	0.0553
FordB(128,3,4)	0.2198	0.1568	0.1170	0.1000	**0.0728**	0.1407	0.1642
GunPoint(64,2,1)	**0.0000**	**0.0000**	**0.0000**	0.0070	0.0033	**0.0000**	**0.0000**
Ham(64,3,4)	0.2000	0.2286	0.2380	0.2190	0.1592	0.1810	**0.1524**
HandOutlines(128,3,4)	0.0622	0.0541	0.2240	0.1390	0.0880	**0.0351**	0.0405
Haptics(64,2,1)	0.5390	0.5260	0.4490	0.4940	0.4697	0.4610	**0.4318**
Herring(64,3,4)	0.2969	0.3281	0.2970	0.4060	0.3661	**0.2656**	**0.2656**
InlineSkate(128,2,1)	0.6436	0.5782	0.5890	0.6350	**0.4741**	0.5418	0.5164
InsectWinSound(48,3,4)	0.3566	0.3586	0.5980	0.4690	0.3646	**0.3561**	0.3626
ItalyPowerDemand(16,2,1)	0.0330	0.0350	0.0300	0.0400	0.0322	**0.0262**	0.0272
LargeKitchenApps(128,2,1)	0.3547	0.1467	0.1040	0.1070	0.0773	**0.0747**	0.0853
Lightning2(64,3,4)	0.1639	0.2131	0.1970	0.2460	0.2030	0.1148	**0.0820**
Lightning7(64,3,4)	0.2600	0.1507	0.1370	0.1640	0.1889	0.1233	**0.1096**
Mallat(128,3,4)	0.0512	0.0337	**0.0200**	0.0210	0.0245	0.0337	0.0247
Meat(64,2,1)	**0.0000**	**0.0000**	0.0330	**0.0000**	0.0128	**0.0000**	**0.0000**
MedicalImages(96,2,1)	0.2658	0.2303	0.2080	0.2280	**0.1846**	0.1947	0.1868
M.PhaAgeGroup(64,3,4)	0.3571	0.3377	**0.2320**	0.2400	0.2951	0.3506	0.3312
Win num	10	10	11	9	19	34	**39**
AVG Arithmetic ranking	4.8118	4.7294	4.2941	4.6941	3.7177	2.2941	**2.1177**
AVG Geometric ranking	4.1346	4.1540	3.6967	4.0921	3.0368	1.9060	**1.7794**
MPCE	0.0450	0.0419	0.0392	0.0404	0.0356	0.0306	**0.0294**

as long sequences. This is an empirical value. It may lead to serious information loss when the RP images larger than 700 × 700 are directly resized to the maximum acceptable size, which is 128 × 128 as presented in Tables 2 and 3.

The network structure of IFCN has been described in Section 3.3. This network is initialized using the Xavier method [48], and optimized using categorical cross entropy and the 'Adam' optimizer [49]. The batch size is set to 16, the learning rate is initially set to 5×10^{-5} and decreased according to the training loss. Training process is stopped after 1500 epoches for each dataset.

The results of compared methods are obtained from their corresponding papers. The missing parts are supplemented using codes released by the authors or reproduced by ourselves. For fair comparison, the sizes of RP images in other articles has been changed to be consistent with this paper. All the methods are implemented in Pytorch on a PC with two Nvidia RTX 2080Ti GPUs.

To evaluate the performance of different methods, the evaluation metrics including Number of Wins, Average Arithmetic Rank, Average Geometric Rank and Mean Per-Class Error (MPCE) are introduced, which are consistent with those metrics in Wang et al. [22]. Among them, MPCE calculates the mean error rates by taking the factor of category amount into account. Then, we follow the recommendations of [50] to adopt the Friedman test for rejecting the null hypothesis [51]. Finally, utilizing a Wilcoxon signed-rank test with Holm correction ($\alpha = 0.05$) [52,53], the significance of differences between compared classifiers is measured. A critical difference (CD) diagram [50] is performed to intuitively visualize the performance of these classifiers.

4.2. Comparisons of different methods

Tables 2 and 3 provide the classification performance of 7 different classifiers on 85 UCR datasets. The best performance of each dataset is highlighted in bold. Besides, the CD diagram and some important pairwise comparisons are shown in Figs. 10 and 11, respectively.

Some observations can be made from the classification results. Firstly, compared with its competitors, MSRP-IFCN achieves the best classification performance over all of the four metrics (see Tables 2 and 3). In the CD diagram of Fig. 10, we also find that

Table 3
Test eesults Part II (Continued from Table 2).

Database	RP-CNN	RP-AlexNet	FCN	ResNet	HIVE-COTE	MSRP-IRN	MSRP-IFCN
M.PhalanxCorrect(64,3,4)	0.1375	**0.1340**	0.2050	0.2070	0.1911	0.1546	0.1409
M.PhalanxTW(64,3,4)	0.4091	0.3896	0.3880	0.3930	0.4288	0.3766	**0.3636**
MoteStrain(80,2,1)	0.1182	0.1398	0.0500	0.1050	0.0532	0.0767	**0.0487**
NonInvThorax1(128,3,4)	0.0580	0.0646	0.0390	0.0520	0.0683	0.0403	**0.0305**
NonInvThorax2(128,3,4)	0.0489	0.0514	0.0450	0.0490	0.0477	0.0427	**0.0336**
OliveOil(96,2,1)	0.1100	0.1333	0.1670	0.1330	0.1023	**0.0667**	**0.0667**
OSULeaf(96,2,1)	0.2900	0.2975	0.0120	0.0210	0.0295	**0.0083**	**0.0083**
PhalangesCorrect(64,2,1)	0.1585	0.1562	0.1740	0.1750	0.1788	**0.1469**	0.1562
Phoneme(128,3,4)	0.8191	0.7284	0.6550	0.6760	**0.6145**	0.7036	0.6698
Plane(64,3,4)	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.9900	**0.0000**	**0.0000**
P.PhalanxAgeGroup(64,2,1)	0.1268	0.1220	0.1510	0.1510	0.1515	**0.1122**	**0.1122**
P.PhalanxCorrect(64,2,1)	0.0756	0.0825	0.1000	0.0820	0.1237	0.0550	**0.0515**
ProPhalanxTW(64,3,4)	0.1854	**0.1707**	0.1900	0.1930	0.1884	0.1805	0.1707
RefriDevices(128,3,4)	0.4427	0.4107	0.4670	0.4720	**0.1990**	0.3947	0.3920
ScreenType(128,3,4)	0.6160	0.5413	0.3330	0.2930	**0.2890**	0.4133	0.3787
ShapeletSim(64,3,4)	0.1389	0.0500	0.1330	**0.0000**	0.0087	**0.0000**	**0.0000**
ShapesAll(64,2,1)	0.1550	0.1633	0.1020	0.0880	0.0741	**0.0550**	0.0633
SmallKitchenApps(128,3,4)	0.3093	0.1840	0.1970	0.2030	0.1629	0.1493	**0.1360**
SonyAIBORobot1(64,2,1)	0.0499	0.0582	0.0320	0.0150	0.1132	**0.0133**	0.0166
SonyAIBORobot2(64,2,1)	0.0923	0.1190	0.0380	0.0380	0.0546	0.0378	**0.0262**
StarLightCurves(128,3,4)	0.0234	0.0211	0.0330	0.0250	0.0185	0.0186	**0.0185**
Strawberry(64,2,1)	**0.0081**	0.0189	0.0310	0.0420	0.0302	0.0162	0.0108
SwedishLeaf(64,2,1)	0.0600	0.0512	0.0340	0.0420	0.0314	0.0288	**0.0192**
Symbols(64,3,4)	0.0824	0.0663	0.0380	0.1280	0.0342	**0.0090**	**0.0090**
SyntheticControl(64,2,1)	0.3433	0.2767	0.0100	**0.0000**	0.0004	**0.0000**	**0.0000**
ToeSegmentation1(64,3,4)	0.2061	0.1009	0.0310	0.0350	0.0450	0.0263	**0.0219**
ToeSegmentation2(64,3,4)	0.0923	0.0769	0.0850	0.1380	0.0336	**0.0154**	**0.0154**
Trace(64,2,1)	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
TwoLeadECG(64,2,1)	0.0026	0.0184	**0.0000**	**0.0000**	0.0065	**0.0000**	**0.0000**
TwoPatterns(64,2,1)	0.4935	0.4795	0.1030	**0.0000**	0.0001	**0.0000**	**0.0000**
UWaveAll(128,3,4)	0.0405	0.0416	0.1740	0.1320	0.0302	**0.0302**	0.0463
UWaveX(64,3,4)	0.3582	0.3322	0.2460	0.2130	**0.1616**	0.1686	0.1901
UWaveY(64,3,4)	0.3439	0.3007	0.2750	0.3320	**0.2245**	0.2362	0.2697
UWaveZ(64,2,1)	0.3317	0.3035	0.2710	0.2450	**0.2217**	0.2295	0.2376
Wafer(64,2,1)	**0.0000**	0.0023	0.0030	0.0030	0.0003	0.0005	0.0005
Wine(64,3,4)	0.0370	0.0185	0.1110	0.2040	0.0880	**0.0000**	0.0370
WordSynonyms(64,2,1)	0.3135	0.2837	0.4200	0.3680	**0.2520**	0.2539	0.2978
Worms(128,3,4)	0.3377	0.2468	0.3310	0.3810	0.2660	0.1818	**0.1558**
WormsTwoClass(128,3,4)	0.2078	**0.1558**	0.2710	0.2650	0.2161	0.1688	0.1688
Yoga(64,2,1)	0.1180	0.1180	0.1550	0.1420	0.0830	**0.0733**	0.0787
Win num	10	10	11	9	19	34	**39**
AVG Arithmetic ranking	4.8118	4.7294	4.2941	4.6941	3.7177	2.2941	**2.1177**
AVG Geometric ranking	4.1346	4.1540	3.6967	4.0921	3.0368	1.9060	**1.7794**
MPCE	0.0450	0.0419	0.0392	0.0404	0.0356	0.0306	**0.0294**

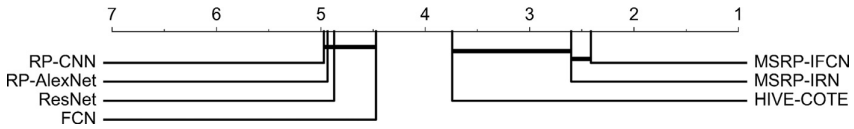


Fig. 10. The CD diagram of 6 compared classifiers and our proposed classifier over 85 UCR datasets, the thick horizontal lines in the diagram indicate a cluster of classifiers that are not significantly different in terms of classification performance.

MSRP-IFCN leads other classifiers, with MSRP-IRN achieving the second best performance. Moreover, pairwise comparisons with HIVE-COTE and RP-AlexNet (see top left and top middle of Fig. 11) indicate the obvious advantage of MSRP-IFCN.

Secondly, according to the comparison of MSRP-FCN and FCN in Fig. 11 (top right), the performance of MSRP-FCN is superior to FCN. This demonstrates that, as a transformation technique, MSRP are very effective for TSC. Some factors lead to the effectiveness. On the one hand, MSRP inherit the advantages of RP, e.g., exposing recurrent patterns and constructing long-term time dependencies. These advantages are highly complementary with the CNN. On the other hand, MSRP handle the weaknesses of RP, e.g., tendency confusion and encoding long sequences. Therefore, they provide better representations for the time series. In Section 4.3.1, MSRP are disassembled and further analysed through the ablation experiments.

Finally, it is found that MSRP-IFCN outperforms MSRP-IRN by a small margin. Pairwise comparison (see top left of Fig. 11) shows that MSRP-IRN achieves superiorities on quite a few datasets. However, it performs worse on more datasets. Consequently, it is indicated that deepening the structure of a TSC network may be helpful, but should be treated cautiously. The effectiveness of IFCN will be further discussed through the ablation experiments of Section 4.3.2 and visualization studies of Section 5.3.

4.3. Ablation studies

In this section, the ablation experiments of MSRP-IFCN are provided. These experiments are divided into two parts. First, compared with RP, the improvements of MSRP are analysed in Section 4.3.1, where each component of MSRP is validated. Second,

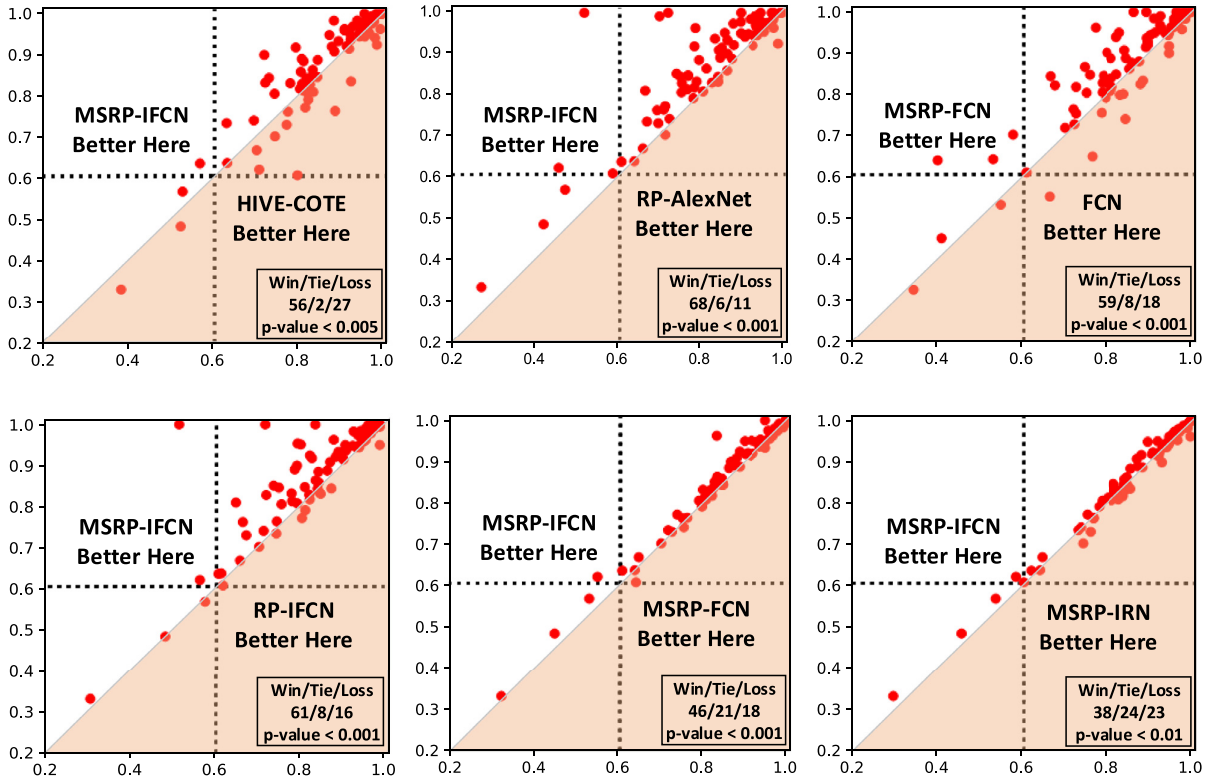


Fig. 11. The critical pairwise comparisons between our proposed classifier and the compared classifiers.

Table 4
Comparison in terms of error rates between different m and τ.

Dataset	Adiac	FaceAll	MoteStrain	OSULeaf	Wine	CricketY	Earthquakes	ElectricDev	FiftyWords	Lightning2
$m = 2, \tau = 1$	**0.1151**	**0.0373**	**0.0487**	**0.0083**	**0.0370**	0.1872	0.2374	0.2943	0.2396	0.1148
$m = 3, \tau = 4$	0.1432	0.1047	0.0887	0.0661	0.1296	**0.1487**	**0.1942**	**0.2360**	**0.1824**	**0.0820**

the effectiveness of IFCN is discussed in Section 4.3.2, where IFCN is compared with FCN and the network composed of three original Inception modules.

4.3.1. The ablation experiments of MSRP

To demonstrate the representation ability of MSRP, four groups of experiments are performed, with IFCN selected as the network classifier. First, the classification performance of MSRP and original RP are compared. Then, MSRP are decomposed into three components, namely the multi-scale RP, the asymmetric RP and the signed RP. Each component is validated separately, with other components being kept consistent.

MSRP vs. RP We firstly compare the classification performance of MSRP and original RP. An intuitional pairwise comparison is provided at the bottom left corner of Fig. 11. Thanks to the improvements of RP, MSRP perform significantly better than RP. Next, the effectiveness of each modification will be verified separately.

Multi-scale RP vs. single-scale RP The key idea of multi-scale RP is, sequences can be represented in multiple scales of images by adjusting the values of m and τ. Then, the most suitable scale will be selected according to the classification performance. In Table 4, the classification results of two different groups of m and τ on ten selected datasets are presented, with lower error rates being highlighted in bold. Distinct gaps between the bold and unbold values can be found in the table. Consequently, it is demonstrated that the adjustment of (m, τ) is essential, which enriches the scales of images, helps to better represent the time series, and improves the classification performance obviously.

Asymmetric RP vs. symmetric RP The asymmetric RP are constructed for long sequences to mitigate the information loss of the image resizing process. To compare the performance of the asymmetric RP and the original symmetric RP, ten UCR datasets with long sequential data are selected. Other configurations are kept consistent with those in Tables 2 and 3. The corresponding error rates are presented in Table 5. It can be observed that, the asymmetric RP obviously promote the performance on some datasets, e.g. 'CinCECGTorso', 'Phoneme', 'ScreenType' and 'UWaveGestureLibraryAll'. Besides, on other datasets, the asymmetric RP are also helpful.

Signed RP vs. unsigned RP The signed RP are constructed to handle the tendency confusion problem of RP. To demonstrate it, we compare the signed and unsigned versions of RP on ten selected datasets, with other configurations kept fixed. The corresponding error rates are presented in Table 6. As shown, the signed RP perform far better than the unsigned RP. Consequently, it is demonstrated that the designed sign masks are very effective. The designed sign masks will be further discussed through a visualization technique in Section 5.

4.3.2. The ablation experiments of IFCN

To demonstrate the effectiveness of IFCN, we compare it with FCN and the network (referred to as Inception) composed of three

Table 5
Comparison in terms of error rates between the symmetric and the asymmetric RP.

Dataset	Cin-wCin-Torso	Mallat	NonInv-Thorax1	NonInv-Thorax2	Phoneme	ScreenType	Sma-KitApp	Star-LiCurves	UWaveAll	Worms-TwoClass
symmetric RP	0.2855	0.0473	0.0478	0.0438	0.7326	0.464	0.1787	0.0189	0.0863	**0.1688**
asymmetric RP	**0.0754**	**0.0247**	**0.0305**	**0.0336**	**0.6698**	**0.3787**	**0.1360**	**0.0185**	**0.0463**	**0.1688**

Table 6
Comparison in terms of error rates between the signed RP and the unsigned RP.

Dataset	CricketX	CricketY	CricketZ	Lightning7	OSULeaf	SynControl	TwoPatterns	UWavX	UWaveY	UWaveZ
Signed RP	**0.1718**	**0.1487**	**0.1538**	**0.1096**	**0.0083**	**0**	**0**	**0.1901**	**0.2697**	**0.2376**
Unsigned RP	0.2128	0.2333	0.2026	0.1370	0.0744	0.3000	0.4858	0.3741	0.3222	0.3322

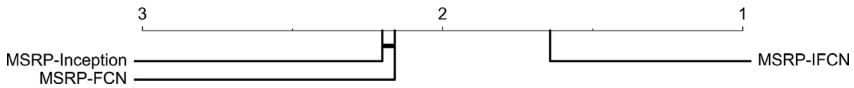


Fig. 12. The CD diagram of IFCN, FCN and Inception on classifying MSRP images.

original Inception modules, respectively. Note that, these three networks are adjusted to have comparable amount of parameters. The classification performance of these networks is presented in Fig. 12. As shown, on the one hand, IFCN performs better than FCN. Moreover, Fig. 11 (bottom middle) provides their pairwise comparison, where IFCN is superior to FCN in more than half of the selected datasets. These results indicate that stronger multi-scale feature extraction is helpful for TSC. On the other hand, IFCN is also obviously better than Inception. Note that, Inception is even slightly worse than FCN. These comparisons demonstrate the effectiveness of our modified Inception module.

5. Visualization

Time series are hard to understand intuitively. Therefore, visually interpreting the decisions made by a TSC network becomes particularly important. As far as we know, Class Activation Map (CAM) is the most commonly used visualization technique for TSC networks. It is originally proposed for image classification analysis [54], then introduced to TSC to highlight the contributing regions of sequences to a predicted category [22]. However, CAM cannot be applied to networks with fully connected layers [3]. Moreover, it is not good at lighting up the fine-grained contributing regions [55]. To handle these issues, this paper introduces a better visualization technique, namely Guided Grad-CAM [55]. This technique is simply modified to adapt to the characteristics of time series and MSRP images. It is briefly described in Section 5.1. Using the modified Guided Grad-CAM, the effect of the designed sign masks for RP is illustrated in Section 5.2. Moreover, the visual comparisons between different TSC networks are provided in Section 5.3.

5.1. Mask guided grad-CAM

Guided Grad-CAM has been proved to be an effective visualization tool for image classification analysis [55], which combines the high-resolution nature of Guided Backpropagation [56] and the class-discriminative nature of Grad-CAM [55]. Besides, it can be applied to arbitrary CNN-based networks without architectural changes.

However, two obvious differences between image and time series data hinder the direct application of Guided Grad-CAM. First, the pixels of time series and MSRP images have lots of important negative values, while all the pixel values of images are non-negative. Second, the discriminative regions of certain sequences can be very sparse, which may invalidate Grad-CAM. To handle these issues, the sign masks of time series and MSRP images are extracted and then multiplied to the gradient maps of Guided-Backpropagation. In this way, the significant negative gradients are preserved. Then, the global average pooling operation of Grad-CAM is omitted, and the gradient maps are directly multiplied to corresponding feature maps. Eqs. (6) and (7) illustrate these two modifications.

$$L^c_{Ma-Gu-Back} = ReLU\left(sign\_mask \cdot \frac{\partial y^c}{\partial I}\right), sign\_mask = sign(I), \quad (6)$$

$$L^c_{Gr-CAM} = ReLU\left(\sum_k \frac{\partial y^c}{\partial S^k} \cdot S^k\right), \quad (7)$$

where L^c and y^c represent the heat maps of category c and the predicted scores, respectively, I represent the input MSRP images or sequences, $sign$ represents an operation extracting the sign masks, S^k represents the kth feature map of the last convolutional layer.

Finally, the heat maps are calculated by multiplying $L^c_{Ma-Gu-Back}$ and L^c_{Gr-CAM}, as Eq. (8) defined. The framework of the modified Guided Grad-CAM, named Mask Guided Grad-CAM, is shown in Fig. 13.

$$L^c_{Ma-Gu-Gr-CAM} = L^c_{Ma-Gu-Back} \cdot L^c_{Gr-CAM}. \quad (8)$$

5.2. Analyzing the rule of signs visually

The ablation experimental results of Section 4.3 have demonstrated that the rule of signs significantly boosts the classification performance. Using Mask Guided Grad-CAM, we can illustrate why the rule of signs works.

As discussed in Section 3.2.3, the designed sign masks supplement the tendency transition information missing from RP. A similar work is presented in Wang and Oates [41], which utilizes Markov Transition Fields (MTF) to describe the tendency transitions of sequences. Its effectiveness has been demonstrated through extensive experiments [41]. An MTF image is constructed by the dynamic transition probabilities between different fragments of a sequence. In Fig. 14(a), some MTF images are presented as examples. If we input the sign masks of MSRP images to IFCN for classification, then highlight the contributing regions of

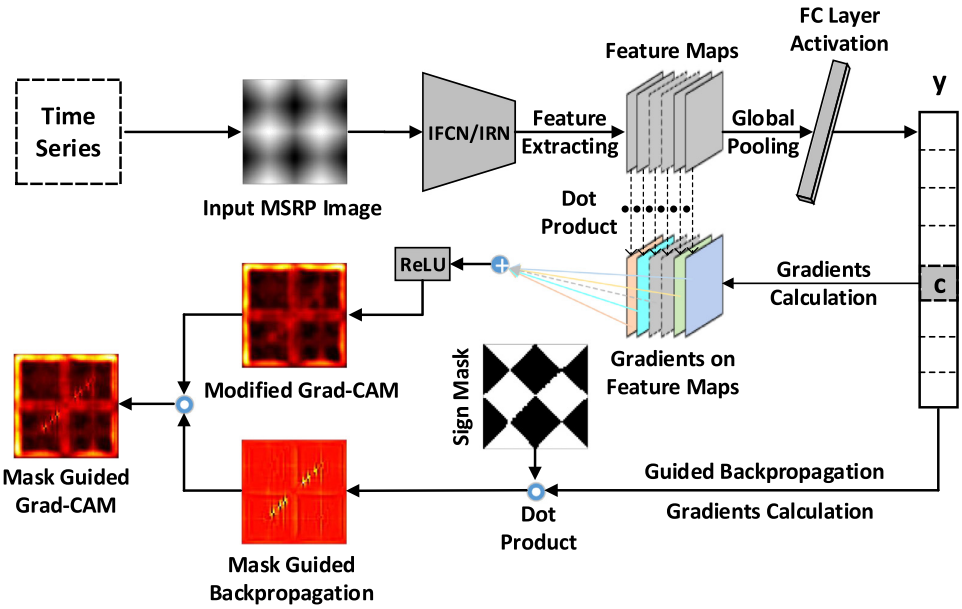
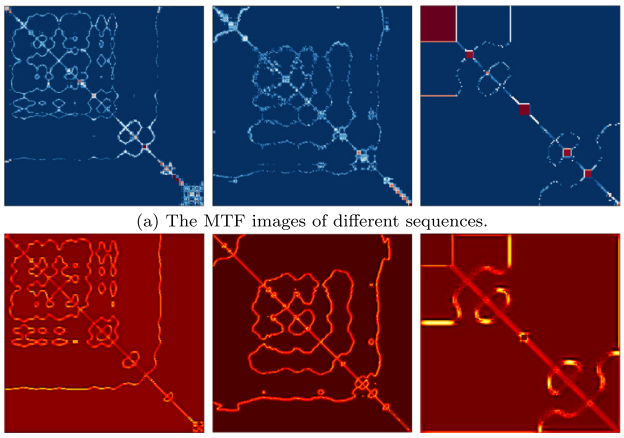


Fig. 13. The framework of mask guided grad-CAM.



(a) The MTF images of different sequences.

(b) The highlighted sign mask images of different sequences.

Fig. 14. The MTF images and highlighted sign mask images of sequences from the 'Beef' (left column), 'Coffee' (middle column) and 'UWaveGestureLibraryY' (right column) datasets.

these sign masks through Mask Guided Grad-CAM, it can be found that these highlighted images are very similar to MTF images (see Fig. 14). A reasonable interpretation to this phenomenon is, the designed sign masks reveal the dynamic tendency transitions of sequences similar with MTF. These interesting experimental results support the rule of signs as an effective supplement to RP.

5.3. Comparison between different DNN classifiers

In this section, the feature extraction performance of four representative CNN-based networks (traditional CNN [15], FCN [22], IFCN and IRN) are visually compared. Concretely, we select two sequences from Fig. 1, and use Mask Guided Grad-CAM to locate the discriminative regions of these sequences and their MSRP images, respectively. If a network has better feature extraction abilities, its heat maps more precisely locate the discriminative regions. The heat maps of different networks are shown in Fig. 15.

It is worth mentioning that all of the investigated classifiers classify the two selected sequences correctly. The discriminative regions of the sequence from 'Adiac' dataset are their corner areas, especially the tiny waves at the bottom of the curve. These waves correspond to the slender straight lines in corresponding MSRP image. The sequence of 'SyntheticControl' dataset can be distinguished by the sharp jitter at the top of the curve, which becomes a 'cross' in the corresponding MSRP image.

Some interesting observations can be made from Fig. 15. First, the heat maps of the traditional CNN are relatively messy, which cannot locate the discriminative regions precisely (see Fig. 15 left column). This network is composed by two convolutional layers, two max-pooling layers and two fully-connected layers [15]. Such unsatisfactory visualization performance may be due to the information loss brought by the pooling operations and the weak feature extraction ability of fully-connected layers. Actually, the traditional CNN is the worst classifier among all the compared networks.

Second, the heat maps of IFCN more precisely locate the discriminative regions of sequences than the heat maps of FCN (see Fig. 15 middle left and middle right columns). Thanks to the rich receptive fields of the modified Inception modules, IFCN is enhanced in extracting multi-scale features. Consequently, compared with FCN, IFCN better adapts to the scale variability of discriminative regions, leading to better visualization performance.

Finally, IRN creates an inconsistence between its heat maps. For the sequence from the 'Trace' dataset, IRN produces a very high-resolution heat map, and most accurately locate the discriminative regions (see Fig. 15 right column). However, for the sequence from the 'Adiac' dataset, the visualization performance of IRN is poor, with fuzzy heat maps and imprecise location (see Fig. 15). A reasonable interpretation is, though a deeper network architecture introduces stronger representation ability, it also gives rise to higher overfitting risk, leading to the instability of IRN.

In general, it can be observed from Fig. 15 that IFCN achieves the best performance among all of the competitors, followed by IRN and FCN. These visualization results are consistent with the experimental results of Section 4, and demonstrate the effectiveness of our proposed networks indirectly.

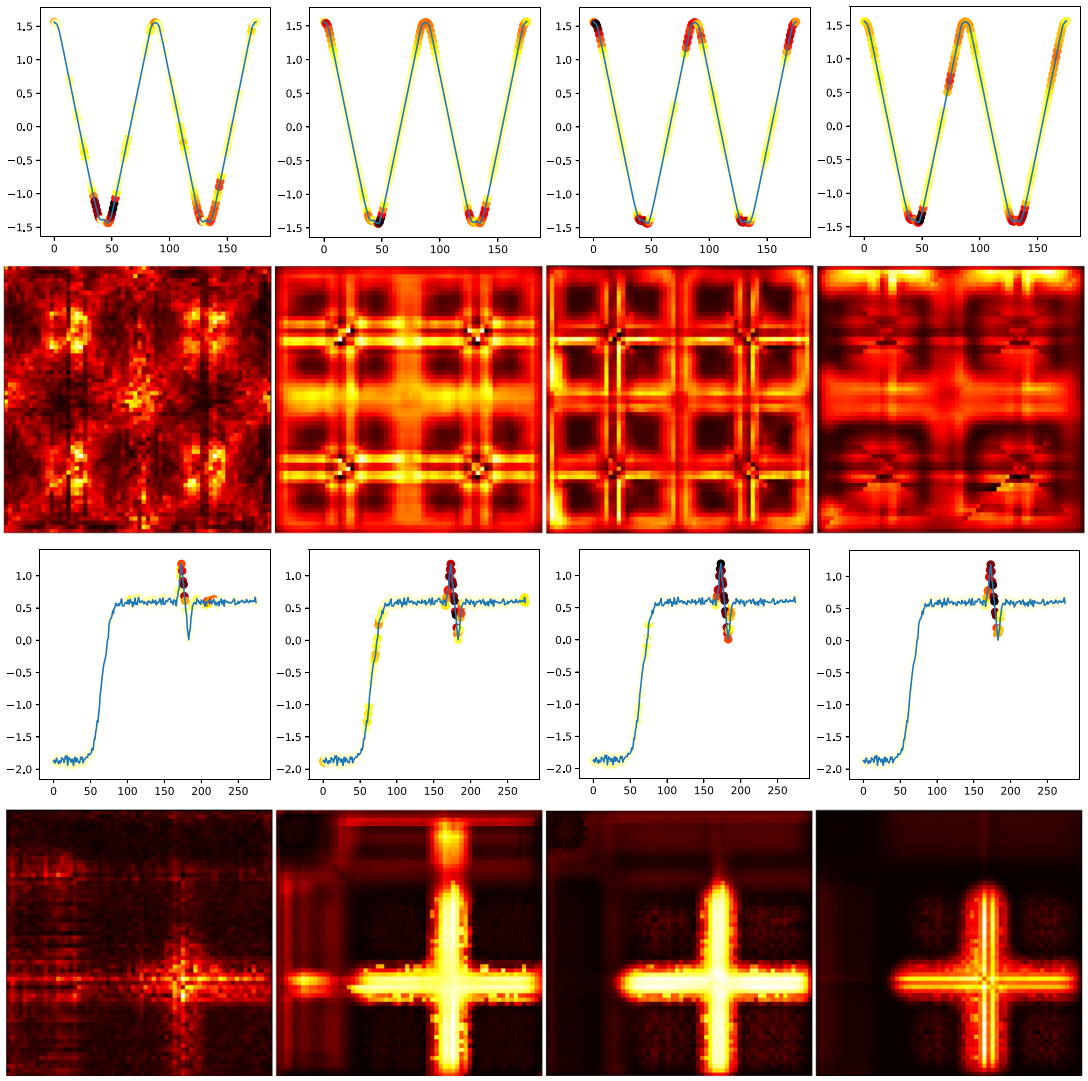


Fig. 15. The heat maps of four networks, CNN (left column), FCN (middle left column), IFCN (middle right column) and IRN (right column). The two sequences in the figure come from the 'Adiac' (first row) and 'Trace' (third row) datasets, respectively. Besides, the heat maps of their MSRP images are also provided (second and last row).

6. Conclusion

In this paper, the time series are encoded as MSRP images for classification. We first comprehensively improve RP to propose MSRP, which not only better adapt to the variability of discriminative region scales and lengths of sequences, but also tackle the tendency confusion problem of RP. Then, to handle the multi-scale problem of MSRP images, we introduce a modified Inception module and propose an Inception architectural network, named IFCN. This network is enhanced in terms of multi-scale feature extraction.

Experimental results on 85 UCR datasets demonstrate that our proposed method outperforms the state-of-the-arts. The effectiveness of each module of our classifier is validated through extensive ablation experiments. Moreover, the visualization results also demonstrate the effectiveness of our method indirectly.

As a future work, we would like to transform MSRP as a network building block for end-to-end classification. The introduction of the MSRP block will avoid complicated data transformations. Besides, similar with MSRP, this block will be used to construct long-term correlations between the extracted features, making up the defects of existing CNN-based TSC networks in this aspect. We believe this future work will be promising, bringing further promotion to TSC networks.

Declaration of Competing Interest

None.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant (No. 61903373, No. 62002372)

References

[1] T. Ying, Y. Shi, Data mining and big data, IEEE Trans. Knowl. Data Eng. 26 (1) (2016) 97–107.
[2] A. Bagnall, J. Lines, A. Bostrom, J. Large, E. Keogh, The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances, Data Min. Knowl. Discov. 31 (3) (2017) 606–660.

[3] H.I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, P.-A. Muller, Deep learning for time series classification: a review, Data Min. Knowl. Discov. 33 (4) (2019) 917–963.
[4] H.A. Dau, A. Bagnall, K. Kamgar, C.-C.M. Yeh, Y. Zhu, S. Gharghabi, C.A. Ratanamahatana, E. Keogh, The UCR time series archive, IEEE/CAA J. Autom. Sin. 6 (6) (2019) 1293–1305.
[5] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, Y. Liu, Deep learning for sensor-based human activity recognition: overview, challenges, and opportunities, ACM Comput. Surv. 54 (4) (2021) 1–40.
[6] M.-P. Hosseini, A. Hosseini, K. Ahi, A review on machine learning for EEG signal processing in bioengineering, IEEE Rev. Biomed. Eng. 14 (2020) 204–218.
[7] X. Liu, H. Wang, Z. Li, L. Qin, Deep learning in ecg diagnosis: a review, Knowl. Based Syst. 227 (2021) 107–187.
[8] S. Bhatt, A. Jain, A. Dev, Continuous speech recognition technologies a review, Recent Dev. Acoust. (2021) 85–94.
[9] A. Abid, M.T. Khan, J. Iqbal, A review on fault detection and diagnosis techniques: basics and beyond, Artif. Intell. Rev. 54 (5) (2021) 3639–3664.
[10] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2016, pp. 770–778.
[11] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2016, pp. 2818–2826.
[12] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Thirty-First AAAI Conference on Artificial Intelligence, AAAI, 2017, pp. 4278–4284.
[13] Y. Chen, S. Su, H. Yang, Convolutional neural network analysis of recurrence plots for anomaly detection, Int. J. Bifurc. Chaos 30 (01) (2020) 201–213.
[14] İ. Cantürk, Fuzzy recurrence plot-based analysis of dynamic and static spiral tests of parkinsons disease patients, Neural Comput. Appl. 33 (2021) 349–360.
[15] N. Hatami, Y. Gavet, J. Debayle, Classification of time-series images using deep convolutional neural networks, in: Tenth International Conference on Machine Vision, SPIE, 2018, p. 106960Y.
[16] E. Garcia-Ceja, M.Z. Uddin, J. Torresen, Classification of recurrence plots distance matrices with a convolutional neural network for activity recognition, Procedia Comput. Sci. 130 (2018) 157–163.
[17] J.P. Eckmann, S.O. Kamphorst, D. Ruelle, Recurrence plots of dynamical systems, Europhys. Lett. 4 (9) (1987) 973–977.
[18] N. Marwan, M.C. Romano, M. Thiel, J. Kurths, Recurrence plots for the analysis of complex systems, Phys. Rep. 438 (5–6) (2007) 237–329.
[19] R. Rajabi, A. Estebsari, Deep learning based forecasting of individual residential loads using recurrence plots, in: IEEE Milan PowerTech, IEEE, 2019, pp. 1–5.
[20] D.F. Silva, V.M. De Souza, G.E. Batista, Time series classification using compression distance of recurrence plots, in: IEEE 13th International Conference on Data Mining, IEEE, 2013, pp. 687–696.
[21] V.M. Souza, D.F. Silva, G.E. Batista, Extracting texture features for time series classification, in: IEEE 22nd International Conference on Pattern Recognition, IEEE, 2014, pp. 1425–1430.
[22] Z. Wang, W. Yan, T. Oates, Time series classification from scratch with deep neural networks: a strong baseline, in: IEEE International Joint Conference on Neural Networks, IEEE, 2017, pp. 1578–1585.
[23] W. Luo, Y. Li, R. Urtasun, R. Zemel, Understanding the effective receptive field in deep convolutional neural networks, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, MIT Press, 2016, pp. 4905–4913.
[24] E. Keogh, C.A. Ratanamahatana, Exact indexing of dynamic time warping, Knowl. Inf. Syst. 7 (3) (2005) 358–386.
[25] Y.-S. Jeong, M.K. Jeong, O.A. Omitaomu, Weighted dynamic time warping for time series classification, Pattern Recognit. 44 (9) (2011) 2231–2240.
[26] J. Zhao, L. Itti, Shapedtw: shape dynamic time warping, Pattern Recognit. 74 (2018) 171–184.
[27] B.J. Jain, D. Schultz, Asymmetric learning vector quantization for efficient nearest neighbor classification in dynamic time warping spaces, Pattern Recognit. 76 (2018) 349–366.
[28] J. Hills, J. Lines, E. Baranauskas, J. Mapp, A. Bagnall, Classification of time series by shapelet transformation, Data Min. Knowl. Discov. 28 (4) (2014) 851–881.
[29] J. Grabocka, M. Wistuba, L. Schmidt-Thieme, Fast classification of univariate and multivariate time series through shapelet discovery, Knowl. Inf. Syst. 49 (2) (2016) 429–454.
[30] H. Wang, Q. Zhang, J. Wu, S. Pan, Y. Chen, Time series feature learning with labeled and unlabeled data, Pattern Recognit. 89 (2019) 55–66.
[31] M.G. Baydogan, G. Runger, E. Tuv, A bag-of-features framework to classify time series, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2796–2802.
[32] P. Schäfer, The boss is concerned with time series classification in the presence of noise, Data Min. Knowl. Discov. 29 (6) (2015) 1505–1530.
[33] P. Schäfer, U. Leser, Fast and accurate time series classification with weasel, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 637–646.
[34] J. Lines, A. Bagnall, Time series classification with ensembles of elastic distance measures, Data Min. Knowl. Discov. 29 (3) (2015) 565–592.
[35] A. Bagnall, J. Lines, J. Hills, A. Bostrom, Time-series classification with cote: the collective of transformation-based ensembles, IEEE Trans. Knowl. Data Eng. 27 (9) (2015) 2522–2535.
[36] J. Lines, S. Taylor, A. Bagnall, Time series classification with hive-cote: the hierarchical vote collective of transformation-based ensembles, ACM Trans. Knowl. Discov. Data 12 (5) (2018) 52.
[37] Z. Cui, W. Chen, Y. Chen, Multi-scale convolutional neural networks for time series classification, arXiv preprint arXiv:1603.06995 (2016).
[38] A. Le Guennec, S. Malinowski, R. Tavenard, Data augmentation for time series classification using convolutional neural networks, in: Proceedings of the ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data, Springer, 2016.
[39] J. Wang, Z. Wang, J. Li, J. Wu, Multilevel wavelet decomposition network for interpretable time series analysis, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 2437–2446.
[40] Z. Wang, T. Oates, Encoding time series as images for visual inspection and classification using tiled convolutional neural networks, in: Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI, 2015, pp. 40–46.
[41] Z. Wang, T. Oates, Imaging time-series to improve classification and imputation, in: Twenty-Fourth International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 2015, pp. 3939–3945.
[42] L.C. Afonso, G.H. Rosa, C.R. Pereira, S.A. Weber, C. Hook, V.H.C. Albuquerque, J.P. Papa, A recurrence plot-based approach for parkinsons disease identification, Future Gener. Comput. Syst. 94 (2019) 282–292.
[43] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, MIT Press, 2012, pp. 1097–1105.
[44] N. Hatami, Y. Gavet, J. Debayle, Bag of recurrence patterns representation for time-series classification, Pattern Anal. Appl. 22 (3) (2019) 877–887.
[45] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: International Conference on Learning Representations, 2016.
[46] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167 (2015).
[47] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR, 2011, pp. 315–323.
[48] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR, 2010, pp. 249–256.
[49] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
[50] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[51] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1) (1940) 86–92.
[52] S. Garcia, F. Herrera, An extension on"statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, J. Mach. Learn. Res. 9 (12) (2008) 2677–2694.
[53] A. Benavoli, G. Corani, F. Mangili, Should we really use post-hoc tests based on mean-ranks? J. Mach. Learn. Res. 17 (1) (2016) 152–161.
[54] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2921–2929.
[55] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad–CAM: visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.
[56] J.T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: the all convolutional net, arXiv preprint arXiv:1412.6806 (2014).

Ye Zhang received the M.Sc. degree from Central South University, China. He is currently pursuing the Ph.D. degree with the College of Electronic Science and Technology, National University of Defense Technology, China. His main research interests include data mining, pattern recognition and deep learning.

Yi Hou received the B.Sc. degree from Wuhan University, China, and the M.Sc. as well as Ph.D. degree from the National University of Defense Technology, China. He held a visiting position with the Department of Computing Science, University of Alberta, Canada, from 2014 to 2016. His main research interests include robot visual SLAM, visual place recognition, time series classification, signal processing, computer vision, deep learning, pattern recognition, and image processing.

Kewei Ouyang received the B.Sc. degree from the Dalian University of Technology, China, and the M.Sc. degree from the National University of Defense Technology, China. He is currently pursuing the Ph.D. degree with the College of Electronic Science and Technology, National University of Defense Technology, China. His main research interests include time series classification and clustering.

Shilin Zhou is currently a Professor with the College of Electronic Science and Technology, National University of Defense Technology, China. His main research interests include pattern recognition, signal processing, computer vision, intelligent information processing, and remote sensing image processing.