




Time Series Classification Based on Image Transformation Using Feature Fusion Strategy

Wentao Jiang¹ · Dabin Zhang¹ · Liwen Ling¹  · Ruibin Lin¹

Accepted: 18 February 2022 / Published online: 15 March 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Time series classification is an important branch of data analysis. Scholars have proposed a large number of time series classification methods in recent years. However, time series classification remains a challenging problem due to feature selection in time series classification. In order to further simplify the feature selection procedure and improve time series classification accuracy, an automatic feature selection of a time series classification method based on an image feature fusion strategy and a deep learning algorithm is proposed. First, a time series is transformed into images using different types of image transformation methods, i.e. the recurrence plot, Gramian angle difference field, Gramian angle summation field and Markov transition field. Second, the above four images are encoded into a new image type, that is the combined image, by a feature fusion strategy. Finally, a convolutional neural network is used for combined image classification and forecasting model selection. Time series from the M1 and M3 competition datasets are used to verify the effectiveness of the proposed method. The experimental results show that the algorithm has a higher classification accuracy and smaller prediction error compared to the benchmark models. Moreover, the forecasting error MAPE of combined image method is reduced by 0.2020 and 1.7454 compared with the traditional image method and single forecasting method respectively.

Keywords Time-series classification · Time-series images · Combined image · Feature fusion · Deep learning

✉ Liwen Ling
linglw@scau.edu.cn

Wentao Jiang
Jiangwt2@163.com

Dabin Zhang
zdbff@aliyun.com

Ruibin Lin
450169049@qq.com

¹ College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China

1 Introduction

As a branch of time series analysis, time series classification has attracted extensive attention in the field of data mining. Time series classification widely exists in many fields in real life, such as electronic health records [1], human activity recognition [2], to acoustic scene classification [3], and cyber-security [4]. Owing to the strong volatility and uncertainty of real time series data, traditional methods such as expert experience consultation and multiple model perform poorly in classification. Therefore, the successful application of a distance based time series classification method [5] provides new ideas for the field of time series classification. One of the most popular and traditional TSC approaches is the use of a nearest neighbour (NN) classifier coupled with a distance function [6]. Later, a improved method based on the NN called the k-nearest neighbor (k-NN) is proposed [7]. Seto et al. indicate that dynamic time warping (DTW) [8] is the best distance-based measure to use with the k-NN, which can greatly improve the classification accuracy. Lines and Bagnall [9] compared several distance measures and showed that there is no single distance measure that significantly outperforms DTW. They also showed that ensembling individual NN classifiers (with different distance measures) outperforms all of an ensemble's individual components. In addition to distance-based time series classification, feature-based algorithms are also widely used. For example, Nanopoulos et al. use the mean, standard deviation, skewness and kurtosis of continuous increments to represent time series and classify them successfully [10]. Morchen et al. use features derived from wavelet and Fourier transforms of a range of time series data sets to classify time series [11]. Wang et al. introduce a more comprehensive feature combination method that contains thirteen features such as the trend, seasonality, periodicity, serial correlation, chaos, nonlinearity, and self-similarity to represent time series [12]. Lin et al. [13] propose bag of words (BOW) method, which quantifies the extracted feature BOW and inputs it into the classifier as a feature attribute. Because of the diversity of feature selection, Baydogan et al. [14] integrate feature engineering works and propose a time series bag of feature (TSBF) to extract multiple subsequences of random local information, and use these subsequences to predict the category of time series. Similar to the choice of representations and distance metrics for time series, features for time-series classification problems are usually selected manually by a researcher for a given dataset.

However, in the coming big data era, manual feature selection is too cumbersome and consumes considerable manpower and computing resources. Therefore, the automatic feature extraction and classification of time series under a deep learning framework has given scholars some inspiration.

The deep learning framework [15, 16] performs well in a variety of classification tasks [17–19]. The performance of CNNs in image recognition tasks reached the human level in 2015 [20]. Also deep CNNs made outstanding contributions to solving many aspects of machine translation [21], natural language processing (NLP) tasks [22], document classification [23], 2-D object recognition [24, 25], image retrieval [26] and underwater image enhancement [27]. Inspired by recent successes of deep learning in computer vision, the idea of transforming time series into images has received much attention in a time series analysis. The images of time series can be trained by CNNs, and the features can be extracted from images automatically [28]. Silva et al. [29] improve the recurrence plot (RP) algorithm by using the compression distance to expand the RP paradigm of time series classification. Campanharo et al. [30] propose a weighted adjacency matrix method based on a RP to extract the features of rotation motion from time series in the first-order Markov transition field (MTF). Wang et al. [31] propose encoding time series as two Gramian angle fields (GAFs) and using deep learning

model to classify. Li Xixi et al. [32] propose the idea of transforming time series into images by using a RP, used a deep learning framework to classify and predict time series, and proposed that a CNN could automatically extract the features in time series images.

In this paper, an improved time series image conversion method is proposed. The four time series imaging methods of the RP, GADF, GASF and MTF are combined into one image using the idea of feature fusion so as to reduce the edge feature loss. Later, three deep learning models, Resnet-18, VGG-11 and DenseNet-121, were used as classifiers. In this paper, the labels of supervised learning are given by six forecasting models, so the classification problem can also be regarded as the problem of forecasting model selection. Furthermore, the proposed combined image algorithm is verified with the M1 and M3 datasets. The experimental results demonstrate that compared with the traditional time series image algorithm and other classification algorithms (Sect. 4.4), the combined image algorithm has a higher classification rate. Also after forecasting selection, compared with traditional time series imaging algorithm and single forecasting models (such as AMRIMA), the proposed combined image algorithm has smaller forecasting error (MAPE). The combined image method based on deep learning significantly improves the time series classification rates and significantly reduces the forecasting error of selecting the optimal forecasting model, which verifies the effectiveness of the combined image method. Therefore, the main contributions of this paper are the following:

- An improved time series image conversion method based on feature fusion is proposed in this paper. Compared with ordinary time series images, the proposed method is more conducive to CNN recognition, and obtain meaningful results.
- This paper presents a simplified time series feature extraction method, which can automatically extract features from time series images during time series classification.

The pseudo code for our proposed framework is presented in Algorithm 1 below

Algorithm 1

Train the classification

Given:

$O = x_1, x_2, \dots, x_n$: the classification of n observed time series;

C : the set of CNNs (e.g. ResNet-18, VGG-19, DenseNet-121);

L : the set of class labels (e.g. ARIMA, ETS, THETA, etc.);

I : the set of time series images (e.g. MTF, RP and GAFs).

Output:

A time series preprocessing method with better classification rate.

Data preprocessing:

For $i=1$ to N

1. Split x_i into a training period, validation period and test period;

2. Transform x_i into four types of images: MTF, RP, and GAFs;

3. Four time series images of x_i are composed into one combined image based on feature fusion;

4. Fit L models to the training period, validation period and test period;

5. Calculate the forecasts for the training period, validation period and test period from each model;

6. Calculate the forecast error measure over the training period, validation period and test period for all models in L ;
 7. Select the model with the minimum forecast error as the labels.
- Prepare the classifier based CNN:
8. Input combined time series images into CNNs;
 9. Train the softmax classifier.
- Forecast a new time series
- Given:
- The data preprocessing method from step 3;
- the trained classifier from step 9.
- Output:
- Class labels from new time series x_{new} .
10. x_{new} repeat step 2 and step 3;
 11. Let combined image of x_{new} get most suitable label(forecasting model), and calculate the forecasting error MAPE.

2 Methods

Our proposed framework, presented in Fig. 1.

2.1 Theory of Information Fusion

Information fusion technology is an emerging data processing technology. Information fusion includes three levels: the pixel level, feature level and decision level. Among them, decision level fusion represented by the multiclassifier combination has become one of the research hotspots of pattern recognition, and has been successfully applied to handwritten character and face recognition [33]. Although the research on feature-level fusion did not earlier than the research on other fusion methods, it has made gratifying developments [34].

The first mock exam is based on feature-level fusion. Obviously, the different feature vectors extracted from the same pattern and the optimized combination of these different features can retain the effective recognition information of multiple features and eliminate

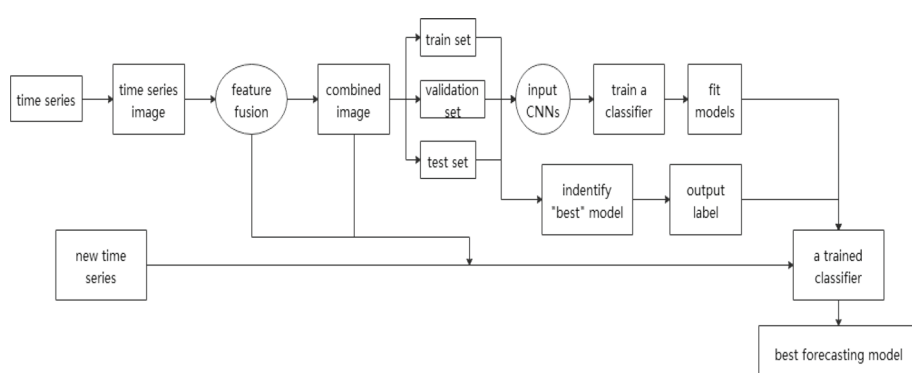


Fig. 1 The flow chart of this paper

redundant information to a certain extent, which is especially important for classification and recognition. There are two existing feature level fusion methods. One is to combine two sets of feature vectors into a joint vector, and then extract features in a high-dimensional real vector space. The other is to combine two groups of feature vectors with complex vectors to extract the features from complex vector space. Both feature fusion methods can improve the recognition rate. The feature fusion method based on a joint vector is called serial feature fusion, and the feature fusion method based on complex vector is called parallel feature fusion [35].

Suppose $\omega_1, \omega_2, \dots, \omega_c$ are c known pattern classes. Let $\{\Omega = \xi | \xi \in R^N\}$ be a training sample space. $A = \{x | x \in R^p\}$ and $B = \{y | y \in R^q\}$, where x and y are the two feature vectors of the same sample ξ extracted by different means. We will discuss the feature fusion in the transformed training sample feature space A and B . Suppose that A and B are regarded as two random vector spaces, we denote them as $\alpha_1^T x$ and $\beta_1^T y$ (the first pair), $\alpha_2^T x$ and $\beta_2^T y$ (the second pair), $\dots, \alpha_d^T x$ and $\beta_d^T y$ (the d th pair). Assume the following:

$$X^* = (\alpha_1^T x, \alpha_2^T x, \dots, \alpha_d^T x) = (\alpha_1, \alpha_2, \dots, \alpha_d)^T x = W_x^T x, \quad (1)$$

$$Y^* = (\beta_1^T x, \beta_2^T x, \dots, \beta_d^T x) = (\beta_1, \beta_2, \dots, \beta_d)^T x = W_y^T y, \quad (2)$$

X^* is a set of feature vectors of feature space A , and Y^* is a set of feature vectors of feature space B . Following two linear transformation (1) and (2):

$$Z_1 = \begin{pmatrix} X^* \\ Y^* \end{pmatrix} = \begin{pmatrix} W_x^T x \\ W_y^T y \end{pmatrix} = \begin{pmatrix} W_x & 0 \\ 0 & W_y \end{pmatrix}^T \begin{pmatrix} x \\ y \end{pmatrix} \quad (3)$$

$$Z_2 = X^* + Y^* = W_x^T + W_y^T = \begin{pmatrix} W_x \\ W_y \end{pmatrix}^T \begin{pmatrix} x \\ y \end{pmatrix} \quad (4)$$

Z_1 is a joint vector in high dimensional space and Z_2 is joint vector of the complex vector space. As the combinatorial feature projected, are used for classification, the transformation matrices are

$$W_1 = \begin{pmatrix} W_x & 0 \\ 0 & W_y \end{pmatrix} \quad \text{and} \quad W_2 = \begin{pmatrix} W_x \\ W_y \end{pmatrix} \quad (5)$$

where $W_x = (\alpha_1, \alpha_2, \dots, \alpha_d)$, and $W_y = (\beta_1, \beta_2, \dots, \beta_d)$.

2.2 Image Coding and Fusion of Time Series

This paper uses four algorithms to transform time series into images. They are Gramian angle summation field (GASF), Gramian angle difference field (GADF) [36], Markov transition field (MTF) [36], and Recurrence plot (RP) [37].

2.2.1 Gramian Angular Field

Given a time-series $X = \{x_1, x_2, \dots, x_n\}$, of n real-valued observations, normalize X so that all values are scaled at $[-1, 1]$ or $[0, 1]$ by:

$$\tilde{X}_{-1}^i = \frac{(x_i - \max(X)) + x_i - \min(X)}{\max(x) - \min(x)} \quad (6)$$

$$\tilde{X}_0^i = \frac{x_i - \min(X)}{\max(x) - \min(x)} \quad (7)$$

Therefore, by encoding the value of the time series \tilde{X} as angular cosine and the time point as radius, the normalized time series can be expressed in polar coordinates, and the formula is as follows:

$$f(x) = \begin{cases} \phi = \arccos(\tilde{x}_i), -1 \leq \tilde{x}_i \leq 1, \tilde{x}_i \in \tilde{X}_{-1}^i \\ r = \frac{t_i}{N}, t_i \in N \end{cases} \quad (8)$$

In the equation above, t_i is the time point and N is a constant parameter used to regularize the radius of the polar coordinate system, which is a novel time series visualization method. In the Cartesian coordinate system, the area formula is expressed as:

$$S_{i,j} = \int_{x(i)}^{x(j)} f(x(t)) dx(t), \quad (9)$$

where

$$S_{i,i+k} = S_{j,j+k} \quad (10)$$

if $f(x(t))$ has the same values on $[i, i+k]$ and $[j, j+k]$. However, in polar coordinates, if the area is defined as

$$S'_{i,j} = \int_{\phi(i)}^{\phi(j)} r[\phi(t)]^2 d\phi(t), \quad (11)$$

then $S'_{i,i+k} \neq S'_{j,j+k}$. That is, the corresponding area from time stamp i to time stamp j is not only dependent on the time interval $|i - j|$, but also determined by the absolute value of i and j .

Rescaled data in different intervals have different angular bounds. $[0, 1]$ corresponds to the cosine function in $[0, \frac{\pi}{2}]$, while cosine values in the interval $[-1, 1]$ fall into the angular bounds $[0, \pi]$. The GAF is defined as follows:

$$G = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \cos(\phi_1 + \phi_2) & \dots & \cos(\phi_1 + \phi_n) \\ \cos(\phi_2 + \phi_1) & \cos(\phi_2 + \phi_2) & \dots & \cos(\phi_2 + \phi_n) \\ \vdots & \vdots & \dots & \vdots \\ \cos(\phi_n + \phi_1) & \cos(\phi_n + \phi_2) & \dots & \cos(\phi_n + \phi_n) \end{bmatrix} \quad (12)$$

The Gramian angular summation field (GASF) and Gramian angular difference field (GADF) are defined as follows:

$$\begin{aligned} GASF &= [\cos(\phi_i + \phi_j)] \\ &= \tilde{X}' \cdot \tilde{X} - \sqrt{I - \tilde{X}^2} \cdot \sqrt{I - \tilde{X}^2} \end{aligned} \quad (13)$$

$$\begin{aligned} GADF &= [\sin(\phi_i + \phi_j)] \\ &= \sqrt{I - \tilde{X}^2} \cdot \tilde{X} - \sqrt{I - \tilde{X}^2} \cdot \tilde{X}' \end{aligned} \quad (14)$$

I is the unit row vector $[1, 1, \dots, 1]$. After transforming to the polar coordinate system, we take time-series at each time step as a 1-D metric space. By defining the inner product $\langle x, y \rangle = x \cdot y - \sqrt{1 - x^2} \cdot \sqrt{1 - y^2}$ and $\langle x, y \rangle = \sqrt{1 - x^2} \cdot y - \sqrt{1 - y^2} \cdot x$ two types

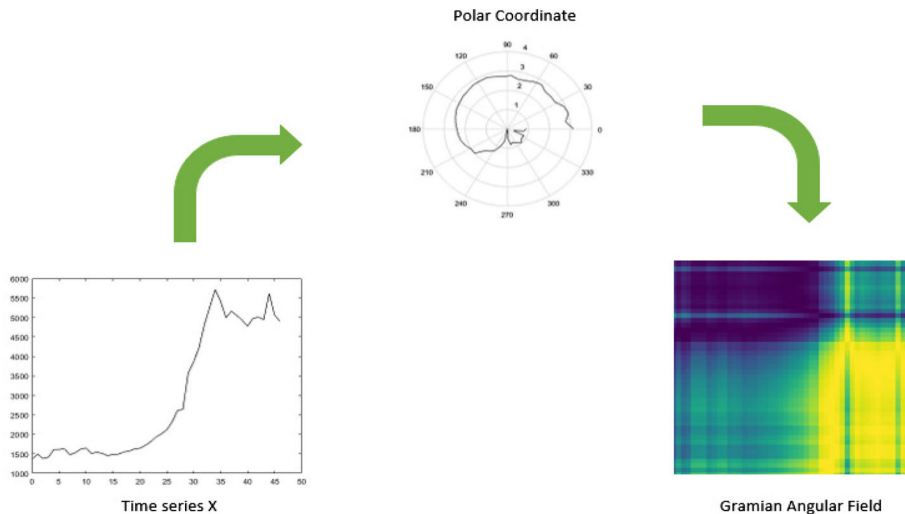


Fig. 2 Illustration of the proposed encoding map of Gramian angular fields. We take the GADF as an example, and the formation of the GASF is similar. X is a rescaled time-series in the M3 dataset. We transform X into a polar coordinate system by Eq. (8) and finally calculate its GASF images with Eq. (13)

of Gramian angular fields (GAFs) are actually quasi-Gramian matrices $[\langle x, y \rangle]$:

$$G = \begin{bmatrix} [\langle \tilde{x}_1, \tilde{x}_1 \rangle] & \dots & [\langle \tilde{x}_1, \tilde{x}_n \rangle] \\ \vdots & \dots & \vdots \\ [\langle \tilde{x}_n, \tilde{x}_1 \rangle] & \dots & [\langle \tilde{x}_n, \tilde{x}_n \rangle] \end{bmatrix} \quad (15)$$

The GAFs have several advantages. First, they provide a way to preserve temporal dependency, since time increases as the position moves from top-left to bottom-right. The GAFs contain temporal correlations because $G_{|i-j|=k}$ represents the relative correlation by superposition/difference of directions concerning time interval k . The main diagonal $G_{i,i}$ is the special case when $k = 0$, which contains the original value/angular information. From the main diagonal, we can reconstruct the time-series from the high-level features learned by the deep neural network. However, the GAFs are large because the size of the Gramian matrix is $n \times n$ when the length of the raw time-series is n .

The transformation maintains the time dependence between the values, and provides time correlation due to the superposition in the direction relative to the time interval, and the resulting matrix is bijective. Thus, the inverse function yields an absolute reconstruction of the original data, as we can see in Fig. 2.

2.2.2 Markov Transition Field

We get inspiration from Campanharo et al. [36]. Given a time-series X , we identify its Q quantile bins and assign each x_i to the corresponding bins $q_i (j \in [1, Q])$. Thus $Q \times Q$ weighted adjacency matrix W is constructed by counting the transitions among quantile bins in the manner of a first-order Markov chain along the time axis. $w_{i,j}$ is given by the frequency with which a point in quantile q_j is followed by a point in quantile q_i . After normalization by $\sum_j w_{i,j} = 1$ W is the Markov transition matrix. It is insensitive to the distribution of X and temporal dependency on time steps t_i . However, our experimental results on W demonstrate

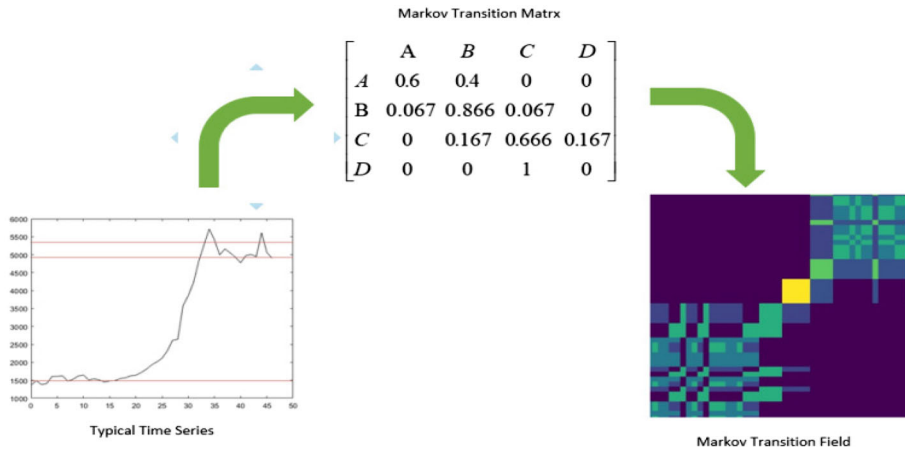


Fig. 3 Illustration of the proposed encoding map of Markov transition fields. X is a time-series in the M3 dataset. X is first discretized into Q quantile bins. In this image, we take $Q = 4$. Then we calculate its Markov transition matrix W and finally build its MTF with Eq. (16)

that getting rid of the temporal dependency results in too much information loss in matrix W . To overcome this drawback, the mathematical formula of Markov transfer field (MTF) is as follows:

$$G = \begin{bmatrix} \omega_{ij}|x_1 \in q_i, x_1 \in q_j & \omega_{ij}|x_1 \in q_i, x_2 \in q_j & \cdots & \omega_{ij}|x_1 \in q_i, x_n \in q_j \\ \omega_{ij}|x_2 \in q_i, x_1 \in q_j & \omega_{ij}|x_2 \in q_i, x_2 \in q_j & \cdots & \omega_{ij}|x_2 \in q_i, x_n \in q_j \\ \vdots & \vdots & \cdots & \vdots \\ \omega_{ij}|x_n \in q_i, x_1 \in q_j & \omega_{ij}|x_n \in q_i, x_2 \in q_j & \cdots & \omega_{ij}|x_n \in q_i, x_n \in q_j \end{bmatrix} \quad (16)$$

A $Q \times Q$ Markov transition matrix (W) is built by dividing the data into Q quantile bins. The quantile bins that contain the data at time stamp i and j (temporal axis) are q_i and q_j ($q \in [1, Q]$). $M_{i,j}$ in the MTF denotes the transition probability of $q_i \rightarrow q_j$ respectively. That is, by considering the time and location, matrix W is extended to an MTF matrix containing the transition probability on the magnitude axis. By assigning the probability from the quantile at time step i to the quantile at time step j at each pixel M_{ij} , the MTF actually encodes the multispan transition probabilities of the time-series. $M_{i,j|i-j|=k}$ denotes the transition probability between the points with time interval k . Figure 3 shows the procedure to encode time-series to MTF.

2.2.3 Recurrence Plot

In this part, we use recurrence plots (RPs) to encode time-series to images. Recurrence plots provide a way to visualize the periodic nature of a trajectory through a phase space [37], and can contain all relevant dynamical information in the time-series [38]. A recurrence plot of time-series x , can be formulated as

$$R(i, j) = \Theta(\epsilon ||x_i - x_j||) \quad (17)$$

where $R(i, j)$ is the element of the recurrence matrix R ; i indexes time on the x-axis of the recurrence plot, and j indexes time on the y-axis. ϵ is a predefined threshold, and $\Theta(\cdot)$ is the

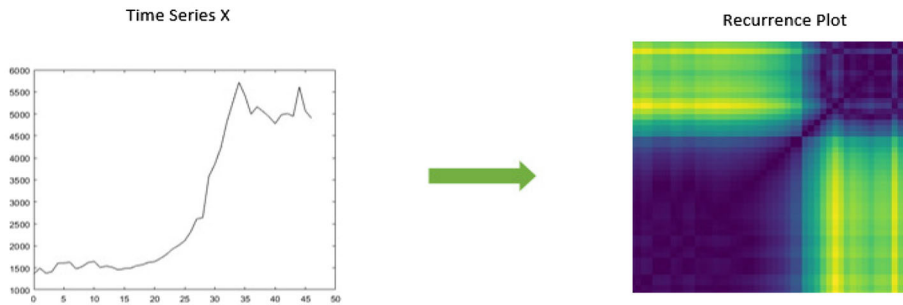


Fig. 4 Illustration of the proposed encoding map of recurrence plots. X is a sequence of time-series in the M3 dataset. We finally build its RP with Eq. (18)

Heaviside function. In short, one draws a black dot when x_i and x_j are closer than ϵ . Instead of a binary output, an unthresholded RP is not binary, but is difficult to quantify. We use the following modified RP, which balances the binary output and the unthresholded RP [39].

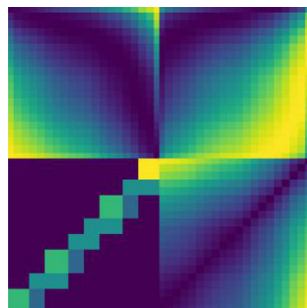
$$R(i, j) = \begin{cases} \epsilon, & ||x_i - x_j| > \epsilon| \\ ||x_i - x_j|, & \text{otherwise} \end{cases} \quad (18)$$

Figure 4 shows that it gives more values than a binary RP and results in coloured plots.

2.2.4 Image Fusion

In order to solve the problem of feature loss of a single image during CNN training and classification, the combined image algorithm converts the time series into four images, and combines them into a square image for feature level fusion. The combined image is GADF, GASF, MTF and RP from the top left to the bottom right. The advantage of this combination is first that, the sine function and cosine function of GAFs are bijective functions in their respective intervals, which can reflect the dependence of time series on time, and the GAFs matrix has the highest level characteristics at the diagonal. An MTF can encode the statistical information, including the transition probability characteristics of the first-order Markov chain of a time series. The recurrence plot provides a method to visualize the periodicity of trajectories through phase space, and contains all relevant dynamic information in time series. Second, the edges and diagonals of time series images contain more abundant classification features. Using the above feature fusion method, the four groups of feature vectors are combined into a joint vector, thus expanding the feature vector space of the original image, which is conducive to the deep learning framework for its classification, so as to improve the classification rate. In Fig. 5, the boundaries of the four combined images can be clearly seen. After the images are spliced, half of the boundary of the original image becomes the combination point of the combined image (for example, the right boundary and lower boundary of the upper left corner image have become the core part of the combined image).

Fig. 5 Combined image of feature fusion



3 Combined Image Recognition Based on Deep Learning

3.1 Convolutional Neural Network (CNN)

In this paper, three deep learning frameworks are applied to test the generalization performance of the proposed algorithm. The three deep convolution neural network models have different network depth and network structures, so if the algorithm can perform well in the three convolution neural network models, it can be applied to other deep learning models. [39].

Basic idea of Residual Learning

In [40], He et al. proposed an improved CNN model for image classification, called a deep residual network. The main difference between a residual network and the classic CNN is that residual networks have a unique network structure, as shown in Fig. 6. The classic CNN model, combines basic units such as convolution, nonlinear mapping, merging or batch normalization in a cascading manner. However the residual network, uses a direct method that can directly connect the input and output in the building block. Mathematically speaking, residual learning is different from directly approximating the basic function $H(x)$, and residual learning focuses on fitting its residual mapping $F(x)$, where:

$$F(x) = H(x) - x \quad (19)$$

The final mapping of a residual learning block is $F(x) + x$, which is the output of a traditional CNN, namely $H(x)$. However, as stated by He et al. in [41], the fitted residual mapping $F(x)$ is more effective than the original mapping $H(x)$, especially when $H(x)$ is an identity or approximate identity mapping. The characteristics of the residual network will increase the depth greatly, but will not reduce the classification accuracy of the network.

Basic Idea of VGGNet

The visual geometry group network (*VGGNet*) is a deep neural network with multilayered operations. Since the 3×3 convolutional layer is set on the top and increases as the depth increases, VGGNet is very effective. In order to reduce the volume, the max pooling layer is used as a handler in the *VGGNet*. Two FC layers with 4096 neurons were used. The network is shown in Fig. 7.

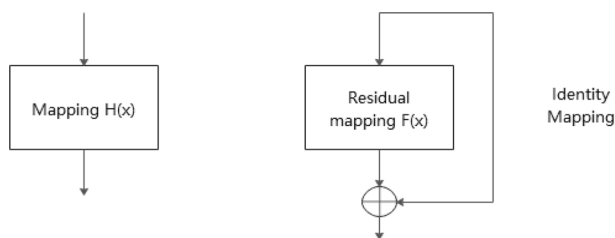


Fig. 6 Basic building blocks in different CNN models. Left: a basic building block in a typical CNN model. Right: a basic building block in a residual network

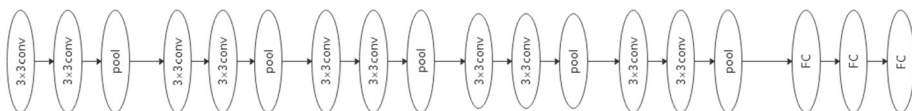


Fig. 7 The architecture of VGGNet

In the training phase, convolutional layers were used for the feature extraction and max pooling layers associated with some of the convolutional layers to reduce the feature dimensionality. In the first convolutional layer, 64 kernels (3×3 filter size) were applied for feature extraction from the input images. Fully connected layers were used to prepare the feature vector. Finally, in the testing phase, based on the softmax activation technology, cross validation is used to classify the images.

VGGNet systematically studied the influence of the network depth on image recognition performance, and constructed and pretrained deeper structures on the basis of shallow networks. Finally, two successful network architectures were proposed in the ImageNet Challenge: *VGG-16* (13 convolutional layers and 3 fully connected layers) and *VGG-19* (16 convolutional layers and 3 fully connected layer). More details can be found in the original paper [41].

Basic Idea of DenseNet

The recently proposed CNN architecture DenseNet [42] has an improved connection mode: in a dense block, each layer is connected to all other layers. In this case, all layers can access the feature map with each other, which reflects the correlation of features. Therefore, the model is more compact and not easy to overfit. In addition, each layer directly accepts the supervision of the loss function through shortcuts, thus providing implicit in-depth supervision. All these good attributes make DenseNet a natural fit for the recognition problem of each pixel. Using DenseNet for image recognition is a parallel work that can achieve the best performance without pretraining or other processing.

Traditional CNNs, such as FlowNets, calculate the output of the l^{th} layer by applying a nonlinear transformation H to the previous layer's output x_{l-1}

$$x_l = H_l(x_{l-1}) \quad (20)$$

Through continuous convolution and pooling, the network achieves spatial invariance and obtains coarse image features at the top. However, fine image details usually disappear at the top of the network. In order to improve the information flow between layers, DenseNet provides a simple connection mode: the l^{th} layer receives the feature mapping of all previous layers as input.

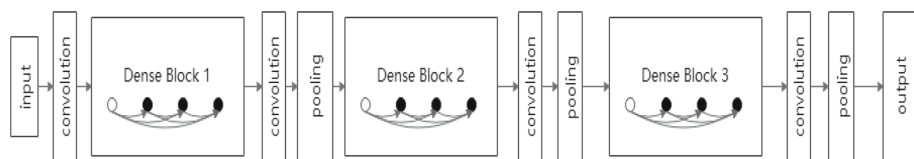


Fig. 8 The architecture of DenseNet

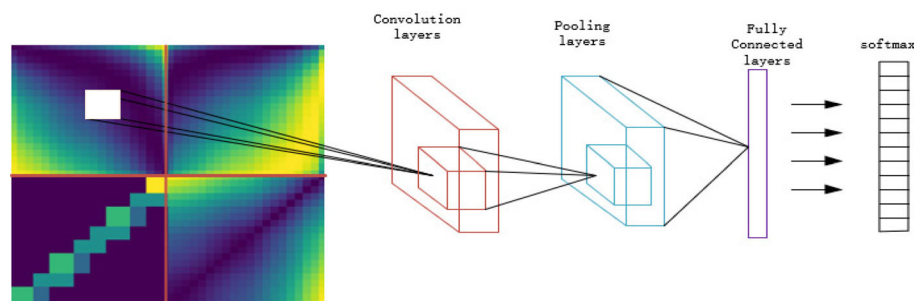


Fig. 9 The figure simply shows the network structure of the CNN, and focuses on the operation of the convolution kernel (white part) on the image, which can repeatedly extract features from the edge part (red line)

$$x_l = H_l(x_0, x_1, \dots, x_{l-1}) \quad (21)$$

where $[x_0, x_1, \dots, x_{l-1}]$ is a single tensor formed by concatenating the output feature maps of the previous layer. In this way, even the last layer can access the input information of the first layer to realize feature exchange. Through the shortcut connection, the loss function directly monitors all layers. $H_l(\cdot)$ is a continuous operations of composite functions, batch normalization (BN), leaky rectified linear units (LReLU), a 3×3 convolution and dropout. We denote such composite function as one layer. This is shown in Fig. 8.

3.2 Recognition of Combined Images Based on Deep Learning Framework

Li Xixi et al. [27] proposed the idea of transforming time series into a single image and using deep learning model for classification. In order to study the advantages of the feature fusion of time series combined images under convolution neural network model, the above three CNN models are used for the classification of combined images, and the classification accuracy is compared with those of single image methods under the same CNNs. The diagonal and border of time series images have a large number of effective features, which are often difficult to fully extract in a single image. The convolution kernel of the three CNN models can repeatedly extract the boundary and diagonal features of the combined image when convoluting the combined image, so as to reduce the loss of features and have more complete and richer boundary features than a single image, finally improving the image classification rate. The feature vectors of four images are combined into one feature vector [30]. The deep learning framework has greater discrimination in feature vector classification, so as to improve the classification rate. We focus on the operation of the convolution kernel (white part) on the image, which can repeatedly extract features from the edge part (red line), as shown in Fig. 9.

3.3 Evaluation Criteria

There are two criteria used to verify the superiority of this algorithm: the classification rate and forecasting error. The classification rate compares between the labels obtained by inputting the test set into the trained classifier and the best label. Because the supervised learning label used in this paper is the best forecasting model, that is, the final classification result can be regarded as the forecasting model selection problem. The forecasting error used in this paper is the mean absolute percentage error (MAPE). The benchmark models in this paper are four different single image generation methods and six econometric model methods.

$$MAPE = \sum_{t=1}^n \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right| \times \frac{100}{n} \quad (22)$$

where Y_t is the real value of the time-series at point t , \hat{Y}_t is the forecast, n is the forecasting horizon.

The classification accuracy can be expressed as

$$accuracy = \frac{(TP + TN)}{All} \quad (23)$$

where true positives (TP) is the number of positive examples correctly divided, and true negatives (TN) is the number of negative cases correctly divided.

4 Experimental Verification

4.1 Time Series Classification

As shown in Fig. 1, in the training phase, four computer vision methods are used to generate four types of images from the time series in the training set. Later, information fusion is applied to the four types of images to form a combined image. Then three CNNs are used as classifiers to train the combined image and classify the combined image into the corresponding category. Because the purpose of this paper is to forecast the time series, the classification is the corresponding optimal forecasting model. In the test phase, the new time series are also transformed into combined images in the same way. Furthermore, the new combined images are input into the trained classifier to get their optimal forecasting models so as to make time series predictions.

4.2 Classified Data Sets and Forecasting Models

The time series of the M1 and M3 competitive datasets were selected for classification. The M1 and M3 competition datasets, including more complex microeconomic and industrial data, are conducive to verifying the generalization ability of the proposed method. The specific classification is shown in Tables 1 and 2. Here 80% of the original time series in M3 is divided into the training set and 20% is divided into verification set, and the M1 data set is defined as the test set. Since the classification is supervised learning, the label of each time series is its optimal prediction model. Here we choose six econometric prediction models: ARIMA, ETS, RW, drift random walk (RW-d), the seasonal component index smoothing algorithm (ETS-s) and white noise (WN). Each time series is marked in the training set M3, then the

Table 1 Category of 111 datasets of the M1 competition

Types	Yearly	Quarterly	Monthly	Total
Micro	6	5	22	33
Industry	4	2	21	27
Macro	7	11	17	35
Demographic	3	5	8	16
Total	10	23	68	111

Table 2 Categories of the 3003 datasets of the M3 competition

Types	Yearly	Quarterly	Monthly	Other	Total
Micro	146	204	474	4	828
Industry	102	83	334		519
Macro	83	336	312		731
Finance	58	76	145	29	308
Demographic	245	57	111		413
Other	11		52	141	204

model is trained, and then the data from test set M1 are input into the trained classifier, so that each time series in the M1 dataset will be classified into corresponding labels.

4.3 Experimental Design and Parameter Setting

In our experiment, we used Python 3.7 and R. The size of the four types of single pictures is 359×359 , and after resizing the size of combined image (GADF-GASF-MTF-RP) is also 359×359 . The parameters for pretrained CNN models are set as follows:

Dimension of the output of the pretrained VGG-11: 1000.

Dimension of the output of the pretrained ResNet-18: 512.

Dimension of the output of the pretrained DenseNet-121: 1000.

The iteration rate of the CNNs is 0.001, and the batch size is 16.

4.4 The Benchmark Models

The benchmark models in this paper are the support vector machine (SVM), dynamic time warping (DTW), multilayer perceptron (MLP). The SVM and DTW represent two different machine learning time series classification algorithms, the MLP represents a deep learning time series classification method, and the SVM + combined image represents a time series image classification method without deep learning processing.

4.5 Experimental Results and Analysis

The Superiority of the Improved Algorithm in the Classification Rate

When step $h = 1$, as shown in Tables 3 and 9, the highest classification rate is 45.95% under Densenet-121 deep learning framework, and the average probability is 16.7% in the six classification problems. Compared with the classification rate of SVM (38.15%) and DTW (36.44%) of machine learning classification algorithm and the classification rate of the MLP

Table 3 Comparison of the classification rates between the combination image algorithm and traditional image algorithms when the step size $h = 1$

Classification rate	ResNet-18	DenseNet-121	VGG-11
GASF	0.3743	0.3604	0.4324
GADF	0.3694	0.3514	0.3423
MTF	0.3379	0.2973	0.4324
RP	0.3238	0.3694	0.3694
Combined	0.4234	0.4595	0.4414

The best experimental results are shown in bold

Table 4 The forecasting error MAPE of the six individual prediction models when the step size $h = 1$

Forecasting model	MAPE
WN	15.6154
RW	13.4052
RW-d	13.2679
ETS-s	12.5063
ETS	12.3900
ARIMA	11.7136
Average	13.1497

The best experimental results are shown in bold

Table 5 Forecasting error MAPE of the CNN + image method when step $h = 1$

MAPE	ResNet-18	DenseNet-121	VGG-11
RP	12.2832	12.3311	12.0918
MTF	12.0618	12.0978	12.0221
GADF	12.0199	11.9888	11.9031
GASF	11.9675	11.8007	11.8898
Combined	11.8994	11.7084	11.6777

The best experimental results are shown in bold

Table 6 Comparison of the classification rate between the combination image algorithm and traditional image algorithms when the step size $h = 3$

Classification rate	ResNet-18	DenseNet-121	VGG-11
GASF	0.3874	0.3243	0.3874
GADF	0.3333	0.3243	0.3604
MTF	0.3694	0.3243	0.3694
RP	0.4054	0.4234	0.3604
Combine	0.4054	0.4234	0.4324

The best experimental results are shown in bold

(40.06%) deep learning classification algorithm, the CNN + combined proposed in this paper has better effect. At the same time, the CNN + combined has the same advantages as SVM + combined algorithm without deep learning processing. When step $h = 3$, CNN + combined algorithm is still better than the other methods, which shows that the algorithm proposed in this paper is universal.

Table 7 The forecasting error MAPE of the six individual prediction models when the step size $h = 3$

Forecasting model	MAPE
WN	21.3482
RW	19.3629
RW-d	19.0634
ETS-s	17.7794
ETS	17.3867
ARIMA	16.1350
Average	18.5126

The best experimental results are shown in bold

Table 8 The forecasting error MAPE of CNN + image method when the step $h = 3$

MAPE	ResNet-18	DenseNet-121	VGG-11
RP	16.5730	16.9400	16.5806
MTF	16.5899	16.4698	17.0998
GADF	16.3339	16.5496	16.1986
GASF	16.6458	16.1796	16.2884
Combined	16.9910	16.0527	16.1854

The best experimental results are shown in bold

Table 9 The classification rate of the benchmark models with different step sizes

Classification rate	$h = 1$	$h = 3$
SVM	0.3815	0.4024
DTW	0.3644	0.3912
MLP	0.4006	0.4210
SVM+combined image	0.3783	0.3895

Table 10 Average time consumed in one iteration

Duration/s	GASF	GADF	MTF	RP	Combined
VGG-11	7.53	7.51	7.51	7.54	7.53
ResNet-18	8.31	8.29	8.31	8.31	8.33
DenseNet-121	10.38	10.38	10.37	10.38	10.37

Table 11 Average generation rate of single image

	GASF	GADF	MTF	RP	Combined
duration/s	5.53	5.48	5.51	5.50	5.53

Error Analysis of Single Forecasting Models

The experimental results MAPE of the six forecasting models in Tables 4 and 6 can be used to briefly analyse the time series. For example, the MAPE of white noise in two tables is the maximum, which implies that the time series has correlation and trend, but there is no randomness. The error of the random walk (RW) is smaller than that of RW-d, which indicates that time series is fractal and obeys Brownian motion. The error of the ETS model is smaller

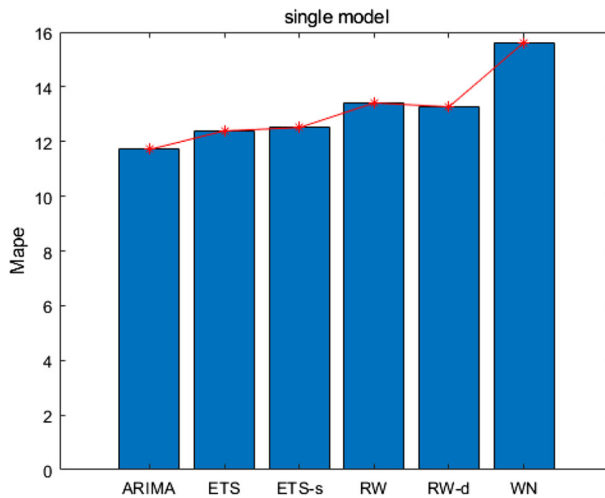


Fig. 10 MAPE histogram of the single models

than that of ETS-s model, which indicates that time series is not seasonal. That is, periodicity is weak.

Error Comparison of Combined Image Algorithm with the Traditional Image Algorithm and Traditional Forecasting Model

From Tables 4 and 5, the forecasting error MAPE after the combined image + CNN forecasting model selection is less than that after traditional image + CNN. The error of VGG-11 is the smallest, which is approximately 11.678. In order to show the error of each model more intuitively, we draw the MAPE in a column chart. As shown in Figs. 10, 11, 12 and 13, after a large number of experiments and analysis, we find that the uncertainty of the volatility and time series will lead to some changes in time series images. On the whole, most of the results support our experimental hypothesis, this small fluctuation will not affect the overall results.

Robustness and Generalization Ability of the Combinatorial Image Algorithm

Tables 5, 8 show that as the network level deepens, the training effect does not improve. This shows that the deepening of the network level is often accompanied by a surge in parameters, which will cause overfitting in the case of insufficient data. When step $h = 1$, the structure of the 11 layer VGG-11 deep learning network is most suitable for the fluctuation of time series. When the step size $h = 3$, DenseNet-121 has the smallest error. Therefore, it is very important to choose the appropriate deep learning framework. In this experiment, we did not separate the data according to the data type, but integrated all the data together. Therefore, the algorithm can be applied to a variety of data types, and can be applied to different fields.

The Comparison of the Combined Image and Time Series Image Algorithm Under Different CNNs

When the step size $h = 1$, the classification rate of the combined image algorithm is 7.2%, 11.49% and 4.73% higher than time series image algorithm under ResNet-18, DenseNet-121 and VGG-11, respectively. Furthermore, when the step size increases to $h = 3$, the classification rate of the combined image algorithm is 3.15%, 7.5% and 6.3% higher than those of the time series image algorithm under ResNet-18, DenseNet-121 and VGG-11, respectively. According to the experimental results, we can see that the promotion effect of DenseNet-121

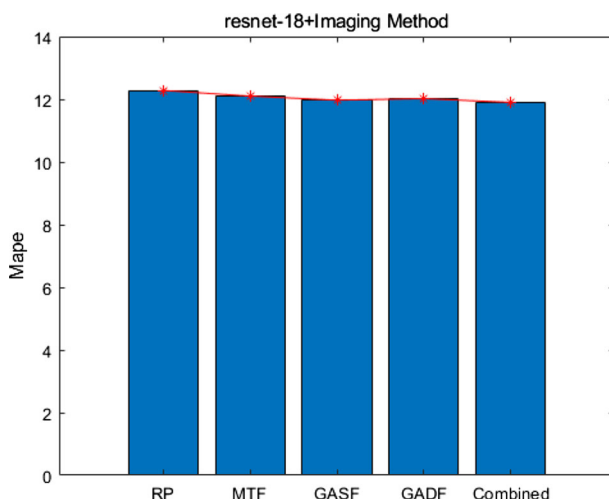


Fig. 11 MAPE histogram of ResNet-18 + imaging method

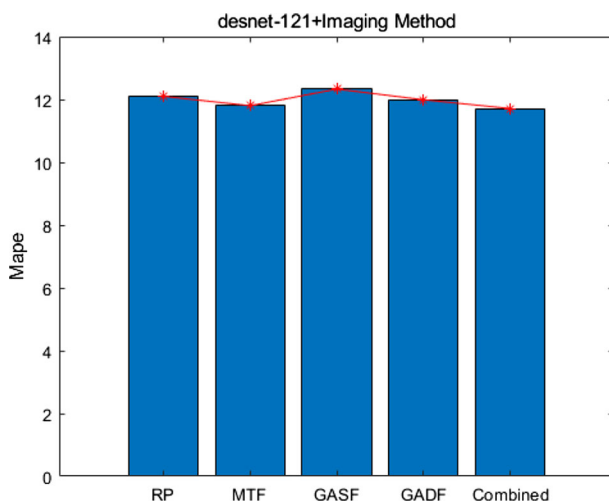


Fig. 12 MAPE histogram of DesNet-121 + imaging method

is the best, because DenseNet-121 has a dense network structure and can extract time series combined image features at multiple levels to achieve the best classification rate.

The Comparison of Combined Image and Time Series Image Algorithm Under Different CNNs

Table 10 shows the average running time of a single different time series image under the three CNN frameworks. As introduced in the experimental setup, this paper uses the control variable method to keep the combined image consistent with the size of other images. The experimental results also show that the image generation speed is almost the same in the same CNN. In contrast, in the CNN with deep network layers, the training time of images is longer, which may be because the deep network structure is more complex and the feature extraction is more sufficient. Table 11 shows that the average generation rate of single image

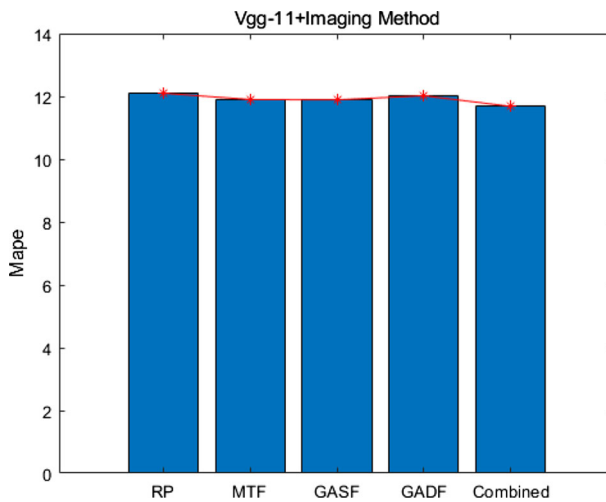


Fig. 13 MAPE histogram of VGG-11 + imaging method

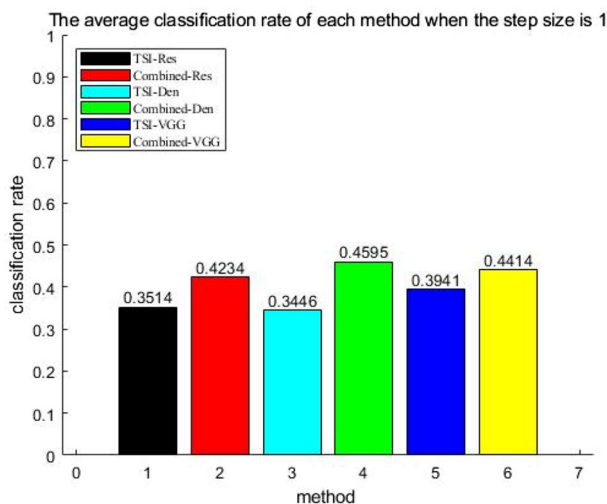


Fig. 14 The average classification rate of each method when the step size is 1

is almost the same. The generation rate of time series image is mainly related to image size and computer hardware, but has little to do with image transformation mode.

4.6 The Insufficiency of the Experiment

Tables 4 and 5 show that the overall results basically meet the assumptions of this paper, but when the combined image algorithm proposed in this paper inputs ResNet-18, the error is greater than that of the single model. After analysing the relevant literature, we find that for the short-term forecasting of time series, the effect of the residual network is not as good as those of DenseNet and VGG. The reason may be that the residual network is good at

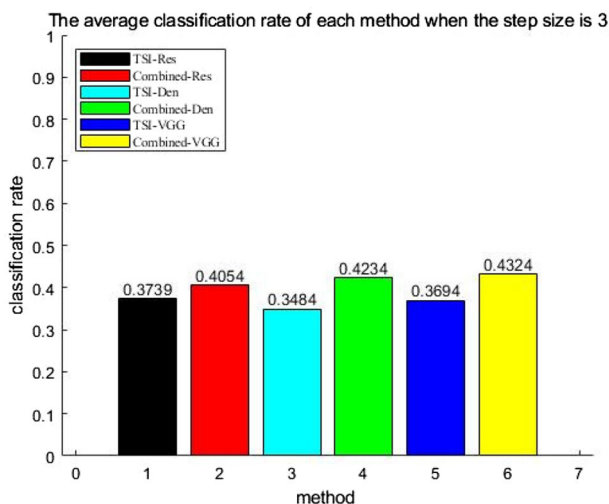


Fig. 15 The average classification rate of each method when the step size is 3

extracting multi-level features in images. When step $h = 1$, the images generated by time series have less deep feature information. Tables 7 and 8 show that the combined image algorithm has better results and only cooperates with DenseNet-121. That is, as the step size increases, the fluctuation of the time series increases, and the image feature information becomes more complex. Therefore, we need a deeper CNN and DenseNet-121 is suitable.

5 Conclusions and Future Work

A new time series image transformation method called combined image based on feature fusion is proposed in our paper. The proposed method enables automated feature exaction, making it more flexible than using manually selected time series features. Furthermore, the proposed method also improves the utilization of images in convolutional neural networks, reduces the feature extraction loss, and improves the time series image classification accuracy. Therefore, the time series can be accurately assigned to the best forecasting model. The experimental results show that the time series combined image + CNN algorithm proposed in this paper achieves better classification rate and forecasting error performance than that of ordinary time series images in CNN. Compared with traditional time series classification algorithms (SVM and DTW), the deep learning time series classification algorithm (MLP) and the non-deep learning time series image classification algorithm (SVM + image), the proposed method also achieve better results.

In other words, the algorithm proposed in this paper makes a certain contribution to time series classification and model selection. However, there are some shortcomings. In the future work, the author will further analyse the algorithm based on the time complexity of deep learning training. Furthermore, deep learning lightweight technology will be used to improve the algorithm.

Acknowledgements This work was supported by the National Natural Science Foundation of China (71971089, 72001083) and Natural Science Foundation of Guangdong Province (No. 2022A1515011612).

Author Contributions WJ Software, Data curation, Writing, Methodology. DZ Writing-review, Supervision. LL Conceptualization, Methodology, Writing—review, Supervision. RL Software, Data curation.

Funding Natural Science Foundation of Guangdong Province (No.2022A1515011612). Thanks for your cooperation.

Declaration

Conflict of interest All Authors declar that they have no conflict of interest.

Ethical approval This artical does not contain any studies with human participants or animals performed by any of the authors.

References

1. Yang S, Zheng X, Ji C, et al (2021) Multi-layer representation learning and its application to electronic health records. *Neural Process Lett* 1–17
2. Liu B, Zhang Z, Cui R (2020) Efficient time series augmentation methods. In: 2020 13th international congress on image and signal processing, BioMedical engineering and informatics (CISP-BMEI)
3. Auge D, Hille J, Mueller E et al (2021) A survey of encoding techniques for signal processing in spiking neural networks. *Neural Process Lett* 53:4693–4710
4. Ghanem W, Jantan A (2019) Training a neural network for cyberattack classification applications using hybridization of an artificial bee colony and monarch butterfly optimization. *Neural Process Lett* 51:905–946
5. Zg A, Vk B, Mi B et al (2020) Weighted kNN and constrained elastic distances for time-series classification - ScienceDirect. *Expert Syst Appl* 162:113829
6. Lines J, Bagnall A (2015) Time series classification with ensembles of elastic distance measures. *Data Min Knowl Disc* 29(3):565–592
7. Xiao X, Lu Y, Huang X et al (2021) Temporal series crop classification study in rural china based on sentinel-1 SAR data. *IEEE J Sel Top Appl Earth Obs Remote Sens* 99:1–1
8. Geler Z, Kurbalija V, Ivanovic M, et al. (2020) Time-series classification with constrained DTW distance and inverse-square weighted k-NN. In: 2020 international conference on innovations in intelligent systems and applications (INISTA)
9. Lines J, Bagnall A (2015) Time series classification with ensembles of elastic distance measures. *Data Min Knowl Disc* 29(3):565–592
10. Buza K, Nanopoulos A, Schmidt-Thieme L (2011) Time-series classification based on individualised error prediction. In: IEEE international conference on computational science engineering. IEEE
11. Morchen F, Ullsch A, Thies M et al (2006) Modeling timbre distance with temporal statistics from polyphonic music. *IEEE Trans Audio Speech Lang Process* 14(1):81–90
12. Wang X, Smith K, Hyndman R (2006) Characteristic-based clustering for time series data. *Data Min knowl discov* 13(3):335–364
13. Lin J, Keogh E, Wei L et al (2007) Experiencing SAX: a novel symbolic representation of time series. *Data Min Knowl Discov* 15(2):107–144
14. Baydogan MG, Runger G, Tuv E (2013) A bag-of-features framework to classify time series. *IEEE Trans Pattern Anal Mach Intell* 35(11):2796–2802
15. 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 Disc* 31(3):606–660
16. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521:436–444
17. Xu J-L et al (2021) Deep learning for classification of time series spectral images using combined multi-temporal and spectral features. *Anal Chim Acta* 1143:9–20. <https://doi.org/10.1016/j.aca.2020.11.018>
18. Yang C, Jiang W, Guo Z (2019) Time series data classification based on dual path cnn-rnn cascade network. *IEEE Access* 7:155304–155312
19. Gupta S, Kumar M, Garg A (2019) Improved object recognition results using SIFT and ORB feature detector. *Multimed Tools Appl* 78(23):34157–34171
20. Kumar M, Gupta S (2021) 2D-human face recognition using SIFT and SURF descriptors of face's feature regions. *Vis Comput* 37(11)

21. Chhabra P, Garg NK, Kumar M (2020) Content-based image retrieval system using ORB and SIFT features. *Neural Comput Appl* 32(7):2725–2733
22. Kumar M, Kumar R, Saluja K K, et al. (2021) Gait recognition based on vision systems: a systematic survey. *J Vis Commun Image Represent* 75(6)
23. Goldberg Y (2016) A primer on neural network models for natural language processing. *Artif Intell Res* 57(1):345–420
24. Bansal M, Kumar M, Kumar M, et al. (2020) An efficient technique for object recognition using Shi-Tomasi corner detection algorithm. *Soft Comput* 1–10
25. Kumar M, Bansal M, Kumar M (2020) XGBoost: 2D-object recognition using shape descriptors and extreme gradient boosting classifier. In: *International conference on computational methods and data engineering (ICMDE 2020)*
26. Kumar Munish, Chhabra et al (2018) An efficient content based image retrieval system using BayesNet and K-NN. *Multimed Tools Appl* 77(16):21557–21570
27. Garg Diksha, Naresh et al (2018) Underwater image enhancement using blending of CLAHE and percentile methodologies[J]. *Multimed Tools Appl* 77(20):26545–26561
28. Lecun Y, Boser B, Denker J et al (2014) Backpropagation applied to handwritten zip code recognition. *Neural Comput* 1(4):541–551
29. Pereira S, Pinto A, Alves V, Silva CA (2016) Brain tumor segmentation using convolutional neural networks in mri images. *IEEE Trans Med Imaging* 35(5):1240–1251
30. Campanharo ASLO, Sirer MI, Malmgren RD, Ramos FM, Amaral LAN (2011) Duality between time series and networks. *PLoS ONE* 6(8):1–13
31. Wang Z, Oates T (2015) 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*
32. Li X, Kang Y, Li F (2020) Forecasting with time series imaging. *Expert Syst Appl* 1(3):113–130
33. Akbar S, Ali F, Khan S et al (2020) Deep-AntiFP: prediction of antifungal peptides using distant multi-information fusion incorporating with deep neural networks. *Chemom Intell Lab Syst* 208:104214
34. Zhang X, Zhao H (2021) Hyperspectral-cube-based mobile face recognition: a comprehensive review. *Inf Fusion* 74(24)
35. Ye TA, Xm A, Hc A et al (2021) Using Z-number to measure the reliability of new information fusion method and its application in pattern recognition. *Appl Soft Comput* 111:107658
36. Campanharo AS, Sirer MI, Malmgren RD, Ramos FM, Amaral LAN (2011) Duality between time series and networks. *PLoS ONE* 6(8):233–248
37. Eckmann JP, Kamphorst SO, Ruelle D (1987) Recurrence plots of dynamical systems. *Europhys Lett* 4(9):973–977
38. Thiel M, Romano MC, Jürgen Kurths (2004) How much information is contained in a recurrence plot? *Phys Lett A* 330(5):343–349
39. Xu JL, Hugelier S, Zhu H et al (2020) Deep learning for classification of time series spectral images using combined multi-temporal and spectral features. *Anal Chim Acta* 1143:9–20
40. He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. *IEEE Comput Soc* 33(5):243–249
41. Jaworek-Korjakowska J, Kleczek P, Gorgon M (2019) Melanoma thickness prediction based on convolutional neural network with VGG-19 model transfer learning. In: *2019 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW)*. IEEE
42. Zhu Y, Newsam S (2017) DenseNet for dense flow. *Comput Sci* 6(2):790–794