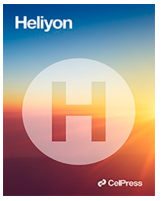
Heliyon 10 (2024) e37655

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

Relative position matrix and multi-scale feature fusion for writer-independent online signature verification

Fangjun Luan [a,b,c], Weiyi Cao [a,b,c], Shuai Yuan [a,b,c,*]

[a] School of Computer Science and Engineering, Shenyang Jianzhu University, Shenyang, China
[b] Liaoning Province Big Data Management and Analysis Laboratory of Urban Construction, Shenyang, China
[c] Shenyang Branch of National Special Computer Engineering Technology Research Center, Shenyang, China

ARTICLE INFO

Keywords:
Writer-independent online handwritten signature verification
Multi-scale feature fusion
Relative position matrix
Siamese neural network

ABSTRACT

Online signature verification (OSV) is widely used in finance, law and other fields, and is one of the important research projects on biological characteristics. However, its data set has a small scale and has high requirements for generalization of certification models. Therefore, how to overcome these problems is of great value to improve the practicality and security of online handwriting signature technology. We propose a writer-independent online handwritten signature verification method, which adopts the relative position matrix method to convert the traditional temporal features into images for processing. This method enriched the features of the signatures, serving the purpose of data augmentation. Then two-dimensional multi-scale feature fusion based Siamese neural network (2D-MFFnet) is built for representing and learning the importance of each channel adaptively combined with the attention mechanism. Finally, a temporal convolutional network is designed to construct the classifier. The results illustrate that compared with traditional time series models, the algorithm has reduced the equal error rate by at least 2.52 % on the open datasets MCYT-100 and SVC2004 task2.

1. Introduction

As a biological behavior characteristic recognition method, handwriting signature verification is adopted to perform identification and authorization according to the user unique signature habits and handwriting characteristics, which ensures that only legal authorization users can access sensitive information or perform specific operations. According to different signature sampling techniques, handwritten signature verification can be categorized into offline and online methods [1]. Offline signatures only store the static image of the signature. In contrast, online handwritten signature verification captures dynamic behavioral features such as the x and y coordinates, pen-tip pressure, and changes in pen lifts and drops, offering a richer set of features and therefore obtaining higher security. They are widely applied in financial services, legal documents, electronic commerce, and other fields [2]. However, the available signature datasets in the field of online handwritten signature verification are generally small in scale, so efficiently utilizing the existing data to achieve higher verification accuracy is an urgent problem to be addressed.

Forgery of signatures generally includes random forgery, simple forgery, and skilled forgery [3]. Random forgery refers to the forger not knowing the relevant information about the true signature and substituting the forger's signature for the true one, thus conducting a random forgery; simple forgery occurs when the forger is aware of the user's real signature but signs according to their

* Corresponding author. School of Computer Science and Engineering, Shenyang Jianzhu University, Shenyang, China.
E-mail address: reidyuan@163.com (S. Yuan).

https://doi.org/10.1016/j.heliyon.2024.e37655
Received 15 April 2024; Received in revised form 6 September 2024; Accepted 7 September 2024
Available online 10 September 2024

own writing habits. Although the name signed is the same, there are significant differences in the signature's form and handwriting; skilled forgers, through observing the dynamic information of the real signature and practicing, can imitate it with minimal differences. Additionally, even when the same user is writing a signature, it can be easily affected by factors such as the writing device, the writing environment, and the psychological state at the time of writing, which leads to significant differences between the written signature and the true signature [4]. Therefore, an online handwritten signature verification system should adequately measure the inter-class and intra-class variability of signatures, allow the characteristics of the true signature to fluctuate within a certain range, while effectively distinguishing forged signatures to improve the accuracy of signature verification.

From the perspective of the verification model, handwritten signature verification models can be further divided into writer-dependent models and writer-independent models [5]. Writer-dependent verification techniques require training a separate model for each individual, which can waste a substantial amount of storage space and computational resources in practical applications. Writer -independent models are trained on existing training samples, and when registering new users, no additional network training is necessary, which makes them more valuable for applications. At the same time, the performance requirements for these writer -independent models are more stringent. The models need to have stronger representational learning capabilities for ensuring their ability of generalizing signature outside of the training data.

Via long-term research and exploration of researchers, the field of online handwritten signature verification has achieved significant development. The discrimination methods for online handwritten signatures can be divided into traditional methods and deep learning methods. Among traditional methods, the Dynamic Time Warping (DTW) algorithm is the most widely used. Khalil et al. [6] employed the DTW method for signature verification using single and multiple feature combinations. Experiments represented that using signature curvature changes and speed features could effectively improve the signature verification success rate. Parziale et al. [7] proposed a DTW algorithm based on signature stability regions (SM-DTW), which uses stroke segmentation to infer the stable regions of a signature, assigns greater weight to stable regions than ordinary regions, and incorporates this into the DTW computation. Experimental results indicate that this algorithm improved the performance of the baseline system. Liu et al. [8] used Discrete Cosine Transform for feature transformation, extracting more effective features to enhance signature verification accuracy. Okawa [9] utilized mean template methods for online signature verification and proposed a time-series averaging method, namely Euclidean barycenter-based DTW barycenter averaging (EB-DBA). Hefny et al. [10] used Legendre polynomials to extract online signature features, enabling traditional methods to achieve high accuracy without the need for extensive training data.

Traditional methods have been proven to achieve satisfactory accuracy in previous studies. However, they also have certain limitations, such as the tedious manual extraction of features, which requires substantial prior expert knowledge.

In recent years, with the rapid development of deep learning in the fields of machine learning and pattern recognition, methods based on deep learning have begun to receive widespread attention in the area of handwritten signature verification. Lai et al. [11] applied RNN and joint learning to online signature verification and proposed a new descriptor called Length Normalized Path Signature (LNPS), which was applied to online signature verification. Experiments have verified that LNPS possesses characteristics such as scale and rotation invariance after linear combination, representing strong application potential in online signature verification tasks. Shen et al. [12] proposed a multi-scale residual attention mechanism module based on a Siamese network for automatically extracting multi-scale features of signatures. They constructed an ABSoftmax classifier using an adaptive boost (AdaBoost) algorithm to realize an integrated decision-making process for writer-independent online signature verification, thus improving the accuracy of online signature verification. Xie et al. [13] converted raw time series feature data into images, added a channel weight learning mechanism, and proposed a Triplet Supervised Network (TSN) containing three weight-sharing convolutional neural networks for measuring the distance between signatures. Vorugunti et al. [14] fused high-level features extracted by Convolutional Autoencoders with manually extracted features to form a hybrid feature set, using a Depthwise Separable Convolutional Neural Network (DWSCNN) for verification. Compared with traditional CNNs, DWSCNN uses fewer training samples and parameters to effectively learn deep representations of signatures, thus forming a lightweight OSV framework.

The attention mechanism is one of the important concepts in the field of deep learning and has shown strong potential in recent years. Hu et al. [15] proposed Squeeze-and-Excitation Networks (SENet), which can implicitly and adaptively predict potential key features and model the interdependence among feature channels. It automatically learns the importance of each channel and then enhances useful features and suppresses features that are not useful for the current task according to this importance. Later, Woo et al. [16] proposed the Convolutional Block Attention Module (CBAM), which combines channel attention and spatial attention and is widely used to improve the representational ability of CNNs. Wang et al. [17] proposed attention residual learning to train very deep residual attention networks, which can be easily extended to hundreds of layers. Cheng et al. [18] proposed a Class-specific Attention Encoding (CAE) module to force CNNs to explicitly encode class attention. The CAE module can be embedded into CNNs to improve their recognition abilities. In the field of signature verification, the attention mechanism is also widely applied [12,19–21], effectively improving verification accuracy.

In summary, previous work has verified the feasibility and potential of deep learning models for online handwritten signature verification. However, this method still faces some problems: how to ensure the model's generalization ability with few samples, and how to extract more robust features and design classifiers with stronger classification performance to improve the model's verification accuracy.

2. Method

The handwritten signature verification problem is a typical small-sample classification problem, requiring the model to focus more on the distinctive features of signatures. Inspired by image classification problems and face recognition issues in the field of computer

vision, we use sequence-to-image conversion technology to transform the original problem into a similarity measure learning problem in the field of computer vision. We propose a representation learning method based on a Siamese network that integrates multi-scale features of signatures and finally employs a Temporal Convolutional Network (TCN) [22] to construct the classifier, which further improves the accuracy of signature authentication.

2.1. Problem description

The problem of writer-independent online handwritten signature verification can be formally described as: given a set of genuine signatures $T_t = \{S^0, S^1, ..., S^m\}$and a set of skilled forged signatures $T_f = \{f^0, f^1, ..., f^m\}$, where m represents the number of users, S^i represents the set of genuine signatures of the i-th user, and f^i represents the set of skilled forged signatures of the i-th user. For ease of explanation, we refer to the genuine signature set as the template set and the signatures to be verified as the test set. Using the above sets, we train a classifier C (template, test) that needs to output whether the input template data and test data belong to the same category. This problem is equivalent to the true-false discrimination of signature data. Furthermore, we hope that C has the ability to generalize beyond the template set and the test set.

The Siamese network [23] is one of the classic methods for solving the above type of problem. Its basic idea is to let the Siamese network extract features from the input template signatures and test signatures and learn their similarity, ultimately using this similarity as the basis for discrimination. We base our network design on the Siamese architecture.

Depending on the architecture of the model, it can be divided into end-to-end models or staged models. The network architecture proposed in this paper is a staged model because the end-to-end discrimination method only includes classification loss. Online handwritten signature verification datasets are scarce, and the model has a high risk of overfitting, lacking the ability to generalize beyond the template set. In addition, the similarity measure problem and classification problem are actually different optimization targets. Using only classification loss cannot enable the model to learn deeper representational information between samples, resulting in poor model generalization and difficulty in application.

2.2. Sequence-to-image conversion

The image features of handwritten signatures play an extremely important role in the verification task. Online handwritten signature verification performs discrimination by finding potential patterns in the sequence. This method ignores the image features of handwritten signatures, so we consider converting the original input sequence into an image, then using two-dimensional Convolutional Neural Networks (CNN) to extract features from the image, which not only retains the original temporal information of the signature, but also adds image information, making the signature features richer.

In previous work, typical methods for converting time-series data into image data include fast Fourier transform (FFT) [24], wavelet transform [25], etc., all of which are based on conversion methods from time domain to frequency domain. Considering our application context, we ultimately used the Relative Positional Matrix of time series [26] for image conversion. This method has several advantages:

(1) Simple calculation.
(2) The values of the matrix are calculated based on relative positions, eliminating the absoluteness of position.

The definition of Relative Position Matrix (RPM) is shown in Eq. (1).

$$M = \begin{pmatrix} x_1 - x_1 & x_2 - x_1 & \cdots & x_m - x_1 \\ x_1 - x_2 & x_2 - x_2 & \cdots & x_m - x_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1 - x_m & x_2 - x_m & \cdots & x_m - x_m \end{pmatrix} \tag{1}$$

It is known from the definition of M that the element of the i-th row and j-th column of M represents the difference in data between the i-th sampling point and the j-th sampling point of the input sequence. Via the abovementioned calculations, matrix M not only contains information from the original sequence but also has redundant features. The redundancy can act as data augmentation, which would improve the model's performance by compensating for the lack of original signature sequence features.

Afterward, M is processed with max-min normalization and multiplied by 255 to convert M into a grayscale value matrix, as shown in Eq. (2), to obtain the final matrix M.

$$M' = \frac{M - \min(M)}{\max(M) - \min(M)} * 255 \tag{2}$$

We only retained the three most significant channel features in the signature sequence: namely the horizontal coordinate (X), vertical coordinate (Y), and pressure (P). By calculating the corresponding RPM M'_x, M'_y, M'_p for each channel of the original sequence and stacking them, the target 3-channel image can be obtained. Fig. 1a-c represents the results of the RPM conversion for some samples (given in RGB format, where M'_x, M'_y, and M'_p correspond to the R, G, and B channels, respectively).

2.3. Downsampling

If the length of the original sequence is L and the number of channels is C, then the size of the converted relative positional matrix is L * L * C. The scale of the matrix data is quadratic to the length of the original sequence, which poses certain challenges for the storage and computation of the converted images. In addition, considering the redundancy of data in RPM, it is necessary to downsample the original time series while preserving the sample's own features as much as possible.

We employed the Largest Triangle Three Buckets (LTTB) algorithm [27], which divides the original sequence into k segments according to time order, considering each data set within a segment as a bucket. After division, starting from the first point of the sequence, thereafter, find a representative point with the smallest Standard Error of the Estimate (SEE) in each bucket according to the order of the buckets, excluding the first and last points, until the last point of the time series is selected. The pseudocode of the algorithm is as follows:

The pseudocode for the LTTB

Input:original signature sequence: $S = (X, Y, P)$, $X = (x_1, x_2, \ldots, x_n)$, $Y = (y_1, y_2, \ldots, y_n)$, $P = (p_1, p_2, \ldots, p_n)$
Output: Downsampled signature sequence: $S' = (X', Y', P')$, $X' = (x'_1, x'_2, \ldots, x'_m)$, $Y' = (y'_1, y'_2, \ldots, y'_m)$, $P' = (p'_1, p'_2, \ldots, p'_m)$
Algorithm:

1.For $i = 1,2,3$ do:
2. Divide the signature sequence into m buckets, with the first bucket containing only the first data point, the last bucket containing only the last data point, and the remaining data points evenly distributed into m-2 buckets.
3. Select the point in the first bucket.
4. for *a* in (Data points, except for the first and last bucket) do:
5. Calculate the area of the triangle formed by a with the selected point in the previous bucket and the average point in the next bucket, sorting each point in the bucket.
6. end for
7. Select the highest ranked point in the bucket.
8.end for
9.Select the point in the last bucket.

We use the LTTB algorithm to downsample the signature sequence. The length is reduced from the original L to m. Let $DR = L/m$, representing the down sampling ratio. Then use the sequence-to-image conversion, and the relative positional matrix with different degrees of data redundancy can be obtained. By this method, the size of the matrix can be reduced by DR times, greatly reducing storage and computational costs. Fig. 2a represents the original signature data, and Fig. 2b represents the signature data after downsampling by a factor of 4.57.

The downsampled data retains the basic structure of the original signature data while also amplifying some of the non-smooth turning points in the original signature. We believe this is more conducive for convolutional neural networks to extract more discriminative texture information. Compared with downsampling methods based on smooth sampling, the data after LTTB downsampling has stronger separability.

2.4. Encoder-classifier architecture

For our proposed model, the discrimination process is divided into encoding and discrimination stages. The learning target of the encoder is to encode the original features into a new feature space, so that the distance between samples of the same category in the new space is as small as possible, and the distance between samples of different categories in that space is as large as possible. Therefore, the Contrastive Loss function is used during the encoding stage.

The contrastive loss function as shown in Eq. (3).

$$L(W,(Y,X_1,X_2)) = \frac{1}{N}\sum_{n+1}^{N} YD_w^2 + (1-Y)max\,(m - D_w, 0)^2 \tag{3}$$

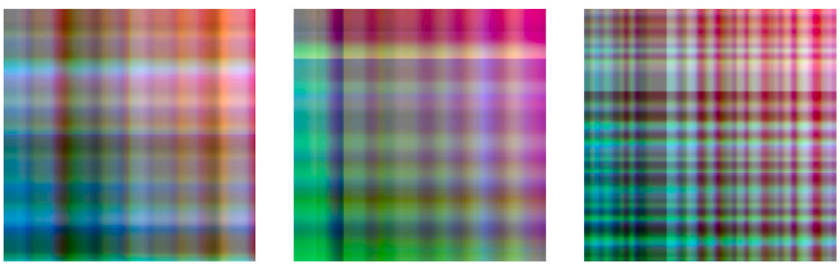


Fig. 1. RPM matrix visualization. (a–c) are the visualization results of three randomly selected samples.

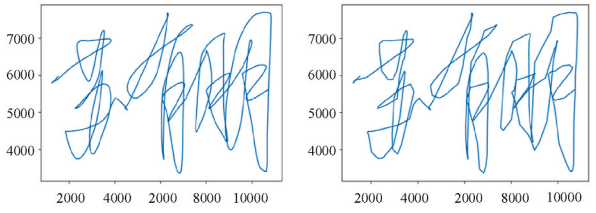


Fig. 2. Downsampling signature comparison chart. (a) is the original signature. (b) represents the signature after downsampling.

$$D_w(X_1, X_2) = \|X_1 - X_2\|_2 = \left(\sum_{i=1}^{P} \left(X_1^i - X_2^i\right)^2\right)^{\frac{1}{2}} \tag{4}$$

in the formula, W represents the weights of the encoder, X_1 and X_2 represent the two input samples to be compared, N represents the number of samples in a particular mini-batch, and Y represents the true label, that is, whether the two input samples belong to the same category. D_w is a distance metric, and we choose to measure the Euclidean distance between samples, as calculated in Eq. (4). m is a set threshold. When the distance between non-similar samples in the new feature space is greater than m, stop optimizing the distance between the input sample pairs, which reduces the difficulty of optimizing this loss.

The discrimination stage uses cross-entropy loss for optimization, aiming to discriminate whether the two encoded sequences belong to the same user.

The learning architecture based on the Encoder-Classifier architecture is shown in Fig. 3. We propose the Two-Dimensional Multi-Scale Feature Fusion Neural Network (2D-MFFnet) as an encoder. Mean-while, TCN is used as a classifier to verify the authenticity of signatures. Each module will be introduced in detail later.

2.5. Two-dimensional multi-scale feature fusion neural network

Our proposed Two-Dimensional Multi-Scale Feature Fusion Neural Network (2D-MFFnet) mainly consists of two types of convolutional blocks, namely the Basic Block and the Dilation Block. In addition, it includes a Squeeze-and-Excitation (SE) module, which is introduced below.

2.5.1. Basic block

The Basic Block contained in our proposed encoder consists of four groups of convolutional units with different sizes, corresponding to different receptive fields, capable of capturing neighborhood features of different scales in the feature map. Each convolutional unit also includes a convolutional layer with a kernel size of 1 to unify the output channel number of the four convolutional units. In each unit, the convolutional layer is followed by a ReLU function max(0, x) as the activation function. Table 1 represents the corresponding parameters for different groups:

After the four convolutional units have processed the input feature maps, the four sets of new features obtained will be stacked. Then, the Channel Attention Module SE Block is used to process them, which can better focus on the importance of features at different scales.

2.5.2. Dilation block

To further improve the model's ability to capture long-term relationships in sequence features, we use the Dilation Block to process features. In traditional time-series analysis tasks, the short-term dependence and long-term relationships of time series are of varying importance in different tasks. Inspired by the feature pyramid modules in computer vision, we designed a multi-scale sequence relationship capturing module. This module uses multiple groups of two-dimensional convolutions with varying dilation rates to

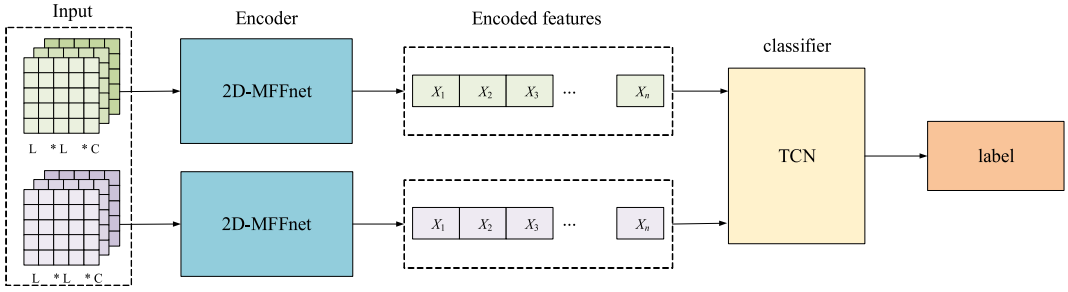


Fig. 3. Encoder-classifier architecture.

Table 1
Basic block parameters.

Group	Kernel size	Padding	Activation function
Group-1	3	1	ReLU
Group-2	5	2	
Group-3	7	3	
Group-4	11	5	

capture different temporal patterns from the input relative positional matrix.

It is known from the definition of the relative positional matrix that the i-th row and j-th column of the matrix represent the difference between the i-th sampling point and the j-th sampling point, that is, the i-th row represents the relative relationship of its data with any sampling point. If a convolution with a dilation rate of 0 and kernel size S, with weights W, is used to compute the convolution at the i-th row and j-th column of the RPM. According to the definition of the convolution operation, the value at that point should be $\sum_{i-\left\lfloor \frac{S}{2} \right\rfloor}^{i+\left\lfloor \frac{S}{2} \right\rfloor} \sum_{j-\left\lfloor \frac{S}{2} \right\rfloor}^{j+\left\lfloor \frac{S}{2} \right\rfloor} W_{i,j} * M_{i,j}$, which means that the convolution operation captures the relationship in the size S neighborhood of each sampling point in the original time series from the $i - \left\lfloor \frac{S}{2} \right\rfloor$ sampling point to the $i + \left\lfloor \frac{S}{2} \right\rfloor$ sampling point. If the dilation rate is increased at this point, the temporal span of the features captured can be further increased without adding model parameters.

For a convolution with a stride of 1, a dilation rate of V, and a convolution kernel size of S, using a padding rate $P = V$ can ensure that the size of the feature map after convolution is consistent with the input size. This is to ensure that the feature maps generated by convolution kernels with different dilation rates can be directly stacked.

Finally, via stacking the local features obtained from convolution blocks with different dilation rates, and using a 1*1 convolution kernel to merge features with different temporal spans, the fused global features can be obtained. Fig. 4 shows the sampling process of dilated convolution with a dilation rate of 2.

Similarly, the Dilation Block also contains four different convolutional units, with related parameters of each convolutional kernel as shown in Table 2.

2.5.3. Squeeze-and-excitation

The SE channel attention mechanism [15] can adaptively increase the weights of important channels, which effectively captures the significant features in the data and enhances the representation capacity of the model. It consists of two parts: squeeze and excitation. In the squeeze stage, global average pooling is used to obtain global information from the three channels. In the excitation stage, fully connected layers and activation functions are used to learn the weights of the channels, thereby obtaining the importance weights of each channel. Finally, the learned weights are multiplied by the corresponding original channels to obtain the weighted feature map.

The Dilation Block mentioned above uses dilation convolutions of various scales to capture different local dependencies of signature features, but it is not clear at this point which scale features are more effective for the task. Therefore, the SE channel attention mechanism is used to adaptively learn the importance of channels corresponding to each scale, which further improves the encoding effect of the encoder. The Dilation Block with the SE module is shown in Fig. 5.

2.5.4. Overall network architecture

The key structures of the designed network have been explained above. Below, we introduce the overall architecture of the 2D-

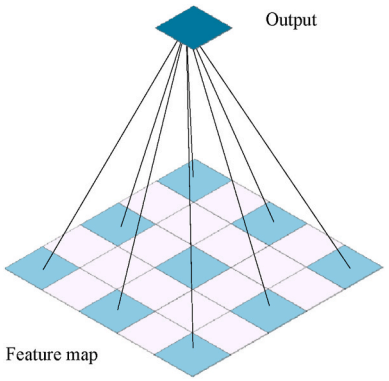


Fig. 4. Dilation convolution sampling.

Table 2
Dilation block parameters.

Group	Kernal size	Dilation rate	Padding	Activation function
Group-1	3	1	1	ReLU
Group-2		2	2	
Group-3		3	3	
Group-4		4	4	

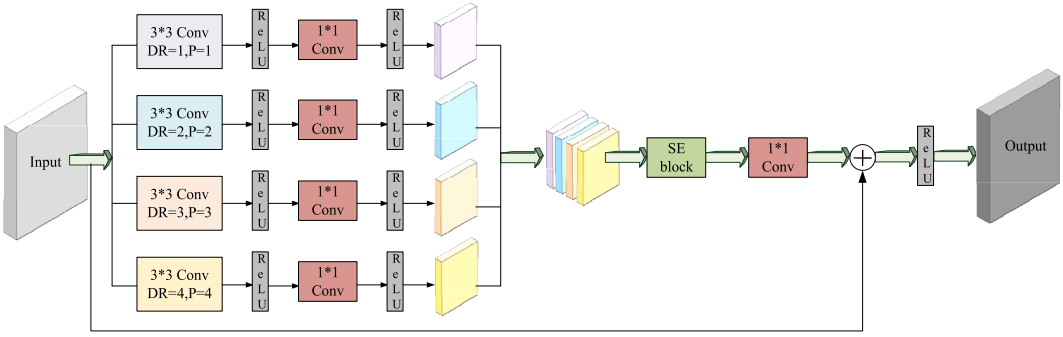


Fig. 5. Dilation block architecture with SE modules.

RMFFnet, as shown in Fig. 6:

The network receives input images converted from sequences to images. After feature extraction, it is mapped to a feature vector in a new feature space through a fully connected layer. In addition, both the Basic Block and the Dilation Block in the network are followed by residual structures to prevent gradient vanishing and further enhance feature extraction capability.

2.6. TCN classifier

Benefiting from the translational invariance of CNN, the features encoded still retain the temporal sequence relationship of the original time series. Therefore, time-series-related models can be used as the final classifier in the classification phase. TCN is a deep learning model for processing time series data and is more suitable for time series prediction and classification tasks [22]. TCN is based on the idea of Convolutional Neural Networks (CNN), but compared with ordinary convolution, it can capture long-term dependencies in time series data. By stacking multiple convolutional layers, each layer has a receptive field that can capture longer time series information. In addition, TCN also uses dilated convolutions technology, which further enhances the model's ability to model long-term dependencies. The TCN architecture used in the experiment is shown in Fig. 7. We used a 5-layer TCN block with the kernel size of 40. Relevant parameters are shown in Table 3.

2.7. MLP classifier

In classification tasks, the Multilayer Perceptron (MLP) [28] is often used as a classifier after the feature extraction phase. It is simple to implement and easy to converge. The parameters of the MLP are shown in Table 4. We used a three-layer MLP, with a hidden layer size of 256 and activated using the ReLU function.

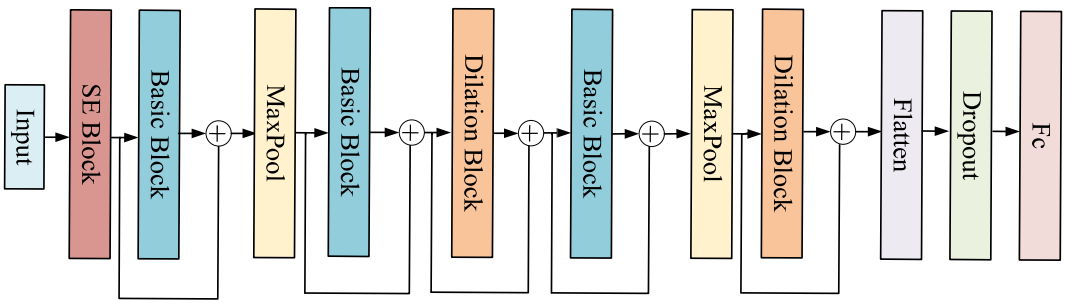


Fig. 6. Encoder overall architecture.

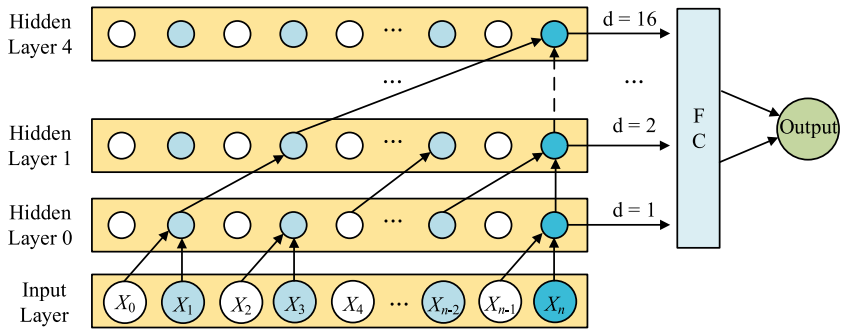


Fig. 7. TCN architecture.

Table 3
TCN parameters.

Parameters	Value
Kernal size	40
Layers	5
Dilation rate	[1,2,4,8,16]
Number of channels	25
Activation function	ReLU
Drop out	0.5

Table 4
MLP experiment parameters.

Parameters	Value
Number of hidden layers	2
Hidden layer size	256
Activation function	ReLU
Dropout	0.5
Python	Python 3.8

3. Experiment

In this section, we will further illustrate the effectiveness of the proposed method based on the datasets used, the experimental platform, the specific experimental parameters, and the comparative experiments. The specific configuration of our experimental platform is shown in Table 5:

3.1. Dataset and signature pairing

MYCT-100 [29] is a Spanish database published by the BiDA laboratory of the Autonomous University of Madrid. The database has signatures from 100 users, each with 25 genuine signatures and 25 skilled forged signatures. The signature collection device is a WACOM pen tablet, and the sampling frequency is 100 Hz. The following information was mainly collected during the signing process: coordinate trajectory, time, pressure value, and knob deflection angle. The ranges of the x-axis and y-axis coordinate values are 0–12700 and 0–9700, respectively. The pressure value ranges from 0 to 1024. The range of horizontal and vertical angles is 0–3600 and 300–900, respectively.

The SVC-2004 task2 [30] Chinese and English handwritten signature database was provided by the Hong Kong University of

Table 5
Experimental platform configuration.

Configuration	Version
CPU	Intel Xeon 64C
GPU	RTX4090 24G
RAM	90 GB DDR4
OS	Ubuntu 20.4
Python	Python 3.8
Pytorch	2.0.0

Science and Technology at the first World Signature Verification Competition held in 2004, with signatures collected using WACOM tablet computers. The database contains a relatively small number of signatures, with 40 users in total. Each user has 20 genuine signature samples and 20 skilled forgery signature samples. The signature feature information mainly includes seven time series features: coordinate trajectory, pressure, time, pen horizontal and vertical declination angles, and pen lifting and lowering marks.

In the MCYT-100 dataset, the signatures of the first 80 users were used to train the model, and the signatures of the remaining 20 users were used for testing. In the SVC-2004 task2 dataset, the signatures of the first 30 users were used as the training set, and the signatures of the remaining 10 users were used as the test set.

To facilitate the network's learning of signature similarity, we need to construct signature sample pairs. Taking the MCYT-100 dataset as an example, the training set has signatures from 80 users, making each user's genuine signatures paired with genuine signatures, resulting in $80 * C_{25}^{2} = 24000$ genuine-genuine sample pairs; each user's genuine signatures paired with forged signatures result in $80 * 25 * 25 = 50000$ genuine-forged signature sample pairs. To ensure data fairness, this experiment randomly discarded excess genuine-forged signature pairs to balance the number of positive and negative samples. The paired quantities afterward are shown in Table 6:

3.2. Preprocessing

In the experiments, we found that the original multi-channel sequence features contained a lot of redundancy, so we only retained the three most significant channel features in the signature sequence, namely the horizontal coordinate, vertical coordinate, and pressure. To eliminate noise in the time series, five point cubic smoothing filter [31] was used to smooth the online handwritten signature data. For the signature sequence feature $S = \{S_1, S_2, \ldots, S_n\}$, where $S_i = \{X, Y, P\}$ and $i = 1, 2, \ldots, n$, with n representing the number of sampling points in the signature, X representing the horizontal coordinate of the signature, Y representing the vertical coordinate, and P representing the pressure during writing. The smoothing method is as presented in Eqs. (5)–(9).

$$S'(1) = (69 \cdot S(1) + 4 \cdot S(2) - 6 \cdot S(3) + 4 \cdot S(4) - S(5))/70 \tag{5}$$

$$S'(2) = (2 \cdot S(1) + 27 \cdot S(2) + 12 \cdot S(3) - 8 \cdot S(4) + 2 \cdot S(5))/35 \tag{6}$$

$$S'(i) = (-3 \cdot S(i-2) + 12 \cdot S(i-1) + 17 \cdot S(i) + 12 \cdot S(i+1) - 3 \cdot S(i+2))/35 \tag{7}$$

$$S'(i-1) = (2 \cdot S(i-4) + 8 \cdot S(i-3) + 10 \cdot S(i-2) + 27 \cdot S(i-1) + 2 \cdot S(i))/35 \tag{8}$$

$$S'(i-2) = (-S(i-4) + 4 \cdot S(i-3) - 6 \cdot S(i-2) + 4 \cdot S(i-1) + 69 \cdot S(i))/70 \tag{9}$$

S_i' represents the signature sequence after smoothing. Subsequently, through normalization operation, as shown in Eq. (10). It is scaled to the range [0, 1] to eliminate the differences in dimensions between features.

$$S_i'' = \frac{S_i' - S_{min}'}{S_{max}' - S_{min}'} \tag{10}$$

Where S_{min}' is the minimum value in the feature sequence and S_{max}' is the maximum value in the feature sequence. For subsequent processing, it is necessary to ensure the equal length of the signature sequences. According to statistics, the length of most samples is less than 1400, so the signature sequence length is unified to 1400. For samples shorter than 1400, padding with zeros was used, and samples longer than 1400 were truncated.

3.3. Evaluation metrics

To evaluate the performance of the model, we utilize Accuracy (ACC) (Eq. (11)), False Accept Rate (FAR) (Eq. (12)), False Reject Rate (FRR) (Eq. (13)), and Equal Error Rate (EER) (Eq. (14)) to assess the proposed model.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$FAR = \frac{FP}{TN + FP} \tag{12}$$

Table 6
Number of pairs.

Dataset	Training set		Testing set	
	G-G	G-F	G-G	G-F
MCYT-100	24000	24000	6000	6000
SVC-2004 task2	5700	5700	1900	1900

G represents genuine signatures; F represents forged signatures.

$$FRR = \frac{FN}{TP + FN} \tag{13}$$

$$EER = FAR_T = FRR_T \tag{14}$$

where TP represents the number of genuine signatures correctly recognized by the model, TN represents the number of forged signatures correctly recognized, FP represents the number of genuine signatures misidentified as forged, FN represents the number of forged signatures misidentified as genuine, and T represents the threshold when the FAR equals the FRR.

3.4. Experimental parameters

We trained the 2D-MFFnet with the relevant parameters as shown in Table 7:

3.5. Experimental results

3.5.1. Comparison with typical convolutional networks

To intuitively compare the effectiveness of our proposed model, we selected the classic convolutional neural networks VGG-11 and ResNet-18 as encoders. At the same time, we implemented two classifiers: MLP and TCN. Subsequently, we combined the encoder and classifier for training. The results of the experiments are shown in Tables 8 and 9. On the MCYT-100 dataset, the 2D-MFFnet + TCN method achieved an accuracy of 93.74 %, an EER of 6.45 %, a FAR of 6.48 %, and an FRR of 6.03 %. On the SVC-2004 task2 dataset, the accuracy was 89.55 %, the EER was 10.57 %, the FAR was 11.21 %, and the FRR was 9.68 %. The results demonstrate that the 2D-MFFnet + TCN proposed in this paper outperforms other convolutional neural networks in terms of discrimination accuracy (ACC) and equal error rate (EER) on all datasets, which confirm the effectiveness of the proposed model. In addition, the TCN classifier achieved certain advantages in terms of accuracy and equal error rate.

3.5.2. Comparison with time series model

We employed popular time series models as encoders, including the Transformer, LSTM, and our proposed 1D-MFFnet, to encode the original time series data that had not undergone sequence-to-image transformation. Subsequently, we used MLP and a TCN as classifiers to discriminate the authenticity of the samples and tested their performances. The structure of 1D-MFFnet is similar to 2D-MFFnet, with all 2D convolutions in 2D-MFFnet replaced with 1D convolutions. The results are shown in Tables 10 and 11. In the MCYT-100 dataset, 1D-MFFnet + TCN achieved an EER of 7.76 %, which is the best result compared with other time series models. On the SVC-2004 task2 dataset, 1D-MFFnet + TCN achieved an EER of 14.68 %, which is the best result. It is evident that our proposed 1D-MFFnet still achieves competitive results compared with Transformer and LSTM.

From the results in Tables 10 and 11, it can be observed that LSTM + TCN achieved an accuracy of 90.2 % when dealing with the MCYT-100 dataset. However, it only achieved an accuracy of 79.97 % on the SVC2004 task2 dataset, which is unsatisfactory. This is because LSTM is more suitable for larger data scales. SVC-2004 task2 contains fewer signature data, and using LSTM poses a serious overfitting problem, which lead to poor results. However, the proposed model still has good verification performance when dealing with small databases, further confirming the effectiveness of the proposed model.

The experiments in this section, compared with the convolutional methods in the previous section (Tables 8 and 9), represent that the methods that underwent sequence-to-image transformation overall outperform the traditional time series methods in all metrics, which indicates that the image representation method effectively improves the signature verification accuracy.

3.5.3. Ablation experiments

To verify the effectiveness of the encoder-classifier architecture, we tested the various indicators of the end-to-end 2D-MFFnet model, as shown in Table 12. The end-to-end model of 2D-MFFnet had an EER of 10.07 % and an accuracy of 90.00 % on the MCYT-100 dataset, and an EER of 15.53 % and an accuracy of 84.55 % on the SVC-2004 task2 dataset. Compared with the end-to-end method of 2D-RMFFnet, the 2D-MFFnet + TCN method reduced the equal error rates by 3.48 % and 4.11 % on the two datasets, respectively. And the accuracy rates were increased by 3.05 % and 5.00 %, respectively. The results represent that the staged training can better learn signature feature representation and improve discrimination accuracy.

Considering that the importance of input channel features and channel features at different scales varies, SENet can enhance the weight of important feature information among different channels. There-fore, the SE structure is introduced into the 2D-MFFnet to

Table 7
2D-MFFnet parameters.

Parameters	Value
Optimizer	Adam
Learning rate (lr)	0.001
Batch size	32
Weight decay	10^{-9}
Dropout	0.5

Table 8
Comparison results with other convolutional networks on the MCYT-100 dataset.

Encoder	Classifier	ACC (%)	FAR (%)	FFR (%)	EER (%)
ResNet-18	MLP	88.95	8.01	14.38	9.92
	TCN	89.83	13.11	7.22	9.92
VGG-11	MLP	90.30	7.35	11.95	8.77
	TCN	91.27	11.71	5.74	8.61
2D-MFFnet	MLP	92.87	6.96	7.28	7.16
	TCN	**93.74**	6.48	6.03	**6.45**

Table 9
Comparison results with other convolutional networks on the SVC-2004 task2 dataset.

Encoder	Classifier	ACC (%)	FAR (%)	FFR (%)	EER (%)
ResNet-18	MLP	86.47	15.00	12.05	13.47
	TCN	87.15	20.58	5.11	11.26
VGG-11	MLP	88.86	14.47	7.78	11.21
	TCN	87.92	21.47	3.26	12.84
2D-MFFnet	MLP	87.47	18.78	6.26	10.68
	TCN	**89.55**	11.21	9.68	**10.57**

Table 10
Comparison with time series model on MCYT-100 dataset.

Encoder	Classifier	ACC (%)	FAR (%)	FFR (%)	EER (%)
Transformer	MLP	89.78	9.32	11.12	9.84
	TCN	90.00	11.64	8.29	10.19
LSTM	MLP	89.84	8.96	11.31	10.35
	TCN	90.2	14.76	4.92	9.11
1D-MFFnet	MLP	**91.98**	8.00	8.04	8.04
	TCN	90.16	17.53	2.16	**7.76**

Table 11
Comparison with time series model on SVC-2004 task2 dataset.

Encoder	Classifier	ACC (%)	FAR (%)	FFR (%)	EER (%)
Transformer	MLP	83.84	27.94	4.37	14.21
	TCN	84.34	26.42	4.89	14.89
LSTM	MLP	80.84	25.63	12.68	19.05
	TCN	79.97	27.68	12.37	19.05
1D-MFFnet	MLP	**85.37**	25.00	4.26	14.89
	TCN	84.61	27.89	2.89	**14.68**

Table 12
End-to-end experimental results.

Method	Dataset	ACC (%)	FAR (%)	FFR (%)	EER (%)
2D-MFFnet	MCYT-100	90.00	13.79	6.29	10.07
	SVC-2004 task2	84.55	21.05	9.84	15.53

further improve the model's representation learning ability. Table 13 shows the experimental results of 2D-MFFnet without the SE structure. The classifiers used were TCN and MLP. The results show that the performance of 2D-MFFnet with the SE structure is significantly improved on the MCYT-100 dataset, with the EER reduced by approximately 5 %. On the SVC-2004 task2 dataset, the EER decreased by about 2 %. This validates the effectiveness of the SE module.

3.5.4. Robustness evaluation of the proposed model

Considering that different signature samples may impact model performance, we used five-fold cross-validation to evaluate the robustness of the 2D-MFFNet + TCN model. For the MCYT dataset, the data was divided into groups of 20 users each (total of 5 groups). For the SVC dataset, which includes 40 users, we split the dataset into 5 groups based on users (8 users per group). During the experiments, each group of users was used as the test set in turn, with the remaining groups serving as the training set. The experimental results are shown in Fig. 8a-d. Differences in validation results can be observed, which are due to some users having challenging

Table 13
Experimental results of 2D-MFFnet without SENet.

Method	Dataset	ACC (%)	FAR (%)	FFR (%)	EER (%)
2D-MFFnet* + TCN	MCYT-100	87.30	12.73	12.61	12.73
	SVC-2004 task2	84.05	15.58	16.32	16.16
2D-MFFnet* + MLP	MCYT-100	88.12	11.96	11.81	11.92
	SVC-2004 task2	88.31	11.32	12.11	12.00

'2D-MFFnet*' indicates 2D-MFFnet without SENet.

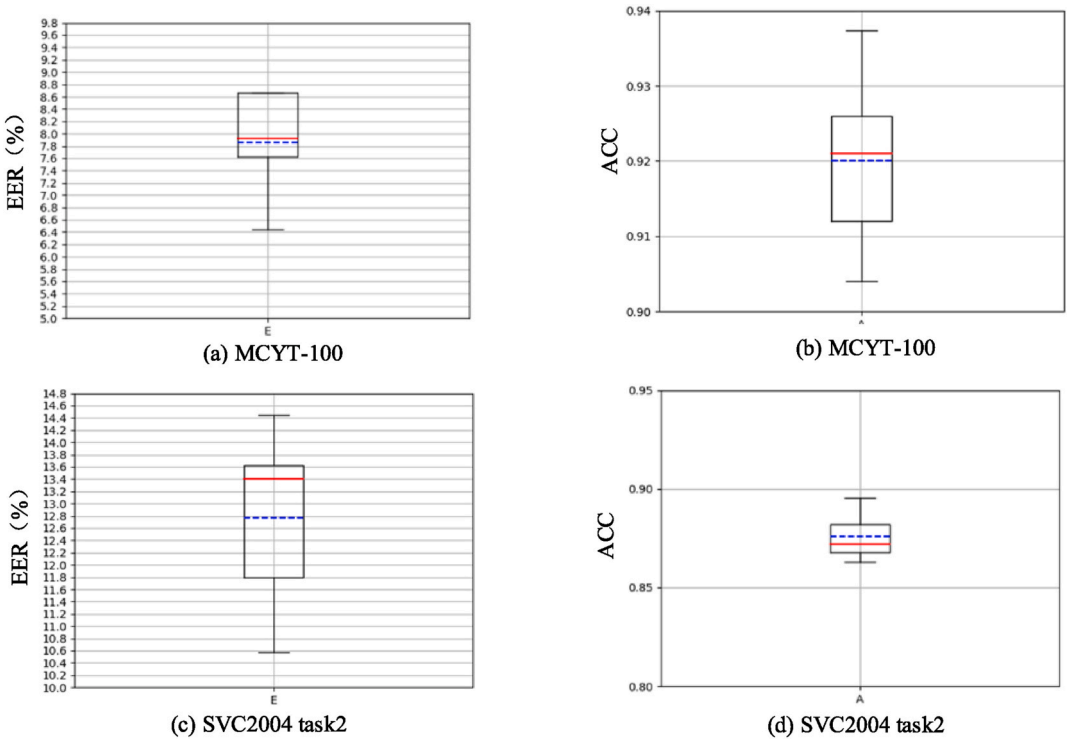


Fig. 8. The five-fold cross-validation results of 2D-FMMnet + TCN.

samples (low distinguishability between genuine and forged signatures). The significant differences among different user groups in different datasets cause some fluctuations in the model's performance. Overall, our proposed model still demonstrates robustness.

3.5.5. Comparison of performance with other methods

In this part, we compare with other representative methods. Tables 14 and 15 respectively present the comparison results of ACC and EER on the MCYT-100 and SVC-2004 task2 datasets. Documents [12,32–35] are deep learning methods, and documents [36,37] use traditional methods based on DTW and machine learning. Among them, the document [37] uses a 5V1 verification method, which involves five reference signatures. In contrast, our method used a 1V1 verification method (with one reference sample), making our verification task more challenging. The results demonstrate the superior performance of our proposed 2D-MFFnet + TCN method.

We then calculated the parameter quantity and computational complexity of the proposed model. For traditional methods such as

Table 14
Comparison with other methods on MCYT-100 dataset.

Method	ACC (%)	EER (%)
Siamese Neural Network [12]	93.53	6.57
Semantic-driven [32]	–	8.79
OSVNet [33]	92.85	–
A stroke-based RNN [35]	–	10.46
Stroke-Wise Distortion [36]	–	13.72
2D-MFFnet + TCN	**93.74**	**6.45**

'-' indicates that the paper does not present relevant results.

Table 15
Comparison with other methods on SVC-2004 task2 dataset.

Method	ACC (%)	EER (%)
Siamese Neural Network [12]	88.23	11.74
OSVNet [33]	68.21	–
Signature2Vec [34]	86.00	–
DTW + SVM(5V1) [37]	88.59	–
Stroke-Wise Distortion [36]	–	18.63
2D-MFFnet + TCN	**89.55**	**10.57**

'-' indicates that the paper does not present relevant results.

DTW [36], the time complexity is determined by the template signature length M and the length of the signature to be verified T. For deep learning models, we measure the computational complexity in terms of GFLOPs, as shown in Table 16. Since the proposed model processes image data, its computational complexity and parameter quantity are significantly higher than those of previous sequential models, which is in line with reality. Additionally, our proposed model requires approximately 1720 s for training and 30.3 s for testing, demonstrating high usability.

3.6. Dataset mixing

In the previous experiments, we have demonstrated the performance of the proposed model on a single dataset. However, whether the model can maintain its accuracy and robustness on a large dataset composed of multiple languages and different collection devices is also an interesting question. Therefore, we conducted more experiments following this line of thought.

We combined the MCYT-100 dataset (Spanish) and the MOBISIG dataset (Hungarian) [38] to create a new dataset, bringing the total number of users to 183 (with 100 from MCYT-100 and 83 from MOBISIG). Unlike other datasets, the MOBISIG database consists of signatures written with a finger on a tablet and received by an Android APP. We randomly discarded the excess genuine signatures from MOBISIG so that the processed dataset contains 20 genuine signatures and 20 forged signatures per user.

We mixed the MCYT-100 and MOBISIG datasets as follows: both datasets were divided into five parts by user (MCYT-100 with 20 users per part and MOBISIG with 16 users per part). During each training session, the training-testing split was applied in the same way for both datasets. For instance, in the first training session, the signatures of users 1–20 from MCYT-100 were used as the test set, and those of users 21–100 were used as the training set. Similarly, the signatures of users 1–16 from MOBISIG were used as the test set, and those of users 17–83 were used as the training set. The results after five-fold cross validation following the aforementioned method are shown in Fig. 9. The EER (Fig. 9a) is 13.6 $\pm$ 0.45, and the ACC (Fig. 9b) is 85.9 $\pm$ 1.

The MOBISIG dataset is more challenging to distinguish. As shown in Table 17, our proposed method achieved an EER of 14.7 $\pm$ 0.5 on this dataset, which is about 1 % lower than the EER of other methods. Overall, the model's performance showed a certain degree of decline when using the mixed dataset. Considering the differences in language and data distribution between MCYT-100 and MOBISIG, we believe that this level of decline is within an acceptable range, demonstrating that our proposed model is robust to signature language and signature devices. This also explains why recognizing mixed language datasets (such as the SVC dataset) is more challenging. Furthermore, we believe that one of the challenges of writer-independent online handwritten signature verification is overcoming data distribution inconsistencies to improve the overall performance of the model. This will be an important research direction in the future.

4. Conclusion and implications

We propose a writer-independent online handwritten signature verification network, where the verification process is divided into two parts: feature encoding and authenticity discrimination. The feature encoding stage focuses on learning a highly discriminative representation of the signature, where the Siamese network serves as the backbone, using dilated convolutions and residual network learning for multi-scale features of signatures. Channel attention mechanisms are introduced to help the network learn the importance between different channels and scales, thus improving the network's representation learning capability. Afterward, a temporal convolution classifier was constructed to verify the authenticity of the signature. Our focus is on the image representation of the

Table 16
Computational cost of each method.

Method	Computational complexity	Parameter quantity
Ours	76.52 GFLops	45.56M
Siamese Neural Network [12]	–	7.44M
TriAlexNet_CWL [13]	9.95 GFLops	47.49 M
Semantic-driven [32]	1.19 GFLops	–
Stroke-Wise Distortion [36]	O(MN)	–
Target-Wise Distortion [36]	O(MN)	–

"-" indicates that the paper does not provide relevant results or the metric is not applicable.

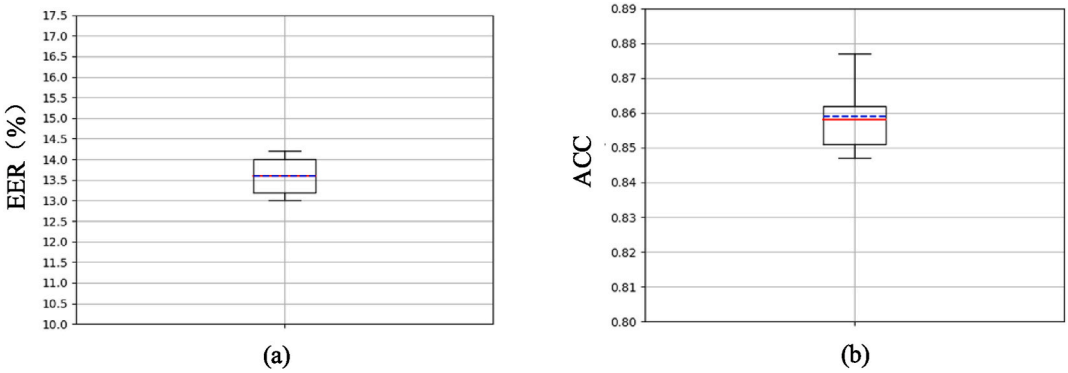


Fig. 9. Five-fold cross-validation results on the hybrid dataset.

Table 17
Comparison experiments on the MOBISIG dataset.

Method	Dataset	EER (%)
Ours	MOBISIG	14.70
	Mixed dataset	13.60
semantic-driven [32]	MOBISIG	15.37
A stroke-based RNN [35]	MOBISIG	16.08

signature. The down sampled signature sequence is converted into an image using the relative position matrix method to obtain richer features. In subsequent experiments, we compared several of the most popular time series models to demonstrate that image models that incorporate timing features have better verification capabilities.

In addition, the writer-independent method effectively alleviates the problem of insufficient signature samples in practical applications. A single template signature can be sufficient to determine the authenticity of a signature, which is more conducive to practical applications and deployment. However, compared with writer-dependent methods, writer-independent methods have lower accuracy, hence further research is needed to enhance the verification accuracy. To address the issue of inconsistent signature data distribution, the approach of transfer learning can be utilized. Other studies have already illustrated the effectiveness of this method, and we will also attempt to use this approach to further improve the validation accuracy in subsequent research.

Ethical statement

The manuscript is submitted for the first time, and all content is original and free from plagiarism.

Data availability statement

The MCYT-100 database is openly at: http://atvs.ii.uam.es/atvs/mcyt100s.html. The SVC-2004 Task2 database is openly at: https://cse.hkust.edu.hk/svc2004/download.html. The MOBISIG database is openly at: https://www.ms.sapientia.ro/~manyi/mobisig.html.

Funding

This work is supported by National Natural Science Foundation of China (62073227, Shuai Yuan), Liaoning Provincial Science and Technology Department Foundation (2023JH2/101300212, Shuai Yuan).

CRediT authorship contribution statement

Fangjun Luan: Supervision, Resources, Methodology, Data curation. **Weiyi Cao:** Writing – original draft, Methodology, Data curation, Conceptualization. **Shuai Yuan:** Writing – review & editing, Resources, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:Shuai Yuan reports financial support was provided by National Natural Science Foundation of China. Shuai Yuan reports financial support was provided by Liaoning Provincial Science and Technology Department. If there are other authors, they declare

that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R. Plamondon, S.N. Srihari, Online and off-line handwriting recognition: a comprehensive survey, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 63–84, https://doi.org/10.1109/34.824821.

[2] M. Sharif, M. Raza, J.H. Shah, M. Yasmin, S.L. Fernandes, An Overview of Biometrics Methods, Handbook of Multimedia Information Security: Techniques and Applications, 2019, pp. 15–35.

[3] D.R. Adithya, V. Anagha, M. Niharika, N. Srilakshmi, S.K. Aditya, Signature Analysis for Forgery Detection, Emerging Research in Computing, Information, Communication and Applications: ERCICA 2018, vol. 2, Springer, 2019, pp. 339–349.

[4] E.A. Soelistio, R.E.H. Kusumo, Z.V. Martan, E. Irwansyah, A review of signature recognition using machine learning. 2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI), IEEE, 2021, pp. 219–223.

[5] S. Bhavani, R. Bharathi, A multi-dimensional review on handwritten signature verification: strengths and gaps, Multimed. Tool. Appl. 83 (1) (2024) 2853–2894, https://doi.org/10.1007/s11042-023-15357-2.

[6] M.I. Khalil, M. Moustafa, H.M. Abbas, Enhanced DTW based on-line signature verification, in: 2009 16th IEEE International Conference on Image Processing (ICIP), IEEE, 2009, pp. 2713–2716.

[7] A. Parziale, M. Diaz, M.A. Ferrer, A. Marcelli, Sm-dtw: stability modulated dynamic time warping for signature verification, Pattern Recogn. Lett. 121 (2019) 113–122, https://doi.org/10.1016/j.patrec.2018.07.029.

[8] Y. Liu, Z. Yang, L. Yang, Online signature verification based on DCT and sparse representation, IEEE Trans. Cybern. 45 (11) (2014) 2498–2511, https://doi.org/10.1109/TCYB.2014.2375959.

[9] M. Okawa, Template matching using time-series averaging and DTW with dependent warping for online signature verification, IEEE Access 7 (2019) 81010–81019, https://doi.org/10.1109/ACCESS.2019.2923093.

[10] A. Hefny, M. Moustafa, Online signature verification using deep learning and feature representation using Legendre polynomial coefficients, in: THe International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019) 4, Springer, 2020, pp. 689–697.

[11] S. Lai, L. Jin, W. Yang, Online signature verification using recurrent neural network and length-normalized path signature descriptor, in: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), IEEE, 2017, pp. 400–405.

[12] Q. Shen, F. Luan, S. Yuan, Multi-scale residual based siamese neural network for writer-independent online signature verification, Appl. Intell. 52 (12) (2022) 14571–14589, https://doi.org/10.1007/s10489-022-03318-5.

[13] L. Xie, Z. Wu, X. Zhang, Y. Li, X. Wang, Writer-independent online signature verification based on 2D representation of time series data using triplet supervised network, Measurement 197 (2022) 111312, https://doi.org/10.1016/j.measurement.2022.111312.

[14] C.S. Vorugunti, V. Pulabaigari, R.K.S.S. Gorthi, P. Mukherjee, Osvfusenet: online signature verification by feature fusion and depth-wise separable convolution based deep learning, Neurocomputing 409 (2020) 157–172, https://doi.org/10.1016/j.neucom.2020.05.072.

[15] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.

[16] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, CBAM: convolutional block attention module, in: Computer Vision - ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII, Springer-Verlag, Munich, Germany, 2018, pp. 3–19, https://doi.org/10.1007/978-3-030-01234-2_1.

[17] F. Wang, M. Jiang, C. Qian, S. Yang, X. Tang, Residual attention network for image classification, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, https://doi.org/10.1109/ICCV48922.2021.00025.

[18] G. Cheng, P. Lai, D. Gao, J. Han, Class attention network for image recognition, Sci. China Inf. Sci. 66 (2023).

[19] K. Ahrabian, B. Babaali, Usage of Autoencoders and Siamese Networks for Online Handwritten Signature Verification, vol. 12, Springer, London, 2019.

[20] S. Chattopadhyay, S. Manna, S. Bhattacharya, U. Pal, SURDS: Self-Supervised Attention-Guided Reconstruction and Dual Triplet Loss for Writer Independent Offline Signature Verification, 2022.

[21] J.X. Ren, Y.J. Xiong, H. Zhan, B. Huang, 2C2S: a two-channel and two-stream transformer based framework for offline signature verification, Eng. Appl. Artif. Intell. 118 (2023) 105639.

[22] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, arXiv preprint arXiv:1803.01271 (2018), https://doi.org/10.48550/arXiv.1803.01271.

[23] D. Chicco, Siamese neural networks: an overview, Artificial neural networks, 73-94, https://doi.org/10.1007/978-1-0716-0826-5_3, 2021.

[24] U. Oberst, The fast Fourier transform, SIAM J. Control Optim. 46 (2) (2007) 496–540, https://doi.org/10.1137/060658242.

[25] D. Zhang, D. Zhang, Wavelet transform, Fundamentals of image data mining: analysis, Features, Classification and Retrieval (2019) 35–44.

[26] W. Chen, K. Shi, A deep learning framework for time series classification using Relative Position Matrix and Convolutional Neural Network, Neurocomputing 359 (2019) 384–394, https://doi.org/10.1016/j.neucom.2019.06.032.

[27] S. Steinarsson, Downsampling Time Series for Visual Representation, 2013.

[28] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323(6088) 1986 533-536..

[29] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.-J. Igarza, C. Vivaracho, MCYT baseline corpus: a bimodal biometric database, IEE Proceedings-Vision, Image and, Signal Process. 150 (6) (2003) 395–401.

[30] D.-Y. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi, T. Matsumoto, G. Rigoll, SVC2004: first international signature verification competition, in: Biometric Authentication: First International Conference, ICBA 2004, Hong Kong, China, July 15-17, 2004. Proceedings, Springer, 2004, pp. 16–22.

[31] P.A. Gorry, General least-squares smoothing and differentiation by the convolution (Savitzky-Golay) method, Anal. Chem. 62 (6) (1990) 570–573, https://doi.org/10.1021/ac00205a007.

[32] J. Long, C. Xie, Z. Gao, High discriminant features for writer-independent online signature verification, Multimed 82 (25) (2023) 38447–38465, https://doi.org/10.1007/s11042-023-14638-0.

[33] C.S. Vorugunti, P. Mukherjee, V. Pulabaigari, Osvnet: convolutional siamese network for writer independent online signature verification, in: 2019 International Conference on Document Analysis and Recognition (ICDAR), IEEE, 2019, pp. 1470–1475.

[34] M.K. Srivastava, D. Reddy, B. Kurma, K. Yeturu, Signature2Vec-An algorithm for reference frame agnostic vectorization of handwritten signatures, in: International Conference on Computer Vision and Image Processing, Springer, 2021, pp. 130–138, https://doi.org/10.1007/978-3-031-11346-8_50.

[35] C. Li, X. Zhang, F. Lin, Z. Wang, J.E. Liu, R. Zhang, H. Wang, A stroke-based RNN for writer-independent online signature verification, in: 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019, pp. 526–532.

[36] M. Diaz, A. Fischer, M.A. Ferrer, R. Plamondon, Dynamic signature verification system based on one real signature, IEEE Trans. Cybern. 48 (1) (2016) 228–239, https://doi.org/10.1109/TCYB.2016.2630419.

[37] K.-K. Tseng, X.-X. An, C. Chen, Online handwritten verification algorithms based on DTW and SVM, J. Internet Technol. 21 (6) (2020) 1725–1732, https://doi.org/10.3966/160792642020112106014.

[38] M. Antal, L.Z. Szabó, T. Tordai, Online signature verification on MOBISIG finger-drawn signature corpus, Mobile Inf. Syst. 2018 (2018) 1–15, https://doi.org/10.1155/2018/3127042.