Pattern Recognition 159 (2025) 111146

Contents lists available at ScienceDirect

Pattern Recognition

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

HTCSigNet: A Hybrid Transformer and Convolution Signature Network for offline signature verification

Lidong Zheng[1], Da Wu[1], Shengjie Xu, Yuchen Zheng *

College of Information Science and Technology, Shihezi University, Shihezi, 832003, China

ARTICLE INFO

Keywords:
Offline signature verification
Convolutional neural network
Transformer

ABSTRACT

For Offline Handwritten Signature Verification (OHSV) tasks, traditional Convolutional Neural Networks (CNNs) and transformers are hard to individually capture global and local features from signatures, and single-depth models often suffer from overfitting and poor generalization problems. To overcome those difficulties, in this paper, a novel Hybrid Transformer and Convolution Signature Network (HTCSigNet) is proposed to capture multi-scale features from signatures. Specifically, the HTCSigNet is an innovative framework that consists of two parts: transformer and CNN-based blocks which are used to respectively extract global and local features from signatures. The CNN-based block comprises a Space-to-depth Convolution (SPD-Conv) module which improves the feature learning capability by precisely focusing on signature strokes, a Spatial and Channel Reconstruction Convolution (SCConv) module which enhances model generalization by focusing on more distinctive micro-deformation features while reducing attention to common features, and convolution module that extracts the shape, morphology of specific strokes, and other local features from signatures. In the transformer-based block, there is a Vision Transformer (ViT) which is used to extract overall shape, layout, general direction, and other global features from signatures. After the feature learning stage, Writer-Dependent (WD) and Writer-Independent (WI) verification systems are constructed to evaluate the performance of the proposed HTCSigNet. Extensive experiments on four public signature datasets, GPDSsynthetic, CEDAR, UTSig, and BHSig260 (Bengali and Hindi) demonstrate that the proposed HTCSigNet learns discriminative representations between genuine and skilled forged signatures and achieves state-of-the-art or competitive performance compared with advanced verification systems. Furthermore, the proposed HTCSigNet is easy to transfer to different language datasets in OHSV tasks.[2]

1. Introduction

Handwritten signature as a method of personal identification, which is widely used in various authentication scenarios, such as commercial transactions, bank procedures, and legal documents, et al. [1]. However, due to the ease of forgery associated with handwritten signatures, this poses a significant challenge in verifying whether a signature is genuine or not. Therefore, designing a robust signature verification system can effectively assist the issues arising from the application of handwritten signatures.

In general, according to different application scenarios, handwritten signature verification systems can be divided into two types: online and offline. For online handwritten signature verification systems, signatures are collected as dynamic information such as positions of the pen, pressure, and stroke, et al. [2]. For Offline Handwritten Signature Verification (OHSV) systems, signatures are collected as static digital images which are more common in real-world scenarios. Since offline signatures lose the dynamic information, OHSV tasks are more challenging and complex compared to online signature verification tasks [3]. Whether online or offline, signature verification is usually defined as a binary classification problem that verifies a query signature as a genuine or forged signature. In addition, forged signatures can be categorized as,

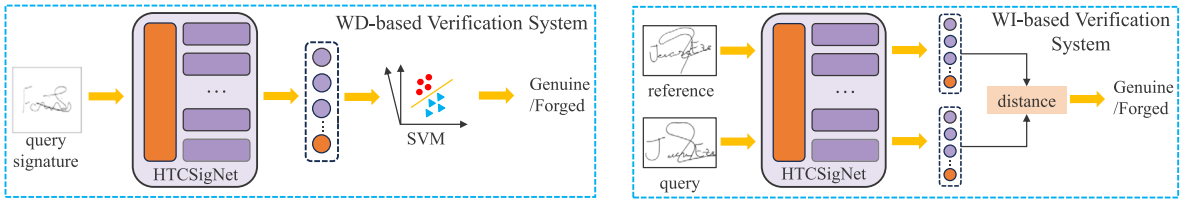
- **Random Forgery:** the genuine signature from other users, which is totally different from a target user.
- **Simple Forgery:** a signature sample written by the person who knows the shape of the original signature without much practice [4].

* Corresponding author.
E-mail addresses: zhenglidong@stu.shzu.edu.cn (L. Zheng), wuda1@stu.shzu.edu.cn (D. Wu), xushengjie@stu.shzu.edu.cn (S. Xu), zhengyuchen@shzu.edu.cn (Y. Zheng).

[1] Equal contribution.

[2] The code is available at https://github.com/copycpp/HTCSigNet-Master.

https://doi.org/10.1016/j.patcog.2024.111146
Received 18 April 2024; Received in revised form 23 September 2024; Accepted 27 October 2024
Available online 5 November 2024



(a) WD-based verification system (b) WI-based verification system

Fig. 1. Different verification system.

- **Skilled Forgery:** the forger has access to authentic signatures and forges after practicing many times.

Since skilled forgeries are deliberately imitated by trained forgers, they are often very similar to genuine signatures. Therefore, discriminating the difference between genuine and skilled forged signatures is often the core problem of OHSV systems.

The task of OHSV usually consists of a preprocessing stage, a feature learning stage, and a decision stage [5]. In the preprocessing stage, it performs necessary operations on signature images, including grayscale conversion, binarization, and normalization, et al. In the feature learning stage, the target is to learn robust representations from different signatures, especially help to discriminate the genuine and forged signatures. The last stage involves the decision-making process, where the previously trained feature extractor is utilized to make decisions on a query signature (determining whether the query signature is genuine or forged). In this paper, the proposed method takes place in the feature learning stage, where we propose a novel method to learn discriminative representations between genuine signatures and skilled forgeries.

To design a discriminated feature extractor for OHSV systems, one idea is to extract global and local features as potential information from signatures. Generally, the global feature involves analyzing the entire signature to capture the overall shape, size, tilt, smoothness of curves, and overall layout features. As for the local feature, it represents a form of localized detail, and in the context of signature images, it can be viewed as a kind of "micro deformation". Specifically, the micro deformations between the genuine signatures and skilled forgeries can be described as small translations, transformations, and special writing habits of different signers, et al. [6].

In past decades, deep learning-based feature learning methods gradually replaced handcraft-based feature learning methods in many real-world applications [7,8]. In particular, Local Binary Patterns (LBP) [9], Histogram of Oriented Gradient (HOG) [10] and Grey-Level Co-occurrence Matrix (GLCM) [11] can only extract partial structural or local features of signature images. However, with the advent of deep learning-based methods, particularly those using Convolutional Neural Networks (CNNs), have achieved state-of-the-art performance in extracting local features [3,5,6,12,13]. In addition, the latest research using transformer-based methods [14,15] have also demonstrated state-of-the-art results in extracting global features. Although some works in offline handwritten signature verification tasks have made certain progress in separately extracting global and local features, there is still little effort to explore multi-scale feature methods that simultaneously extract global by transformers and local features by CNNs.

Therefore, the objective of this study is to combine the strengths of CNN and transformer-based architectures and those advantages to learn the discriminative representation between genuine signatures and skilled forgeries. Here, a novel Hybrid Transform and Convolution Signature Network (HTCSigNet) is proposed to capture global and local multi-scale features for OHSV tasks. The hybrid framework idea does not simply stack the transformer and CNN but undergoes deep integration and interaction. This novel framework is designed as a parallel of two streams, which are the transform and the CNN-based streams. Detailedly, the transformer-based stream utilizes the self-attention mechanism to focus on global information such as the overall shape, contour features, layout, general direction, coherence, and other spatial information of the signature. The CNN-based stream utilizes the receptive field mechanism to focus on local information such as stroke details, connections, thickness, spacing, and shapes of signature strokes. In addition, we conduct a pre-convolution process before the transformer to capture multi-scale representations, aiming to achieve composite features including local information at a global level. Furthermore, we apply the SPD-Conv [16] and SCConv [17] modules into the CNN-based stream. The SPD-Conv module enhances the model feature learning ability to focus on signature strokes based on non-stride convolution modules, while the SCConv reconstructs spatial and channel modules to focus more on discriminative features between genuine signatures and skilled forgeries, while ignoring common features, thereby reducing feature redundancy and enhancing model generalization in the OHSV task. Their effectiveness is fully demonstrated in subsequent ablation experiments.

After the feature learning phase, the proposed HTCSigNet is evaluated separately in the verification systems based on Writer-Dependent (WD) is shown in Fig. 1(a) and Writer-Independent (WI) is shown in Fig. 1(b). Thorough experimentation and analysis have been conducted on GPDSsynthetic, CEDAR, UTSig, and BHSig260 datasets, the proposed HTCSigNet achieves state-of-the-art performance compared with other advanced verification systems. The main contributions of this study are as follows.

- A novel hybrid transformer and convolution framework called HTCSigNet is proposed to learn discriminative representations between genuine signatures and skilled forgeries. We demonstrate HTCSigNet has exceptional feature learning and generalization abilities and it is easy to transfer to other signature datasets of different languages.
- A novel CNN-based composite block is proposed for capturing distinctive and generalized features of different signatures. It comprises the SPD-Conv module to enhance the learning capability by focusing on signature strokes, the SCConv module to improve generalization by reducing attention to common features, and general convolution layers for aggregating the final local features.
- Comprehensive experiments with multiple evaluation metrics in WD and WI verification systems proved that the proposed HTCSigNet can learn discriminative representations between genuine signatures and skilled forgeries and achieve state-of-the-art performance on different OHSV tasks.

The rest of the paper is arranged as follows: Section 2 discusses the advantages and disadvantages of some existing feature extractors and advanced offline signature verification systems. Section 3 introduces the proposed HTCSigNet in detail and how to train and evaluate it. Section 4 presents the experiment and results. Section 5 concludes this paper and describes possible future works.

2. Related work

As described in Section 1, offline signature verification systems usually consist of three stages, and the feature learning stage plays a crucial

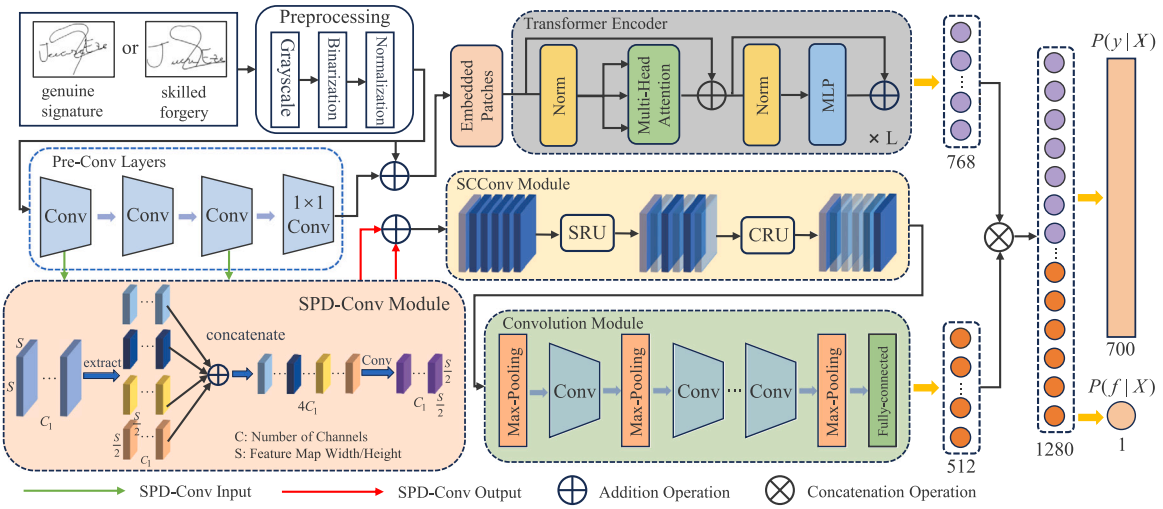


Fig. 2. The overview architecture of the proposed HTCSigNet. In the SCConv module, SRU represents Spatial Reconstructive Unit, and CRU represents Channel Reconstructive Unit. And, $P(y|X)$ and $P(f|X)$ are estimated by performing forward propagation through the HTCSigNet.

role in building verification systems. In this paper, the main contribution is combining multi-scale features from CNN and transformer-based architectures in OHSV tasks. Therefore, we introduce the CNN-based feature extraction in the first. Then, we summarize some transformer-based feature learning methods in OHSV systems.

2.1. CNN-based feature extractions in OHSV tasks

The target of feature learning in OHSV tasks is to learn discriminative representations between genuine signatures and skilled forgeries. Throughout an extended period in the past, handcraft-based methods were the popular approach for feature extraction. For example, Local Binary Pattern (LBP) [9], Grey-Level Co-occurrence Matrix (GLCM) [11], and DAISY [18], which are used to extract texture features from different signatures. Furthermore, there are some handcraft-based methods to extract local, shape, and edge orientation features of the signatures, such as Histogram of Oriented Gradient (HOG) [10] and Scale-invariant Feature Transform (SIFT) [19]. In recent years, although handcraft-based feature extraction methods [20–22] have seen some development, the performance in terms of verification still falls short of expectations.

In the past decade, most of the studies were focused on using deep learning-based methods to improve verification performance in OHSV tasks. Firstly, abundant works are based on CNNs [3,5,6,13,23,24], conducting a wide of studies by harnessing the capture local features ability of CNNs. In [3], Hafemann et al. proposed a CNN-based backbone network named SigNet and trained it with genuine signatures and skilled forgeries. They demonstrated, through experiments and visualization of feature spaces, that the proposed SigNet exhibits excellent discriminative feature learning and generalization capabilities. In [5], Tsourounis et al. focused on transferring domain knowledge from handwritten text-based writer verification to OHSV tasks. They found that the proposed framework can dramatically improve the efficiency of CNN in offline signature verification tasks. In [6], Zheng et al. examined the positional coordinates of the highest value within the max-polling window. This is done to identify micro-deformations within similar strokes or unique writing tendencies among distinct writers. In [13], Liu, et al. proposed a Mutual Signature DenseNet (MSDN) for feature extraction from localized regions rather than the whole signature. Through the summary of local regions, the resulting similarity scores from numerous regions are integrated to make the conclusive verification determination. Parcham et al. [23] proposed a novel Composite Backbone Capsule Neural Network (CBCapsNet) which combines CNN and Capsule Networks. It can capture spatial properties of signature features, improve the feature extraction phase, and reduce the complexity of the network.

2.2. Transformer-based feature extractions in OHSV tasks

Transformer is first introduced in [25] and achieved tremendous success in natural language processing. With the successful applications of the transformer in computer vision tasks, some methods based on the transformer have begun to be proposed in offline signature verification tasks [14,15]. Since research on transformer-based approaches in the field of OHSV is in its early stages, studies leveraging the transformer hold significant importance. In [14], Ren et al. proposed a Two-Channel and Two-Stream (2C2S) based on the Swin transformer approach as the feature extraction framework. The proposed 2C2S established the associations among feature channels and steers the model to focus on useful information to distinguish genuine and forged signatures. In [15], Li et al. proposed a novel holistic part unified model based on Vision Transformer (VIT) [26] named TransOSV in OHSV tasks. The proposed TransOSV uses a holistic encoder to learn the global signature representations and a part decoder to learn the subtle local difference between genuine signatures and forgeries.

Although these feature extraction methods achieve good performance, they often focus on single-depth features, which may suffer from overfitting and poor generalization. Additionally, in OHSV tasks, signatures come in various languages, and it is challenging to train a feature extractor for each signature dataset, limiting the real application of these methods. Therefore, we propose a novel feature extractor based on the fusion of CNNs and transformer, aiming to deeply integrate signature local and global features to generate multi-scale features. These features simultaneously preserve local and global characteristics while demonstrating excellent generalization performance, thereby promoting the practical application of the method.

3. Hybrid transformer and convolution signature network (HTCSigNet)

In this section, we introduce how to use the proposed HTCSigNet to extract multi-scale features in OHSV tasks in detail. First, we describe the structure of the proposed HTCSigNet. Then, we delve into the specifics of the transformer block and the proposed novel CNN block

which includes the SPD-Conv module, SCConv module, and general convolution module, respectively. Finally, we introduce how to train the proposed HTCSigNet and evaluate its performance.

3.1. Overview

The proposed HTCSigNet model is illustrated in Fig. 2. The model accepts a single signature (genuine or forged) after preprocessing as input and the input signature undergoes an initial local feature extraction through a pre-convolution module. Then, two different blocks, one based on the transformer and the other based on the CNN, extract global features $\mathbf{f}_g$ and local features $\mathbf{f}_l$ respectively. To further extract local features, the SPD-Conv and SCConv modules are used to enhance the capability to extract subtle distinguishing features of the proposed HTCSigNet. Finally, multi-scale feature $\mathbf{f}$ is obtained by concatenating $\mathbf{f}_g$ and $\mathbf{f}_l$, which includes both global and local features of the signature.

3.2. Learning global representations by a transformer-based block

To learn the global information such as the overall shape of different signatures, a transformer block based on ViT [26] is introduced in the proposed HTCSigNet. As shown in Fig. 2, it can be seen that a pre-convolution module is added before the transformer encoder, which consists of three 3×3 and a 1×1 convolution. The purpose of adding this module is to enhance the generalization capability of the model, avoiding the issue of a single deep model performing well on one dataset but poorly on others.

In addition, since the convolution operation often loses global information of the image, we design a residual connection to preserve as much global information of the signature as possible in the input of the ViT, which learns global representations of signatures through the self-attention mechanism. First, the whole signature image is divided into a series of equally sized image blocks, with each block representing a region of the signature image. Next, each image block is embedded into a vector, and positional relationships are incorporated to preserve the spatial relationships between different blocks. Finally, the input sequence is processed through multiple layers of a transformer encoder composed of multi-head attention mechanisms and feedforward neural networks. Throughout the learning process, the model can focus on relationships between different positions, aiming to capture global representations.

3.3. Learning local representations by a novel composite CNN-based block

Although transformer-based block capture the global features of signatures, some local features are also important, such as the shape and morphology of strokes. To further capture the local features of signatures, we propose a novel composite CNN block to address this issue. As shown in Fig. 2, learning local representations is constructed by three parts which are SPD-Conv, SCConv, and general convolution modules, respectively. Therefore, we respectively introduce the principles and functions of these three modules in the proposed hybrid framework.

3.3.1. SPD-conv-based module

SPD-Conv module consists of a Space-to-depth (SPD) layer followed by a non-stride convolution layer [16]. As for SPD, it performs on-step downsampling on the original input or feature maps through slicing techniques. When feature map $\mathbf{X}$ of size $S \times S \times C_1$ and sampling scale is n. Let $\{i, j\} \in \{0, 1, \ldots, n-1\}$, the representation of each element sliced out from a sequence of sub-feature maps can be expressed as

$$\mathbf{f}_{i,j} = \mathbf{X}[i : S : n, j : S : n]. \tag{1}$$

Therefore, given any $\mathbf{X}$ (the original image or feature map), $\mathbf{f}_{x,y}$ is formed by all the entries $\mathbf{X}(i, j)$ that $i + x$ and $j + y$ are divisible by n and each sub-map downsamples $\mathbf{X}$ by the factor of n [16]. Then, concatenating these sub-feature maps along the channel dimension named $\mathbf{X_2}$ which compares to $\mathbf{X}$, its spatial dimension has decreased by a factor of n, while the channel dimension has increased by a factor of n^2. In other words, undergoing SPD operation, the size of $\mathbf{X}$ from $\mathbf{X}(S \times S \times C_1)$ to $\mathbf{X_2}(\frac{S}{n} \times \frac{S}{n} \times n^2 C_1)$.

After the SPD operation, to retain all the discriminative feature information as much as possible, a convolution layer with stride=1 and C_2 filters where $C_2 < n^2 C_1$ are added. Finally, it converts feature map from $\mathbf{X_2}(\frac{S}{n} \times \frac{S}{n} \times n^2 C_1)$ to $\mathbf{X_3}(\frac{S}{n} \times \frac{S}{n} \times C_2)$. To sum up, after SPD-Conv operation, the feature map is converted from $\mathbf{X}(S \times S \times C_1)$ to $\mathbf{X_3}(\frac{S}{n} \times \frac{S}{n} \times C_2)$ which can be seen in SPD-Conv module of Fig. 2.

3.3.2. SCConv-based module

SCConv module is a plug-and-play spatial and channel reconstruction convolution module that consists of two units, the Spatial Reconstruction Unit (SRU) and the Channel Reconstruction Unit (CRU) [17]. Its main properties include reducing redundant features and lowering computational complexity. Here, the purpose of using the SCConv module is to prevent model overfitting and enhance the generalization capability of the proposed HTCSigNet for extracting discriminative features from different signatures.

SRU in the SCConv module, exploits the spatial redundancy of features through a separate-and-reconstruct approach. Given a feature map $\mathbf{X} \in R^{N \times C \times H \times M}$, where N is the batch axis, C is the channel axis, H and M are the spatial height and width axes. It multiplies $\mathbf{X}$ by $\mathbf{W}_1$ and $\mathbf{W}_2$ respectively to separate the input features into $\mathbf{X}_1^w$ which has informative contents and $\mathbf{X}_2^w$ which has little or no information. $\mathbf{W}_1$ and $\mathbf{W}_2$ are the informative and no-informative weights. The whole process of acquiring $\mathbf{W}$ is as follows,

$$\mathbf{W} = Gate(Sigmoid(\mathbf{W}_\gamma(GN(\mathbf{X})))), \tag{2}$$

where $GN(\cdot)$ is Group Normalization (GN) [27] layers to assess the information of different feature maps. $\mathbf{W}_\gamma$ is the normalized correlation weights, it can be denoted as,

$$\mathbf{W}_\gamma = \omega_i = \frac{\gamma_i}{\sum_{j=1}^{C} \gamma_j}, i, j = 1, 2, \ldots, C, \tag{3}$$

where γ is the trainable parameters in GN layers to measure the variance of spatial pixels for each batch and channel. Then, it uses a cross reconstruct operation to sufficiently combine the two different weighted informative features and enhance the information flow between them, which can be expressed as,

$$\begin{cases} \mathbf{X}_1^w &= \mathbf{W}_1 \otimes \mathbf{X}, \\ \mathbf{X}_2^w &= \mathbf{W}_2 \otimes \mathbf{X}, \\ \mathbf{X}^{w1} &= \mathbf{X}_{11}^w \oplus \mathbf{X}_{22}^w, \\ \mathbf{X}^{w2} &= \mathbf{X}_{21}^w \oplus \mathbf{X}_{12}^w, \\ \mathbf{X}^w &= \mathbf{X}^{w1} \cup \mathbf{X}^{w2}, \end{cases} \tag{4}$$

where $\otimes$ is the element-wise multiplication, $\oplus$ is the element-wise summation, $\cup$ represents concatenation operation, $\mathbf{X}_{11}^w \cup \mathbf{X}_{12}^w = \mathbf{X}_1^w$ and $\mathbf{X}_{21}^w \cup \mathbf{X}_{22}^w = \mathbf{X}_2^w$.

After the SRU module, a CRU module is connected. The purpose of using the CRU module is to further diminish the redundancy of spatial-refined feature maps $\mathbf{X}^w$ in the channel dimension. In addition, CRU extracts rich features and proceeds redundant features through lightweight convolutions and feature reuse schemes. Normally, it uses repetitive standard $k \times k$ convolution to extract features, resulting in some relatively redundant feature maps in the channel dimension. Let $\mathbf{M}^k \in R^{c \times k \times k}$ as a $k \times k$ convolution kernel and $\mathbf{X}, \mathbf{Y} \in R^{c \times h \times w}$ as the input and output respectively. A standard convolution can be denoted as $\mathbf{Y} = \mathbf{M}^k \mathbf{X}$. However, three operations (Split, Transform, and Fusion are noted as STF) which in the CRU module are described in [17] and replaced the standard convolution. Therefore, the final output of the SCConv module is,

$$\mathbf{Y} = STF(\mathbf{X}^w). \tag{5}$$

Table 1
Summary of general Convolution layers.

Layer	Size	Other Parameters
Input	$128 \times 112 \times 112$	
Max-Pooling	$128 \times 3 \times 3$	Stride=2
Convolution (C1)	$256 \times 5 \times 5$	Stride=2
Max-Pooling	$128 \times 3 \times 3$	Stride=2
Convolution (C2)	$384 \times 3 \times 3$	Stride=1, padding=1
Convolution (C3)	$384 \times 3 \times 3$	Stride=1, padding=1
Convolution (C4)	$256 \times 3 \times 3$	Stride=1, padding=1
Max-Pooling	$256 \times 3 \times 3$	Stride=2
Fully Connected	512	

3.3.3. Generating final local features

After the SCConv module, a convolution module is employed to extract the final local feature representations, which is described in Table 1. In OHSV tasks, with only the features extracted by the SPD-Conv module, the smaller receptive field makes it hard to adequately capture subtle deformations of signature strokes. The SCConv module acts as an abstract transformation of feature maps, aiming to enhance the model's generalization capability. It is well-known that CNNs [3,6] have made breakthroughs in capturing local features of signatures. Therefore, we incorporate final convolution layers to capture richer local information in signatures. Here, there are multiple layers (convolution, max-pooling, and fully-connected layers), where convolutional layers and fully-connected layers have learnable parameters, which are optimized during training. Therefore, after general convolution layers, a final feature vector of 512 dimensions is generated.

3.4. Training process of the HTCSigNet

In the training stage of HTCSigNet, we only use part of the GPDSsynthetic dataset [28] to train the proposed model. The training data includes both genuine signatures and skilled forgeries, we choose the classification-based loss which is used in [3] as the loss function during the model training process. More specifically, the classification-based loss is divided into L_g and L_f. The one part of the classification-based loss L_g can be denoted as,

$$L_g = -\sum_i y_i log P(y_i|\mathbf{x}), \tag{6}$$

where y_i is the true signer for signature $\mathbf{x}$, $P(y_i|\mathbf{x})$ is the probability assigned to signer i for the signature $\mathbf{x}$. The purpose of L_g is to distinguish different signers. In other words, it learns discriminative representations between genuine signatures and random forgeries. Another part of the classification-based loss L_f can be described as,

$$L_f = -f log(P(f|\mathbf{x})) - (1-f) log(1 - P(f|\mathbf{x})), \tag{7}$$

where f is a binary label that represents whether a signature is a forgery or not. If $f = 1$, it represents that the signature is a forged signature. The purpose of L_f is to discriminate whether a signature is genuine or not. In other words, it learns discriminative representations between genuine signatures and skilled forgeries. To combine the two parts of classification-based losses, we use hyper-parameter λ to balance them as a final loss that is denoted as,

$$L = \lambda L_f + (1-\lambda) L_g, \tag{8}$$

where $\lambda \in [0, 1]$ is a hyper-parameter to control the relative importance of L_f and L_g.

3.5. Establishing the completed verification system

To evaluate the performance of the proposed method, we train both WD-based and WI-based classifiers to construct comprehensive verification systems after the feature learning stage. It should be noted that for this stage, the feature of signatures which are not included in the training process. For the WD-based verification system, Support Vector Machines (SVMs) with Radial Basis Function (RBF) kernels as the WD-based classifiers for each user. In the process of training SVMs, genuine signatures of signers from the reserved dataset are employed as positive samples, while genuine signatures from other signers within the same dataset are treated as negative samples. Given the substantial imbalance between the negative and positive samples, distinct weights are applied to the respective classes to address this issue. The objective function of the SVM can thus be described as,

$$\begin{aligned} \min\ & \frac{1}{2}\|\mathbf{w}\|^2 + C_+ \left(\sum_{i:y_i=+1} \xi_i \right) + C_- \left(\sum_{i:y_i=-1} \xi_i \right), \\ s.t.\quad & y_i(\mathbf{w}\mathbf{x}_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \end{aligned} \tag{9}$$

where, $\mathbf{x}_i$ is a training sample with target label y_i, ξ_i is the slack variables, C_+ and C_- are the weights for the positive and negative classes,

$$C_+ = \frac{N}{P} C_-, \tag{10}$$

where, P and N are the numbers of the positive and negative samples.

For the WI-based verification system, we use a distance-based method to decide whether a signature pair (i, j) belongs to a similar or dissimilar class. It selects m reference sample(s) (genuine signatures) and calculates the Euclidean distance between the reference signature and the remaining signatures. Find the maximum distance noted as d_{max} and the minimum distance noted as d_{min} from the results. It can be seen as,

$$\begin{aligned} d_{max} &= \max(d_1, d_2, \ldots, d_{n-1}), \\ d_{min} &= \min(d_1, d_2, \ldots, d_{n-1}), \end{aligned} \tag{11}$$

where n represents the number of signature pairs. Then, we employ a sliding mechanism to obtain a series threshold D between the d_{min} and d_{max} which can be described as,

$$D = \{d_{min}, d_{min} + step, d_{min} + 2step, \ldots, d_{max}\}, \tag{12}$$

where $step$ represents the sliding size. Let $d \in D$, given a pair signature $(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{x}_i$ presents reference signature and $\mathbf{x}_j$ presents query signature, the decision rule can be described as,

$$f(x_i, x_j) = \begin{cases} 0 & \text{if } dist(\mathbf{x}_i, \mathbf{x}_j) > d, \\ 1 & \text{if } dist(\mathbf{x}_i, \mathbf{x}_j) \leq d, \end{cases} \tag{13}$$

where $dist(\cdot)$ is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. $f(\mathbf{x}_i, \mathbf{x}_j) = 0$ represents the query signature $\mathbf{x}_j$ is a forged signature and $f(\mathbf{x}_i, \mathbf{x}_j) = 1$ represents the query signature $\mathbf{x}_j$ is a genuine signature. Experimenting with each threshold in the set D using the above method, selecting the optimal threshold d_{opt} as the threshold for the WI-based verification system.

4. Experiment

In this section, we conduct several experiments on different signature datasets to evaluate the proposed HTCSigNet. First, we introduce the experimental protocol. Then, we present and discuss verification performance by WD-based and WI-based methods in different language datasets, respectively. Next, we evaluate the verification performance of the proposed HTCSigNet with state-of-the-art verification systems. Finally, the effectiveness of each module in the proposed methods is demonstrated through ablation experiments.

4.1. Experimental protocol

This section mainly introduces experimental settings in detail. Firstly, it describes the datasets and preprocessing procedures used in this work. Next, detailed information is provided on the experimental settings, including data partition, training process, and verification process. Finally, the evaluation metrics and experimental conditions are introduced to evaluate the performance of the proposed HTCSigNet.

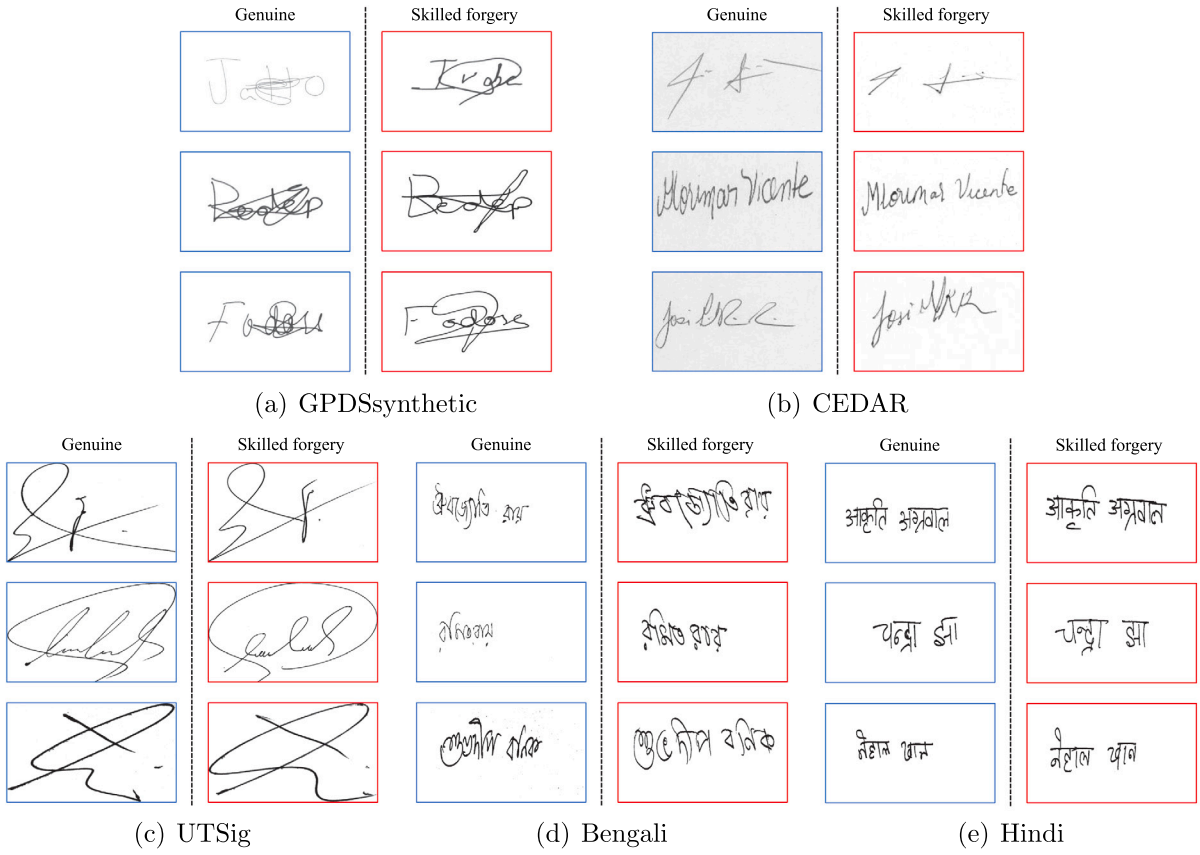


Fig. 3. Example of signature samples from different datasets.

4.1.1. Datasets and preprocessing

To demonstrate the robustness and generalization ability of the proposed method, we conduct experiments using datasets from five different languages, namely GPDSsynthetic [28], CEDAR [29], UTSIG [30], BHSig260-B [31], and BHSig260-H [31].

- **GPDSsynthetic**. It is a large-scale synthetic dataset consisting of 10,000 users, each with 24 genuine signatures and 30 skilled forged signatures. However, this work only utilized users from 0 to 1,000 for the experiments, referred to as GPDSsynthetic-1000.
- **CEDAR**. It is a English dataset and consists of 55 users and each with 24 genuine signatures combined with 24 skilled forged signatures.
- **UTSig**. It is a Farsi dataset consisting of 115 users, each with 27 genuine signatures, 36 simple signatures and 6 skilled signatures. However, experiments only utilized genuine and skilled forged signatures.
- **BHSig260-B** and **BHSig260-H**. They are two subsets in the BHSig260 dataset. BHSig260-B contains 24 genuine signatures and 30 skilled forgeries for each of 100 users. BHSig260-H consists of 160 users, each with 24 genuine signatures and 30 skilled forgeries.

Fig. 3 shows examples of genuine signatures and skilled forgeries from these five datasets. Since the original signature images are different sizes while often including different backgrounds and noise and the proposed HTCSigNet expects the inputs of a fixed shape, we construct several preprocessing steps before training the HTCSigNet. First, we use the OTSU [32] algorithm to remove the background, set its pixel as white, and set the foreground pixel to gray-scale. Then, the signature images undergo inversion by subtracting the value of each pixel from the maximum brightness, i.e., $I(x, y) = 255 - I(x, y)$, resulting in the background having a zero value [3]. Finally, signatures are uniformly resized to 256×256.

4.1.2. Experimental settings

First and foremost, it needs to be emphasized that this work only trains one feature extractor on the GPDSsynthetic-1000 dataset and other datasets are used to test the generalization performance of the feature extractor. Therefore, we further partition the GPDSsynthetic-1000 dataset in this way: the users from No. 301 to No. 1000 are used to train the HTCSigNet-based feature extractor, and the users from No. 1 to No. 300 are used to evaluate the verification performance of the trained feature extractor. During the training process, 90% of the data is used for model training, and 10% is used for validation. Therefore, the users used for feature learning and verification are completely different.

Then, as described in Section 3.4, there are two different losses in the training process and they are controlled by the hyper-parameter λ. To optimize them on the same scale, we set the $\lambda = 0.95$. We perform all experiments over NVIDIA 4090 GPU and the batch size of training is set to 32. We employ basic translations for data augmentation, selecting random crops sized 224×224 from the 256×256 signature image. The transformer encoder weights are pre-trained on the ImageNet21k dataset. We train the proposed model with Adam optimizer with a learning rate of $1e-5$.

Finally, we evaluate the performance of the proposed HTCSigNet using two approaches which are WD and WI-based verification systems. Since the contribution of this work is how to learn discriminative representations between genuine signatures and skilled forgeries in the feature extraction stage, the design of classifiers follows the same criteria in [3,6,33,34]. In the WD-based scenario, we choose the SVM with RBF kernel as the WD classifier for each user. In the SVM training process, positive samples consist of 5, 10, and 12 genuine signatures from the target user, while choosing 5, 10, and 12 genuine signatures from all remaining users as negative samples for GPDSsynthetic, CEDAR, and UTSig datasets. As for BHSig260-B and BHSig260-H datasets, positive samples consist of 5, 8, and 10 genuine signatures from the target

Table 2
Performance of WD classifiers based on HTCSigNet and baseline on the GDPSsynthetic dataset(%).

Model	#Refs	FRR	FAR	AUC(%)	EER(%)
Baseline (SigNet-F)	5	35.33	2.47	98.20	4.34(0.19)
	10	18.40	3.24	98.56	3.53(0.20)
	12	15.23	3.48	98.70	3.33(0.21)
Baseline (ViT)	5	19.77	2.02	99.26	1.95(0.23)
	10	8.22	2.89	99.45	1.54(0.12)
	12	6.28	3.08	99.49	1.41(0.10)
HTCSigNet	5	21.47	1.80	99.37	1.75(0.16)
	10	7.95	2.63	99.54	1.29(0.12)
	12	6.39	2.82	99.59	1.16(0.12)

user, while 5, 8, and 10 genuine signatures from all remaining users as negative samples. We select 10 genuine signatures and all skilled forgeries from each user, distinct from the training samples, to verify the performance of the trained SVM, respectively. We conduct the experiment by training 10 times, employing various data splits. Furthermore, the experimental results are averaged over 10 iterations. In the WI-based scenario, we use a distance-based verification method which is denoted in Section 3.5 to verify the performance of the proposed HTCSigNet.

4.1.3. Evaluation metrics

To evaluate verification performance, the following metrics are considered. **Accuracy**: the radio of correctly classified samples, which is only shown in WI-based scenario. **AUC**: area under the ROC [35] curve, which is only shown in WD-based scenario. **False Rejection Rate (FRR)**: rate at genuine signatures are incorrectly classified as forgeries. **False Acceptance Rate (FAR)**: ratio of skilled forged signatures are incorrectly classified as genuine signatures. **EER**: the error when FAR = FRR.

4.2. WD-based verification system results on the gpdssynthetic dataset

Since the proposed HTCSigNet is constructed through CNN and ViT, we set the state-of-the-art CNN-based method SigNet-F [3] and ViT as the baselines of this work. We trained SigNet-F, ViT, and HTCSigNet under the same experimental conditions, and the verification performance is shown in Table 2. From those verification results, it can be concluded that the proposed HTCSigNet outperforms both SigNet-F and ViT on most evaluation metrics.

In addition, we also compare the feature vectors from the baselines (CNN and transformer-based architectures) feature extractors with the proposed feature extractor by t-SNE [36]. The visualization results are shown in Fig. 4. We can see that genuine signatures mix with a multitude of forged ones, devoid of the distinct boundary between genuine signatures and skilled forgeries, and there are some indistinguishable points in Figs. 4(a) and 4(b). However, in Fig. 4(c), it is evident that genuine signatures and skilled forgeries can be more easily separated, and the genuine signatures of different users are more distinctly spaced apart. Some points that are difficult to distinguish have also been effectively resolved. This means that the proposed HTCSigNet not only better learns the feature representations of signatures across different users, but also learns discriminative representations more efficiently between genuine signatures and skilled forgeries.

Finally, we also compare the proposed HTCSigNet with state-of-the-art WD-based verification systems on the GPDS dataset, which includes GPDS960 [37] and GPDSsynthetic datasets. The former is a manually collected handwritten dataset, while the latter is a synthetic dataset. Table 3 shows the summary and comparison results, which demonstrate that the proposed method achieves the best performance in the GPDS dataset. It is worth noting that in each of the tables below, "Source" denotes the dataset used for training the feature extractor, and "#Refs" represents the number of real signatures used for training the SVM.

Table 3
Summary and comparison with state-of-the-art systems on GPDS dataset.

Method	Source	#Refs	EER(%)
AIRSV [38]	GPDS960Gray	5	4.53(0.14)
AIRSV [38]	GPDS960Gray	14	3.47(0.16)
SigNet-F [3]	GPDS960Gray	5	2.42(0.24)
SigNet-F [3]	GPDS960Gray	12	1.69(0.18)
SigNet-F [3]	GPDSsynthetic	5	4.34(0.19)
SigNet-F [3]	GPDSsynthetic	10	3.53(0.20)
SigNet-F [3]	GPDSsynthetic	12	3.33(0.21)
Meta-learning [39]	GPDSsynthetic	8	6.28(0.14)
Meta-learning [39]	GPDSsynthetic	10	6.09(0.25)
RBP [40]	GPDSsynthetic	5	22.13(0.42)
RBP [40]	GPDSsynthetic	12	14.93(0.18)
Zheng et al. [6]	GPDSsynthetic	5	7.11(0.41)
Zheng et al. [6]	GPDSsynthetic	10	5.38(0.36)
Zheng et al. [6]	GPDSsynthetic	12	4.52(0.42)
ViT. [26]	GPDSsynthetic	5	1.95(0.23)
ViT. [26]	GPDSsynthetic	10	1.54(0.12)
ViT. [26]	GPDSsynthetic	12	1.41(0.10)
Proposed	GPDSsynthetic	5	1.75(0.16)
Proposed	GPDSsynthetic	10	1.29(0.12)
Proposed	GPDSsynthetic	12	**1.16(0.12)**

4.3. WD-based verification results on other datasets

Evaluating the generalization performance of the proposed method on other datasets to verify whether there is an overfitting risk is a significant research aspect of this work. A state-of-the-art feature extractor should perform good performance not only on internal datasets (the dataset used for training the model) but also on external datasets (datasets not involved in the training). Therefore, we design the following experiments by using CEDAR, UTSig, and BHSig260 (BHSig260-B and BHSig260-H) datasets to explore the generalization ability of the feature extractor which is trained exclusively on the GPDSsynthetic-1000 dataset. By analyzing whether the model exhibits overfitting on these datasets, we assess the overfitting risk of the model.

For comparison with state-of-the-art systems, we employ a random selection of 8, 10, and 12 genuine signatures from both the target user and remaining users to train SVMs on CEDAR and UTSig datasets. We employ a random selection of 5, 8, and 10 genuine signatures from each user, the genuine signatures from the target user as positive samples, and the remaining users as negative samples for training the SVMs on BHSig260-B and BHSig260-H datasets. In addition, we select 5 users from the other datasets (CEDAR, UTSig, and BHSig260) to finetune the feature extractor (HTCSigNet) which is trained on the GPDSsynthetic-1000 dataset.

Table 4 shows the summary and comparison between the proposed HTCSigNet and other state-of-the-art methods on the CEDAR dataset. Here, it can be seen that the HTCSigNet outperforms the baseline (SigNet-F and ViT) when using 8, 10, and 12 reference signatures to train the SVM classifiers. It should be noted that system [5] achieves the best performance when using 3, 5, and 10 reference signatures to build a verification system. However, system [5] based on domain adaptation techniques and finetuned using signatures and the proposed method demonstrates significantly improved performance on remaining datasets compared to system [5]. In addition, the proposed method achieves competitive performance after finetuning.

Tables 5 and 6 show the performance of the proposed HTCSigNet and state-of-the-art systems on UTSig and BHSig260 datasets. From the verification results on the CEDAR, UTSig, and BHSig260 datasets, it can be seen that the proposed HTCSigNet outperforms the baseline model in all cases, even when transferred directly to these datasets without finetuning. Compared with other state-of-the-art systems, it still achieves advanced or competitive results. Therefore, it can be concluded that the proposed HTCSigNet does not exhibit overfitting risk and has good generalization ability across different scenarios.

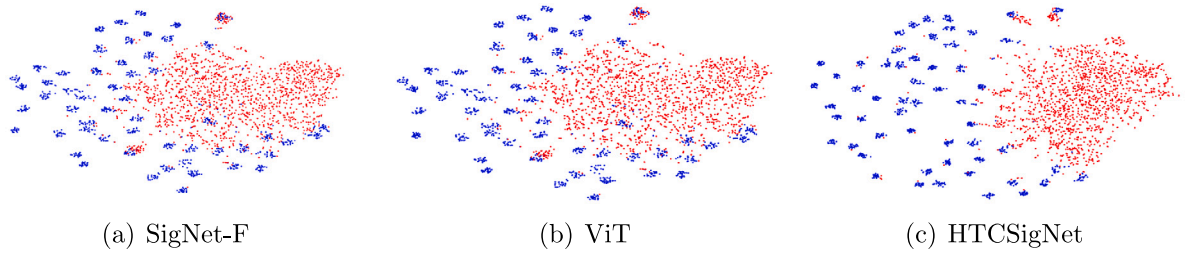


Fig. 4. The visualization of feature vectors from the first 50 users on the GPDSsynthetic-1000 dataset by t-SNE. (a) SigNet-F (CNN-based feature extractor). (b) ViT (transformer-based feature extractor). (c) the proposed feature extractor. Each blue point represents a genuine signature and the red point represents a skilled forgery. The more dispersed the blue and red points are, the better the model can distinguish between genuine signatures and skilled forgeries.

Table 4
Summary and comparison with state-of-the-art systems on CEDAR dataset.

Method	Source	#Refs	EER(%)
SigNet-F [3]	GPDSsynthetic	10	5.26(0.44)
SigNet-F [3]	GPDSsynthetic	12	5.33(1.02)
SigNet-F [3]	GPDS960Gray	8	5.37(0.46)
SigNet-F [3]	GPDS960Gray	12	4.32(0.49)
Graph-Based [41]	GPDSsynthetic	10	5.76
Meta-Learning [39]	GPDS960Gray	4	8.27(1.45)
Meta-Learning [39]	GPDS960Gray	8	7.07(1.08)
SigNet-SPP-300dpi. [42]	GPDS960Gray	10	3.60(1.26)
CCA-SigNet-F [43]	GPDS960Gray	12	2.35(0.17)
Zheng et al. [6]	GPDSsynthetic	5	3.89(0.45)
Zheng et al. [6]	GPDSsynthetic	10	2.95(0.38)
Zheng et al. [6]	GPDSsynthetic	12	2.76(0.43)
CNN-CoLL [5]	CVL-database	3	2.50
CNN-CoLL [5]	CVL-database	5	2.03
CNN-CoLL [5]	CVL-database	10	**1.66**
ViT. [26]	GPDSsynthetic	8	5.88(0.81)
ViT. [26]	GPDSsynthetic	10	5.34(0.60)
ViT. [26]	GPDSsynthetic	12	5.18(0.62)
Proposed	GPDSsynthetic	8	4.54(0.51)
Proposed	GPDSsynthetic	10	4.32(0.78)
Proposed	GPDSsynthetic	12	3.99(0.58)
Proposed (finetuned)	GPDSsynthetic	12	2.36(0.47)

Table 5
Summary and comparison with state-of-the-art systems on UTSig dataset.

Method	Source	#Refs	EER(%)
DMML [44]	GPDSsynthetic	12	17.45
HOC & HOG [45]	GPDSsynthetic	12	16.00
Graph-Based [41]	GPDSsynthetic	12	11.75
SigNet-F [3]	GPDSsynthetic	10	14.21(0.86)
SigNet-F [3]	GPDSsynthetic	12	14.21 (0.69)
SigNet-F [3]	GPDS960Gray	8	11.08(1.04)
SigNet-F [3]	GPDS960Gray	12	10.10(0.52)
ViT [26]	GPDSsynthetic	8	11.17(0.80)
ViT [26]	GPDSsynthetic	10	10.95(0.73)
ViT [26]	GPDSsynthetic	12	10.47(0.90)
Proposed	GPDSsynthetic	8	9.50(0.90)
Proposed	GPDSsynthetic	10	9.08(0.90)
Proposed	GPDSsynthetic	12	**9.05(0.65)**
Proposed(finetuned)	GPDSsynthetic	12	**8.84(0.72)**

4.4. WI-based verification system results

Although the WD-based verification system achieves better verification performance, it requires training a classifier for each user in the test set, inevitably incurring additional memory and time overheads. On the other hand, the WI-based verification system determines a decision boundary by finding a threshold value, resulting in significantly lower time and memory overheads compared to the WD-based verification system. Therefore, we use the distance-based method which is described in Section 3.5 to build a WI signature verification system and follow the same principles as [14,15,33,34], training feature extractors for each dataset which means that within each dataset, a part of the data is used for training the feature extractor, while the remains are used for testing. For the GPDSsynthetic-1000 dataset, we follow Section 4.1.2, using the last 700 users to train the feature extractor and the first 300 users for testing. For the CEDAR dataset, 50 users are employed to train the feature extractor, and 5 users are used to test. For the UTSig dataset, 90 users are employed to train the feature extractor, and 25 users are used to test. Furthermore, 50 or 80 users in the Bengali dataset are used to train the proposed model and the remaining is used to test. And 100 or 125 users in the Hindi dataset are used to train the proposed model and the remaining is used to test. In the testing phase, we randomly select 1 to 12 genuine signatures as reference samples for 12 sets of experiments and showcase the best results. The comparison between the proposed method and state-of-the-art results on 5 signature datasets are shown in Table 7.

From Table 7, we can see that the proposed HTCSigNet achieves the best EER on all datasets. It should be noted that the performance is not good on the UTSig dataset and the proposed method cannot achieve

Table 6
Summary and comparison with state-of-the-art systems on BHSig260 dataset.

Dataset	System	Source	#Refs	EER(%)
Bengali	SigNet-F [3]	GPDSsynthetic	5	10.55(0.87)
Bengali	SigNet-F [3]	GPDSsynthetic	8	8.50(0.93)
Bengali	SigNet-F [3]	GPDSsynthetic	10	7.82(0.90)
Bengali	SigNet-F [3]	GPDS960Gray	5	7.33(0.93)
Bengali	SigNet-F [3]	GPDS960Gray	10	5.21(0.38)
Bengali	Zheng et al. [6]	GPDSsynthetic	5	8.92(0.41)
Bengali	Zheng et al. [6]	GPDSsynthetic	8	8.21(0.38)
Bengali	ViT. [26]	GPDSsynthetic	5	5.33(0.55)
Bengali	ViT. [26]	GPDSsynthetic	8	4.26(0.58)
Bengali	ViT. [26]	GPDSsynthetic	10	3.86(0.44)
Bengali	Proposed	GPDSsynthetic	5	4.54(0.53)
Bengali	Proposed	GPDSsynthetic	8	3.43(0.44)
Bengali	Proposed	GPDSsynthetic	10	**3.12(0.30)**
Bengali	Proposed(finetuned)	GPDSsynthetic	10	**2.97(0.42)**
Hindi	SigNet-F [3]	GPDSsynthetic	5	8.96(0.45)
Hindi	SigNet-F [3]	GPDSsynthetic	8	7.40(0.56)
Hindi	SigNet-F [3]	GPDSsynthetic	10	6.78(0.50)
Hindi	SigNet-F [3]	GPDS960Gray	5	7.86(0.75)
Hindi	SigNet-F [3]	GPDS960Gray	10	7.40(0.56)
Hindi	Zheng et al. [6]	GPDSsynthetic	8	9.84(0.42)
Hindi	Zheng et al. [6]	GPDSsynthetic	10	9.01(0.39)
Hindi	ViT. [26]	GPDSsynthetic	5	4.03(0.26)
Hindi	ViT. [26]	GPDSsynthetic	8	3.28(0.27)
Hindi	ViT. [26]	GPDSsynthetic	10	3.02(0.25)
Hindi	Proposed	GPDSsynthetic	5	3.74(0.24)
Hindi	Proposed	GPDSsynthetic	8	3.00(0.48)
Hindi	Proposed	GPDSsynthetic	10	**2.78(0.27)**
Hindi	Proposed(finetuned)	GPDSsynthetic	10	**2.66(0.32)**

Table 7
Results of WI-based systems on the five different datasets.

Dataset	Systems	Accuracy(%)	FRR(%)	FAR(%)	EER(%)
GPDS	SigNet [33]	77.76	22.24	22.24	
	SigCNN [12]		39.54	**1.21**	7.34
	MSDN [13]				8.24
	CBCapsNet [23]	92.94	6.86	7.26	
	CBCapsNet [23]	90.87	9.45	8.81	
	TransOSV [15]		10.64	10.64	10.64
	Proposed (700/300)	**95.7**	**6.64**	2.59	**4.61**
CEDAR	SigNet [33]	100	0	0	0
	IDN [24]		2.17	5.87	3.62
	2C2S [14]	100	0	0	0
	SigCNN [12]	100	0	0	0
	CBCapsNet [23]	100	0	0	0
	Proposed (50/5)	100	0	0	0
UTSig	SigNet [33]	67.0			32.3
	LwR [46]	72.7			29.7
	Top [46]	64.2			34.5
	Proposed (90/25)	66.19	14.96	1.33	**8.15**
Bengali	SigNet [33]	86.11	13.89	13.89	
	IDN [24]	95.32	5.24	4.12	
	2C2S [14]	93.25	5.37	8.11	6.75
	SURDS [34]	87.37	19.89	5.42	
	CBCapsNet. [23]	94.3	5.11	6.29	
	TransOSV [15]		3.56	3.56	3.56
	LwR [46]	94.5			6.1
	Top [46]	88.2			11.2
	Proposed (50/50)	91.17	7.83	9.2	8.52
	Proposed (80/20)	**98.63**	**1.9**	**1.4**	**1.44**
Hindi	SigNet [33]	84.64	15.36	15.36	
	IDN [24]	93.04	4.93	8.99	
	2C2S [14]	90.68	8.66	9.98	9.32
	SURDS [34]	89.50	12.01	8.98	
	CBCapsNet [23]	100	0	0	
	TransOSV [15]		3.24	3.24	3.24
	LwR [46]	92.6			8.8
	Top [46]	83.8			15.3
	Proposed (100/60)	95.26	4.65	4.61	4.63
	Proposed (125/35)	**97.81**	**1.90**	**2.29**	**2.10**

the best EER when 50 users in the Bengali dataset and 100 users in the Hindi dataset to train the feature extractor. However, the proposed HTCSigNet achieves the best EER while adding the training samples. The reasons for this situation are as follows: first, the WI verification of this work is based on distance, but we do not consider the relationships between different pairs of signatures and there is also no optimization for the similarity or distance of different signatures in the feature space. Next, transformer-based structures require a large amount of data for training except for the GPDS dataset. However, all the other datasets are small, with very few samples available for training. For instance, in [23], there are $\binom{24}{2}$ genuine-genuine signature pairs and 24×24 genuine-skilled forged signature pairs for each user to train the feature extractor, but only $24 + 24$ signatures for each user are obtained to train the feature extractor in this work. From another perspective, the proposed method achieves competitive results with less training data, which is one of the highlights of this work.

4.5. Ablation study

The proposed HTCSigNet is mainly composed of a transformer-based block and a CNN-based block, with each block further consisting of multiple components. Therefore, to verify the effectiveness of each block, we design a series of ablation experiments. As shown in Fig. 2, the transformer encoder part consists of Norm-1, Multi-Head Attention, Norm-2, and MLP modules, while the CNN-based block is primarily made up of SPD-Conv, SCConv, and Convolution modules. These components are the key elements of the HTCSigNet model, so the ablation studies in this paper will be based on these modules. The results of ablation studies are shown in Table 8 and all presented results in this section are based on WD settings.

From Table 8, it can be seen that the proposed method achieves the best validation performance under all circumstances. In addition, we design visualization experiments to prove the effectiveness of SPD-Conv and SCConv modules. For the SPD-Conv module, we design two validation methods to verify its effectiveness in OHSV tasks. The first method involves visualizing the weights of the convolution before and after inserting this module using Grad-CAM [47]. The second method evaluates the performance of the model with and without inserting this module. The results are reported as shown in Figs. 5 and 6. For the SCConv module, we separately test the generalization performance of the model under the conditions of adding or not adding it, and the result is reported as shown in Fig. 7.

From the third row of visualization results of Fig. 5, it can be observed that the model focuses more accurately on signature strokes to learn signature features after inserting the SPD-Conv module compared to not inserting it. Warmer colors indicate larger weights. In addition, from Fig. 6, it can be found that, when SVMs are trained with different numbers of reference signatures, the performance of introducing the SPD-Conv module consistently outperforms that of not introducing the SPD-Conv module. Therefore, we prove the effectiveness of using the SPD-Conv module in OHSV tasks.

Similarly, it can also be observed from the last row of Fig. 5, that after inserting the SCConv module, the model pays more attention to the distinctive micro-deformations between genuine signatures and skilled forgeries, reducing the focus on common stroke features. In addition, although the EER obtained without the SCConv module is better than with the SCConv module in the UTSig dataset, the poor performance is primarily attributed to the unstable nature of the dataset and the significant differences among the genuine signatures of the same user, as shown in Fig. 9. It should be noted that incorporating the SCConv module demonstrates significantly better generalization performance on the remaining datasets compared to the model without the SCConv module, which can be seen in Fig. 7. Furthermore, to demonstrate that the improvement in generalization is attributed to the SCConv module rather than the SPD-Conv module, further generalization performance experiments are conducted on type (2) and (3) from Table 8. The results are shown in Fig. 8 and they indicate that the generalization performance of the proposed model is superior after inserting the SCConv module. In other words, from the above experimental results, it can be seen that the SPD-Conv and SCConv modules play a role in preventing model overfitting. By enhancing the model's focus on signature stroke features and key local area features, these modules improve the model's generalization performance and help prevent overfitting. Therefore, we have demonstrated the effectiveness of using the SCConv module in OHSV tasks.

5. Conclusion

In this paper, a novel hybrid framework based on CNN and transformer is proposed to learn discriminative representations in OHSV tasks. The main contribution involves extracting multi-scale features from handwritten signatures, which deeply integrate both local and global information of the signatures, enabling more precise verification. It is worth noting that the proposed novel composite convolution block provides more distinctive and generalized local information for the multi-scale features of signatures. In addition, the extracted multi-scale features overcome the issue of poor generalization ability of single-depth features, allowing them to be well transferred to datasets in various languages. Furthermore, the utilization of SPD-Conv and SCConv modules significantly enhances the feature learning capability by precisely focusing on signature stroke and generalization ability by focusing on more distinctive micro-deformation features while reducing attention to common features of the model in OHSV tasks.

For learning the feature representations, we use genuine and skilled forged signatures and classified-based losses to train the proposed HTCSigNet. After the feature learning stage, we build both WD-based

Table 8
Results of different combinations of modules on the GPDSsynthetic dataset.

Type	Transformer-based block				CNN-based block			EER(%)
	Norm-1	Multi-Head	Norm-2	MLP	SPD-Conv	SCConv	Convolution	
Proposed	✔	✔	✔	✔	✔	✔	✔	**1.16(0.12)**
(1)	✔	✔	✔	✔		✔	✔	1.21(0.10)
(2)	✔	✔	✔	✔				1.43(0.17)
(3)	✔	✔	✔	✔	✔		✔	1.35(0.14)
(4)	✔	✔	✔	✔			✔	1.33(0.14)
(5)					✔	✔	✔	3.72(0.19)
(6)		✔	✔	✔	✔	✔	✔	9.44(0.36)
(7)	✔		✔	✔	✔	✔	✔	3.68(0.26)
(8)	✔	✔		✔	✔	✔	✔	2.66(0.23)
(9)	✔	✔	✔		✔	✔	✔	4.53(0.25)

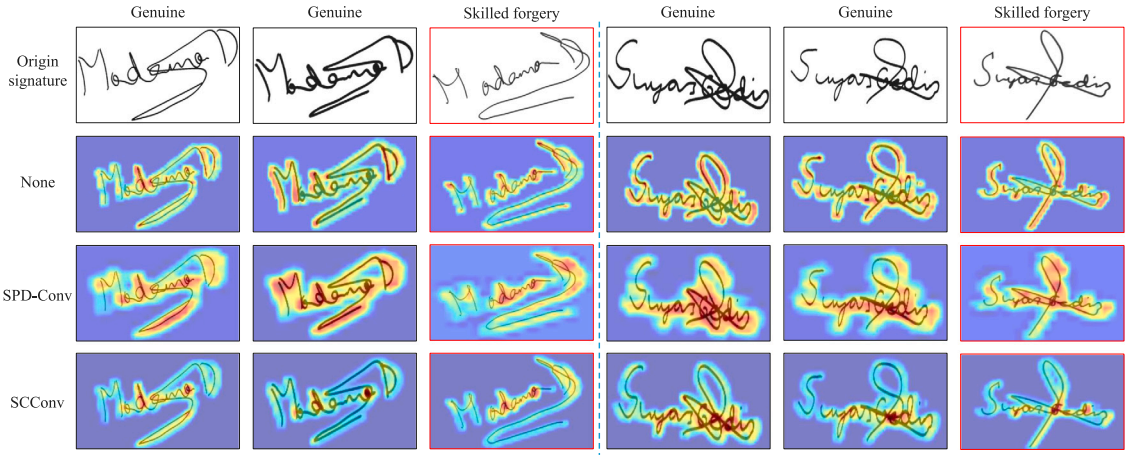


Fig. 5. Visualization results of attention maps on the GPDSsynthetic dataset. The first row shows the original signature images, the second row shows the images overlaid without SPD-Conv and SCconv modules. The third row shows the images overlaid with the SPD-Conv module and the fourth row shows the image overlaid with the SCConv module. The initial three columns represent signatures from one writer, while the subsequent three columns represent signatures from another writer. Within the set of signatures for each writer, black boxes represent genuine signatures, and red boxes represent skilled forgeries.

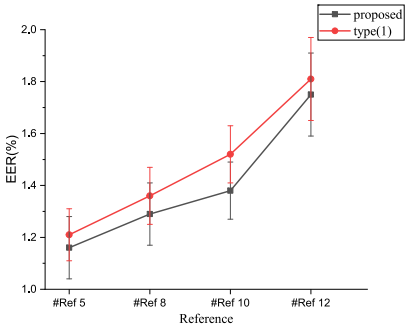


Fig. 6. Effectiveness results of the SPD-Conv module on the GPDSsynthtic dataset.

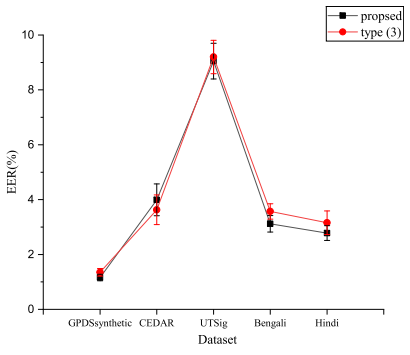


Fig. 7. Generalization results comparison of the proposed with and without the SCConv module.

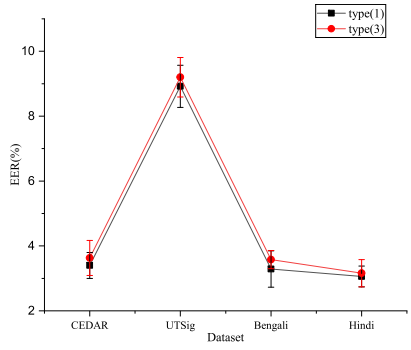


Fig. 8. Comparison of generalization performance between SCConv and SPD-Conv modules.

and WI-based verification systems to verify the performance of the proposed method. Through extensive experiments and analysis, it can be concluded that the proposed HTCSigNet achieves state-of-the-art or competitive results compared with the baseline and other advanced methods.

For future work, we will consider optimizing the distance in the feature space between different signature combinations during the feature learning stage. In addition, we will further modify the structure of the transformer encoder to explore a new structure more suitable for signature verification. Finally, we will consider proposing an end-to-end signature verification system.

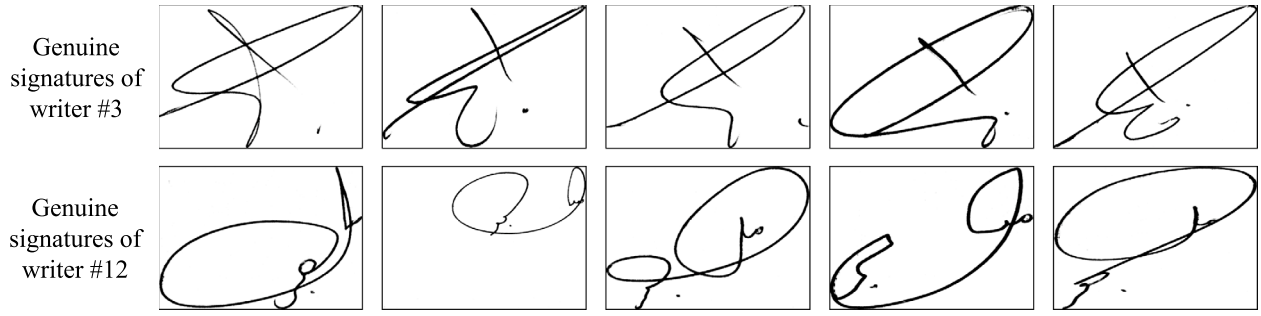


Fig. 9. Randomly-chosen genuine signatures of two writers from the UTSig dataset. Each row represents different genuine signatures of the same user, with significant stylistic variations. This is an unstable dataset, making it difficult for the model to learn effective feature representations from such inconsistent data.

CRediT authorship contribution statement

Lidong Zheng: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization . **Da Wu:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization . **Shengjie Xu:** Writing – original draft, Validation, Supervision, Software, Resources . **Yuchen Zheng:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

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

Acknowledgments

This work was supported by Innovation and Cultivation Project for Youth Talents of Shihezi University, China (Grant Number CXPY202117), Startup Project for Advanced Talents of Shihezi University, China (Grant Number RCZK2021B21).

Data availability

The authors do not have permission to share data.

References

[1] H. Suwanwiwat, A. Das, U. Pal, M. Blumenstein, ICFHR 2018 competition on thai student signatures and name components recognition and verification, in: International Conference on Frontiers in Handwriting Recognition, 2018, pp. 500–505.
[2] L.G. Hafemann, R. Sabourin, L.S. Oliveira, Offline handwritten signature verification—literature review, in: International Conference on Image Processing Theory, Tools and Applications, 2017, pp. 1–8.
[3] L.G. Hafemann, R. Sabourin, L.S. Oliveira, Learning features for offline handwritten signature verification using deep convolutional neural networks, Pattern Recognit. 70 (2017).
[4] S. Pal, M. Blumenstein, U. Pal, Off-line signature verification systems: A survey, in: International Conference & Workshop on Emerging Trends in Technology, 2011, pp. 652–657.
[5] D. Tsourounis, I. Theodorakopoulos, E.N. Zois, G. Economou, From text to signatures: Knowledge transfer for efficient deep feature learning in offline signature verification, Expert Syst. Appl. 189 (2022) 116136.
[6] Y. Zheng, B.K. Iwana, M.I. Malik, S. Ahmed, W. Ohyama, S. Uchida, Learning the micro deformations by max-pooling for offline signature verification, Pattern Recognit. 118 (2021) 108008.
[7] M.M. Hameed, R. Ahmad, M.L.M. Kiah, G. Murtaza, Machine learning-based offline signature verification systems: A systematic review, Signal Process., Image Commun. 93 (2021) 116139.
[8] D. Engin, A. Kantarci, S. Arslan, H.K. Ekenel, Offline signature verification on real-world documents, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 808–809.
[9] T. Ojala, M. Pietikainen, D. Harwood, Performance evaluation of texture measures with classification based on Kullback discrimination of distributions, in: International Conference on Pattern Recognition, Vol. 1, 1994, pp. 582–585.
[10] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vol. 1, 2005, pp. 886–893.
[11] F. Albregtsen, et al., Statistical Texture Measures Computed from Gray Level Coocurrence Matrices, vol. 5, (no. 5) Image Processing Laboratory, Department of Informatics, University of Oslo, 2008.
[12] Q. Wan, Q. Zou, Learning metric features for writer-independent signature verification using dual triplet loss, in: International Conference on Pattern Recognition, 2021, pp. 3853–3859.
[13] L. Liu, L. Huang, F. Yin, Y. Chen, Offline signature verification using a region based deep metric learning network, Pattern Recognit. 118 (2021) 108009.
[14] 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.
[15] H. Li, P. Wei, Z. Ma, C. Li, N. Zheng, TransOSV: Offline signature verification with transformers, Pattern Recognit. 145 (2024) 109882.
[16] R. Sunkara, T. Luo, No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2022, pp. 443–459.
[17] J. Li, Y. Wen, L. He, Scconv: Spatial and channel reconstruction convolution for feature redundancy, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 6153–6162.
[18] E. Tola, V. Lepetit, P. Fua, Daisy: An efficient dense descriptor applied to wide-baseline stereo, IEEE Trans. Pattern Anal. Mach. Intell. 32 (5) (2009) 815–830.
[19] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[20] M. Okawa, Synergy of foreground–background images for feature extraction: Offline signature verification using Fisher vector with fused KAZE features, Pattern Recognit. 79 (2018) 480–489.
[21] D. Banerjee, B. Chatterjee, P. Bhowal, T. Bhattacharyya, S. Malakar, R. Sarkar, A new wrapper feature selection method for language-invariant offline signature verification, Expert Syst. Appl. 186 (2021) 115756.
[22] M. Ajij, S. Pratihar, S.R. Nayak, T. Hanne, D.S. Roy, Off-line signature verification using elementary combinations of directional codes from boundary pixels, Neural Comput. Appl. (2021) 1–18.
[23] E. Parcham, M. Ilbeygi, M. Amini, CBCapsNet: A novel writer-independent offline signature verification model using a CNN-based architecture and capsule neural networks, Expert Syst. Appl. 185 (2021) 115649.
[24] P. Wei, H. Li, P. Hu, Inverse discriminative networks for handwritten signature verification, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5764–5772.
[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.
[27] Y. Wu, K. He, Group normalization, in: European Conference on Computer Vision, 2018, pp. 3–19.
[28] M.A. Ferrer, M. Diaz, C. Carmona-Duarte, A. Morales, A behavioral handwriting model for static and dynamic signature synthesis, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2016) 1041–1053.

[29] M.K. Kalera, S. Srihari, A. Xu, Offline signature verification and identification using distance statistics, Int. J. Pattern Recognit. Artif. Intell. 18 (07) (2004) 1339–1360.
[30] A. Soleimani, K. Fouladi, B.N. Araabi, UTSig: A Persian offline signature dataset, IET Biom. 6 (1) (2017) 1–8.
[31] S. Pal, A. Alaei, U. Pal, M. Blumenstein, Performance of an off-line signature verification method based on texture features on a large indic-script signature dataset, in: International Association for Pattern Recognition, 2016, pp. 72–77.
[32] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.
[33] S. Dey, A. Dutta, J.I. Toledo, S.K. Ghosh, J. Lladós, U. Pal, Signet: Convolutional siamese network for writer independent offline signature verification, 2017, arXiv preprint arXiv:1707.02131.
[34] S. Chattopadhyay, S. Manna, S. Bhattacharya, U. Pal, Surds: Self-supervised attention-guided reconstruction and dual triplet loss for writer independent offline signature verification, in: International Conference on Pattern Recognition, 2022, pp. 1600–1606.
[35] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (8) (2006) 861–874.
[36] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).
[37] F. Vargas, M. Ferrer, C. Travieso, J. Alonso, Off-line handwritten signature GPDS-960 corpus, in: International Conference on Document Analysis and Recognition, Vol. 2, 2007, pp. 764–768.
[38] Y. Serdouk, H. Nemmour, Y. Chibani, Handwritten signature verification using the quad-tree histogram of templates and a support vector-based artificial immune classification, Image Vis. Comput. 66 (2017) 26–35.
[39] L.G. Hafemann, R. Sabourin, L.S. Oliveira, Meta-learning for fast classifier adaptation to new users of signature verification systems, IEEE Trans. Inf. Forensics Secur. 15 (2019) 1735–1745.
[40] M.B. Yılmaz, K. Öztürk, Recurrent binary patterns and cnns for offline signature verification, in: Future Technologies Conference, 2020, pp. 417–434.
[41] P. Maergner, N.R. Howe, K. Riesen, R. Ingold, A. Fischer, Graph-based offline signature verification, 2019, arXiv preprint arXiv:1906.10401.
[42] L.G. Hafemann, L.S. Oliveira, R. Sabourin, Fixed-sized representation learning from offline handwritten signatures of different sizes, Int. J. Document Anal. Recognit. 21 (2018) 219–232.
[43] X. Zhao, C. Liu, B. Zhang, L. Yuan, Y. Zheng, Multi-view representation learning with deep features for offline signature verification, in: International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2021, pp. 261–275.
[44] A. Soleimani, B.N. Araabi, K. Fouladi, Deep multitask metric learning for offline signature verification, Pattern Recognit. Lett. 80 (2016) 84–90.
[45] A. Soleimani, K. Fouladi, B.N. Araabi, Persian offline signature verification based on curvature and gradient histograms, in: International Conference on Computer and Knowledge Engineering, 2016, pp. 147–152.
[46] X. ji, D. Suehiro, S. Uchida, Paired contrastive feature for highly reliable offline signature verification, Pattern Recognit. 144 (2023) 109816.
[47] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: International Conference on Computer Vision, 2017, pp. 618–626.

Lidong Zheng received his B.E. degree from School of Computer Science, HuaiNan Normal University of China in 2017. He is currently working toward the M.E degree from the College of Information Science and Technology, Shihezi University of China. His research interests include document analysis and recognition, and pattern recognition.

Da Wu received his B.E. degree from College of Information Science and Technology, Shihezi University of China in 2019. He is currently working toward the M.E degree from the College of Information Science and Technology, Shihezi University of China. His research interests include change detection, semantic segmentation, and pattern recognition.

Shengjie Xu received his B.E. degree from School of Electronic Engineering, Jiangsu Ocean University of China in 2017. He is currently working toward the M.E degree from the College of Information Science and Technology, Shihezi University of China. His research interests include change detection, semantic segmentation, and pattern recognition

Yuchen Zheng received his B.E. degree and M.E. degree from the Department of Computer Science and Technology, Ocean University of China in 2014 and 2017, and Ph.D. degree from the Department of Advanced Information Technology, Kyushu University, Japan in 2020. He is currently work as an associate professor at Shihezi University, Shihezi, China. His research interests include document analysis and recognition, pattern recognition and neural networks.