

Static-Dynamic Interaction Networks for Offline Signature Verification

Huan Li, Ping Wei*, Ping Hu

Xi'an Jiaotong University, Xi'an, China

huanli@stu.xjtu.edu.cn, pingwei@xjtu.edu.cn, helenhu@xjtu.edu.cn

Abstract

Offline signature verification is a challenging issue that is widely used in various fields. Previous approaches model this task as a static feature matching or distance metric problem of two images. In this paper, we propose a novel Static-Dynamic Interaction Network (SDINet) model which introduces sequential representation into static signature images. A static signature image is converted to sequences by assuming pseudo dynamic processes in the static image. A static representation extracting deep features from signature images describes the global information of signatures. A dynamic representation extracting sequential features with LSTM networks characterizes the local information of signatures. A dynamic-to-static attention is learned from the sequences to refine the static features. Through the static-to-dynamic conversion and the dynamic-to-static attention, the static representation and dynamic representation are unified into a compact framework. The proposed method was evaluated on four popular datasets of different languages. The extensive experimental results manifest the strength of our model.

Introduction

Offline handwritten signature verification is one of the most important biometric technologies in forensic, commercial, and financial applications. Given a reference signature image and a test signature image, signature verification is to discriminate whether the test signature is forged or genuine with respect to the reference signature (Wei, Li, and Hu 2019), as shown in Fig. 1 (a).

This is a challenging problem for the arbitrariness of personal signing habits, the sparsity of stroke information, and the camouflage of skillful forgery. The essence of signature verification is to compare the similarity of the subtle styles hidden in the reference signature and the test signature, rather than the specific contents of the signatures (Wei, Li, and Hu 2019). Previous studies model signature verification as a static feature matching problem of the two signature images. However, static features mainly describing the global patterns are insufficient to characterize the signature styles. The dynamic signing process which strings up a series of local stroke structures such as corners, curves, and

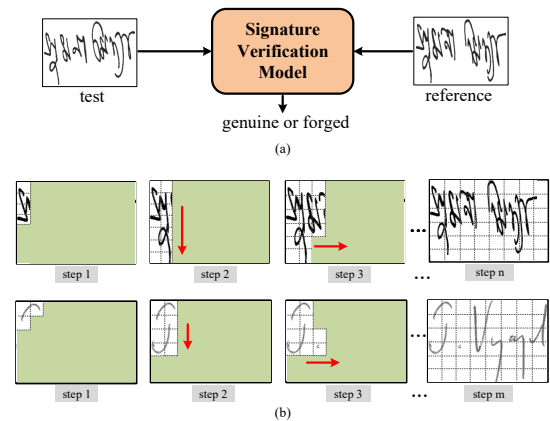


Figure 1: (a) Static signature verification. (b) Two examples that signatures are signed part by part in a sequential order.

zigzags contains abundant style information. Modeling the dynamic process could be helpful for signature verification.

However, the real signing process of finishing a signature can not be observed in the static signature image. Driven by this consideration, we can assume a pseudo dynamic process in the static image since any handwritten signature is signed part by part in sequential steps. Fig. 1 (b) illustrates the pseudo dynamic signing processes of two static signatures. In a sequential order these two signatures are signed part by part and eventually the signatures are finished. By assuming the pseudo signing process, the local patches in a static image can be converted into ‘dynamic’ sequences from which the effective signature features can be mined.

Inspired by this idea, we propose a novel Static-Dynamic Interaction Network (SDINet) model for offline signature verification. The SDINet model hypothesizes a pseudo dynamic process in a signature image and converts the signature image into a sequence of image patches. Based on this hypothesis, the model is composed of a static representation, a dynamic representation, and the interactions between the two representations. The static representation extracts deep convolutional features from the signature images and describes the global static information of signatures. The dynamic representation extracting sequential features with long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) characterizes the sequential and local informa-

*Ping Wei is the corresponding author.

tion. A dynamic-to-static attention is proposed to learn from the feedback features for re-weighting the static features. This attention mechanism is designed to intensify the effective information and weaken the redundant information in the static features. Through the static-to-dynamic conversion and the dynamic-to-static attention, the static representation and dynamic representation are unified into a deep framework. The static and dynamic features are combined to make the signature verification decision.

It should be noted that introducing the pseudo dynamic process in a signature image is not to model or characterize the real physical dynamic process of writing a signature. After all this real process cannot be observed in the static image. The objective of introducing the pseudo dynamic process is to mine the local and sequential features of signature strokes. These features are integrated with static global features to improve the performance of signature verification.

The proposed method was tested on four challenging signature datasets of different languages: CEDAR Dataset (Kalera and Xu 2004), BHSig-B Dataset (Pal et al. 2016), BHSig-H (Pal et al. 2016), and GPDS Synthetic Signature Database (Ferrer, Diaz-Cabrera, and Morales 2015a). The experimental results manifest the strength of our model.

This paper makes **three major contributions**:

1. It introduces pseudo dynamic processes into static images and proposes a novel formulation to convert static images to dynamic features.
2. It proposes a novel Static-Dynamic Interaction Network model which integrates static, dynamic, and static-dynamic interaction features for signature verification.
3. The proposed method outperforms the other comparison methods on four different language signature datasets.

Related Work

According to data collection, handwritten signature verification approaches can be roughly divided into the online ones (Lai and Jin 2018b; Guru and Prakash 2009) and the offline ones (Justino et al. 2000; Yilmaz et al. 2011; Serdouk, Nemmour, and Chibani 2015). The online approaches collect the sequence data of the whole dynamic writing process with special devices and make verification decision based on the sequence data. The offline approaches take static signature images as inputs. The online approaches require the users to sign on special devices, which is inapplicable to the document-based scenarios. The offline approaches can be used in more occasions. However, the offline approaches only have static data.

Early traditional studies extract hand-crafted features to verify signatures (Baltzakis and Papamarkos 2001; El-Yacoubi et al. 2000; Justino et al. 2000; Yilmaz and Yanikoğlu 2016; Yilmaz et al. 2011; Serdouk, Nemmour, and Chibani 2014, 2015). These hand-crafted features perform well on some datasets but the performance drops for complicated signatures with heavy noise or skillful forgery.

Recently, instead of the hand-crafted features, deep features extracted by neural network models have been widely used for signature verification (Zhang, Liu, and Cui 2017; Zagoruyko and Komodakis 2015; Gabe Alvarez 2015; Li,

Wei, and Hu 2021). Dey et al (Dey et al. 2017) proposed to compare the reference signature and the test signature with siamese networks. Hafemann et al (Hafemann, Sabourin, and Oliveira 2016, 2017) trained different convolutional neural networks to extract features for signatures verification. While these existing offline neural network approaches achieved better performance than the methods based hand-crafted features, they did not model the dynamic information hidden in static signatures. Our model simulates dynamic processes to assist signature verification and therefore acquires better performance.

Attention mechanism (Fu, Zheng, and Mei 2017; Woo et al. 2018; Hu, Shen, and Sun 2018; Wang et al. 2017) is an effective strategy to enhance useful features. We propose a novel dynamic-to-static attention which learns attention maps from sequential data to re-weight static features.

Approach

We propose a novel Static-Dynamic Interaction Network (SDINet) for offline handwritten signature verification. The model hypothesizes a pseudo dynamic process in a static signature image and jointly exploits the static features, dynamic features, and static-dynamic interactions to make verification decision. Under this framework, the signature styles are deeply learned from both the static and sequential perspectives and therefore the verification performance is improved.

Architecture

The architecture of the SDINet is shown in Fig. 2. The inputs of the model are a test signature \boldsymbol{x} and a reference signature \boldsymbol{r} . The output is a verification decision result $\bar{y} \in \{1, 0\}$, where 1 indicates \boldsymbol{x} is genuine compared with \boldsymbol{r} and 0 indicates \boldsymbol{x} is forged. The overall architecture is comprised of four major functional components: *static representation (SR)*, *static-to-dynamic conversion (StD)*, *dynamic representation (DR)*, and *dynamic-to-static attention (DtS)*, as shown in Fig. 2. The relations of the four components are: SR extracts static feature maps from the input images; StD converts the static feature maps into sequences which are fed to DR to extract sequential features and feedback features; the feedback features are fed to DtS to learn attention maps for re-weighting the static features. In this way, the four components form a closely-interacting loop. The whole model is trained in an end-to-end way.

The *static representation (SR)* component, shown as the SR blocks in Fig. 2, contains multiple cascaded convolutional layers which extract the static features from the test signature and the reference signature, respectively. Through the static representation, the static features $\boldsymbol{F}_x \in \mathbb{R}^{H \times W \times C}$ of the test signature and $\boldsymbol{F}_r \in \mathbb{R}^{H \times W \times C}$ of the reference signature are extracted from the two signature images, respectively. H , W , and C are the height, width, and channel number of the feature map, respectively.

The *static-to-dynamic conversion (StD)* component, shown as the StD blocks in Fig. 2, receives the static feature maps \boldsymbol{F}_x and \boldsymbol{F}_r from the static representation and converts \boldsymbol{F}_x and \boldsymbol{F}_r into sequences. Corresponding to the two static feature extraction streams, there are two StD compo-

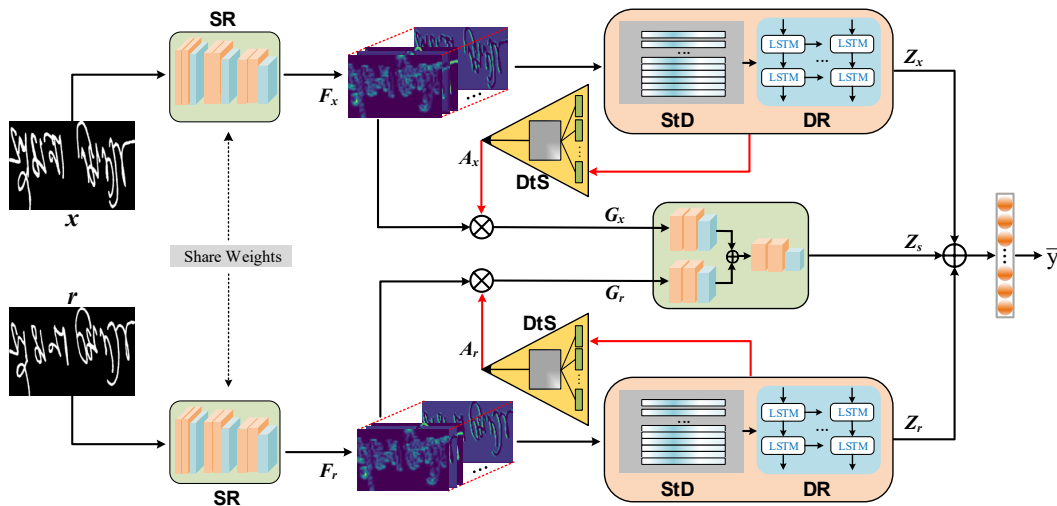


Figure 2: Overall architecture of the Static-Dynamic Interaction Network. ‘SR’, ‘StD’, ‘DR’, and ‘DtS’ indicate static representation, static-to-dynamic conversion, dynamic representation, and dynamic-to-static attention, respectively.

nents which address F_x and F_r respectively. The structure of StD will be detailed in the following section.

The *dynamic representation (DR)* component, shown as the DR blocks in Fig. 2, receives the converted sequences from the StD component and outputs the sequential features Z_x and Z_r of the test signature and the reference signature, respectively. It aims to extract the sequential features of the signatures. A LSTM module is utilized to formulate the sequence data of signatures and outputs the dynamic features Z_x and Z_r . Meanwhile, the DR component also produces feedback features which are used to generate attention maps for re-weighting the static features.

The *dynamic-to-static attention (DtS)* component, shown as the DtS triangular blocks in Fig. 2, receives feedback features from the DR component and learns the attention maps A_x and A_r for the static features F_x and F_r . The DtS component seeks to enhance the informative static features and restrain the ineffective features. It associates the dynamic representation with the static representation in a feedback way, which feeds the information hidden in dynamic sequences back to the static features.

Multiplying the static features F_x and F_r by the attention weight maps A_x and A_r generates the refined static features G_x and G_r , respectively. A convolutional network module is used to further extract deep features from G_x and G_r , respectively. The extracted deep features from G_x and G_r are concatenated by channels into unified feature maps, which are sent to another convolutional module to fuse the features. The fusion features are sent to a global average pooling (GAP) layer to generate the final static features Z_s of the input reference and test signature images.

The concatenation of the static features Z_s and the dynamic features Z_x , Z_r is sent to a fully-connected layer. The final result is computed with a binary softmax classifier.

Static-To-Dynamic Conversion

The static-to-dynamic conversion component is designed to convert the static feature maps of signatures into dynamic sequences which simulate the hidden process of signing the signatures. Handwritten signatures are written by humans in a sequential way. The dynamic process of signing the signatures contains rich information about the writing styles of the signers. However, in static signature images, the real signing process cannot be observed and all the dynamic information has been hidden in the complete static signatures. How to convert the static signatures into dynamic sequences and how to simulate the signing process hidden in the image are the key issues for signature verification.

We observe that humans write an entire signature part by part. Fortunately, the convolutional feature extraction also follows the “block by block” way, i.e each local receptive field is a block and the convolutional computation is sequentially implemented block by block in both the vertical and horizontal directions. In this way, we can imagine that each local receptive field in the convolution computation corresponds to a small part of the signature and the convolution computation process characterizes the signing process.

Thus, a feature map extracted from the static signature image is uniformly split into rows and columns. We assume that each row or column represents one dynamic unit of the signing process. All of the rows and columns form the dynamic sequences of signing the signature. Since a static signature image has multiple channels of convolutional feature maps, to enhance the descriptive ability of the dynamic sequences, we use a convolutional network to fuse the multi-channel feature maps and convert them to dynamic sequences. The conversion is illustrated in Fig. 3.

We use the static feature maps $F_x \in \mathbb{R}^{H \times W \times C}$ of the test signature to explain the static-to-dynamic conversion. We use a 1×3 kernel and a 3×1 kernel respectively to implement convolution on F_x and generate two $H \times W \times 1$

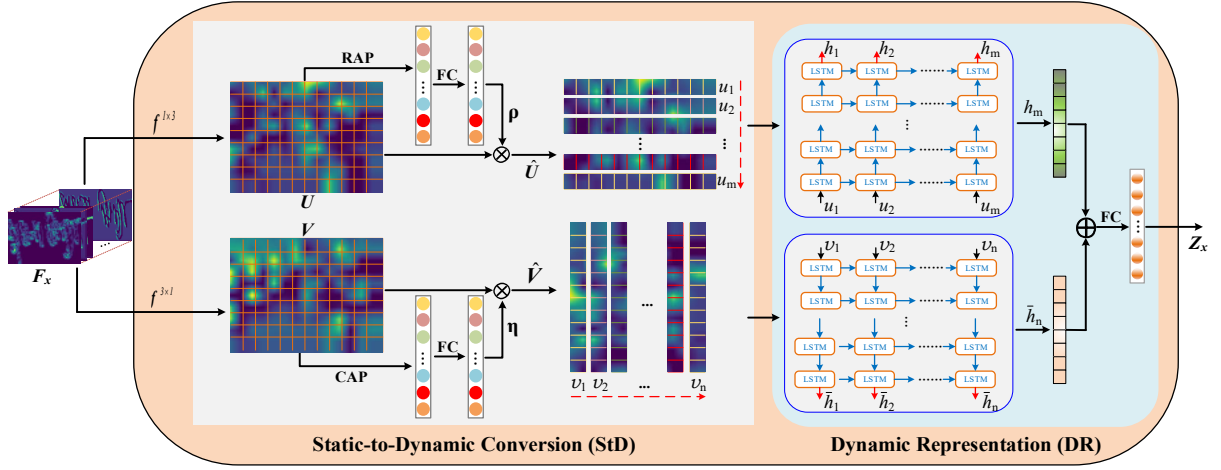


Figure 3: The inside structures of the static-to-dynamic conversion and dynamic representation components.

feature maps. This process is summarized as

$$U = f^{1 \times 3}(F_x), \quad V = f^{3 \times 1}(F_x) \quad (1)$$

where $f^{1 \times 3}$ and $f^{3 \times 1}$ represent the convolutions along the rows and columns, respectively. $U \in \mathbb{R}^{H \times W \times 1}$ and $V \in \mathbb{R}^{H \times W \times 1}$ are the 2D feature maps of F_x along the rows and columns, respectively, as shown in Fig. 3.

For the sequence representation, the weight of each row or column in the 2D feature maps should be different. Therefore we introduce a process that learns weights for each row and column of the 2D feature maps. Firstly, a row average pooling (RAP) and a column average pooling (CAP) are applied to U and V respectively. A following fully-connected layer (FC) with sigmoid function computes the row weights ρ and the column weights η , which multiply U and V respectively to generate the weighted 2D feature maps \hat{U} and \hat{V} . This process is formulated as follows:

$$\hat{U} = \rho \odot U, \quad \hat{V} = \eta \odot V \quad (2)$$

where $\rho \odot U$ means multiplying each row of U with each item in ρ respectively. $\eta \odot V$ means multiplying each column of V with each item in η respectively.

Then we generate the dynamic sequence by splitting the \hat{U} and \hat{V} in different dimensions. Uniformly splitting \hat{U} into m rows generates a sequence $u = [u_1, u_2, \dots, u_m]$, and splitting \hat{V} into n columns produces another sequence $v = [v_1, v_2, \dots, v_n]$, where m and n denote the sequence lengths. Each sequence consists of a series of fixed-length vectors. Each vector u_t ($t = 1, 2, \dots, m$) or v_t ($t = 1, 2, \dots, n$) can be regarded as a sequence frame at time point t .

The two sequences u and v acquired from different dimensions of the static features simulate the signing processes in different directions. They are used for the dynamic representation of signature images.

Dynamic Representation

Given the dynamic sequence $u = [u_1, u_2, \dots, u_m]$ and $v = [v_1, v_2, \dots, v_n]$, we seek to learn the dynamic features of the signatures. We use a deep Long Short Term Memory

(LSTM) module to represent the signature sequences and learn the deep dynamic features, as shown in Fig. 3. It is a multilayer network architecture receiving the sequences u or v as inputs, where each item of u or v is taken as a frame at a time point. The proposed module has three stacked LSTM layers to learn the dynamic features. The LSTM architecture computes the hidden features at each time point and outputs the feature sequences, which is represented as:

$$h = \phi(u), \quad \bar{h} = \phi(v) \quad (3)$$

where ϕ represents the proposed LSTM architecture. $h = [h_1, h_2, \dots, h_m]$ is the output feature sequence for u , where h_t ($t = 1, 2, \dots, m$) is a feature at time point t . $\bar{h} = [\bar{h}_1, \bar{h}_2, \dots, \bar{h}_n]$ is the output for the sequence v .

h_m and \bar{h}_n are the dynamic features output at the last time point in the feature sequence h and \bar{h} , respectively. The concatenation of h_m and \bar{h}_n is fed to a fully-connected layer to output the final dynamic features of the signature. For the reference signature r and the test signature x , the corresponding dynamic features are denoted as Z_r and Z_x , respectively. The signature verification decision is made based on static features and these two dynamic features.

The learned h and \bar{h} at all time points contain rich information about the signature in the signing process. Thus h and \bar{h} are fed back to learn the attention maps for the static features. In this sense, we regard h and \bar{h} as the feedback sequence features, which will be detailed as follows.

Dynamic-To-Static Attention

The dynamic representation (DR) component produces the dynamic features Z_r and Z_x for verification decision, and also the feedback sequence features for learning attention maps for static features.

Learning static attention from sequence features is inspired by the characteristic of signatures. On a static feature map, the features at different locations should have different weights. This is because based on the dynamic signing process the effects of features at successive locations on signature verification are closely correlated. For example, if the authenticity score at one location is very low, the score at

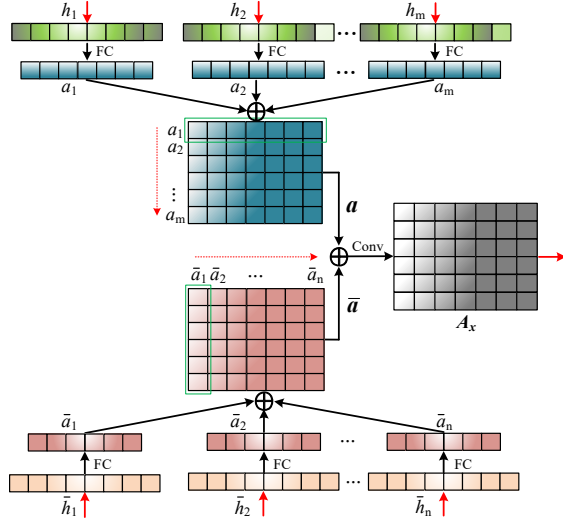


Figure 4: Dynamic-to-static attention.

the next signing location would not be very high. This is determined by the smoothness of the signing process and the coherence of the signing style. We propose a Dynamic-To-Static Attention (DtS) to learn this attention maps from the feedback sequence features, as shown in Fig. 4.

The sequence features $\mathbf{h} = [h_1, h_2, \dots, h_m]$ and $\bar{\mathbf{h}} = [\bar{h}_1, \bar{h}_2, \dots, \bar{h}_n]$ correspond to the row sequence and column sequence respectively. h_t or \bar{h}_t represents the long-range temporal relationships that indicate all of the previous signing steps have impacts on the current signing step. To learn the attention maps for the static features, the proposed DtS component accepts \mathbf{h} and $\bar{\mathbf{h}}$ as inputs, which are fed to the fully-connected layers with sigmoid function to generate the attention sequences $[a_1, a_2, \dots, a_m]$ and $[\bar{a}_1, \bar{a}_2, \dots, \bar{a}_n]$, respectively. Combining the attention sequences of $[a_1, a_2, \dots, a_m]$ by rows generates the attention map $\mathbf{a} = [a_1, a_2, \dots, a_m]$. Combining the sequences of $[\bar{a}_1, \bar{a}_2, \dots, \bar{a}_n]$ by columns generates the attention map $\bar{\mathbf{a}} = [\bar{a}_1, \bar{a}_2, \dots, \bar{a}_n]$. These two maps are shown in Fig. 4.

We regard \mathbf{a} and $\bar{\mathbf{a}}$ as two channels of feature maps which are fed to a convolutional layer with sigmoid function. The output of this convolutional layer is the final attention map \mathbf{A}_x , which is used to refine the static features \mathbf{F}_x . For the reference signature image, the attention map is denoted as \mathbf{A}_r . The refined static features are

$$\mathbf{G}_r = \mathbf{F}_r \otimes \mathbf{A}_r, \quad \mathbf{G}_x = \mathbf{F}_x \otimes \mathbf{A}_x \quad (4)$$

where \otimes denotes the element-wise multiplication for each channel of the static features. \mathbf{G}_r and \mathbf{G}_x are the re-weighted static features corresponding to the reference signature and test signature, respectively. \mathbf{G}_r and \mathbf{G}_x are fed to a convolutional module to generate the combined static feature \mathbf{Z}_s , as described in Section Architecture.

The proposed dynamic-to-static method learns attention for static features from sequence features. It is a novel mechanism and different from the previous approaches which learn attention for static features from static data or learn attention for sequence features from sequences.

Loss Function

For model learning, suppose a training set $\{(\mathbf{x}^i, \mathbf{r}^i, y^i) | i = 1, \dots, N\}$ is consisted of N signature samples, where \mathbf{x}^i is the test signature and \mathbf{r}^i is the reference signature. $y^i \in \{0, 1\}$ is a binary ground truth label of the test signature, where 1 indicates the genuine one and 0 means the forged one. We define \hat{y}^i as the probability value predicted by the model for the reference-test pair $(\mathbf{x}^i, \mathbf{r}^i)$. For all the training samples, the loss function is

$$\mathcal{L} = - \sum_{i=1}^N [y^i \log \hat{y}^i + (1 - y^i) \log(1 - \hat{y}^i)]. \quad (5)$$

Experiments

We test our SDINet model on four public signature datasets: CEDAR Dataset (Kalera and Xu 2004), BHSig-B Dataset (Pal et al. 2016), BHSig-H (Pal et al. 2016), and GPDS Synthetic Signature Database (Ferrer, Diaz-Cabrera, and Morales 2015a). We also perform extensive ablation study experiments. The results manifest the effectiveness of SDINet.

Experimental Setup

Five widely-used metrics are adopted to evaluate the method: False Acceptance Rate (FAR), False Rejection Rate (FRR), Accuracy (Acc), Equal Error Rate (EER), and Area Under Curve (AUC). All signature images are preprocessed by removing backgrounds using OTSU algorithm (Otsu 1979) and non-standard Binarization that is the same as (Wei, Li, and Hu 2019). We resize all images to the same size of 155×220 . We construct the proposed model based on TensorFlow 1.8.0. The parameters of batch normalization layer are set as decay=0.99 and $\epsilon = 10^{-5}$ respectively.

CEDAR Dataset

Each individual on CEDAR signature dataset has 24 genuine and 24 forged signatures. Referring to previous approaches, 50 people's signatures are used to train our model and the rest of 5 people's signatures for test. According to writer-independent signature verification method, a positive sample is consisted of a reference signature and a genuine signature; a reference signature and a forged signature are paired as a negative sample. Each signatory has 276 positive pairs and 276 negative pairs.

We compare our SDINet model with other typical approaches such as Morphology (Kumar et al. 2010), IDN (Wei, Li, and Hu 2019), Surroundness (Kumar, Sharma, and Chanda 2012), and SigNet-F (Hafemann, Sabourin, and Oliveira 2017). Table 1 shows the results of different approaches. The experiment results in this table manifest that our SDINet model outperforms the existing approaches. Although the 2.17% FRR in IDN (Wei, Li, and Hu 2019) is better than our model, FRR and FAR are the mutually restricted indicators. The comprehensive metric EER of our model is the best, which convincingly proves the effectiveness of our method.

The reason that SDINet model outperforms other approaches is that our model introduces the dynamic process

Model	FRR	FAR	EER
IDN (Wei, Li, and Hu 2019)	2.17	5.87	3.62
Morphology (Kumar et al. 2010)	12.39	11.23	11.59
Graph Matching (Chen and Srihari 2006)	7.7	8.2	
HOCCNN (Shariatmadari, Emadi, and Akbari 2019)	4.79	5.07	-
Partially Ordered Sets (Zois, Alewijnse, and Economou 2016)	5.83	11.52	3.02
Chain Code (Bharathi and Shekar 2013)	9.36	7.84	-
Tree structured sparsity (Zois et al. 2018)	6.95	6.06	2.30
Archetypes (Zois, Theodorakopoulos, and Economou 2017)	2.07	2.07	2.07
Surroundness (Kumar, Sharma, and Chanda 2012)	8.33	8.33	-
Distance Statistics (Kalera and Xu 2004)	20.62	23.18	21.90
SigNet-F (Hafemann, Sabourin, and Oliveira 2017)	-	-	4.63
PDSN (Lai and Jin 2018a)	-	-	4.37
Triplet Nets-Graph (Maergner et al. 2019)	12.21	12.35	12.27
Our SDINet	3.42	0.73	1.75

Table 1: Signature verification comparison on CEDAR(%).

and mines the dynamic features. Other approaches only extract static features which cannot sufficiently characterize the essential styles of signatures.

BHSig-B and BHSig-H Dataset

BHSig-B data and BHSig-H data are contained in the whole BHSig260 dataset. BHSig-B Dataset contains 100 people’s signatures signed by Bengal. Each signer has 24 genuine signatures and 30 forged signatures. We use 50 people’s signatures for training and the rest of people’s signature images for test. BHSig-H Dataset contains 160 peoples’ signature images. Each signatory also has 24 genuine signatures and 30 forged signatures. We use 100 people’s signatures to train our model and the rest 60 persons’ signatures as testing data. On these two datasets, we compare our method with approaches such as SigNet (Dey et al. 2017), FHTF (Bhunias, Alaei, and Roy 2019), Correlated Feature (Dutta, Pal, and Lladós 2016) and others.

Table 2 shows the comparison results. On BHSig-B Dataset, our method obtains 94.42% Acc, 7.86% FRR and 3.30% FAR; on BHSig-H, it obtains 95.00% Acc, 3.77% FRR and 6.24% FAR. Our model achieves better performance than other approaches, which proves the effectiveness of the proposed strategies and modules.

GPDS Synthetic Dataset

GPDS Synthetic Dataset (Ferrer, Diaz-Cabrera, and Morales 2015b) is a large-scale and challenging signature

BHSig-B Dataset			
Model	FRR	FAR	Acc
SigNet (Dey et al. 2017)	13.89	13.89	86.11
Texture Feature (Pal et al. 2016)	33.82	33.82	66.18
FHTF (Bhunias, Alaei, and Roy 2019)	11.46	10.36	-
IsRFsM (Alaei et al. 2017)	30.12	16.18	-
Correlated Feature (Dutta, Pal, and Lladós 2016)	14.43	15.78	84.90
Our SDINet	7.86	3.30	94.42
BHSig-H Dataset			
IDN (Wei, Li, and Hu 2019)	4.93	8.99	93.04
SigNet (Dey et al. 2017)	15.36	15.36	84.64
Texture Feature (Pal et al. 2016)	24.47	24.47	75.53
Correlated Feature (Dutta, Pal, and Lladós 2016)	15.09	13.10	85.90
FHTF (Bhunias, Alaei, and Roy 2019)	11.46	10.36	-
IsRFsM (Alaei et al. 2017)	30.12	16.18	-
Our SDINet	3.77	6.24	95.00

Table 2: Signature verification comparison on BHSig-H and BHSig-B Dataset (%).

Model	FRR	FAR	Acc
SigNet (Dey et al. 2017)	22.24	22.24	77.76
Correlated Feature (Dutta, Pal, and Lladós 2016)	27.62	28.34	73.67
Our SDINet	8.32	12.37	89.66

Table 3: Signature verification comparison on GPDS (%).

verification dataset of Spanish. It is composed of 4000 people’s signature samples, of which each person has 24 genuine signatures and 30 forged signatures. On this dataset, 3200 people’s signatures are utilized to train the model and the rest people’s signatures as testing set.

On this database, our model is compared with SigNet (Dey et al. 2017) and Correlated Feature (Dutta, Pal, and Lladós 2016). Table 3 shows the experiment results. Our model achieves 8.32% FRR, 12.37% FAR, and 89.66% Acc, respectively, which outperforms the comparison methods under all metrics by a remarkable margin.

Ablation Study

Our model introduces novel pseudo dynamic process and dynamic-to-static attention for signature verification. It includes static representation (SR), dynamic representation (DR), and dynamic-to-static attention (DtS) function components. To analyse the effects of different components, we carried out four group ablation experiments on the four signature datasets: (1) ‘SR’ is the base network model only using static representation; (2) ‘SR + DR’ uses static representation with dynamic representation; (3) ‘SR + DtS’ uses static representation with dynamic-to-static attention; (4) ‘SR + DtS + DR’ is our overall model (SDINet).

Model	CEDAR			BHSig-Bengali			BHSig-Hindi			GPDS		
	EER	Acc	AUC	EER	Acc	AUC	EER	Acc	AUC	EER	Acc	AUC
SR	3.56	96.54	99.43	8.71	91.39	97.36	8.21	91.66	97.74	10.97	88.98	95.83
SR + DR	2.18	97.86	99.82	7.51	92.48	97.90	6.15	93.84	98.60	10.36	89.48	96.19
SR + DtS	2.40	97.60	99.69	5.43	94.41	98.86	5.74	94.09	98.77	10.40	89.58	96.20
SR + DtS + DR (Our SDINet)	1.75	97.93	99.84	5.39	94.42	98.89	5.11	95.00	98.84	10.43	89.66	96.10

Table 4: Ablation analysis of dynamic representation (DR) and dynamic-to-static attention (DtS) (%).

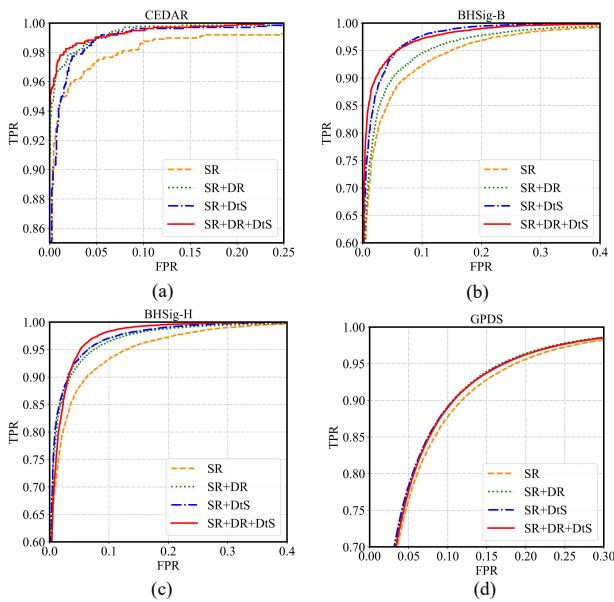


Figure 5: The ROC curves of different methods.

Table 4 shows the ablation experiment results and Fig. 5 shows the ROC curves of different methods. The method ‘SR + DR’ outperforms the ‘SR’ method on all the four datasets of all evaluation metrics. ‘SR + DtS + DR’ performs much better than ‘SR + DtS’. These results prove that introducing the dynamic representation can greatly improve the performance. The DR extracts dynamic features that provide important information of signing styles and therefore the incorporation of DR greatly improves performance.

Furthermore, the method ‘SR + DtS’ outperforms the method ‘SR’ and the method ‘SR + DtS + DR’ outperforms the method ‘SR + DR’. These two groups of comparison manifest the effect of the dynamic-to-static attention (DtS). DtS learns weight maps to refine static features, which helps the model to gain impressive performance improvement.

In addition, we show the features to visually analyse the effect of the dynamic-to-static attention. Fig. 6 shows examples of original signature images (the first row), the static feature maps (the second row ‘SR’), and the refined static features by the dynamic-to-static attention (the third row ‘SR+DtS’), where warmer colors indicate larger weights. As is shown, compared with SR, the model with dynamic-to-static attention can exactly focus on the signature stroke

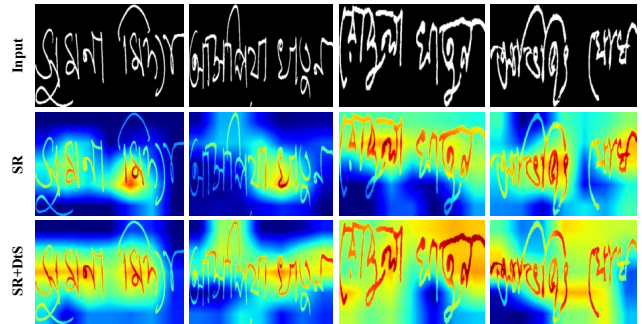


Figure 6: Visualization analysis for effectiveness of the dynamic-to-static attention (DtS).

areas with informative features. The visual results further prove the effect of dynamic-to-static attention.

The above ablation experiments comprehensively prove the effects of the dynamic representation and the dynamic-to-static attention.

Conclusion

In this paper, we formulate a novel Static-Dynamic Interaction Network model for offline writer-independent signature verification that contains four key functional components: static representation, static-to-dynamic conversion, dynamic representation, and dynamic-to-static attention. Our model utilizes static features and dynamic features to make decision. The proposed model is evaluated on four popular signature datasets and the experiment results have manifested the effectiveness of our SDINet model. The future work will concentrate on the extension of the dynamic representation to other relevant tasks.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (No. 61876149).

References

- Alaei, A.; Pal, S.; Pal, U.; and Blumenstein, M. 2017. An Efficient Signature Verification Method Based on an Interval Symbolic Representation and a Fuzzy Similarity Measure. *IEEE Transactions on Information Forensics and Security* 12(10): 2360–2372.

- Baltzakis, H.; and Papamarkos, N. 2001. A new signature verification technique based on a two-stage neural network classifier. *Engineering Applications of Artificial Intelligence* 14(1): 95–103.
- Bharathi, R. K.; and Shekar, B. H. 2013. Off-line signature verification based on chain code histogram and Support Vector Machine. In *International Conference on Advances in Computing, Communications and Informatics*.
- Bhunia, A.; Alaei, A.; and Roy, P. 2019. Signature Verification Approach using Fusion of Hybrid Texture Features. *Neural Computing and Applications* (5): 8737–8748.
- Chen, S.; and Srihari, S. 2006. A New Off-line Signature Verification Method based on Graph. In *International Conference on Pattern Recognition*.
- Dey, S.; Dutta, A.; Toledo, J. I.; Ghosh, S. K.; Lladós, J.; and Pal, U. 2017. SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification. *CoRR* abs/1707.02131.
- Dutta, A.; Pal, U.; and Lladós, J. 2016. Compact correlated features for writer independent signature verification. *International Conference on Pattern Recognition*.
- El-Yacoubi, A.; Justino, E. J. R.; Sabourin, R.; and Bortolozzi, F. 2000. Off-line signature verification using HMMs and cross-validation. *Proceedings of the IEEE Signal Processing Society Workshop 2*: 859–868 vol.2.
- Ferrer, M. A.; Diaz-Cabrera, M.; and Morales, A. 2015a. Static Signature Synthesis: A Neuromotor Inspired Approach for Biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(3): 667–680.
- Ferrer, M. A.; Diaz-Cabrera, M.; and Morales, A. 2015b. Static Signature Synthesis: A Neuromotor Inspired Approach for Biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(3): 667–680.
- Fu, J.; Zheng, H.; and Mei, T. 2017. Look Closer to See Better: Recurrent Attention Convolutional Neural Network for Fine-grained Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- Gabe Alvarez, Blue Sheffer, M. B. 2015. Offline Signature Verification with Convolutional Neural Networks. *Technical report, Stanford University*.
- Guru, D. S.; and Prakash, H. N. 2009. Online Signature Verification and Recognition: An Approach Based on Symbolic Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(6): 1059–1073.
- Hafemann, L. G.; Sabourin, R.; and Oliveira, L. S. 2016. Writer-independent feature learning for Offline Signature Verification using Deep Convolutional Neural Networks. In *International Joint Conference on Neural Networks*.
- Hafemann, L. G.; Sabourin, R.; and Oliveira, L. S. 2017. Learning Features for Offline Handwritten Signature Verification using Deep Convolutional Neural Networks. *Pattern Recognition* 70: 163–176.
- Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Comput.* 9: 1735–80.
- Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-Excitation Networks. *IEEE Conference on Computer Vision and Pattern Recognition*.
- Justino, E. J. R.; Yacoubi, A. E.; Bortolozzi, F.; and Sabourin, R. 2000. An Off-Line Signature Verification System Using HMM and Graphometric Features.
- Kalera, M.; and Xu, A. 2004. Offline Signature Verification And Identification Using Distance Statistics. *International Journal of Pattern Recognition and Artificial Intelligence* 18: 1339–1360.
- Kumar, R.; Kundu, L.; Chanda, B.; and Sharma, J. D. 2010. A writer-independent off-line signature verification system based on signature morphology. In *International Conference on Intelligent Interactive Technologies and Multimedia*.
- Kumar, R.; Sharma, J. D.; and Chanda, B. 2012. Writer-independent off-line signature verification using surroundedness feature. *Pattern Recognition Letters* 33: 301–308.
- Lai, S.; and Jin, L. 2018a. Learning Discriminative Feature Hierarchies for Off-Line Signature Verification. In *International Conference on Frontiers in Handwriting Recognition*.
- Lai, S.; and Jin, L. 2018b. Recurrent Adaptation Networks for Online Signature Verification. *IEEE Transactions on Information Forensics and Security* 14(6): 1624–1637.
- Li, H.; Wei, P.; and Hu, P. 2021. AVN: An Adversarial Variation Network Model for Handwritten Signature Verification. *IEEE Transactions on Multimedia* doi:10.1109/TMM.2021.3056217.
- Maergner, P.; Pondenkandath, V.; Alberti, M.; Liwicki, M.; Riesen, K.; Ingold, R.; and Fischer, A. 2019. Combining graph edit distance and triplet networks for off-line signature verification. *Pattern Recognition Letters* 125: 527–533.
- Otsu, N. 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man and Cybernetics* 9(1): 62–66.
- Pal, S.; Alaei, A.; Pal, U.; and Blumenstein, M. 2016. Performance of an Off-Line Signature Verification Method Based on Texture Features on a Large Indic-Script Signature Dataset. In *12th IAPR Workshop on Document Analysis Systems*.

- Serdouk, Y.; Nemmour, H.; and Chibani, Y. 2014. Combination of OC-LBP and Longest Run Features for Off-Line Signature Verification. In *Tenth International Conference on Signal-Image Technology and Internet-Based Systems*.
- Serdouk, Y.; Nemmour, H.; and Chibani, Y. 2015. Orthogonal Combination and Rotation Invariant of Local Binary Patterns for Off-line Handwritten Signature Verification. In *the International Conference on Telecommunications and ICT*.
- Shariatmadari, S.; Emadi, S.; and Akbari, Y. 2019. Patch-based offline signature verification using one-class hierarchical deep learning. *International Journal on Document Analysis and Recognition* 22: 375–385.
- Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; and Tang, X. 2017. Residual Attention Network for Image Classification. *IEEE Conference on Computer Vision and Pattern Recognition*.
- Wei, P.; Li, H.; and Hu, P. 2019. Inverse Discriminative Networks for Handwritten Signature Verification. In *The IEEE Conference on Computer Vision and Pattern Recognition*.
- Woo, S.; Park, J.; Lee, J.; and Kweon, I. S. 2018. CBAM: Convolutional Block Attention Module. *European Conference on Computer Vision*.
- Yilmaz, M. B.; Yanikoglu, B.; Tirkaz, C.; and Kholmatov, A. 2011. Offline signature verification using classifier combination of HOG and LBP features. In *International Joint Conference on Biometrics*.
- Yilmaz, M. B.; and Yanikoğlu, B. 2016. Score level fusion of classifiers in off-line signature verification. *Information Fusion* 32: 109–119.
- Zagoruyko, S.; and Komodakis, N. 2015. Learning to compare image patches via convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- Zhang, Z.; Liu, X.; and Cui, Y. 2017. Multi-phase Offline Signature Verification System Using Deep Convolutional Generative Adversarial Networks. In *International Symposium on Computational Intelligence and Design*.
- Zois, E. N.; Alewijnse, L.; and Economou, G. 2016. Offline signature verification and quality characterization using poset-oriented grid features. *Pattern Recognition* 54: 162–177.
- Zois, E. N.; Papagiannopoulou, M.; Tsourounis, D.; and Economou, G. 2018. Hierarchical Dictionary Learning and Sparse Coding for Static Signature Verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*.
- Zois, E. N.; Theodorakopoulos, I.; and Economou, G. 2017. Offline Handwritten Signature Modeling and Verification Based on Archetypal Analysis. In *Proceedings of the IEEE International Conference on Computer Vision*.