A Stroke-based RNN for Writer-Independent Online Signature Verification

Chuang Li, Xing Zhang, Feng Lin, Zhiyong Wang, Jun'E Liu, Rui Zhang
China Financial Certification Authority (CFCA)
Beijing, China
{lichuang,zhangxing,linfeng,wangzhiyong,liujune,zhangrui}@cfca.com.cn

Haiqiang Wang
Tencent Media Lab
Shenzhen, China
tommyhqwang@tencent.com

***Abstract*—In the field of online handwritten signature verification, it is challenging to verify handwritten signature in a writer-independent scenario. In recent years, many researchers have been applying deep neural network methods to the signature verification task. However, these methods have not outperformed traditional methods, especially when the training samples are limited. In this paper, we propose a novel stroke-based bidirectional RNN architecture. The main idea is to split the signature into multiple patches using strokes. Concatenation of query and reference signature pairs are used as input. The proposed method uses two LSTM RNN networks to extract different features. The first one extracts the features of the strokes and the latter extracts the global features of the whole signatures. The results on the BiosecureID dataset demonstrate that our proposed method can reduce the EER by 33.05%, from 5.6% to 3.75% with fewer features and less training samples. Besides, we find that the proposed stroke based RNN network is 5x faster in training and testing time than Non stroke-based RNN network.**

***Index Terms*—online handwritten signature verification, writer-independent, recurrent neural network, Stroke-based**

I. Introduction

An automatic handwritten signature verification system aims to verify whether a particular signature truly belongs to an individual or not by utilizing the unique aspects of the signature. The system can be classified as either online (dynamic) [1], [2] or offline (static) [3] based on the format of the input signature data. An offline signature verification system works with static images. The signature is written on paper and scanned into a computer system. Thus only the shape of the signature is available to the system. By contrast, an online handwritten signature is acquired through dedicated devices, such as pen tablets, PDAs or handheld devices. The dynamic features are captured during the writing process, which include both the spatial and the dynamic information of a signature. Thus an online signature includes more information than an offline signature, such as time, pressure, pen up and down, azimuth, velocity, stroke order, etc. Due to the additional information, online systems achieve higher accuracy than offline ones [4]. Moreover, some pen tablets also detect the pen trajectory when the tip is not in contact with the writing surface, which is called the pen-up trajectory. It is shown [5] that pen-up trajectory information is useful to an online signature verification system and the lack of trajectories during pen-ups deteriorates verification performance. However, most mobile devices, such as smartphones and PDAs, cannot record the pen-up trajectory. In this paper, we focus on online signature verification without the pen-up trajectory.

Online signature verification systems can also be divided into writer-dependent [6], [7] and writer-independent [8] [9] based on the approach to train the model. In the writer-dependent case, a specific model is trained for each subject using signatures from that user. Whereas, a global model is built based on the whole training set in the writer-independent case. Thus the model is independent of writers and could be used to verify signatures from new writers without retraining. The writer-independent approaches are more flexible than writer-dependent ones: 1) It is inconvenient to ask a user to provide sufficient signature samples to train a personal writer-dependent model. However, a writer-independent model can be trained by using signatures from all users enrolled in the system. 2) The writer-independent system would not need to be retrained when a new user enrolls in. The convenience makes it suitable for real consumer-facing systems, e.g., a new client who opens an account in a bank.

A typical signature verification system calculates the similarity between enrolled signature and query signature, then judges whether the query signature is a forgery signature or a genuine signature by thresholding. The threshold can be writer-dependent or global. The former maintains a threshold for each writer and therefore requires each writer to enroll multiple samples. The latter uses a global threshold for the entire authentication system. Thus it is suitable for real life applications, especially when there is just one enrolled signature available for verification.

In this paper, we propose a stroke-based bidirectional RNN architecture to evaluate the similarity between enrolled signature and query signature. The system compares the similarity with a global threshold and decides whether the query signature is a forgery or genuine signature. Our main contributions are as follows: 1) We proposed a novel Stroke-based bidirectional RNN architecture that uses fewer samples yet gets higher accuracy. 2) The proposed stroke-based bidirectional RNN architecture can achieve higher accuracy with fewer features, which makes the model more suitable for deployment. 3) By conducting multi-datasets verification, we found that the proposed method has better generalization ability than other RNN structures.

The remaining part of the paper is organized as follows.

2379-2140/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDAR.2019.00090

Section II describes the related work on online handwritten signature verification. The proposed method is presented in the section III. Section IV introduces the dataset, experimental protocol and the evaluation results. Finally, section V gives concluding remarks.

II. Related Work

There are many approaches for online signature verification have been proposed in last decades. Broadly speaking, these approaches can be divided into two categories according to how the feature extraction is performed.

Feature-based approaches The main task of a feature-based approach is to select features that can describe a signature. Typically, a fixed-length feature vector derived from signature is used to determine whether it is a genuine or forgery signature. In [10], 100 global features have been proposed and sorted by their discrimination weight, such as average velocity, average pressure, etc. In [11], the number of strokes is used as a global feature. In [12], the occurrence of pen-ups and pen downs during signing along with average velocity and average pressure are selected as global features. Also, it is a common practice to apply some transformations on the signature for feature extraction. For example, a wavelet transform has been used [13] to extract a feature vector from the entire sample. In [14], the author applies DCT transform on the signature to obtain feature vectors.

Function-based approaches These approaches represent a signature as the time sequence and calculate the distance between two signatures to determine whether it is a genuine-genuine pair or genuine-forgery pair. There are multiple time functions [10] at each point to describe its features. The DTW is a very well-known algorithm for calculating the similarity between two time sequences with unequal lengths. It has been widely used in many studies [15], [16]. HMM (Hidden Markov Model) and GMM (Gaussian Mixture Model) are another two kinds of algorithms that have been used for signature verification in many works [17]–[21]. These two algorithms train a probabilistic model to calculate the probability that a query signature is genuine. The decision is made by comparing the probability with a threshold. In recent years, deep learning based approaches are getting more and more popular in signature verification because of their superior performance. An auto-encoder network and a siamese LSTM network [22] are proposed for signature verification. They also use the attention mechanism and apply down-sampling to improve the performance. Tolosana *et al.* [8] study the effectiveness of different kinds of RNN networks in the signature verification task.

III. Proposed Methods

We proposed a stroke-based bidirectional RNN architecture for online signature verification in this work. The system is illustrated in Figure 1. It consists of two cascaded bidirectional LSTM networks. The first network extracts the features of the strokes of enrolled signature and query signature and the second network calculates the overall features of the signature, respectively. The output of the network are two values. They are used to represent the dissimilarity degree between enrolled signature and query signature pair. A global threshold is used to determine whether the query signature is genuine or forgery.

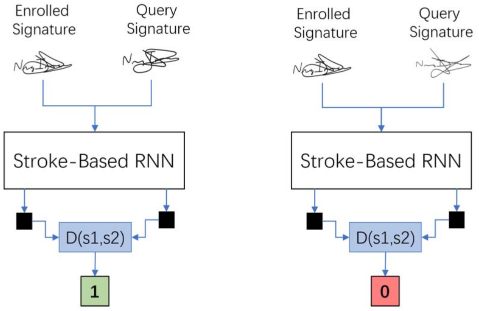

Fig. 1. Overview of proposed Stroke-based RNN. Two logits s1 and s2 are output of convolutional layers. Network output 1 and 0 represent the query signature is genuine and forgery, respectively.

$$D(s1, s2) = logit1 - logit2 \tag{1}$$

Forgery signature can be divided into random forgery and skilled forgery. Random forgery is the situation when an impostor, without any previous knowledge of a specific signature, tries to verify the identity of a signer by using his own genuine signature. However, the impostor knows the signature of the impersonated signer in the case of skilled forgery. The imposter is allowed to mimic the genuine signature. Thus the forgery is very similar to the impersonated signer. We extend the network from 2 logits to 3 logits in order to accommodate random forgery, i.e. there are 3 logits after the last fully connected layer. Then the dissimilarity can be evaluated by the following equation:

$$D(s1, s2) = logit1 + logit3 - logit2 \tag{2}$$

Intuitively, logit 1, 3 and 2 correspond to probability of skilled forgery(SF), random forgery(RF), and genuine signature(G), respectively. Thus, Eq.2 is a measurement of $prob(SF) + prob(RF) - prob(G)$. In the 2 logits network, Eq.1 is a measurement of $prob(SF) - prob(G)$. Both of them are distance measurement of input signature pairs. The 3 logits framework is effectively a generalized version of 2 logits network as random forgery might be the potential inputs. We experiment with 3 logits framework in this paper.

A. LSTM

RNN is widely used to process sequence data due to its ability to capture sequential relationships. However, RNN has only short-term memory due to the disappearance of the gradient [23], [24]. The long short-term memory(LSTM) [25] network is a variant of RNN. It combines short-term memory with long-term memory through subtle gate control and solves the problem of gradient disappearance to some extent. Each cell of LSTM consists of three gates: a forget gate f, an input gate i and an output gate o. The key to LSTM is the cell state

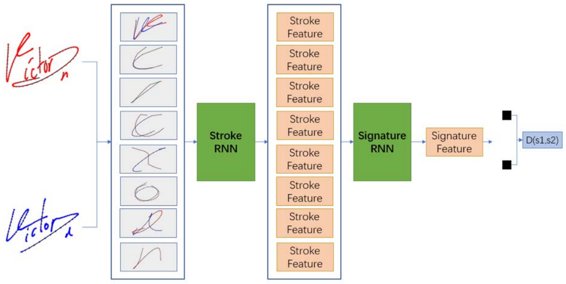

Fig. 2. Architecture of the proposed Stroke-based RNN. Patches of concatenated signature pair are input to Stroke RNN. The input to Signature RNN is the output of Stroke RNN. A global feature vector is extracted to measure the similarity of signature pair with fully concatenated layer.

C. The three gates are designed to protect and control cell state C.

B. Early Concatenated RNN

The idea of early concatenation is proposed in a network called 2-channel CNN [26]. It is shown to be more accurate in measuring the similarity between two signature images than Siamese [27], [28] networks. If there are two single-channel signature images with dimension (64, 64, 1), the concatenation involves combining them into a two-channel matrix (64, 64, 2) before feeding them into the network. Then the system evaluates the similarity metrics to determine whether they are from the same signer. We combine the concatenation technique with RNN and empirically find that the concatenation plays an important role in online signature verification. The system is more resistant to over-fitting, especially when the training data are limited.

C. Stroke-based RNN

The proposed stroke-based RNN network structure is shown in Figure 2. It consists of two RNN networks. The first one is called Stroke RNN. The input is the concatenation of enrolled signature and query signature. It contains two LSTM hidden layers with 150 and 100 memory blocks, respectively. After processing of these two hidden layers, a vector of stroke features is calculated and used as the input of the second RNN network. The second RNN, namely Signature RNN, extracts a global feature of two input signatures and outputs two logits representing the dissimilarity of input signature pair. The subtraction of the two logits is used to determine whether the two input signatures are genuine-genuine pair or genuine-forgery pair. A larger value indicates the two signatures are more likely to be a genuine-forgery pair. Signature RNN also consists of two LSTM hidden layers with 150 and 100 memory blocks, respectively. Moreover, there is a fully connected layer which uses Softmax as loss function. Detailed structure of the proposed network is given in Figure 3.

The LSTM networks used in Stroke-based RNN are all bidirectional. We share the weights of the forward cell and backward cell to reduce the number of parameters to train. The structure details of the proposed method are given in Table I.

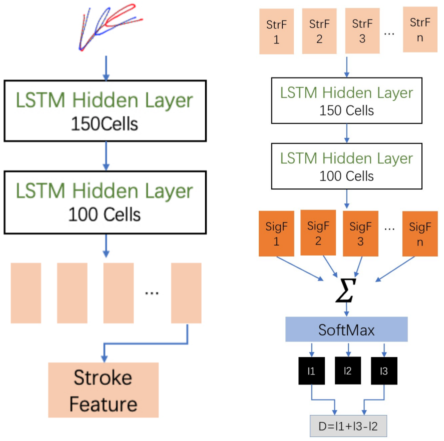

Fig. 3. Detailed structure of the proposed Stroke RNN and Signature RNN. StrF represents stroke features from Stroke RNN and SigF represents global features of the signature, respectively.

TABLE I
Architecture of the proposed Stroke-based RNN.

Type	remarks or drop rate	Input size
concatenate		2×20×400×23
bidirectional LSTM	150 cells	20×400×46
dropout	0.2	20×400×300
bidirectional LSTM	100 cells	20×400×300
dropout	0.6	20×200
bidirectional LSTM	150 cells	20×200
dropout	0.2	20×300
bidirectional LSTM	100 cells	20×300
dropout	0.7	200
fully connect		200
softmax	classifier	3

IV. Experimental Results

A. Datasets

There are several public datasets for online handwritten signature verification. The BiosecureID [29], MCYT-100 [30], SCUT-MMSIG [31] and MOBISIG [32] datasets are used to verify the performance of the proposed network.

BiosecureID The BiosecureID (400) dataset is no longer accessible to the public. Right now only a subset BiosecureID (132) that has signatures of 132 users is available. There are 16 original signatures and 12 skilled forgeries for each user. The signature was acquired in an office-like scenario. Users were asked to sign on paper with an inking pen, inside a grid that marked the valid signing space. The paper was placed on a Wacom Intuos 3 pen tablet that captured X and Y pen coordinates (resolution of 0.25 mm), pressure (1024 levels) and timestamp (100 Hz). In addition, pen-ups trajectories are also available.

TABLE II
Set of time functions used in this work.

#	Feature
1	x-coordinate: x_n
2	y-coordinate: y_n
3	Pen-pressure: z_n
4	Path-tangent angle: θ_n
5	Path velocity magnitude: v_n
6	Log curvature radius: ρ_n
7	Total acceleration magnitude: a_n
8-16	First-order derivate of features 1-7:$\dot{x_n}$,$\dot{y_n}$,$\dot{z_n}$,$\dot{\theta_n}$,$\dot{v_n}$,$\dot{\rho_n}$,$\dot{a_n}$
17	Ratio of the minimum over the maximum speed over a 5-samples window: v_n^r
18-19	Angel of consecutive samples and first order difference: α_n,$\dot{\alpha_n}$
20	Sine: s_n
21	Cosine:c_n
22	Stroke length to width ratio over a 5-samples window: r_n^5
23	Stroke length to width ratio over a 7-samples window: r_n^7

MCYT-100 The MCYT-100 dataset is a signature dataset collected by the BiDA Laboratory of the Autonomous University of Madrid. The acquisition device used is a WACOM tablet. During the process of writing, the tablet collects the following information: coordinates, time, pressure, and the declination of the pen. The x-axis coordinate and the y-axis coordinate range are 0 12700 and 0 9700, the pressure range is 0 1024, and the horizontal declination and vertical declination range are 0 3600 and 300 900, the sampling frequency is 100Hz, respectively. The dataset consists of signatures about 100 users and there are 25 real samples and 25 forged samples for each user.

SCUT-MMSIG The SCUT-MMSIG dataset was created by the South China University of Technology. The dataset includes signatures of 50 students. There are 20 real signature samples and 20 signature samples forged by the other four people for each student. It defines 5 verification protocols, namely, single-session, across-session (5), across-session (10), mixed-session and random forgery protocols.

MOBISIG The MOBISIG dataset contains signatures of 83 subjects: 49 men and 34 women. Data collection was performed using a Nexus 9 tablet. Each user has a dedicated folder that contains the 45 genuine signatures of the user and the 20 forgeries made by other users. Each signature was stored as a sequence of discrete values [x_t, y_t, t, p_t, fa_t, vx_t, vy_t, ax_t, ay_t, az_t, gx_t, gy_t, gz_t], where x_t, y_t are the coordinate values, t is the time stamp, p_t, fa_t are the pressure and finger area (these are normalized values [0, 1] and can be obtained through the standard Android API), vx_t, vy_t are the directional velocities, ax_t, ay_t, az_t are the directional acceleration of the device, and gx_t, gy_t, gz_t are the values obtained from the gyroscope sensor, respectively.

B. Preprocessing

1) Time Functions Representation
We used a set of 23 time functions [10], [33] that are extracted from the x, y coordinate and pen pressure value. All time functions are listed in Table II.

TABLE III
Details of experimental protocol on different datasets. PP and NP represent positive and negative pairs, respectively.

Dataset	Train(subjects)	Total(subjects)	PP	NP
BiosecureID	110	132	120	120
MCYT-100	85	100	600	600
SCUT-MMSIG	40	50	380	380
MOBISIG	70	85	1980	1980

2) Z-score Normalization
Considering that the data magnitudes of different devices may be different, we used Z-score transformation for the data normalization. Z-score is a commonly used normalization method. It converts the data into normal-distributed data with a mean of 0 and a standard deviation of 1. Its conversion function is:

$$z_i = (x_i - \mu)/\sigma \tag{3}$$

where μ and σ represen the mean and standard deviation of all sample data, respectively.

3) Construct Stroke Data
We need to segment each piece of stroke data from signature data and use it as input to Stroke-based RNN network. The point with a pressure of 0 is used as the separator of a stroke.

4) Data Alignment
The number of strokes segmented from the enrolled signature and query signature may be different and the number of points in each stroke may differ as well. So data alignment is necessary after the signature are segmented for early concatenated. We limit the maximum number of points per stroke to 400. It will be truncated if the number exceeds. On the other hand, it will be filled with 0. The maximum number of strokes per signature is limited to 20.

C. Experiment Protocol

Two types of forgeries are considered in this work: *Skilled Forgeries* and *Random Forgeries*. In order to train the network to verify both skilled and random forgeries, we have to construct three types of samples: genuine-genuine pairs, genuine and skilled-forgery pairs, genuine and random-forgery pairs. We will use the BisecureID dataset as an example to give more details about the construction of signature pairs.

There are signatures collected from 132 people in the BiosecurID dataset. We randomly take the signatures of 110 users as the training set and the signatures of the remaining 22 users as the test set, respectively. Each user has 16 original signatures and 12 skilled forgeries. Thus, there are $C_{16}^2 = 120$ genuine-genuine pairs and $16 * 12 = 192$ genuine-and-skilled-forgery pairs for each user. Then, we randomly selected five other users and constructed $16 * 5 * 16 = 1280$ genuine-and-random-forgery pairs. For the training state, a certain number of pairs need to be randomly removed so that the number of all three types of pairs is 120 to keep the samples balanced.

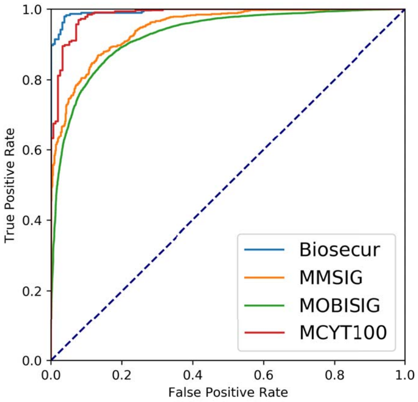

Fig. 4. ROC curves of the proposed method on BiosecurID, SCUT-MMSIG, MOBISIG and MCYT-100 datasets.

TABLE IV
Results of Stroke-based RNN network in terms of EER (%) on Biosecure dataset.

	EER (skilled)	EER (random)
DTW [34] [8]	10.17	**0.94**
Tolosana *et al.* [8]	5.6	5.2
Proposed Method	**3.74**	1.92

Therefore, we have $120 * 2 * 110 = 26400$ pairs for training and $(120 + 192 + 1280) * 22 = 35024$ pairs for testing. Table III shows the division of training set and test set on different datasets.

D. Evaluation Metrics

The Equal error rate (EER) is often used to measure the accuracy of a biometric system. The value indicates that the proportion of false acceptances is equal to the proportion of false rejections. It can be obtained by intersecting the ROC curve with a diagonal of the unit square. The lower the equal error rate value, the higher the accuracy of the biometric system. In this paper, EER is used to measure the accuracy of signature verification.

E. Implementation Details

The experiment was performed using Tensorflow 1.7, and the AdamOptimizer without batch normalization is used for training. The batch size is set to 24. The weights of the model are initialized according to Xavier and Yoshua's method [35] with the initialization bias value is equal to 0. The initial learning rate is set to be 2e-4 and momentum [36] with a decay of 0.9. The training was done using 3 TITAN X Pascal GPUs, and it takes 3.24 hours to train the model on the Biosecure dataset, which is about five times faster than other RNN architectures.

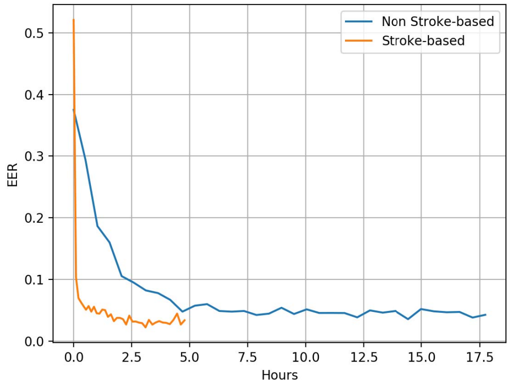

Fig. 5. Training performance on BiosecureID dataset using Stroke-based RNN and Non Stroke-based RNN. Stroke-based RNN converges faster and get better results than Non Stroke-based RNN.

TABLE V
Comparison of the proposed method with the state-of-the-art methods on various signature datasets in terms of EER (%).

Datasets	State-of-the-art Methods	EER (Skilled)	nV1
BiosecureID	Tolosana *et al.* [8]	5.6	1v1
	Proposed Method	**3.74**	1v1
MCYT-100	Moises *et al.* [37]	13.56	1v1
	Proposed Method	**10.46**	1v1
SCUT-MMSIG	Xinyi *et al.* [31]	20.66	1v1
	Proposed Method	**13.90**	1v1
MOBISIG	Antal*et al.* [32]	19.27	5v1
	Proposed Method	**16.08**	1v1

F. Results

The verification performance in terms of EER on the BiosecureID dataset using proposed Stroke-based RNN network is shown in Table IV. Two different types of forgery are considered: skilled and random. To the best of our knowledge, the best result on the BiosecureID dataset was reported in [8]: EER of skilled forgery is 5.6%. We achieved much better performance with an EER of 3.74%. For random forgery cases, our method still got better result than [8]. However, traditional DTW based method [34] remains state-of-the-art.

In order to evaluate the generalization ability of the proposed network, we also performed experiments on MCYT-100, SCUT-MMSIG and MOBISIG datasets. Table V illustrates the results on different datasets in terms of EER. The results of state-of-the-art methods are also listed for comparison. It can be seen that Stroke-based RNN achieved the best results on all datasets. It is worth to mention that some other report 5v1 results rather than 1v1 results. 5v1 means 5 reference signature are used during the verification stage and 1v1 uses only 1 reference signature. Even though, the proposed method outperforms other 5v1 approaches on the MOBISIG dataset.

G. Efficiency

The stroke-based RNN is much faster than former deep learning methods. We trained two models on the BiosecureID dataset. The model for comparison is called Non Stroke-based RNN, which is a 3 layers stacked LSTM RNN architecture

TABLE VI
Comparison of running time and performance. S and NS represent Stroke-based RNN and Non Stroke-based RNN, espectively. Batch, Total and Steps columns indicate the time cost of a batch, total training time and number of training steps.

Dataset			EER(%)	Batch	Total	Steps
BiosecurID	Training	S	-	$3.93s$	$3.24h$	2500
		NS	-	$21.46s$	$17.74h$	3300
	Prediction	S	3.749	$0.97s$	$133.12s$	-
		NS	4.513	$6.5s$	$885.01s$	-
MCYT-100	Training	S	-	$3.69s$	$8.18h$	7000
		NS	-	$15.57s$	$48.88h$	9900
	Prediction	S	10.462	$0.76s$	$341.79s$	-
		NS	11.090	$5.09s$	$2268.53s$	-
SCUT-MMSIG	Training	S	-	$2.67s$	$1.53h$	1800
		NS	-	$14.35s$	$10.97h$	2500
	Prediction	S	13.906	$0.60s$	34.87	-
		NS	16.921	$4.83s$	$285.15s$	-
MOBISIG	Training	S	-	$3.01s$	$3.23h$	3400
		NS	-	$15.55s$	$20.03h$	4200
	Prediction	S	16.086	$0.685s$	$493.18s$	-
		NS	16.261	$4.96s$	$3570.96s$	-

with 100-100-50 memory cells. The signature data to Stroke-based RNN is segmented by strokes, the input of Non Stroke-based RNN is original signature data that is not segmented. Using Stroke-based RNN for training is five times faster than Non Stroke-based RNN with better performance. Figure 5 shows the training performance of Stroke-based RNN and Non Stroke-based RNN on the BiosecureID dataset. The comparison of running time on 4 databases are shown in Table VI.

V. Conclusion

In this paper, we proposed a novel network model called Stroke-based RNN for writer-independent online signature verification. The proposed approach involves signature segmentation using strokes, early concatenation of signature pair and deeper RNN architecture. It is shown that the proposed method could significantly improve the performance on the task. The EER of skilled forgery on the BiosecureID dataset is 3.74% which is 33.05% lower than the state-of-the-art. What's more, the proposed stroke based RNN network is 5 times faster than plain RNN network.

References

[1] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 1, pp. 63–84, 2000.

[2] A. Parziale, M. Diaz, M. A. Ferrer, and A. Marcelli, "Sm-dtw: Stability modulated dynamic time warping for signature verification," *Pattern Recognition Letters*, vol. 121, pp. 113–122, 2019.

[3] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in neural information processing systems*, 2009, pp. 545–552.

[4] N. Aqili, A. Maazouzi, M. Raji, A. Jilbab, S. Chaouki, and A. Hammouch, "On-line signature verification using point pattern matching algorithm," in *2016 International Conference on Electrical and Information Technologies (ICEIT)*. IEEE, 2016, pp. 410–413.

[5] M. Martinez-Diaz, J. Fierrez, R. P. Krish, and J. Galbally, "Mobile signature verification: Feature robustness and performance comparison," *IET Biometrics*, vol. 3, no. 4, pp. 267–277, 2014.

[6] L. Yang, Y. Cheng, X. Wang, and Q. Liu, "Online handwritten signature verification using feature weighting algorithm relief," *Soft Computing*, vol. 22, no. 23, pp. 7811–7823, 2018.

[7] M. Diaz, M. A. Ferrer, D. Impedovo, M. I. Malik, G. Pirlo, and R. Plamondon, "A perspective analysis of handwritten signature technology," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, p. 117, 2019.

[8] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia, "Exploring recurrent neural networks for on-line handwritten signature biometrics," *IEEE Access*, vol. 6, pp. 5128–5138, 2018.

[9] K. Ahrabian and B. Babaali, "Usage of autoencoders and siamese networks for online handwritten signature verification," *Neural Computing and Applications*, pp. 1–14, 2017.

[10] J. Fierrez-Aguilar, L. Nanni, J. Lopez-Peñalba, J. Ortega-Garcia, and D. Maltoni, "An on-line signature verification system based on fusion of local and global information," in *International Conference on Audio- and Video-Based Biometric Person Authentication*. Springer, 2005, pp. 523–532.

[11] A. K. Jain, F. D. Griess, and S. D. Connell, "On-line signature verification," *Pattern recognition*, vol. 35, no. 12, pp. 2963–2972, 2002.

[12] L. L. Lee, T. Berger, and E. Aviczer, "Reliable on-line human signature verification systems," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 643–647, 1996.

[13] D. Z. Lejtman and S. E. George, "On-line handwritten signature verification using wavelets and back-propagation neural networks," in *Proceedings of Sixth International Conference on Document Analysis and Recognition*. IEEE, 2001, pp. 992–996.

[14] Y. Liu, Z. Yang, and L. Yang, "Online signature verification based on DCT and sparse representation," *IEEE transactions on cybernetics*, vol. 45, no. 11, pp. 2498–2511, 2014.

[15] A. Kholmatov and B. Yanikoglu, "Identity authentication using improved online signature verification method," *Pattern recognition letters*, vol. 26, no. 15, pp. 2400–2408, 2005.

[16] C. Vivaracho-Pascual, M. Faundez-Zanuy, and J. M. Pascual, "An efficient low cost approach for on-line signature recognition based on length normalization and fractional distances," *Pattern Recognition*, vol. 42, no. 1, pp. 183–193, 2009.

[17] J. Fierrez, J. Ortega-Garcia, D. Ramos, and J. Gonzalez-Rodriguez, "HMM-based on-line signature verification: Feature extraction and signature modeling," *Pattern recognition letters*, vol. 28, no. 16, pp. 2325–2334, 2007.

[18] E. A. Rua and J. L. A. Castro, "Online signature verification based on generative models," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 4, pp. 1231–1242, 2012.

[19] B. L. Van, S. Garcia-Salicetti, and B. Dorizzi, "On using the viterbi path along with HMM likelihood information for online signature verification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 5, pp. 1237–1247, 2007.

[20] A. Humm, J. Hennebert, and R. Ingold, "Gaussian mixture models for chasm signature verification," in *International Workshop on Machine Learning for Multimodal Interaction*. Springer, 2006, pp. 102–113.

[21] O. Miguel-Hurtado, L. Mengibar-Pozo, M. G. Lorenz, and J. Liu-Jimenez, "On-line signature verification by dynamic time warping and gaussian mixture models," in *2007 41st Annual IEEE International Carnahan Conference on Security Technology*. IEEE, 2007, pp. 23–29.

[22] K. Ahrabian and B. Babaali, "Usage of autoencoders and siamese networks for online handwritten signature verification," *Neural Computing and Applications*, pp. 1–14, 2017.

[23] Y. Bengio, P. Frasconi, and P. Simard, "The problem of learning long-term dependencies in recurrent networks," in *IEEE international conference on neural networks*. IEEE, 1993, pp. 1183–1188.

[24] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, 2013, pp. 1310–1318.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4353–4361.

[27] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Advances in neural information processing systems*, 1994, pp. 737–744.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[29] J. Fierrez, J. Galbally, J. Ortega-Garcia, M. R. Freire, F. Alonso-Fernandez, D. Ramos, D. T. Toledano, J. Gonzalez-Rodriguez, J. A. Siguenza, J. Garrido-Salas *et al.*, "BiosecurID: a multimodal biometric database," *Pattern Analysis and Applications*, vol. 13, no. 2, pp. 235–246, 2010.

[30] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.-J. Igarza, C. Vivaracho *et al.*, "MCYT baseline corpus: a bimodal biometric database," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 150, no. 6, pp. 395–401, 2003.

[31] X. Lu, Y. Fang, W. Kang, Z. Wang, and D. D. Feng, "SCUT-MMSIG: a multimodal online signature database," in *Chinese Conference on Biometric Recognition*. Springer, 2017, pp. 729–738.

[32] M. Antal, L. Z. Szabó, and T. Tordai, "Online signature verification on MOBISIG finger-drawn signature corpus," *Mobile Information Systems*, vol. 2018, 2018.

[33] M. Martinez-Diaz, J. Fierrez, and S. Hangai, "Signature features," *Encyclopedia of biometrics*, pp. 1375–1382, 2015.

[34] M. Gomez-Barrero, J. Galbally, J. Fierrez, J. Ortega-Garcia, and R. Plamondon, "Enhanced on-line signature verification based on skilled forgery detection using sigma-lognormal features," in *2015 International Conference on Biometrics (ICB)*. IEEE, 2015, pp. 501–506.

[35] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[36] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147.

[37] M. Diaz, A. Fischer, M. A. Ferrer, and R. Plamondon, "Dynamic signature verification system based on one real signature," *IEEE transactions on cybernetics*, vol. 48, no. 1, pp. 228–239, 2016.