2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS) | 979-8-3503-3071-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICPADS60453.2023.00291

Deep Frame-Point Sequence Consistent Network for Handwriting Trajectory Recovery

1st Yu-Jie Xiong
Shanghai University of Engineering Science
Shanghai, China

2nd Yu-Fan Dai
Shanghai University of Engineering Science
Shanghai, China

3rd Dan Meng*
OPPO Research Institute
Shenzhen, China

***Abstract*—Intelligent Cyber-Physical Systems relies heavily on data for real-time monitoring, analysis, and control of physical systems. By converting offline handwriting into online handwriting, it provides ICPS with more diverse and abundant input data, enriching the variety and quantity of available information. Based on the acquisition approach, there are two kinds of handwriting data: online and offline data. Generally, online data which contains pen trajectory of static character, has more advantages than offline data in terms of character recognition and analysis. Due to limited means of acquiring online data, inferring from offline data is an attractive approach. In this paper we introduce a novel framework to recover handwriting trajectory from single static character image. The trajectory can be represented by two types of sequence: points sequence and frame sequence. Therefore, we design two streams: points sequence prediction stream and frame sequence prediction stream, based on the encoder-decoder structure. We combine the two streams by a novel sequence consistent module to synchronize the training process. With the two streams and their complementary advantages, our methods can predict trajectory with both high spatial and temporal accuracy. Extensive experiments demonstrate the effectiveness of our network through qualitative and quantitative comparison.**

***Index Terms*—sequence consistent, frame-Point sequence, handwriting trajectory recovery**

I. Introduction

Handwriting recognition and analysis has been a subject of intense research over the last two decades [1]. According to sampling devices, handwritten character data can be classified into two categories: offline and online data. Offline data refers to static images captured by a scanner of a camera. On the other hand, online data are points sequence sampled by custom acquisition devices such as electrostatic or electromagnetic tablets. The sequential data record coordinate information of pen tip moving trajectory of handwriting. Comparing to static images, online data provides additional dynamic information, which makes handwriting character recognition tasks much easier. However due to the limited application of such special sampling devices, acquiring online data is much more difficult than offline images. Therefore if we can recover the stroke trajectory from static 2D images, the offline recognition problem could be transferred to online recognition. Achieving this would not only bridge the gap between offline and online recognition but also hold profound implications for enhancing the capabilities of Intelligent Cyber-Physical Systems [2], [3].

In recent years, with the blooming of deep learning, convolutional neural network (CNN) has been widely applied for handwriting recognition. A number of CNN based models [4], [5] were proposed for offline character recognition. As for online data, the input change to point sequence. Thus recurrent neural network (RNN) based models were utilized to handle sequential information [6], [7]. In comparison, online handwriting recognition generally performs much better in terms of precision and robustness to noise. Especially, for Chinese characters, online data has intrinsic advantage. Different from alphabetic letters, Chinese characters are composed of many disconnected strokes. Moreover, different characters have distinct stroke order, defined in the dictionary. Hence the sequential points provide valuable stroke order information, which can highly boost recognition performance. Besides, images sampled in real scenes are noisy and corrupted, which complicates training. Online data which consists of trajectory sequence does not have those problems. With a model trained on online data, character recognition will become more robust and applicable in many real scenes.

In order to recover trajectory information from static images, some RNN based frameworks were proposed to predict point sequence. In [8], the authors designed an encoder-decoder short term memory networks (LSTM) network to solve this problem by introducing two networks. An encoder network is used to encode the extracted image feature sequence to a hidden representation, and a decoder network takes the representation and sequentially predicts the data points. In this way, the two-dimensional image matrix can be converted to one-dimensional sequence which consists of successive coordinate positions of the pen-tip. However this method can only recover the trajectory within a single stroke. Multiple strokes of one character could not be separated. Besides due to the simple L1 distance loss used in this framework, the output coordinate from decoder may deviate from the actual skeleton of original images.

On the other hand, instead of predicting points coordinate directly, we can also generate frame sequence of strokes. With the recovered sequence, we can obtain the difference image between each continuous frame, and extract point coordinates from it. Recently, researchers propose to decompose the motion and content flow to effectively handle complex evolution of pixels. In [9], the authors designed a content

2690-5965/23/$31.00 ©2023 IEEE
DOI 10.1109/ICPADS60453.2023.00291

bridge module (CBM), which contains two units taking charge of representation flow and temporal flow respectively. By focusing on representation and temporal information separately, training process can be eased. After the two units, a merge unit is applied to synthesize the two flows. Based on this work, we applied deep RNN network to predict trajectory sequence frames. This method is able to successfully generate handwriting frames with high correlation to the actual skeleton. However sometimes the temporal information can not be accurately captured, due to the reason that image generation loss functions lack the ability to constrain the temporal variation of stroke sequence.

We observe that both point sequence and image sequence have their pros and cons. Point prediction stream gives clear motion flow, but lacks the ability to preserve character shape information. Image prediction stream performs well in generating good shape, but produces vague temporal flow. This situation is explainable concerning these two intrinsic data forms. For point sequence, the points are connected to each other in a sequential modeling, giving explicit temporal information. As to frame sequence, there is no sequential connection between successive images. The loss function works on inner-frame level, instead of inter-frame. Thus the temporal constrain is implicit. To take advantage of both networks to resolve this problem, we propose a deep frame-point sequence consistent network to combine both streams. We design a new module, named **frame-point sequence consistent module (FPSCM)**, in which the two streams share the same hidden state. In this way, point and image stream will constrain each other and keep consistent for spatial representation and temporal information. We will demonstrate the effectiveness of our network in the experiment section.

In summary, in this paper, we propose a novel two-stream-style framework which keeps inner consistency to deal with handwriting trajectory problems. Our contribution includes: 1) a new network to predict trajectory frames from a single character image; 2) a novel frame-point sequence consistent module to synchronize two stream training. Extensive experiments demonstrate the effectiveness of our network by quality and quantity comparison.

II. Related Work

A. Handwriting Analysis and Understanding

For the task of handwriting analysis and understanding, there are two main categories of methods: offline and online approaches. Offline character recognition is a specific task of image classification. Traditional methods use low level features to encode characters or words. Almazan et al. [10] encodes the input word image as fisher vectors (FV), i.e., as an aggregation of the gradients of a Gaussian mixture model (GMM) over a few low-level descriptors, such as SIFT. Then a set of linear SVM classifiers are trained for binary attribute classification. Fang et al. [11] proposed a handwriting recognition authentication scheme named HandiText based on behavior and biometrics features. Xiong et al. [12] introduced an effective attention module into multipule siamese network (MSN) to extract discriminative information from offline handwritten signatures.

Online handwriting data are composed of point sequences of pen tips. Since the offline characters are naturally represented as scanned images, a number of papers tried to apply CNN to online characters. However, in this way, the online handwriting trajectory should firstly be transformed to static image representations, such as the path signature maps [13] and the directional feature maps [14]. Whereas for sequence input, it is more reasonable to use RNN based network. Wang et al. [15] also proposed a new method of online Tibetan handwriting sample generation based on component combination for large character set, in which linear discriminate analysis (LDA) is used to reduce the dimension of features. Zhang et al. [16] focused on Chinese characters which are composed of multiple strokes. The authors applied gated recurrent unit (GRU) to replace LSTM. They also proposed a conditional generative model to automatically draw recognizable Chinese characters.

B. Sequence Prediction

The problem of future sequence prediction has received growing interest in computer vision. Chen et al. [17] represented 3D structures by 2D maps of pairwise residue distances and developed a new method to predict protein sequence profiles based on an image captioning learning frame. Balderas et al. [18] evaluated the combination of artificial neural networks (ANNs) into two novel algorithms developed with the aim of improving image sequence prediction. Sharma et al. [19] constructed a novel framework that employs a neural image compressor to preserve the spatial relationships between patches and generate a compressed representation of the whole-slide image, and a customized deep-learning regressor to predict RNA sequence from the compressed representation by learning both global and local features. ProtConv [20] employed convolutional neural network to predict the functionality of proteins by converting the amino-acid sequences to a two dimensional image. In 2017, Villegas et al. [21] proposed to decompose spatial layout and temporal dynamics. By independently modeling motion and content, predicting next frame reduces to converting the extracted content feature into the next frame content by identified motion features, which simplifies the task. Pang et al. [9] designed a content bridge module (CBM) which splits the temporal flow and spatial representation flow. By balancing these two directions, it is easier to train the network when building deep.

III. Methodology

Our task is to recover the trajectory that finally generates the character image, but learning temporal information from static image is a challenging task. As a result, we need the help of carefully designed data structure that suits for a powerful sequence prediction model, which is able to capture

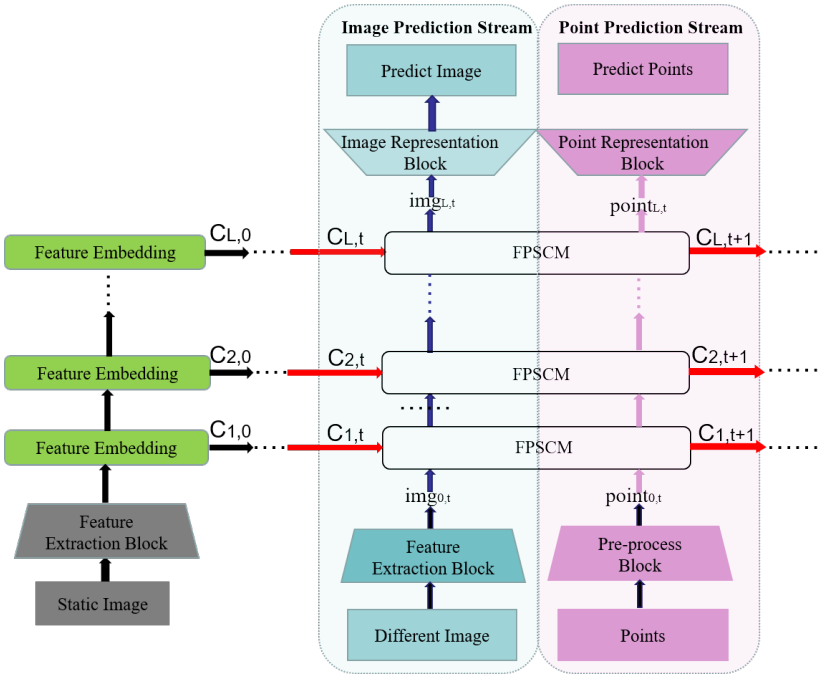


Fig. 1. Deep frame-point sequence consistent network. The proposed network uses image features as initial states of FPSCM. The image prediction stream and point prediction stream take in different inputs but are trained simultaneously. FPSCM is introduced to synchronize the two streams.

both temporal and content information at the same time. In the following section, we firstly introduce two streams data structure in Section 3.1. Then we detail our proposed frame-point sequence consistent module (FPSCM) in Section 3.2. Finally, we briefly describe the method of weights initializing for the proposed frame-point sequence consistent network in Section 3.3.

A. Two-Stream Data Structure

We design two-stream data structure, namely image prediction stream and point prediction stream, for the proposed model as shown in Figure 1. During the training time, the input of the image prediction stream is the difference images, while the input of the point prediction stream is the start and end point coordinates, with pen tip states of each segment in the difference image respectively. The target output of the two streams is the ground truth difference image frames and start/end points sequence. Consequently, we use MSE loss for image prediction stream, L1 regression loss and softmax classification loss for point prediction stream. During the testing time, we predict image frame and point coordinate for the time $t+1$ based on the previous t timestamps.

1) Image Prediction Stream.: It is easy to come up with an idea of solving the trajectory recovery problem as a video frame sequence prediction problem, i.e. we can try to recover image frame sequence that finally generating such static character image. It can be realized by sending image frame sequence into a RNN style network. So we design an image prediction stream to fulfill this task based on the architecture of deep RNN network [9].

During training time, in order to better capture the motion variation, we use difference image of neighboring frame $D = [diff\_img_1, diff\_img_2, \cdots, diff\_img_N]$ as the input of image prediction stream.

As shown in Figure 1, at timestamp t, the input different image is $diff\_img_t$, and after two convolutional and max pooling layers, we have:

$$img_{0,t} = Pool(R(diff^{'}\_img_t; W_{diff^{'}\_img_t})),$$
$$diff^{'}\_img_t = Pool(R(diff\_img_t; W_{diff\_img_t})).$$

where R is the "R" unit detailed in Section 3.2, $W_{diff^{'}\_img_t}$ and $W_{diff\_img_t}$ are the parameters of R, and Pool means max pooling. With $img_{0,t}$ as the input of deep RNN network at time t, we have $img_{L,t}$ when the depth of deep RNN network is L. After the image representation block, we finally get the predicted image $\hat{diff}\_img_{t+1}$ for timestamp $t+1$: $\hat{diff}\_img_{t+1} = f^{deconv}(img_{L,t})$, where $f^{deconv}$is composed of stacked up-pooling and deconvolutional layers, and the predicted image $\hat{diff}\_img_{t+1}$ has the same size of $diff\_img_t$. MSE loss is used for image prediction stream: $L_{img} = \frac{1}{N-1}\sum_{t=1}^{N-1} \|\hat{diff}\_img_{t+1} - diff\_img_{t+1}\|^2$.

2) Point Prediction Stream.: The static character image is generated by concatenating a sequence of point coordinates and pen tip states, which could be seen as a matrix $P = [p_1, p_2, \cdots, p_{N'}] \in R^{3\times N'}$. In $P, p_t = (x_t, y_t, s_t), t \in$

$[1, 2, \cdots, N']$ is a tuple, meaning point coordinates and pen tip state at time t. In each tuple, $x_t \in [0, w], y_t \in [0, h]$, where w and h are the width and height of the character image. $s_t = 1, 2, 3$ means pen-down (pen is drawing on paper), pen-up (pen is leaving from paper) and end-of-char. We can further transfer s_t into one-hot style $\tilde{s_t}$:

$$\tilde{s_t} = \begin{cases} (1,0,0), & pen-down \\ (0,1,0), & pen-up \\ (0,0,1), & end-of-char \end{cases} \quad (1)$$

Given the point coordinates and states of timestamps t, the point prediction stream gives prediction of next t + 1 timestamp . As shown in Figure 1, image and point prediction stream shares the same temporal flow by FPSCM. In Section 3.2, unit T is designed as a convolutional layer with activation function. As a result, we should convert the input of point prediction stream in_cort into the same shape with $img_{0,t}$ in the image prediction stream (denoted as pre-process block in point prediction stream), where

$$\begin{aligned} in\_cor_t &= [\text{start}_t, end_t] \\ &= \begin{cases} [(x_t, y_t), (x_{t+1}, y_{t+1})] & if \quad s_t = 1, \\ [(x_{t+1}, y_{t+1}), (x_{t+2}, y_{t+2})] & otherwise. \end{cases} \end{aligned} \quad (2)$$

We also have the pen tip states in_state_t for in_cor_t:

$$\begin{aligned} in\_state_t &= \left[start_t^{state}, end_t^{state}\right] \\ &= \begin{cases} [\tilde{s}_t, \tilde{s}_{t+1}] & if \quad s_t = 1, \\ [\tilde{s}_{t+1}, \tilde{s}_{t+2}] & otherwise. \end{cases} \end{aligned} \quad (3)$$

Figure 2 details the converting process of getting $point_{0,t}$, which is the same size of $img_{0,t}$ in image prediction stream. This converting procedure is indispensable, firstly it provides feature maps with the same shape of image prediction stream. Secondly, it forces the deep RNN network to focus on the start and end point of current different images in image prediction stream with the help of FPSCM. This indicates that point prediction stream plays the role of attention mechanism.

B. Frame-Point Sequence Consistent Module

To model image and points sequence inputs, we need to make sure that they can be trained efficiently and are in consistent with each other along the temporal flow. At each timestamp, we have a multi-modal data stream: image frame and points sequence. If we handle image frame and points sequence separately, it is hard to keep the consistency of image and point stream in temporal flow, especially during the testing phrase. This may caused by fact that there is no communication mechanism that synchronize the hidden states of these two streams. Therefore, we start to consider how to break the communication gap between these two streams, through which image frame and points sequence can complement each other and the training process can converge. Inspired by CBM [9], we design FPSCM to make sure the temporal coherency between these two streams.

As mentioned before, CBM simplifies the training process when building deep by balancing temporal flow and spatial representation flow. However, in our scenario, we expect not only to capture both temporal and spatial information, but also to learn these information from image frame and points sequence at the same time. So the architecture of our proposed FPSCM is shown in Figure 3, in which there are two "representation" units R and one "temporal" unit T. R units can be seen as content feature extractor for image prediction stream and point prediction stream respectively, and T acts as sequence consistent supervisor on the sequential dimension. With sequence consistent unit T, parameters in content representation flow and temporal flow can be co-updated. Besides, R unit in image and point stream is treated as a content representation bridge over the temporal information. In this way, the two streams can be synchronized and trained simultaneously by sharing the same T unit.

In the i^{th} layer of deep RNN network ($i \in [1, 2, ..., L]$) at timestamp t, we denote $img_{i-1,t}$ and $point_{i-1,t}$ as the input of image and point prediction stream respectively, then we have:

$$img'_{i,t} = R\left(img_{i-1,t}; W_{img_i}\right),$$
$$\text{point }'_{i,t} = R\left(\text{ point }_{i-1,t}; W_{\text{point }_i}\right).$$

where $R = ReLU(conv(\cdot)), W_{img_i}$and W_{point_i} are the parameters of R in i^{th} layer of image and point stream. The sequence consistent information calculated by unit T can be represented as: $c_{i,t} = T\left(c_{i,t-1}, \text{ img}_{i-1,t}, \text{ point}_{i-1,t}; W_{t_i}\right)$, where $T = sigmoid(conv(\cdot)), c_{i,t}$ is the memory state on temporal flow in i^{th} layer at timestamp t, and W_{t_i} is the parameters of T in i^{th} layer.

Finally, in order to keep frame-point sequence coherence, we use unit M to merge information from image frame and points sequence as:

$$img_{i,t} = M(img'_{i,t}, c_{i,t}),$$
$$point_{i,t} = M(point'_{i,t}, c_{i,t}).$$

where M is the element-wise production, $img_{i,t}$ and $point_{i,t}$ are the output of image and point prediction stream in i^{th} layer at time t step. For each timestamp t, we stack several FPSCMs to construct a deep RNN network.

C. Initializing Weights for the Proposed Network

It is hard to learn temporal information from a static image or its feature representation, but we can treat feature representation as an embedding vector and encode it into a coarse temporal sequence. This kind of coarse temporal sequence contains right-to-left and top-to-down sequential information of a static character image, since each element in the embedding vector can be mapped to a particular rectangular part of the image. Due to the reason that convolutional neural network (CNN) can adapt its weights to obtain robust feature representations, we use a deep CNN to learn feature representation from the input character image directly. The deep CNN is composed of eight convolutional layers with

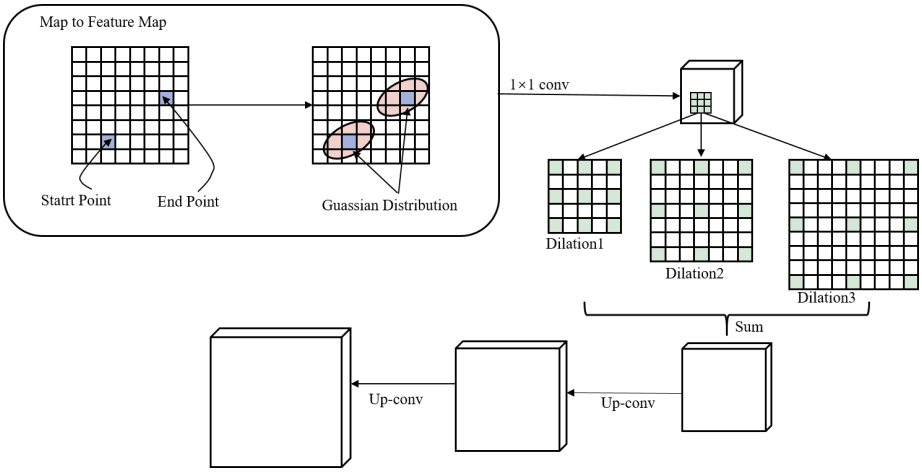


Fig. 2. Pre-process of point prediction stream. Input points are transferred into feature maps with the same size of $img_{0,t}$ in image prediction stream. The start and end points are first mapped into a 8 × 8 matrix, and then Gaussian distribution is applied in the mapped points' 3×3 neighborhoods. After that, a 1×1 convolutional layer is followed, cascading a multi-dilation layer with dilation rates ranging from 1 to 3. With the summation of the dilated feature maps, two up convolutional layers are used to complete the feature maps with the same size of the input in image prediction stream.

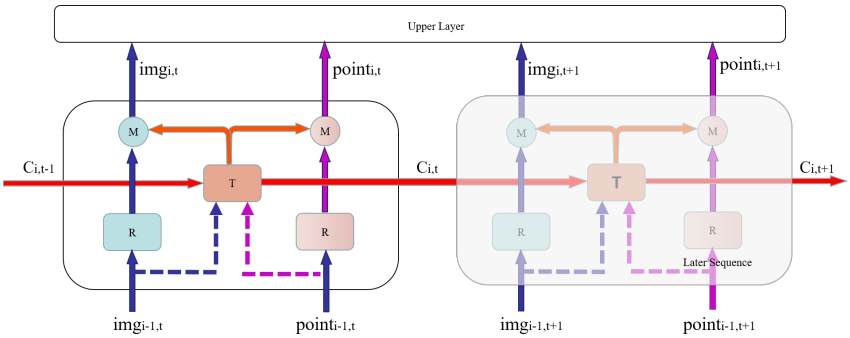


Fig. 3. Frame-point sequence consistent module. FPSCM is designed to guarantee the coherence between image and point prediction streams. The red lines represent temporal flows, the blue lines and purple lines are the representation of image and point stream respectively. R, T , and M denote content feature extractor, sequence consistent unit, and merge function respectively. The dashed blue and purple line means feeding forward is allowed but back-propagation is inhibited with a certain probability, denoted as temporal dropout.

kernel size of 3 ×3. Rectified linear unit (ReLU) and max pooling are applied after all except the $3^{rd}, 6^{th}$, and 8^{th} convolutional layers, while batch norm layer is employed after only these three layers to speed up the converge process during training stage. For an input character image, after feature extraction block, a 1×2 pooling layer is used to yield long sequence vector. This long sequence vector is further sent into a stacked multi-layer LSTM network for feature embedding, which generates the initial states $C_{i,0}, i \in [1, L]$ for FPSCMs.

IV. EXPERIMENT

The feature extraction network takes input images of 64 × 64, and LSTM cells of 512 hidden units are stacked in multi-layer LSTM network. There are five layers in both multi-layer LSTM network and deep RNN network, which means $L = 5$. We set $\alpha = 0.5, \beta = 0.5$, and weights for s_1, s_2, s_3 are 1, 20, 100.

A. Dataset

Training data plays an important role in deep learning based framework in terms of data quality and quantity. To the best of our knowledge, there exists no such dataset with sufficient number of data containing trajectory information and corresponding offline images. Online data contains sampled point coordinates through writing time, and we can convert online data to its offline images. Hence, we have large amount number of data to train and test our proposed network. We use OLHWDB1.1 [22], UNIPEN [23], and LIPI Toolkit

dataset[1] to evaluate the performance and generalization of our proposed network.

The OLHWDB1.1 dataset is a Chinese online handwriting dataset, including 3755 characters in the GB2312-80 level-1 set written by 240 people. UNIPEN contains English characters (lowercase and uppercase), and digits. LIPI Toolkit dataset includes Tamil, Telugu, and Devanagari characters. Since the number of the mentioned datasets is huge, we only select a small subset (around 10,000 samples) for training, and the other 1,000 samples for testing.

B. Evaluation Matrix

Suppose the number of testing set is denoted as N_{test}, we measure the performance of the proposed network based on two evaluation metrics:

(1) Start Point (SP) accuracy: For every handwriting character, there is only one start point. So if the network can predict the start point correctly of the offline image, we regard it as a positive result, otherwise negative. As a result, we can calculate the start point accuracy as:

$$SP = \frac{\text{Number of images with correct SP prediction}}{N_{\text{test}}}. \quad (4)$$

(2) End Point (EP) accuracy: Similar to SP, for every offline image, there is only one end point. Hence we can calculate the end point result as follows:

$$EP = \frac{\text{Number of images with correct EP prediction}}{N_{\text{test}}}. \quad (5)$$

C. Baselines

We compare our FPSCM with two baselines:

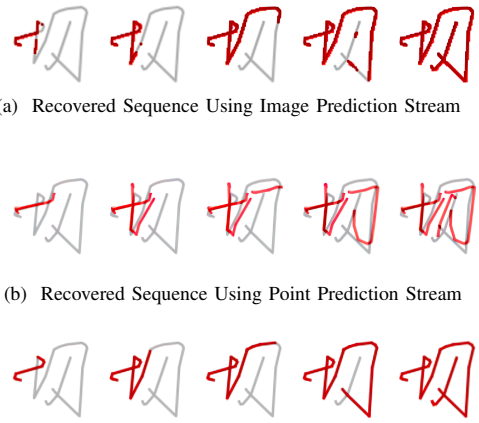
(1) SSM-MCE [24]: This approach develops compact classifiers using deep neural networks for online handwritten Chinese character recognition.

(2) ATR-CNN [25]: This method extends CNN model for offline handwritten Chinese character classification.

D. Results and Analysis

1) Impact of Different Components in Deep Frame-Point Sequence Consistent Network.: From Figure 4, we can easily draw the conclusion that image prediction stream has the superior ability in capturing the shape of the character image, while inferior in memorizing the temporal information. On the other hand, point prediction stream has advantages in terms of temporal flow, and disadvantages in the aspects of keeping character's shape. The reason can be traced down to the data structure and loss function of image stream and point stream. As well known, image frame covers rich content information, and point sequence carries temporal information in an explicit way. In order to complement one another, we combine data streams from image frame and points sequence. However, we find the model can hardly converge since there is no mechanism to co-update parameters of hidden states along the temporal flow. FPSCM is specially designed to solve this problem, and experiment results show that multi-modal data sequence can yield satisfied results by sharing the hidden states' parameters along temporal flow.

[1]http://lipitk.sourceforge.net/hpl-datasets.htm



(a) Recovered Sequence Using Image Prediction Stream

(b) Recovered Sequence Using Point Prediction Stream

(c) Recovered Sequence Using Both Stream with FPSCM

Fig. 4. Comparison of prediction results from different streams. Above are recovered sequence using (a) image prediction stream only, (b) point prediction stream only, and (c) image and point prediction stream with FPSCM.

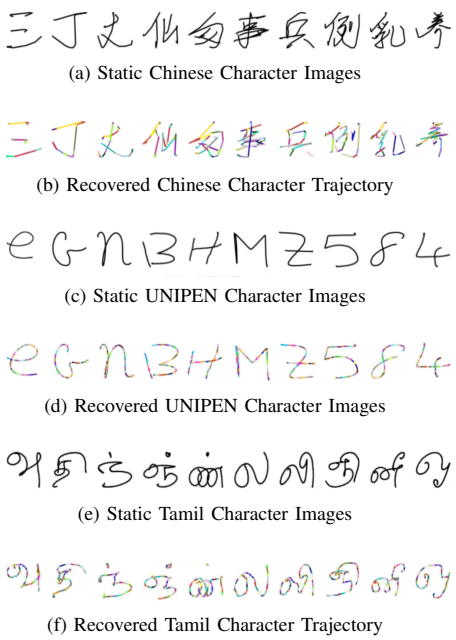
TABLE I
EVALUATING THE GENERALIZATION ABILITY OF THE PROPOSED MODEL.

Training Dataset	Evaluation Metrix	Testing Dataset		
		OLHWDB1.1	UNIPEN	LIPI Indic
OLHWDB1.1	SP(%)	93.86	31.67	43.70
	EP(%)	86.61	36.30	55.14
UNIPEN	SP(%)	24.21	97.57	27.44
	EP(%)	16.64	93.87	32.03
LIPI Indic	SP(%)	49.55	25.75	98.13
	EP(%)	29.69	40.49	96.71

2) Generalization Ability of the Proposed Model.: To study the generalization ability of the proposed model, we set several kinds of experiments as follows: train by Chinese, test by Chinese (C-C), C-U, C-I, U-C, U-I, and I-U, where C, U and I shorts for Chinese OLHWDB1.1, UNIPEN English and digits, and LIPI Toolkit Indic dataset respectively. Table I shows that the proposed model has stable generalization ability to a variety of language datasets. Besides, the proposed model trained by Chinese handwriting online dataset can achieve satisfactory performance on UNIPEN and Indic dataset. However, model trained on neither UNIPEN nor Indic datasets gets pool performance. This phenomenon indicates that, Chinese character covers much more complicated structure than English, digits and Indic. Besides, our proposed deep frame-point sequence consistent network has a strong power in recovering the temporal information from a static image. As a result, the proposed network can perform well on the unknown dataset given the fact that the training set already covers sufficient information.

TABLE II
COMPARISON THE RECOGNITION RESULTS.

Dataset	Recognition Rate(%)			
	Online	Offline	Recovered Trajectory	Offline + Recovered Trajectory
OLHWDB1.1	95.43	90.77	94.67	95.40
LIPI Indic	96.61	92.35	95.54	96.69



(a) Static Chinese Character Images

(b) Recovered Chinese Character Trajectory

(c) Static UNIPEN Character Images

(d) Recovered UNIPEN Character Images

(e) Static Tamil Character Images

(f) Recovered Tamil Character Trajectory

Fig. 5. Examples of the recovered handwriting trajectory. Character images in (a), (c) and (e) are the static images randomly chose from OLHWDB1.1, UNIPEN, and LIPI Toolkit Tamil Indic dataset. Colored images in (b), (d) and (f) are the recovered trajectory sequence, and each color represents one stroke.

3) Comparison of Recognition Rate.: To further analyze the quality of the recovered handwriting trajectory, we first list some recovered trajectory results in Figure 5, and then compare the recognition performance of using static character images, original points sequence, recovered points sequence, and the combination of static images and recovered points sequence in Table II.

In Figure 5, we randomly choose several static images from OLHWDB1.1, UNIPEN, and LIPI Toolkit Tamil Indic dataset (see Figure 5(a), 5(c) and 5(e)) to recovery their handwriting trajectory. The recovered trajectory is marked in different colors, and each color represents one stroke. In fact, each stroke is the connection of start and end point obtained by point prediction stream. From Figure 5, we can see that examples in UNIPEN dataset preserve the best shapes since data in UNIPEN dataset is much simpler than the other compared datasets. We also validate the rationality of improving the recognition results by utilizing both static offline images and the recovered trajectory online sequence. In Table II, we can see that online data achieves the best recognition results, while the performance of static offline images declines a lot. Recovered trajectory online sequence obtained by our proposed network shows comparable or even better recognition results than the static offline images. If we combine the recognition results of offline images and recovered trajectory online sequence, the recognition rate can be further improved.

V. CONCLUSION

Handwriting trajectory recovery provides much merit for character recognition. With the dynamic information of pen tip moving order, characters with multiple strokes could be easily classified and recognized. In this paper, we introduce a novel two-stream framework to address sequence prediction problem. Our main contribution includes: 1) A new image frame prediction network for trajectory recovery; 2) A novel FPSCM to synchronize two-stream training on temporal flow. We propose and implement a carefully designed deep frame-point sequence consistent architecture to accomplish this task. Extensive experiments demonstrate that our network is able to recover high quality handwriting trajectory. With the success in this prediction task, we believe that our two-streams training architecture has great potential for other deep learning problems, such as video frame prediction, voice recognition, NLP and so on. In these tasks temporal information plays an important role. Thus different form of original input could boost each other if synchronized in the training process. We will continue exploring the possibility in our future work.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant No. 62006150, and the Science and Technology Commission of Shanghai Municipality under Grant No. 21DZ2203100.

REFERENCES

[1] P. Melanie, B. Berangere, G. Jordan, D. Aurelia and P. Lionel, "Dual-Memory Model for Incremental Learning: The Handwriting Recognition Use Case," in: International Conference on Pattern Recognition, 2021, pp. 5527–5534.

[2] J. L. Zhou, Y. F. Shen, L. Y. Li, C. Zhuo and M. S. Chen, "Swarm Intelligence-Based Task Scheduling for Enhancing Security for IoT Devices," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,vol. 42, no. 6, pp. 1756–1769, 2023.

[3] P. J. Cong, J. L. Zhou, J. L. Wang, Z. B. Wu and S. Y. Hu, "Learning-based Cloud Server Configuration for Energy Minimization under Reliability Constraint," IEEE Transactions on Reliability, Early Access, 2023.

[4] M. Pavlo, Z. Q. You and K. Q. Li, "A high-performance CNN method for offline handwritten Chinese character recognition and visualization," Soft Comput, vol. 24, pp. 7977–7987, 2020.
[5] Z. Li, N. Teng, M. Jin and H. Lu, "Building efficient CNN architecture for offline handwritten Chinese character recognition," Int. J. Document Anal. Recognit, vol. 21, pp. 233–240, 2018.
[6] Z. M. Zhang, G. J. Wu, Y. H. Li, Y. Yue and X. Zhou, "Deep Incremental RNN for Learning Sequential Data: A Lyapunov Stable Dynamical System," in: IEEE International Conference on Data Mining, pp. 966–975, 2021.
[7] Q. Cui, S. Wu, Q. Liu, W. Zhong and L. Wang, "MV-RNN: A Multi-View Recurrent Neural Network for Sequential Recommendation," IEEE Trans. Knowl. Data Eng, vol. 32, pp. 317–331, 2020.
[8] K. B. Ayan, B. Abir, K. B. Ankan, K. Aishik, B. Prithaj, P. R. Partha and P. Umapada, "Handwriting Trajectory Recovery using End-to-End Deep Encoder-Decoder Network," in: International Conference on Pattern Recognition, pp. 3639–3644, 2018.
[9] B. Pang, K. W. Zha, H. W. Cao, C. Shi and C. W. Lu, "Deep RNN Framework for Visual Sequential Applications," in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 423–432, 2019.
[10] A. Jon, G. Albert, F. Alicia and V. Ernest, "Word Spotting and Recognition with Embedded Attributes," IEEE Trans. Pattern Anal. Mach. Intell, vol. 36, pp. 2552–2566, 2014.
[11] L. M. Fang, H. W. Zhu, B. Q. Lv, Z. Liu, W. Z. Meng, Y. Yu, S. L. Ji and Z. H. Cao," HandiText: Handwriting Recognition Based on Dynamic Characteristics with Incremental LSTM," Transactions on Data Science, vol. 1, pp. 1–18, 2020.
[12] Y.-J. Xiong and S. Y. Cheng, "Attention Based Multiple Siamese Network for Offline Signature Verification," in: Proceedings of the International Conference on Document Analysis and Recognition, pp. 337–349, 2021.
[13] L. Colmenarejo and R. Prei, "Signatures of paths transformed by polynomial maps," Contributions to Algebra and Geometry, vol. 61, pp. 695–717, 2020.
[14] Z. P. Liu, L. Zhang and Y. Yang, "Hierarchical Bi-Directional Feature Perception Network for Person Re-Identification," in: ACM International Conference on Multimedia, pp. 4289–4298, 2020.
[15] W. L. Wang, Z. J. Li, Z. Q. Cai, X. B. Lv, C. K. Zhaxi and Y. H. Han, "Online Tibetan Handwriting Recognition for Large Character Set on New Databases," Int. J. Pattern Recognit. Artif. Intell, vol. 33, pp. 1953003:1–1953003:21, 2019.
[16] X. Y. Zhang, F. Yin, Y. M. Zhang, C. L. Liu and B. Yoshua, "Drawing and Recognizing Chinese Characters with Recurrent Neural Network," IEEE Trans. Pattern Anal. Mach. Intell, vol. 40, pp. 849–862, 2018.
[17] S. Chen, Z. Sun, L. H. Lin, Z. F. Liu, X. Liu, Y. T. Chong, Y. T. Lu, H. Y. Zhao and Y. D. Yang, "To Improve Protein Sequence Profile Prediction through Image Captioning on Pairwise Residue Distance Map," J. Chem. Inf. Model, vol. 60, pp. 391–399, 2020.
[18] B. S. David, P. Pedro and M. Arturo, "Convolutional long short term memory deep neural networks for image sequence prediction," Expert Syst. Appl, vol. 122, pp. 152–162, 2019.
[19] S. Shreya, R. Srikanth, V. Abhishek, M. Devraj and M. Shantanu, "Spatial-context-aware RNA-sequence prediction from head and neck cancer histopathology images," in: Annual International Conference of the IEEE Engineering in Medicine & Biology Society, pp. 1711–1714, 2021.
[20] T. S. Samia, M. H. Md, A. Ahsan and S. Swakkhar, "Convolutional neural networks with image representation of amino acid sequences for protein function prediction," Comput. Biol. Chem, vol. 92, pp. 107494, 2021.
[21] V. Ruben, Y. Jimei, H. Seunghoon, L. Xunyu and L. Honglak, "Decomposing Motion and Content for Natural Video Sequence Prediction," in: International Conference on Learning Representations, 2017.
[22] F. Yin, Q. F. Wang, X. Y. Zhang and C. L. Liu, "ICDAR 2013 Chinese Handwriting Recognition Competition," in: International Conference on Document Analysis and Recognition, pp. 1464–1470, 2013.
[23] A. R. Ahmad, B. K. Marzuki, R. Yusof and C. Viardgaudin, "Online handwriting recognition using support vector machine," in: IEEE Region 10 Conference, 2004.
[24] D. Jun, H. Jinshui, Z. Bo, W. Si and D. Lirong, "A Study of Designing Compact Classifiers Using Deep Neural Networks for Online Handwritten Chinese Character Recognition," in: International Conference on Pattern Recognition, pp. 2950–2955, 2014.
[25] C. Wu, W. Fan, Y. He, J. Sun, N. Satoshi, "Handwritten Character Recognition by Alternately Trained Relaxation Convolutional Neural Network," in: International Conference on Frontiers in Handwriting Recognition, pp. 291–296, 2014.