Handwriting Trajectory Reconstruction Using Spatial-Temporal Encoder-Decoder Network

Feilong Wei and Yuanping Zhu(✉)

Tianjin Normal University, No. 393 Binshuixi Road, Xiqing District, Tianjin, China
zhuyuanping@tjnu.edu.cn

Abstract. Chinese handwriting characters have complex strokes and various writing styles, which makes it difficult to reconstruct handwriting. Aiming at this problem, we propose a handwriting reconstruction method based on a spatial-temporal encoder-decoder network with constrains. Different from other models that generate trajectory coordinates through a fully connected network, the method proposed in this paper outputs heat map sequence. The model is consists of three modules: key point detection module, spatial encoder-decoder module and reconstruction constraint module. The key point detector module and the spatial encoder part of encoder-decoder module are composed of a full convolutional network. The former generates heat maps of all key points which is a branch of the spatial encoder, and the mainly encoding the spatial information of each position on the offline image. The temporal decoder module is composed of a GRU network and an MLP network. Finally, we combine temporal information and reconstruction constraints to generate the final sequence. At each time, the features encoding by the spatial encoder module are combined with the features at the previous time that generate a corresponding heat map. The main contribution of the work of this paper is to propose a method that more suitable for handwriting reconstruction of Chinese handwritten characters. Experimental results show that the CT [6] accuracy of our method has already reached 87.6% on OLHWDB1.1 dataset.

Keywords: Handwriting trajectory reconstruction · Full convolutional network · Encoder-decoder network · Deep learning.

1 Introduction

Handwritten text analysis and recognition [20] has always been an important field of OCR [9], and it has also been the focus of research by scientists in the past decade [14]. Handwriting analysis has been studied for a long time. From the initial rule-based method to the current deep learning network-based methods, the accuracy of recognition has been continuously improved. According to different representations, handwritten text recognition is divided into online

H. Ma et al. (Eds.): PRCV 2021, LNCS 13019, pp. 342–354, 2021.
https://doi.org/10.1007/978-3-030-88004-0_28

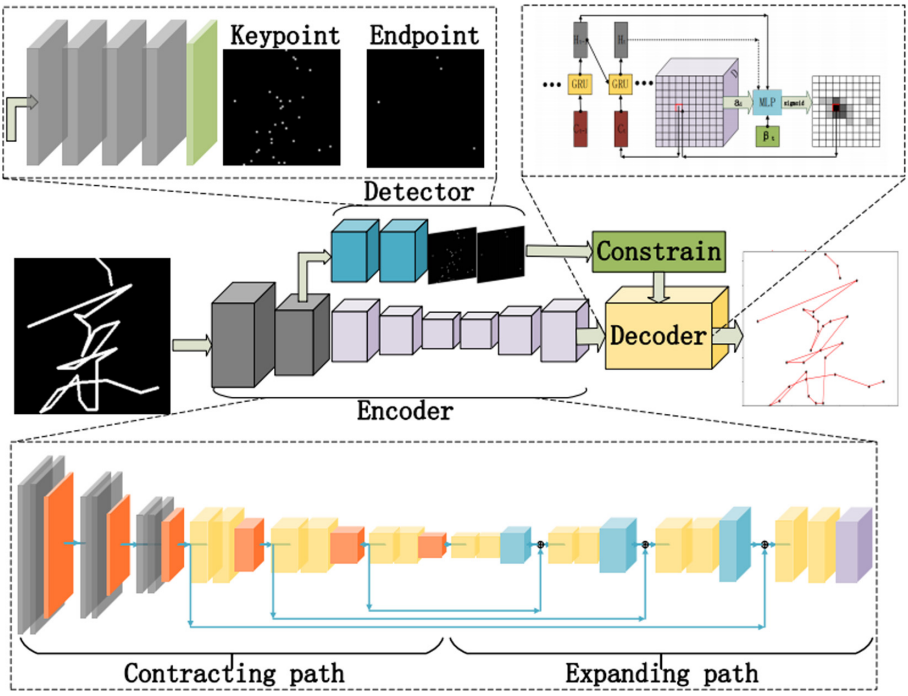

Fig. 1. The framework of handwriting reconstruction method.

handwriting recognition and offline handwriting recognition. Offline characters are represented by two-dimensional static images, while online characters are represented by a continuous coordinate sequence. Online characters also covers the trajectory, speed and angle of the handwriting during writing. Therefore, compared to offline characters, the accuracy of online handwriting recognition is usually higher than offline handwriting recognition. However, offline text collection is more convenient, more suitable for actual application scenarios, and its applications are more extensive. If the dynamic information of the text can be recovered from the two-dimensional static image, the static and dynamic information can be combined to further improve the accuracy of recognition. Moreover, handwriting reconstruction is widely used in smart writing and handwriting identification [7].

Currently, character handwriting reconstruction methods include graph search, template matching and writing rules, as well as deep learning based methods [6,20]. The method based on graph search [17] is to find a path with the least cost according to the minimum energy cost criterion. It is only suitable for the restoration of the writing order of numbers and alphabets. The method based on template matching [11] needs to build a stroke template library, and restore the handwriting by comparing the input image with the template. This method has a wider application range and higher accuracy, but it is too complicated

to calculate the best path in the matching process. The method [2] based on writing rules uses the structural characteristics of characters to express the relationship between character strokes, and then uses rules to restore their order. Its disadvantage is that it cannot adapt to changes in writing styles and cannot handle text with broken pens. The method [19] based on deep learning performs a series of preprocessing on the image, and finally performs the arrangement prediction of the order relationship of each pixel through the network. It has poor adaptability to the complicated text with many strokes. Other methods based on deep learning like [6,6] extracted the feature sequence of the two-bit static image, and finally the handwriting sequence is generated through RNN and fully connected network. It is not very adaptable to samples which have complex font and a wide range of stroke's number.

When a person is writing, the visual attention will move with the movement of the handwriting. In machine vision, according to this feature, we express it as the response probability of corresponding position at different times, and should be concentrated in a certain area or point. Therefore, this paper proposes a handwriting reconstruction method based on spatial-temporal encoder-decoder network, which simulates the process of human visual attention movement [18] by predicting the probability of each point on the image at different times.

The rest of this article is organized as follows. The second section introduces Spatial-Temporal Encoder-Decoder Network model proposed in this article in detail. The third section explains in detail the reconstruction constraint proposed in this paper. The fourth section introduces the composition of the loss function in detail. The fifth section is the detail of experiment and results. The last section is the conclusions.

2 Spatial-Temporal Encoder-Decoder Network

In this section, we introduce in detail how our proposed method generates online handwriting sequences based on offline pictures. As mentioned earlier, we did not directly output, but output the absolute position of the maximum probability point at different temporal steps. The Spatial-Temporal Encoder-Decoder Network is divided into three modules to generate handwriting sequence:key point detector module, spatial encoder module, and temporal decoder module. The spatial encoder module is the backbone network of the model, which is essentially a variant FCN network and outputs the spatial features of each position of the offline image. Figure 3 shows its structure. The key point detector module is a branch of the backbone network, which outputs and classifies all key points of the font. Recurrent neural network GRU [3] and Multi-layer perceptron MLP form temporal decoder, which combines spatial features to output the heat map sequence. The overall framework as show as Fig. 1.

2.1 Key Point Detector

The key point detector [1,5,16] module is to return the position of each candidate point through the FCN network [13]. Fully convolutional networks have better

spatial generalization capabilities than fully connected networks. This module detects all key points and divides them into two categories:end points and connection points. And then provide this information to the reconstruction constraint module. Since the full convolutional network FCN is more stable than the fully connected network in terms of position regression, more and more people use the full convolutional network FCN when studying key point detection [1]. Fully convolutional network only contains convolutional layer, pooling layer and activation layer. Specific parts are selected through the information connection between parts. This method is very meaningful for target detection (Fig. 2).

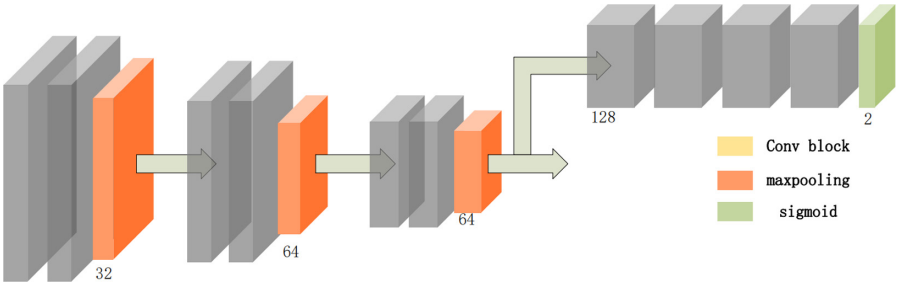

Fig. 2. The structure diagram of key point detector network

The detection network can identify the overall frame of the font and filter out the key parts. It has a certain sensitivity to the turning points of the line segments. Even a curve with a small curvature can identify subtle turning points, which are finally reflected in the output heat map. While detecting the key points, the detection network also determines the length of the output coordinate sequence, and realizes the self-variable length sequence generation.

2.2 Spatial Encoder Network

The key point detector network can find the key points of the font, but only extracts the position information, and cannot analyze the relationship between them (the feature map size is the same as the output image size, and the receptive field is limited, and deeper sequential features are not extracted). FCN cannot extract deeper-scale features well without changing the size of the feature map, so this work will be completed by the spatial encoder network. The spatial encoder network is a special full convolutional network. In order to extract the deep-level visual features of the image and obtain a larger receptive field while keeping the feature map at a certain size. So this part is composed of FCN and U-Net [12]. FCN [13] made a brief introduction in the front. And U-Net [12] is an FCN network with a special structure. U-Net [12] consists of two parts:one is the contracting path and the other is the expanding path. The contraction path can obtain contextual information of different scales, and the expansion

path can supplement some of the deep-level information of the image. But the supplement of information is definitely incomplete, so skip connect is needed to combine the higher resolution pictures on the contraction path. Since the method proposed in this paper is to generate a heat map of handwriting points through FCN. It limits the output size of FCN must take into account the size ratio of handwriting points in the original image. And the addition of U-Net [12] is to maintain a certain size scale feature map while obtaining the deeper features of the image. Meanwhile, the stroke texture information of the image can be obtained by expanding the receptive field.

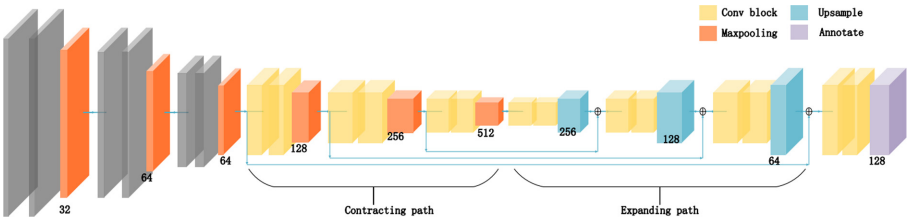

Fig. 3. The structure of spatial encoder network as showing as picture.

The specific structure of spatial encoder network shown in Fig. 3. The image through the spatial encoder network, is encoded as a tensor of size $d \times H' \times W'$. We denote these coding features as Eq. 1,

$$a = \{a_1, a_2, a_3, ..., a_n\}, a_i \in R^d, L = H' \times W' \tag{1}$$

where d is the dimension of a_i.

2.3 Temporal Decoder Network

The Temporal Decoder Network is essentially a candidate determiner composed of MLP and GRU [3]. In order to link the offline image with the variable-length output sequence, the paper [18] calculates an intermediate vector to provide a regional feature filter for subsequent recognition and classification. But we use this intermediate vector to output the coordinate points we need in the image heatmap. Figure 4 shows the work flow of the temporal decoder network, where MLP is an multi-layer perceptron composed of multiple fully connected networks and is the output layer of the temporal decoder network. GRU [3] is an improved version of cyclic neural network RNN, which solves the problem of gradient disappearance or gradient explosion during RNN training, and the space occupancy rate is much smaller than LSTM [4] while achieving the same effect. The hidden state calculation equation of GRU [3] see Eq. 2 $\sim$ Eq. 5.

$$z_t = \sigma\left(W_{hz} H_{t-1} + U_{cz} C_t + b_z\right) \tag{2}$$

$$r_t = \sigma\left(W_{hr} H_{t-1} + U_{cr} C_t + b_r\right) \tag{3}$$

$$\widetilde{H_t} = \tanh\left(W_h\left(H_{t-1} \otimes r_t\right) + U_h C_t + b_h\right) \tag{4}$$

$$H_t = \left(1 - z_t\right) \otimes H_{t-1} + z_t \otimes \widetilde{H_t} \tag{5}$$

Among them,$\sigma\left(\cdot\right)$ is the sigmoid function, $z_t, r_t, \widetilde{H_t}$ which is the update gate, reset gate and candidate state. When the temporal decoder network predicts the position of the handwriting point at each time step, it outputs the probability of each area, so the output only needs to maximize the probability of the candidate area. The temporal decoder network combines spatial features a_i and the hidden state H_{t-1} of the current GRU to calculate the maximum probability position of the handwriting point at the current moment (see Eq. 6∼Eq. 7),

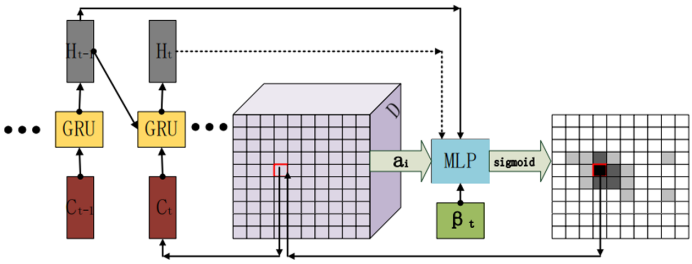

Fig. 4. The calculation process of the temporal decoder network.

$$e_{ti} = v_a^T \tanh\left(W_a H_{t-1} + U_a a_i\right) \tag{6}$$

$$p_{ti} = \frac{\exp\left(e_{ti}\right)}{\sum_{i=0}^{L} \exp\left(e_{ti}\right)} \tag{7}$$

where $v_a \in R^n, W_a \in R^{n' \times n}, U_a \in R^d$. And then we will the most probable point as the C_t to strengthen the relationship between points, like Eq. 8.

$$C_t = a\left[\max\left(p\right)\right] \tag{8}$$

In order to strengthen the path information, this article proposes a handwriting trend feature. The handwriting trend feature is a blank graph (β) whose size is the output scale, and the corresponding position is marked at each time step. Then trend feature are extracted by convolution and be send to the MLP. The final calculation formula is as shown as Eq. 9 ∼ Eq. 12.

$$\beta = (0) \in F^{H' \times W'} \tag{9}$$

$$\beta_t = (1_{ij}) \in F^{H' \times W'}, i \in (0, W'), j \in (0, H') \tag{10}$$

$$F_t = f\left(\beta_t\right) \tag{11}$$

$$e_{ti} = v_a^T \tanh\left(W_a H_{t-1} + U_a a_i + U_f f_t\right), f_t \in F_t \tag{12}$$

where β_t is the trajectory picture in the time step t, and $f\left(\cdot\right)$ is a convolution module. Finally the output e_{ti} will be used in Eq. 11 which represents the response of each position of the corresponding time step t.

3 Handwriting Reconstruction Constraints

Although the Spatial-Temporal Encoder-Decoder Network has a certain adaptability to the reconstruction of handwriting Chinese characters with complex fonts and broken pens, the handwriting point probability based on the whole image will be chaotic when faced with such samples. In order to constrain the chaos of handwriting, we designed a connection rule based on different handwriting points, as shown in Fig. 5.

In the key point detection module, we divide all points into connection points and end points. So, we defined two rules:

Theorem 1. *The starting point of the line segment must be the end point*

Theorem 2. *There must be a solid line in the line segment.*

In practical applications, we select candidate points based on the rules and the output of the model (See Eq. 13∼ Eq. 15),

$$p_{ti} = \begin{cases} \frac{\exp(e_{ti}) \times d_i}{\sum_{i=0}^{L} \exp(e_{ti}) \times d_i}, \; if \; last \; point \in endpoints, \\ \frac{\exp(e_{ti}) \times k_i}{\sum_{i=0}^{L} \exp(e_{ti}) \times k_i}, \; if \; last \; point \in connectionpoints. \end{cases} \tag{13}$$

$$l = \frac{1}{n} \phi \left(last \; point, candidate \; point \right) \tag{14}$$

$$P_{ti} = \begin{cases} \max \left(p_{ti} \right), & if \; last \; point \in endpoints, \\ \max \left(p_{ti} \times l \right). & if \; last \; point \in connectionpoints. \end{cases} \tag{15}$$

where $\phi(l, c)$in Eq. 14 means interpolating sampling between two points in the original image and n is the number of samples. In addition, $k_i \in key \; point \; map$ and $d_i \in end \; point \; map$. P_{ti} is the final predicted value.

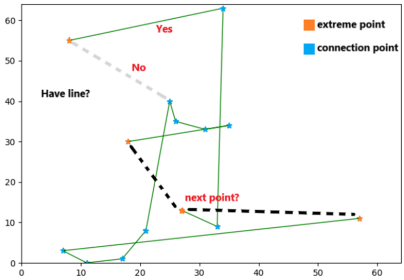

Fig. 5. Example of the situation in the process of reconstruction

4 Loss Function

Since the Spatial-Temporal Encoder-Decoder Network needs to learn key point detection and key point sorting, these two tasks are different. So we define the final loss function as Eq. 16 like [8], where L_{det} represents the loss in the key point detection task and L_{sq} represents the loss in the key point sorting task.

$$L = L_{det} + L_{sq} \tag{16}$$

In order to measure the gap between the predicted map and the label and balance the quantitative relationship between key points and the background, focal loss [10] is used as the loss function, as shown in the Eq. 17.

$$L_{det} = \frac{-1}{N} \sum_{c=1}^{C} \sum_{h=1}^{H} \sum_{w=1}^{W} \begin{cases} \beta \left(1 - p_{cij}\right)^{\alpha} \log\left(p_{cij}\right), & if\ y_{cij} = 1, \\ \left(1 - \beta\right) p_{cij}^{\alpha} \log\left(1 - p_{cij}\right), & otherwise. \end{cases} \tag{17}$$

Different from the traditional sorting loss function, our sorting task is to maximize the probability of the label point at each time, so we directly adopt the cross-entropy loss function.(L_{sq} see Eq. 18)

$$L_{sq} = \frac{-1}{N} \sum_{t=1}^{N} \log\left(p_{t,label[t]}\right) \tag{18}$$

5 Experiment

In order to verify the effectiveness of the proposed method in handwriting reconstruction, this chapter conducts ablation experiments and comparative experiments.

5.1 Dataset Processing

OLHWDB1.1 and Tamil dataset are used in experiment.OLHWDB1.1 includes 3755 types of Chinese characters, which is written on a separate page. The stroke coordinates of the pen tip are recorded. Tamil dataset is from paper [6], which can explore the reconstruction effect on other languages, is a dataset of HP Company‘s compete. We should convert it to offline form because all of them are trajectory sequence.

Different from offline handwriting characters saved in the form of static images, online handwriting characters retain richer dynamic information when writing in the form of handwriting point sequences. We save the original data in the form of a formula [20] like Eq. 19,

$$\left[\left[x_1, y_1, s_1\right], \left[x_2, y_2, s_2\right], ..., \left[x_n, y_n, s_n\right]\right] \tag{19}$$

where x_i and y_i is the coordinate,s_i is the point state. And then, we convert the data set into a specific form for training.

We think that the points that are too dense or the intermediate points on the same line are redundant points. In order to filter excess points, we set two conditions [20] as shown in Eq. 20 and Eq. 21,

$$\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \leq T \tag{20}$$

$$\frac{\Delta x_{i-1}\Delta x_i + \Delta y_{i-1}\Delta y_i}{\sqrt{\left(\Delta x_{i-1}^2 + \Delta x_i^2\right) \cdot \left(y_{i-1}^2 + y_i^2\right)}} \geq C \tag{21}$$

where T is the threshold to filter out points with too dense distance and C is the threshold to filter out the middle point in the same straight line. In order to protect the starting point and end point of the strokes from being screened out, the screening operation is carried out when $s_i = s_{i-1} = s_{i+1}$.

Offline Character Generation. In order to make the key point heat map and label correspond to the offline image, map the preprocessed handwriting point sequence coordinates to the image whose size is $H' \times W'$. Then we resize the image to $H \times W$. (See (a) in Fig. 6) According to the corresponding label, generate a heatmap of key point [15] based on the Gaussian distribution on the image.(See (b) in Fig. 6)

5.2 Implementation Details

The network model in this article is built under the pytorch framework, and the GPU model used by the platform is NVIDIA 1080Ti, which runs on a 64-bit Linux system. In the data preprocessing in this paper, the parameter T in Eq. 20 is $0.05 \times \max(H, W)$ and the parameter C in Eq. 21 is -0.9. The size of image is $H \times W = 512 \times 512$, the heat map and the output scale are $H' \times W' = 64 \times 64$. The dimension of the output a_i of the spatial encoder network is $d = 128$. The hidden state H_t of GRU is a 64-dimensional tensor. Finally, we use the optimizer Adam to set the initial learning rate $lr = 0.001$ and decay to 0.1 every 10 rounds.

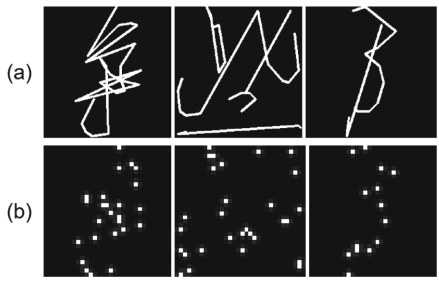

Fig. 6. Offline characters (a) and corresponding heatmap labels (b).

5.3 Evaluation Metrics

At present, there is no unified standard for the evaluation of online handwriting generation problems, such as paper [6,20]. This is also due to the large differences in the methods of generating handwriting.

Due to the particularity of the method proposed in this article. We use the average probability of the corresponding position of each handwriting point of the character as the criterion for the quality of the model.(See Eq. 22)

$$meanP = \frac{1}{K}\sum_{t=1}^{K} p_{t,indice} \tag{22}$$

where K is the number of trajectory points. Although $meanP$ cannot fully represent the recovery degree of a font handwriting, it can reflect the response degree of the model to handwriting points.

In addition, in order to facilitate the comparison with the paper [6], we also adopted their evaluation index(See Eq. 23 $\sim$ Eq. 24),

$$Starting\ Point\ Accuracy = \frac{Number\ of\ correct\ SP}{Total\ number\ of\ test\ images} \tag{23}$$

$$Junction\ Point\ Accuracy = \frac{Number\ of\ correct\ JP}{Total\ number\ JP\ points\ in\ test\ data} \tag{24}$$

when complete trajectory (CT) of an offline character image is perfectly retrieved along with the correct starting point,we evaluate this metric as a positive result.

5.4 Experiment and Result Analysis

In order to verify the necessity of the handwriting trend characteristics (TC) in Eq. 11 and reconstruction constraints (RC) in the model proposed in this article, we conducted corresponding ablation experiments with $meanP$ as the evaluation index. We randomly select 5000 samples from the test set for evaluation (see Table 1) The results of the Table 1 are predictable. The handwriting trend characteristics provides model with the features formed by the handwriting points of all previous moments, and provides enlightening information for the next moment. And reconstruction constraints can properly correct its errors and provide more accurate information for the next moment.

Table 1. The meanP of each method combination

Method	meanP
Decoder	0.559
Decoder +TC	0.723
Decoder +RC	0.753
Decoder +TC+RC	**0.796**

Table 2. Stroke recovery accuracy

	Evaluation Metrics	Tamil	OLHWDB1.1
Bhunia et al. [6].	Starting Point (SP)Accuracy	98.12%	85.00%
	Junction Points (JP) accuracy	**97.02%**	70.40%
	Complete trajectory (CT) retrieval accuracy	95.54%	64.30%
Ours.	Starting point (SP) accuracy	**99.10%**	**93.80%**
	Junction points (JP) accuracy	97.00%	**89.30%**
	Complete trajectory (CT) retrieval accuracy	**95.60%**	**87.60%**

We also conducted a comparative experiment with the paper [6] in Tamil dataset and OLHWDB1.1. The results are from 1,000 randomly selected samples. (see Table 2)

We have compared the accuracy of our proposed method with the method from [6]. We selected it because it is the latest methods in this field as comparison objects and implemented them on the dataset. That ours model are more suitable for trajectory reconstruction of Chinese characters. A few qualitative results of methods are shown in Fig. 7 and Fig. 8.

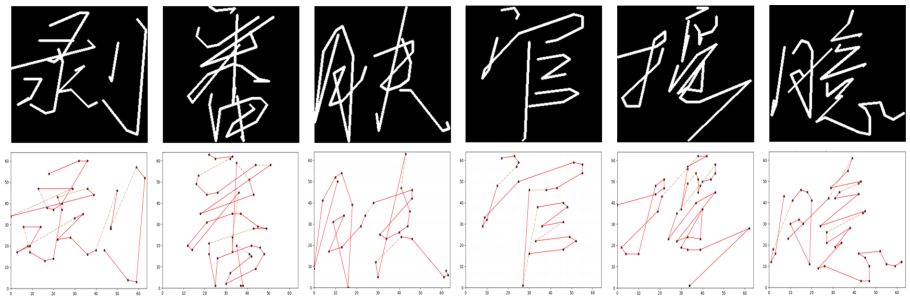

Fig. 7. Examples of the recovery trajectory from offline character on OLHWDB dataset.

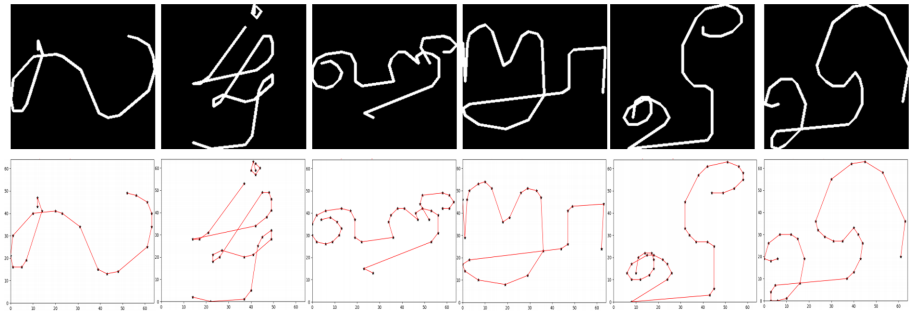

Fig. 8. Examples of the recovery trajectory from offline character on Tamil dataset.

6 Conclusion

This paper proposes a method of regression trajectory sequence that generates heatmap base on Spatial-Temporal Encoder-Decoder Network. The reconstruction results is better than the method [6] on OLHWDB. The coordinates generated in this way cannot be directly trained in the network, and the gradient must also be faulted by generating heat map labels. In future work, we will focus on solving this problem and combine it with GAN to generate a more complete trajectory sequence. In addition, whether the model can completely recover the handwriting of characters that have not been touched before will also be a future research direction.

Acknowledgement. This work was supported by the Natural Science Foundation of Tianjin(Grant No.18JCYBJC85000)

References

1. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: Openpose: Realtime multi-person 2d pose estimation using part affinity fields. IEEE Trans. Pattern Anal. Mach. Intell. **43**(1), 172–186 (2021)
2. Cao, Z., Su, Z., Wang, Y.: An offline handwritten chinese character writing sequence recovery model. J. Image Graphics **10**(1), 2074–2081 (2009)
3. Cho, K., et al.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. In: Proceedings of the Empiricial Methods in Natural Language Processing (EMLP) (2014)
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
5. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B.: Deepercut: a deeper, stronger, and faster multi-person pose estimation model. In: Proceeding of European Conference on Computer Vision, pp. 34–50 (2016)
6. Kumar Bhunia, A., et al.: Handwriting trajectory recovery using end-to-end deep encoder-decoder network. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 3639–3644 (2018)
7. Lai, S., Jin, L., Zhu, Y., Li, Z., Lin, L.: Synsig2vec: Forgery-free learning of dynamic signature representations by sigma lognormal-based synthesis. IEEE Trans. Pattern Anal. Mach. Intell. **8**(1), 99–112 (2021)
8. Law, H., Deng, J.: Cornernet: detecting objects as paired keypoints. Int. J. Comput. Vis. **128**(3), 642–656 (2020)
9. Li, L., Gao, F., Bu, J., Wang, Y., Yu, Z., Zheng, Q.: An End-to-End OCR Text Re-organization Sequence Learning for Rich-Text Detail Image Comprehension. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12370, pp. 85–100. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58595-2_6
10. Papandreou, G., et al.: Towards accurate multi-person pose estimation in the wild. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
11. Qiao, Y., Yasuhara, M.: Recover writing trajectory from multiple stroked image using bidirectional dynamic search. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR), pp. 970–973 (2006)

12. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, pp. 234–241 (2015)
13. Schwing, A.G., Urtasun, R.: Fully connected deep structured networks. In arXiv:1503.02351. (2015)
14. Shi, B., Xiang, B., Cong, Y.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(11), 2298–2304 (2016)
15. Shi, X., et al.: Deep learning for precipitation nowcasting: A benchmark and a new model. In: 31st Annual Conference on Neural Information Processing Systems, NIPS, pp. 5617–5627 (2017)
16. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4724–4732 (2016)
17. Yu, Q., Yasuhara, M.: Recovering drawing order from offline handwritten image using direction context and optimal euler path. In: IEEE International Conference on Acoustics, pp. 765–768 (2006)
18. Zhang, J., et al.: Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition. Pattern Recognit. **71**(11), 196–206 (2017)
19. Zhang, R., Zhan, Y., yang, M.: A method for restoring handwritten strokes based on endpoint sequence prediction. Comput. Sci. **046**(4), 264–267 (2019)
20. Zhang, X.Y., Yin, F., Zhang, Y.M., Liu, C.L., Bengio, Y.: Drawing and recognizing chinese characters with recurrent neural network. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 849–862 (2018)