HANDWRITING TRAJECTORY RECOVERY FROM OFF-LINE MULTI-STROKE CHARACTERS BY DEEP ORDERING PREDICTION AND HEURISTIC SEARCH

Tie-Qiang Wang[*,†], *Cheng-Lin Liu*[*,†,‡]

[*]National Laboratory of Pattern Recognition (NLPR),
Institute of Automation of Chinese Academy of Sciences, Beijing 100190, China
[†]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
[‡]CAS Center for Excellence of Brain Science and Intelligence Technology, Beijing 100190, China
Email: {tieqiang.wang, liucl}@nlpr.ia.ac.cn

ABSTRACT

Stroke order recovery from off-line multi-stroke characters is a great challenge due to the ambiguity in intersection and connection among strokes. In this paper, we propose a novel framework for handwriting trajectory recovery from off-line handwritten characters by deep neural network (DNN) based ordering prediction and heuristic search, where several DNN modules are designed to extract stroke skeleton, ambiguous zones and starting points, respectively. Then, the ordering matrix M_o among all the stroke segments is calculated by a pointer network (Ptr-Net). Besides, a convolutional neural network (CNN) is used to measure the time adjacency between two arbitrary segments. Based on these necessary measurements, the final writing order is decided via searching for the optimal permutation by heuristic A^* search. Experiments on handwriting images synthesized from the public online handwriting datasets CASIA-OLHWDB1.1, ICDAR13-Online and UNIPEN show that the proposed method yields superior performance on Chinese and English/Arabic handwriting.

Index Terms— handwriting trajectory recovery, pointer network, ordering matrix, heuristic search

1. INTRODUCTION

Handwritten character recognition can now yield high accuracies using deep neural networks trained with large datasets [1]. However, the recovery of handwriting trajectory from offline handwritten image is still an unsolved problem despite the numerous efforts devoted in the past decades [2]. Handwriting trajectory recovery (HTR) is needed for many applications, such as stroke trajectory based font design [3], writing quality estimation and calligraphy analysis [4]. Technically, HTR is a stroke ordering problem while stroke parsing from offline handwriting images is difficult due to the ambiguity of intersection and connection among strokes [5].

The existing HTR methods can be categorized into single-stroke character based methods [6] and multi-stroke character based ones [4, 7]. Generally, single-stroke based methods trace the whole handwriting trajectory from a starting point (SP) to an ending point (EP) [6], and global search based methods [5] outperform rule-based methods [6] by searching a path to optimize a pre-defined criterion.

More challenges exist in multi-stroke handwritten characters such as Chinese characters, where the continuity among strokes and the ordering of disconnected strokes are main issues [7]. To cope with this, Qiao and Yasuhara [7] search for the best writing paths between handwritten strokes and the pre-defined template strokes, but it still suffers from excessively distorted and intersected strokes. Recently, Zhao et al. [8, 9] proposed a CNN-based model (named DEN), which predicts the probability of the next stroke point position from previous frames of part-drawn handwriting images. Despite its ability of modeling instant writing states from offline handwritten character images, DEN ignores the ambiguous zones caused by the intersections of different strokes.

To provide a practical solution of HTR from multi-stroke offline images, we propose a framework based on stroke segment ordering. Particularly, HTR is formulated as a problem of stroke segment ordering and starting point prediction in a heuristic search framework [10], where the search space is built on the ordering matrix learned by a pointer network [11]. By integrating the deep ordering and time-adjacency relation prediction between stroke segments, the handwriting trajectory is given by the optimal ordering of stroke segments found by heuristic search.

The main contributions of this work are as follows: (1) We first formulate the HTR problem as an optimal ordering of stroke segments; (2) We predict the ordering matrix among stroke segments using a pointer network and search for the optimal ordering using heuristic search. (3) Experiments on multi-lingual handwritten characters demonstrate the superiority of the proposed method.

The rest of this paper is organized as follows: Section 2 reviews related works; Section 3 describes our proposed HTR method; Section 4 presents experimental results; and Sec-

978-1-6654-3864-3/21/$31.00 ©2021 IEEE

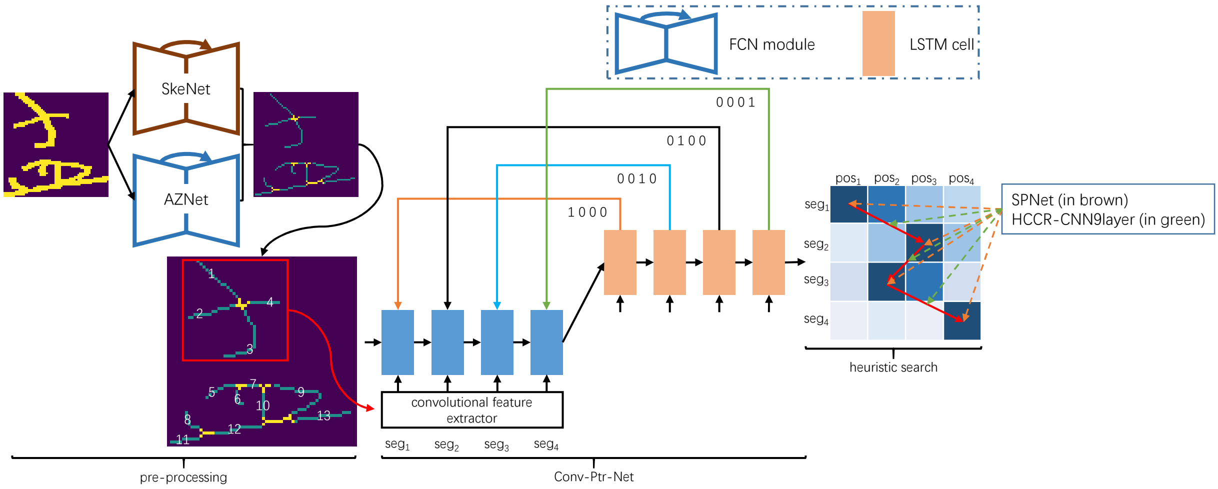

Fig. 1. The pipeline of the proposed HTR method.

tion 5 draws concluding remarks.

2. RELATED WORK

Ambiguous Zone Detection. Ambiguous zones (AZs) [12] are the unwanted artifacts or distortions at the regions with intersections of strokes, and obstacle HTR task heavily. Earlier methods detected AZs using the feature points or contour information around them [12]. Recently, Kim et al. proposed an ambiguous zone detection method based on FCN [13], which also inspires us in this paper.

Pointer Networks for Sequence Re-Ordering. In our experiments, $S_{in} = (s_1, ...s_i, ..., s_N)$ and S_{out} denote a sequence of stroke segments and their ground-truth handwriting order, respectively. Technically, Pointer networks (Ptr-Nets) re-order S_{in} in an end-to-end manner [11]: the training target of S_{in} can be treated as a one-hot vector sequence $S_{out} = (y_1, ...y_i, ..., y_T)$, $y_i \in \{0, 1\}^{N\times 1}$, where $y_i[j] = 1$ means that the i-th item in S_{in} is in the j-th position in the writing order.

3. METHODOLOGY

3.1. System Overview

As shown in Figure 1, the proposed method consists of pre-processing, Conv-Ptr-Net based ordering and heuristic search. In pre-processing, two isomorphic fully convolutional networks (FCNs) named SkeNet [14] and AZNet extract the stroke skeleton and ambiguous zones [15], respectively; Then the skeleton image is split into stroke segments at the detected ambiguous zones (AZs), and another FCN named SP-Net (sharing the same architecture with SkeNet) is used to find the starting point (SP) of each segment; The ordering matrix [7] M_o among all the segments is measured by a convolutional pointer network (Conv-Ptr-Net). Besides, the time-adjacency relations between two segments seg_p, seg_q is also measured by a CNN $f(seg_p, seg_q)$. Based on M_o and f, the optimal recovered trajectory with the minimal cost is obtained by heuristic search.

3.2. Pre-Processing

The architectures of the SkeNet/AZNet were proposed by [14], and are briefly outlined in Figure 2: (1) the convolution filters initialized from the network HCCR-CNN9Layer of [1] (the 1st row); (2) the convolution filters for feature maps of multiple scales (the 2nd row); (3) learnable upsampling for enlarging feature maps (the 4th row); (4) convolution kernels for the final fusion (the 5th row).

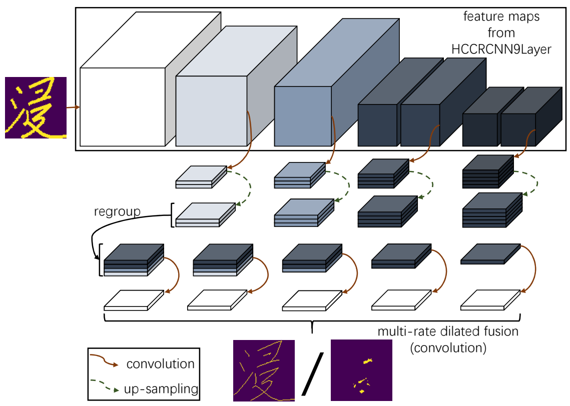

Fig. 2. Architectures of the SkeNet and AZNet [14].

3.3. Starting Point Detection

After skeletonization, ambiguous zone detection and stroke segments splitting, HTR needs to know the order of all the stroke segments and the starting point for each segment. Therefore, we formulate starting point detection using an SPNet. As shown in Figure 3, we take the character image and a stroke segment as input, and the earlier written half of the input segment as the training target. In testing, the input channels of SPNet are a character image and a stroke segment map, the segment is split into two equal parts at its middle point. For each part, the pixel values of SPNet output map are accumulated. Thus, the part with a larger summation is treated as earlier written.

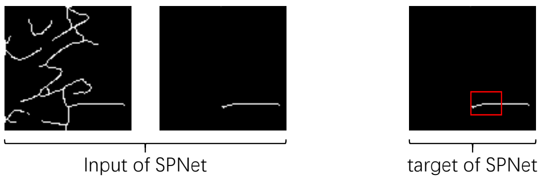

Fig. 3. Data preparation for the proposed SPNet. The red box indicates the earlier written half of a stroke segment.

3.4. Stroke Segment Ordering via Pointer Network

We re-order stroke segments into writing orders via Conv-Ptr-Net, which consists of a convolutional feature extractor Φ (the convolutional part of HCCR-CNN9Layer [1]) and an encoder-decoder module [11]. Φ provides stroke segment feature sequence $X = \Phi(S_{in}) = \{x_1, ..., x_N\}$. However, the output of Ptr-Net still relies on the initial orders of input sequences [11]. So, we calculate the minimal distance between each segment and the top-left corner of the character image. Then, we pre-order all the segments by this distance in ascending order.

We use LSTM [11] to build our encoder $\mathbb{E}$ and decoder $\mathbb{D}$, whose hidden states are $\{e_1, ..., e_N\}$ and $\{d_1, ..., d_N\}$, respectively. Now we can formulate the encoding operations:

$$f_i = \sigma(W_f[e_{i-1}, x_i] + b_f),\ i\text{-th output of forget gate,} \quad (1)$$
$$\gamma_i = \sigma(W_\gamma[e_{i-1}, x_i] + b_\gamma),\ \alpha_i = tanh(W_\alpha[e_{i-1}, x_i] + b_\alpha),$$
$$c_i = c_{i-1} \odot f_i + \gamma_i \odot \alpha_i,\ i\text{-th state of LSTM cells,} \quad (2)$$
$$o_i = \sigma(W_o[e_{i-1}, x_i] + b_o),\ i\text{-th output of output gate,} \quad (3)$$
$$e_i = o_i \odot tanh(c_i),\ i\text{-th output of hidden state.} \quad (4)$$

The decoder $\mathbb{D}$ then treats $\{o_1, ..., o_N\}$ as its input sequence, and accepts the final states e_N and c_N of $\mathbb{E}$ as its initialized states. Following the same computations as Eq. 1-4, $\mathbb{D}$ produces $\{d_1, ..., d_N\}$. Thus, attention mechanism can fulfill the ordering task [11]:

$$\mu_i^j = \upsilon^T tanh(W_1 e_i + W_2 d_j), \quad (5)$$

where υ, W_1 and W_2 are learnable. μ_i^j measures the probability of the j-th predicted stroke being located at the i-th position of S_{in}. The final output of $\mathbb{D}$ can be written as $\{\mu_1^j, ..., \mu_N^j\}$. We express μ^j as a probability distribution:

$$M_o^{ij} = p(j\text{-th written segment is } seg_i) = softmax(\mu^j)[i]. \quad (6)$$

Base on Eq. 6, we train Ptr-Nets by minimizing:

$$L_{ptr} = -\sum_j \sum_i M_o^{ij} y_i^j. \quad (7)$$

Algorithm 1 A^* search for stroke segment ordering

INPUT: $OPEN = \{BEGIN\}$, $CLOSED = \varnothing$
while $OPEN \neq \varnothing$ **do**
  choose seg_{cur} with minimal $C(BEGIN - \cdots - seg_{cur})$ from $OPEN$, seg_{cur} belongs to pos_{cur}
  if $seg_{cur} = END$ **then**
    trace from seg_{cur} back to $BEGIN$, and RETURN P
  else
    move seg_{cur} into $CLOSED$
    for seg in pos_{cur+1} **do**
      if $seg \notin CLOSED$ and $seg \notin OPEN$ **then**
        set parent of seg as seg_{cur}, put seg with $C(BEGIN - \cdots - seg_{cur} > seg)$ into $OPEN$
        if $seg \in OPEN$ **then**
          if $C(BEGIN - \cdots - seg_{cur} - seg) < C(BEGIN - \cdots - seg)$ **then**
            change the parent node of seg as seg_{cur}
          end if
        end if
      end if
    end for
  end if
end while

3.5. Heuristic Search for Optimal Ordering

Though the stroke segments can be ordered solely from M_o, the measures of time-adjacent relations between stroke segments can be complementary, since the M_o does not differentiate time-adjacent segment pairs from those non-adjacent pairs. Therefore, for a pair of stroke segments seg_p, seg_q, we use a two-channel HCCR-CNN9Layer architecture [1] $f(seg_p, seg_q)$ to judge whether they are time-adjacent (1) or not (0). The network is trained by minimizing the prediction loss on data of stroke segment pairs.

On obtaining M_o and f, A^* search aims to seek the optimal stroke order with minimal cost. For a partial (currently searched but incomplete) path P, we should accumulate the cost $C(P)$ of all the searched segments in P. Besides, A^* search also provides a heuristic function $H(P)$ to estimate the remaining cost of a complete solution containing P. Given a currently searched path $P = \{g_1, ..., g_j\}$ where g_j indicates the segments seg_{g_j} is placed at the j-th position, we calculate its cost $C(P)$ as:

$$C(P) = \sum_{i=1}^{j}[1 - p(i\text{-th written segment is } seg_{g_i})] + \sum_{i=1}^{j-1}[1 - f(seg_{g_i}, seg_{g_{i+1}}))]. = \sum_{i=1}^{j}[1 - M_o^{g_i,i}] + \sum_{i=1}^{j-1}[1 - f(seg_{g_i}, seg_{g_{i+1}})], \quad (8)$$

where the first item means that a higher likelihood of placing seg_{g_i} at pos_i should cause less cost. Similarly, the second item in Eq. 9 means that time-adjacent stroke segments can also reduce the cost. For P, we need to use the heuristic function $H(P)$ in A^* search [10] to estimate remaining cost:

$$H(P) = \sum_{i=j+1}^{N}[1 - max_k M_o^{k,i}] + \sum_{i=j-1}^{N-1}[1 - max_{g_i} f(seg_{g_i}, seg_{g_{i+1}})], \quad (9)$$

where all the involved stroke segments are out of P. Algorithm 1 shows the whole searching process. After selecting the best stroke segment from $OPEN$ with a minimum path cost, we expand it with other segments.

3.6. Data Generation for Training

The training of our deep models demands plenty of ground-truthed skeletons, ambiguous zones and the writing orders of stroke segments in handwritten images. Fortunately, the on-line handwritten samples [16] record split strokes by (x, y)-coordinate sequences, which can be viewed as the ideal skeletons of synthesized character images (generated by dilating the (x, y)-coordinate lines) and contains the ground-truth writing orders of human. The off-line image in Figure 1 is generated by dilating the strokes of plotted online handwritten character with appropriate control of stroke width and edge smoothness.

4. EXPERIMENTS

We synthesize off-line handwritten images from CASIA-OLHWDB1.1 (∼1.121 millions, 3755 classes) [16] for the training and test our models on the synthesized data (224,590 samples, 3755 classes) from ICDAR-2013 Online HCCR Competition Database [17]. Besides, we take the same 10000 English/Arab symbols with [8, 9] from UNIPEN (8000 as training set and 2000 as testing set). The size of all the synthesized images is 64 × 64. We reproduced the model proposed by [14] with comparable results on the skeletonization task successfully and only report the AZ detection task in this section.

4.1. Ambiguous Zone Detection

Table 1. Results of Ambiguous Zone Detection.

models	ICDAR13-Online			UNIPEN		
	P (%)	R (%)	F-measure (%)	P (%)	R (%)	F-measure (%)
OverlapNet [13]	70.2	66.1	68.0	84.5	82.3	83.3
HED-Net [18]	79.6	75.6	77.5	87.4	85.1	86.2
DeepSke-Net [19]	80.3	76.3	78.2	88.0	85.7	86.8
AZNet (ours)	**81.0**	**76.9**	**78.8**	**88.4**	**86.1**	**87.2**

We use precision (P), recall (R) and F-measure ($= \frac{2 \times P \times R}{P+R}$) to evaluate the foreground pixels of ambiguous zones detected by different models [18, 19] in Table 1. Compared with DeepSke-Net [19], SkeNet adds a multi-rate convolutional fusion layer to generate the final output [14]. OverlapNet works without any task-oriented designs [13], so, it misses more pixels than others. Comparing with HED-Net and DeepSke-Net, we can tell that the inter-channel regrouping of feature maps [19] handles the multi-scale strokes better [19]. Furthermore, our AZNet proves that the multi-rate dilated fusion finds more details at the AZs in handwritten character images. Some samples processed by SkeNet and AZNet are shown in Figure 4. Though the results extracted by our models do not fit the ground-truthed pixels perfectly, they still hold unambiguous structures for the subsequent task.

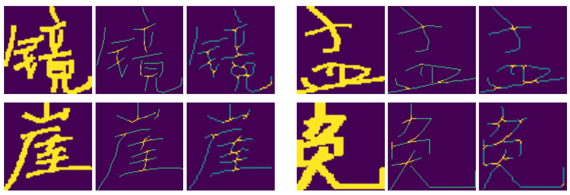

Fig. 4. Some results of SkeNet and AZNet. In each triplet, the input image, ground-truthed image and the result are listed from left to right (the blue pixels: skeleton segments, the bright pixels: ground-truthed/detected AZs).

4.2. Starting Point Detection and Time-Adjacent Relation Prediction

Table 2. Accuracies of Starting Point Detection and Time-Adjacent Relation Prediction.

Starting Point Detection		
models	ICDAR13-Online (%)	UNIPEN (%)
SEN [8]	96.2	97.8
DEN [9]	97.0	98.6
SEN+DEN [9]	97.1	98.6
SPNet (ours)	**97.6**	**98.8**
Time-Adjacent Relation Prediction		
HCCR-CNN9Layer (f)	97.2	99.4

The results of starting point detection and time-adjacent relation prediction are shown in Table 2. In the comparison

Table 3. Accuracies of HTR.

models	ICDAR13-Online			UNIPEN		
	Acc_{sp} (%)	Acc_{seg} (%)	Acc_t (%)	Acc_{sp} (%)	Acc_{seg} (%)	Acc_t (%)
SEN [8]	95.3	92.0	90.4	98.2	97.4	97.1
DEN [9]	96.0	93.1	92.1	98.2	97.6	97.4
SEN+DEN [9]	96.8	93.4	92.3	98.3	98.0	97.5
Conv-Ptr-Net (ours)	92.9	85.7	83.3	95.3	92.6	90.8
ours	**97.1**	**96.5**	**94.9**	**98.8**	**98.3**	**97.9**

methods, the training target of SEN is a heatmap, where the t-th written pixel is label as $\exp(-kt)$ [9], and DEN aims to predict the starting pixel alone of a stroke segment [8]. SEN can roughly provide the drawing trend but not details, while the learning targets of DEN are undersized (only one pixel). Therefore, the SPNet can even outperform the incorporation of SEN and DEN. Besides, HCCR-CNN9Layer $f(seg_p, seg_q)$ successfully predicts the time-adjacent relationships of most stroke segments pairs on ICDAR13-Online (97.2%) and UNIPEN (99.4%), which ensures the effectiveness of Eq. 9.

4.3. Handwriting Trajectory Recovery

We measured the performance of HTR based on these evaluation metrics: (1). Starting point accuracy (Acc_{sp}) [8]: If the network can predict the earliest handwriting point of an off-line image correctly, we count it as a positive result. (2). Stroke segment ordering accuracy (Acc_{seg}) [11]: A positive result is counted when all the stroke segments are correctly ordered. (3). Complete trajectory accuracy (Acc_t) [20]: A positive result is obtained when stroke segments in a character are correctly placed with correct starting points.

Table 4. Accuracies of HTR. $[min\ length, max\ length]$ denotes the length range of the input stroke segment sequences in ICDAR13-Online dataset.

$[min\ length, max\ length]$	percentage (%)	Acc_t (%)
$[1, 8]$	22	95.6
$[9, 16]$	39	95.9
$[17, 24]$	18	94.2
$[25, 32]$	14	92.2
$[32, 36]$	7	91.4

From results show in Table 3, we can see that the proposed method outperforms the baseline methods in off-line handwritten Chinese character HTR task and yields competitive result compared to the baseline methods on UNIPEN dataset. Compared to a stand-alone Conv-Ptr-Net, the inclusion of time-adjacent segments measurement in A^* search significantly improves the performance. Moreover, though the F-measures of SkeNet and AZNet range from 0.7∼0.8 in [14] and Table 1, the accuracies of HTR in Table 3 can still achieve $> 90\%$. This implies that the modules measuring stroke segment ordering and time-adjacent relationship play robustly. In Table 3, a standalone Conv-Ptr-Net can reach $> 80\%$ Acc_{seg}, which shows that it learns plenty of the widely-accepted handwriting laws.

Table 4 presents the Acc_t for characters with variable numbers of stroke segments. The accuracies with different lengths are apparently imbalanced: samples with larger numbers of stroke segments report lower recovery accuracies. This is reasonable because it is more difficult to order larger number of stroke segments correctly. Some factors that cause failure cases are shown in Figure 5: (1) Cursive handwriting styles may cause some extremely indistinguishable ambiguous zones, whose adjacent stroke segments are hard to parse (the 1st sample); (2) Some tiny stroke segments are mis-predicted because they are covered by larger segments (the 2nd sample); (3) Some strokes with the same direction appear as one smooth line, whose ambiguous zones are undetected (the 3rd sample).

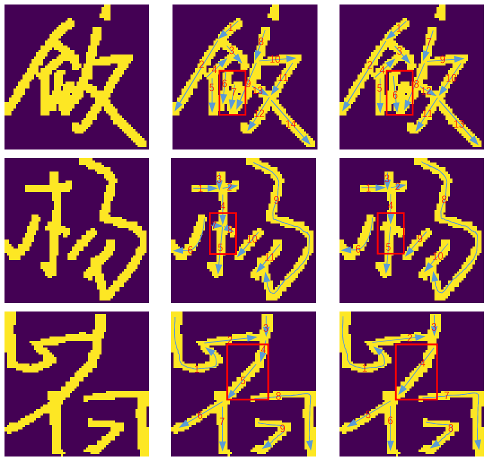

Fig. 5. Some failure results of our method. In each row, the input images, ground-truthed stroke order and the ordering result are listed from left to right. Orders, directions and failure cases are labeled by numbers, arrows and boxes, respectively.

5. CONCLUSION

In this paper, we propose a novel framework for HTR using heuristic search with deep models measuring orders and time-adjacency relationships among stroke segments. On the stroke segments produced by SkeNet and AZNet, SPNet is used to find their starting points, and a convolutional pointer network (Conv-Ptr-Net) is used to measure the ordering matrix M_o. The final handwriting trajectory is decided by searching the optimal permutation heuristically with costs formed by M_o and time-adjacency relationship measurements. The proposed method has demonstrated superior performance on both Chinese handwriting and UNIPEN datasets. In the future, end-to-end training is to be explored for accelerating the proposed framework.

Acknowledgements

This work has been supported by the Major Project for New Generation of AI under Grant No. 2020AAA0109702, the National Natural Science Foundation of China (NSFC) grants 61936003, 61721004.

References

[1] X. Xiao, L. Jin, Y. Yang, W. Yang, J. Sun, and T. Chang, "Building fast and compact convolutional neural networks for offline handwritten Chinese character recognition," *Pattern Recognition*, vol. 72, pp. 72–81, 2017.

[2] 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.

[3] J.-W. Lin, C.-Y. Hong, R.-I. Chang, Y.-C. Wang, S.-Y. Lin, and J.-M. Ho, "Complete font generation of Chinese characters in personal handwriting style," in *IEEE International Performance Computing and Communications Conference*, 2015, pp. 1–5.

[4] L. P. Cordella, C. De Stefano, A. Marcelli, and A. Santoro, "Writing order recovery from off-line handwriting by graph traversal," in *International Conference on Pattern Recognition*, 2010, pp. 1896–1899.

[5] Yu Qiao, M. Nishiara, and M. Yasuhara, "A framework toward restoration of writing order from single-stroked handwriting image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1724–1737, 2006.

[6] L. Rousseau, A. Éric, and J. Camillerapp, "Recovery of a drawing order from off-line isolated letters dedicated to on-line recognition," in *International Conference on Document Analysis and Recognition*, 2005, vol. 2, pp. 1121–1125.

[7] Y. Qiao and M. Yasuhara, "Recover writing trajectory from multiple stroked image using bidirectional dynamic search," in *International Conference on Pattern Recognition*, 2006, vol. 2, pp. 970–973.

[8] B. Zhao, M. Yang, and J. Tao, "Pen tip motion prediction for handwriting drawing order recovery using deep neural network," in *International Conference on Pattern Recognition*, 2018, pp. 704–709.

[9] B. Zhao, M. Yang, and J. Tao, "Drawing order recovery for handwriting Chinese characters," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 3227–3231.

[10] M. Nosrati, R. Karimi, and H. A. Hasanvand, "Investigation of the *(star) search algorithms: Characteristics, methods and approaches," *World Applied Programming*, vol. 2, no. 4, pp. 251–256, 2012.

[11] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2692–2700.

[12] Z.-W. Su, Z.-S. Cao, and Y.-Z. Wang, "Interpretation of ambiguous zone to improve thinning results of handwritten Chinese characters," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 24, no. 02, pp. 303–320, 2010.

[13] B. Kim, O. Wang, A. C. Öztireli, and M. Gross, "Semantic segmentation for line drawing vectorization using neural networks," *Computer Graphics Forum*, vol. 37, no. 2, pp. 329–338, 2018.

[14] T.-Q. Wang and C.-L. Liu, "Fully convolutional network based skeletonization for handwritten Chinese characters," in *AAAI Conference on Artificial Intelligence*, 2018, pp. 2540–2547.

[15] 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.

[16] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, "CASIA online and offline Chinese handwriting databases," in *International Conference on Document Analysis and Recognition*, 2011, pp. 37–41.

[17] 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*, 2013, pp. 1464–1470.

[18] S. Xie and Z. Tu, "Holistically-nested edge detection," in *IEEE International Conference on Computer Vision*, 2015, pp. 1395–1403.

[19] W. Shen, K. Zhao, Y. Jiang, Y. Wang, Z. Zhang, and X. Bai, "Object skeleton extraction in natural images by fusing scale-associated deep side outputs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 222–230.

[20] A. Kumar Bhunia, A. Bhowmick, A. Kumar Bhunia, A. Konwer, P. Banerjee, P. Pratim Roy, and U. Pal, "Handwriting trajectory recovery using end-to-end deep encoder-decoder network," in *International Conference on Pattern Recognition*, 2018, pp. 3639–3644.