Two-stage Fully Convolutional Networks for Stroke Recovery of Handwritten Chinese Character

Yujung Wang[1], Motoharu Sonogashira[2], Atsushi Hashimoto[3], and Masaaki Iiyama[2][0000−0002−7715−3078]

[1] Graduate School of Informatics, Kyoto University
[2] Academic Center for Computing and Media Studies, Kyoto University
[3] Graduate School of Education, Kyoto University

Abstract. In this paper, we propose a method to recover strokes from offline handwritten Chinese characters. The proposed method employs a fully convolutional network (FCN) to estimate the writing order of connected components in offline Chinese character images and a multi-task FCN to estimate the writing order and directions of strokes in each connected component. Online dataset CASIA-OLHWDB1.0 from the CASIA database is hired as the training set. Because the network produces discontinuous strokes, we refine the estimated writing orders using a graph cut (GC), in which the estimated directions are used for calculation of smoothness term. Experimental results with test dataset of CASIA-OLHWDB1.0tst demonstrate the effectiveness of our method.

Keywords: Handwriting Trajectory Recovery · Semantic Segmentation · Fully Convolutional Networks · Graphcut.

1 Introduction

Handwritten Chinese character recognition has been studied for the last two decades. This has been considered as a difficult problem, owing to target's complicated structures, leading to a large number of character classes, and the variability in writing style [1]. The approaches to handwriting recognition can be classified into two categories: offline recognition and online recognition. Offline recognition involves digital character images which are previously written on a paper and then captured by a scanner or a camera, while online recognition involves a sequence of coordinate points of pen trajectories which are captured on-line through special devices such as digitizer tablets.

In general, offline recognition is more difficult than online recognition. With character images only, offline recognition usually concentrates on complicated image processing for feature extraction [2]. On the other hand, using information of the pen trajectories, online recognition can dynamically analyze the structure of characters, which results in a capability to recognize the cursive script and an increase in the recognition rate [3]. Therefore, by extracting the information

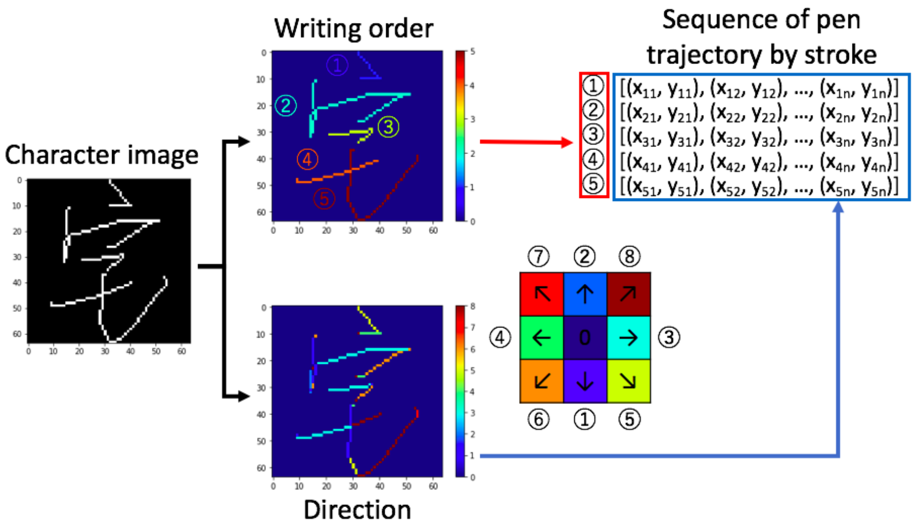
Writing order
Sequence of pen trajectory by stroke
Character image
① ② ③ ④ ⑤
① $[(x_{11}, y_{11}), (x_{12}, y_{12}), ..., (x_{1n}, y_{1n})]$
② $[(x_{21}, y_{21}), (x_{22}, y_{22}), ..., (x_{2n}, y_{2n})]$
③ $[(x_{31}, y_{31}), (x_{32}, y_{32}), ..., (x_{3n}, y_{3n})]$
④ $[(x_{41}, y_{41}), (x_{42}, y_{42}), ..., (x_{4n}, y_{4n})]$
⑤ $[(x_{51}, y_{51}), (x_{52}, y_{52}), ..., (x_{5n}, y_{5n})]$
⑦ ② ⑧
④ ← 0 → ③
⑥ ① ⑤
0 10 20 30 40 50 60
0 1 2 3 4 5
0 1 2 3 4 5 6 7 8
Direction

Fig. 1: Chinese characters can be represented as a sequence of strokes consisting of points with similar direction. The writing order is the order of strokes, and the direction is the direction a stroke written.

of strokes from character images, an offline recognition problem can be virtually viewed as online recognition.

In this paper, we propose a method to recover stroke of handwritten Chinese character images by estimating the information of stroke such as writing order and direction. A Chinese character is composed of one or more subcharacters consisting one or more strokes, which are mostly straight lines with a few polylines. In other words, a Chinese character can be represented as a sequence of strokes, consisting of one or more sub-strokes with similar directions. This is shown in Fig.1. We model the stroke recovery as a semantic segmentation problem, and employ two-stage fully convolutional networks (FCN) [4]. The first FCN estimates the writing order of connected components approximating the subcharacters from character images, and the second FCN estimates the writing order and directions of strokes of connected component images. Both FCN are trained using online data. Because the FCN could produce discontinuous strokes, a graph cut (GC) [5] is utilized to refine the estimated writing order by the estimated directions. The overview of our method is shown in Fig.2.

The contributions of our method are as follows: first, we estimate the writing order and directions of strokes using FCNs. Previous solutions [7–9, 11, 12, 14, 15] typically involve pixelwise sequence-to-sequence models, which assume a two-stage solution of stroke segmentation and direction estimation. In other words, such an approach assumes that strokes can be reliably segmented in the first stage. In contrast, our method recovers the strokes from a raw character image. Stroke segmentation is done simultaneously with estimation of stroke direction

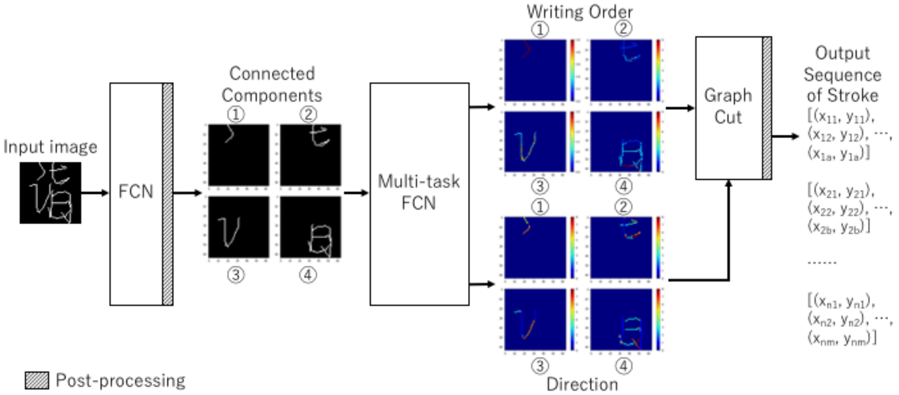
Input image
FCN
Connected Components
①
②
③
④
Multi-task FCN
Writing Order
①
②
③
④
Direction
Graph Cut
Output Sequence of Stroke
$[(x_{11}, y_{11}), (x_{12}, y_{12}), \cdots, (x_{1a}, y_{1a})]$
$[(x_{21}, y_{21}), (x_{22}, y_{22}), \cdots, (x_{2b}, y_{2b})]$
......
$[(x_{n1}, y_{n1}), (x_{n2}, y_{n2}), \cdots, (x_{nm}, y_{nm})]$
▨ Post-processing

Fig. 2: Overview of the proposed method.

by trained FCNs. The only required preprocessing is a connected component extraction, which is feasible in many cases. Second, we successfully recover the writing order and directions of strokes simultaneously for multi-stroke images. Because the writing order and direction relate to each other, the direction refines the connectivity problem of the writing order, and the writing order limits possible choices of the direction. As a result, both the writing order and the direction are improved.

2 Related Work

The recovery of online information from offline images has been studied for the last two decades [7–9, 11–16]. Features such as endpoints, trajectories of strokes, and relations between strokes have been proven to be useful in online recognition [6]. Research on stroke recovery aims to extract these features from offline images.

Most previous research works on tracing the trajectory, or extracting directions of pixels in other words. Methods involving the continuity between a pixel and its neighbors are first considered. Lee and Pan [7] traced the offline signatures using hierarchical decision making with a set of heuristic rules. Boccignone et al. [8] reconstructed trajectories based on some continuity criteria. Lallican and Viard-Gaudin [11] recovered the trajectories using the Kalman filter.

After a number of studies, researchers have become to focus more on the continuity between strokes, which is proved to improve not only the performance of trajectory recovery on single-stroked character images but also multi-stroked character images by the following methods. Qiao et al. [12] restored the trajectories of single-stroked handwriting images within the framework of edge continuity relations. Kato and Yasuhara [14] traced the drawing order of offline multi-stroke handwriting images. Qiao and Yasuhara [15] recovered writing trajectories from multi-stroked images using a bidirectional dynamic search, to find the best matching with the writing paths of template strokes.

On the other hand, some previous research has worked on either of estimating the writing order of the strokes or segmenting strokes. Liu et al. [13] utilized a model-based structural method to estimate the writing order of offline Chinese characters. Kim et al. [16] segmented the strokes from hand drawings using neural networks without ordering.

Only a few previous research has worked on estimating both directions and the writing order. Abuhaiba et al. [9] proposed a method to extract temporal information from offline Arabic characters through stroke segmentation, straight-line approximation, and the learning of token strings extracted from strokes by a fuzzy sequential machine, transforming the character images into sequences of coordinate points. [10]

Our method is designed to estimate the writing order and directions of strokes from offline multi-stroke character images using semantic segmentation, and then recover the pen trajectory using the writing order and directions of strokes. Moreover, with the multi-task FCN, our method estimate the writing order and directions simultaneously.

3 Proposed Method

In this section, we elaborate on our proposed stroke recovery method for offline handwritten Chinese characters.

A Chinese character is composed of one or more subcharacters, which are often common among many Chinese characters. Given subcharacters instead of the original Chinese character as the network input inherently decreases the number of character classes to learn. Therefore, we employ two-stage writing order recovery.

First, given bitmaps of offline character images, an FCN is utilized to estimate the writing order of the connected components approximating subcharacters in the character images. Next, a multi-task FCN is utilized to estimate the writing order and directions of strokes in each component. The FCNs are trained separately. Then, a graph cut (GC) is utilized to refine the estimation of the writing order by using the estimation of the direction for weighting. Finally, combining the results for the writing order and direction, the character images can be described in a sequence of strokes composed of coordinate points. The flow chart of the proposed method is presented in Fig.2.

3.1 Estimation of writing order of connected components

As the first step, we intend to extract subcharacters from the offline image with its writing order. However, we have no ground truth to train such FCN. Since the strokes of a subcharacter are typically written continuously, we alternate subcharactor segmentation for a connected components extraction in a binarized character image. Then, we employ an FCN that predicts their writing order.

Given a bitmap of an offline character image, the first FCN outputs an estimation of the writing order at the level of connected components. The connected components are represented as $P = \{p_1, p_2, ..., p_k\}$, and collection of their writing order is represented as $C = \{c_1, c_2, \ldots c_k\}$. The first FCN estimates the writing order at the level of connected components by predicting the pixel-wise likelihood of label assignments $P(C|v)$ for a pixel v. After predicting the order $l_{cv} = \mathrm{argmax}_{c_i} P(c_i|v)$ as the label with the maximum likelihood, we aggregate them to obtain a relative order of each connected component l_{p_i} by the following criteria:

$$l_{p_i} = \frac{1}{N(p_i)} \sum_{v \in p_i} l_{cv}, \tag{1}$$

where $N(p_i)$ is the number of pixels in the connected component p_i. We determine the writing order of P by matching l_{p_i} in ascending order to $c_1, c_2, \ldots, c_k$. Pixels in the connected component p_i are all reassigned to the label of writing order of p_i.

The architecture of the first FCN is illustrated in Fig.3, which is adapted from [4]. The architecture of the first FCN consists of five blocks of convolutional layers; three split layers placed after the third, forth, and fifth blocks of layers; a merge layer merging the split layers; and an output layer. The Rectified Linear Unit (ReLU) activation function is adopted in each Conv2D layer, and the softmax activation function is adopted in the output layer. Then, the connected components are sorted based on Eq. (1) and its post-process. The ordered components are input to the second FCN independently.

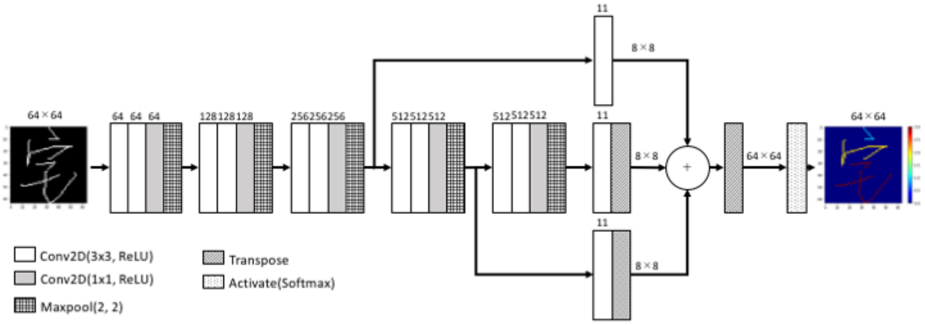

Fig. 3: Structure of the first FCN.

3.2 Estimation of writing order and stroke direction of each connected components

The second FCN is a multi-task FCN, and predicts the writing order and directions of strokes in the connected components. The architecture of the second

FCN is illustrated in Fig.4. The layers are the same as in the first FCN, except that the layers are split for the writing order and the direction, respectively. The sigmoid activation function is used in the output layer. Because the overlaps between strokes should have more than one labels, the sigmoid activation function, which gives probabilities ranging from 0 to 1 and not necessarily summed up to 1, can give these pixels more than one labels with similarly large probabilities.

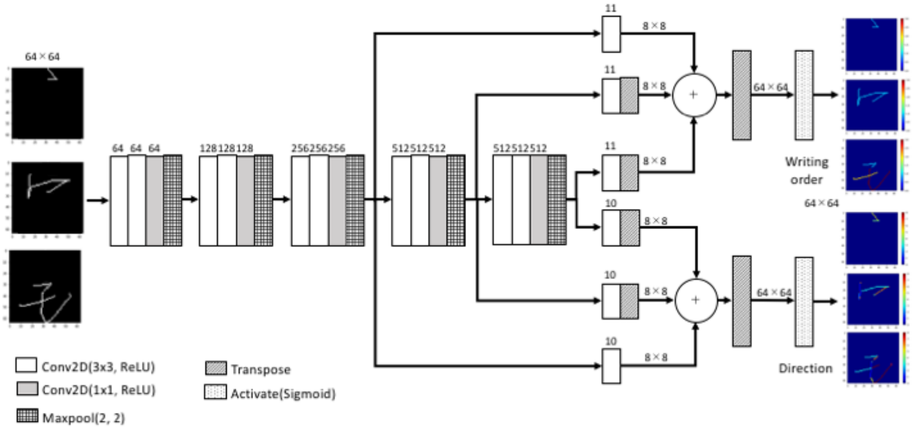

Fig. 4: Structure of the second FCN.

Both FCNs are trained using the binary cross-entropy loss function. In the estimation of the second FCN, each pixel v has a likelihood for the writing order denoted by $P(O|v)$ and a likelihood for the direction denoted by $P(D|v)$, where $O = \{o_1, o_2, ..., o_N\}$ is the collection of the writing orders and $D = \{d_1, d_2, ..., d_8\}$ are the discrete stroke directions, quantized into eight directions. We applied thresholds to the likelihoods for eliminating labels with small likelihoods. The thresholds are $0.8 \times \max(P(O|v))$ and $0.8 \times \max(P(D|v))$, respectively. The label assignment candidates of pixel v is represented as $L_{ov} = \{l_{ov}^1, l_{ov}^2, ..., l_{ov}^n\}$ for the writing order and $L_{dv} = \{l_{dv}^1, l_{dv}^2, ..., l_{dv}^m\}$ for the direction, ordered by the likelihoods, where n and m are the number of labels whose likelihoods are larger than the thresholds.

3.3 Label smoothing with graph cut

Then, given the label assignment candidates L_{ov} and L_{dv} for all pixels v from the multi-task FCN as input, the graph cut (GC) [5] is utilized to smooth the estimation of the writing order. The energy function of GC for a graph $G = (V, E)$ in our method is defined as follow:

$$\mathrm{E}(l) = \sum_{v \in V} \mathrm{E}_v(l_v) + \sum_{(u,v) \in E} \mathrm{E}_{uv}(L_{du}, L_{dv})\kappa \qquad (2)$$

where vertices V and edges E correspond to pixels and the relations between pixels and their eight neighbors, respectively, κ is the parameter, $\mathrm{E}_v(l_v)$ is the data term that measures the label assignments l_v, and $\mathrm{E}_{uv}(L_{du}, L_{dv})$ is the smoothness term that measures the penalties of difference between the input label set of directions L_{du} and L_{dv}.

The GC is executed separately for each label of the writing order. In the ith execution, pixels with label i are considered as the foreground, and others as the background. Pixels with label i but resulted in background in the GC calculation will move to the next execution with label $i+1$. The data term indicates the difference between the input label i and the assigned label l_v. Therefore, the data term is set as zero if the input label set L_{ov} of a pixel v contains the assigned label l_v, and a constant λ if it does not. However, in order to maintain the connectivity between pixels with the same assigned label, even though l_v is not in L_{ov}, data term of a pixel v is set as zero if there exists a neighbor pixel n with input label set L_{on} containing the assigned label l_v. The smoothness term is weighted as the difference between the directions of a pixel and those of its neighbors. The GC outputs the smoothed result of the estimation of the writing order. The equation of the data term and the smoothness term can be written as follows:

$$\mathrm{E}_v(l_v) = \begin{cases} \lambda \text{ if } l_v \notin L_{ov} \text{ and } l_v \notin L_{on} \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{3}$$

$$\mathrm{E}_{uv}(L_{du}, L_{dv}) = \begin{cases} w_{dd} \text{ if } (u,v) \in E \\ 9.0 \quad \text{otherwise} \end{cases}, \tag{4}$$

where w_{dd} is the weight of the difference between the directions of pixel u and pixel v, defined as

$$w_{dd} = \begin{cases} 1.0 & \text{if } u \rightarrow v \text{ and } v \rightarrow u \\ \min(w_a) & \text{if } u \rightarrow v \text{ or } v \rightarrow u \\ 9.0 & \text{otherwise} \end{cases} \tag{5}$$

$$w_a = \begin{cases} 1.0 \ \text{ if } \varDelta\theta = 0^\circ \\ 2.0 \text{ if } \varDelta\theta = 45^\circ \\ 3.0 \ \text{ otherwise} \end{cases} \tag{6}$$

where $u \rightarrow v$ indicates that the direction of pixel u pointing to pixel v is in the input label set L_{du}, and $\varDelta\theta$ is the minimum angular difference between L_{du} and L_{dv}.

Finally, with the result of the writing order and directions of strokes, pixels with the same writing order are put into a sequence with corresponding writing order to form a stroke, and the directions of each pixels are presented as a list of label L_{dv} for each pixel v.

4 Experiments

4.1 Implementation

In this section, we demonstrate the effectiveness of our method in extracting the online information from offline Chinese character images.

The online character dataset OLHWDB1.0 in the CASIA Chinese handwriting database was utilized in the experiment. The online data of characters were composed of sequences of coordinate points as strokes. We constructed character images with online data, and resized them to a size of 64×64. The FCN networks were trained by 87,600 character images, with the corresponding online information containing 300 classes. For the multi-task FCN, the character images are cut into connected components for training. Because the numbers of connected components with the same number of strokes differ a lot, we limited the numbers of connected components to be the same. For the parameters of the GC, we set $\lambda = 1500.0,$ and $\kappa = 5.0$ where the values were obtained by optimizing the results. Then, 21,700 character images were used for testing.

4.2 Evaluation

We evaluated the estimation as follow:

1. Direction accuracy: If the label of the direction of a point corresponds to the ground truth, the label assignment is considered to be correct. A pixel may have multiple labels owing to the overlaps, so the labels are counted separately.
2. Relative writing order accuracy: Because our method may cut a stroke into several strokes due to large changes in direction, which means that the number of strokes may be different from ground truth but the relative ordering of strokes is the same, relative writing order is used instead of absolute writing order. We employ two criteria for evaluating the writing order accuracy. One is Pearson correlation coefficient between the predicted writing order and its ground truth. The other is the relative ordering of each pixel v.
   (a) Pearson correlation coefficient: Pearson correlation coefficient globally represent the relation between the prediction and the ground truth. Value of coefficient near to 1 means that the number and the relative relation between strokes of prediction are similar to the ground truth, while value of coefficient near to 0 means that number or the relative relation between strokes of prediction are different from the ground truth. For value of coefficient smaller than 0, it usually means that the relative order between connected components of prediction are different from the ground truth.
   (b) Relative writing order accuracy: Relative writing order accuracy locally represent the pixel-based relation between the prediction and the ground truth. we calculate the mean of the predicted writing order of each stroke s_i with pixels within labeled with i in ground truth and compare it with

the predicted writing order of each pixel. The mean of the predicted writing order of pixels that belongs to the ith stroke s_i, which is denoted by l_{s_i}, is calculated as

$$l_{s_i} = \frac{1}{N(s_i)} \sum_{v \in s_i} l_{ov}^{S'} \tag{7}$$

where $N(s_i)$ is the number of pixels in the stroke s_i and $l_{ov}^{s'}$ is the corresponding label of pixel v in the estimated sequence of strokes S'. For a pixel v assigned with label $l_{ov}^{S'}$ and belongs to stroke s_i, the assignment is considered to be correct if $l_{ov}^{S'} < l_{s_{i+1}}$ and $l_{ov}^{S'} > l_{s_{i-1}}$, which is presented in Fig.5(a). We compare the ordering in the level of stroke instead of the level of pixels because when multiple pixels are assigned with the same label, the relative ordering of pixels in the same stroke will all be counted as positive, which is indicated in Fig.5(b).

We evaluated the writing order in a relative manner for two reasons. First, as described above, some of the strokes are cut into several sub-strokes by our method, owing to large changes in direction, which means that the points of these strokes have the same order as in the online sequences of characters, but belong to different strokes in the estimations. These points can be correctly evaluated using the relative writing order. Second, for points belonging to the same strokes, the order between them is determined by the direction, which means that the correctness of the absolute ordering between the points depends on the direction accuracy. Therefore, for the writing order, we do not consider the ordering between points in the same strokes.

4.3 Result

First, our method achieved an accuracy of 96.97% in direction estimation, which means most of the directions of strokes are recovered. The trajectories of strokes are successfully extracted from the offline character images by semantic segmentation.

Second, with the Pearson correlation coefficient of 0.52, which means that the corresponding estimation of strokes is related to the ground truth, our method achieved an accuracy of 75.87% in relative writing order estimation. Several results are shown in Fig.6. Result (a) to (c) are successful results, where strokes are successfully extracted and the ordering between the estimation and the ground truth are closely related. Result (d) and (e) are results with high accuracy but poor correlation coefficient, where strokes are well extracted but the ordering of continuous part does not match with the ground truth, which leads to the poor correlation. Result (c) and result (d) are differ from the ground truth but are actually more precise than the ground truth. Since our method recover strokes by the estimation of writing order and directions, it is more easier to find where to start and end with precise estimation.

In addition, the peak of the relative writing order accuracy remains at 98% to 100% for characters with one to five strokes and 70% to 80% for characters

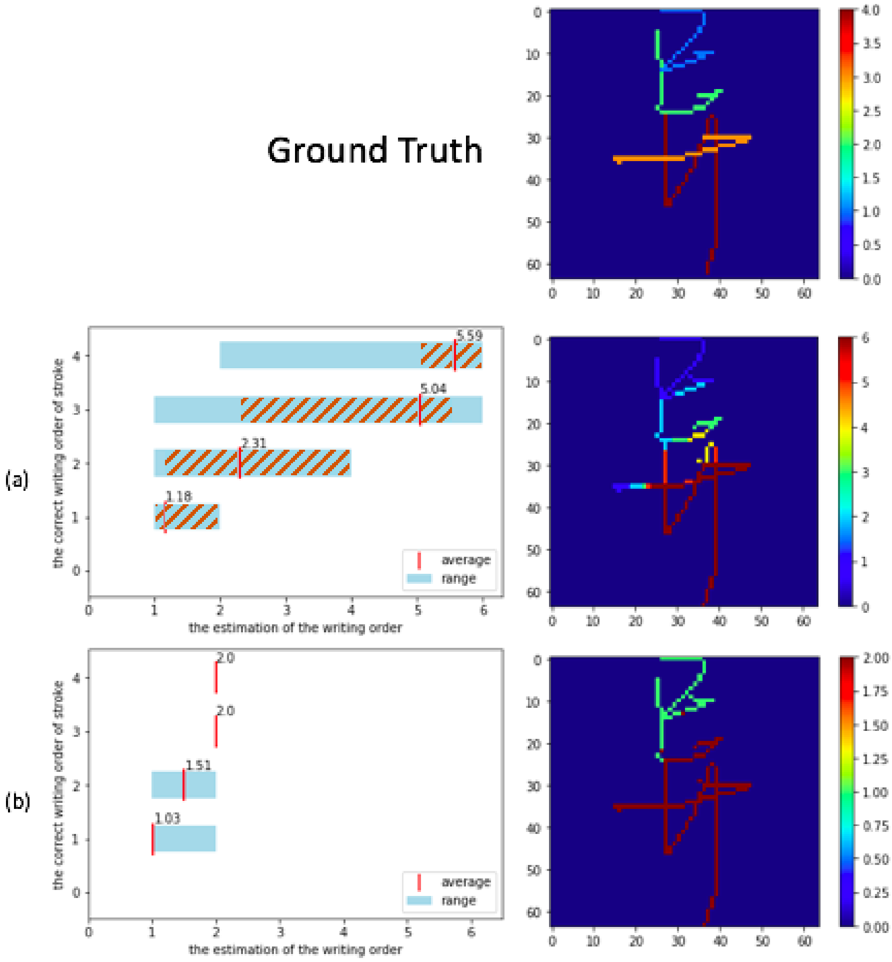

Fig. 5: Description of the relative writing order accuracy. In the left graphs, the vertical axis is the correct writing order of strokes s_1, s_2, s_3, s_4, the horizontal axis is the estimation of the writing order $l_{ov}^1, l_{ov}^2, ..., l_{ov}^6$, the blue bar is the range of the estimation in the same stroke, and the red line is the mean. The slash line area in (a) is the area counted as positive. We use the range instead of checking the label directly to prevent the situation of (b) which will make all points in s_3, s_4 to be correct.

with six to nine strokes. Our method performs promising results for characters with one to five strokes. For characters with six to nine strokes, though the accuracy is not higher than characters with less strokes, the average correlation coefficient remains 0.58 to 0.61, indicating the close relation between the ground

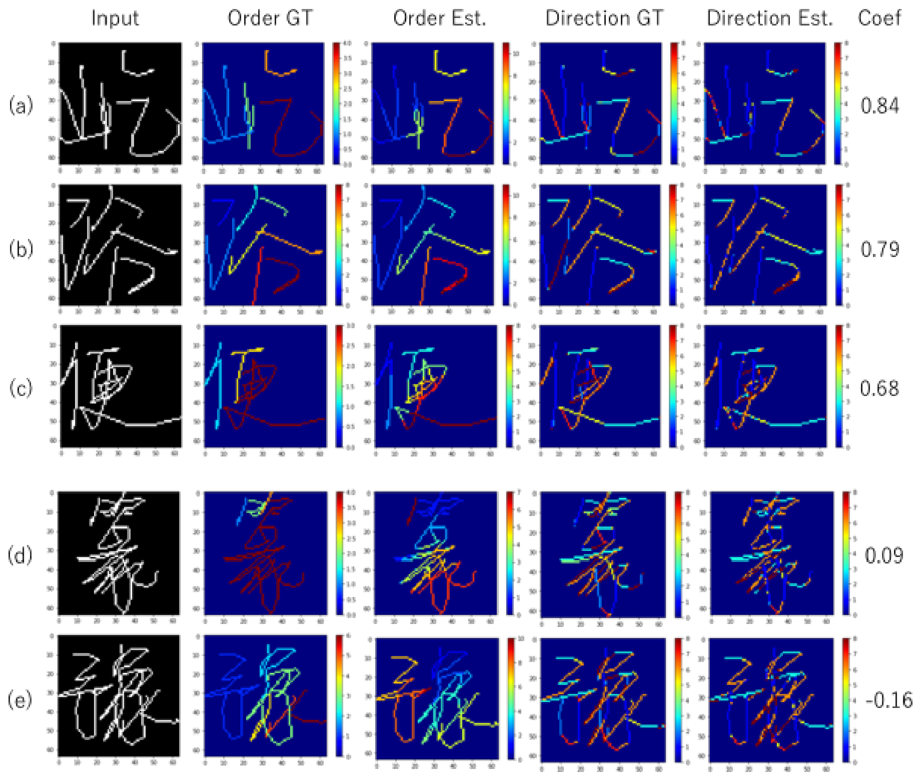

Fig. 6: Results of the proposed method. From left to right are the input, the ground truth of the writing order, the estimation of the writing order, the ground truth of directions, the estimation of directions, and the Pearson correlation coefficient. (a), (b), and (c) are successful results, and (d) and (e) are results with high relative writing order accuracy but poor correlation coefficient.

truth and the estimation. Therefore, our method is proved to work on characters with different numbers of strokes.

Third, Fig.7 shows the results using only multi-task FCN, multi-task FCN with GC, and the proposed FCN-FCN with GC. The results of the proposed FCN-FCN with GC are obviously better than the others. Extracting a character into continuous part and then into strokes is proved to achieve better results than extracting strokes directly.

In total, our method successfully estimates online information from offline handwritten character images.

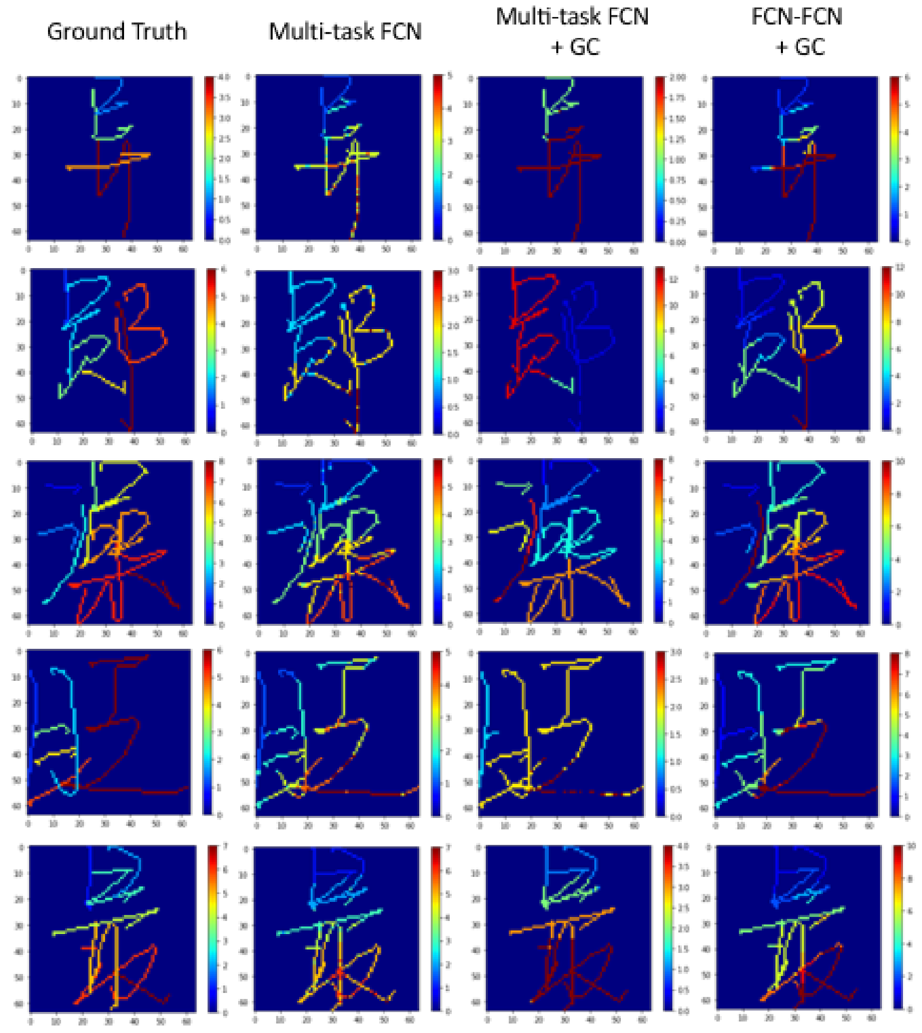

Fig. 7: Images of the ground truth, estimation using multi-task FCN only, estimation using multi-task FCN and GC, and the estimation using the proposed FCN-FCN model with GC. FCN-FCN model with GC obviously performs better than the others.

5 Conclusion

In this study, we introduced a deep learning method using FCN for stroke recovery of handwritten Chinese characters. Our method is based on semantic segmentation using two FCNs, respectively extracting connected components and strokes, and the estimations are refined by a GC. We demonstrated through the experiments that our method successfully extracts online information from offline character images.

However, for characters separated into left part and right part such as Fig.6(e), in which the correct ordering is left to right, FCN sometimes gives the reversed results. In the future work, we plan to improve the performance on ordering, and compare the result of online recognition using output of our method as input with the result of offline recognition using the original character images as input to evaluate our method on improving the accuracy of character recognition.

6 Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 17H06288.

References

1. LIU Chenglin,DAI Ruwei,XIAO Baihua. *Chinese character recognition: history, status and prospects[J].* Front. Comput. Sci., 2007, 1(2): 126-136.
2. C.-L. Liu, S. Jaeger and M. Nakagawa, *Online recognition of Chinese characters: the state-of-the-art*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 2, pp. 198-213, Feb. 2004.
3. Xu-Yao Zhang, Yoshua Bengio, Cheng-Lin Liu, *Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark, Pattern Recognition*, vol. 61, pp. 348-360, 2017
4. J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 3431-3440.
5. Yuri Boykov, Olga Veksler, Ramin Zabih, *Fast approximate energy minimization via graph cuts*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.23, no.1, pp.1222-1239, 2001.
6. D. S. Doermann and A. Rosenfeld, *Recovery of temporal information from static images of handwriting*, Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 162-168., Champaign, IL, USA, 1992
7. S. Lee and J. C. Pan, *Offline tracing and representation of signatures*, in IEEE Transactions on Systems, Man, and Cybernetics, vol. 22, no. 4, pp. 755-771, July-Aug. 1992.
8. G. Boccignone, A. Chianese, L.P. Cordella, A. Marcelli, *Recovering dynamic information from static handwriting*, Pattern Recognition, vol. 26, Issue 3, pp. 409-418, 1993
9. I.S.I Abuhaiba, M.J.J Holt, S Datta, *Recognition of Off-Line Cursive Handwriting*, Computer Vision and Image Understanding, vol. 71, Issue 1, pp. 19-38, 1998

10. B. Zhao, M. Yang and J. Tao, *Drawing Order Recovery for Handwriting Chinese Characters*, ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom, pp. 3227-3231, 2019
11. P. M. Lallican and C. Viard-Gaudin, *A Kalman approach for stroke order recovering from off-line handwriting*, Proceedings of the Fourth International Conference on Document Analysis and Recognition, Ulm, Germany, pp. 519-522 vol.2. , 1997
12. Y. 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(11), pp. 1724-1737, 2000
13. Cheng-Lin Liu, In-Jung Kim, Jin H. Kim, *Model-based stroke extraction and matching for handwritten Chinese character recognition*, Pattern Recognition, vol. 34, Issue 12, pp. 2339-2352, 2001
14. Y. Kato and M. Yasuhara, *Recovery of drawing order from scanned images of multi-stroke handwriting*, Fifth International Conference on Document Analysis and Recognition, pp. 261-264, 1999
15. Yu Qiao and M. Yasuhara, *Recover Writing Trajectory from Multiple Stroked Image Using Bidirectional Dynamic Search*, 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, pp. 970-973, 2006
16. Byungsoo Kim, Oliver Wang, A. Cengiz ztireli, Markus Gross, *Semantic Segmentation for Line Drawing Vectorization Using Neural Networks*, Computer Graphics forum, vol. 37, Issue 2, pp. 329-339, May 2018