A Method to Identify the Cause of Misrecognition for Offline Handwritten Japanese Character Recognition using Deep Learning

Keiji Gyohten, Hidehiro Ohki and Toshiya Takami
Faculty of Science and Technology, Oita University, Dannoharu 700, Oita 870-1192, Japan

Keywords: Offline Handwritten Character Recognition, Deep Learning, Convolutional Neural Network, Stroke Recognition, Identifying the Cause of Misrecognition.

Abstract: In this research, we propose a method to identify the cause of misrecognition in offline handwritten character recognition using a convolutional neural network (CNN). In our method, the CNN learns not only character images augmented by applying an image processing method, but also those generated from character models with stroke structures. Using these character models, the proposed method can generate character images which lack one stroke. By learning the augmented character images lacking a stroke, the CNN can identify the presence of each stroke in the characters to be recognized. Subsequently, by adding dense layers to the final layer and learning the character images, obtaining the CNN for the offline handwritten character recognition becomes possible. The obtained CNN has nodes that can represent the presence of the strokes and can identify which strokes are the cause of misrecognition. The effectiveness of the proposed method is confirmed from character recognition experiments targeting 440 types of Japanese characters.

1 INTRODUCTION

Recent progress in deep learning is remarkable, and is driving the current artificial intelligence boom. Deep learning techniques can be applied to various pattern recognition problems, resulting in significant breakthroughs. Character recognition, one of the pattern recognition problems, also benefits from the advancement of deep learning techniques.

In order to achieve high recognition performance in deep learning, a large amount of training data is required. Also, in the character recognition problem, the number of handwritten images that can be collected as training data is important. For example, He et al.'s (2015) research achieves high character recognition performance by collecting a large amount of handwritten character images and using them as training data. However, there are limitations in collecting a large amount of handwritten character images.

From this background, data augmentation has become common for enlarging training data for deep learning. When increasing image data for training, it is normal to apply various image processing methods to the given image data. In the case of character recognition problems, methods such as applying geometrical transformation to given character images and generating character images of various handwritings using character models have been proposed.

In particular, we consider that generating training character images using character models could contribute to the interpretability of deep learning for character recognition. Today, a neural network is expected to provide human-understandable justifications for its output, leading to insights about its inner workings. This is called the interpretability of deep learning (Chakraborty et al. 2017). In order to tackle this approach, various studies on object recognition have been undertaken in recent years. However, since various conceptual semantic structures are possible in a natural scenario, it is difficult to clearly identify judgment parameters for recognizing objects. On the other hand, in the case of character recognition, since the characters are composed of strokes arranged in clear logical structures based on the positional relationships, image features characterizing the recognition judgment can be clearly identified. We consider the possibility of recognizing the logical structure of the strokes in characters by learning the character images generated by combining the presence or absence of strokes from the character model.

Gyohten, K., Ohki, H. and Takami, T.
A Method to Identify the Cause of Misrecognition for Offline Handwritten Japanese Character Recognition using Deep Learning.
DOI: 10.5220/0008949004460452
In *Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2020)*, pages 446-452
ISBN: 978-989-758-397-1; ISSN: 2184-4313

In this paper, we propose a character recognition method that can grasp the logical structures of the strokes in the character by learning the augmented images generated from the character models. The proposed method consists of two phases: the phase of learning the structures of the strokes in the characters and the phase of learning the characters in consideration from the results of stroke recognition. In the phase of learning strokes, each node in the convolutional neural network's (CNN's) output layer corresponds to each stroke in the characters. In the learning process, character images lacking one stroke are provided to learn the presence or absence of a stroke. Thereafter, dense layers are added to output the final result of character recognition. The nodes in the output layer of the added network correspond to each character to be recognized.

2 RELATED WORKS

With the progress of research on deep learning, handwriting recognition using CNN has achieved remarkably high performance. Excellent performance has also been achieved in handwritten Chinese character recognition, for example, in the study of Zhang et al. (2017). This method achieved a recognition rate of 97.37% using 720 training images and 60 evaluation images for each of the 3,755 Chinese characters. Since Chinese characters have thousands of character types and their structure is complicated, a larger amount of training character images will be needed to improve handwritten Chinese character recognition performance using CNN. To solve this problem, it is common to apply data augmentation on the training data for CNN. Applying various image processing methods to the given character images is the easiest way to increase the training image data.

However, a problem in the data augmentation of training character images for Chinese character recognition is that various handwritings cannot be generated only by applying general image processing methods to original images. The training data generated by general image processing methods could be correlated with each other and overfitting may occur in the learning. We consider the use of character models to be efficient in generating various handwritings. These methods have been mainly used for script recognition (Bhattacharya and Chaudhuri 2009; Saabni and El-Sana 2013), font generation (Miyazaki et al. 2017), and so on. In addition, Liu et al. 2018; Ly et al. 2018; Shen and Messina 2016; and Wigington et al. 2017, use character models for the data augmentation in deep learning. Further progress in this type of approach is expected in the future.

Furthermore, the augmentation of character images using stroke-based character models may also contribute to constructing interpretable neural networks in character recognition using deep networks. The interpretable neural networks provide human-understandable justifications for its output, leading to insights about their inner workings (Chakraborty et al. 2017). If it is possible to make the network learn the training character images generated by controlling their logical character structure using the character models, the resulting network might be able to explain why such a judgment is made.

3 PROPOSED METHOD

3.1 Outline

Our method is characterized by recognizing characters after understanding the stroke structure of each character. This approach is realized by learning not only character images augmented by applying image processing, but also ones generated from character models having stroke structure. The method is based on the traditional CNN and is learned through the following two phases. The network first learns the augmented character images to be able to recognize the presence of each stroke in characters. Then, a dense layer is added to the network as the new output layer where each node corresponds to a character to be recognized. By learning the augmented character images, the network can recognize not only the characters but also the presence of each stroke in them.

3.2 Character Image Augmentation

The character image augmentation based on image processing is done by applying projective transformations to the original character images to imitate various handwritings. After obtaining a four-sided figure whose corners are placed randomly near the corners of the original character image, our method applies the projective transformation to the image to fit the distorted figure as shown in Fig. 1. Fig. 2 shows some examples of the augmented character images, where various handwritings can be imitated.

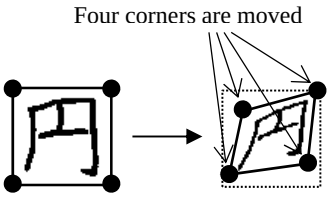


Figure 1: Character image augmentation based on projective transformation.

Figure 2: Examples of the character images generated by applying projective transformation.

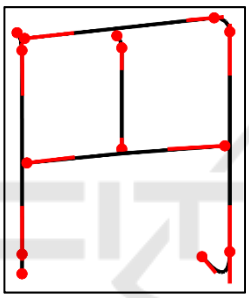


Figure 3: Example of characters represented by KanjiVG.

In the character image augmentation using stroke-based character models, we use KanjiVG[1] created by Ulrich Apel. KanjiVG is the character model that provides shapes and orders of strokes in each Japanese character as SVG (scalable vector graphics) files. The strokes are represented as Bezier curves and their orders for writing a character are described in the file. In addition, the data of the radicals that compose the characters are also described in the file, but not used in our method. Fig. 3 shows an example of a character represented by KanjiVG.

Our method generates character images using KanjiVG according to the procedure described below. First, our method generates strokes of a character from KanjiVG and approximates them to polylines. In order to change the appearance of the character, our method generates a Voronoi diagram whose seeds correspond to the vertices on the polylines, and moves each vertex randomly in the cell. Each vertex is moved according to the stroke order of the character, and the Voronoi diagram is updated with each movement of the vertex. This procedure makes it possible to change the appearance of the character without breaking the positional relationships among the strokes.

In order to construct the CNN that can identify the presence of each stroke in the characters to be recognized, our method needs to use augmented character images lacking a stroke as the training data. Since KanjiVG is composed of Bezier curves corresponding to each stroke, these models can be used to easily generate character images lacking a stroke. Fig. 4 shows the examples of character images generated using KanjiVG.

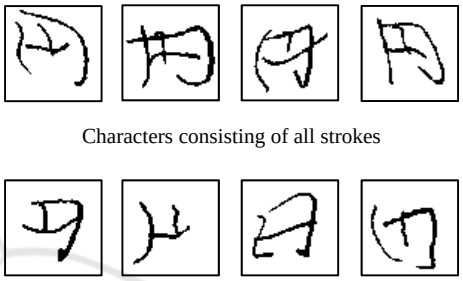


Figure 4: Examples of the character images generated using KanjiVG.

3.3 Construction of the CNN for Stroke Recognition

Fig. 5 shows the CNN that can identify the presence of each stroke in the characters to be recognized. Nodes in the output layer correspond to each stroke in the characters. Thus, the number of nodes in the output layer is the sum of the strokes in the characters to be recognized.

The CNN first learns to recognize the presence of strokes in the characters using the character images augmented with image processing. Since the nodes in the output layer correspond to each stroke in the characters, the teaching data are given as multi-hot vectors as shown in Fig. 6(a). We used Adam optimizer and binary cross-entropy as the loss function. In this step, the CNN cannot recognize the presence of each stroke but tries to select features in the images to recognize the characters.

Next, this CNN learns to identify the presence of each stroke in the characters using the character images generated with KanjiVG. The characters in these augmented images lack one of the strokes.

[1] KanjiVG https://kanjivg.tagaini.net/

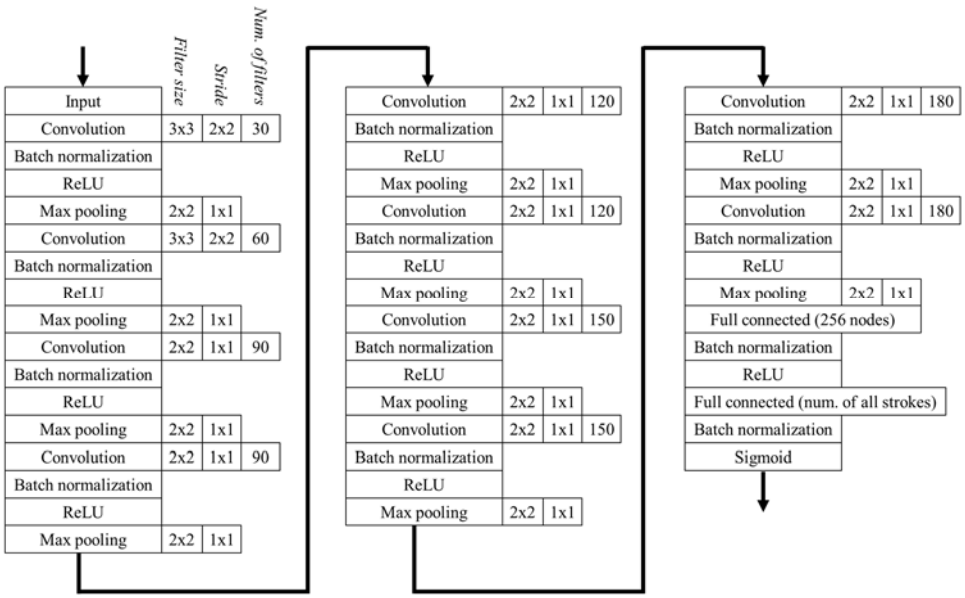
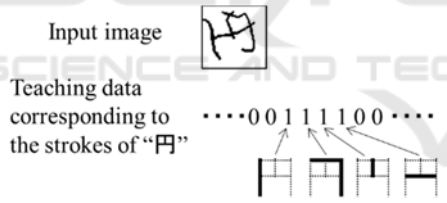
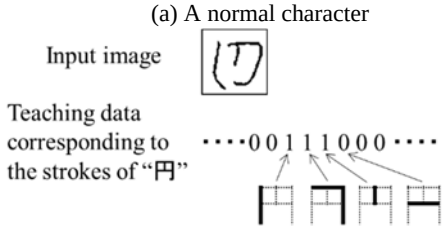


Figure 5: The CNN for identifying the presence of each stroke in the characters.

Along with this, our method uses teaching data where the value of the node corresponding to the lacking stroke is 0, as shown in Fig. 6(b). In this step, the CNN attempts to recognize the presence of each stroke by selecting the features obtained in the previous step.



(a) A normal character



(b) A character lacking one stroke

Figure 6: The teaching data in training data for learning the presence of each stroke in the characters.

3.4 Construction of the CNN for Character Recognition

As shown in Fig. 7, our method builds the CNN for character recognition by adding two dense layers to the output layer of the network obtained in section 3.3. The last of the added layers becomes the new output layer with nodes corresponding to each character to be recognized. Thus, the number of nodes in the output layer is that of the characters to be recognized.

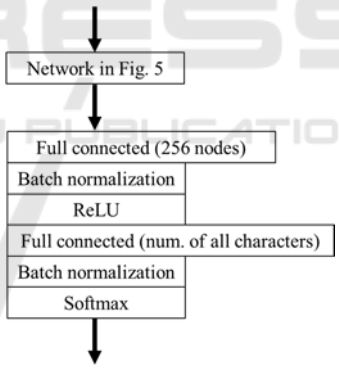


Figure 7: The CNN for recognizing the characters.

After transferring the weights obtained in section 3.3 to this network, our method learns to recognize the characters using the character images augmented through image processing. In this learning, the transferred weights are frozen and only the weights in the last two added layers are optimized. As shown in Fig. 8, the teaching data are given as one-hot vectors representing the corresponding characters. We used Adam optimizer and categorical cross-entropy as the loss function. In this step, the CNN attempts to recognize the characters by considering firing of the nodes corresponding to each stroke in the characters.

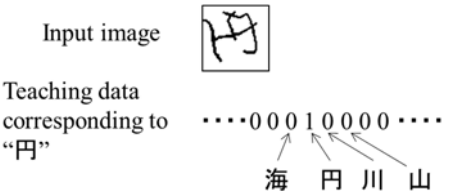


Figure 8: The teaching data in the training data for learning the characters.

4 EXPERIMENTAL RESULTS

In this section, we describe the experimental results of the proposed method. The proposed method is implemented on Intel core i7-6700 3.40GHz CPU with 16.0GB RAM and GeForce GTX1080 GPU chip. The character image augmentation method is coded in Python and uses OpenCV. The CNN for character recognition is implemented with Keras and Tensorflow in Python.

In this experiment, ETL9B was used as the handwritten character image dataset. This is a handwritten Japanese character database composed of 201 binary character images, sized 64x63 for each of the 3,036 character types, including Chinese characters called Kanji in Japan. In this experiment, 440 types of Kanji characters learned at an elementary school were used as the characters to be recognized, as shown in Fig. 9. The 201 images for each character type were divided into 120 training images, 60 test images, and 21 verification images. As shown in Table 1, a set of character images for learning and verification and their corresponding teaching data for stroke and character recognition were used for learning. We call them data set A.

Figure 9: Japanese characters to be recognized.

First, we increased the volume of training data by applying the projective transformation described in section 3.2 to the images in data set A. As shown in Table 1, the number of augmented character images for each character type is 50 times the number of its strokes. We call these along with their corresponding teaching data as set B. In data set B, 90% and 10% of the number of character images for each character type were used for training and verification, respectively.

Table 1: Data set for the experiments.

Data set	Types of the input images / Num. of images for each character
A	The original character images in ETL9B / 120
B	The character images augmented by applying projective transformations / 50 x the number of strokes in a character
C	The character images augmented using KanjiVG (one of the strokes is lacking) / 50 x the number of strokes in a character
D	The character images augmented using KanjiVG / 50 x the number of strokes in a character

Next, we increased the volume of training data by applying the character augmentation method using the character model KanjiVG described in section 3.2. The character images of data set C in Table 1 lack one stroke as shown in Fig. 4. In data set C, the number of the augmented character images for each character type lacking a specific stroke is 50. Therefore, the number of the augmented character images for each character type lacking a stroke is 50 times the number of its strokes. We also generated the character images consisting of complete strokes using the character model KanjiVG. We call these images and their corresponding teaching data as set D. In data set D, the number of the augmented character images for each character type is 50 times the number of its strokes. In data set C and D, 90% and 10% of the number of character images for each character type were used as training and verification data, respectively.

First, we compared the character recognition performance of the conventional CNN and the proposed method. The conventional CNN structure is exactly the same as the CNN shown in Fig. 5 except that each node in the output layer corresponds to each character to be recognized and uses the softmax function as the activation function. Table 2 shows the recognition rates of the conventional CNN using dataset A and dataset A + B. In order to avoid overlearning, we applied early stopping strategies to the learning phase using validation image data in all experiments. The recognition rate was obtained by testing 60 test images for each character in ETL9B described above. In addition, after constructing the CNN for stroke recognition described in section 3.3 by learning data set A + B + C, the CNN for character recognition described in section 3.4 was constructed by learning data set A + B + D. Its recognition rate is also shown in Table 2. From these results, it was confirmed that the recognition rate significantly improved after using the augmented character image.

However, the proposed method could not exceed the performance of the conventional CNN that had learned the data set A + B. This result was unexpected because our method is designed to overcome the traditional CNN for character recognition considering the positional relationships of the strokes. One possible cause is that the character images generated from the character models are unnatural, as shown in Fig. 4. The generation of more natural character images should be considered in future. Furthermore, it is possible that the hyperparameters of the CNN for stroke recognition were inappropriate. It is also necessary to adjust the appropriate values of the hyperparameters to correctly recognize a large number of strokes.

Next, we examined the firing status of the nodes corresponding to each stroke in the characters. Fig. 10 shows some of the results of recognized characters misclassified by the proposed method. In Fig. 10, we selected those results where the outputs of the nodes corresponding to the strokes in a desired character are unbalanced. Strokes surrounded by red circles indicate that firing was weaker than other strokes. These results show the possibility of identifying unclear parts in the character images by analyzing the firing status of the nodes in the middle layer. However, in most misrecognition results, all nodes corresponding to the stroke in the correct character did not fire. In the future, we think that it would be necessary to examine the structure of CNN that can recognize strokes more carefully.

Table 2: Recognition rates of the experiments.

Types of methods	Recognition rate
The conventional CNN obtained by learning A	97.8%
The conventional CNN obtained by learning A+B	99.7%
The proposed method obtained by learning A+B+C and A+B+D	99.5%

Figure 10: Firing status of the nodes corresponding to the strokes in the characters misclassified by the proposed method.

5 CONCLUSIONS

In this study, we proposed a method to investigate the cause of false recognition in offline handwritten character recognition using CNN. Our approach is based on a CNN which can recognize stroke structure by learning character images generated from stroke-based character models. The resulting CNN has nodes that can represent the presence of strokes, and can identify which stroke is the cause of the misrecognition. Consequent to the application of the proposed method to the Japanese character recognition problem, the possibility of identifying which stroke caused the misrecognition was confirmed to a certain extent.

However, since many misrecognitions were not caused by specific strokes, all nodes corresponding to the strokes in the desired character did not fire in most cases. Therefore, it was impossible to identify the cause of all misrecognitions by the proposed method. In order to explain the cause of misrecognition, it would be necessary to adopt a completely different approach. We think that the proposed method can point to mistakes in writing characters, and can be applied to support the writing of beautiful characters. In the future, we plan to improve the method assuming such types of applications.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant Number JP 19K12045.

REFERENCES

Bhattacharya, U. and Chaudhuri, B. B. (2009). Handwritten Numeral Databases of Indian Scripts and Multistage Recognition of Mixed Numerals. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(3):444-457

Chakraborty, S., Tomsett, R., Raghavendra, R., Harborne, D., Alzantot, M., Cerutti, F., Srivastava, M., Preece, A., Julier, S., Rao, R. M., Kelley, T. D., Braines, D., Sensoy, M., Willis, C. J. and Gurram, P. (2017). Interpretability of deep learning models: A survey of results. In Proceedings of IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 1-6.

He, M., Zhang, S., Mao, H. and Jin, L. (2015). Recognition confidence analysis of handwritten Chinese character with CNN. In Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR), 61-65.

Liu, M., Xie, Z., Huang, Y., Jin, L. and Zhou, W. (2018). Distilling GRU with Data Augmentation for Unconstrained Handwritten Text Recognition. In Proceedings of 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), 56-61.

Ly, N. A., Nguyen, C. T. and Nakagawa, M. (2018). Training an End-to-End Model for Offline Handwritten Japanese Text Recognition by Generated Synthetic Patterns. In Proceedings of 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), 74-79.

Miyazaki, T., Tsuchiya, T., Sugaya, Y., Omachi, S., Iwamura, M., Uchida, S. and Kise, K. (2017). Automatic Generation of Typographic Font from a Small Font Subset. In CoRR arXiv: 1701.05703.

Saabni, R. M. and El-Sana, J. A. (2013). Comprehensive synthetic Arabic database for on/off-line script recognition research. In International Journal on Document Analysis and Recognition (IJDAR), 16(3):285-294.

Shen, X. and Messina, R. (2016). A Method of Synthesizing Handwritten Chinese Images for Data Augmentation. In Proceedings of 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), 114-119.

Wigington, C., Stewart, S., Davis, B., Barrett, B., Price, B. and Cohen, S. (2017). Data Augmentation for Recognition of Handwritten Words and Lines Using a CNN-LSTM Network. In Proceedings of 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 639-645.

Zhang, X.-Y., Bengio, Y. and Liu, C.-L. (2017). Online and Offline Handwritten Chinese Character Recognition: A Comprehensive Study and New Benchmark. In Pattern Recognition, 61(1):348-360.