Automatic Font Generation For Early-Modern Japanese Printed Books

Y. Takemoto[1], Y. Ishikawa[1], M. Takata[1], and K. Joe[1]
[1]Nara Women's University

Abstract – *The National Diet Library makes images of early-modern Japanese printed books public on the Internet so that anyone can read the valuable books, which have been published during Meiji era to early Showa era. For the efficient and convenient use of the data, it is necessary to convert image data into text data. However, it is difficult to recognize character images of early-modern printed books correctly by existing OCRs. The multi-fonts character recognition method for early-modern printed books has been proposed. The method requires huge number of character images of those old books while the current methods to collect character images are very limited. In this paper, we propose a new method to complement learning data for the multi-fonts character recognition method by automatically generating character images with deep learning. For that purpose, we construct the neural network that automatically generates character images which have the features similar to the corresponding characters to be found in other early-modern Japanese printed books.*

Keywords: Font Generation, Deep Learning, Character Recognition, and Early-Modern Japanese Printed Books

1 Introduction

The book of the old days is one of means to learn history, culture, thought and so on of the time. The books published during Meiji era to early Showa era are called early-modern Japanese printed books. Because almost all of early-modern Japanese printed books are out of print, they are very variable and difficult to obtain. Therefore, National Diet Library [1] makes image data of those books public on the Internet [2]. That makes it easy for anyone to read early-modern Japanese printed books without worrying about damage, loss, or theft like paper books. However, in image data, it is impossible to perform a full text search for the books. In order to improve the convenience, it is necessary to convert image data into text data as soon as possible.

A typical way to convert characters image data into text data is optical character recognition. However, accurate recognition is difficult with commercially available OCRs because of several reasons. First of all, heterogeneous and old character types are often used in early-modern books. Second, fonts used in early-modern books differ depending on publishers and published ages. So a huge number of fonts used on early-modern books are required. Furthermore, early-modern books are printed by typographical printing. In typographical printing, metal types, which construct carved characters, are painted with ink and pressed on papers to print the carved characters. As a result, even if the same character is printed, slight differences arise due to ink spreading and rubbing each time printing is performed as shown in Fig. 1. Therefore, the multi-fonts character recognition method for Early-Modern Japanese Printed Books has been proposed [3][4][5]. The method requires various character images with various fonts of early-modern books as learning data. The current method to collect learning data is to clip character images from early-modern books manually while manual collection is limited in terms of cost. Publishers of early-modern books tend to issue their books by field. Since there are many cases of personal or private publish, the number of font types used for the publication is over 20,000. In addition, the number of collectable character images for the type of a font is at most 2,000 from our experience. That is based on Zipf's law [6][7]: the number of elements whose appearance frequency is k-th is proportional to $1/k$ of the total. Namely, characters with lower appearance frequency are printed fewer times in the entire books. Therefore, it is very difficult to collect such rarely appearing characters. In addition, characters that are never printed in a book cannot be collected from the book because there is no metal type. Furthermore, some of collected character images include the loss of features because of ink spreading or rubbing. Such images are inappropriate for learning data. Examples of such inappropriate character images are shown in Fig. 2. In the image on the left side, the character is collapsed with ink. In the image on the right side, a part of the character is missing with ink rubbing. Thus, in order to improve the accuracy of character recognition, a method for efficiently increasing learning data is required. In this paper, we aim to complement learning data of the multi-fonts character recognition method by automatically generating character images with deep learning. Generated images have the similar features to the font of learning data in terms of publisher and published age. We construct a neural network that automatically generates character images whose fonts are not found by giving learning data whose fonts are available.

The structure of this paper is as follows. In section 2, we introduce research reports on image generation. In section 3, we propose a method to automatically generate character

Fig. 1 Character images in early-modern Japanese printed books.

Fig. 2 Character images inappropriate for learning data.

images of a font in the particular publisher and the published age of early-modern Japanese printed books with deep learning. In section 4, we explain the experiment to verify the usefulness of the constructed neural network and show results.

2 Related Works

In this section, we introduce several research reports on image generation. For image generation research, deep learning, which is one of machine learning method using multilayered neural networks, is often used. It contributes to the achievements of various research [8][9][10].

There is an example that succeeded in complementing learning data with images automatically generated by a neural network called GAN [11]. The characteristic of GAN is to learn two neural networks, called Generator and Discriminator, together. Generator automatically generates images which have features learned from the learning data. Discriminator judges the input image is given or generated by Generator. Making two neural networks learn together enables Generator to automatically generate images with higher reproducing accuracy. In this example, a neural network automatically generates images to complement learning data. Since learning data is object images whose shape is not clearly defined, they are used for GAN. In order to make Generator learn so that it can generate an image from a noise of random numbers, GAN requires huge amount of learning data and computational cost. In this paper, since we have concrete target images of early-modern Japanese printed books, it is not necessary to use GAN.

The convolution neural network is sometimes used for image generation. It extracts features from input image. The features are transformed into an image by the deconvolution neural network. With those neural networks, it is possible to learn features from learning data and automatically generate various kinds of image. There is a service to color a line drawing automatically by using those neural networks [12]. In the research to automatically generate images of objects [13], for example, if images of chairs are used as learning data, images of chairs with various shapes and colors can be generated. There is also a research report to automatically generate room images and face images with DCGAN, which is a GAN extension with the convolution neural network [14]. In the research of artistic style transformation, the artistic style is extracted from the landscape photography by the

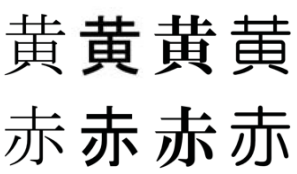


Fig. 3 Character images of various fonts

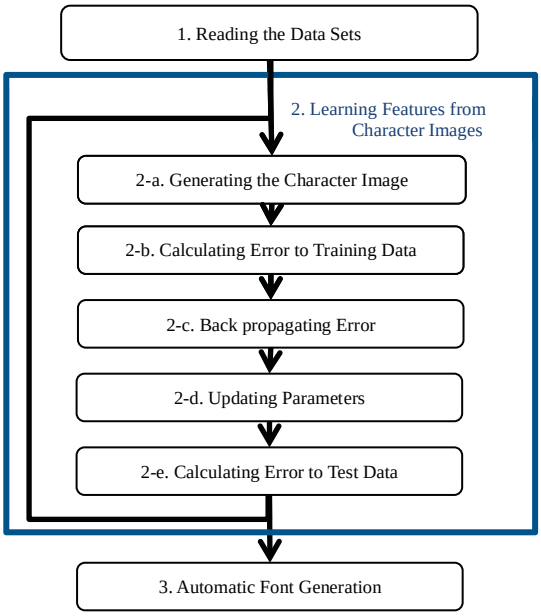


Fig. 4 The algorithm of automatic font generation

convolution neural network, and images of the landscape applied with the style are generated [15].

In this paper, we extract some features from character images by using a convolution neural network. We aim to generate character images by converting the font with a deconvolution neural network based on the extracted features.

3 A Method of Automatic font Generation with Deep Learning

3.1 The algorithm of automatic font generation

In order to convert the font of the character image, we focus on the features that the character image has. Fig. 3 shows character images in which two types of characters are depicted with four kinds of fonts. We find two sets of features construct the character image: Character specific feature and font specific feature. The shape of the entire character depends on the kind of the character while the shapes of points and lines making up a character depend on the kind of font. It is necessary for our font conversion of character images to transform just the font specific feature whith keeping the character specific feature.

We use deep learning to transform the font specific feature. The process for the automatic generation of character

images of the particular font of early-modern Japanese printed books with deep learning is described as follows. The flow chart is shown in Fig. 4.

1. Read the data sets
2. Learn features from character images
    a. Generate character images
    b. Calculate the errors of the training data
    c. Back propagate the errors
    d. Update parameters
    e. Calculate the errors of the test data
    f. Go to b. until convergence
3. Automatic font generation
    a. Read the character data
    b. Generate the character image

We read the data sets of character images in step 1, and divide the data sets into training data and test data. Training data is used for learning the neural network. Test data is used for evaluating the reproducing accuracy of the neural network against unknown data. Next, learning of the neural network is performed in step 2. In step 2-a, the character image with transformed font is generated from the character image of training data. In step 2-b, the errors between the automatically generated image and the character image of an early-modern Japanese printed book are calculated. In step 2-c, the errors calculated in step 2-b is back propagated to the neural network. In step 2-d, parameters are updated. In step 2-e, the errors of the automatically generated image from the test data are calculated to evaluate the neural network. The steps 2-a to 2-e are repeated until the learning is converged. After learning, character images with font transformation can be automatically generated using the neural network with the high reproducing accuracy against test data in step 3. In step 3-a, character images as a source of font generation are read. In step 3-b, character images with the transformed font are generated. Details of each process are explained in the following sections 3.2 to 3.4.

3.2 Read the data sets

The data used for the neural network learning consists of two character image sets: the character image as a source of the font generation and the character image of the font in a particular publisher and published age of early-modern Japanese printed books. The font used as a source of the font generation has to satisfy the following three conditions.

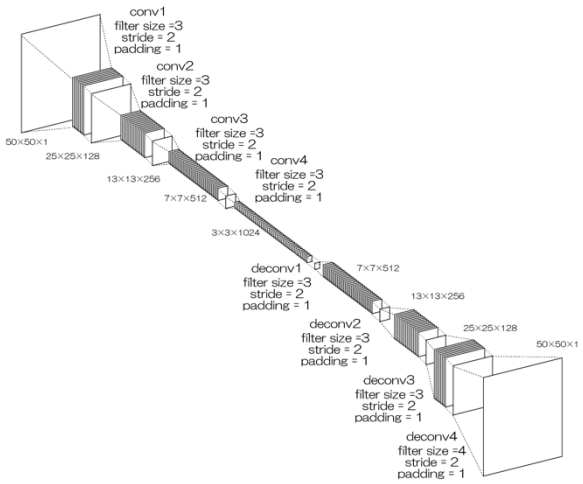


Fig. 5 The neural network that automatically generates character images converted the font.

- Easy to obtain
- Enough kinds of available characters
- Include heterogeneous and old-fashioned characters

In this paper, we use the Gothic type, which is a widely used standard Japanese font type including more than 6,000 characters, as a font that fulfills the above conditions.

Since the image format of the multi-font type recognition method is based on bmp, the character images to be prepared and generated images by the neural network are also in the bmp format. The character images are to be binarized and the image size is to be unified to 50 pixels vertically and horizontally. The data sets are divided into training data and test data.

3.3 Learning features from character images

Using Gothic font character images, the neural network is to be trained so that it automatically generates character images of the particular font. The neural network constructed in this paper is shown in Fig. 5. The middle layer of the neural network consists of 4 convolution layers and 4 deconvolution layers. We use "deconvolution" in the meaning written in [12]. In the both layers we adopt the setting stride of 2 for pooling and unpooling. In order to unify the size of the output image and input image, the padding in all layers is set to 1, and the filter size of the last deconvolution layer is set to 4. The filter size of all layers except the last deconvolution layer is set to 3. We use the ReLU function expressed by the following equation as the activation function.

$$f(x) = max(0, x) \quad (1)$$

We apply the dropout function [16] to all convolution layers. The dropout function randomly excludes the neurons of connected layer with a fixed rate. The dropout function has an effect to prevent overfitting where a neural network has excessively learned features of training data to fetch in meaningless features of individual training sample. When the overfitting occurs, the neural network can reproduce all the features of training data in perfect while it becomes very difficult to reproduce the features of unknown data. Though the dropout in the higher rate can prevent overfitting, the neural network reproduces the features of training data with a less reproducing accuracy. Therefore, it is necessary to adjust the dropout rate with learning appearance.

In order to generate a character image closer to the character image of early-modern Japanese printed books used for learning data, the error between the generated character image and the original character image of early-modern books is to be calculated. The error is back propagated from the output layer to the input layer of the neural network, and parameters are updated so that the error becomes smaller. The error calculation in this paper is the root mean square error of the pixel values. The root mean square error is calculated by the following equation.

$$RMSE(z) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z - z_i)^2} \quad (2)$$

where z is a true value, $z_i(i = 1, 2, \dots, n)$ quantitatively expresses the distribution state of the measurement values.

From the error back propagated, new parameters are to be calculated with some learning optimization methods. In this paper, Adam [16] is used as a learning optimization method. Though there are various learning optimization methods, Adam uses less memory and converges learning faster. The equation for updating parameters by Adam is shown as follows.

$$g_t = \nabla_\theta f_t(\theta_{t-1}) \quad (3)$$

$$\theta_t = \theta_{t-1} - \alpha\, \hat{m}_t / \sqrt{\hat{v}_t} \quad (4)$$

$$\hat{m}_t = m_t / \left(1 - {\beta_1}^t\right) \quad (5)$$

$$\hat{v}_t = v_t / \left(1 - {\beta_2}^t\right) \quad (6)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (7)$$

$$v_t = \beta_1 v_{t-1} + (1 - \beta_2) {g_t}^2 \quad (8)$$

where $g_t$is the gradient of the target function $f_t(\theta_{t-1})$, θ is the learning parameter, α is the learning rate, $m_t$and $v_t$are approximate values of the first moment and the second moment of the gradient, and β_1 and β_2 are parameters which reduce influence by past gradients.

By repeating learning, we obtain models of the neural network with various parameters. The most suitable model for automatic font generation is with the highest reproducing accuracy for unknown data. After learning, in order to select the most suitable model, we evaluate the reproducing accuracy to test data. Therefore, we calculate the error between the character image automatically generated from test data and the same kind of character images from early modern books. The error is the root mean square error of the pixel values as shown in equation (2). Therefore, the model that minimizes this error is the most suitable model for automatic font generation.

3.4 Automatic font generation

After learning, the neural network can generates character images with transforming the font found in an early modern printed book. We use the neural network model with the highest reproducing accuracy to test data. When we input any Gothic font type character image to the neural network, we obtain the character image whose font is transformed into the font that is learned by the neural network. Therefore, it becomes possible to automatically generate character images, which are difficult to collect manually, by preparing Gothic font type character images. This makes it possible to complement the learning data of the multi-font type recognition method with automatically generated character images. We expect to improve the recognition rate of the multi-font type recognition method by complementing the learning data that have not been obtained so far.

4 Experiment

4.1 Method of experiment

In order to verify the effectiveness of the proposed method, we perform the experiment of automatic font generation. Character images with a particular font of an early modern printed book are automatically generated and we judge whether the generated images can be used as learning data of the multi-fonts character recognition method. The font learned through the experiment is taken from several books published by Shinshindo (publisher) in the middle of Meiji era (1880's to 1890's). Hereafter, we call this font as MMS (Middle Meiji Shinshindo) font. The data set used for learning is character images of Gothic font and MMS font. We prepare 1,297 pairs of data sets, each of which constitutes two character images. We divide the data set into training data and test data. 1,200 pairs selected randomly are used as training data, and the remaining 97 pairs are used as test data. The learning is performed 5,000 epochs and the dropout rate is set to 0.7.

After learning, we compare generated character images with the MMS font images to judge whether generated character images are available as learning data of the multi-fonts character recognition method. The comparison criterion is the Euclidean distance of the PDC (peripheral direction

Table. 1 Result of font generation using training data

Character images of Gothic font	Character images of MMS font	generated character images
遙		
光		
禮		
猫		

Table. 2 Result of font generation using test data

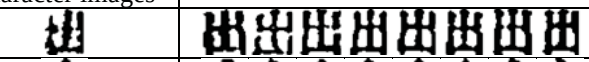
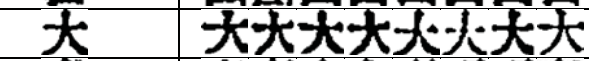
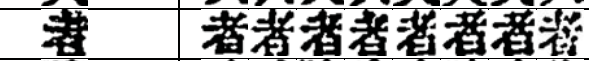
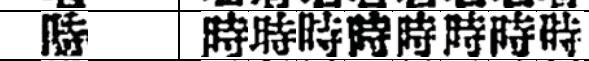
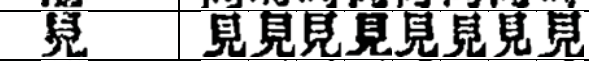
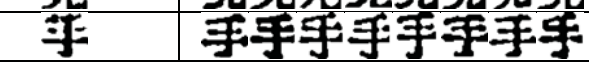
Character images of Gothic font	Character images of MMS font	generated character images
紀		
公		
嬉		
家		

contributivity) features [18] extracted from both images, which is often used for the recognition of Japanese handwritten characters. The reason why we use the criterion is that the PDC feature is used for our multi-fonts character recognition method as a vector of 1,536 dimensions. Character images of early modern printed books have different influences of ink spreading and rubbing at every printing. Therefore, in the PDC feature vector space, even though several character images are taken from the same printed books namely with the same font, the PDC features of those images do not match completely. The distribution of the differences among the above PDC feature vectors is considerably large but within a certain error range. If the PDC feature vectors of generated character images are within this error range, we can assume that the generated character images are available as learning data of our multi-fonts character recognition method.

4.2 Results

We show the experiment results of automatic font generation of the neural network model with the highest reproducing accuracy to test data. Table. 1 shows examples of character images of Gothic font, MMS font and generated font using training data. Table. 2 shows examples of character images of Gothic font, MMS font and generated font using test data. We observe that generated character images from training data almost completely reproduce the character images of FFS font. Although the generated character images from test data have noises, we find that the image features of Gothic font that is used for the source of font-conversion are nicely transformed into the features of MMS font.

Table. 3 List of character images to compare the PDC features

generated character images	Character images of MMS font
出	
大	
者	
時	
見	
手	

In order to check whether the neural network can automatically generate character images close to character images of early modern printed books, we compare pixel values of those character images. As a result, the average matching rate of pixel values in training data is 99.79% while the rate in test data is 73.69%. The average matching rate of pixel values in test data is obviously lower than in training data. The reason is that the neural network could not accurately reproduce unknown features not found in training data. However, character images clipped out from early modern printed books are extremely affected by spreading and rubbing of ink, and sagging of pages when photography. Even with the same character, the thickness of the line and the distortion of the shape differ for each character image. Therefore, even if character images that have features of MMS font were automatically generated, the pixel matching rate of sometimes decreases.

In order to determine if automatically generated character images are available as learning data of our multi-fonts character recognition method, we compare the PDC features. We choose six kinds of character images out of character images generated from test data to compare the PDC feature with character images of MMS font. The selected six characters have various numbers of strokes and various shapes. Table. 3 shows the character images of the selected six characters to compare. Prepared character images of MMS font include 51 images of "出", 68 images of "大", 33 images of "者", 36 images of "時", 39 images of "見", and 32 images of "手". We explain the way to compare the PDC features. First, we calculate the PDC features of generated character images and MMS font character images. Next, we get two kinds of Euclidean distances from these PDC features. One is the Euclidean distance between generated character image and optional character image of MMS font. The other is the Euclidean distance between optional two character images of MMS font. We call the first one Euclidean distance 1 and the second one Euclidean distance 2. If the distribution range of Euclidean distance 1 exists within one of Euclidean distance 2, we can assume that generated character images are available as learning data of our multi-fonts character recognition method. The histograms of calculated two kinds of Euclidean distances for six kinds of

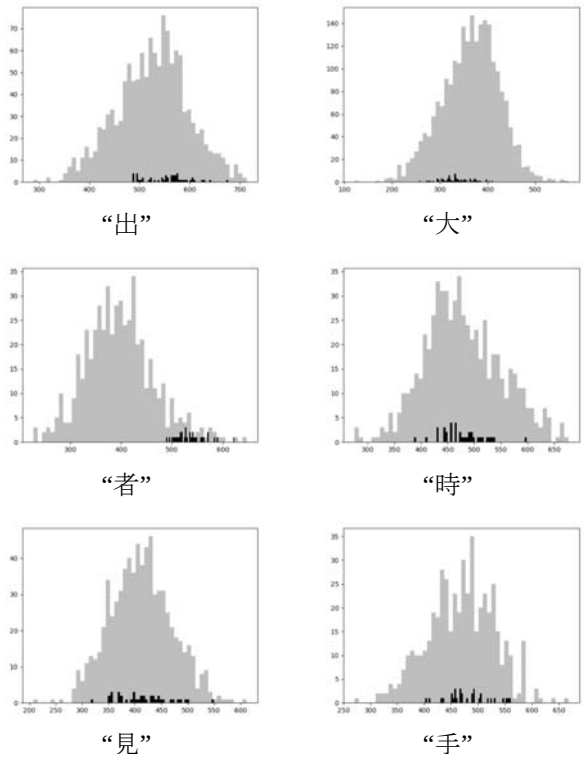


Fig. 6 Histograms of two kinds of Euclidean distances for six kinds of characters

Table. 4 Examples of character images that are failed to convert the font.

Character images of MMS font	generated character images
十	
寧	
云	
崎	

characters are shown in Fig. 6. The value of the horizontal axis is the Euclidean distance, and the value of the vertical axis is the frequency. From the histogram, we find that for all the six kinds of characters the distribution range of Euclidean distance 1 exists within one of Euclidean distance 2. Therefore, the proposed method is confirmed to be automatically generate character images that can be used for learning of our multi-fonts character recognition method.

In the meanwhile, not all character images automatically generated by the neural network are available for learning data. Some of these character images cannot be correctly transformed to the target font. Table. 4 shows examples of character images that are failed to be transformed to the MMS font. All of those character images collapse or loss the shape of characters. In this case, the neural network cannot reproduce these characters with the features learned from training data. In order to solve this problem, it is necessary to increase learning data or improve the structure of the neural network.

5 Conclusions

In this paper, we propose a method to complement learning data of the multi-fonts character recognition method with character images generated by a neural network. For this purpose, we construct the neural network that automatically generates character images, which have features of a particular publisher and published age of early modern printed books. Convolution layers and deconvolution layers constitute the neural network. When a character image is inputted to the neural network, the character image with the transformed font is outputted. The transformed font is the particular font of early modern printed books used for learning data. Furthermore, an easily obtainable font is used for the input image. The data set used for the learning is a pair of character images: with the font of early modern printed books and the easily obtainable font. We divide the data set into training data and test data. Training data is used for learning the neural network, and test data is used for evaluating the reproducing accuracy of the neural network. After learning, the model with the highest reproducing accuracy to test data is the most suitable neural network for automatic font generation.

In order to verify the effectiveness of the proposed method, we perform the experiment of automatic font generation. For the font to learn by the neural network we adopt publishing books by Shinshin-do in the middle of the Meiji era (MMS font). In addition, we use Gothic font as the font for input image. 1,297 pairs of character images of Gothic and MMS font are prepared as the data set. 1,200 pairs randomly selected from the data set are used as training data, and the remaining 97 pairs are as test data. After learning, we judge whether character images generated from the model with the highest reproducing accuracy to test data can be used for learning data of the multi-fonts character recognition method where the PDC feature is used to represent the feature of character. Therefore, we compare the PDC features of generated character images and of MMS font.

We show the result of the experiment of automatic font-generation. Comparing the pixel values of character images, the average matching rate is 99.79% for training data and 73.69% for test data, respectively. The average matching rate for test data is extremely lower than for training data. The reason is that the neural network couldnot reproduce unknown features not found in learning data. However, even with the same character, the thickness of the line and the distortion of the shape differ for each character image in real early modern printed books. For this reason, even though character images with the feature of MMS font can be

generated, the pixel matching rate decreases in some character images. Therefore, in order to judge whether generated character images are available as learning data of the multi-fonts character recognition method, we compare the PDC features. We select six kinds of character images out of character images generated from test data by the most suitable model. As a result, we find that generated character images contain the features of MMS font in all six kinds of characters. This shows that generated character images can be used as learning data of the multi-fonts character recognition method. We consider that complementing learning data of the multi-fonts character recognition method enables to correctly recognize characters of early modern Japanese printed books. However, the neural network of the proposed method cannot correctly generate some character images the transformed font. In order to generate more character images correctly, it is necessary to increase the amount of learning data or improve the structure of the neural network.

6 References

[1] National Diet Library. http://www.ndl.go.jp. Accessed 2018-03-30.

[2] National Diet Library Digital Collection. http://dl.ndl.go.jp. Accessed 2018-03-30.

[3] Chisato Ishikawa, Naomi Ashida, Yurie Enomoto, Masami Takata, Tsukasa Kimesawa and Kazuki Joe. "Recognition of Multi-Fonts Character in Early-Modern Printed Books". Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA09), Vol. Ⅱ, pp. 728-734(2009).

[4] Manami Fukuo, Yurie Enomoto, Naoko Yoshii, Masami Takata, Tsukasa Kimesawa and KazukiJoe. " EvaluaTion of the SVM based Multi-Fonts Kanji Character Recognition Method for Early-Modern Japanese Printed Books". Proceedings of The 2011, International Conference on Parallel and Distributed Processing Technologies and Applications (PDPTA2011), Vol. Ⅱ, pp. 727-732(2011).

[5] Taeka Awazu, Kazumi Kosaka, Masami Takata and Kazuki Joe. "A Multi-Fonts Kanji Character Recognition Method for Early-Modern Japanese Printed Books". Information Processing Society of Japan TOM, Vol. 9(2), pp.33-40(2016).

[6] Zipf, G. K. "The Psycho-Biology of Language, Boston-Cambridge Mass". Houghton Mifflin. 1935.

[7] Zipf, G. K. "Human Behavior & The Principle of Least Effort, An Introduction to Human Ecology". Addison-Wesley Press Inc. 1949.

[8] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. "Mastering the game of Go with deep neural networks and tree search". Nature 529, pp.484–489, 2016.

[9] Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeff Dean, and Andrew Y. Ng. "Building High-level Features Using Large Scale Unsupervised Learning". ICML, 2012.

[10] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu and G. Zweig. "Achieving Human Parity in Conversational Speech Recognition". MSR-TR-2016-71, 2016.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair‡, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets". NIPS, 2014.

[12] PaintsChainer. https://paintschainer.preferred.tech/index_ja.html. Accessed 2018-03-30.

[13] Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, and Thomas Brox. "Learning to Generate Chairs, Tables and Cars with Convolutional Networks". CVPR, 2017.

[14] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". ICRL, 2016.

[15] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. " A Neural Algorithm of Artistic Style". CVPR, 2017.

[16] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". Journal of Machine Learning Research 15 (2014) 1929-1958.

[17] Diederik P. Kingma, and Jimmy Lei Ba. "Adam: A Method for Stochastic Optimization". ICLR, 2015.

[18] Norihiro Hagita, Seiichiro Naito and Isao Masuda. "Handprinted Chinese Characters Recognition by Peripheral Direction Contributivity Feature". The Institute of Electronics, Information and Communication Engineers, Vol.J66-D, No.10, pp.1185-1192(1983).

Acknowledgment

This work is partially supported by Grant-in-Aid for scientific research from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT) No. 17H01829.