Data Augmentation for Ancient Characters via Semi-MixFontGan

Zhiyi Yuan
Graduate School of Information,Production and Systems
Waseda University
Kitakyushu, Japan
yuanzhiyi@akane.waseda.jp

Sei-ichiro Kamata
Graduate School of Information,Production and Systems
Waseda University
Kitakyushu, Japan
kam@waseda.jp

***Abstract*—The ancient documents provide people a way to understand history. However, the existing materials are suffering from unbalanced characters dataset, as well as intra-class multimodality fonts. As a result, humans and recognition systems are unable to identify these characters effectively. Based on these problems, we propose Semi-MixFontGan: a font generation method based on Semi-Supervised strategy that can learn from a small number of labeled font data to aggregate subclasses' information of categories and generate characters. In generating new samples from ancient books that have a small amount of labeled font data, the model can automatically learn the difference between them and generate font-consistent characters. The model is composed of two parts. In the first part, we propose a MixFont method to mix labeled and unlabeled and generated data. Then use a convolutional autoencoder to learn the font information. In the second part, the generator network can generate reasonable and realistic images by Font and Content Discriminator. Through this model, we can make the ancient book dataset more balanced. Experiments show that the generated characters by our model can get good visual effects and maintain font consistency with training data. With the augmented data, the accuracy of the recognition network has increased.**

***Contribution*—**We propose a novel font generation method with semi-supervised learning to generate characters from small labeled font Kuzushiji dataset.

***Keywords*—GAN, Style Transfer, Semi-Supervised Learning**

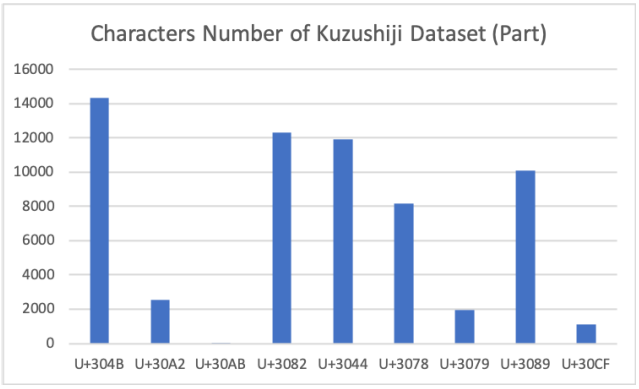

Fig. 1. Distribution of 10 classes of Kuzushiji Dataset

Fig. 2. Multi-modality in Left: U+304B Right: U+306E

I. Introduction

Japan has millions of ancient books and over a billion historical documents, such as personal letters or diaries preserved nationwide. However, most of the Japanese people nowadays cannot read these books because they are written in "Kuzushiji" which were deprecated in the 1900s. Moreover, history books are lack of protection [1]. Due to the lack of sufficient human resources, people are interested in using deep learning methods to identify these historical documents automatically. In Japanese history books, the writing style changes significantly, while there are also degradation, character overlap, and other difficulties. Furthermore, languages like Chinese and Japanese, which have a massive number of characters and intricate strokes, are harder to recognize.

In the handwritten character recognition system, a large number of unbalanced and unmarked datasets lead to system performance degradation [2]. Since Kuzushiji requires expert knowledge to identify font labels, the acquisition is expensive and time-consuming. The semi-supervised learning model is used to make use of unmarked data and reduce the need for marked data [3]. As shown in Fig.I, in the Kuzushiji dataset of historical documents provided by the Japanese authorities, the data distribution is very uneven. For example, the number of characters (U+304B) is as high as 14,342, while some characters (U+30AB) are only 17. Wang *et al.* pointed out that insufficient training data would lead to a decrease in the recognition rate [4]. That is, there is no way for recognition systems to learn enough pattern information from small or unbalanced amounts of data.

To overcome the existing drawbacks of unbalanced character data occur in the historical documents, a reasonable method of data augmentation is needed. Traditional data augmentation methods are mainly used to deform data images, such as size, inversion, blur, and rotation. This kind of operation is feasible for natural images, but based on the spatial particularity of text symbols, a small amount of rotation cannot add meaningful data, and robust operation such as inversion may cause the change of glyph meaning. Aside from this, another problem in the historical books is that a single character often con-

978-1-7281-9331-1/20/$31.00 ©2020 IEEE

tains many different fonts, which is known as the intra-class multimodality problem, as shown in Fig.2.

In this paper, we propose a semi-supervised encoder and a font character generation model which can well generate clear and consistent samples from a small amount of intra-class multimodality data to augment the dataset. The previous supervised networks require a paired dataset. However, due to the lack of fonts labels in the Kuzushiji dataset, we use a small number of datasets with manual labels for training. A special "multi-tasks" encoder is used to learn tagged datasets for fonts classification and information extraction, and in combination with glyphs to generate stylized characters. After a period of learning, the network begins to send the unlabeled data and generated images into the encoder for classification and labeling and then continue to train the network by mixing these three types of data with a unique mixing method. Finally, the classification results of all datasets are obtained, and the generated results can be used for data augmentation.

We have three main contributions in our work:

1) Propose a semi-supervised font generation model to create samples from a small labeled font dataset.
2) We reconsider a Semi-Supervised Learning(SSL) method to mix the labeled and unlabeled and generated data in order to improve the encoder's extraction and classification performance.
3) Our experiments show better performance than the previous historical character data augmentation method.

II. RELATED WORK

A. Semi-Supervised Learning

In recent years, the number of data has increased a lot due to the rapid development of the Internet, but clean labeled data is difficult to obtain. At present, deep learning methods need to use a large amount of labeled data, so semi-supervised learning using labeled data mixed with unlabelled data is gradually becoming popular [5]. Lee proposed a method to make the network work in semi-supervised fashion [6]. The network's prediction of unlabeled data is used to train the network as a label of unlabeled data (pseudo label). This cost of tableless data achieves a regularization effect, which reduces the overfitting of the network under limited labeled data, and makes the network generalization better. But the number of unlabeled data addition needs to be carefully controlled. To construct a better pseudo label, Tarvainen *et al.* proposed a way to move average model parameters [7]. Since the integration of multiple independently trained networks can achieve better prediction, better pseudo-tags can be constructed by integrating the same model under different iteration periods, different data enhancement, and regularization conditions. However, it is relatively time-consuming to predict the same input twice in each iteration under different regularization and data enhancement conditions.

MixMatch [8] method improves the previous algorithm by first augmenting the labeled and unlabeled data, and then classifying the expanded unlabeled data using the entropy algorithm for entropy reduction, and finally mixing the two datasets. But since the augmented unlabeled data is forced to self-consistently regularize, the predicted results are meaningless. In order to overcome the lack of font labels in Kuzushiji dataset, we propose a new hybrid approach in the unsupervised strategy.

B. Font Generation via Deep Learning Methods

The current work of generating characters contains different ways. Variational AutoEncoder(VAE) [9], in the way of reconstruction of input images, the potential representation of data is obtained, and the learning of small data is supervised, but fuzzy boundaries are generated. To avoid this, Conditional AutoEncoder(CVAE) [10] was proposed to separate the font from the class label, which eventually learns the gaussian distribution of fonts, while class labels are manually controlled. However, due to the uncertainty of the writing style of historical documents in Japan, this is not an ideal character generation model. Generative Adversarial Networks (GANs) were originally proposed by allowing two-game between neural network approach to learning [11] [12], which can generate a good visual image. Combined with class labels, Mirza *et al.* proposed a Conditional GAN(CGAN) to handle multimodality learning [13]. Combining CAVE with CGAN, CVAE-GAN can produce a more precise and more realistic font image [14], but it also faces the same problem of extensive class label dataset requirements. However, writing fonts in Japanese historical documents have no labels.

Azadi *et al.* proposed a GAN-based font transfer network to generate letters of the same font as a given letter [15]. This is a domain-adaptive approach that effectively migrates missing letters from glyphs, but still requires a lot of stylistic characters to learn. Zhang *et al.* proposed a set of font and content encoder [16], which uses a small number of reference images to extract the information of font and glyph and generate the glyph with target font. However, due to its simple structure, it is easy to generate meaningless images. Inspired by these works, our font generation model uses a semi-supervised approach and a new network structure to combine glyphs and fonts for reconstruction effectively.

III. PROPOSED METHOD

In this section, we describe the detailed architecture of the proposed Semi-MixFontGan system that contains three major components, data mixing part, font feature extraction network, and font transfer network.

The architecture of our model is shown in Fig.3. To be specific, the MixFont part tries to mix labeled, unlabeled, and generated data in a unique method, and the font feature extraction network classfies mixing data and extracts font features. In this way, the font transfer network can generate font characters based on font features and content images.

A. MixFont

Since the Kuzushiji dataset does not contain font labels, we manually annotate about 10% of the data, learning the entire dataset in a semi-supervised way.

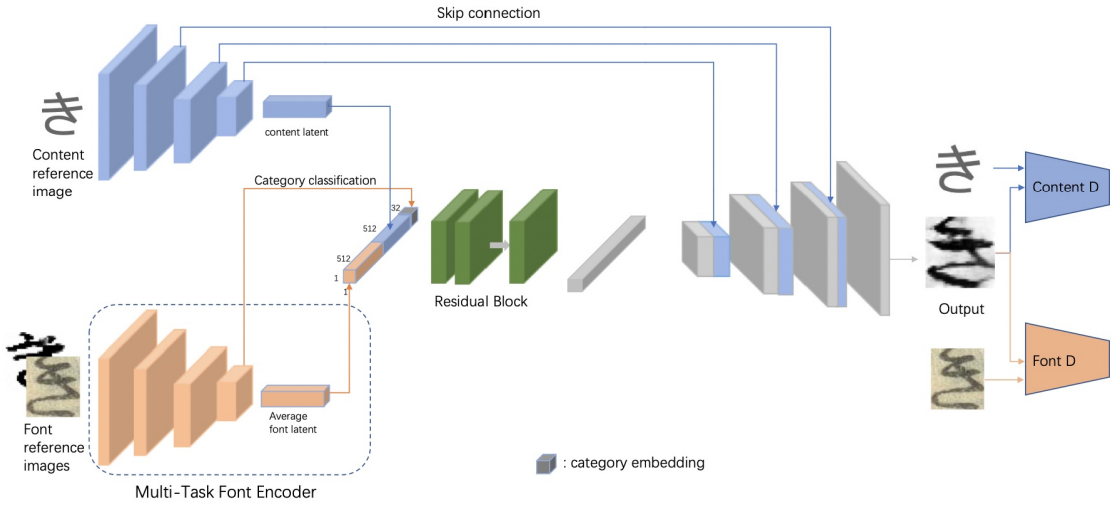

Fig. 3. The architecture of the proposed MixFontGan system that migrates font image which is combined with labeled, unlabeled and generated images information to content image. We use a font encoder to estimate the depth characteristics and categories of the characters, then connect the content and font representations with category embeddings, and then embeds the categories into the remaining blocks and decoders to generate synthetic handwriting.

Based on the recent MixMatch [8] method, we expand the way that we mixed the data so that the generated font images could also be added to the dataset for training.

For recognition system of identifying intra-class font, we define $\mathcal{X}$ is a batch of labeled data and an equally-sized batch $\mathcal{U}$ of unlabeled examples and generated font images $\mathcal{Z}$ which are classified by multi-tasks font encoder. Using augmentation method to generate a batch of augmented label examples $\mathcal{X}'$ from $\mathcal{X}$. At the same time, using augmentation method to generate a batch of augmented unlabeled examples with "guessed" labels $\mathcal{U}'$ from $\mathcal{U}$. Since $\mathcal{Z}$ is generated data, we also think it as unlabeled and needs to be guessed. Different augmentation method for data should always get consistent prediction.

$$\mathcal{X}', \mathcal{U}', \mathcal{Z}' = MixFont(\mathcal{X}, \mathcal{U}, \mathcal{Z}, T, K, \beta) \tag{1}$$

Mix(·) means the whole algorithm process.

For labeled data, we will use traditional method to get classification prediction, denoted as p. For unlabeled data, implement K times random augmentation excludes crop and flip. Input augmentation data to classifier to get average prediction and apply temperature sharpening (T is temperature parameter) to receive "guessed" label q. Same as for generated data, predicted label as r. For this point, we have a batch of augmented labeled and generated data $\hat{\mathcal{X}}$, K batches of augmented unlabeled data $\hat{\mathcal{U}}$ and augmented generated data $\hat{\mathcal{Z}}$. Combine $\hat{\mathcal{X}}$ and $\hat{\mathcal{U}}$ and $\hat{\mathcal{Z}}$, and shuffle them to get dataset $\mathcal{W}$. The output is a batch of $\mathcal{X}'$ which is implemented Eq.(2) on $\hat{\mathcal{X}}$ and $\mathcal{W}$. For two kinds of examples and their labels (x_1, p_1) and (x_2, p_2), the mixing examples can be described as:

$$\begin{aligned} x' &= \lambda' x_1 + (1-\lambda') x_2, \\ p' &= \lambda' p_1 + (1-\lambda') p_2. \end{aligned} \tag{2}$$

where weighting factor λ' is produced by Beta function using hyperparameter β. This equation allows model can linearly process regions between samples.

The combined loss $\mathcal{L}$ for semi-supervised learning is defined as:

$$\mathcal{L}_{\mathcal{X}} = \frac{1}{|\mathcal{X}'|} \sum_{x,p \in \mathcal{X}'} H(p, \mathrm{p_{model}(y \mid x; \theta)}) \tag{3}$$

$$\mathcal{L}_{\mathcal{U}} = \frac{1}{L|\mathcal{U}'|} \sum_{u,q \in \mathcal{U}'} ||q, \mathrm{p_{model}(y \mid u; \theta)}||_2^2 \tag{4}$$

$$\mathcal{L}_{\mathcal{Z}} = \frac{1}{L|\mathcal{Z}'|} \sum_{z,r \in \mathcal{Z}'} ||r, \mathrm{p_{model}(y \mid z; \theta)}||_2^2 \tag{5}$$

$$\mathcal{L}_{encoder} = \mathcal{L}_{\mathcal{X}} + \lambda_{\mathcal{U}} \mathcal{L}_{\mathcal{U}} + \lambda_{\mathcal{Z}} \mathcal{L}_{\mathcal{Z}} \tag{6}$$

where H(p,q) indicates the cross-entropy between p and q, and T are hyperparameters described below. For each unlabeled example in $\mathcal{U}$ and generated sample in $\mathcal{Z}$, we first get the prediction for the example's label using the model. The prediction will be used in the unsupervised cross-entropy part. We compute the model's average predicted class distributions across all the K augmentations of u_b and z_b by

$$\bar{q}_{b_u} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{p_{model}(y \mid \hat{u}_{b,k}; \theta)} \tag{7}$$

$$\bar{q}_{b_z} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{p_{model}(y \mid \hat{z}_{b,k}; \theta)} \tag{8}$$

and then use Eq.(7) to reduce the entropy of the label distribution:

$$Sharpen(p, T)_i = p_i^{\frac{1}{T}} \; / \; \sum_{j=1}^{L} p_j^{\frac{1}{T}} \tag{9}$$

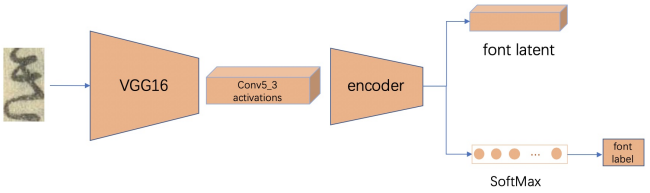

Fig. 4. Multi-Tasks Font Encoder for extract font information and classification

where p is some input categorical distribution, T is temperature hyperparameter to regulate classification entropy. If T is tending to 0, the output will approach "one-hot" distribution.

In this way, we can train all the dataset with generated images in order to increase the recognition accuracy of our font encoder.

B. Font Feature Extraction Network

Recently, CNN has been widely used to extract features of images. Here we introduce our multi-tasks font encoder which is based on VGG16 [17] network without pretrain. As shown in Fig.4. We choose this deep network to represent each character's fonts because we want to extract high-level latent features and retain more spatial information. The outputs of the encoder are font latent and predicted label. For early training, we use labeled dataset to improve recognition accuracy, which means to improve the ability to extract font information. The inputs to the font encoder are samples that randomly selected from the font reference set $\mathcal{X}_f$ and the input to the content encoder is a binary glyph image from the content reference set x_c. $\mathcal{X}_f = \{x_1, x_2, x_3, \dots\}$ which are selected from previous data mixing output with the same label.

During each training, we randomly choose N images from $\mathcal{X}_f$ as input, then output a font latent for the subsequent generation. We also set a confidence threshold such as 0.8 to judge whether the probability of classification of generated images is high enough. If the probability is higher than threshold, then the generated images will be added to the training dataset. The structure of content encoder is similar to font encoder but it is pretrained only to give a content latent.

C. Font Transfer Network

As shown in Fig.3. Our model is composed of a Generator G and two discriminators: Font Discriminator D_{font} and Content Discriminator $D_{content}$.

The generator G has several up-sampling layers. We apply a series of deconvolutional layers to achieve the output sample. The skip-connection method can make the model learn information from different scales. The features in the different layers present different information. In the low-level feature maps, the features contain more specific glyph information. Otherwise, the abstract font information is preserved in the high-level feature maps. We combine these information to make the whole network more robust.

For adversarial training, we set two discriminators in model. Due to the particularity of our historical documents dataset, there is often no paired training data, that is, given a content reference image, we don't have any corresponding ground truth image in the font dataset. Therefore, the transferred image need to be compared with font images and content image respectively. The adversarial loss can be described as:

$$\begin{aligned}\mathcal{L}_{font\_adv} = \mathbb{E}_{x_f}[log(D_{font}(x_f))] + \\ \mathbb{E}_y[log(1 - D_{font}(y))]\end{aligned} \tag{10}$$

$$\begin{aligned}\mathcal{L}_{content\_adv} = \mathbb{E}_{x_c}[log(D_{content}(x_c))] + \\ \mathbb{E}_y[log(1 - D_{content}(y))]\end{aligned} \tag{11}$$

where y is transfered image, x_f are font reference images, x_c is content reference image.

To stablize our training, we use an L_1 loss only to font reference images in our objective function because the goal is to constraint generated image to be as similar as reference images. Since we have multiple font reference images, the loss can be described as:

$$\mathcal{L}_{pixel} = \frac{1}{N}\sum_{n=1}^{N} ||x_{f_n} - y||_1 \tag{12}$$

where N indicates the number of font reference images.

Finally, The whole loss is defined as:

$$\begin{aligned}\mathcal{L} = \lambda_e\mathcal{L}_{encoder} + \lambda_f\mathcal{L}_{font\_adv} + \\ \lambda_c\mathcal{L}_{content\_adv} + \lambda_{pixel}\mathcal{L}_{pixel}\end{aligned} \tag{13}$$

IV. EXPERIMENT

A. Implementation details

The purpose of our work is to overcome the intra-class multimodality of the dataset from Japanese historical documents, generate new characters in order to increase the imbalanced dataset.

The dataset that we use is Kuzushiji Dataset [19], published by official National Institute in 2016. Training samples are from the Kuzushiji Dataset cut to single characters with different sizes. Until now (March 2020), the dataset contains 1,086,326 characters of 4,328 categories from 44 ancient Japanese documents. We did experiment on the hiragana part which occurs frequently in the historical documents with different fonts, totaling over one million images in 46 categories. The example of characters used in experiments are shown in Fig.5.

TABLE I
COMPARSION EXPERIMENT ON DIFFERENT DATA MIXING METHOD

Methods	Accuracy
No data-mixing method	0.9254
Pseudo label [18]	0.9332
MixMatch [8]	0.9547
Our data mixing	0.9586

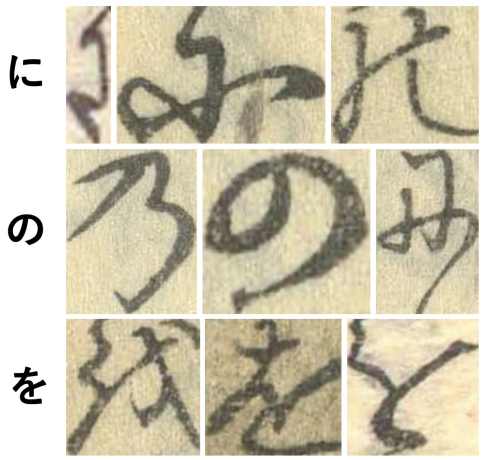

Fig. 5. Example of U+306B, U+306E and U+3092(from top to bottom) used in the experiments.

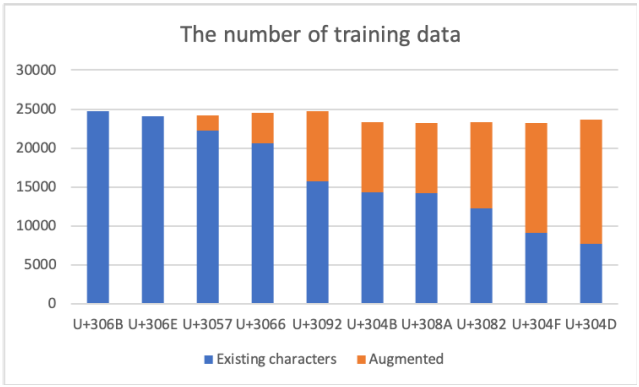

Fig. 6. Data augmentation for 10 Kuzushiji samples

For the semi-supervised part. The hyper-parameters are followed: sharpening temperature T=0.5, number of unlabeled augmentations K=2, β=0.75, $\lambda_{\mathcal{U}}$ and $\lambda_{\mathcal{Z}}$ start as 100 and goes linearly to 15000. As for the structure, we have five convolution layers and five transposed convolution layers in the generators, each layer has stance Normalization and ReLU. For loss, we set λ_{pixel}=10, λ_e, λ_f and λ_c equal to 1.

B. Semi-supervised method evalution

We first test our semi-supervised method to see how much recognition accuracy improvements can be achieved. We use VGG16 based network as our font encoder. We trained three classes of Kuzushiji dataset, which has only ten percent of the labeled dataset. As shown in Table I, we proved the performance of our data mixing method is better than [6].

C. Ablation experiment

We verify the performance of our method on the larger dataset. Since we use the Kuzushiji dataset, we pick out ten categories from original Kuzushiji dataset and compose them as training dataset. There are 165,046 images in total. The probability distribution of the original dataset and the augmented dataset is shown in Fig.6. We keep the ratio among these ten characters and augment the least eight categories to make their number almost equal.

Fig. 7. Example of reasonable outputs (a) and bad outputs (b)

TABLE II
ABLATION EXPERIMENT ON DATA AUGMENTATION

Methods	Accuracy
Original dataset	0.9352
Augmented dataset with traditional methods	0.9543
Cascade VAE [1]	0.9631
Our method	0.9669

Good results should produce meaningful and similar internal font to reference data. These samples should help train the classifier system. On the contrary, bad results mean that they can hardly be called ”characters”, such as Fig.7. The bad results mostly occurred in the beginning five epochs because the performance of the whole generation network is not yet trained well. With further training, the results are getting better and the training process tends to be stable.

To evaluate our generated samples with font can contain vivid and meaningful information, we augment the minimum of 8 categories of Kuzushiji dataset with three different methods. Then the ResNet18 [20] network is used to measure the recognition performance on the dataset. The results of the experiments are shown in Table II.

ResNet shows the powerful performance but due to various shortcomings of the historical documents. The performance can still not compare to neat and balanced handwriting datasets such as MNIST. The result also shows that embedding characters' font features to the generated result will provide recognition networks more information to learn.

D. Transfer experiment

Since the font in Kuzushiji dataset is unlabeled, there is no paired data to evaluate. We also test our model on a paired cursive dataset to see the transfer result. As shown in Fig.8. For the cursive font like this, the image still has a good visual effect.

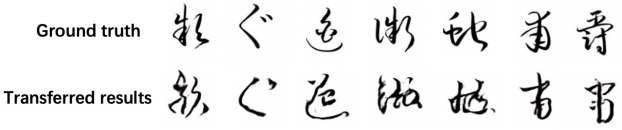

Fig. 8. Transfered font images of our method

V. CONCLUSION

Our goal is to extend the dataset of Japanese historical documents with font from a few stylized character images which can overcome the disturbance of intra-class multi-modality phenomenon. We propose a semi-supervised network structure with MixFont method for extracting character fonts and generating them end-to-end. Through the ablation study, it is proved that the consistency of image fonts and the enhancement of datasets are helpful to improve the accuracy of the recognition systems. In further study, we plan to introduce an unsupervised approach so that the system can automatically learn and migrate fonts from data without manual labels.

REFERENCES

[1] G. Cao and S.-I. Kamata, "Data augmentation for historical documents via cascade variational auto-encoder," pp. 340–345, 2019.

[2] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3722–3731.

[3] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[4] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017.

[5] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1–130, 2009.

[6] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[7] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in neural information processing systems*, 2017, pp. 1195–1204.

[8] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Advances in Neural Information Processing Systems*, 2019, pp. 5049–5059.

[9] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[10] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in neural information processing systems*, 2015, pp. 3483–3491.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[12] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[13] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[14] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "Cvae-gan: fine-grained image generation through asymmetric training," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2745–2754.

[15] S. Azadi, M. Fisher, V. G. Kim, Z. Wang, E. Shechtman, and T. Darrell, "Multi-content gan for few-shot font style transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7564–7573.

[16] Y. Zhang, Y. Zhang, and W. Cai, "Separating style and content for generalized style transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8447–8455.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2, 2013.

[19] Kuzushiji Dataset, National Institute of Japanese Literature, Center for Open Data in the Humanities. doi:10.20676/00000340. [Online]. Available: http://codh.rois.ac.jp/kmnist/index.html.en

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.