Data Augmentation with Automatically Generated Images for Character Classifier Model Training

Chee Siang Leow
University of Yamanashi
Kofu, Yamanashi, Japan
cheesiang_leow@alps-lab.org

Tomoki Kitagawa
University of Yamanashi
Kofu, Yamanashi, Japan
kitagawatomoki@alps-lab.org

Hideaki Yajima
University of Yamanashi
Kofu, Yamanashi, Japan
hideaki_yjm@alps-lab.org

Hiromitsu Nishizaki
University of Yamanashi
Kofu, Yamanashi, Japan
hnishi@yamanashi.ac.jp

***Abstract*—This paper presents a novel data augmentation technique crucial for training AI-OCR systems for handwritten character classification. Using a Y-autoencoder (Y-AE) enhanced with Adaptive Instance Normalization, diverse handwriting styles are generated to improve the breadth of handwriting representations. A filtering mechanism is introduced to include only valid character images for training. The method was tested on a subset of the ETL Character Database, featuring 92 unique Japanese Hiragana and Katakana characters. The baseline classifier achieved an accuracy of 0.9061. However, when using the augmented dataset, which included Y-AE model-generated and filtered images, the accuracy improved to a maximum 0.9555 with data augmentation technique. These results showcase the potential of this data augmentation technique in consumer electronics, particularly in AI-OCR software. Despite needing some noise removal, the approach significantly boosts classifier accuracy, suggesting an efficient way forward for document processing in various sectors.**

***Index Terms*—Data augmentation, Handwritten character recognition, Image generation, Y-autoencoder**

I. Introduction

The challenge of constructing adequate datasets for training deep learning models has been addressed in recent studies. A proposed solution is the pre-trained model, a method that enables model training even with a small training dataset. However, data augmentation methods specifically tailored for character classification tasks remain largely undeveloped. This paper aims to address this gap by proposing a data augmentation method that uses automatic character image generation to improve model accuracy for Japanese handwriting classification tasks. The model is trained on the ETL Character Database [1], which contains about 200 images per character, a number insufficient for robust model training. By applying Adaptive Instance Normalization (AdaIN) [2] to a Y-autoencoder (Y-AE) [3] model, we are able to generate handwritten character images with variations for the training recognition model.

Various data augmentation techniques applied in image classification models, such as Cutmix [5], mixup [6], and RandomErasing [7], have been proposed. Other techniques include ScrabbleGAN [9], which uses generative adversarial neural networks (GAN) [8] to generate variable length handwritten characters, and CycleGAN [10], a style transfer method for generating diverse Chinese characters. However, despite the plethora of data augmentation methods available for image classification and recognition tasks, few have been proposed for training character classifiers/recognizers. This paper expands on a previous study that used Y-AE for automatic character image generation for model training, extending the scope from Japanese Hiragana characters to include Katakana characters, thereby increasing the complexity of the classification task. The paper also provides a detailed analysis of the proposed method.

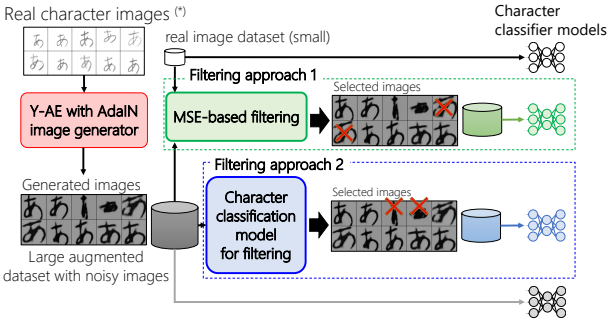

Fig. 1. The framework for Y-AE-based handwritten character image generation and character classifiers training with augmented images. (*)The real images are used to extract the handwriting style. From a single-character image, 92 different character images can be generated.

II. Generate Images Based on Y-Autoencoder with Filtering Methods

A. Model Architecture

The architecture of the Y-AE model used in this study, which was employed to generate Hiragana and Katakana character images, is shown in Fig. 2. This model is based on the original Y-AE architecture, comprising an encoder and a conditional decoder. The encoder utilizes a VGG16 [4] backbone feature extractor to encode the RGB image of dimensions $(128, 128, 3)$ and outputs the style representation s and the estimated character label e of the input image. The style representation s and the character label c serve as inputs to the decoder, which subsequently generates a handwritten character image. The character label c is transformed into a 512-dimensional embedding vector by the embedding layer, which is equipped with fully-concatenated (FC) layers. As shown in Fig.2, three FC layers is employed to convert the vector for injection into AdaIN, which is the content feature f_c, as depicted in equation 2. This allows for the creation of the intended character images.

$$f_c = FC(Emb(c)) \tag{1}$$

$$\text{AdaIN}(f_c, s) = \sigma(s)\left(\frac{f_c - \mu(f_c)}{\sigma(f_c)}\right) + \mu(s) \tag{2}$$

where the s is the style expression vector extracted by encoder. The model is trained same as the original Y-AE [3] with training conditions of number of epochs was 500, mini-batch

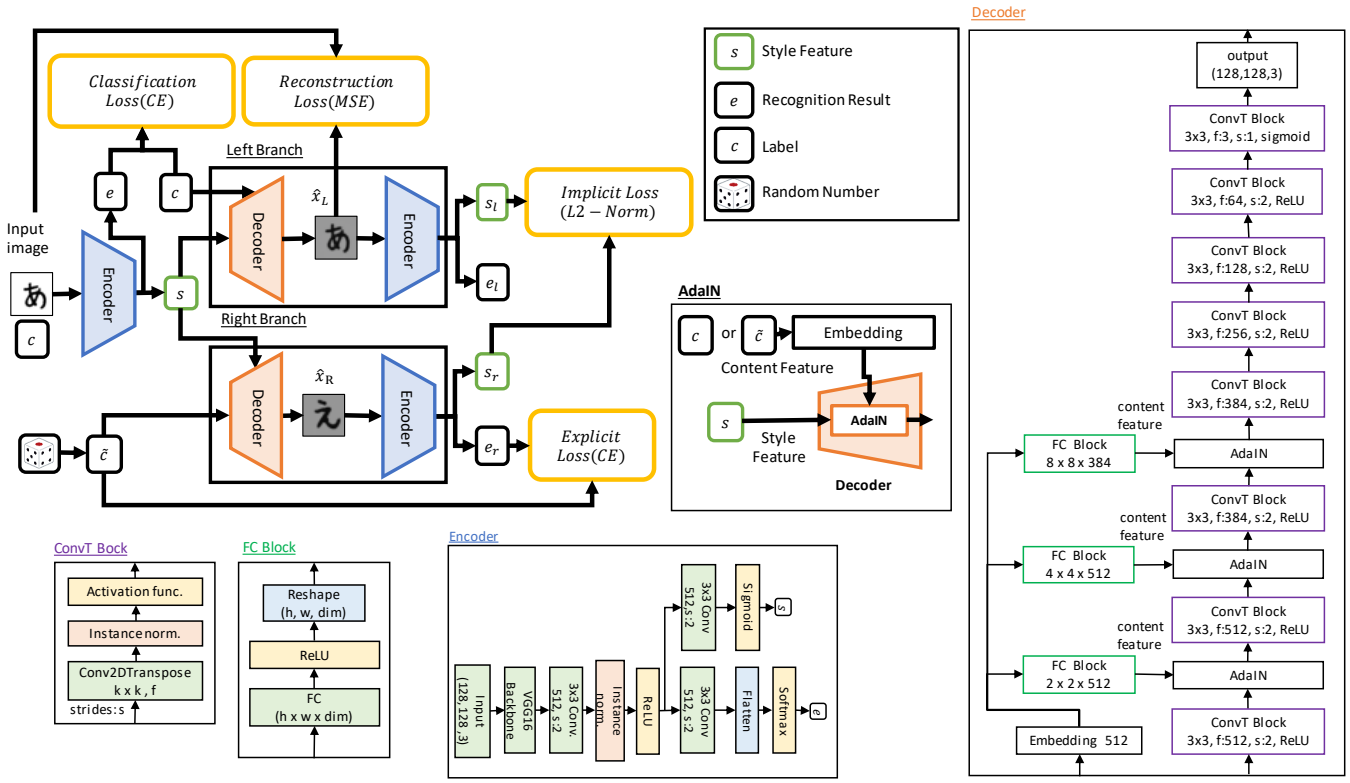

Fig. 2. The Y-autoencoder architecture.

size was eight, Adam was used as the optimization function, and the learning rate was set to 1e-4. During training, we employed three data augmentation of ElasticTransform, Affine, and GaussianBlur from a tool for image data augmentation called Albumentations [11]. Each of these three functions was applied with a probability of 50% during the generation of a mini-batch at the time of model training.

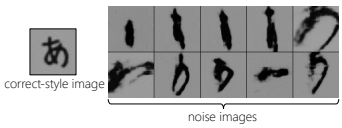

Fig. 3. Example of generated images of Hiragana character "あ."

B. Methods to filter generated images

The character images generated by the Y-AE models do not always represent the correct character form. For example, Fig. 3 shows some of the results of the generated character "あ." As shown in Fig. 3, the correct character is not generated in some images, and if these images are used to augment the data for training a character classifier, a highly accurate character classifier may not be able to be trained because of noise images.

Therefore, we introduce the a filtering method of the generated images. Two filtering methods are considered in this paper: one is MSE-based approach and the other is a character classifier-based approach. The MSE-based approach employs a generated image whose MSE scale to the real images is large. In the character classifier-based approach, a character classifier trained with the original character images (i.e., the baseline classifier) is used to recognize character images, and only correctly recognized character images are adopted for data augmentation.

1) MSE-based filtering: In the MSE-based filtering approach, the distance between two images is calculated using the following equation:

$$\mathrm{MSE}(A,B) = \frac{1}{w \times h} \sum_{x=0}^{w} \sum_{y=0}^{h} \{A(x,y) - B(x,y)\}^2 \qquad (3)$$

where w and h are the width and height of an image, respectively and $A(x,y)$ or $B(x,y)$ is the pixel value of the (x,y) coordinate in images A and B, respectively. The MSE value is 0 for images in which A and B are exactly the same, and this increases for images in which A and B are different. In other words, we consider the generated images with larger MSE values to be more suitable for data augmentation.

By calculating the MSE between the generated images and all the real images of the same character type, the generated images with the a high average MSE value are adopted as the image for data augmentation.

Note that when calculating the MSE, a pre-processing as shown in Fig. 4 is performed to eliminate factors due to the background of the generated images and the size of the characters. As shown in Fig. 4, the pre-processing was performed by the following steps:

1) A character image is converted to a binary image using Otsu's binarization [12].
2) Extraction of the character box in the image.
3) Cropping the character border area.

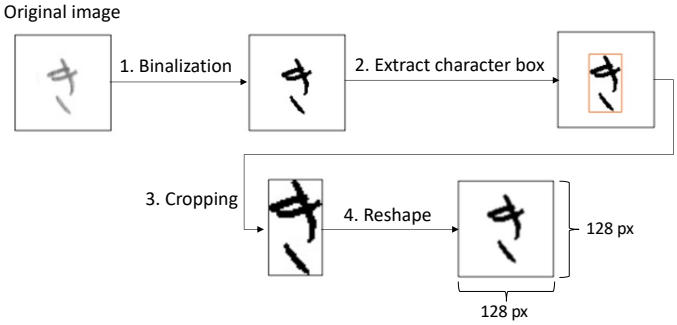

Fig. 4. Pre-processing of handwritten character images for calculating the MSE.

4) Reshape the image into a (128, 128) square so that margins of at least 10 pixels are added to the top, bottom, left, and right sides of the image.

C. Classifier-based filtering

A generated image is input to the character classifier, and if the classifier can correctly classify the image into the proper class, then the generated image is not noise and is considered to have retained the style of the characters. The character classifier used for filtering is trained on the same dataset used to train the Y-AE models. The architecture of the character classifier is described in Section II-D. A generated image is input to the character classifier, and if the classification result is correct with a posterior probability of 90% or higher at the time, the image is adopted as the image for data augmentation. Note that in training the character classifier using the generated images, only the top n images with the highest posterior probability are used for data augmentation in order to keep the number of images per character class the same. n is explained in Section III-B.

D. Handwritten Character Classifier

Because the present paper aims to confirm whether the automatically generated character images are effective for data augmentation, we use the simple ResNet-152 model [13]. The number of character types to be classified is 92 in total, consisting of 46 types of Hiragana and 46 types of Katakana characters of the Japanese language. The ResNet-152 model accepts 128×128 handwritten character images as the input, resulting in 92 output nodes. The model structure is exactly the same as the original ResNet-152 model except for the input and output layers. The ResNet-152 model is trained from scratch without pre-training.

III. Experiments

A. Y-AE model training

In the current study, we do not train a Y-AE model that can generate Hiragana and Katakana characters simultaneously, but we train two Y-EA models, one can generate a Hiragana character image while the other can also generate a Katakana character image. This is because there are some characters in Japanese with similar shapes in Hiragana and Katakana (e.g., "へ" and "ヘ"), and these characters may not be generated well if Hiragana and Katakana images are trained simultaneously using a single Y-AE model.

ETL has a total of nine subsets. We use 46 Hiragana characters from ETL-9 and 46 Katakana characters from ETL-5. For training the Y-AE model for generating Hiragana images, 200 handwritten character images are used for each Hiragana, for a total of 9,200 images. To train the Y-EA model for generating Katakana images, 208 handwritten characters are used for each Katakana, here for a total of 9,568 images.

When generating character images using the trained Y-AE models for Hiragana and Katakana characters generation, the same character images as used for model training are also used. In other words, the maximum number of character images generated is 423,200 (=9,200×46) for Hiragana and 440,128 (=9,568×46) for Katakana.

TABLE I
Number of character images used in training for each model.

Model no.	number of original ETL images	number of generated images
(1), (2)	18,768 (ETL-5, ETL-9)	—
(3), (4)	—	863,328
(5), (6)	18,768 (ETL-5, ETL-9	863,328
(7), (8)	18,768 (ETL-5, ETL-9)	24,196
(9), (10)	18,768 (ETL-5, ETL-9)	24,196

B. Character classifier training

First, in this paper, we trained character classification models for 92 Japanese Hiragana and Katakana characters using multiple datasets for comparison. In addition, exactly the same data augmentation functions as used to train the Y-AE models were applied during training. Figure 5 shows a list of the training conditions for the character classification models. In addition, six different datasets were used in this study. The number of images used in the training of each model is summarized in Table I. All character classifiers were subjected to the same training conditions except for the training data and the number of epochs. The mini-batch size was set to 8, Adam was used as the optimization function, and the learning rate was set to 1e-4. The ETL and generated images input to the model were binary images of a 128×128 image size, applying Otsu's binarization method to eliminate factors other than character shape.

The generated images used in the training of Models (7), (8), (9), and (10) were filtered using the MSE scale and the baseline character classifier. Note that in this case, the filtering was performed so that there would be 263 images per character class ($n = 263$). The reason for limiting the number to 263 is that the character class with the lowest number of images was 263 when the baseline character classifier was used for filtering. The number of images per character was exactly the same, and there was no difference in the number of images per class.

The validation and test sets for testing the classifier models consisted of real character images of Hiragana and Katakana characters included in ETL-1 and ETL-7. Each character was evaluated with 200 images, totaling 18,400 images. The classification accuracy is used as the evaluation measure.

C. Results and Discussions

1) Character generation results: Figure 6 shows the handwritten character images generated by the Y-AE generators with AdaIN for Japanese Hiragana and Katakana. As shown in Fig. 6, both Hiragana and Katakana handwritten characters were generated as if they were real. The original Y-AE model did not use AdaIN; the handwritten character images generated

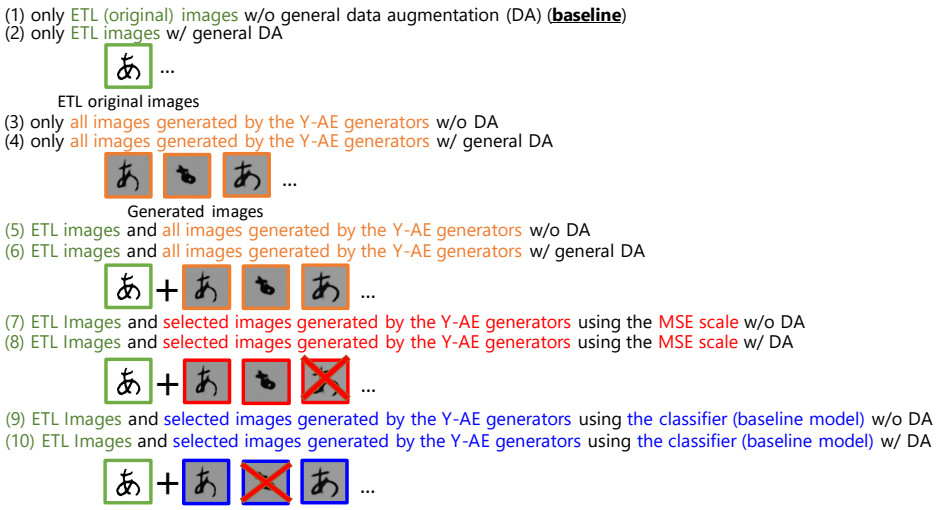

Fig. 5. List of training conditions for character classification models.

TABLE II
CHARACTER CLASSIFICATION ACCURACY (ACC.) FOR EACH MODEL. THE ARCHITECTURE OF THE CLASSIFICATION MODEL WAS THE SAME FOR ALL.

Model no.	DA	Valid. acc.	Test acc.
(1) ETL only (base)	✗	0.8832	0.9061
(2) ETL only	✓	0.9159	0.9302
(3) Generated images (GIs) only	✗	0.7979	0.8035
(4) GIs only	✓	0.8637	0.8620
(5) ETL + GIs (all)	✗	0.8910	0.8993
(6) ETL + GIs (all)	✓	0.9079	0.9127
(7) ETL + GIs (filtered MSE)	✗	0.9066	0.9217
(8) ETL + GIs (filtered MSE)	✓	0.9411	0.9474
(9) ETL + GIs (filtered baseline model)	✗	0.9176	0.9310
(10) ETL + GIs (filtered baseline model)	✓	**0.9428**	**0.9555**

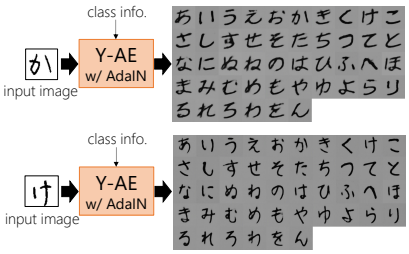

Fig. 6. Example of handwritten character images generated by the Y-AE with AdaIN.

by the Y-AE generator without AdaIN are shown in Fig. 7. As can be seen by comparing Fig. 6 and Fig. 7, the use of AdaIN clearly enabled the generation of a wide variety of handwritten characters.

Next, the MSE scale was also used to evaluate how much the generated handwritten character images differed from the real images used to train the Y-AE models. The MSE was the same as the calculation method used in the image filtering described in Section II-B1. Table III shows the statistics of the MSE scale. For images of the same character type, the smaller

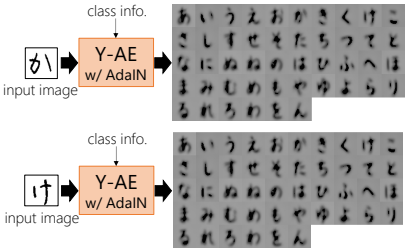

Fig. 7. Example of handwritten character images generated by the Y-AE without AdaIN.

the MSE value, the more the characters can be considered to be of the same handwriting style. Conversely, the larger the MSE value, the more likely it is that the characters had a completely different handwriting style. In Table III, the MSE values between images of the same character type (200 images for each character) were calculated on an all-possible combinations of all the images, and the mean, variance, and minimum MSE values are shown. The total number of real images is 18,400, including 92 types of Hiragana and Katakana characters in ETL-5 and ETL-9, 200 images for each character. The number of generated images is also 18,400, including 200 randomly selected images for each character type from the generated images by the Y-AE generators.

As shown in Table III, the MSE values between the real images in ETL had a larger mean and variance, indicating that there was more variation in the handwriting style. On the other hand, the statistics of MSE between the real and the generated images show that the values were smaller than those of MSE between the real images, and it can be considered that the variation of handwriting style was more limited than that of the real images. However, since the minimum value was 1.280, which is non-zero, no character was output exactly the same as the images used in the Y-AE model training. This shows that the Y-AE generator can be used to generate character images of handwriting style that are different from the training dataset.

2) Character classification results: Table II shows the character classification accuracy of each model for the test set. The baseline model (Model(1)) was trained from only

TABLE III
STATISTICS OF THE MSE SCALE BETWEEN CHARACTER IMAGES OF THE SAME CHARACTER TYPE.

Comparison target	Average	Variance	Minimum
Real images vs. real images	9.671	5.972	2.224
Real images vs. generated images	5.330	2.255	1.280

ETL-5 and ETL-9 real character images without any data augmentation, resulting in an accuracy of 0.8832 and 0.9061 on the validation and test sets, respectively. By applying three typical data augmentation functions on the same training set (Model (2)), the classification accuracies improved to 0.9159 and 0.9302. On the other hand, the model trained by adding images generated by our proposed Y-AE character generator as data augmentation images (Model(5)) showed an accuracy of 0.8993 on the test set, which was worse than the baseline. The model (Model (6)) trained by applying the data augmentation functions to this training set improved the accuracy to 0.9127, but was not as good as the model trained from the ETL alone. From this result, it is assumed that the character images generated by the Y-AE models contained many characters that were not well formed. In other words, there were a certain number of noise images that are not useful for training the character classification model. These noise images can be considered to be an obstacle to the training of the character classifier model.

In fact, the accuracy of the classification models trained using only images generated by the Y-AE model alone was 0.8035 (Model(3)) and 0.8620 (Model(4)) on the test set. Considering that the number of images was 46 times larger than the baseline but worse than the baseline, it can be concluded that there were a lot of noise images in the automatically generated images.

Therefore, when we trained the classification models using the ETL and the image data filtered from the generated images using the MSE scale and the baseline character classifier, the accuracies of the classifiers trained with the MSE scale-based and the baseline-based filtering methods improved to 0.9217 (Model(7)) and 0.9310 (Model(9)), respectively, on the test set. Furthermore, applying the same three data augmentation functions as in Model(2) further improved the accuracies to 0.9474 (Model(8)) and 0.9555 (Model(10)). The same results were obtained for the validation set.

These results indicate that the generated handwritten character images from the Y-AE generators trained on a limited data set can be sufficiently used as image data for data augmentation by eliminating noise can improve more by applying the image based data augmentation.

IV. CONCLUSIONS

In this paper, we proposed a data augmentation technique to train a character classifier with deep learning by automatically adding generated images to the training set using a Y-AE-based conditional generation model. Because the original Y-AE model [3] could not represent handwriting with rich variations in handwritten style, we made an improvement by applying AdaIN.

Our Y-AE model successfully generated handwritten character images with a wide variety of handwriting. On the other hand, the generated character image set contained noise (not suitable for training a character classifier); therefore, we applied character similarity using the MSE scale and character filtering using the character classifier trained with the real handwritten character images dataset only.

The effectiveness of the proposed method as a data augmentation was evaluated in terms of the accuracy of the character classifier. The experimental results showed that the character images generated by the Y-AE generator alone were not as good as the character classifier trained only with real handwritten character images; however, they were very useful as an extension to the dataset of real handwritten character images. In addition, it was also shown that existing data augmentation functions, such as Affine transformations, could also be applied to the generated character images. Finally, the character classification accuracy of the baseline model on the test set was 0.9061, while our proposed method achieved 0.9555, which was a significant improvement of 0.0494 points. This was a 47.4% improvement in the character error rate.

In the future, we plan to develop a more realistic character generation model using the diffusion-based model [14], [15] and a character generation model that can generate as many as 7,000 Japanese Kanji characters.

REFERENCES

[1] ETL Character Database, http://etlcdb.db.aist.go.jp/, Referred on 27/May/2023.

[2] X. Huang et al., "Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization," Proc of ICCV, 2017, pp.1510-1519.

[3] M. Patacchiola et al., "Y-Autoencoders: disentangling latent representations via sequential-encoding," Pattern Recognition Letters, vol.140, 2020, pp.59-65.

[4] K. Simonyan et al., "Very Deep Convolutional Networks for Large-Scale Image Recognition," Proc. of ICLR, 2015, pp.1-14.

[5] S. Yun et al., "CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features," Proc. of ICCV, 2019, pp.6022-6031.

[6] H. Zhang et al., "mixup: Beyond Empirical Risk Minimization," Proc. of ICLR, 2018, pp.1-13.

[7] Z. Zhong et al., "Random Erasing Data Augmentation," Proc. of AAAI, 2020, pp.13001-13008.

[8] I. Goodfellow et al., "Generative Adversarial Nets," Advances in Neural Information Processing Systems 27 (NIPS 2014), vol.27, 2014, pp.2672-2680.

[9] S. Fogel et al., "ScrabbleGAN: Semi-Supervised Varying Length Handwritten Text Generation," Proc. of CVPR, 2020, pp.4323-4332.

[10] B. Chang et al., "Generating Handwritten Chinese Characters Using CycleGAN," Proc. of WACV, 2018, pp.199-207.

[11] Buslaev, Alexander and Iglovikov, Vladimir I. and Khvedchenya, Eugene and Parinov, Alex and Druzhinin, Mikhail and Kalinin, Alexandr A., "Albumentations: Fast and Flexible Image Augmentations," https://doi.org/10.3390/info11020125, Referred on 31/July/2023.

[12] O.Nobuyuki., "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, 1979, vol.9, pp.62-66.

[13] K. He and X. Zhang and S. Ren and J. Sun., "Deep Residual Learning for Image Recognition," Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp.770-778.

[14] Ho, Jonathan and Jain, Ajay and Abbeel, Pieter., "Denoising Diffusion Probabilistic Models," Advances in Neural Information Processing Systems 33 (NIPS 2020), 2020, pp.6840-6851.

[15] Y. Song and J. Sohl-Dickstein and D. P. Kingma and A. Kumar and S. Ermon and B. Poole., "Score-Based Generative Modeling through Stochastic Differential Equations," Proceedings of the International Conference on Learning Representations (ICLR 2021), 2021, pp.1-36.