Automatic Characteristic Line Drawing Generation using Pix2pix

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

Keywords: Neural Network, Image Synthesis, Line Drawing Generation, Automatic Coloring, Pix2pix.

Abstract: A technology known as pix2pix has made it possible to automatically color line drawings. However, its accuracy is based on the quality of the characteristic lines, which emphasize the characteristics of the subject drawn in the line drawing. In this study, we propose a method for automatically generating characteristic lines in line drawings. The proposed method uses pix2pix to learn the relationship between the contour line drawing and line drawing with characteristic lines. The obtained model can automatically generate a line drawing with the characteristic lines from the contour line drawing. In addition, the quality of the characteristic lines could be adjusted by adding various degrees of blurring to the training images. In our experiments, we qualitatively evaluated the line drawings of shoes generated using the proposed method. We also applied an existing automatic coloring method using pix2pix to line drawings generated using the proposed method and confirmed that the desired colored line drawing could be obtained.

1 INTRODUCTION

Pix2pix is a method for acquiring a generator that performs a desired image conversion by learning paired images before and after applying the conversion (Phillip, I. et al. 2017). It uses generative adversarial networks (GANs) (Ian, J.G. et al. 2014). The automatic coloring of drawings is one of the image conversions that can be realized using pix2pix. With this technology, realistic illustrations can now be created simply by drawing line drawings.

However, there are two problems with this automatic coloring realized by pix2pix. The first is that the quality of the coloration of the image obtained by automatic coloring depends on the quality of the line drawing to be input. When only the basic characteristics of a subject are depicted in the line drawing, the coloring result tends to be simple. When the line drawing captures various characteristics of the subject, realistic coloring results can be generated. We refer to these lines as the characteristic lines. The other is that the line drawing to be input must be manually prepared. Therefore, to obtain a sophisticated illustration image, it is necessary to manually prepare a line drawing that captures various characteristics. However, for beginners learning to create illustrations, creating such line drawings by themselves is a complicated task. At present, there are few conventional techniques that can support the creation of line drawings that capture the characteristics of the subjects.

From this perspective, we propose a method to support the creation of a line drawing that captures the characteristics of the subject. The main advantage of the proposed method is that the quality of the characteristic lines can be adjusted when line drawing is automatically generated. This is actualized by applying various levels of blur to the training images to be trained by pix2pix. By changing the level of blur, our method can control the amount and precision of the generated lines and change the quality of the generated images.

The remainder of this paper is organized as follows. Section 2 describes related work, including pix2pix. Section 3 explains the proposed method, which consists of a set of pix2pix, creates line drawings, and colors them automatically. Section 4 introduces the experimental results of the proposed method and evaluates the results of the generated line drawings and their colored images.

2 RELATED WORKS

2.1 Pix2pix

Our method uses pix2pix, which provides an easy implementation of the desired image transformation.

Yanagida, K., Gyohten, K., Ohki, H. and Takami, T.
Automatic Characteristic Line Drawing Generation using Pix2pix.
DOI: 10.5220/0010776700003122
In *Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2022)*, pages 155-162
ISBN: 978-989-758-549-4; ISSN: 2184-4313

Pix2pix is based on GAN, which is a type of algorithm used for unsupervised learning and can generate pseudo-images that resemble the training images. The basic structure and learning process of pix2pix are almost the same as those of the normal GAN. However, there were two differences between them, as shown in Figure 1.

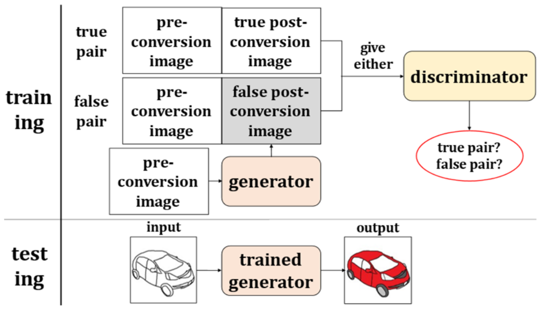

Figure 1: Training and testing of pix2pix.

First, the input to the generator for learning is not a random noise vector but a real pre-conversion image. When a random noise vector is input to the generator, the GAN cannot control the types of image output from the generator according to the input. On the contrary, the generator in pix2pix can generate a highly accurate pseudo-image by performing appropriate image conversion according to the input image. In addition, while GAN has to generate an image from a simple random noise vector, pix2pix only has to convert an input image into the desired image. Because the process of image conversion in pix2pix is simpler than that of GAN, its learning time can be shortened.

Second, the image provided to the discriminator is not a single image but a pair of images. The discriminator in pix2pix only requires to solve a conversion problem by deriving the correspondence between the images. This makes it relatively easy to capture the features in them. By clarifying what the discriminator requires to learn, pix2pix can produce more complete fake images than GAN.

Many image transformations can now be easily realized using pix2pix. As shown in Phillip, I. et al. (2017), pix2pix can be applied to a wide range of image transformations, such as converting a black-and-white image into a color image, converting a daytime sky pattern into a nighttime sky pattern, and converting a label image into a real image. As mentioned in Section 1, we focused on the automatic coloring of line drawings.

The automatic coloring of a line drawing using pix2pix has the advantage that it can automatically color parts where the lines are missing in the given line drawing. Figure 2 shows an example of this advantage. This figure shows that the hair and nose can be complemented and colored automatically, even if the input line drawing image does not include them. However, whether the missing line information is complemented depends on the training images. The boundaries of the parts obtained by completion are often ambiguous. Clearly, it is preferable to draw as many lines as possible in the input line drawing.

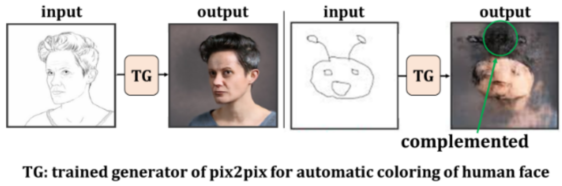

Figure 2: Line completion, the advantage of automatic coloring using pix2pix.

2.2 Control of Generated Images in GAN

To obtain the generated color illustrations to the quality that the user wants, it is necessary to control the image output by pix2pix. The following methods were proposed to control the images obtained by the GAN.

In cGANs (Mirza, M. and Osindero, S. 2014), the generator controls the generation process by learning supplementary information regarding the data as a conditional probability distribution. DCGAN (Radford, A. et al. 2016) attempted to solve the problem in which a single pair of generators and classifiers fluctuated and did not converge like a discriminatively trained network. This method controlled the generation of larger images by training with multiple generators and classifiers. VAE-GAN (Larsen, A.B.L. et al. 2016) learned features from latent or image space to address the GAN mode collapse and encoder-decoder architectures. EBGAN (Zhao, J. et al. 2016) used an autoencoder to control captured images to address the issue of mode collapse owing to insufficient capacity or poor architecture selection. MemoryGAN (Kim, Y. et al. 2018) incorporated a storage module to handle this problem in which the structural discontinuity of classes was not clear and made the generated images unstable because the discriminator forgot the previously generated sample. DeLiGAN (Gurumurthy, S. et al. 2017) generated a variety of images by re-parameterizing the latent space.

The above existing methods modify the GAN structure to control the acquired images. On the contrary, this study is characterized by applying various image processing techniques, such as blurring the training images and controlling the quality of the generated images by adjusting the degree of blurring.

3 PROPOSED METHOD

In this section, we propose a method to support the creation of line drawings with characteristic lines, featuring the ability to adjust the level of detail in the drawing. In this method, a contour image was input as a part of the line drawing to be drawn, which served as a clue for line drawing generation. Figure 3 shows an overview of the proposed method. The proposed method consisted of two pix2pix, PBB and PBC. First, we input a line drawing with only contour lines and applied a pix2pix, which output the outlines of the characteristic lines (PBB). Next, the other pix2pix was applied to convert the result from PBB to a final line drawing with the characteristic lines (PBC). In addition, the line drawing obtained from the PBC was colored using a conventional pix2pix. The dataset and the architecture used in the coloring pix2pix were based on the model proposed by Phillip Isola et al. (2017). In the following, we show how PBB and PBC can be combined with the original pix2pix to automatically generate a colored image from a line drawing where only the outlines are drawn.

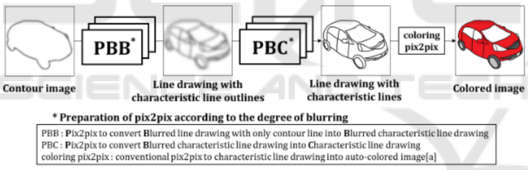

Figure 3: Overview of the proposed method.

3.1 Automatic Generation of Line Drawing with Characteristic Lines

To implement pix2pix, training images should be prepared and trained. In PBB, we applied three processes to the training images: contour-only line drawing generation, projection transformation, and bounding rectangle extraction. This is explained in Sections 3.1.1 to 3.1.3.

3.1.1 Obtaining Contours

It is necessary to prepare line drawings with only the contours as the pre-conversion image and line drawings with characteristic lines drawn as the post-conversion image. Figure 4 shows examples of prepared pre-conversion and post-conversion images. We applied common edge extraction methods to the original color images and obtained line-drawing images with characteristic lines. Then, we obtained line drawings with only contours by extracting the contours of the line drawings with characteristic lines.

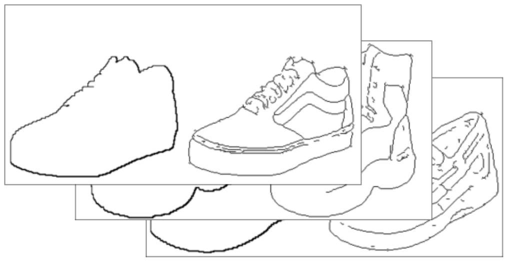

Figure 4: Examples of the training dataset for PBB.

3.1.2 Projection Transformation

If the subjects of the prepared training images are all taken from almost the same direction, the proposed method may not be able to achieve reliable training for the generation of images from various perspectives. In addition, it may not be possible to prepare a sufficient number of training images that can flexibly generate the desired characteristic lines. Therefore, we propose a method to augment the training images by applying a random projection transformation to the training images to generate line drawings as if they were drawn from various perspectives and to prepare sufficient number of images. The projection transformation used in our method was used in a two-dimensional projection space and generated twisted images by randomly changing the four corners of the given images.

The transformation process is described in detail below. The process is illustrated in Figure 5. First, the input image was reduced to half its original size, and the areas within the green frames shown in Figure 5 were reserved. Within these areas, the positions of the four corners of the transformed image were randomly determined and transformed by projection transformation. For example, the top-right point of the reduced image in Figure 5 was moved in the upper-right direction, as indicated by the red arrow. Using the above process, our method can create a line drawing with a different appearance from the original. The results of applying various projection transformations are shown in Figure 6.

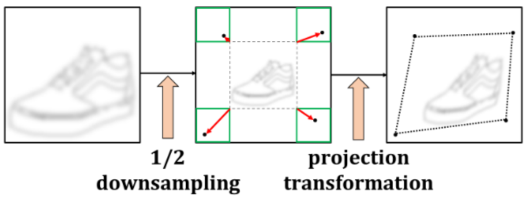

Figure 5: Projection transformation.

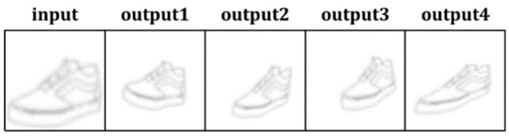

Figure 6: Image after applying the projection transformation.

3.1.3 Extraction of Bounding Rectangles

After applying the projection transformation described in Section 3.1.2, the size of the subject in the line drawing becomes smaller than that of the subject in the original line drawing. When we trained the pix2pix on these line drawings as the training images, pix2pix could not output appropriate line drawings. From this result, it was inferred that the size of the subject drawn in the image provided to pix2pix must be normalized. Therefore, as illustrated in Figure 7, we obtained the bounding rectangle of the subject in the line drawing reduced by the projection transformation and normalized its size.

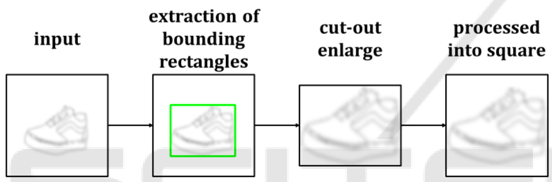

Figure 7: Image normalization using minimum bounding rectangle.

3.2 Control of Generated Line Drawings

We conducted a basic experiment in which we trained pix2pix on binary images of contour-only line drawings and binary images of line drawings with characteristic lines, which were prepared using the method described in Section 3.1. The result was that pix2pix could not output the line drawing with characteristic lines. This was probably because the line drawing almost consisted of white pixels and did not have spatial gradients in the pixel value distribution, so the learning algorithm based on gradient descent did not work effectively. To effectively learn pix2pix, it is necessary to add information that indicates the existence of black pixels in the neighbor of black pixels in the training image to make it easier to find the relationship between lines.

To solve this problem, our method applied Gaussian blur, a type of blurring process, to both pre- and post-transformed images for training in PBB (Chung, M.K. "3. *The Gaussian kernel*"). This approach could make the pix2pix learn the relationship between lines and convert an image with only contour lines to a line drawing with the outline of the characteristic lines.

Furthermore, by varying the degree of blurring, the level of detail of the generated line drawing and the colored image can be controlled. The blurring process replaces a pixel value with the average of neighboring pixel values. By adjusting the kernel size, which is the size of the neighbor for calculating the average value, we can change the degree of blurring and control the level of detail of the generated line drawing and colored image. As shown in Figure 3, the proposed method provided PBBs and PBCs for each kernel size.

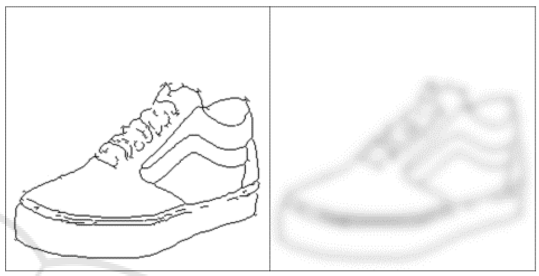

Figure 8: Blurring using Gaussian blur (blur degree: 16).

3.3 Converting Blurred Images to Clear Line Drawings

The method described in Section 3.2 can generate line drawings that include the outline of characteristic lines. However, because the acquired line drawing is blurred, it must be converted to a clear line drawing. Therefore, we introduced pix2pix (PBC), which converted the blurred image into a line drawing.

We prepared training images in which the output images were line drawings obtained by the method described in Section 3.1, and the input images were the blurred line drawings obtained by applying the blurring process described in Section 3.2. By training pix2pix on these training images, we obtained a model that could convert blurred images to clear line drawings. Figure 9 shows the results of restoring a line drawing from a blurred image.

Figure 9: Results of pix2pix that converts blurred images to line drawings.

4 EXPERIMENTS

In this section, we explain the outline of the experiments to verify the proposed method and describe the experimental results and discussion. In the experiments, we used a dataset of shoe images from the training dataset[1], which has been published in studies on automatic coloring using pix2pix (Phillip, I. et al. 2017). First, we experimented with PBB to generate line drawings containing characteristic line outlines from contour-only line drawings, and with PBC to convert line outlines into line drawings. We applied the automatic line drawing coloring proposed in Phillip, I. et al. (2017) to the generated line drawings and verified the quality of the final images. We also investigated how the quality of the acquired images changed by adjusting the kernel size in the PBB. Four types of shoes were used in these experiments: leather shoes, heels, sandals, and sneakers.

The experiments were conducted in the same computing environment. Our method was implemented on a Windows 10 operating system, a Core i9-9900k CPU, and a GeForce RTX 2080 Ti GPU, using Python as the programming language and TensorFlow as the deep learning framework. Figure 11 shows the results of generating colored images from contour-only line drawings from those experiments. The kernel size for the blurring process described in Section 3.2 can be varied from 1 to 16. Figure 10 shows some of the results obtained using different kernel sizes.

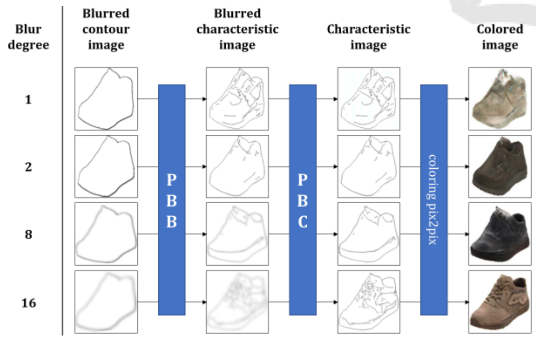

Figure 10: Coloring results of proposed method using different kernel sizes.

4.1 Line Drawing Generation

First, we generated line drawings using the PBB and PBC. The four processes described in Section 3, that is, contour-only line drawing generation, blurring, projection transformation, and size normalization by extracting bounding rectangles, were applied to each of the training and test images in the dataset. The actual dataset used is shown in Figure 11. The left image in the training image is the pre-transformed image from Figure 1, and the right image is the real post-transformed image from Figure 1. The left image of the test image is the input image for testing as shown in Figure 1, and the right image is the correct image of the input image.

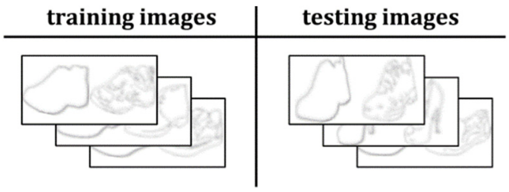

Figure 11: Datasets for PBB.

A total of 1,000 shoe images generated by applying 10 random projection transformations to 100 shoe images were used as training images. The number of images for each type of shoe was 10 heels, 40 leather shoes, 40 sneakers, and 10 sandals.

To determine the optimal number of epochs, we trained PBB and PBC using 1,000 training images for 25, 50, 100, 1,000, and 10,000 epochs. Line drawings were generated using the models obtained from the training and verified their accuracy. The images produced by the models trained for 25 and 50 epochs often had unnatural lines. The images produced by the models trained for 100, 1,000, and 10,000 epochs produced appropriate line drawings. Because there was no difference in the quality of the line drawings according to the number of epochs, we set the number of epochs to 100.

The experimental results are presented in Figure 12. The resulting line drawing is shown as the output image, compared to the input image of the test image, which is a contour-only line drawing. The original line drawing, which was the basis of the input image, is shown below the output image as the correct answer image. A total of 5,000 test images were obtained by applying 10 projection transformations to 500 different shoe images.

It is difficult to quantitatively evaluate the accuracy of the generated line drawings. Therefore, we qualitatively evaluated whether the results were "good looking," "not good looking," or "poor looking," and discussed the results. For heels and

[1] https://people.eecs.berkeley.edu/~tinghuiz/projects/pix2pix /datasets/edges2shoes.tar.gz

leather shoes, the output results were good-looking because the shapes were not complicated and the number of characteristic lines were relatively small. On the contrary, sneakers and sandals showed output results with complicated shapes and poor appearance. Particularly for sneakers, the appearance of many of the output images was poor because of the complexity of the characteristic lines.

Next, we describe the differences in the generated line drawings by adjusting the kernel size in the blurring process. When the kernel size was set to one, several characteristic lines were drawn using PBB, but they contained some noise. When the kernel size was increased from two to eight, the output results from the PBC did not contain noise, but the number of characteristic lines obtained from PBB was very small. When the kernel size was increased to eight or more, the number of characteristic lines drawn gradually increased as the kernel size increased. When the kernel size was increased to the maximum value of 16, a very large number of characteristic lines were drawn, as shown in Figure 10.

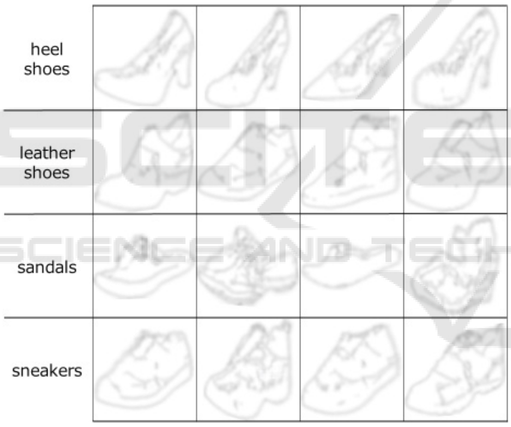

Figure 12: Experimental results of line drawing generation.

4.2 Coloring

For all line drawings generated in Section 4.1, we applied the automatic coloring of line drawings proposed in Phillip, I. et al. (2017). The training image for the coloring experiment is the training image from the dataset to which the projection transformation is described in Section 3.1.2, and the size normalization based on the bounding rectangle is described in Section 3.1.3. Figure 13 shows the training and test images. As shown in Figure 11, the left image of the training image is the image before the transformation, and the right image is the image after the transformation. In the test image, the image generated by the proposed method is used as the input image. Therefore, unlike in Figure 11, as in the experiment in Section 4.1, we used images of heels, leather shoes, sneakers, and sandals as shoe types.

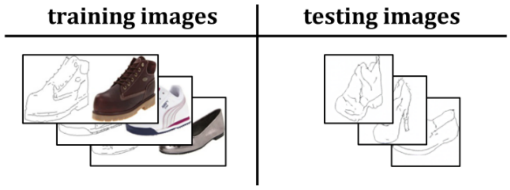

Figure 13: Dataset for coloring pix2pix.

Figure 14 shows some of the coloring results. Because the output results of Section 4.1.1 were used as the input images, the total number of images in the coloring result was 5,000.

The heels and leather shoes, which were evaluated to have a good appearance in the line drawing generation experiment, were colored without any problems. For sandals, which were evaluated to have a poor appearance in some cases owing to their complex shape, the unnatural parts were well complemented during the coloring process, and a good appearance was obtained. For sneakers, some processing results were successfully colored, as shown in Figure 14. However, there were some coloring results that did not look good owing to the extreme complexity of the characteristic lines of the sneakers.

Next, we discuss the differences in the results of coloring for each generated line drawing obtained by adjusting the kernel size. As described in Section 4.1, the line drawing generated using the blurring process with a kernel size of one contained noise. Therefore, it cannot color line drawings naturally. The line drawing generated by the blur process with a kernel size of two contained very few characteristic lines. Therefore, the resulting colorized images were plain and unnatural. As shown in Figure 10, although the coloring result was generated from the contour lines of a sneaker, the number of characteristic lines generated from PBB and PBC was very small, therefore the coloration was similar to that of leather shoes.

This problem was resolved when the kernel size was set to eight or more. Although the number of characteristic lines obtained was small, more naturally colored images were obtained. As the kernel size increased, the number of characteristic lines obtained increased, and the accuracy of conversion to natural-colored images could be maintained. When the kernel size was set to the maximum value of 16, a large number of characteristic lines were obtained, as shown in Figure 10. As a result, we were able to reproduce natural coloring images with high accuracy, even for shoelaces, which are difficult to convert because of their complex structure.

In addition, as mentioned in Section 2.1, we confirmed that the areas where characteristic lines were not drawn were complemented and automatically colored. For sneakers and sandals, which were evaluated as having poor appearance in the line drawing results, their missing characteristic lines were complemented through the coloring process, resulting in colored images with good appearance. This result shows that the final appearance should be judged not by the output result shown in Figure 12, but by the automatically colored image shown in Figure 12.

Figure 14: Experimental results of coloring.

5 CONCLUSION

In this study, we proposed a method for automatically generating feature-captured line drawings using simple operations. The proposed method generated an outline of characteristic lines from a contour-only line drawing using a model obtained by training pix2pix on a training image to which four processes were applied: acquisition of a contour-only line drawing, blurring, projection transformation, and image size normalization based on the bounding rectangle. Then, our method applied pix2pix to generate a final line drawing from the outline of the characteristic lines and produced a line drawing with characteristic lines. Colored illustrations can be generated for line drawing by applying pix2pix, which has already been proposed for color line drawings. In addition, the level of detail of the lines and those of the coloring can be adjusted by changing the degree of blurring in the blurring process.

In the experiments, we evaluated line drawings with characteristic lines generated from contour-only line drawings and their colored images generated from the line drawings. In addition, we examined how the acquired images were changed by adjusting the degree of blurring. As a result, we observed that if the degree of blur was weak, noise would be mixed in with the line drawing, making it look bad. However, when the degree of blurring was increased by increasing the kernel size, the number of lines that captured the features was reduced, and noiseless line drawings were obtained. By making increasing the degree of blurring, the number of lines that captured the features in the generated line drawing increased.

In this study, contour lines were input as part of the subject as a starting point for line drawing generation. In the future, it will be necessary to survey designers and others to determine what type of line drawing is appropriate for use as a starting point for line completion. Because the subject of the experiment was only shoe images, which is not practical, we would like to verify it with various practical images. In addition, it was necessary to quantitatively evaluate the obtained results.

ACKNOWLEDGEMENTS

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

REFERENCES

Phillip, I., Jun-Yan, Z., Tinghui, Z., Alexei, A. E. (2017). Image-to-Image translation with conditional adversarial networks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1125–1134.

Ian, J.G., Jean, P-A., Mehdi, M., Bing, X., David, W-F., Sherjil, O., Aaron, C., Yoshua, B. (2014). Generative adversarial networks, *Advances in Information Processing Systems 27(NIPS)*.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets, arXiv Preprint arXiv1411.1784.

Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks, *In 4th International Conference on Learning Representations (ICLR'16)*.

Larsen, A.B.L., Sønderby, S.K., Larochelle, H., and Winther, O. (2016). Autoencoding beyond pixels using a learned similarity metric, *In 33rd International Conference on Machine Learning (ICML'16)*, pp. 2341–2349.

Zhao, J., Mathieu, M., and LeCun, Y. (2016). Energy-based generative adversarial networks, arXiv preprint arXiv:1609.03126.

Kim, Y., Kim, M., and Kim, G. (2018). Memorization precedes generation: Learning unsupervised GANs with memory networks, arXiv preprint arXiv:1803.01500.

Gurumurthy, S., Sarvadevabhatla, R.K., and V. Babu, R. (2017). DeLiGAN: Generative adversarial networks for diverse and limited data, *In 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, pp. 4941–4949.

Chung, M.K. University of Wisconsin-Madison – STAT 692 Medical Image Analysis – 3. *The Gaussian kernel*.