International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

A Study on Image Quality Improvement for 3D Pagoda Restoration

Byong-Kwon Lee

Submitted: 09/11/2022 **Accepted:** 11/02/2023

Abstract: The SRGAN algorithm based on the Generative Adversarial Networks (GANS) algorithm is used for image generation, image restoration, and resolution improvement. In this study, we proposed a method to improve the quality of pagoda images to create a 3D model by combining several 2D pagoda images. In the study, the SRGAN artificial intelligence algorithm was used to minimize the noise generated when converting low-quality images into 3D models. Low resolution, high resolution and super resolution results were obtained with the SRGAN algorithm by selecting the low quality of the Pagoda image dataset. In addition, the degree of resolution improvement was confirmed by collecting quantified R, G, and B information through histogram analysis. The research results will be used as a dataset for converting 2D images into 3D models.

Keywords*: SRGAN algorithm, Pagoda image resolution, image histogram, super resolution algorithm*

1. Introduction

There are many ways to restore the pagoda, a traditional cultural heritage of Korea. Algorithms used to restore the 3D form of pagoda include point cloud method, graph method, and mesh method. Figure 1 schematically shows the classification system for the 3D data expression method (Johnson, Justin & Alahi, Alexandre & Fei-Fei, Li. (2016). In this study, a method of improving the image quality of images taken in all directions (east, west, north, south) as a pre-processing process for pagoda restoration was studied. In this study, the dataset for pagoda restoration is made by improving the image quality of the damaged and indistinguishable top image and the low-resolution image. In addition, the histogram information was checked and evaluated through analysis of the improved image quality of the pagoda image, and it was used for 3D modeling of the pagoda (J. Li, L. Wu, S. Wang, W. Wu, F. Song and G. Zheng (2019). In Section 2, research on related technologies, in Section 3, the RSGAN algorithm introduction, results, and improvement directions are derived, and in Section 4, problems are analyzed and future development directions are presented.

School of Media Contents, Dept. of Multimedia Major, Seowon University
sonic747@daum.net

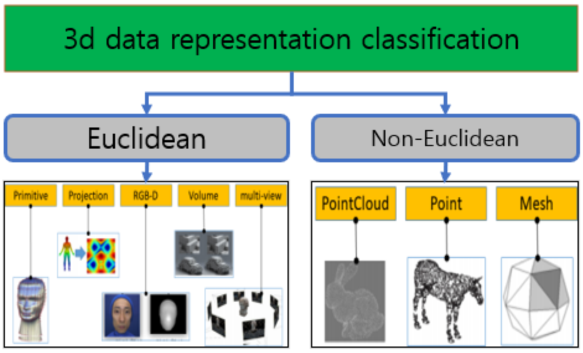


Fig 1. The Classification System for the 3D data expression method

2. Preliminaries

2.1. GAN (Generative Adversarial Networks)

Generative Adversarial Networks (GANs) are algorithms that create, combine, and transform datasets that lack deep learning skills using artificial intelligence (S. H. Salem Hussin and R. Yildirim (2021)). In addition, GAN technology predicts the original image with a small amount of data and transforms the model required for training in a specific way. GAN is widely used in image processing and learns one multi-layered artificial neural network for training data. However, GAN is a single generative neural network that performs interactive learning of two artificial neural networks and finally creates a fake model that is difficult to visually distinguish from the real one. GAN uses Generator and Discriminator to train models against each other (J. Osorio Ríos, A. Armejach, G. Khattak, E. Petit, S. Vallecorsa and M. Casas (2020)). Since the first appearance of the GAN algorithm, various studies have been conducted (Y. Lecun, L. Bottou, Y. Bengio and P. Haffner (1998), Salamani et al.,(2018)). Fig.1 shows the development process of GAN algorithm. This study conducted a study on SRGAN in Figure 2.

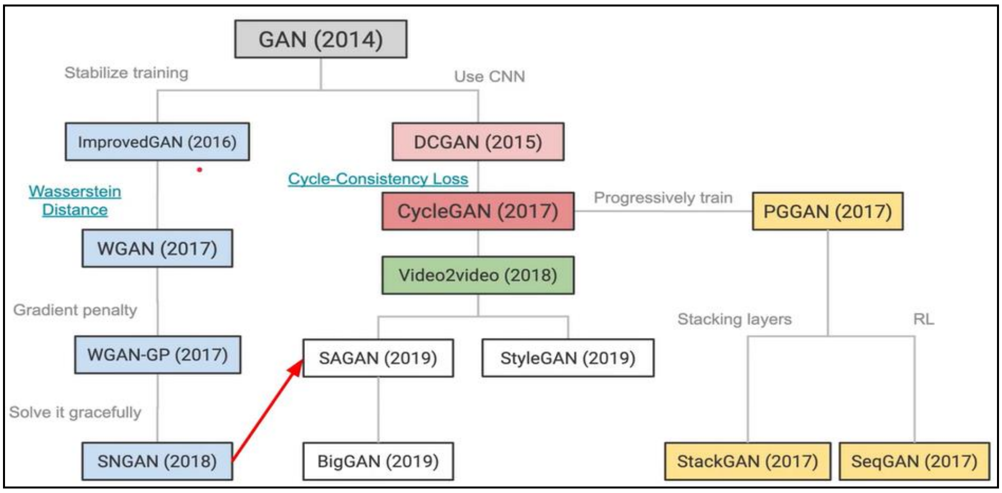


Fig. 2. A development process of GAN algorithm

2.2. SRGAN (Super Resolution GAN)

SRGAN (Super Resolution GAN) is an algorithm that converts a low-resolution image into a high-resolution image. The SRGAN AI algorithm makes SR (super resolution) models that are difficult to identify due to the small size of the subject of old, damaged images and videos, images used in the past, and captured images. Equation 1 is a loss function and consists of a content loss and an adversarial loss (X. Jiang, Y. Xu, P. Wei and Z. Zhou (2020), N. Xiang, B. Tang and L. Wang (2022)).

$$l^{SR} = \underbrace{\underbrace{l^{SR}_{X}}_{\text{content loss}} + \underbrace{10^{-3} l^{SR}_{Gen}}_{\text{adversarial loss}}}_{\text{perceptual loss (for VGG based content losses)}}$$

……………….. (1)

The loss function uses a perceptual loss and consists of a content loss and an adversarial loss. The adversarial loss is similar to the GAN loss we know in general, and a slightly special part is the content loss. (Formula 1) is the Loss function - Perceptual loss.

$$l^{SR}_{Gen} = \sum_{n=1}^{N} - \log D_{\theta_D} \left(G_{\theta_G} \left(I^{LR} \right) \right)$$

………(2)

In (Formula 2), logD of Adversarial loss formula is the probability of judging the image generated by the generator as genuine, and it learns in the direction of minimizing it by adding a (-) in front. The existing GAN loss has the disadvantage of slowing down at the beginning of training if it is in the form of log (1-x). Changing this to -log (x) will speed up the learning.

$$l^{SR}_{VGG/i.j} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \left(\phi_{i,j} \left(\left(I^{HR} \right)_{x,y} \right) - \phi_{i,j} (G_{\theta_G} (I^{LR}))_{x,y} \right)^2$$

……(3)

In (Formula 3), Content loss is almost identical to the perceptual loss first presented in the paper called Perceptual Losses for Real-Time Style Transfer and Super-Resolution.[3]

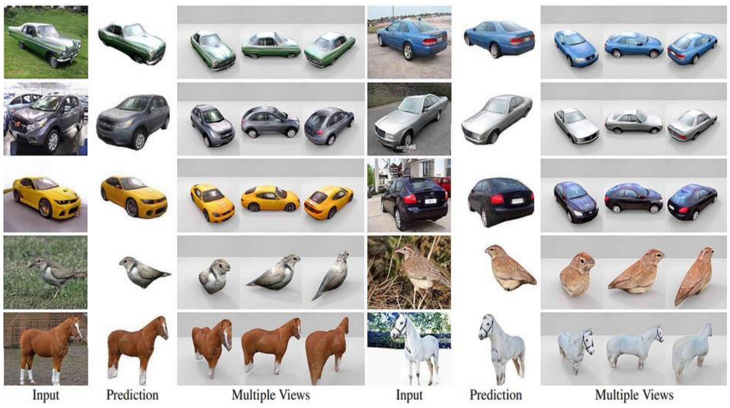


Fig. 3. High resolution using SRGAN

2.3. Important Information

2.3.1 Establishment of Experimental Environment

Artificial intelligence learning proceeds with learning based on a large amount of dataset. However, the performance of AI learning depends on the resolution of the collected data. In this study, an experiment was conducted to restore a low-resolution Pagoda image to a high-resolution using the SRGAN artificial intelligence algorithm (X. Hou, T. Liu, S. Wang and L. Zhang (2021), N. A. Gowtham, S. Deepakq and D. Patra (2020)). In Table 1, Python language was used as an experimental environment for testing, and Tensorflow and Pytorch were used as artificial intelligence learning platforms.

Table 1. Development Environment

DIV	Configuration
OS	Windows 10
RAM	16G
VGA	GTX1080ti
Language	Python 3.9.12
Platform	Pytorch 1.11.0/ Tensorflow-gpu 2.3.1
CUDA version	CUDA 11.7
CUDDNN version	cuDNN v8.4.1

Figure 4 is an experimental procedure for improving the image quality of the tower.

(a) Colab is installed, (b) Colab is installed, and after uploading the source from the cloud, (c) selects and changes the runtime type between CPU and GPU (GPU type is selected in this study). (d) modifies the cell part attached to Colab and completes runtime preparation. (e) executes the modified code.

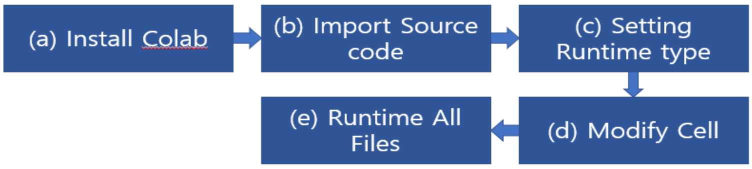


Fig. 4. an experimental procedure

in this study, the experiment was conducted through Google’s Colaboratory's Jupyter Notebook. Google's Colab has the same functionality and performance as Jupyter Notebook without installing Python and using a graphics card(D. Lee, S. Lee, H. Lee, K. Lee and H. -J. Lee(2019), S. N. Ferdous, A. Dabouei, J. Dawson and N. M. Nasrabadi (2021), W. Ma, Z. Pan, J. Guo and B. Lei (2018)). The journal needs to be purchased with discount for author(s) through an order form accompanying the acceptance letter.

Fig. 5. The Pagoda Dataset for SRGAN

Figure 5 is an image file with improved image quality for SRGAN learning. The training data is divided into those with and without background. In addition, low-resolution and high-resolution input data were tested by type for test evaluation.

2.3.2 Upload Data for Training

AI learning using Google's colab uploads data to the server (FTP) and proceeds after connecting the URL in colab. Figure 6 shows the connection link between colab and server (FTP). Figure 7 shows the Connect to Server for AI Learning of SRGAN.

```
1 !wget "http://beom2581.dothome.co.kr/test1.jpg" -O original.png
```

Fig. 6: Editable Image Path for SRGAN

```
--2022-08-19 06:24:13--  http://beom2581.dothome.co.kr/test1.jpg
Resolving beom2581.dothome.co.kr (beom2581.dothome.co.kr)... 112.175.184.60
Connecting to beom2581.dothome.co.kr (beom2581.dothome.co.kr)|112.175.184.60|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 233643 (228K) [image/jpeg]
Saving to: 'original.png'

original.png        100%[==================>] 228.17K   754KB/s    in 0.3s

2022-08-19 06:24:13 (754 KB/s) - 'original.png' saved [233643/233643]
```

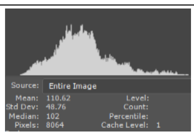
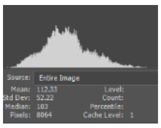
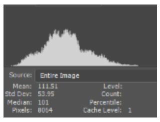
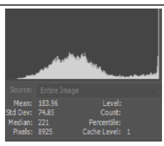
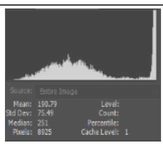
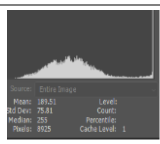
Fig. 7: Connect to Server for SRGAN Learning of AI

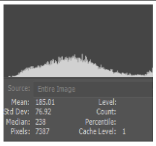
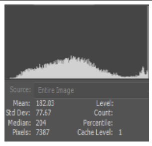
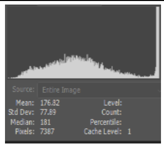
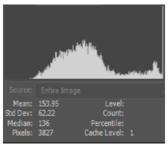
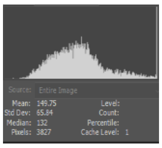
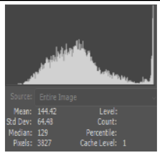
A model trained on the DIV2K dataset (bicubic down sampling images) on a 128 x 128 image patch.

2.3.3 Analysis of Experimental Results

The final result performed through training is output in three resolutions: high-resolution, low-resolution and super-resolution. Figure 8 shows the images for each resolution.

Table 2. pagoda resolution and histogram analysis

DIV	Original Resolution Image	High Resolution Image	Super Resolution Image
Entire Image			
Center			
			
Top			
			

Down			
			
Pedestal			
			

In this study, as a result of applying the SRGAN algorithm, it was confirmed that the resolution improvement was clear in the low-resolution Pagoda image. Also, it was confirmed that the number of learning iterations was affected by the resolution (H. N. Pathak, X. Li, S. Minaee and B. Cowan (2018), Z. San-You, C. De-Qiang, J. Dai-Hong, K. Qi-Qi and M. Lu(2020)).

Table 3. Gray and Color histogram analysis code for the Pagoda Images

Gray Histogram Analysis Code

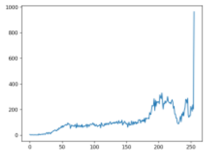
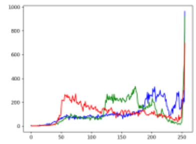
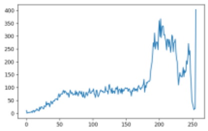
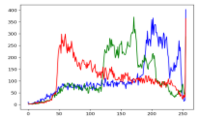
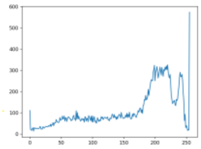
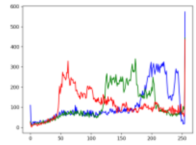
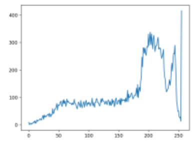
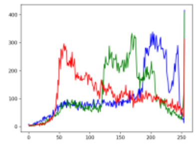
```
1   import cv2
2   import matplotlib.pyplot as plt
3   src1=cv2.imread('4.jpg',cv2.IMREAD_COLOR)
4   hist=cv2.calcHist([src1],[0],None,[256],[0,256])
5   cv2.imshow('src1', src1)
6   cv2.waitKey(1)
7   plt.plot(hist)
8   plt.show()
9   cv2.destroyAllWindows()
10  img = [src1]
11  for i in img:
12      hist = cv2.calcHist([i], [0], None, [
13          256], [0, 256])
14      plt.plot(hist)
```

Color Histogram Analysis Code

```
1   import cv2
2   import matplotlib.pyplot as plt
3   src1 = cv2.imread('4.jpg')
4   color = ['b', 'g', 'r']
5   channel = cv2.split(src1)
6   for (i, j) in zip(channel, color):
7       hist = cv2.calcHist([i], [0], None, [
8           256], [0, 256])
9       plt.plot(hist, color=j)
10  cv2.imshow('src1', src1)
11  cv2.waitKey(1)
12  plt.show()
13  cv2.destroyAllWindows()
```

The image histogram analysis source, Table 3, is interpreted as follows: Line 4: cv2.calcHist(images, channel, mask, histSize, ranges [, hist [, accumulate]]) []: optional, Arg.: images : Input image - list format[]/ channels: List indicating channels to obtain histogram [0], [0,1], [0,1,2]/[0] for GRAYSCALE image/[0] for BGR image: B, [1] : G, [2] : R/ mask : To obtain a histogram for the entire mask image and input image None/histSize : A list indicating the size of each dimension of the histogram. If it is set to 64, then 64 pieces are grouped by 4 in such a way as 0~3, 4~7, etc. If it is set to 128, it is expressed as 128, grouped by 2 in such a way as 0~1, 2~3, etc. ranges: A list of the minimum and maximum values of each dimension of the histogram. hist: computed histogram (numpy.ndarray) accumulate : “True” to accumulate on an existing histogram, “False” to create a new one. Also, In the color image, each channel was expressed by dividing the channel through cv2.split. The cv2.split function splits channels, and splits b, g, and r channels in a list format.

Table 4. Gray and Color histogram for the Pagoda Images

Item	Image	Histogram	
		Gray	Color (RGB)
Iteration- 50 **Time Taken** **03:17**			
Iteration- 100 **Time Taken** **08:33**			
Iteration- 250 **Time Taken** **15:24**			
Iteration- 500 **Time Taken** **28:03**			

Experimental results did not show a significant effect when the number of lessons was repeated 50 times. Rather, it was confirmed that the result was similar to that of the LR (Low Resolution) image. It was confirmed that SRGAN works when the number of times of learning is set to 100, but it was confirmed that the overall image was unclear than that of 250 times. Finally, the learning was doubled and the result of learning 500 times was confirmed with high-resolution image results. In this study, color distribution was confirmed using histogram analysis to objectively analyze the resolution. Table 4 shows Gray and Color analysis of pagoda by number of repetitions.

2.3.4 Conclusion

The restoration of pagoda proceeds with research based on proven data and the thoughts of expert engineers. This restoration work needs to restore the resolution of the data collected by the preprocessing process. In particular, the resolution of old image data is low, making it difficult to restore, and artificial intelligence restoration based on images may produce inaccurate results. In this study, the quality of the pagoda image was improved through the SRGAN algorithm among GAN algorithms. In the experiment, image quality improvement was attempted for the whole, top, middle, and pedestal of pagoda, and analysis was conducted through histograms. The experimental result confirmed that the resolution improvement was good for the low-quality image, and the degree of improvement of the image quality was different according to the number of repetitions of the AI. Gray and Color (Red, Green, Blue) histogram analysis was performed to be used as an index of pagoda image quality. As a future research area, it is considered necessary to conduct an experiment comparing artificial intelligence restoration with improved image quality and artificial intelligence restoration without image quality improvement.

3. Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT). (No. 2020R1A2C1007668)

References

[1] Johnson, Justin & Alahi, Alexandre & Fei-Fei, Li. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. 9906. 694-711. 10.1007/978-3-319-46475-6_43. DOI: https://doi.org/ 10.48550/ arXiv.1603.08155

[2] J. Li, L. Wu, S. Wang, W. Wu, F. Song and G. Zheng(2019), "Super Resolution Image Reconstruction of Textile Based on SRGAN," 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), 2019, pp. 436-439, doi: 10.1109/SmartIoT.2019.00078.

[3] S. H. Salem Hussin and R. Yildirim (2021), "StyleGAN-LSRO Method for Person Re-Identification," in IEEE Access, vol. 9, pp. 13857-13869, doi: 10.1109/ACCESS.2021.3051723.

[4] J. Osorio Ríos, A. Armejach, G. Khattak, E. Petit, S. Vallecorsa and M. Casas(2020), "Evaluating Mixed-Precision Arithmetic for 3D Generative Adversarial Networks to Simulate High Energy Physics Detectors," 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 49-56, doi: 10.1109/ICMLA51294.2020.00017.

[5] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner (1998), "Gradient-based learning applied to document recognition", Proceedings of the IEEE, pp. 2278-2324.

[6] D. Salamani et al., (2018), "Deep Generative Models for Fast Shower Simulation in ATLAS", Proceedings 14th International Conference on e-Science: Amsterdam Netherlands October 29-November 1 2018, pp. 348.

[7] X. Jiang, Y. Xu, P. Wei and Z. Zhou (2020), "CT Image Super Resolution Based on Improved SRGAN," 2020 5th International Conference on Computer and Communication Systems (ICCCS), 2020, pp. 363-367, doi: 10.1109/ICCCS49078.2020.9118497.

[8] N. Xiang, B. Tang and L. Wang (2022), "Image Super-Resolution Method Based on Improved Generative Adversarial Network," 2022 IEEE 5th International Conference on Electronics Technology (ICET), 2022, pp. 1207-1212, doi: 10.1109/ICET55676.2022.9824258.

[9] X. Hou, T. Liu, S. Wang and L. Zhang (2021), "Image Quality Improve by Super Resolution Generative Adversarial Networks," 2021 2nd International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI), 2021, pp. 117-121, doi: 10.1109/ICHCI54629.2021.00031.

[10] N. A. Gowtham, S. Deepakq and D. Patra(2020), "Super-Resolution Generative Adversarial Network with Modified Architecture for Single Image Super-Resolution," 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP), 2 pp. 1-6, doi: 10.1109/ICCCSP49186.2020.9315285.

[11] D. Lee, S. Lee, H. Lee, K. Lee and H. -J. Lee(2019), "Resolution-Preserving Generative Adversarial Networks for Image Enhancement," in IEEE Access, vol. 7, pp. 110344-110357, doi: 10.1109/ ACCESS.2019.2934320.

[12] S. N. Ferdous, A. Dabouei, J. Dawson and N. M. Nasrabadi (2021), "Super-resolution Guided Pore Detection for Fingerprint Recognition," 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 8085-8092, doi: 10.1109/ICPR48806.2021.9413043.

[13] W. Ma, Z. Pan, J. Guo and B. Lei (2018), "Super-Resolution of Remote Sensing Images Based on Transferred Generative Adversarial Network," IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, 2018, pp. 1148-1151, doi: 10.1109/IGARSS.2018.8517442.

[14] H. N. Pathak, X. Li, S. Minaee and B. Cowan (2018), "Efficient Super Resolution for Large-Scale Images Using Attentional GAN," 2018 IEEE International Conference on Big Data (Big Data), pp. 1777-1786, doi: 10.1109/BigData.2018.8622477.

[15] Z. San-You, C. De-Qiang, J. Dai-Hong, K. Qi-Qi and M. Lu(2020), "Adaptive Diagonal Total-Variation Generative Adversarial Network for Super-Resolution Imaging," in IEEE Access, vol. 8, pp. 57517-57526, 2020, doi: 10.1109/ACCESS.2020.2981726.