Generating New Ulos Motif with Generative AI Method in Digital Tenun Nusantara (DiTenun) Platform

Humasak Simanjuntak[1]*, Evelin Panjaitan[2], Sandraulina Siregar[3], Unedo Manalu[4], Samuel Situmeang[5], Arlinta Barus[6]
Department of Information Systems, Faculty of Informatics and Electrical Engineering, Institut Teknologi Del[1,2,3,4,5]
Department of Informatics, Faculty of Informatics and Electrical Engineering, Institut Teknologi Del[6]
Jl. Sisingamangaraja, Sitoluama, Laguboti, Toba, Indonesia, 22381[1,2,3,4,5,6]

***Abstract*—DiTenun is a startup developing a platform that utilizes artificial intelligence to create innovative digital textile patterns for woven fabrics. One of the woven motifs produced is the Ulos motif, a traditional weaving from the Batak tribe that consists of various types, patterns/motifs, and sizes. Currently, DiTenun platform applies two methods to generate Ulos motifs: image quilting and SinGAN. The image quilting method uses synthetic textures to form a new texture by combining blocks from the original texture. The SinGAN is a Generative Adversarial Network (GAN) method that accepts one image motif as input to generate a new motif that resembles the training motif. The new motifs generated by both methods are still repetitive and not diverse (less variation). Therefore, this paper focuses on improving the StyleGAN method, which utilizes two or more Ulos motif images as input to produce new innovative motifs by mixing regularization. Six experimental scenarios are carried out on the Ulos motif image dataset with different numbers of input motifs and hyperparameter tuning. The experiment results are new images with diverse patterns, colour combinations, and merge motif elements. The StyleGAN performance is measured with Frechet Inception Distance (FID) and Kernel Inception Distance (KID) to find the best-quality motif generated based on the six hyperparameter tuning scenarios. The results show that the fourth scenario on Ulos Batak Karo, Gundur Category (Min Max Resolution: 8 and 256, number images 4, on training iteration per resolution = 100000 and max iteration = 50000000) is the best motif generated, based on FID and KID score, are 91.32 and 0.04, respectively.**

Keywords—Generate Ulos motif; StyleGAN; DiTenun; generative AI Ulos motif

I. Introduction

DiTenun is a start-up that creates digital weaving motifs based on artificial intelligence applications. The start-up was established to support the woven clothing industry in Indonesia, mainly traditional cultural weaving so that weavers could produce more diverse designs and adapt to customer preferences. One of the traditional weaves that are culturally popular in Indonesia is Ulos. Ulos weaving is a typical cloth of the Batak tribe, consisting of various types, sizes, patterns, or motifs. Traditional Ulos are made using a loom instead of a machine, and red, black, and white colours usually dominate the patterns. Most Tapanuli people consider Ulos a symbol of bonds of affection, position, and communication in the Batak traditional community.

Currently, the DiTenun platform applies two methods to create digital weaving motifs: image quilting and GAN algorithm. Image quilting exploits synthetic textures to form a new larger texture by combining blocks from the original [1] [2]. The weakness of this process is produces images with patterns that repeatedly appear in a particular direction or unnatural transitions because of poor patch selection. Furthermore, image quilting has limited generalization, so it struggles with more complex, non-repetitive textures with significant structural elements, like the Ulos motifs.

Generative Adversarial Networks (GANs) stand out as a robust deep-learning model for image generation. Comprising a generator and a discriminator, GANs are trained and tested to produce images that are indistinguishable from real ones. The versatility of the GAN algorithm is evident in its applications, which span image generation [3] [4] [5], image in painting [6] [7] [8] [9], test generation [10], medical image processing [11], semantic segmentation [12] [13], image colourization [14] [15], image-to-image translation [16] [17], and art generation [18] [19]. Implementing GAN also involves translating sketch images, which do not precisely represent the real entity boundaries and are not spatially aligned with the entity, synthesizing more realistic images, and creating diverse images [20]. Moreover, generative AI for text-to-image conversion using diffusion models is also emerging as significant research [21] [22]. Notable examples include DALL-E, which creates imaginative and contextually accurate images from text descriptions. These generative AI systems are at the forefront of AI technology, demonstrating their 'creativity' and adaptability across various applications.

The GAN algorithm has also been implemented on the DiTenun platform using the SinGAN algorithm. SinGAN is a GAN development that learns from a single natural image and produce new high-quality with the same visual content as the input [23] [24]. Based on the DiTenun platform, SinGAN employs one image as input to generate a new motif image. Thus, the diversity and quality of the generated motifs heavily depend on the input motif. The generated motifs may also lack diversity and richness if the input motifs lack sufficient detail or variation. Although SinGAN can generate diverse motifs, the quality of these generated motifs may not always match with the input motifs. In particular, generated motifs might exhibit artefacts, especially at larger scales or when the input motifs have complex textures and structures. We also found

*Corresponding Author.

that difficult to control the attributes of the generated motifs because the output is more stochastic, making it harder to direct the synthesis operation towards specific desired characteristics. Therefore, this research addresses these challenges and further unlocks the potential of GANs by proposing implementing the StyleGAN method to generate new Ulos motifs.

StyleGAN is a GAN method that uses two or more images as input to generate new motifs by employing mixing regulations [25]. Mixing regulation is the merging of image features to produce new images so that the output obtained has features from the input image. The input image consists of the original and style images. The input images can be a reconstructed result of the original or image style. Not only improving the style in mixing regularization, the StyleGAN method also produced images with a high resolution of 1024 × 1024 pixels [26] [27]. Furthermore, hyperparameter tuning, which consists of iteration, resolution, and batch size, is required to enhance the style of the image produced by the StyleGAN method.

This research marks a pioneering effort in the field, being the first to apply generative AI to create Ulos motifs. Prior studies on Ulos motifs have focused on traditional methods and manual design techniques. However, by leveraging the capabilities of generative AI, this study introduces an innovative approach that enhances the design process and preserves the intricate patterns and cultural significance of Ulos motifs. The integration of AI technology in this context represents a significant advancement, opening new avenues for artistic expression and cultural preservation through modern computational techniques.

Furthermore, this paper contributes to improving the StyleGAN algorithm and conducts an extensive experiment to produce new Ulos motifs on DiTenun. The experiment expected to generate a good-quality, rich, diverse Ulos motif by testing more Ulos categories and hyperparameter tuning. The StyleGAN algorithm results are evaluated with Frechet Inception Distance (FID), Kernel Inception Distance (KID), and graph loss metrics. To manage and enrich ulos image collections, an ulos repository is developed by collecting, storing, and combining the real ulos motif with the new motif generated by the StyleGAN.

II. MATERIALS AND METHOD

A. Dataset

The Ulos dataset was built based on surveys to the weaver and images generated by the platform. The dataset was collected in a repository and categorized according to the types of Ulos motifs from 3 Batak tribes: Toba, Karo, and Simalungun. The dataset contains 20 Ulos types, each consisting of 1 to 29 motifs. The StyleGAN algorithm trains with Ulos images from this dataset to learn the pattern of the Ulos motif and to generate new motifs. Fig. 1 is an example of an input image for several Ulos motifs.

Fig. 1. The example of Ulos motif.

Table I details the types of Ulos used and the number of motifs for each type.

TABLE I. TYPE ULOS MOTIFS

Ulos Type	Motif Code	#Motif
Boolean	BL	7
Bintang Maratur	BM	5
Marpisoran	MP	5
Mangiring Simareurreur	MS	4
Pinucca	PNC	24
Sadum Angkola	SA	11
Suri-suri	SS	1
Sumbat	S	5
Uis	U	6
Julu	JL	3
Bekan Bulu	BB	2
Sigara-gara	SGR	2
Gundur	GDR	4
Indung Bayu	IB	2
Sori-sori Simalungun	SR	29
Hati Rongga	HR	16
Tapak Catur	TC	3
Gobar	GBR	4
Sori-sori Pakpak Barat	SSi	17
Perbunga Mbacang	PM	6

The Ulos motif is used as a type of geometric motif, consisting of a combination of lines and dots which form a geometric pattern such as curved lines, circles, triangles, and other geometric shapes. The Ulos motif is a cross-stitch pattern, making it easy to weave later. The image is in PNG format, and the size is not specified, while the resolution used is a minimum of 8 x 8 and a maximum of 1024 x 1024. The high or low image resolution is used to inspect the influences of the resolution on the output results. We use progressive growing to produce high-resolution images. This method starts with a low-resolution image and gradually adds generators and discriminators to increase the image resolution. Then, the image styles are combined into each layer using AdaIN. Fig. 2 shows the sample Ulos motif dataset used as input to the StyleGAN model.

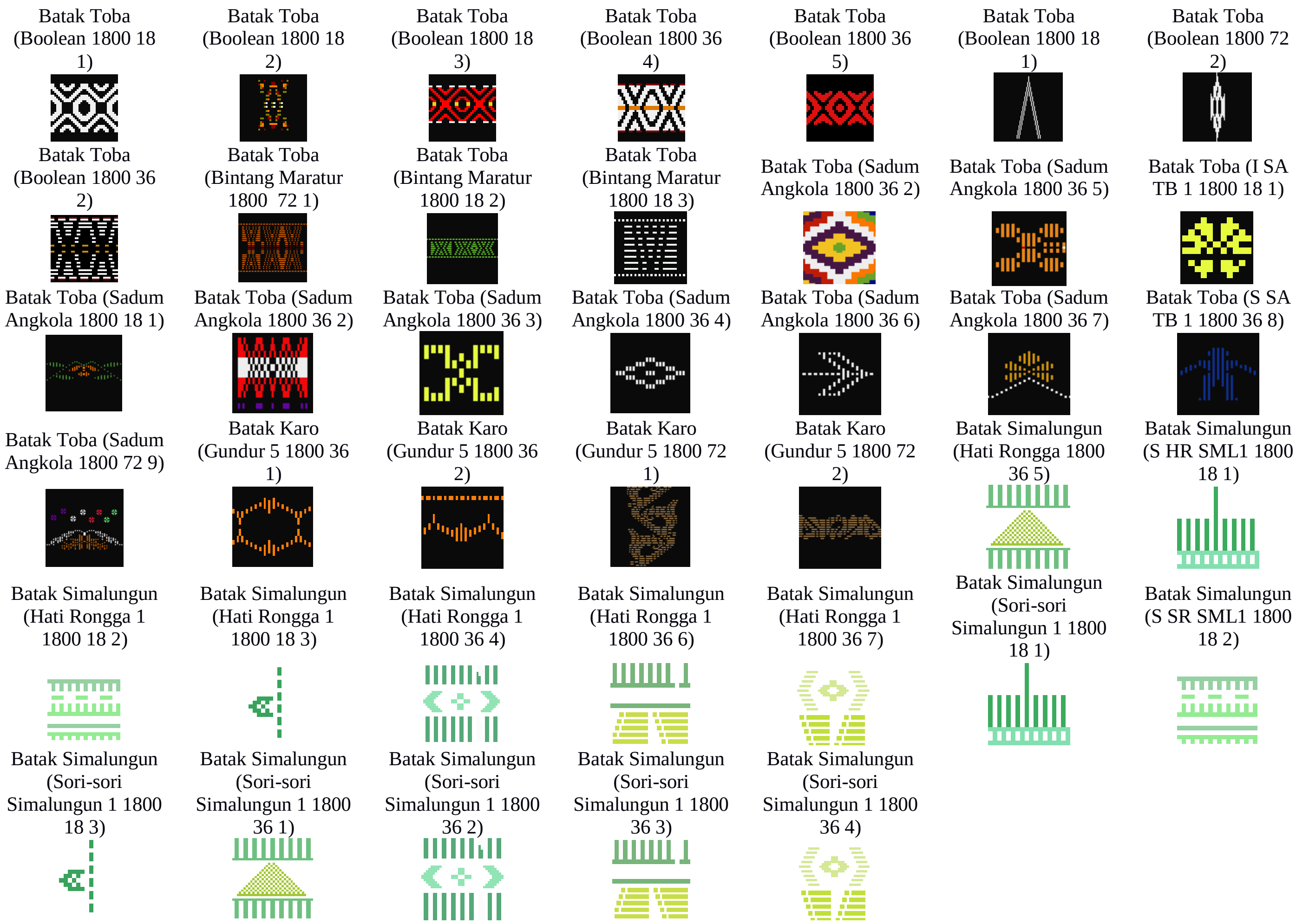


Fig. 2. The sample of Ulos motif dataset.

B. Architecture of StyleGAN

StyleGAN (Style-Based Generator Architecture for Generative Adversarial Networks) is a GAN development that can control image synthesis. It consists of 2 networks: a generator and a discriminator. The generator generates a realistic sample of random noise and tries to trick the discriminator. The generator network accepts a vector number z as input, where z is a three-dimensional image. The input vector z is randomly generated, and the generator can create an arbitrary image different from the input [4]. The second network, the discriminator, is used to identify whether the sample generated by the generator is genuine or fake. The discriminator network, a binary classification network, accepts the input of a three-dimensional image and states that the input is an original image from the dataset, or an image made by a generator. The discriminator network is a combination of several generator networks. This means that this network stores all data from the input and sends it to the generator network. If the discriminator declares a fake image, it will be returned as feedback to the generator network. Fig. 3 shows the general architecture of StyleGAN method.

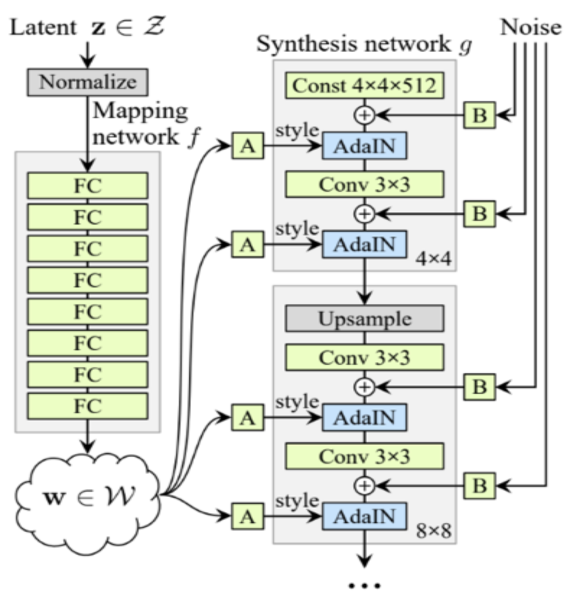


Fig. 3. Architecture of StyleGAN method [25].

StyleGAN is a revolutionary generator that autonomously dissects various aspects of the image without any human intervention. In the context of StyleGAN, an image is a collection of 'styles', each controlling the effect at a specific scale. This innovative generator separates unimportant variations from high-level attributes, enhancing image quality significantly. Moreover, it controls visual features by modifying the input of each level in the network separately, from coarse features to fine details [25].

We designed experiments with StyleGAN using input from Ulos motifs. The general flow of the implementing style to generate new motif Ulos can be seen in Fig. 4.

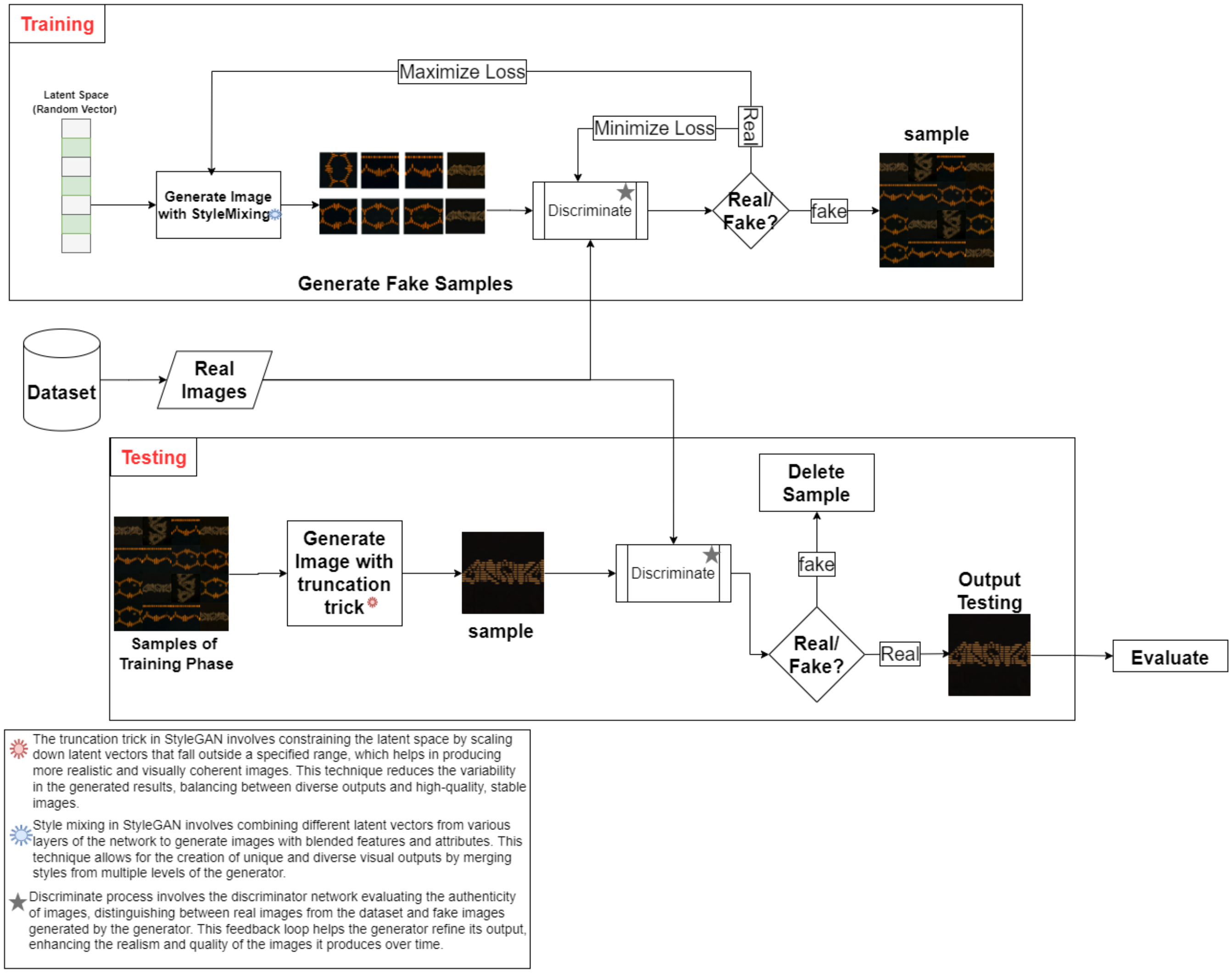


Fig. 4. System design architecture.

C. Design Experiment

As seen in Table II, we meticulously defined six scenarios to determine the best image combination input and tuning hyperparameter in generating the new motif. The hyperparameters used in this experiment are resolution, iteration, and batch size. At the training stage, we rigorously tune the resolution and iteration hyperparameters to compare the resulting motif for iterations in each experiment in different settings. We also measure the effect of iteration on the number of sample images produced and its impact on training time. We used 1024 in two experiments and 256 resolutions in the other four experiments. The StyleGAN method utilises more than one input Ulos motif to determine the effect of the process performance on the number of datasets used. At the testing stage, batch sizes 1 and 4 are used to determine the effect of the batch size on the test results in terms of frame and time.

TABLE III. HYPERPARAMETER SETTING EXPERIMENT

#Scenario	Resolution		Dataset			Training		Testing
	Min	Max	Ulos Type	Category	#Image	Iteration per Resolusi	Max Iteration	Batch Size
1	8	1024	Batak Toba	Sadum Angkola	2	100000	25000000	1 and 4
2	16	256	Batak Toba	Sadum Angkola	11	1200000	25000000	1 and 4
3	16	256	Batak Toba	Sadum Angkola	11	100000	2000000	1 and 4
4	8	256	Batak Karo	Gundur	4	100000	5000000	1 and 4
5	8	1024	Batak Toba	Boolean dan Bintang Maratur	11	100000	25000000	1 and 4
6	8	256	Batak Simalungun	Hati Rongga dan sori-sori	14	100000	3000000	1 and 4

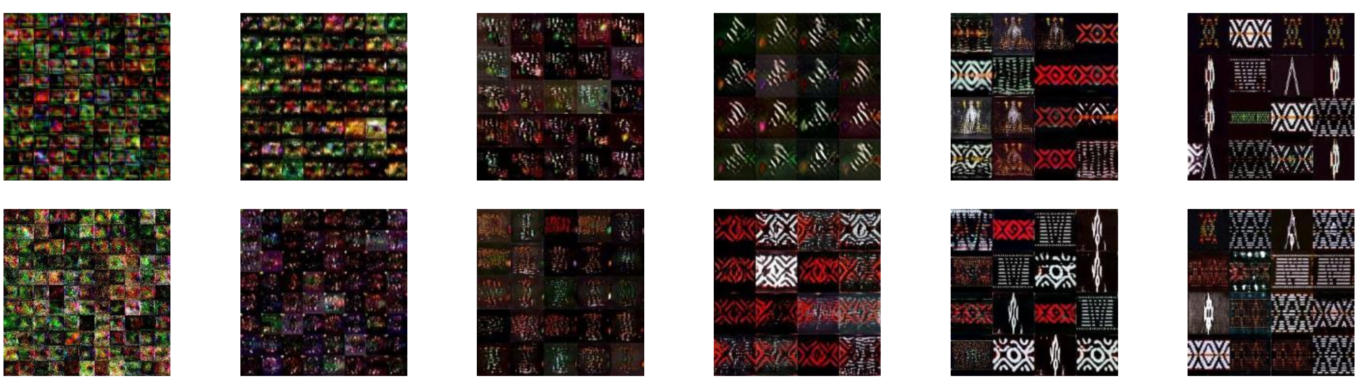
III. RESULT AND DISCUSSION

The main aim of this research is to leverage the power of the StyleGAN architecture in generating new and more diverse images of Ulos motifs by considering the Ulos motifs provided as input. As mentioned in section IIB, the StyleGAN architecture, a key tool in this research, was developed and trained using Ulos motifs in the repository. Applying the StyleGAN method to produce new Ulos motifs is a multi-stage process. It begins with training, which requires more than one image input with the intention to modify and combine one motif with another. Modifications involve adding colour, changes in pattern or shape, or reducing patterns and colours in an image. This process is then followed by testing, drawing, and evaluation, each contributing to the method's overall success in generating new Ulos motifs. All experimental studies were carried out on Ulos motifs, and StyleGAN results were evaluated using Frechet Inception Distance (FID) and Kernel Inception Distance (KID). FID quantifies the realism (realistic) and variety of motifs that StyleGAN generates. KID computes the square of the maximum mean difference of inception representation to measure the distinction between the generated and actual samples.

A. Training Result

Two critical networks work in the StyleGAN training process: the generator and the discriminator. The generator network creates or modifies the dataset, producing new image samples. Simultaneously, the discriminator network plays a crucial role in verifying the authenticity of the generator's output, distinguishing between real and fake images. If all the components of the new generated image are derived from the images in the input dataset, it is deemed real.

Fig. 5 illustrates the fascinating evolution of image clarity as the resolution increases. At a resolution of 16, the displayed images, each with 32 frames, show a hint of colour, but the dataset's pattern remains elusive. Stepping up to resolution 32, the pattern starts to emerge, albeit with colours from both datasets. The pattern becomes more discernible at resolution 64 despite minor changes in colour or shape. By the time we reach resolution 128, style mixing comes into play, combining the source and destination images to create a new one. However, the sample result is still somewhat blurry. At resolution 256, style mixing continues, and the sample results show noticeable improvement. The same holds for 512 and 1024 resolutions, where style mixing is present, but the image from the 1024 sample stands out with its sharpness and quality, surpassing the results from 64, 128, and 256 resolutions.



a. 16 b. 32 c. 64 d. 128 e. 256 f. 512

Fig. 5. Sample training result for 16 - 512 resolution.

Iteration hyperparameters affect the training process. The more iterations, the more image samples are generated. The training results in experiments 1, 2, 5, and 6 had a more significant number of iterations and delivered a greater number of motif samples. Fig. 6 shows the duration of the training time, the total number of iterations, and the total number of samples generated from the training process.

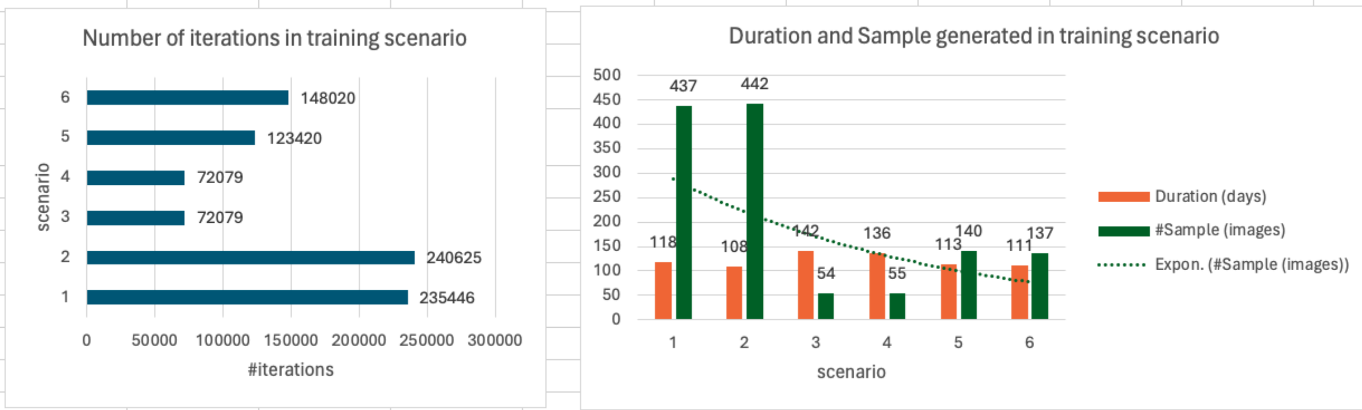


Fig. 6. Number of iterations, duration, and sample generated in the training scenario.

B. *Testing Result*

The testing was conducted using a sample produced during the training process, which will serve as the dataset. During the testing phase, the two GAN networks operate in the same manner as they do during the training process. The outcome of the testing process is a new and realistic image motif. Although this new motif exhibits some alterations, it retains characteristics from the original dataset, such as partially missing motifs, combined motifs, and altered pattern directions. Additionally, there are color changes resulting from color blending. Below is a discussion of the new motif images generated from the six experiments conducted.

1) Motif alteration: Alterations in motif patterns involve directional changes, such as shifting from left to right. Additionally, some patterns may be missing, or new patterns added. Fig. 7 illustrates one of the motifs with altered patterns. These changes pertain to the motif patterns themselves and do not involve combinations with other datasets.

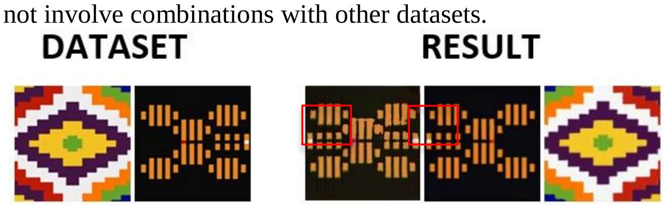


Fig. 7. Testing result for motif alterations.

2) Discoloration: The color changes observed in the test results are influenced by the colours present in other input motifs. Fig. 8 demonstrates the application of these results on one of the datasets used, ulos Batak Toba (Boolean 1800 36 2). Initially, the motif comprised only two colors: white and cream. Following the training and testing phases, the resulting motif image incorporates red and orange, blending with colors from other input motifs.

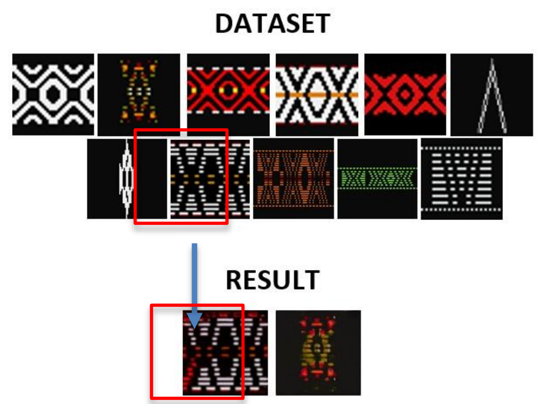


Fig. 8. Testing result for motif discoloration.

3) Merging of Ulos motifs: The implementation results demonstrate the merging of the ulos motif, resulting from the input of multiple motifs and the combination of two or more motifs. As shown in Fig. 9, there are 11 input motifs. After the training and testing process, StyleGAN generates several outputs. One of the test results combines two input motifs, specifically Batak Toba (Boolean 1800 18 3) and Batak Toba (Boolean 1800 18 1), producing a new motif that blends these two input motifs.

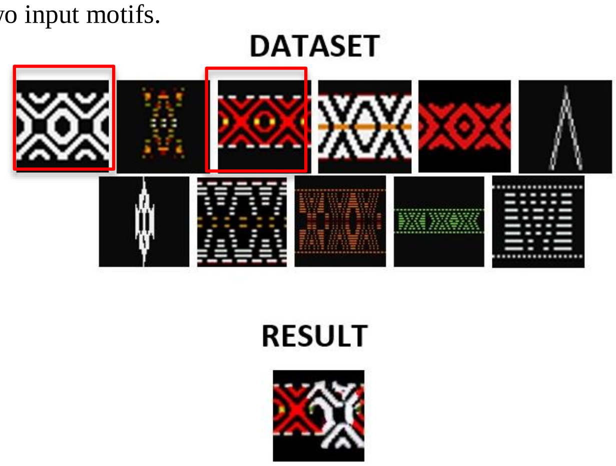
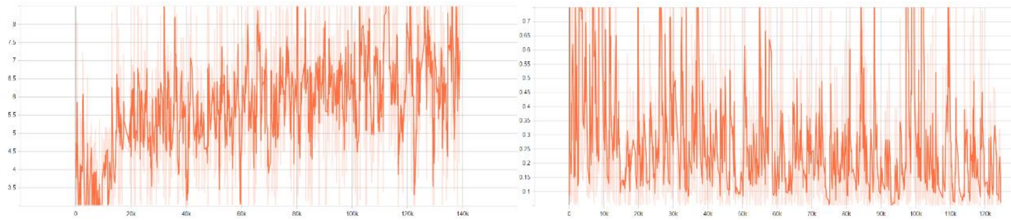


Fig. 9. Merging Ulos motif testing result.

C. Evaluation

1) Training phase evaluation: Following the testing phase, an evaluation process was conducted. The metrics employed for this evaluation included graph loss, FID, and KID. The graph loss metric encompasses both the generator loss and the discriminator loss. Experimental results indicated that iteration is the hyperparameter influencing the loss graph, as the number of iterations impacts the generator's performance. The results for the generator and discriminator loss are presented in Fig. 10.

Fig. 10 displays the generator and discriminator loss from scenario 4. The fluctuations in the generator and discriminator graphs reflect the competition between the two networks to create a realistic sample. The generator attempts to create a motif identical to the original so that the discriminator cannot distinguish that the motif is fake. When attempting to obtain a sample, the generator's graph rises while the discriminator's graph falls. According to the generator loss graph, there is an upward trend up to iteration 110000, with the generator loss value reaching 7.02 and the discriminator loss value dropping to 0.06. The sample image at iteration 110000 is shown in Fig. 11.



(a) (b)

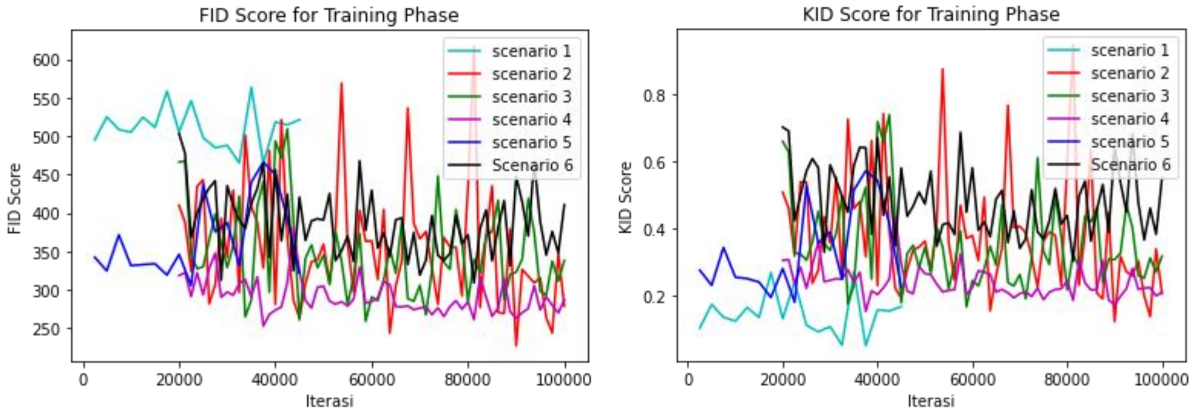
Fig. 10. Generator loss (a) and discriminator loss (b).

In the discriminator loss, it is observed that no sample motif is generated when the discriminator loss peaks at iteration 92622 with a value of 15.75 and the generator loss is 2.99. This occurs because when the discriminator maximizes its performance, it identifies the generator's motif as "fake," causing it to fail the discriminator's scrutiny, and thus, no sample motif is produced. According to the primary theory of loss functions, a lower generator loss typically indicates higher-quality output, while a higher generator loss suggests lower-quality output, which may lead the discriminator to classify it as fake. Similarly, a lower discriminator loss indicates superior performance by the discriminator in accurately distinguishing real image motifs.

Fig. 11. Sample generated motif from scenario 4.

The subsequent evaluation uses FID and KID metrics to assess the diversity of motifs generated by the model and gauge the disparity between the generated motifs and actual samples. When FID and KID scores approach 0, it indicates that the generated images closely resemble the input dataset, making it more challenging for the discriminator to differentiate them as fake. Conversely, higher FID and KID values indicate more significant dissimilarity between the generated and actual motifs. Scenario 4 with the Batak Karo dataset exhibits the lowest FID and KID scores among the six experiments conducted. Fig. 12 below illustrates the graph depicting FID and KID scores during the training phase, showcasing the highest resolution achieved in each experiment.

As depicted in Fig. 12, each graph varies in length due to differences in the number of iterations across experiments. Observing the height of each graph in the scenarios, scenario 4 (represented by the purple line) utilizing the Batak Karo motif dataset exhibits a consistent graph that approaches a value of 0 steadily up to 100000 iterations. On the other hand, scenario 1, which employs the Ulos Batak Toba dataset, achieves a commendable KID score but displays the worst FID score. This outcome indicates that the generated motifs are inconsistent and differ significantly based on the input dataset. Furthermore, this training scenario is only effective up to 60000 iterations. Other scenarios also demonstrate inconsistent results and perform less effectively than scenario 4. Therefore, increasing the number of iterations tends to improve the resulting sample's FID and KID scores, bringing it closer to 0 and enhancing its similarity to the original dataset.



Iteration Iteration

Fig. 12. Frechet inception distance and kernel inception distance score for training phase.

Overall, all experiments' FID and KID values remain above zero or significantly distant from zero. Nevertheless, the resulting motif images appear to bear resemblance to the input dataset. Upon analysis, the test image motifs exhibit background images resulting from style mixing between each dataset. This contrasts with the comparison to dataset images lacking such backgrounds. Moreover, the training experiment reached a maximum of 100,000 iterations. Therefore, additional iterations are advisable to achieve optimal results comparable to previous research with 1,200,000 iterations. This extended training process exceeding 100,000 iterations demands a longer duration and high computational specifications, particularly necessitating sufficient GPU memory for expediting training times.

There are three crucial things to pay attention to related to the fluctuating FID value in the training stage:

- Unstable Training Dynamics

  The fluctuating FID score shown in Fig. 12 might be associated with the unstable training dynamics shown in Fig. 10. Even after thousands of iterations, the generator and discriminator are still in flux, with neither achieving a clear dominance. This state can lead to the FID score bouncing around as the quality of generated images fluctuates. Therefore, learning rates, optimizers, and network architectures need further experimentation for a more stable training process. Techniques such as gradient penalty usually improve the stability of GAN training, so we also suggest further experimentation on this aspect.

- Batch Size Issues

  The FID score is calculated based on batches of images. If the batch size is too small, statistical noise can lead to significant fluctuations in the score, even if the overall quality improves. As shown in Table II, batch sizes used in all experiment scenarios are relatively small, 1 and 4. Therefore, further experimentation with larger batch sizes might help.

- Mode Collapse

  Mode collapse happens when the generator gets stuck in a loop, producing only a limited variety of images. This state can happen even with high iteration counts. While the generated images might be high quality within this limited range, the lack of diversity will cause the FID score to be unreliable.

2) Testing phase evaluation: During the testing process, three hyperparameters are utilized: iteration, resolution, and batch size. However, for this experiment, we focused on using the highest resolution value with default iterations, and batch sizes set to 1 and 4. According to the findings, smaller batch sizes result in longer testing times and fewer samples successfully producing results. This occurs because smaller batch sizes lead to more specific style generation for each sample. Fig. 13 illustrates the testing time required for each experimental scenario.

Fig. 14 displays the evaluation results of the StyleGAN algorithm using FID and KID metric. The FID score is calculated by comparing motifs generated during testing with the input dataset. The highest FID score of 336.03 was observed in batch size 1 within scenario 2, using the Batak Toba Ulos (Sadum Angkola) experiment. Conversely, the lowest FID score of 99.37 was achieved in scenario 4, which involved more than two similar inputs using the Batak Karo Ulos (Gundur). For batch size 4, scenario 1 with the Batak Toba Ulos (Sadum Angkola) experiment obtained the highest FID score of 388.69. Meanwhile, scenario 4 with the Batak Karo (Gundur) experiment had the lowest FID score, amounting to 153.25.

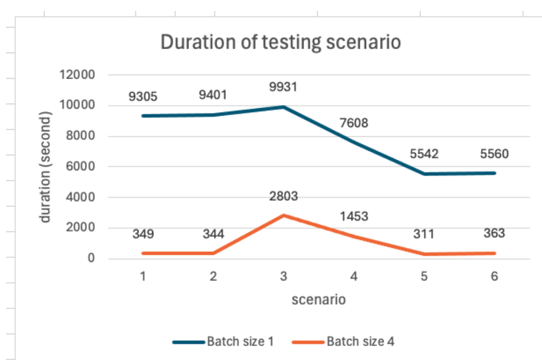


Fig. 13. Duration of testing scenario.

Similar to the FID score, in batch size 1, scenario 2 attained the highest KID score of 0.24, while scenario 4 which involving multiple similar inputs with the Batak Karo Ulos (Gundur) dataset, achieved the lowest KID score of 0.04. For batch size 4, scenario 1 using the Batak Toba Ulos (Sadum Angkola) experiment yielded the highest KID score of 0.34. Conversely, scenario 4 with the Batak Karo (Gundur) experiment had the lowest KID score of 0.08.

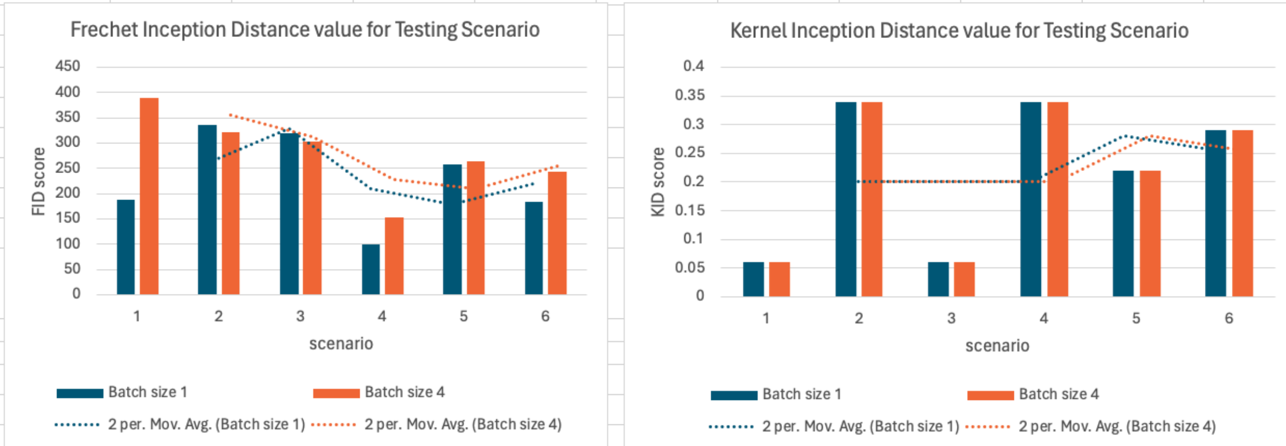


Fig. 14. Frechet inception distance and kernel inception distance value for testing scenario.

Based on the evaluation results, scenario 4 outperforms the other scenarios. To further validate this finding, the researcher distributed a questionnaire to an Ulos expert to determine if there were any changes in motifs, discoloration, or merging of motifs in the generated images.

According to Fig. 15, 83.87% of respondents indicated that there are motif changes in scenario 4. Additionally, 13% of respondents noted motif merging, 3.23% observed discoloration, and none reported no changes.

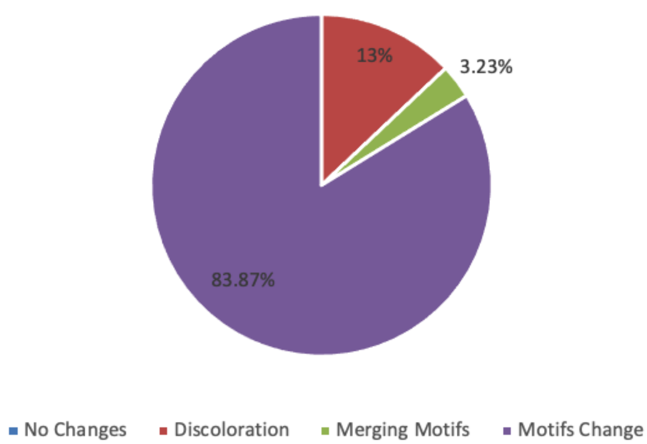


Fig. 15. Questionnaire for scenario results.

IV. Conclusion

Applying the StyleGAN method has resulted in the creation of novel Ulos motifs for the DiTenun application. These new motifs display unique characteristics, such as moving in the opposite direction, partially missing elements, and increased motifs. Additionally, the new motifs blend of colors from various datasets, effectively merging motifs to produce fresh Ulos designs. The experiment was conducted with three key hyperparameters: iteration, resolution, and batch size. The number of iterations directly influenced the number of samples generated. Scenario 2, with the highest iteration count of 240625, produced a substantial 442 samples, outperforming other experiments.

Based on the evaluation process, the best scenario is scenario 4, with an FID and KID score of 91.32 and 0.04, respectively. Scenario 4 produced better new motifs regarding image quality and similarity to the dataset. However, the experiment has yet to reach the maximum result, indicated by the training evaluation, which is still fluctuating. The image of the Ulos motif is not a natural image but a visual representation of the traditional decoration typical of the Batak tribe, which often displays complex geometric patterns and traditional symbols. This complex characteristic makes it challenging for generative AI algorithms to learn existing patterns, requiring further experiments. Potential future experiments related to learning rates, optimizers, batch size, and network architectures can be done for a more stable training process.

Furthermore, other generative AI methods have the potential to be implemented, especially the diffusion model, to get more interpretable latent space in order to capture variations and generate diverse samples, along with enriching the repository with more Ulos motifs. Moreover, the generative ulos motif necessitates thoroughly considering ethical and cultural dimensions to ensure these technologies serve diverse populations fairly and responsibly. By employing inclusive and diverse datasets, implementing bias mitigation strategies, and adhering to established ethical guidelines, developers can address potential biases and cultural insensitivities.

Acknowledgment

This work was supported by The Research and Community Service Body (LPPM) Institut Teknologi Del and Ministry of Education, Culture, Research and Technology of the Republic of Indonesia, through Kedaireka - Matching Fund Program 2022.

REFERENCES

[1] Efros and W. . T. Freeman, "Image Quilting for Texture Synthesis," in SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, Los Angeles, 2001.

[2] L. Raad and B. Galerne, "Efros and Freeman Image Quilting Algorithm for Texture Synthesis," Image Processing On Line, vol. 7, no. 2017, pp. 1-22, January 2017.

[3] 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 (ICCV), 2017.

[4] L. Wang, W. Chen, W. Yang, F. Bi and F. R. Yu, "A State-of-the-Art Review on Image Synthesis with Generative Adversarial Networks," IEEE Access, vol. 8, pp. 63514 - 63537, 20 March 2020.

[5] B. Zhang, S. Gu, B. Zhang, J. Bao, D. Chen, F. Wen, Y. Wang and B. Guo, "StyleSwin: Transformer-Based GAN for High-Resolution Image Generation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[6] H. Liu, Z. Wan, W. Huang, Y. Song, X. Han and J. Liao, "PD-GAN: Probabilistic Diverse GAN for Image Inpainting," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[7] X. Zhang, X. Wang , C. Shi, Z. Yan, X. Li, B. Kong, S. Lyu, B. Zhu, J. Lv, Y. Yin, Q. Song, X. Wu and I. Mumtaz, "DE-GAN: Domain Embedded GAN for High Quality Face Image Inpainting," Pattern Recognition, vol. 124, p. 108415, April 2022.

[8] Y. Yu, L. Zhang, H. Fan and T. Luo , "High-Fidelity Image Inpainting with GAN Inversion," in Computer Vision – ECCV 2022, Tel Aviv, 2022.

[9] A. Sargsyan, S. Navasardyan, X. Xu and H. Shi, "MI-GAN: A Simple Baseline for Image Inpainting on Mobile Devices," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023.

[10] W. Hu and Y. Tan, "Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN," in International Conference on Data Mining and Big Data, Singapore, 2022.

[11] Z. Ren, S. X. Yu and D. Whitney, "Controllable Medical Image Generation via GAN," Journal of perceptual imaging, vol. 5, p. 0005021–50215, January 2022.

[12] S. Wen, W. Tian, H. Zhang, S. Fan, N. Zhou and X. Li, "Semantic Segmentation Using a GAN and a Weakly Supervised Method Based on Deep Transfer Learning," IEEE Access, vol. 8, pp. 176480 - 176494, September 2020.

[13] H. Li, "Image semantic segmentation method based on GAN network and ENet model," The Journal of Engineering, p. 594–604, AUgust 2021.

[14] S. Huang, X. Jin, Q. Jiang, J. Li, S.-J. Lee, P. Wang and S. Yao , "A fully-automatic image colorization scheme using improved CycleGAN with skip connections," Multimedia Tools and Applications, vol. 80, p. 26465–26492, 04 May 2021.

[15] T. Bana, J. Loya and S. Kulkarni , "ViT - Inception - GAN for Image Colourisation," Machine Learning, Optimization, and Data Science, vol. 13163, p. 105–118, 02 February 2022.

[16] F. Xiong, Q. Wang and Q. Gao, "Consistent Embedded GAN for Image-to-Image Translation," IEEE Access, vol. 7, pp. 126651 - 126661, 05 September 2019.

[17] H. Emami, M. M. Aliabadi, M. Dong and R. B. Chinnam, "SPA-GAN: Spatial Attention GAN for Image-to-Image Translation," IEEE Transactions on Multimedia, vol. 23, pp. 391 - 401, 24 February 2020.

[18] W. Xu, C. Long, R. Wang and G. Wang, "DRB-GAN: A Dynamic ResBlock Generative Adversarial Network for Artistic Style Transfer," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021.

[19] Z. Jiang and P. Sweetser, "GAN-Assisted YUV Pixel Art Generation," In Australasian Joint Conference on Artificial Intelligence , vol. 13151, p. 595–606, 19 March 2022.

[20] W. Chen and J. Hays, "SketchyGAN: Towards Diverse and Realistic Sketch to Image Synthesis," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[21] R. Rombach, A. Blattmann, D. Lorenz, P. Esser and B.¨ Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.

[22] A. Ramesh, D. Prafulla, A. Nichol, C. Chu and M. Chen, "Hierarchical Text-Conditional Image Generation with CLIP Latents," arXiv preprint, vol. 2022, p. arXiv:2204.06125.

[23] T. R. Shaham, T. Dekel and T. Michaeli, "SinGAN: Learning a Generative Model From a Single Natural Image," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[24] H. Jain, R. Patil, I. D. Mastan and S. Raman, "Blind Motion Deblurring through SinGAN Architecture," arXiv preprint arXiv:2011.03705, 2020.

[25] T. Karras, S. Laine and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019.

[26] L. Han, S. H. Musunuri, M. R. Min, R. Gao, Y. Tian and D. Metaxas, "AE-StyleGAN: Improved Training of Style-Based Auto-Encoders," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2022.

[27] H. Li, J. Liu, X. Zhang, Y. Bai, H. Wang and K. Mueller , "Transforming the latent space of stylegan for real face editing," The Visual Computer, pp. 1-16, 22 August 2023.