Ancient Javanese Manuscript Reconstruction Using Generative Adversarial Network with StarGAN v2 Variations

Kukuh Cokro Wibowo[1*], **Fitri Damayanti**[2], **Fanky Abdilqoyyim**[3]

[1,2,3]Department of Information Systems, University of Trunojoyo Madura, Bangkalan, East Java, Indonesia.
E-mail: [1*]kukuhcokro411@gmail.com, [2]fitrid@trunojoyo.ac.id, [3]fankyab@gmail.com

(Received: 23 Dec 2024, revised: 17 Feb 2025, accepted: 19 Feb 2025)

Abstract

Ancient Javanese manuscripts are part of Indonesia's cultural heritage; most of them are usually in bad condition due to the age and environmental surroundings. This paper presents a manuscript reconstruction using the Generative Adversarial Network model, using the variation of StarGAN v2. The primary objective of this research is to assist philologists in reconstructing damaged manuscripts more efficiently, reducing the time and effort compared to manual reconstruction methods. The training for 100 epochs is performed by the model in order to generate the reconstruction image closest to ground truth. This study is done on a dataset that consists of a set of damaged manuscript images. In this dataset, 80% is for training, 20% is for validation, and 10 images are used for testing. Quality assessment will be made on image outputs during training, based on PSNR, SSIM, and LPIPS metrics. The results indicate that the PSNR increases from 16.1234 dB at the 50th epoch to 17.5588 dB at the 100th epoch, while the SSIM increases from 0.8374 to 0.8519, showing a strong improvement in image quality. Despite the LPIPS having a very slight increase from 0.1020 to 0.1051, this evidences that the model can be further improved. Overall, this study demonstrates that the StarGAN v2 model is effective in reconstructing ancient Javanese manuscripts-a great contribution to the field of cultural heritage preservation using modern technology.

Keywords: Ancient Javanese Manuscripts, Image Reconstruction, Generative Adversial Network, StarGAN v2.

I. INTRODUCTION

Ancient Javanese manuscripts are a very valuable cultural heritage, recording the history and knowledge of humanity for centuries. The existence of these manuscripts in Indonesia has to be preserved and maintained, as they are able to reveal the mindset of the people at that time [1]. However, many of these ancient Javanese manuscripts have been physically damaged over time due to age, unfavorable environmental conditions, and suboptimal storage methods. This is because damage causes the loss of an important part of the text contained in it, making it difficult to read and understand again.

Ancient Javanese manuscripts contain not only historical and mythological stories, but also religious, philosophical, and scientific knowledge of the past [1][2]. Most of these manuscripts are written on palm leaves, daluang paper, or other organic materials that are easily fragile. Moreover, the use of ancient Javanese script, which is rarely studied, makes it difficult for modern researchers to interpret and transliterate the manuscript.

Physical deterioration in manuscripts includes staining, tears, faded ink, and loss of certain portions. Causal factors involve the age of the organic material on which manuscripts were prepared, environmental conditions pertaining to light exposure, temperature, humidity, and infestation of insects, improper storage or absence of protective facilities such as a temperature-controlled room [3].

Over the last decades, the development of Artificial Intelligence technology, especially in the field of deep learning, has opened new opportunities for solving damaged image reconstruction problems. The Generative Adversarial Network has become a leading model in image data processing. GAN works with two networks: a Generator that generates new data and a Discriminator that evaluates whether the data is real or artificial [4]. The interaction between these two networks enables GANs to generate high-quality data, especially in the context of images. In this case, GANs can be used to reconstruct missing or damaged parts based on patterns present in the original data.

StarGAN V2 is a variant of GAN designed for multitransform image-to-image translation [5]. Compared to traditional GANs, StarGAN V2 has the advantage of generating more varied and realistic results by using style vectors, and provides cross-domain adaptability that enables image reconstruction with a variety of styles [5][6]. In this research, StarGAN V2 is used to reconstruct ancient Javanese

DOI: 10.34148/teknika.v14i1.1182

manuscripts by generating parts of the text that were damaged or missing according to the context and style of the original writing.

The paper, "StarGAN v2: Diverse Image Synthesis for Multiple Domains" by Yunjey Choi, et al., used the CelebA-HQ and AFHQ datasets to test the GAN model variations of StarGAN V2 [5]. The research focuses on the generation of high image diversity within the target domain and scalability across multiple target domains with only one model generator. In the results, StarGAN V2 integrates domain-specific style codes that enable the creation of style variations within each target domain. But beyond that, it also includes baselines such as MUNIT, DRIT, and MSGAN in terms of much better visual quality (Frechét Inception Distance) while sustaining image diversity (Learned Perceptual Image Patch Similarity), with an FID on CelebA-HQ and on AFHQ of 13.7 and 16.2, respectively, showing significantly high performances.

Another research, by Yannis Assael, et al., entitled "Restoring And Attributing Ancient Texts Using Deep Neural Networks," introduced a deep learning model for three principal tasks in epigraphy: restoring damaged ancient texts, geographic attribution, and chronological attribution of texts [7]. Ithaca was designed to increase accuracy and efficiency in restoring, locating the original place, and dating ancient inscriptions, particularly in ancient Greek. The results were a 26.3% CER, far better than human historians' 59.6% and the previous model, Pythia, at 47.0%. The top-1 accuracy was 61.8%, whereas for Top-20 predictions, it rose to 78.3%. Human historians assisted by Ithaca also resulted in increased accuracy of 71.7% and a CER of 18.3%.

The research gap from this study is that there is no method of reconstructing Javanese script that can automatically repair the damage accurately. This research proposes a novelty in applying a customized StarGAN V2 model to handle various types of script damage in terms of size and shape. The novelty in this research is the construction of a StarGAN V2 model adapted for Javanese script, which has not yet been widely explored in the literature related to ancient manuscript reconstruction. Using this technology, it is expected to play an important role in preserving Indonesia's cultural heritage and to give a chance for further research in the field of image processing and ancient text reconstruction.

II. RESEARCH METHOD

The method used in this research is Generative Adversarial Network, or GAN, with a variation called StarGAN v2, which consists of four main components in the reconstruction process. Figure 1 illustrates the steps taken in this research.

The architecture includes a style encoder, a mapping network, a generator, and a discriminator, working collaboratively to reconstruct damaged manuscripts. The style encoder and mapping network capture style variations, while the generator synthesizes reconstructed images that resemble the ground truth. The discriminator evaluates the generated images to improve the model's performance.

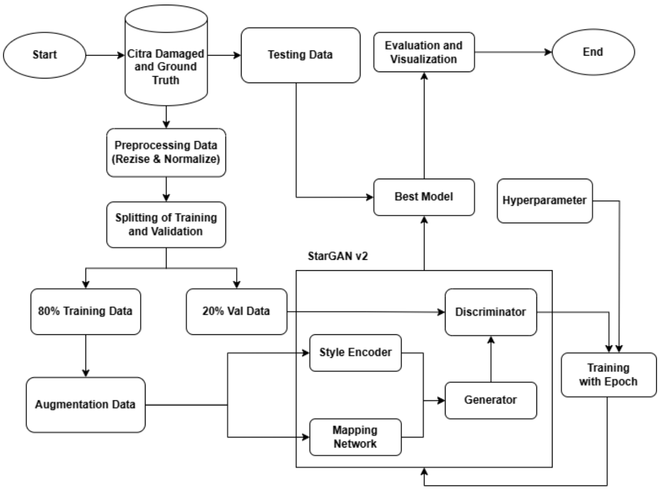

Figure 1. Research Architecture

A. Data Collection

The dataset of this research comes from ancient Javanese manuscripts stored at the MPU Tantular Museum, Sidoarjo, East Java. Those manuscripts include Jajusalatin, Samkok, Serat Ramayana, and Kitab Ramayana, with several types of damage, such as spots on the paper, ink bleeding, holes, and blurred characters (Figure 2).

Figure 2. Photographing the Javanese Script Manuscript

The data was obtained by direct photography of the manuscript pages using a camera. Each manuscript was photographed up to 10 pages, with an initial total of 40 photos. In addition, the museum added 13 pages from related collections to the manuscript, amounting to 53 pages in total. To provide more focused data, each page was cropped into five parts, yielding 265 Javanese script images as shown in Figure 3.

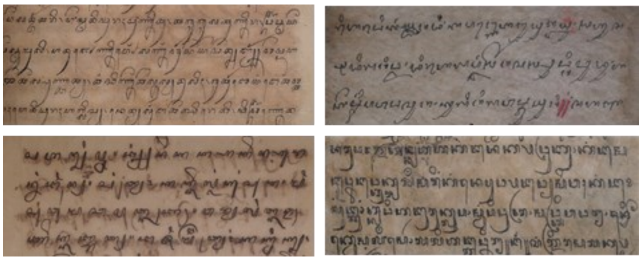

Figure 3. Image of Javanese Manuscript

DOI: 10.34148/teknika.v14i1.1182

B. Preprocessing Data

From the 265 already-prepared Javanese script images, a problem was found in the length-width dimension of the images. All images were resized to 512x256 pixels. Besides, the images are normalized in the range of [0, 1] by dividing the original pixel values which are in the range of [0, 255]. The aim is to accelerate the convergence of the model during training. Then, they are converted into grayscale images and extra channels are added to make sure all images are three-channel formats, such as RGB.

C. Splitting Data Training

The dataset of 265 Javanese script images was split into three parts: training, validation, and testing. In total, 25 images were allocated for testing with non-randomized data selection to ensure that the images represented various levels of damage and script characteristics. The remaining 240 images were divided in a proportion of 80% for training, amounting to 192 images, and 20% for validation, amounting to 48 images.

D. Augmentation Data

In the augmentation of data, the augmentation techniques will be applied to the training data, both for damaged images and ground truth images. The augmentation used will include several techniques: padding, rotation, Gaussian blur, and Gaussian noise. The padding technique is useful in adding margins around the image so that the size and proportion of the image remain consistent despite the rotation [8]. A rotation of 5 degrees is applied to increase the variation in image orientation, so that the model becomes more robust against small shifts in script orientation. To simulate damage which obscures the text, Gaussian blur is used, while Gaussian noise is added to simulate disturbances that may appear in digitized images of old manuscripts, such as spots or artifacts [9].

After the process of augmentation is complete, the augmented results are combined with the original training data to enrich the amount of training data. It aims to improve the generalization ability of the model in recognizing and reconstructing Javanese characters in various conditions of damage and interference so that the model can be more effective in handling data that has never been seen before.

E. StarGAN v2 Models

StarGAN v2 is an extension of StarGAN, a Generative Adversarial Network-based model designed to handle image-to-image translation tasks. StarGAN v2 offers significant improvements over previous versions by introducing the ability to generate images with richer stylistic variations in multiple domains using a single model. Unlike the regular StarGAN, which only translates images between specific domains without considering stylistic diversity, StarGAN v2 allows controlling stylistic variations within the target domain through the integration of style codes [5]. This makes StarGAN v2 more flexible and suitable for various creative applications, such as face manipulation, art style changes, or image reconstruction.

The StarGAN v2 architecture consists of four main components: Style Encoder, Mapping Network, Generator, and Discriminator, which work together to produce images with high quality and diversity [5][10]. Each component plays an important role in ensuring the success of the image-to-image translation process, including generating style variations that match the target domain and maintaining the authenticity of the resulting images.

The used Style Encoder consists of four convolution blocks, each using a convolution layer with a kernel size of 4x4, stride 2, and padding 1, followed by Leaky ReLU activation function and Batch Normalization to maintain stability during training. The last block is completed with an Adaptive Average Pooling layer that reduces the spatial dimension to a fixed size of 1×1, so the output remains consistent even if the input size varies. This layer's output is then flattened and fed to the fully connected layer to produce a 512-dimensional style code. The architecture of the Style Encoder can be seen in Table 1 below.

Table 1. Encoder Style Architecture

Layer	Activation	Additional Info	Output Shape
Conv2D	LReLU	-	(B, 64, H/2, W/2)
Conv2D	LReLU	BatchNorm2D (128)	(B, 128, H/4, W/4)
Conv2D	LReLU	BatchNorm2D (256)	(B, 256, H/8, W/8)
Conv2D	LReLU	BatchNorm2D (512)	(B, 512, H/16, W/16)
Adaptive AvgPool 2D	-	Output size: (1, 1)	(B, 512, 1, 1)
Flatten	-	Flattens the tensor	(B, 512)
Fully Connecte d (FC)	-	Maps to style vector	(B, 512)

Mapping Network: The used network is an MLP consisting of eight fully connected layers. The input, hidden layers, and output for style code have a fixed dimension of 512. All the layers are unshared, and ReLU activation has been applied to all layers except the last one. The input style code is processed to get a more complex representation of the style without using pixel or feature normalization since such approaches do not improve performance and can actually degrade performance. The architecture can be seen in Table 2 below.

Table 2. Architecture Mapping Network

Type	Layer	Activation	Output Shape
Unshared	Linear	ReLU	512
Unshared	Linear	ReLU	512

DOI: 10.34148/teknika.v14i1.1182

Unshared	Linear	ReLU	512
Unshared	Linear	ReLU	512
Unshared	Linear	ReLU	512
Unshared	Linear	ReLU	512
Unshared	Linear	ReLU	512
Unshared	Linear	-	512

The generator starts with the process of encoding the input image using three consecutive convolution layers, each followed by a ReLU activation function. During this stage, the image undergoes downsampling to reduce resolution and extract important features. Then, the image goes through several residual blocks, where each block consists of two convolution layers equipped with AdaIN (Adaptive Instance Normalization) normalization to adapt the feature representation to a given style. These residual blocks go deeper into the network to enhance the model with the capability of reconstructing images with finer details [11]. After that, the image is processed through the decoder, which uses the convolution transposition layer for upsampling to increase the resolution of the image back to its normal size. The architecture of the Generator can be visualized in Table 3 below.

Table 3. Architecture Generator

Type	Layer	Resample	Output Shape
Encoder	Conv2D	-	(64, H, W)
Encoder	ReLU	-	(64, H, W)
Encoder	Conv2D	Down	(128, H/2, W/2)
Encoder	ReLU	-	(128, H/2, W/2)
Encoder	Conv2D	Down	(256, H/4, W/4)
Encoder	ReLU	-	(256, H/4, W/4)
ResBlock	Conv2D	-	(256, H/4, W/4)
ResBlock	AdaIN	-	(256, H/4, W/4)
ResBlock	ReLU	-	(256, H/4, W/4)
ResBlock	Conv2D	-	(256, H/4, W/4)
Decoder	ConvTransp 2D	Up	(128, H/2, W/2)
Decoder	ReLU	-	(128, H/2, W/2)
Decoder	ConvTransp 2D	Up	(64, H, W)
Decoder	ReLU	-	(64, H, W)
Decoder	Conv2D	-	(out_channels, H, W)
Decoder	Tanh	-	(out_channels, H, W)

Discriminator's architecture starts with a convolutional layer with Conv2D - downsampling images with a feature stride of 2 - followed by nonlinear activation through LeakyReLU. Further, batch normalization was performed after the second, third, and fourth convolution layers so that the training is stabilized and the convergence of the model is accelerated. The image then passes through a series of consecutive convolution layers with a kernel size of 4x4; this causes the size of the image to decrease with an increase in the depth of the layer. The final output of the Discriminator is a feature map of depth one that gives a judgment on whether the image is real or fake based on the results of the model evaluation. Architecture of the Discriminator: Architecture of the Discriminator is shown in Table 4 below.

Table 4. Architecture Discriminator

Layer	Resample	Norm	Output Shape
Conv2D (3 → 64)	Stride 2	-	128 x 128 x 64
LReLU	-	-	128 x 128 x 64
Conv2D (64 → 128)	Stride 2	BatchNorm	64 x 64 x 128
LReLU	-	-	64 x 64 x 128
Conv2D (128 → 256)	Stride 2	BatchNorm	32 x 32 x 256
LReLU	-	-	32 x 32 x 256
Conv2D (256 → 512)	Stride 2	BatchNorm	16 x 16 x 512
LReLU	-	-	16 x 16 x 512
Conv2D (512 → 1)	Stride 2	-	8 x 8 x 1

F. Training

Two epoch schemes were conducted in the training stage, namely 50 and 100 epochs. Training was conducted using Google Colab Pro with hardware specifications including 51 GB RAM, a T4 GPU with a capacity of 15 GB, and a 235.7 GB disk. Data used for training was stored in Google Drive and loaded into the training program. While launching the code, the whole training program includes batch processing: corrupted images, ground truth images, followed by further processing using the generator network combined with style encoding and generating new images. In the following order, there is an implementation of a Discriminator on the authenticity of the final obtained image and an evaluation process- loss calculation in each of both models: Discriminator and Generator. The code is elaborated on an epoch-based and batch basis.

In the calculation stage of the loss function, there are two types of loss involved, namely adversarial loss and reconstruction loss. The Adversarial Loss is used to measure the extent to which the generator can produce images that are considered original by the discriminator [4][12], while Reconstruction Loss is used to measure the similarity between the generated image and the ground truth [13]. The loss function works by maximizing the validity of the original image and minimizing the validity of the fake image generated by the generator [14][15]. The reconstruction loss has been weighted in the generator to stress how much more similar the generated image has to be compared with the

DOI: 10.34148/teknika.v14i1.1182

original one. This gives a weight of 50 on the reconstruction loss.

G. Best Model

If it's already in the fine tuning phase, it generates the weight for the trained model by combining weights from both models - the generator and discriminator in a file with the extension of.pth. The essence of storage will have the model at hand with any instance of reuse because of avoidance from retraining challenges. After saving the trained model used from an inference on the testing dataset,. The best model learned during training can be used for evaluation on the test data in order to measure its performance in producing better images according to predefined metrics such as PSNR, SSIM, and LPIPS.

H. Evaluation and Validation

Model evaluation and validation are performed using three main metrics, namely PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index Measure), and LPIPS (Learned Perceptual Image Patch Similarity). These three metrics were chosen to measure the quality of image reconstruction from various aspects, including pixel similarity, visual structure, and human perception.

1) **PSNR** is used to measure the pixel similarity between the reconstructed image and the ground truth. The higher the PSNR value, the smaller the difference between the two images [16]. PSNR focuses on pixel errors, but does not consider the similarity of visual structures. PSNR is defined by the Formula 1:

$$PSNR = 10 \cdot log_{10}\left(\frac{MAX^2}{MSE}\right) \qquad (1)$$

Where :

- **MAX** is the maximum pixel value (e.g. 255 for 8-bit images)
- **MSE** (Mean Squared Error) between the reconstructed image and the ground truth.

2) **SSIM** The SSIM evaluates the structural similarity of the reconstructed image and the ground truth, taking into consideration the aspects of luminance, contrast, and structure. SSIM more accurately reflects human-perceived visual quality compared to PSNR [17]. SSIM is defined by Formula 2:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \qquad (2)$$

Where :

- μ_x and μ_y is the local average.
- σ_x^2 and σ_y^2 is the local variance.
- σ_{xy} is the covariance between *x* and *y*.
- C_1 dan C_2 used to stabilize the calculation when the denominator value approaches zero.

3) **LPIPS** is a perception-based metric that measures visual similarity between images based on representational features in deep learning networks [18]. Unlike PSNR and SSIM, LPIPS compares images on a patch level using the trained network. The smaller the LPIPS is, the more perceptually similar the images are. Formally, LPIPS can be defined as: a feature distance between two images in the representation space as defined in Formula 3:

$$d(x, x_0) = \sum_j \frac{1}{W_j H_j} \sum_{h,w} \left\| \phi^j(x) - \phi^j(x_0) \right\|_2^2 \qquad (3)$$

Where :

- $d(x, x_0)$ Represents the LPIPS score between image *x* (prediction) and x_0 (ground truth).
- $\phi^j(x)$ i.e. The representation of image feature *x* at layer *j* of the backbone network.
- $\| \cdot \|_2^2$ The L2 squared norm, which calculates the difference between features.
- $W_j H_j$ is the feature dimension at layer *j*

With $\phi^j(\cdot)$ defined as in Formula 4:

$$\phi^j(x) = w_j \odot o_{hw}^j(x) \qquad (4)$$

Where :

- $o_{hw}^j(x)$ is the raw feature output at spatial coordinates h,w in layer *j*.
- $\odot$ is an element-wise operation (usually multiplication).
- w_j which is the weight applied to the feature for human perception adjustment.

III. RESULTS AND DISCUSSION

A. Model Training Results

The StarGAN v2 model managed to produce the reconstructed images of damaged ancient Javanese manuscripts after the 50- and 100-epoch training process. Figure 4 and Figure 5 show the loss graphs of the generator and discriminator at the 50th and 100th epochs.

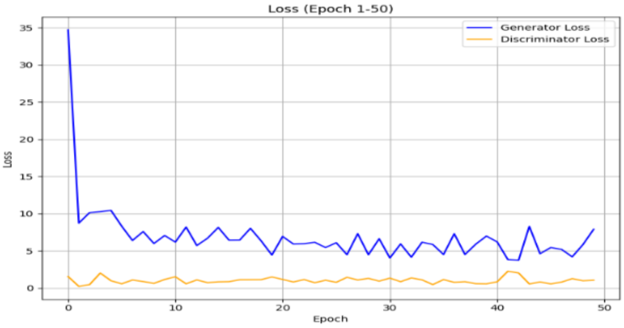

Figure 4. Graph of Generator and Discriminator Loss at Epoch 50

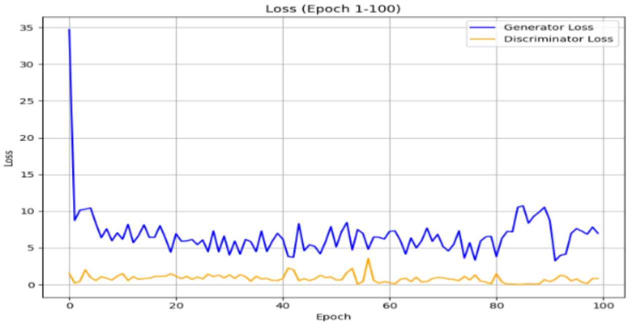

Figure 5. Graph of Generator and Discriminator Loss at Epoch 100

Until the 50th epoch, it gives D loss as 1.0382 while the G loss is reading as 7.8943. From this low D loss, it is very evident that the discriminator has learned very well to tell between a reconstructed image and a ground truth. On the other hand, considering its high G loss values, it still seems rather challenging for the generator to be good at generating images that should more or less look like their corresponding targets.

After training is extended to the 100th epoch, it is recorded that the D Loss value is at 0.8166, showing an improvement whereby the discriminator can better discriminate on images. Meanwhile, for G Loss, it stood at 6.9848, which has dropped significantly since the beginning of training. The above depicts a healthy competition between the generator and the discriminator; however, the generator will need more optimization to generate much more realistic images.

B. Reconstruction Results

Figure 6 and Figure 7 show the image reconstruction results at the 50th and 100th epochs, respectively. These images show the model's ability to reconstruct damaged text into an image that is close to the ground truth.

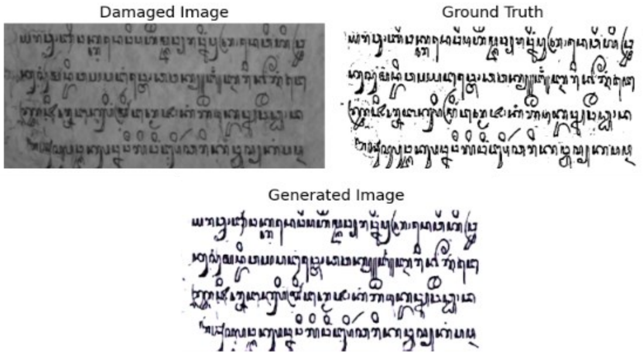

Figure 6. Model Reconstruction Results at the 50th Epoch.

At the 50th epoch, the model was already capable of reconstructing most of the text structure with fairly good accuracy, although there was some minor noise at certain characters. The resulted image shows that the model had been able to comprehend the pattern and structure of Old Javanese, including in areas that usually received significant distortion.

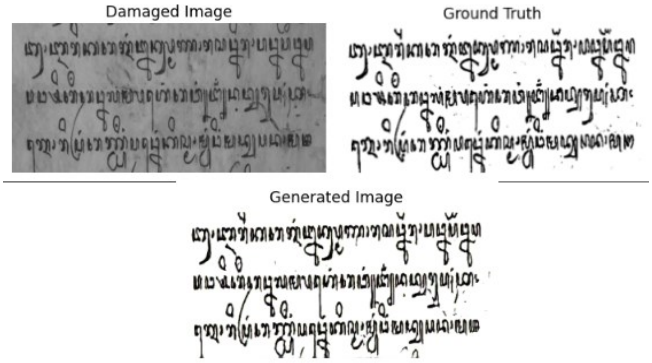

Figure 7. Model Reconstruction Results at the 100th Epoch

After training up to the 100th epoch, the reconstruction results significantly improved. The model manages to produce clearer and closer images to the ground truth with less noise. That means longer training has a positive effect on the quality of reconstruction.

C. Image Quality Matrix Evaluation

The PSNR, SSIM, and LPIPS metrics are calculated to assess the quality of the reconstructed images. Each metric's evaluation result from the average over the testing data is included in Table 5, using two schemes.

Table 5. Image Quality Evaluation Results

Epoch	PNSR	SSIM	LPIPS
50	16.1234	0.8374	0.1020
100	**17.5588**	**0.8519**	**0.1051**

Results of evaluation as shown in Table 5 indicate an increase for all metrics of evaluation from the 50th to the 100th epoch. At the 50th epoch, the PSNR value was 16.1234 dB, which means the quality of the image reconstructed was still relatively poor. However, the PSNR value increases to 17.5588 dB after continuing the training until the 100th epoch, which indicates that the resulting image is closer to the original image with low noise.

In addition, the SSIM at the 50th epoch is 0.8374, which can be argued as a very good structural similarity between the reconstructed image and ground truth. The increased value of 0.8519 in the 100th epoch hence shows that the model is getting better at maintaining the structure and details of the original image. While the LPIPS at the 50th epoch was 0.1020, which already had good perceptual similarity, at the 100th epoch it slightly improved to 0.1051. Although a small improvement, it is apparent that the model can further generate perceptually better images. Overall, evaluation results illustrate that the StarGAN v2 model improves the quality of the reconstructed images as more epochs are used. While this represents a considerable improvement, further enhancements can be made to decrease the LPIPS value even further and increase the PSNR. Model optimization and

DOI: 10.34148/teknika.v14i1.1182

exploring data augmentation techniques would possibly make for better results in future research.

IV. CONCLUSION

This paper successfully applied the StarGAN v2 model to reconstruct damaged ancient Javanese manuscripts. The result of training the model shows that it is able to generate high-quality reconstructed images with significant improvements in the evaluation metrics.

In the training process, two types of loss functions are used: adversarial loss and reconstruction loss. In this case, adversarial loss will indicate that a generator is getting better at generating images considered genuine by a discriminator, while reconstruction losses indicate how well the generated images are close to the ground truth. The adversarial loss and reconstruction loss values, at the end of training, show a consistent decrease, hence showing that the model has learned well to reconstruct missing or corrupted images.

Regarding the evaluation metrics, the PSNR value increased from 16.1234 dB at the 50th epoch to 17.5588 dB at the 100th epoch, reflecting a significant improvement in the quality of the images. The SSIM increased from 0.8374 to 0.8519, showing the structural similarity between the generated image and the ground truth. However, the LPIPS increased from 0.1020 to 0.1051, which indicated that though there is an improvement, as far as perceptivity is concerned, further development must be made.

In general, this study has shown the effectiveness of the StarGAN v2 model in reconstructing ancient Javanese manuscripts, with promising results both in image quality and evaluation metrics. Further research is recommended to explore more diverse data augmentation techniques and model optimization for better reconstruction results.

REFERENCES

[1] T. Asrianti and P. Y. Fauziah, “Pendampingan Belajar Aksara Jawa dalam Upaya Pelestarian Budaya Jawa,” *Abdi J. Pengabdi. dan Pemberdaya. Masy.*, 2023, doi: 10.24036/abdi.v5i3.472.

[2] N. Nofrizal, “Pelestarian Manuskrip Kuno Melayu Nusantara Perspektif Industries,” *Al-Adyan J. Stud. Lintas Agama*, 2020, doi: 10.24042/ajsla.v15i2.6110.

[3] A. R. Himamunanto, “Restorasi Digital Pada Model Kerusakan Citra Aksara Jawa Cetak,” *J. Teknol. Informasi-Aiti* |, 2016.

[4] R. Raut, A. Devkar, P. S. Borkar, R. Deoghare, and S. Kolambe, “Generative Adversial Network Approach for Cartoonifying image using CartoonGAN,” *J. Electr. Syst.*, 2023, doi: 10.52783/jes.666.

[5] Y. Choi, Y. Uh, J. Yoo, and J. W. Ha, “StarGAN v2: Diverse Image Synthesis for Multiple Domains,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020. doi: 10.1109/CVPR42600.2020.00821.

[6] K. Holmes, P. Sharma, and S. Fernandes, “Facial skin disease prediction using StarGAN v2 and transfer learning,” *Intell. Decis. Technol.*, 2023, doi: 10.3233/IDT-228046.

[7] Y. Assael *et al.*, “Restoring and attributing ancient texts using deep neural networks,” *Nature*, 2022, doi: 10.1038/s41586-022-04448-z.

[8] A. Mumuni and F. Mumuni, “Data augmentation: A comprehensive survey of modern approaches,” *Array*. 2022. doi: 10.1016/j.array.2022.100258.

[9] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, 2019, doi: 10.1186/s40537-019-0197-0.

[10] B. Han and M. Hu, “The Facial Expression Data Enhancement Method Induced by Improved StarGAN V2,” *Symmetry (Basel).*, 2023, doi: 10.3390/sym15040956.

[11] S. Park and Y. G. Shin, “Generative residual block for image generation,” *Appl. Intell.*, 2022, doi: 10.1007/s10489-021-02858-6.

[12] L. Xinwei, G. Jinlin, D. Jinshen, and L. Songyang, “Generating Constrained Multi-target Scene Images Using Conditional SinGAN,” in *2021 IEEE 6th International Conference on Intelligent Computing and Signal Processing, ICSP 2021*, 2021. doi: 10.1109/ICSP51882.2021.9408686.

[13] Y. Li, N. Xiao, and W. Ouyang, “Improved generative adversarial networks with reconstruction loss,” *Neurocomputing*, 2019, doi: 10.1016/j.neucom.2018.10.014.

[14] K. Janocha and W. M. Czarnecki, “On loss functions for deep neural networks in classification,” *Schedae Informaticae*, 2016, doi: 10.4467/20838476SI.16.004.6185.

[15] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, “Loss Functions for Image Restoration With Neural Networks,” *IEEE Trans. Comput. Imaging*, 2017, doi: 10.1109/TCI.2016.2644865.

[16] National Instruments Australia, “Peak Signal-to-Noise ratio as an image quality metric,” *Natl. Instruments*, 2013.

[17] H. B. Sumarna, E. Utami, and A. D. Hartanto, “Tinjauan Literatur Sistematik tentang Structural Similarity Index Measure untuk Deteksi Anomali Gambar,” *Creat. Inf. Technol. J.*, 2021, doi: 10.24076/citec.2020v7i2.248.

[18] H. Park and S. Park, “Improving Monocular Depth Estimation with Learned Perceptual Image Patch Similarity-Based Image Reconstruction and Left–Right Difference Image Constraints,” *Electron.*, 2023, doi: 10.3390/electronics12173730.