IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 11, No. 12, December, 2024, pp. 3447-3457
Journal homepage: http://iieta.org/journals/mmep

Restoration of Javanese Characters Based on Wasserstein Generative Adversarial Network-Gradient Penalty

Fitri Damayanti[1*], Ika Oktavia Suzanti[2], Achmad Jauhari[2], Sri Herawati[1]

[1] Department of Information System, Faculty of Engineering, University of Trunojoyo Madura, Bangkalan 69162, Indonesia
[2] Department of Informatics, Faculty of Engineering, University of Trunojoyo Madura, Bangkalan 69162, Indonesia

Corresponding Author Email: fitrid@trunojoyo.ac.id

Copyright: ©2024 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/mmep.111223

Received: 23 August 2024
Revised: 15 October 2024
Accepted: 22 October 2024
Available online: 31 December 2024

Keywords:
Javanese characters, Multi-Style Ancient Javanese Characters Dataset, restoration, Structural Similarity Index Measure, Wasserstein Generative Adversarial Network-Gradient Penalty

ABSTRACT

Ancient manuscripts are a medium for understanding the history and life of past societies. Over a long storage period, the ancient manuscript is at risk of physical damage, which can result in the loss of the information contained within it. Therefore, it is deemed important to repair the damaged letters or characters in the ancient manuscript to restore its original strokes and texture. To address this issue, this research proposes a restoration method based on Generative Adversarial Network (GAN), specifically the Wasserstein Generative Adversarial Network-Gradient Penalty method (WGAN-GP). This method is capable of addressing the issues of gradient vanishing and mode collapse that occur in traditional GANs. The Gradient Penalty function in this method is used to control the gradients produced during the training process, thereby helping to maintain the consistency of stroke shapes and font styles. The evaluation of this method was conducted qualitatively and quantitatively on the Multi-Style Ancient Javanese Characters Dataset (MSAJCD). This dataset consists of printed Javanese script and handwritten Javanese script. The WGAN-GP method is capable of restoring Javanese script with extensive damaged areas and can ensure the correct stroke shapes of the repaired script. The results of the experiments show that the WGAN-GP method is superior to the Deep Convolution Generative Adversarial Network (DCGAN) method and the Text Block Identification method. With a Structural Similarity Index Measure (SSIM) index increase of 7.07% for the printed Javanese script dataset, and 10.48% for the handwritten Javanese script dataset.

1. INTRODUCTION

Precious artifacts containing the history of different eras are old texts that document human culture. Age and deterioration are the two main issues facing the preservation of old manuscripts nowadays. Therefore, we must immediately improve the preservation of these texts. The use of computers to store digital photographs of old manuscripts has emerged as a new conservation strategy with the advent of digitization. However, many letters undergo destruction during transportation. As a result, fixing broken letters is a crucial procedure. We must restore the damaged letters before we can study and share these old manuscripts.

Today, professionals primarily restore ancient manuscripts manually. They work by completing their reasoning based on the various information provided by the residual strokes of the damaged letters and other letters around them. They then use image processing tools to perform manual restoration. This approach is inefficient and time-consuming, hence the need for an automated restoration process.

Several researchers have attempted to restore letters using current image restoration techniques, with the aim of enhancing the effectiveness of damaged letter repair procedures. Weng et al. [1] recovered Chinese characters using the Conditional Generative Adversarial Network (CGAN) technique. Jo et al. [2] repaired faulty Chinese letters in 2021 using the Variational Autoencoder with Classification (VAE-C) model. Song et al. [3] attempted to repair damaged Chinese characters in the same year by employing a strategy based on adversarial classification loss and attention. Su et al. [4] achieved the authentic restoration of ancient Yi alphabets by constructing a multi-stage restoration network that combines form and texture restoration networks. Sun et al. [5] studied the RubbingGAN method's efficacy in Chinese character restoration in 2022. A rubbing GAN consists of one generator, G, and two discriminators, D1 and D2. The generator uses the U-Net architecture. Discriminator 1 uses the PatchGAN architecture, while Discriminator 2 uses the Auto Encoder Decoder architecture. These studies present a number of challenges. The first issue is that relying solely on edge information, which already exists in a wide, damaged area containing significant information, makes effective restoration challenging to accomplish. The second issue is that repairing broken letters requires not only maintaining visual consistency, but also restoring the proper topology of the letter strokes.

Researchers have exclusively used Chinese characters as

their dataset. Chinese characters are a group of hieroglyphics. Characters, not letters, comprise the Chinese script. Unlike an alphabet, these characters represent words and concepts. In Indonesia, there are several tribes, and each tribe has its own letter or character. The traditional Indonesian characters, originally known as Hanacaraka, Carakan, and Dentawyanjana, originated on the island of Java. The Javanese characters are descendants of Brahmi. Javanese characters, an Abugida writing system, consist of 20 to 33 basic letters, depending on the language being written [6]. Himamunanto and Setyowati conducted research on Javanese character restoration in 2018 using the Text Block Identification method [7]. The research used datasets taken from ancient Javanese document sheets entitled "Hamong Tani." We used 452 damaged basic Javanese characters as our datasets. Figure 1 shows some examples of datasets with a small area of damage. The resulting accuracy is 82.07%. The research was only able to perform restoration when the damage area was small.



Figure 1. Some examples of Javanese script corruption used in the research [7]

Some of the previous research resulted in poor accuracy for restoration when the damaged letter area is large. In reality, a significant number of letters in ancient manuscripts sustain damage across a substantial area. Furthermore, research [7] yields restoration results that closely resemble damaged Javanese characters, albeit with structural errors. So, to get more accurate results when fixing large areas of damage in Javanese script and make sure the results were structurally correct, we used the Wasserstein Generative Adversarial Network-Gradient Penalty (WGAN-GP) method in our research.

One way to get around these problems is to use a Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) to fix the characters in Javanese. WGAN-GP is better than regular GANs because it fixes problems like gradient vanishing and mode collapse, makes training more stable, and boosts the accuracy of restoration. By incorporating gradient penalties, this method can better preserve the fine details and structural integrity of Javanese characters, even in cases of severe damage.

In summary, while previous methods have made strides in digital restoration, they often fall short in handling large areas of damage and ensuring structural correctness, especially for character sets like Javanese scripts. This study aims to fill this gap by leveraging WGAN-GP, offering a robust solution for the restoration of ancient Javanese manuscripts with large damaged areas.

GANs essentially consist of two components: a generator and a discriminator. The discriminator distinguishes between the created image and the training image, while the generator generates high-quality fake images to prevent the discriminator from accurately determining whether an image is real or fake. Following training, the GAN can archive the reconstructed images using a generator and extract features from the training set. As one of the unsupervised learning techniques, GAN does not require knowledge of the training data set beforehand. It functions by implicitly summarizing the training data set's features through discriminator and generator confrontations, and it then uses the generator to provide feedback on the summarized features for the reconstruction [8]. Several variants of GAN exist, such as Deep Convolutional GAN (DCGAN) [9], Wassertein GAN (WGAN) [10], and Wassertein GAN with Gradient Penalty (WGAN-GP) [11].

The generative adversarial network (GAN) variant WGAN-GP trains generative models. Traditional GANs encounter several issues, including gradient vanishing and mode collapse problems. Gradient vanishing occurs when learning stops progressing because the discriminator is too strong, and mode collapse occurs when the generator only produces very limited variations of the data. WGAN-GP addresses these issues by using the Wasserstein approximation (or Wasserstein distance) to measure how similar the distribution of data generated by the generator is to the distribution of the training data. WGAN-GP uses a loss function that focuses on reducing the Wasserstein distance between these two distributions. The "Gradient Penalty" in WGAN-GP is a significant innovation that restricts the gradient values during training. By applying a penalty to the gradient of the discriminator model, WGAN-GP ensures that it remains stable during training, which helps avoid instability problems that often occur with traditional GANs [12].

We used the Wasserstein Generative Adversarial Network-Gradient Penalty (WGAN-GP) method to improve the accuracy of Javanese character restoration with a large area of damage and get results that were structurally correct. The main contributions in this research are:

- The primary contributions of this research include the proposal of a WGAN-GP-based restoration technique, which is capable of restoring Javanese characters with extensive damage to their texture and shape.
- We aim to enhance the restoration accuracy of Javanese characters, whether they are handwritten or printed.

Section 1 of the introduction, which provides background information on the research-related studies, is where the methodical writing process for this piece starts. Section 2 describes related work. Section 3 provides an explanation of the study methodology for restoring damaged Javanese characters. Section 4 explains the experiment and discussion. Section 5 includes conclusions and suggestions for further research.

2. RELATED WORK

Since the 1960s, a number of techniques have been developed for picture inpainting, including the texture-based texture synthesis inpainting method and the nontexture-based image restoration model. In the fields of computer vision and deep learning, inpainting technology has emerged as a captivating research topic [13, 14]. Various real-world situations can apply the findings of image restoration research, such as restoring outdated photographs, fixing partial images, changing the style of images, and enhancing image clarity. Pathak et al. [15] first suggested the GAN-based structural image inpainting technique in 2016. This technique maps an image with missing portions to a low-dimensional feature space using an encoder and then reconstructs the output image using a decoder. The study's inadequacy lies in its inability to maintain local consistency between the residual region and the recently recovered zone. By employing global and local discriminators to assess opponent loss during structural

restoration, Iizuka et al. [16] were able to strengthen the weakness. While the local discriminator verifies the image's local consistency, the global discriminator assesses the recovered image's overall semantic consistency.

Yang et al. [17] used multilayer neural networks to mimic texture consistency in high-resolution photos. Yeh et al. [18] employed the Deep Convolution Generative Adversarial Network (DCGAN) to retrieve semantic images in the same year. Yu et al. [19] conducted research on picture inpainting with a two-step network (gated convolution) in 2019. Li et al. [20] created a recurrent feature reasoning network (RFR-net) in 2020 to carry out large-scale restoration of missing regions.

The GAN method has advantages over traditional diffusion-based [21-25] and patch-based [26, 27] image inpainting methods. Traditional diffusion-based image inpainting methods work by smoothly expanding and filling damaged regions with edge information. This method can recover images with regular texture changes, but it cannot recover sections with semantic structure. Patch-based techniques replicate comparable data from the image to fill up the damaged area. This technique expedites the search for matched patches by utilizing the texture description. This technique can restore photos with repeating patterns, but it cannot restore photos with distinct structures in damaged areas.

The problems that come with regular GANs can be fixed with GAN-based methods, especially the Wasserstein GAN with Gradient Penalty (WGAN-GP). These methods can fix problems like gradient vanishing and mode collapse. By using the Wasserstein distance to measure the similarity between the real and generated distributions, WGAN-GP ensures more stable training and better performance in scenarios involving large areas of damage. Furthermore, the gradient penalty helps maintain consistent gradients during training, allowing the generator to produce more accurate restorations while preserving the structural details of Javanese characters.

The primary advantage of WGAN-GP over previous GAN-based models is its ability to handle large, damaged areas by focusing on minimizing the Wasserstein distance, which offers a more robust measure of how well the generated data matches the real data. This approach allows the model to preserve both global and local structure, making it particularly suitable for Javanese character restoration, where stroke accuracy and font consistency are critical.

While previous studies have made significant progress in character restoration and image inpainting, many face limitations when dealing with large-scale damage and complex character structures. This research builds on the strengths of these methods while addressing their shortcomings through the use of WGAN-GP, offering a more effective solution for restoring severely damaged Javanese characters.

3. METHOD

In order to address the issue of restoring damaged Javanese characters, this research introduces a GAN-based restoration model known as WGAN-GP (Wasserstein Generative Adversarial Network-Gradient Penalty). WGAN-GP is a type of Generative Adversarial Network (GAN) that was created to fix some of the problems with training and stability that come up with regular GANs. There are two main parts to WGAN-GP: Wasserstein distance, which is used to measure errors, and gradient projection, which is used to keep training regular [11].

The main goal of a GAN is to increase the produced data's similarity to the genuine data by bringing its distribution closer to the true data. Conventional GANs use a metric known as Jensen-Shannon divergence (JS divergence), whose calculation is based on Kullback-Leibler divergence (KL divergence), as illustrated in Eq. (1), to compare the true data distribution with the produced data distribution. The asymmetric distance between the probability distributions $P_r(x)$ and $P_g(x)$, where $P_g(x)$ denotes the generated data distribution and $P_r(x)$ represents the genuine data distribution, is computed using KL divergence. Eq. (2) [12] illustrates how the asymmetric KL divergence is resolved by the JS divergence. If the true data distribution and the generated data distribution do not overlap, or if the overlap is too tiny, then this measuring metric is flawed. When the GAN updates the weights, this will result in the gradient being 0 or very small in value, which means the weights cannot be updated. When the discriminator is significantly stronger than the generator, this issue arises.

To solve the problem, WGAN proposed the Wasserstein distance, also known as the Earth-Mover (EM) distance, which is shown in Eq. (3). In Eq. (3), P_r is the true data distribution and P_g is the resulting data distribution. $\Pi(P_r, P_g)$ is the set of all possible joint distributions that combine the distributions of P_r and P_g. $E_{(x,y)\sim\gamma}[\|x-y\|]$ is the expected distance value for each joint distribution γ. Wassertein distance is the minimum value among the expected values of all joint distributions. The Wassertein distance is achieved by removing the log of the loss function of the generator and the discriminator. When the two data distributions do not overlap, Wassertein distance has a benefit over JS divergence in that it can still capture the proximity of the data distributions $P_r(x)$ and $P_g(x)$. Maximizing the EM distance between the generated data and the genuine data is the aim of the WGAN discriminator. On the other hand, the generator's objective is to reduce the EM distance between the created and real data [28].

In addition, WGAN also uses the weight clipping method to overcome the weight update gradient when the weights are updated. It is used to constrain the discriminator weights to conform to the 1-Lipschitz continuity shown in Eq. (4). 1-Lipschitz continuity is used to measure the extent to which a function maintains the difference between its inputs. Formally, a function $f: X \to Y$ is said to be Lipschitz with Lipschitz constant $K = 1$, if for any two points x_1 and $x_2$in the domain X, and any value in the output space Y, the difference between the results $f(x_1)$ and $f(x_2)$ is not greater than K times the difference between x_1 and x_2.

Problems with gradient expansion or gradient loss could arise from the weight clipping procedure. Consequently, in order to address the issue brought about by weight clipping, WGAN-GP employs the Gradient Penalty technique [29]. The original WGAN's loss function, L_{WGAN} is displayed in Eq. (5). Eq. (6) shows the weight penalty term of WGAN-GP. Eq. (7) shows how WGAN-GP is improved by including a new penalty term (GP) in the original WGAN loss function. Gradient penalty makes the training of the network model more stable and easier to converge.

Where, $\tilde{x}$ is the interpolation point between the real data and the generated data. λ is a hyperparameter that controls how much gradient penalty will be applied (usually 10 to 1000). While $P_{\tilde{x}}$ is the joint distribution generated through interpolation.

The architecture of WGAN-GP in this research is shown in Figure 2. The input of the generator $G(z)$ is a corrupted Javanese characters image. The data generated by the generator (generated samples) is given to the discriminator. Discriminator $D(x)$ receives input in the form of real and generated samples. Discriminator loss is obtained by adding the Wassertein distance between the real data distribution and the generated data distribution with a gradient penalty. Generator loss is the negative value of discriminator loss. This process is done repeatedly until convergence conditions. The algorithm for the training process for restoring damaged Javanese characters using the WGAN-GP method is shown in Algorithm 1.

$$KL(P_r||P_g) = \int \log\left(\frac{P_r(x)}{P_g(x)}\right) P_r(x)du(x) \quad (1)$$

$$JS(P_r||P_g) = \frac{1}{2} KL(P_r||\frac{P_r + P_g}{2}) + \frac{1}{2} KL(P_g||\frac{P_r + P_g}{2}) \quad (2)$$

$$W(P_r, P_g) = inf_{\gamma \sim \Pi(P_r,P_g)} E_{(x,y)\sim\gamma}[\|x - y\|] \quad (3)$$

$$|f(x_1) - f(x_2)| \leq K\,|(x_1) - (x_2)| \quad (4)$$

$$L_{WGAN} = E_{\tilde{x}\sim g}\,[D(\tilde{x})] - E_{\tilde{x}\sim P_r}\,[D(x)] \quad (5)$$

$$GP = \lambda\, E_{\tilde{x}\sim P_{\tilde{x}}}\,[\,\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1]^2 \quad (6)$$

$$L_{WGAN-GP} = E_{\tilde{x}\sim P_g}\,[D(\tilde{x})] - E_{\tilde{x}\sim P_r}\,[D(x)] + \lambda\, E_{\tilde{x}\sim P_{\tilde{x}}}\,[\,\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1]^2 \quad (7)$$

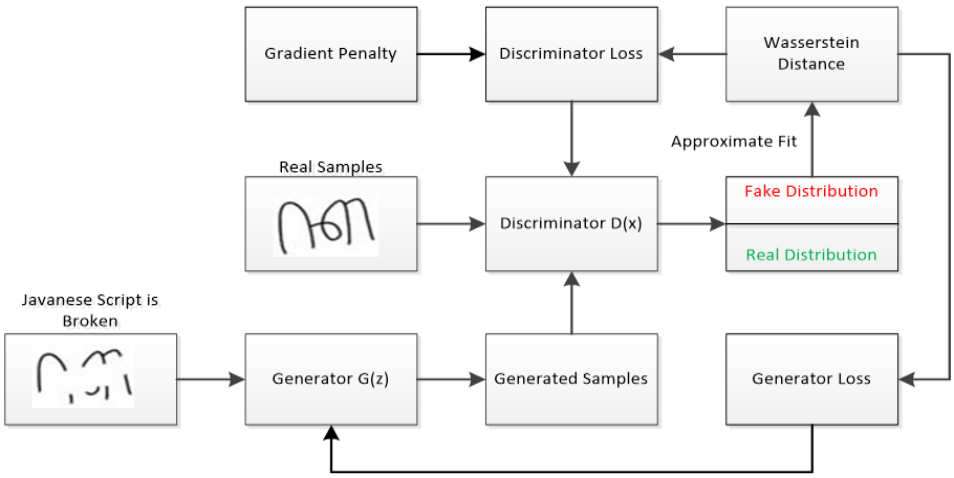


Figure 2. The architecture of the method

Algorithm 1: Training process

***Input*:** Image of original Javanese characters ($x \sim P_r$), The image of Javanese characters is damaged ($z \sim P_z$), $\lambda = 10$, $n_{critic} = 5$, $\alpha = 0.0001, \beta_1 = 0.5, \beta_2 = 0.999$, the batch size ($m$) = 64, $\in \sim \cup\,[0,1]$

***Output*:** The image of Javanese characters is restored ($x \sim P_g$)

***Initialization*:** discriminator parameters w_0, generator parameters θ_0

***Repeat*:**

for $t = 0, \ldots, n_{critic}$ ***do:***

for $i = 0, \ldots, m$ ***do:***

$\tilde{x} = G_\theta(z)$

$\tilde{x} = \varepsilon x + (1 - \varepsilon)\,\tilde{x}$

$L^{(i)} = D_w(\tilde{x}) - D_w(x) + \lambda\,(\|\nabla_{\tilde{x}} D_w(\tilde{x})\|_2 - 1)^2$ using Eq. (7)

end for

$w = Adam\,(\nabla_{\tilde{x}} \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, w, \alpha, \beta_1, \beta_2)$

end for

Sample a batch of damaged Javanese characters image $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$

$\theta = Adam\,(\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} - D_w(G_\theta(z)), \theta, \alpha, \beta_1, \beta_2)$

Until convergence

The model was trained using the Adam optimizer with parameters β_1=0.5, β_2=0.999, and a learning rate of 0.0001. The batch size was set to 64, and the image size was normalized to 64×64 pixels. The hyperparameter λ for the gradient penalty was set to 10, and the number of iterations (epochs) was 500. We updated the discriminator five times (n_{critic} = 5) for each generator update, ensuring it was well-optimized before updating the generator.

The generator architecture consists of 4 convolution layers. We use a 2-dimensional convolution with a kernel size of stride 2 for each convolution layer. The selection of stride 2 aims to perform downsampling without using spatially deterministic functions, such as maxpooling. Then carry out the batch normalization (BN) process and the ReLU activation function. The generator output employs the tanh function as its activation function.

The Discriminator architecture used is identical to the encoder form. The difference lies in the last layer used as output. In the encoder, the output is a variable z which is the result of encoding the input image in the latent domain z while

in the discriminator, the output is the result of classifying the class of the input, namely the original image class or the synthesized image. The discriminator architecture consists of 4 convolution layers. Each convolution layer used is a 2-dimensional convolution with a kernel size of 4×4, stride 2. Then carry out the Batch-Normalization (BN) process with the LeakyReLU activation function.

4. EXPERIMENT AND DISCUSSION

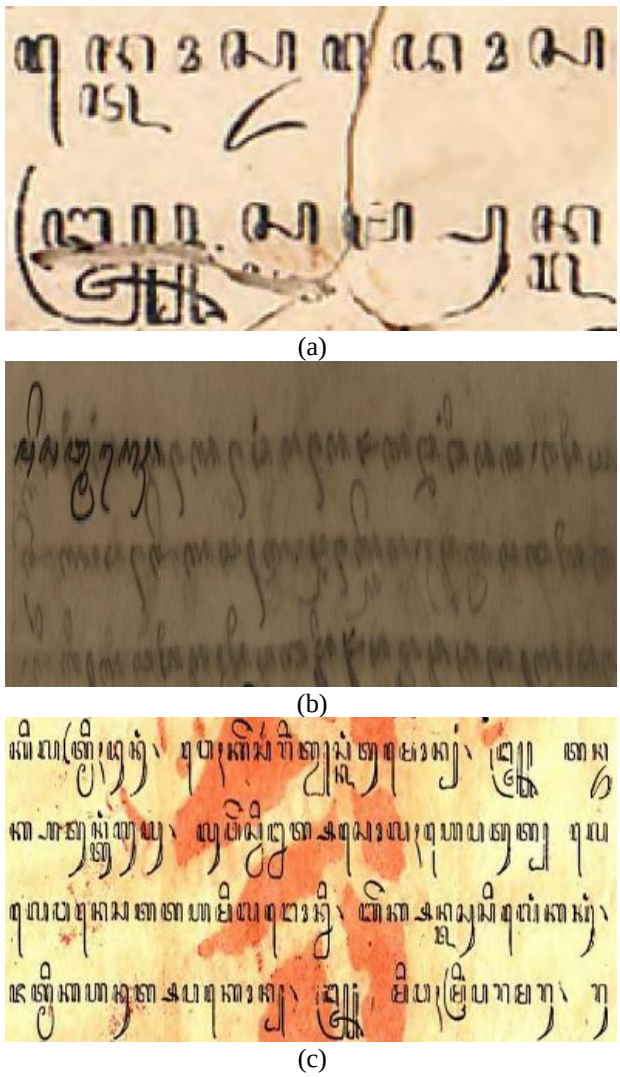
4.1 Dataset and experimental environment

The public dataset does not contain the Javanese character corruption dataset. Therefore, this research utilizes the self-built Multi-Style Ancient Javanese Character Dataset. The dataset contains ancient manuscripts taken from the National Digital Library of Indonesia (https://khastara.perpusnas.go.id). Damage to the ancient manuscripts includes perforated paper, ink bleeding through from previous pages, and red or brown spots. Figure 3(a) illustrates the cutting of ancient manuscripts with perforated paper conditions, while Figure 3(b) displays the ink bleeding through from previous pages, and Figure 3(c) displays the ancient manuscripts with red or brown spots. We segment the images based on foreground and background characteristics. We segment the foreground results to obtain Japanese characters.

To expand the types of Javanese characters used as datasets, this study also took data from a website that contains several styles of Javanese letters (https://aksaradinusantara.com). The Javanese font styles used in this study consist of ABWulang, Carakan, Damarwulan, Djoharuddin, Jamawi, Nyk, and Sehulbari. ABWulang is a Javanese character's font with the handwriting style of the Yogyakarta region adapted from manuscripts and letters of the Yogyakarta Palace around the 1800s, which can be seen in Figure 4(a). Carakan is a Javanese script font that has used the Unicode Standard for the purposes of writing Javanese / Carakan Java, shown in Figure 4(b). The Damarwulan font shown in Figure 4(c) is inspired by the handwriting on the Damarulan manuscript. Djoharuddin is a Javanese character's font with the handwriting style of the Cirebon region adapted from letters from the Cirebon Kraton (Sultan sepuh Djoharuddin) to Thomas Stamford Raffles in the 19th century. Figure 4(d) is an example of Djoharuddin's Javanese characters font. The Jamawi font is based on the manuscript of Serat Imam Nawawi (1852 AD), which can be seen in Figure 4(e). The Sehulbari font based on the Suluk Bonang manuscript is shown in Figure 4(g). Meanwhile, Figure 4(f) is an example of the Nyk Javanese font style.

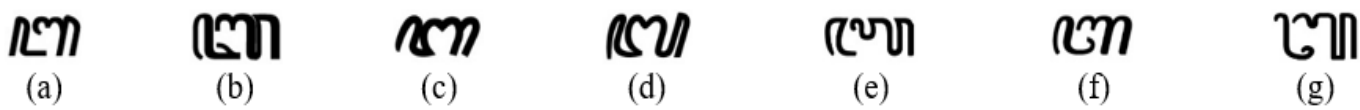
The data obtained from the National Digital Library of Indonesia and the website https://aksaradinusantara.com, is Javanese characters in the form of printed letters. The Javanese characters used in this research is Nglegana characters. Nglegana characters (basic characters), which is a character that functions to connect consonant closed syllables with the next syllable, except for syllables closed by wgyan, layar, and cecak. The 20 basic Javanese characters or Nglegana characters are shown in Figure 5. They consist of Ha, Na, Ca, Ra, Ka, Da, Ta, Sa, Wa, La, Pa, Dha, Ja, Ya, Nya, Ma, Ga, Ba, Tha, Nga.

This research also uses Javanese characters dataset in the form of handwriting taken from https://www.kaggle.com/datasets/hannanhunafa/javanese-script-aksara-jawa-augmented. The Javanese characters are written by several people. The handwritten Javanese characters are in several positions, namely upright position, left tilt, and right tilt. Some examples of handwritten Javanese characters dataset are shown in Figure 6. Similar to the printed Javanese character dataset, this handwritten Javanese characters also uses Nglegana characters (basic characters) which consists of 20 letters.



(a) (b) (c)

Figure 3. Some of the ancient manuscripts used in the research



(a) (b) (c) (d) (e) (f) (g)

Figure 4. Several examples of Javanese letter styles used in research

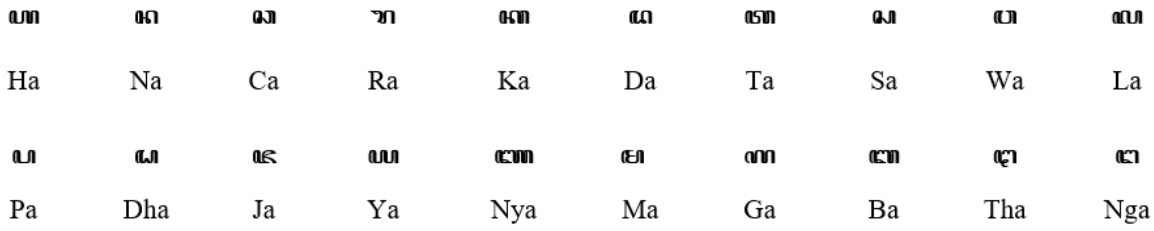


Figure 5. Nglegana Javanese characters

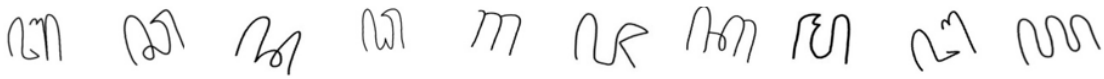


Figure 6. Some examples of handwritten Javanese character datasets



Figure 7. Several examples of damaged Javanese characters in five areas

Table 1. Distribution dataset for experimental scenarios

Dataset	Nglegana Script	Area of Damage	Shape Variation	Total Data
Printed Letters	20	5	10	1000
Handwriting	20	5	50	5000

Model training in this research requires a lot of paired data of original Javanese characters (good) with damaged Javanese characters, which is difficult to obtain in reality. So we make damage in five areas, namely: top left, top right, bottom left, bottom right, and center. Figure 7 shows some examples of corrupted Javanese characters in five areas. In Figure 8, the first row is an example of a corrupted Javanese characters dataset in the form of printed letters, while the second row is an example of a corrupted Javanese characters dataset in the form of handwriting. These images appear to have a fairly large area of damage. Table 1 shows the amount of data in each dataset used in this research.

The experiments were all implemented using Python scripts running on Google Colab Professional. Training procedures were conducted in the Windows system with NVIDIA Intel Core i5 laptop. This research was trained using the Adam optimizer with a batch size of 64. The two parameters of the Adam optimizer are 0.5 and 0.999. Learning rate during training is 0.0001. Hyperparamater λ is 10. The image size is normalized to 64×64 pixels. Using epoch 500 and the ratio between training data and test data is 80% and 20%.

4.2 Measure indicators

The primary metric for evaluating the quality of the restoration outcome is the dissimilarity in structure between the original and restored Javanese character images. While computers struggle to distinguish between two images, the human brain does so with ease. Methods other than distance computation can achieve image similarity measurement. Nowadays, the most widely utilized size indicators are Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM).

We use the Structural Similarity Index Measure (SSIM) [30, 31] to calculate the similarity between two images. The SSIM value is in the range of -1 to 1, with 1 indicating perfect similarity and -1 indicating total dissimilarity. The higher the SSIM value, the higher the similarity of the two images. SSIM can be calculated using Eq. (8), where μ_A is the average pixel value of image A, μ_B is the average pixel value of image B. σ_{AB} is the covariance of image A to image B. σ_A is the variance of image A and σ_B is the variance of image B. $C_1 = (k_1 L)^2$ and $C_2 = (k_2 L)^2$, where L is the image dynamic range (2^{bit}- 1). The default values of $k_1 = 0.01$ and $k_2 = 0.03$.

$$SSIM\ (A, B) = \frac{(2\mu_A\mu_B + C_1)(2\sigma_{AB} + C_2)}{(\mu_A^2 + \mu_B^2 + C_1)(\sigma_A^2 + \sigma_B^2 + C_2)} \quad (8)$$

The PSNR [32] is an objective technique used to quantify image quality, and its scores closely resemble those of methods used by humans to detect images. We use the PSNR metric to measure the uniformity of an image compared to the reference image. We express PSNR in logarithmic decibels (dB) [29]. The degree of resemblance between the two images increases with increasing PSNR value. Eq. (10) can be used to calculate PSNR, with C_{max}^2 representing the largest pixel

value throughout the image. The MSE (Mean Square Error) number is needed to compute the PSNR. The squared error number resulting from all observed pixels is known as the mean square error, or MSE. Eq. (9), where x and y are the image's pixel coordinates, shows the MSE formula. The image's dimensions are M and N. The original image (C) is the ground truth, while S is the repaired image.

$$MSE = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}(S_{xy} - C_{xy})^2}{MN} \quad (9)$$

$$PSNR = 10\log_{10}\left(\frac{C_{max}^2}{MSE}\right) \quad (10)$$

4.3 Comparative experiments

We applied the model evaluation from this research to the damaged Javanese character dataset, which took the form of printed and handwritten letters. Figure 8 illustrates the restoration outcomes of the corrupted Javanese character dataset, represented as printed letters. Figure 9 shows the restoration results of handwritten corrupted Javanese characters. If you use the Wasserstein Generative Adversarial Network-Gradient Penalty (WGAN-GP) method to fix broken Javanese characters, you get an image that looks a lot like the original image (Ground Truth). The red boxed image area shows the damaged image area and the restored image area.

The trial using the printed Javanese characters dataset yields a restoration image that closely resembles the original image (GT), but the texture of the restored image in the damaged area lacks smoothness. This is different from the restoration results using the handwritten Javanese character dataset. In addition to conforming to the original image's shape, the restored image's texture in the damaged area is smooth.

Figure 8 and Figure 9 show the restoration results whose structure matches the original image (Ground Truth), whether the damage is in the upper left, upper right, lower left, lower right, and center areas. Figure 9 shows that even though the position of the Javanese characters is not upright, which is tilted to the left and tilted to the right, the restoration result is similar to the shape of the original image.

4.3.1 Quantitative experiments

Currently, there is no uniform metric to evaluate the results. In this research, a quantitative comparison is made using SSIM and PSNR. To show the effectiveness of the proposed method, this study compares it with other methods, namely the Deep Convolution Generative Adversarial Network (DCGAN) method [14] and the Text Block Identification method [7]. The comparison with the two methods was tested on the printed Javanese characters dataset. While the test on the handwritten Javanese characters dataset was only compared with the DCGAN method. The results of the quantitative comparison between the proposed method in this research with the DCGAN method and the Text Block Identification method are shown in Table 2. While the results of the quantitative comparison of the test on the handwritten Javanese characters dataset between the proposed method in this research and the DCGAN method can be seen in Table 3.

Table 2 and Table 3 show that the SSIM and PSNR results of the method in this research, WGAN-GP, are superior to other methods. The SSIM index increased by 2.07% for the printed Javanese research dataset, and 7.48% for the handwritten Javanese characters dataset. The PSNR index increased by 3.35 dB and 6.04 dB for the printed Javanese characters dataset and the handwritten Javanese characters dataset, respectively.

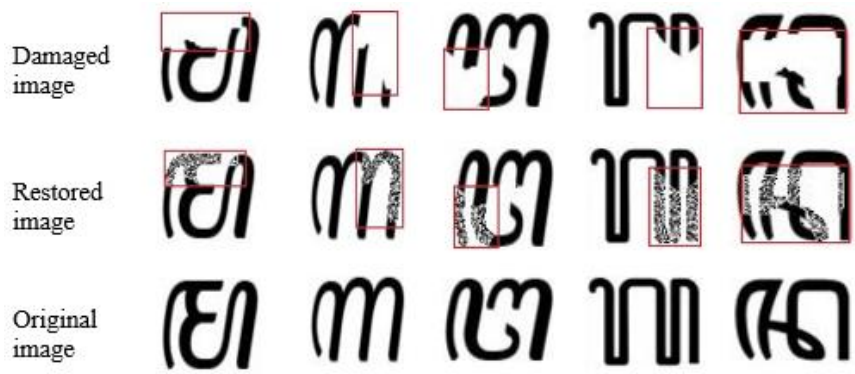


Figure 8. Example of restoration results for a dataset of damaged Javanese characters in the form of printed letters

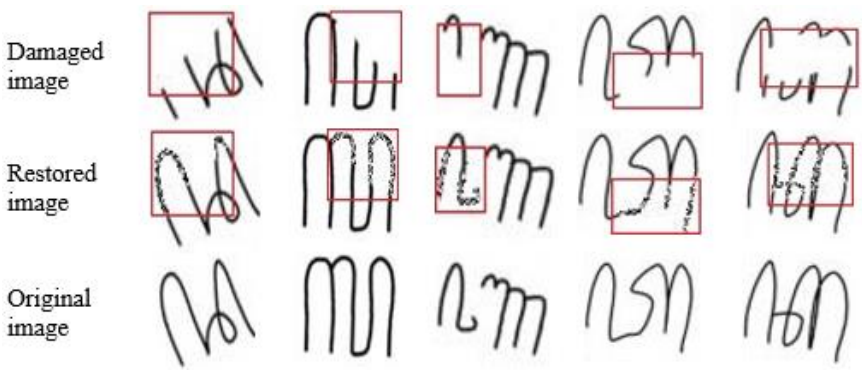


Figure 9. Example of restoration of damaged Javanese characters in the form of handwriting

Table 2. Results of quantitative experiments on printed Javanese characters dataset

Performance Metrics	Methods in Research (WGAN-GP)	DCGAN	Text Block Identification
Accuracy	-	-	82.07%
SSIM	85.12%	83.05%	-
PSNR	23.52 dB	20.17 dB	-

Table 3. Results of quantitative experiments on handwritten Javanese characters dataset

Performance Metrics	Methods in Research (WGAN-GP)	DCGAN
SSIM	90.67%	83.19%
PSNR	27.13 dB	21.09 dB

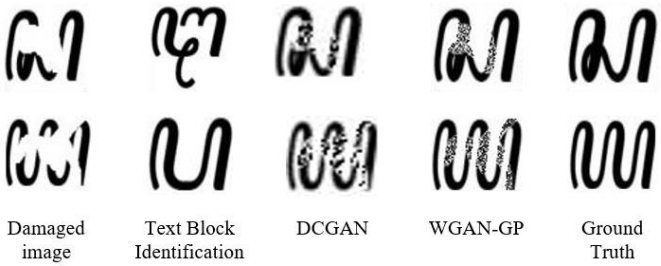


Figure 10. Example of qualitative comparison on printed Javanese characters dataset

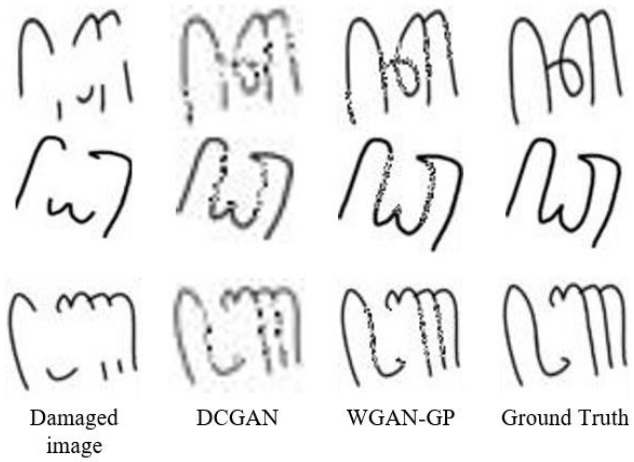


Figure 11. Example of qualitative comparison on handwritten Javanese characters dataset

4.3.2 Qualitative experiments

Figure 10 and Figure 11 show examples of comparison of experimental results between the proposed method in this research and other methods. From left to right shows the damaged Javanese characters (Input), the restoration result using Text Block Identification method, Deep Convolution Generative Adversarial Network (DCGAN) method, the method used in this research Wasserstein Generative Adversarial Network-Gradient Penalty (WGAN-GP), and the last column is the original Javanese characters (Ground Truth).

The Text Block Identification method simply infers probable structures from the remaining information in the broken Javanese characters, and the findings are likely to be incorrect. This is in contrast to the results of the qualitative comparison on the printed Javanese characters dataset, which guarantee visuals. The damaged Javanese character "Sa" can be seen in the first row of the second column of Figure 10. "Tha" are the Javanese characters that arise from the restoration process using Text Block Identification. Similar to Figure 10, where "Ya" is the damaged Javanese character in the second row of the second column and "Pa" is the outcome of repair. The missing portion of the character is not entirely restored when utilizing the DCGAN approach. Although the method partially restores the destroyed character structure, there are several inaccuracies in the specifics. As a result, the restored image is poor. The WGAN-GP approach, which was employed in this study, can restore the majority of damaged areas and guarantee that the strokes are shaped correctly. The WGAN-GP method's restoration results essentially resemble the original character (Ground Truth).

Figure 11 contains an example of a qualitative comparison on a handwritten Javanese characters dataset between the DCGAN method and the WGAN-GP method (the method used in this study). The handwritten characters have a thinner letter thickness compared to the printed characters. The restoration results using the DCGAN method are less perfect but the shape is similar to the original characters (Ground Truth). The same is the case with the restoration results on the printed Javanese characters dataset. The DCGAN method repairs the damaged character structure to some extent, but there are many errors in the details. The restoration results using the WGAN-GP method are smooth in texture. The restoration results on the damaged areas are not visibly different from the undamaged areas.

In general, trials using both printed and handwritten Javanese characters datasets show that the WGAN-GP method is superior to the other methods (Text Block Identification and DCGAN) in every indicator. Figures 8-11 show that the WGAN-GP method is able to restore Javanese characters with a large area of damage and can guarantee the correct stroke shape of the repaired characters. In addition, the WGAN-GP method is also able to restore damaged Javanese characters with left-skewed and right-skewed positions.

The SSIM and PSNR values reported in this study are comparable to those found in similar works on character restoration. For instance, in the restoration of Chinese characters using CGAN, Weng et al. [1] reported an SSIM of around 84% and a PSNR of 22 dB, indicating that the WGAN-GP approach provides a more robust restoration framework. Additionally, in a study on Yi alphabets by Su et al. [4], the reported SSIM and PSNR values ranged between 80-85% and 20-23 dB, further underscoring the effectiveness of the proposed WGAN-GP method for Javanese character restoration.

The SSIM metric measures how structurally similar the restored and original images are, with 1 indicating perfect similarity. The PSNR value is expressed in decibels (dB), where higher values indicate better image quality, closer to the original. Typically, PSNR values above 20 dB indicate good restoration quality, while values above 30 dB are considered near perfect in most image restoration tasks.

The qualitative and quantitative results combined demonstrate the superior performance of the WGAN-GP method. The gradient penalty employed in the WGAN-GP model ensures stable training, leading to more accurate restoration, particularly in preserving the stroke shapes and font styles critical to Javanese characters. The method's ability to handle large areas of damage, where other methods fairly

make it a robust solution for digital manuscript restoration.

The WGAN-GP's ability to maintain consistency in the restored character's structure, even in cases of severe damage or character tilt, is a key observation from both the printed and handwritten datasets. This ensures that the restored character remains true to the original font and style, which is crucial for historical manuscripts where accuracy is paramount.

However, the texture quality in some of the printed Javanese characters still presents a minor challenge. While the structural accuracy is maintained, the lack of smoothness in certain areas of the restored texture indicates that future work could focus on incorporating texture-specific features in the model to further improve visual consistency.

In conclusion, the results clearly show that the WGAN-GP method outperforms existing methods in terms of both visual and quantitative metrics, with superior accuracy in restoring large damaged areas while maintaining structural integrity.

5. CONCLUSION

5.1 Contributions

This study presents the WGAN-GP method as a novel approach for restoring Javanese characters with a large area of damage. The main contributions of this research are as follows:

(a) Restoration of large-damaged characters: The WGAN-GP-based restoration method successfully restores Javanese characters with significant structural damage, addressing both visual consistency and the correct topology of the strokes, an area where previous methods have struggled.
(b) Handwritten and printed character restoration: The method is versatile, performing well on both printed and handwritten Javanese character datasets, showing improvements over methods like DCGAN and Text Block Identification.
(c) Gradient stability: By introducing the gradient penalty, the proposed method ensures stable training, avoiding common issues such as gradient vanishing and mode collapse, which are typical in traditional GAN models.

The results, both qualitative and quantitative, show that the WGAN-GP method outperforms other restoration techniques. The SSIM and PSNR values achieved indicate high restoration accuracy, with values of 85.12% (SSIM) and 23.52 dB (PSNR) for printed characters, and 90.67% (SSIM) and 27.13 dB (PSNR) for handwritten characters.

5.2 Limitations

While the proposed method demonstrates significant improvements, there are several limitations that should be addressed in future work:

(a) Dataset bias: The dataset used in this study may have inherent biases, as it consists of a limited range of Javanese characters from specific manuscripts. The model's performance on other types of ancient manuscripts or non-Javanese scripts has not been tested, which limits the generalizability of the findings.
(b) Texture quality in printed characters: The results for printed Javanese characters indicate that while the structural restoration is accurate, the texture quality in the restored areas is not always smooth. This could be attributed to the model's focus on stroke formation over texture consistency. Further improvements could be made by incorporating texture-specific features in the model.
(c) Applicability to other scripts: While the method performs well on Javanese characters, it has not been applied to other types of ancient scripts with different structural and stylistic features. Future research should explore the adaptability of the model to other languages and manuscripts.

5.3 Future work

To address these limitations and further enhance the capabilities of the WGAN-GP model, several avenues for future research are proposed:

(a) Expanding the dataset: Future work should include a more diverse dataset encompassing various types of Javanese manuscripts, as well as other ancient scripts from different cultural and linguistic backgrounds, to test the generalizability of the model.
(b) Incorporating texture features: Enhancing the model by incorporating texture-specific generative features could improve the quality of texture restoration, particularly for printed manuscripts where smoothness is critical.
(c) Real-time restoration systems: Developing a real-time restoration system using this model could be beneficial for practical applications, especially for museums and libraries. Exploring lightweight versions of the model suitable for deployment on edge devices could also be considered.
(d) Comparative studies with newer models: As new GAN variants continue to be developed, comparing WGAN-GP with more advanced techniques could provide insights into further improvements in accuracy and speed.

ACKNOWLEDGMENT

This research was financially supported by Bima Kemdikbudristek Republic of Indonesia (Grant No.: 101/E5/PG.02.00.PL/2023(009/UN46.4.1/PT.01.03/2024)).

REFERENCES

[1] Weng, Y., Zhou, H.W., Wan, J. (2019). Image inpainting technique based on smart terminal: A case study in CPS ancient image data. IEEE Access, 7: 69837-69847. https://doi.org/10.1109/ACCESS.2019.2919326
[2] Jo, I.S., Choi, D.B., Park, Y.B. (2021). Chinese character image completion using a generative latent variable model. Applied Sciences, 11(2): 624. https://doi.org/10.3390/app11020624
[3] Song, G., Li, J., Wang, Z. (2020). Occluded offline handwritten Chinese character inpainting via generative adversarial network and self-attention mechanism. Neurocomputing, 415: 146-156. https://doi.org/10.1016/j.neucom.2020.07.046
[4] Su, B.P., Liu, X.X., Gao, W.Z., Yang, Y., Chen, S.X. (2022). A restoration method using dual generate adversarial networks for Chinese ancient characters. Visual Informatics, 6(1): 26-34. https://doi.org/10.1016/j.visinf.2022.02.001
[5] Sun, G.B., Zheng, Z.J., Zhang, M. (2022). End-to-end rubbing restoration using generative adversarial

networks. ArXiv, 1-8. https://doi.org/10.48550/arXiv.2205.03743
[6] Arismadhani, A., Prananto, B.B., Susanti, R. (2013). Aplikasi belajar menulis aksara Jawa menggunakan Android. Jurnal Teknik Pomits, 2(1): 1. http://ejurnal2.its.ac.id/index.php/teknik/article/view/2732.
[7] Himamunanto, A.R., Setyowati, E. (2018). Text block identification in restoration process of Javanese script damage. Journal of Physics: Conference Series, 1013(1): 012210. https://doi.org/10.1088/1742-6596/1013/1/012210
[8] Cao, Y.J., Jia, L.L., Chen, Y.X., Lin, N., Yang, C., Zhang, B., Liu, Z., Li, X.X., Dai, H.H. (2018). Recent advances of generative adversarial networks in computer vision. IEEE Access, 7: 14985-15006. https://doi.org/10.1109/ACCESS.2018.2886814
[9] Radford, A., Metz, L., Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In 4th International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, pp. 1-16. https://doi.org/10.48550/arXiv.1511.06434
[10] Arjovsky, M., Chintala, S., Bottou, L. (2017). Wasserstein GAN. arXiv. https://doi.org/10.48550/arXiv.1701.07875
[11] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C. (2017). Improved training of Wasserstein GANs. In Advances in Neural Information Processing Systems, pp. 5678-5778. https://proceedings.neurips.cc/paper/2017/hash/892c3b1c6dccd52936e27cbd0ff683d6-Abstract.html.
[12] Lee, G.C., Li, J.H., Li, Z.Y. (2023). A Wasserstein generative adversarial network—Gradient penalty-based model with imbalanced data enhancement for network intrusion detection. Applied Sciences, 13(14): 8132. https://doi.org/10.3390/app13148132
[13] Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B. (2018). Image inpainting for irregular holes using partial convolutions. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, pp. 85-100. https://doi.org/10.1007/978-3-030-01252-6_6
[14] Sabini, M., Rusak, G. (2018). Painting outside the box: Image outpainting with gans. ArXiv:1808.08483. https://doi.org/10.48550/arXiv.1808.08483
[15] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A. (2016). Context encoders: Feature learning by inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, pp. 2536-2544. https://doi.org/10.1109/CVPR.2016.278
[16] Iizuka, S., Simo-Serra, E., Ishikawa, H. (2017). Globally and locally consistent image completion. ACM Transactions on Graphics (ToG), 36(4): 1-14. https://doi.org/10.1145/3072959.3073659
[17] Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., Li, H. (2017). High-resolution image inpainting using multi-scale neural patch synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA, pp. 6721-6729. https://doi.org/10.1109/CVPR.2017.434
[18] Yeh, R.A., Chen, C., Lim, T.Y., Schwing, A.G., Hasegawa-Johnson, M., Do, M.N. (2017). Semantic image inpainting with deep generative models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA, pp. 5485-5493. https://doi.org/10.1109/CVPR.2017.728
[19] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T. (2019). Free-form image inpainting with gated convolution. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea (South), pp. 4470-4479. https://doi.org/10.1109/ICCV.2019.00457
[20] Li, J.Y., Wang, N., Zhang, L.F., Du, B., Tao, D.H. (2020). Recurrent feature reasoning for image inpainting. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, pp. 7757-7765. https://doi.org/10.1109/CVPR42600.2020.00778
[21] Liu, J.Y., Yang, S., Fang, Y.M., Guo, Z.M. (2018). Structure-guided image inpainting using homography transformation. IEEE Transactions on Multimedia, 20(12): 3252-3265. https://doi.org/10.1109/TMM.2018.2831636
[22] Kawai, N., Sato, T., Yokoya, N. (2015). Diminished reality based on image inpainting considering background geometry. IEEE Transactions on Visualization and Computer Graphics, 22(3): 1236-1247. https://doi.org/10.1109/TVCG.2015.2462368
[23] Ružić, T., Pižurica, A. (2014). Context-aware patch-based image inpainting using Markov random field modeling. IEEE Transactions on Image Processing, 24(1): 444-456. https://doi.org/10.1109/TIP.2014.2372479
[24] Levin, A., Zomet, A., Weiss, Y. (2003). Learning how to inpaint from global image statistics. In Proceedings of the IEEE International Conference on Computer Vision, Nice, France, pp. 305-312. https://doi.org/10.1109/iccv.2003.1238360
[25] Zhao, H.L., Guo, H.Y., Jin, X.G., Shen, J.B., Mao, X.Y., Liu, J.R. (2018). Parallel and efficient approximate nearest patch matching for image editing applications. Neurocomputing, 305: 39-50. https://doi.org/10.1016/j.neucom.2018.03.064
[26] Fan, Q., Zhang, L. (2018). A novel patch matching algorithm for exemplar-based image inpainting. Multimedia Tools and Applications, 77: 10807-10821. https://doi.org/10.1007/s11042-017-5077-z
[27] Li, H.D., Luo, W.Q., Huang, J.W. (2017). Localization of diffusion-based inpainting in digital images. IEEE Transactions on Information Forensics and Security, 12(12): 3050-3064. https://doi.org/10.1109/TIFS.2017.2730822
[28] Zhai, J.T., Lin, P., Cui, Y.F., Xu, L.L., Liu, M. (2023). GraphCWGAN-GP: A novel data augmenting approach for imbalanced encrypted traffic classification. CMES-Computer Modeling in Engineering & Sciences, 136(2): 2069-2092. https://doi.org/10.32604/cmes.2023.023764
[29] He, S., Schomaker, L. (2019). DeepOtsu: Document enhancement and binarization using iterative deep learning. Pattern Recognition, 91: 379-390. https://doi.org/10.1016/j.patcog.2019.01.025
[30] Féraud, J.F. (2017). S—Sim. Dictionnaire Critique de la Langue Française, 13(4): 506-575. https://doi.org/10.1515/9783110914252-043
[31] Abd Alhussain, Z.F., Hassan, A.F. (2023). A binary relation fuzzy soft matrix-theoretic approach to image

quality measurement: Comparison with statistical similarity metrics. Mathematical Modelling of Engineering Problems, 10(3): 799-804. https://doi.org/10.18280/mmep.100309

[32] Fatman, A.N., Ahmad, T., Jean De La Croix, N., Hossen, M.S. (2023). Enhancing data hiding methods for improved cyber security through histogram shifting direction optimization. Mathematical Modelling of Engineering Problems, 10(5): 1508-1514. https://doi.org/10.18280/mmep.100502