

Enhancing the Scanability of Damaged QR Codes Through Image Restoration Using GANs Combined with the Spectral Normalization Technique

Puwadol Sirikongtham, Apichaya Nimkoompai

Faculty of Information Technology, Thai-Nichi Institute of Technology, Thailand

Abstract—QR Codes are widely used in the digital era for storing and sharing information in various applications. However, they are often susceptible to physical damage such as scratches, tears, or fading, which can result in scanning failures and limit their usability. To overcome this issue, this research introduces a Generative Adversarial Network (GAN) model integrated with Spectral Normalization to restore damaged QR Code images. The model was trained and evaluated using a dataset of QR Codes with simulated damage ranging from 1% to 60%. Experimental results demonstrate that the proposed approach effectively reconstructs missing parts of QR Codes while preserving structural details and module sharpness. The model achieved an average PSNR of 28.5 dB, SSIM of 0.91, and a scanning success rate of 88%, outperforming U-Net (68%) and a baseline GAN (75%). Although the processing time is slightly longer, the model offers superior accuracy and robustness, particularly for severely damaged QR Codes (40% to 60% damage). These findings confirm that GANs enhanced with Spectral Normalization offer a promising solution for QR Code restoration, with potential uses in digital marketing, payment systems, and inventory management.

Keywords—QR Code restoration; Generative Adversarial Networks; spectral normalization; image inpainting; deep learning; damage reconstruction

I. INTRODUCTION

QR Codes (Quick Response Codes) have become a widely popular technology in the digital age due to their ability to store diverse information and their user-friendly nature [1]. This has led to their application in various fields, such as marketing, payment systems, product data verification, and information access systems.

However, in real-world environments, QR Codes often face problems of physical damage, such as missing parts or scratches. These can be caused by various environmental factors, leading to QR Code scanning failures and user inconvenience.

This damage can arise from several factors, including poor QR Code printing quality or contact and abrasion with various surfaces [2]. When a QR Code is damaged, scanning to read the data may fail or result in errors, which impacts operational efficiency and the user experience.

Although deep learning technology is highly potential and widely used in general image restoration and inpainting tasks [3], its specific application to address physically damaged QR

Codes, such as those with scratches or missing sections, requires further study and development. For this reason, this study utilized Generative Adversarial Networks (GANs)—a deep learning technique with a remarkable ability to learn and generate complex image data to reconstruct and regenerate the missing parts of a QR Code. GANs have the potential to produce sharp details, which is a critical feature for preserving and inpainting the black-and-white module structure of a QR Code, essential for scanning. This contrasts with some conventional CNN architectures, which may tend to produce blurrier results when filling in missing areas.

This research developed a specialized GAN model incorporating Spectral Normalization to enhance the model's stability and performance. The objective was to test the effectiveness of GANs in improving the scanability and data recovery from damaged QR Codes.

II. THEORY AND RELATED RESEARCH

A. QR Code and basic Structure

QR Code (Quick Response Code) is a two-dimensional barcode technology developed by the company Denso Wave in 1994. QR Codes can store more data than traditional barcodes [4] and can be read by scanning devices or smartphones. The information is encoded in a pattern of black and white dots arranged in a two-dimensional grid. The basic structure of a QR Code consists of several key parts: For Finder Patterns, these are square patterns located at the three corners of the QR Code. They help scanning devices quickly detect and determine the position of the QR Code. For Alignment Patterns, these are smaller patterns that help adjust the position and correct for distortion that may occur from image skewing. For Timing Patterns, these are vertical and horizontal lines that help define the size and position of the data modules. For Data and Error Correction Codes, this section stores the data and error correction codes. These codes allow the QR Code to be read even if it is partially damaged, but it becomes unreadable if the damage is too extensive.

B. QR Code Damage and Its Impact

As shown in Fig. 1, QR Codes often suffer from physical damage in real-world environments, such as missing parts or scratches caused by external factors. This damage can be categorized into several main types: Physical Damage, caused by physical wear and tear, such as scratching or tearing; Environmental Damage, caused by unsuitable environmental conditions; Printing Defects, caused by low-quality printing,

which makes the QR Code unclear or flawed [5]. These types of damage can lead to QR Code scanning failure or data reading errors, causing inconvenience and reducing operational efficiency.



Fig. 1. An example of a damaged QR Code on an EV charging station at a gas station.

C. Image Inpainting: Convolutional Neural Networks (CNNs) Technique

Convolutional Neural Networks (CNNs) are a deep learning technique that has revolutionized image inpainting, or image restoration. CNNs have a remarkable ability to automatically learn features and spatial relationships within an image, allowing them to understand the image context and realistically generate missing parts. Typically, a CNN for inpainting uses an encoder-decoder structure [6]. The encoder extracts important information from the area surrounding the damaged portion, and the decoder uses that information to create new pixels to replace the missing ones. Training with a large amount of image data helps the model learn to fill in images accurately and seamlessly. CNNs can create complex structures and natural-looking textures better than simply estimating values from neighboring pixels. They can also complete objects or even parts that require semantic understanding, such as faces.

D. Image Inpainting: GANs Technique

Generative Adversarial Networks (GANs) are a type of neural network architecture consisting of two main networks that operate in an adversarial manner: the Generator network, which is responsible for producing new data, and the Discriminator network, which determines whether the generated data is real or fake [7]. This adversarial learning mechanism enables GANs to achieve outstanding performance in generating highly realistic data, making them particularly effective for image inpainting or repairing missing parts of an image. In this type of work, the Generator receives an image with a damaged portion (often accompanied by a mask that indicates the location of the damage) as input. It then learns to "imagine" and create pixels to fill that area, attempting to produce a result that looks as realistic and coherent with the rest of the image as possible. Meanwhile, the Discriminator is trained to distinguish between real images (undamaged original images) and fake images (images completed by the Generator). The Discriminator's goal is to catch the images created by the Generator, while the Generator continuously improves itself to create images that are so convincing that the Discriminator cannot tell the difference. This competitive process continues, making both networks more proficient. The Generator becomes progressively better at creating completed images with

complex details, natural-looking textures, and structures that align with the image's context, while the Discriminator's ability to distinguish between real and fake images becomes sharper.

E. Enhancing GAN Stability with Spectral Normalization

Spectral Normalization (SN) is a crucial technique for improving the stability of GAN training by controlling the Lipschitz constant of network weights [8]. While SN is generally applied to the Discriminator to directly regulate gradients and enhance training stability, in this study we extend its application to the Convolutional layers of the Generator (as illustrated in Algorithm 1: Build_Generator). In the context of QR Code inpainting, our initial hypothesis was that stabilizing the Generator's gradients with SN could improve the synthesis of fine details and repetitive patterns that are critical to QR Code structures (e.g., clear black-and-white modules). This stabilization helps reduce artifacts and enables the Generator to better capture the underlying distribution of complete QR Codes. Preliminary experiments support this hypothesis, demonstrating improved structural reconstruction when SN was applied to the Generator compared with cases where SN was applied only to the Discriminator or not used at all. Consequently, this configuration was adopted for the main experiments.

F. Applying Image Inpainting to QR Code Images

Although most research in QR Code image processing has primarily focused on detection or error correction during decoding, some studies have begun to explore the use of deep learning techniques to improve QR Code image quality in different contexts. For instance, study [9] demonstrated the potential of deep learning in recovering invisible QR Codes displayed on specialized screens, while study [10] employed deep learning methods for rapid deblurring of QR Code images. Although these works did not directly address physical damage (e.g., scratches or missing parts) as this study does, they highlight the capability of neural networks to learn and adapt to the unique structural properties of QR Codes. However, based on a review of the related literature, no prior research has specifically applied Generative Adversarial Networks (GANs) combined with Spectral Normalization (SN) for inpainting physically damaged QR Codes. This represents an innovative direction. While SN has been widely used to stabilize the Discriminator in general image generation tasks, this study is the first to apply SN to the Generator for inpainting images with highly geometric structures, such as QR Codes, which require sharpness and well-preserved edges. This introduces a distinct challenge: the primary difficulty in QR Code restoration lies in maintaining edge boundaries and the sharpness of individual modules. In this regard, GANs, particularly when combined with loss functions such as L1 Loss that promote sharpness, tend to outperform traditional CNN encoder-decoder architectures, which often produce blurry results in the inpainted regions. This research, therefore, aims to develop and evaluate such an approach to fill this gap.

III. RESEARCH METHODOLOGY

A. Data Preparation

For the data preparation of QR Code restoration, we created a large dataset of QR Codes using the QR Code Python library

and then simulated damage on them. We used OpenCV to add various types of damage to the QR Codes, simulating missing parts by creating a mask to define the missing areas and filling them with white. This was done in both random patches and scratches with varying directions and thicknesses, as shown in Fig. 2.

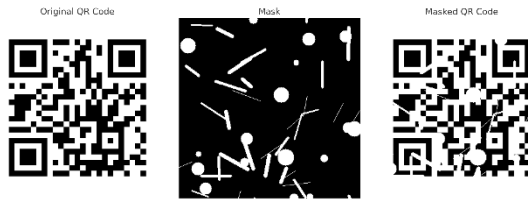


Fig. 2. Example of a simulated damaged QR Code.

1) *QR Code dataset creation*: In the data preparation phase, the researchers created a dataset of 1,000 QR Code images, as shown in Fig. 3, using the qrcode Python library. As shown in Table I, the images were then divided into three groups based on their damage level: minor damage (1 to 20%) with 333 images, moderate damage (20 to 40%) with 333 images, and severe damage (40 to 60%) with 334 images. Each group was then randomly and equally split into a Training Set (70%), a Validation Set (15%), and a Test Set (15%) to ensure a balanced representation of various damage characteristics across all datasets. Each QR Code contained a unique URL.

TABLE I. DATA SPLITTING

Dataset	Number of Images	Proportion (%)
Training Set	700	70%
Validation Set	150	15%
Test Set	150	15%

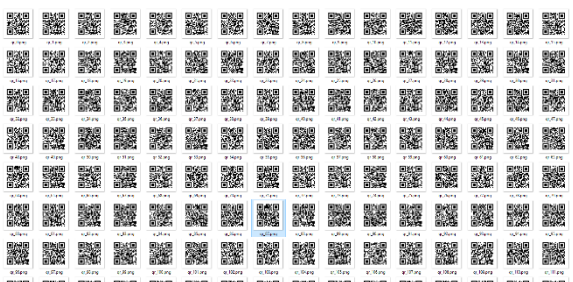


Fig. 3. Simulation of creating 1,000 QR Code images.

2) *Creating damage in QR Codes*: The researchers used OpenCV to add various types of damage to the QR Codes. Missing parts were created by generating a mask to define the missing areas and filling them with white. This was done in both random patch shapes and linear shapes (simulating scratches) with random thickness and direction across the QR Code. The damage was randomly assigned a severity level between 1% and 60%, as shown in Fig. 4. The purpose of setting the damage level from 1% to 60% was to cover a wide range of real-world scenarios: Minor damage (1 to 20%) reflects common issues like color fading, light scratches, or dust. Moderate damage (20 to 40%) simulates a QR Code that

has been in use for some time or has multiple scratches. Severe damage (40 to 60%) is used to test the system's performance when the QR Code has lost critical data. The reason for limiting the damage level to a maximum of 60% is that it represents a realistic boundary for data recovery in a general use context. QR Codes are embedded with an Error Correction Code that can withstand a certain level of damage. Even the highest level of error correction (level H) can only handle approximately 30% damage. Therefore, creating damage higher than this, such as 40 to 60%, simulates a more severe situation than the standard threshold, testing the model's capacity to repair data in cases where the QR Code is damaged beyond what the original system can correct on its own. Conversely, if the damage exceeds 60%, the core structure of the QR Code may be destroyed to a point where it is not reasonably recoverable, even with deep learning techniques. Thus, the damage level was limited to a maximum of 60% to maintain a balance between the model's capability and the practical possibility of image recovery.

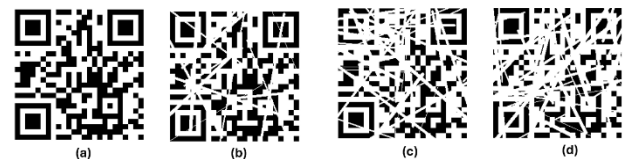


Fig. 4. An image showing the simulation of missing parts of a QR code. Code (a) original image, (b) damaged 1-20%, (c) damaged 20-40%, (d) damaged 40-60%.

B. Model Design

For the design of the model in QR Code Restoration, we developed a Generative Adversarial Network (GAN) architecture implemented using TensorFlow and Keras, consisting of two main components: the Generator and the Discriminator. As illustrated in Pseudo-code 5, the Generator is responsible for reconstructing the missing regions of the QR Code. It adopts an Encoder-Decoder structure with Spectral Normalization applied to every Convolutional layer to control the gradient distribution and enhance training stability. This architecture enables the Generator to produce realistic outputs, particularly in the critical regions of QR Codes such as Finder Patterns and Alignment Patterns.

The Discriminator, on the other hand, is designed to distinguish between real images and those generated by the Generator, thereby forcing the Generator to create outputs with higher realism.

- The Generator is the model responsible for reconstructing the missing regions of the QR Code. It is based on a Convolutional Neural Network (CNN) with an Encoder-Decoder architecture. The Encoder compresses the image while extracting meaningful features, whereas the Decoder reconstructs the image from these extracted features. The model employs LeakyReLU and ReLU as activation functions. Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance

dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.

- The Discriminator is a model designed to distinguish between real images and those generated by the Generator. It employs a CNN architecture to extract features from the images. The activation functions used are LeakyReLU and Sigmoid (see Algorithm 2).
- The GAN Model integrates the Generator and Discriminator into a complete Generative Adversarial Network (GAN). The Discriminator is compiled with the binary cross-entropy loss function and the Adam optimizer. The GAN is compiled with a combination of binary cross-entropy loss and L1 Loss (pixel-wise difference loss), along with the Adam optimizer (see Algorithm 3).

Algorithm 1: Build_Generator

Input: input_shape

Output: Generator model

1. Apply spectral normalized Conv2D (64 filters, kernel=4, stride=2) on input_shape
2. Apply LeakyReLU activation ($\alpha = 0.2$)
3. Apply spectral normalized Conv2D (128 filters, kernel=4, stride=2)
4. Apply Batch Normalization
5. Apply LeakyReLU activation ($\alpha = 0.2$)
6. Apply spectral normalized Conv2DTranspose (128 filters, kernel=4, stride=2)
7. Apply Batch Normalization
8. Apply ReLU activation
9. Apply spectral normalized Conv2DTranspose (64 filters, kernel=4, stride=2)
10. Apply Batch Normalization
11. Apply ReLU activation
12. Apply Conv2D (1 filter, kernel=4, activation = sigmoid)
13. Return **Generator Model**

Algorithm 2: Build_Discriminator

Input: input_shape

Output: Discriminator model

1. Apply Conv2D (64 filters, kernel=4, stride=2) on input_shape
2. Apply LeakyReLU activation ($\alpha = 0.2$)
3. Apply Conv2D (128 filters, kernel=4, stride=2)
4. Apply Batch Normalization
5. Apply LeakyReLU activation ($\alpha = 0.2$)
6. Apply Flatten operation
7. Apply Dense layer (1 unit, activation = sigmoid)
8. Return **Discriminator Model**

Algorithm 3: Build_GAN

Output: GAN model

1. Build Generator using **Algorithm 1**
2. Build Discriminator using **Algorithm 2**

-
3. Set Discriminator as **non-trainable**
 4. Connect Generator output $\rightarrow$ Discriminator input
 5. Compile the GAN model using the Adam optimizer and binary cross-entropy loss
 6. Return **GAN Model**
-

C. Model Training

The model training process utilized the concept of Generative Adversarial Networks (GANs), which consists of two main parts: a Generator and a Discriminator. They were trained in an adversarial manner, starting with loading the complete and damaged QR Code images prepared in the previous step. The pixel values were then normalized to a range of $[0, 1]$ to enhance computational efficiency. During training, the data was divided into mini-batches of 32 images for each epoch. The Generator received the damaged image as input and attempted to reconstruct a complete image. Simultaneously, the Discriminator received images from both the Generator (generated images) and the real images (complete images), and learned to distinguish between the two. The learning of these two components was driven by different loss functions:

- The Discriminator used Binary Cross-Entropy Loss to measure the difference between its predictions and the ground truth. This loss function effectively helped the model learn to distinguish between real and generated QR Code images.
- The goal of the Generator is to produce images that are both visually realistic (capable of fooling the Discriminator) and structurally close to the original ground truth. Therefore, its loss function is defined as a combination of:
- Adversarial Loss, computed from the output of the Discriminator (commonly using Binary Cross-Entropy), this loss encourages the Generator to synthesize more realistic images.
- L1 Loss (Mean Absolute Error). This measures the average pixel-wise difference between the generated image and the ground truth. Incorporating L1 Loss is critical, as it has been shown to preserve details and edge sharpness better than L2 Loss—an essential factor for maintaining the distinct black-and-white modules of QR Codes. Furthermore, L1 Loss provides pixel-level constraints that complement the Adversarial Loss, promoting sharper reconstructions and preserving the fine-grained structural integrity required for QR Code restoration.

The key hyperparameters were optimized through preliminary validation experiments, with the learning rate set to 0.0002 and the optimizer chosen as Adam, following best practices in GAN training.

Training was conducted for approximately 100 epochs, during which both Generator and Discriminator losses were computed and weights updated via backpropagation. This iterative process enables the Generator to learn to produce increasingly realistic and accurate reconstructions, while

simultaneously enhancing the Discriminator's ability to distinguish between real and generated images.

During training, Spectral Normalization was applied to all convolutional layers of the Generator in order to regulate the Lipschitz constant of its weight matrices. This improves gradient stability, prevents imbalance with the Discriminator, and mitigates mode collapse, a common challenge in GAN training without introducing additional computational overhead. In addition, a Model Checkpoint callback was employed to save the best-performing model weights throughout the training process.

D. Model Evaluation

The evaluation focuses on assessing the model's ability to restore/inpaint corrupted QR Code images back to a complete and functional state. To achieve this, standard image restoration metrics and task-specific measures (such as the scanning success rate) relevant to QR Code usability were employed:

PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index): Both are standard metrics in image restoration tasks. PSNR, measured in decibels (dB), quantifies the pixel-level difference between the generated image and the ground truth, with higher values indicating better reconstruction quality. SSIM evaluates structural similarity between the two images, ranging from 0 to 1, where values closer to 1 indicate higher structural fidelity.

Scanning Success Rate is the metric directly evaluating whether the restored QR Codes (from the test set inputs) can be successfully decoded. Each output image was tested using the widely adopted QR Code decoding library pyzbar. The number of successfully decoded images was counted and expressed as a percentage relative to the total number of test images (150). This measure reflects the model's effectiveness in restoring the core functional property of QR Codes.

Average Inference Time per Image measures the processing speed of trained models (GAN+SN, baseline GAN, and U-Net) when applied to real QR Code restoration tasks. Evaluations were conducted using an NVIDIA GeForce RTX 3080 GPU. Before measurement, the GPU was warmed up to ensure stable maximum performance. Then, the forward-pass time of each image in the test set (150 images) was recorded, excluding file loading and preprocessing. The results were averaged to obtain the model's mean inference time per image.

IV. RESULTS

After completing the model training process, the evaluation phase was conducted to verify the model's performance. The test set, consisting of 150 QR Code images not previously seen during training, was used for this assessment. In this process, the corrupted QR Code images were fed into the trained model, and the reconstructed outputs were compared against the corresponding ground truth images.

As shown in Table II, during the initial training stage (Epoch 1), the Generator Loss started at 1.452 and the Discriminator Loss at 0.693, values typically observed when training begins for GAN models. The initial image quality was relatively low, with PSNR at 18.21 dB and SSIM at 0.72. By

Epoch 10, the Generator Loss had decreased to 0.985, while the Discriminator Loss slightly increased to 0.752, suggesting that the Generator was beginning to produce more realistic images. This improvement was reflected in the image quality metrics (PSNR: 22.45 dB, SSIM: 0.83). At Epoch 25, the Generator Loss continued to decrease to 0.712, while the Discriminator Loss increased to 1.103. This behavior is expected, as the Discriminator becomes less effective at distinguishing images once the Generator improves significantly. The image quality improved considerably (PSNR: 25.67 dB, SSIM: 0.89), and the scanning success rate on the validation set reached 82.1%, indicating increasing model stability. By Epochs 50 and 100, the Generator and Discriminator Loss values stabilized at approximately 0.5 and 1.45, respectively, suggesting convergence. The reconstructed image quality also reached consistently high levels (PSNR ~29 dB, SSIM ~0.93), and the validation set scanning success rate exceeded 90%. These results confirm the model's effectiveness in restoring corrupted QR Code images to complete and scannable forms. This reveals the system's enhanced performance. The Scan Success Rate on the test set was lower than on the validation set, which may be attributed to the diverse damage in critical areas like Finder Patterns or the irregular scratches present in the test set, as shown in Table III and Fig. 5.

TABLE II. GAN+SN MODEL TRAINING RESULTS

Epoch	G Loss	D Loss	PSNR (dB)	SSIM	Validation Set (%)
1	1.452	0.693	18.21	0.72	55.3%
10	0.985	0.752	22.45	0.83	68.7%
25	0.712	1.103	25.67	0.89	82.1%
50	0.501	1.452	28.92	0.92	90.5%
100	0.493	1.487	29.15	0.93	91.2%

TABLE III. EVALUATION RESULTS ON TEST SET (150 IMAGES)

Metric	Average	Range	Interpretation
PSNR (dB)	28.5	25.2 - 32.1	Values > 25 dB are considered acceptable
SSIM	0.91	0.85 - 0.95	Values close to 1 indicate high structural similarity
Scan Success Rate	88%	81 - 95%	Percentage of successful QR code scans

According to standard benchmarks, a PSNR greater than 25 dB is generally considered acceptable for image restoration tasks.



Fig. 5. Image showing the reconstruction of missing parts in a QR code using the GANs + SN technique: (a) Damaged QR code, not scannable; (b) GANs + SN reconstruction, scannable.

TABLE IV. COMPARISON OF INPAINTING METHODS WITH OTHER TECHNIQUES

Comparison Item	CNNs (U-Net)	GANs (Baseline)	GANs + Spectral Normalization
Algorithm Type	Learning-Based (CNN)	Learning-Based (GAN)	Learning-Based (GAN + Stabilization)
Architecture	U-Net (5 layers)	Encoder-Decoder	Encoder-Decoder + Spectral Norm
Loss Function	MSE + Perceptual Loss	Adversarial Loss + L1 Loss	Adversarial Loss + L1 Loss
Working Principle	Leams to restore images directly	Constructs new images through Generator–Discriminator	Similar to baseline GAN with stabilization (using Spectral Norm)
Average SSIM	0.82	0.87	0.91
Success Rate	68%	75%	88%
Processing Time	0.15 s	0.30 s	0.35 s
Strengths	- Faster than GANs - Suitable for less severe image Damage	- Generates more natural details - Produces sharper and more realistic structures	- High structural similarity - Reduces mode collapse problem - Achieves highest SSIM and success rate
Weaknesses	- Results may appear blurred - Ineffective with severe Damage	- Requires adversarial training - May encounter instability	- Slower processing time - Computational cost increases due to SN
Suitable for	Damage 1–20%	Damage 20–40%	Damage 40–60% (Highest Efficiency)

TABLE V. PERFORMANCE COMPARISON ACROSS DIFFERENT LEVELS OF DAMAGE

Damage Level	GAN+SN	GAN (Baseline)	CNN (U-Net)
1–20%	95%	85%	80%
20–40%	88%	75%	70%
40–60%	81%	65%	54%
Average	88%	75%	68%

From Table IV and Table V, the performance comparison of the proposed GAN+SN model with other baseline methods is presented as follows: CNNs (U-Net): Utilizes a standard 5-layer U-Net architecture commonly applied in image-to-image translation tasks, trained with a combined loss function of Mean Squared Error (MSE) and Perceptual Loss. GANs (Baseline): Employs the same Generator and Discriminator architectures as the proposed model but without Spectral Normalization, trained with Adversarial Loss combined with L1 Loss to enhance image sharpness. GANs + Spectral Normalization (Proposed Model): Trained with Adversarial Loss, L1 Loss, and applies Spectral Normalization to the Generator. All models were trained and evaluated under the same conditions and dataset to ensure fairness in comparison.

The high scanning success rate (88%) demonstrates our model's effectiveness in reconstructing not just visual

appearance but the functional data structure of QR codes, analogous to how shadow tomography reconstructs 3D information from limited 2D projections in microscopic imaging [13].

V. DISCUSSION

The comparison of the three main techniques for QR Code restoration showed that each method had different strengths and limitations. CNN (U-Net) was the fastest method, taking only 0.15 seconds per image. It achieved an SSIM of 0.82 and a scanning success rate of 68%, making it suitable for applications requiring high speed. However, it had limitations when the damage exceeded 20%, because although Perceptual Loss helped preserve the overall structure, the pixel-level accuracy enforced by Adversarial Loss and L1 Loss (especially when stabilized by SN in the GAN model) appeared to be more effective in reconstructing the fixed geometric patterns of QR Codes, which are essential for successful scanning. The baseline GAN provided a good balance between quality and speed, achieving an SSIM of 0.87 and a scanning success rate of 75%, taking 0.30 seconds per image. However, it sometimes faced training stability issues. The GAN + Spectral Normalization (GAN+SN), which was developed in this study, achieved the highest performance with an SSIM of 0.91 and a scanning success rate of 88%. It was able to handle damage up to 40–60%, although it required the longest processing time of 0.35 seconds per image. This illustrated the trade-off between speed and robustness to damage. GAN+SN was a more suitable choice when restoring heavily damaged QR Codes, providing better stability and reducing mode collapse. The choice of technique should depend on specific requirements: if high speed is prioritized, CNN (U-Net) is recommended for minor damage; for a good balance between quality and speed, the baseline GAN is appropriate; and if maximum performance is desired without much concern for processing time, GAN + Spectral Normalization provides the best results in terms of both accuracy and robustness to severe damage.

TABLE VI. COMPARISON OF PERFORMANCE WITH OTHER STUDIES BY CRITERIA

Criteria	GAN+SN	EHFP-GAN	SRGAN
Main Objective	Reconstruct missing regions	Reconstruct missing regions	Enhance image details
Performance (Success Rate)	Minor Damage (1–20%): 95% Moderate Damage	Mild Damage 95.35% [11]	Not directly evaluated [12]
Key Advantage	Resistant to severe Damage	Performs best with mild Damage	Most suitable for distorted QR Code
Main Limitation	Longer processing time compared to other methods	Performance significantly decreases with heavy Damage	Cannot recover information that is completely missing

From Table VI, when compared with the study by Zheng et al. (2023), which proposed the Edge-Enhanced Hierarchical Feature Pyramid GAN (EHFP-GAN), a two-stage architecture that focused on repairing edges first and then reconstructing the QR Code, it was observed that the EHFP-GAN approach was more complex [11]. In contrast, our study used a single-stage GAN architecture, but emphasized stabilizing the training

process by applying Spectral Normalization (SN) directly to the Generator layers. Our approach focused on improving the overall image reconstruction quality and the sharpness of the black-and-white structure, which is crucial for QR Codes, rather than separating the edge repair process.

In terms of performance, our GAN+SN model achieved an average scanning success rate of 88% for up to 60% damage, whereas the EHFP-GAN model reported a recognition rate of 95.35% for minor damage and dropped to 31.94% for severe damage [11]. Although EHFP-GAN achieved higher success rates on minor damage, our model demonstrated better robustness when facing high levels of damage (40 to 60%), which represented a particularly challenging scenario. Furthermore, compared with studies using Super-Resolution techniques such as SRGAN [12], which achieved a PSNR of 30.06 and SSIM of 0.936 for QR Code upscaling, our model, focusing on inpainting missing parts, still provided competitive results (PSNR 28.5, SSIM 0.91) and, importantly, verified effectiveness through the scanning success rate, the primary practical metric.

A key strength of our study was the use of Spectral Normalization, which reduced unstable training and mitigated Mode Collapse commonly encountered in GAN training. This allowed the model to better learn the distribution of complete QR Code data, resulting in improved reconstruction of fine details and sharp black-and-white modules. In contrast, more complex approaches like EHFP-GAN may require additional data and fine-tuning to achieve maximum performance. However, a limitation of our study was that processing time could be slightly longer than the CNN (U-Net) method (0.35 seconds compared to 0.15 seconds per image), although the improved accuracy was more significant for practical applications.

Although the model performed well on synthetic data, its effectiveness on real-world damage, such as stains, sun fading, or perspective distortions, remained a challenge that required further investigation.

VI. CONCLUSION

This study presented an approach for restoring damaged QR Codes using Generative Adversarial Networks (GANs) combined with Spectral Normalization (SN) to enhance training stability. Experiments on a dataset with simulated damage showed that the developed model could generate restored QR Codes with high sharpness and strong similarity to the original QR Codes, achieving an average PSNR of 28.5 dB and SSIM of 0.91. Moreover, the model significantly improved the scanning success rate of damaged QR Codes, reaching 88%, which outperformed the CNNs (U-Net) and basic GANs methods under the same experimental conditions.

The use of Spectral Normalization in the Generator enabled the model to effectively reconstruct complex details in the restored regions, reduce artifacts, and allowed the model to better learn the distribution of complete QR Code data. Although the proposed model required slightly longer processing time than other methods, the improved accuracy

was highly significant for practical applications, particularly in scenarios where QR Codes were severely damaged.

For limitations and future work, this study had certain limitations that should be considered. The dataset used for training consisted of synthetic data, which may not cover the full variety of real-world damage. The model was also unable to handle complex damage, such as QR Code distortions or damage caused by reflections.

Overall, this study proved a promising approach for restoring damaged QR Codes using GANs combined with Spectral Normalization, which could be applied across various industries and scenarios where QR Codes play a critical role.

REFERENCES

- [1] H. Kato and K. T. Tan, "Pervasive 2D Barcodes for Camera Phone Applications," *IEEE Pervasive Computing*, vol. 6, no. 4, pp. 76-85, Dec. 2007, doi: 10.1109/MPRV.2007.80.
- [2] L. Belussi and N. S. T. Hirata, "Fast QR Code Detection in Arbitrarily Acquired Images," in *Proc. 24th SIBGRAPI Conf. on Graphics, Patterns and Images (SIBGRAPI)*, Maceió, Brazil, Aug. 2011, pp. 281-288, doi: 10.1109/SIBGRAPI.2011.16.
- [3] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image Inpainting for Irregular Holes Using Partial Convolutions," in *Computer Vision – ECCV 2018, Lecture Notes in Computer Science*, vol. 11215, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer, 2018, pp. 89-105, doi: 10.1007/978-3-030-01252-6_6.
- [4] ISO/IEC 18004:2024, Information technology — Automatic identification and data capture techniques — QR code bar code symbology specification, 4th ed., Aug. 2024. Geneva, Switzerland: International Organization for Standardization, 2024. [Online]. Available: <https://standards.iteh.ai>.
- [5] S. Tiwari, "An Introduction to QR Code Technology," in *Proc. 2016 Int. Conf. on Information Technology (ICIT)*, Bhubaneswar, India, Dec. 2016, pp. 39-44, doi: 10.1109/ICIT.2016.021.
- [6] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative Image Inpainting with Contextual Attention," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 5505-5514, doi: 10.1109/CVPR.2018.00577.
- [7] Y. Yu, F. Zhan, and S. Lu, "EdgeConnect: Generative image inpainting with adversarial edge learning," *IEEE Trans. Image Process.*, vol. 30, pp. 2322-2332, 2021, doi: 10.1109/TIP.2021.3050689.
- [8] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral Normalization for Generative Adversarial Networks," in *Proc. Int. Conf. on Learning Representations (ICLR)*, Vancouver, BC, Canada, Apr. 2018, arXiv:1802.05957, doi: 10.48550/arXiv.1802.05957.
- [9] K. Song, N. Liu, Z. Gao, J. Zhang, G. Zhai and X. -P. Zhang, "Deep Restoration of Invisible QR Code from TPVM Display," *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, London, UK, 2020, pp. 1-6, doi: 10.1109/ICMEW46912.2020.9105961.
- [10] Q. Ma, R. Chen, and W. Wang, "QR code images fast blind deblurring based on specific symbol prior and L2 norm minimization," in *Proc. 2023 4th Int. Conf. on Computer Engineering and Application (ICCEA)*, Guangzhou, China, Apr. 2023, pp. 89-94, doi: 10.1109/ICCEA58433.2023.10135499.
- [11] J. Zheng, R. Zhao, Z. Lin, S. Liu, R. Zhu, Z. Zhang, Y. Fu, and J. Lu, "EHFP-GAN: Edge-Enhanced Hierarchical Feature Pyramid Network for Damaged QR Code Reconstruction," *Mathematics*, vol. 11, no. 20, p. 4349, Oct. 2023, doi: 10.3390/math11204349.
- [12] Y. Sancar, "Reconstructing unreadable QR codes: a deep learning based super resolution strategy," *PeerJ Computer Science*, vol. 2025, Art. no. e2841, Apr. 2025, doi: 10.7717/peerj-cs.2841.
- [13] Mateev, Valentin, and Iliana Marinova. "Depth field reconstruction by shadow tomography for small digital microscope." *AIP Conference Proceedings*. Vol. 2505. No. 1. AIP Publishing LLC, 2022.