2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

RINQC: A Robust Invisible Network for Quick Response Code

Chaoen Xiao[1], Ruiling Luo[1], Lei Zhang[1], Jianxin[1] Wang and Duo Zhang[1]

[1] Beijing Electronic Science and Technology Institute, Beijing 100070, China
xce@besti.edu.cn

Abstract. The widespread application of Quick Response (QR) Code urgently necessitates enhancing their traceability and anti-counterfeiting capabilities. However, traditional QR code-based watermark protection technology is susceptible to interference during cross-media transmission. To address the above mentioned issue, this paper proposes a QR image watermark algorithm based on the UNet++ network. First, leveraging the multi-angle and high-speed recognition characteristics of QR codes, targeted improvements are made to the network structure and training process, with dilated convolutions incorporated into the encoder to enhance detail precision. Then, a refined local discrimination is achieved through the integration of PatchGAN, continuously optimizing the watermark embedding method to improve the imperceptibility of the watermark. Finally, a distortion network mechanism is introduced during the training process to simulate the environment of capturing QR codes from different angles, thereby enhancing the robustness of the images. Experiments demonstrate that the proposed method achieved PSNR and SSIM values of 36.27 dB and 0.978 respectively, with better robustness and imperceptibility.

Keywords: Quick Response Code, Image Watermark, UNet++ Network, Deep Learning.

1 Introduction

Quick Response (QR) code images [1] possess several remarkable features as carriers for transmitting secret messages, including their strong error tolerance, fast recognition, and wide range of delivery channels [2]. These characteristics make QR code images ideal for message hiding and covert communication [3]. However, traditional QR codes also face a significant challenge, the information embedded in QR codes is in plaintext, which poses a security risk to a certain extent due to the openness of its encoding method, which is not inherently forgery-proof and is susceptible to the risk of tampering and information theft by attackers. Therefore, ensuring the security and reliability of QR codes has become an urgent problem to be solved at present [4].

In addition to robustness and imperceptibility, QR code-based information hiding also needs to have the function of not affecting the recognition of QR code [5]. A watermark algorithm that can be effectively transmitted across media is a difficult point in information hiding technology because it has very stringent requirements on

robustness. In 2011, Vongpradhip [6] proposed that QR Code is embedded with an invisible watermark by using DCT for an information hiding within the group through the QR Code image with invisible watermark. In 2020, Tancik [7] converted the information into hyperlinks, and embedded the hyperlinks in the image using the UNet [8] network, and through the screen or printing to extract the information, the method is applied on color images with better results. Jia [9] designed a new distortion network based on Tancik to simulate the process of camera imaging, the distortion network adopts the idea of three-dimensional rendering, which is more flexible than the two-dimensional perspective distortion in Tancik. In 2023, Fang [10] proposed a DCT based watermarking method, the method uses Scale Invariant Feature Transform to defend against perspective distortions introduced by the camera. In 2024, Du [11] proposed an improved hybrid watermark optimization scheme,which took advantage of the multi-level wavelet transform to embed watermark information in the fractional Fourier transform domain by modifying the singular values of the image.

The existing watermarking techniques face challenges when dealing with the cross-media transmission of QR codes embedded with watermarks, and robustness needs to be further improved to better adapt to variable transmission environments. To address the above mentioned issues, the major contributions of this work are as follows:

- The high-speed multi-angle recognition feature of the QR code is taken advantage of, and targeted improvements are made to the UNet++ [12] network by incorporating dilated convolutions into the encoder in order to capture richer image information.
- To improve the imperceptibility of the watermarks, we combined with PatchGAN [13], which can perform fine local discrimination and continuously optimize watermark embedding.
- To improve the robustness of the image, a distortion network mechanism is introduced during the training process to simulate the environment of shooting QR codes at different angles.

2 Proposed method

2.1 Network Architecture

The QR image watermark traceability scheme is shown in Fig. 1, which contains 3 steps of encoder, decoder distortion network and discriminators. The first is to preprocess the watermark, that is, to generate a watermark code based on the watermark information, and use the UNet++ network with added dilated convolution [14] to combine the watermark information with the QR image to generate a watermarked image. Then the next step is to put the image embedded with the watermark in the distortion network to simulate the noise caused by shooting. Finally, the image enters the decoder after cross-media transmission, and the decoder decodes the noisy image to extract the watermark information in the QR code image.

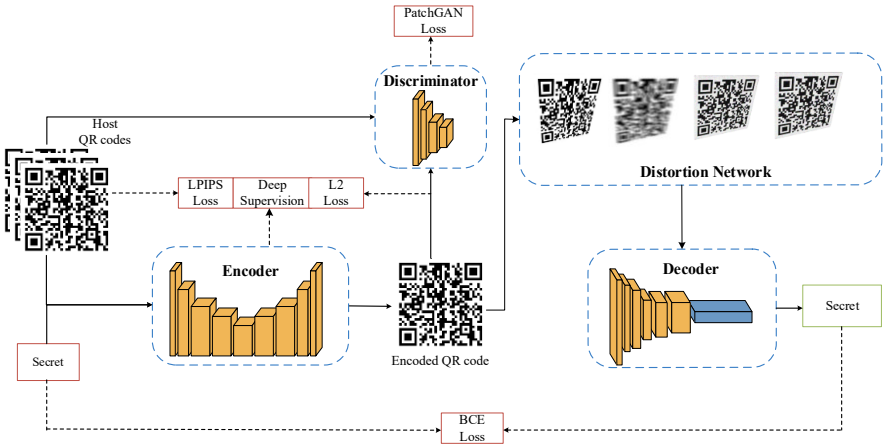

Fig. 1. Embedding and extraction system.

The goal of the encoder is to reconstruct a new image with an original image and a hidden message, ensuring good quality for the reconstructed image. The input of the encoder is a QR code image with embedded watermark information represented as a binary string of N bits. Each neuron in the previous layer is connected through a fully connected layer, and then up sampled to form a 400 × 400 tensor of the same size as the input image. The watermark information image is entered into $X^{1,1}$, as shown in the figure. In the down sampling process, $X^{4,1}$ and $X^{5,1}$ are hollow convolutions. Hollow convolutions increase the receptive field to obtain multiscale pixel information without increasing the amount of calculation and the number of parameters, so as to avoid repeated pooling operations that cause the loss of image spatial information. This paper introduces deep supervision, and adds a 1×1 convolution layer after each output $X^{1,2}$, $X^{1,3}$, $X^{1,4}$, and $X^{1,5}$ of the first layer. The QR image otained by each deep supervision subnetwork is compared with the original QR image to obtain the loss error function and then back-propagated, which alleviates the problem of gradient disappearance while accelerating the convergence speed.

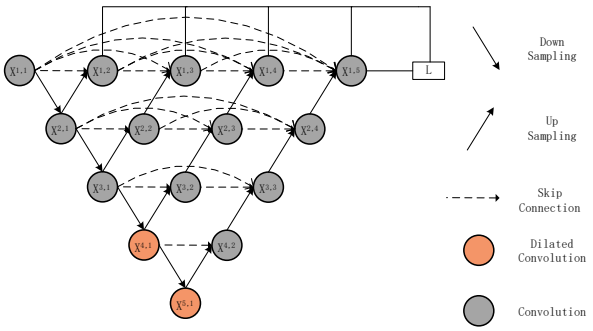

Fig. 2. Watermark embedding network.

The decoder receives the encoded image under distortion attack and is trained to retrieve the hidden string. The entire decoding process consists of several key modules, namely input, Spatial Transformer Network, convolutional network, flatten operation Flatten, Dense, and the final decoded output. Among them, STN is a neural network structure that can learn to perform spatial transformation on input data. It introduces a spatial transformation module that can adaptively perform geometric transformations on the input, such as rotation, scaling, and translation, thereby improving the model's ability to process input data. The main function of the convolutional network is to extract high level features of the image. The black and white modules of the QR code have obvious structured patterns. The convolutional layer can extract these features layer by layer, from local edge information to global encoding patterns, to form a high-dimensional feature representation. The multi-layer design of the convolutional network enables the model to capture important information in the QR code and provide support for subsequent decoding. The output of the convolutional network is a multi-dimensional feature map, which needs to be converted into a one-dimensional vector through a flattening operation to facilitate input into the fully connected layer. The fully connected layer is responsible for mapping the extracted high-dimensional features to the specific representation of the hidden information. It associates the features in the QR code with its corresponding hidden information by learning nonlinear combinations, and finally decodes the target information. Finally, the decoder outputs the hidden watermark information.

GAN discriminators focus on the global image and only output a discrimination result value, thus ignoring the local part of the image. Therefore, PatchGAN is introduced to evaluate the local area, which can better capture the details and local information of the image.

2.2 Distortion Network

We establish a coordinate space to simplify the shooting of QR code perspective transformation, QR code rotation of the rotation matrix by the three basic rotation matrix composite, first of all, to generate the three matrices R_x, R_y, R_z and the three Euler angles θ_x, θ_y, θ_z, respectively, around the x, y, z axis of the rotation matrix, $R = R_x R_y R_z$ for the three-dimensional rotation matrix.

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) & 0 \\ 0 & \sin(\theta_x) & \cos(\theta_x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$R_y = \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$R_z = \begin{bmatrix} \cos(\theta_z) & \sin(\theta_z) & 0 & 0 \\ -\sin(\theta_z) & \cos(\theta_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$\theta_i = 21$ is the semi-viewable angle at which exactly the whole image can be displayed, the distance d between the lens and the image is:

$$d = \frac{\sqrt{w^2 + h^2}}{2 \tan(\theta_i)} \tag{4}$$

R_1, R_2 are compensation matrices that enable projection of the image to the imaging plane:

$$R_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{w}{2} & -\frac{h}{2} & -d & 1 \end{bmatrix} \tag{5}$$

$$R_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{w}{2} & \frac{h}{2} & d & 1 \end{bmatrix} \tag{6}$$

The new coordinates of the image are $[x', y', z', 1]^T = [x, y, z, 1] R_1 R_2 R_3$, and then the three-dimensional rotation matrix is transformed to the plane, and by deleting the z-axis, we can get u and v:

$$u = \frac{f_x x'}{z'} + \frac{\omega}{2}\left(1 - \frac{f_x}{z'}\right) \tag{7}$$

$$v = \frac{f_y y'}{z'} + \frac{h}{2}\left(1 - \frac{f_y}{z'}\right) \tag{8}$$

f_x and f_y are the focal lengths in the horizontal and vertical directions. The coordinates of the new four corners of the image can be calculated using a perspective matrix, which can simulate the perspective transformation in a shooting situation.

As shown in Fig 3, inaccuracies in the photographic process may lead to image blurring, and a variety of noises are introduced during the imaging process. Firstly, in order

to simulate out of focus, a Gaussian blur kernel [15] is introduced in this paper and the standard deviation is chosen to be sampled randomly between 1-3 pixels. Secondly, to simulate motion blur, a random angle is sampled during model training and a linear blur kernel with a width between 3-7 pixels is generated. In addition to that, in this paper, Gaussian noise is used to add noise perturbation to the QR code image by adding a noise tensor to the target tensor that conforms to a Gaussian distribution. The Gaussian perturbation range used during model training is $\sigma \sim U[0, 0.02]$. Finally, the JPEG [16] quality was randomly sampled in the range [30,100] during model training to resist the loss of the image by the camera shot when the model was used.

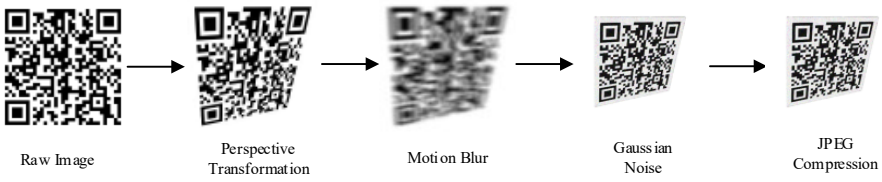

Fig. 3. Training process.

2.3 Loss Function

To evaluate the difference between the original image and the watermarked image in the image, a loss function is introduced to evaluate the difference between the two images as well as the difference between the encoded and decoded information. The loss functions include L_2 loss and LPIPS loss, which are used to effectively evaluate image quality and information fidelity. The L_2 norm loss function minimizes the sum of squares of the difference between the estimated value $y_{\text{true}}^{(i)}$ and the target value $y_{\text{pred}}^{(i)}$, and regularizes the L_2 residual to obtain L_R as the edge loss function of the image.

$$\mathrm{L}_2 \ \mathrm{Loss} = \frac{1}{N}\sum_{i=1}^{N}(y_{\text{pred}}^{(i)} - y_{\text{true}}^{(i)})^2 \tag{9}$$

LPIPS loss [17] is a perceptual loss function based on deep features. In LPIPS loss, features are first extracted in the image layer and these features are unit-normalized in the channel dimension. Then, the spatial average of these features is calculated and summed by channel. Through LPIPS loss, the model can pay more attention to the image quality perceived by the human eye, rather than just the pixel-level differences, thereby improving the perceptual quality of the watermarked image.

L_2 loss produces blurry results on image generation problems. Although these losses fail to encourage high frequency crispness, in many cases they nonetheless accurately capture the low frequencies. The changes in the QR code are mainly concentrated in the high frequency parts of its black and white areas, so we borrowed the design idea of PatchGAN and built a discriminator architecture targeting these high-frequency changes. This discriminator tries to classify if each N × N patch in an image is real or

fake. We run this discriminator convolutationally across the image, averaging all responses to provide the ultimate output of L_D.

$$L_D = -E_{x\sim p_{\text{data}}(x)}[\log D(x)] - E_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{10}$$

The evaluation of the loss function is not only used to measure the difference between images, but also covers the extraction and recovery process of watermark information. In this process, the Cross Entropy Information loss is used to quantify the difference between the extracted watermark information and the original watermark information, thereby providing guidance for the accuracy of the decoder in the watermark recovery stage. The Cross Entropy loss ensures that the watermark information can be accurately extracted by evaluating the deviation between the predicted value and the true label. The Cross Entropy Information loss L_M is used to represent the difference between the extracted watermark information and the original watermark information to ensure the accuracy of the recovery process.

$$L_M = -\sum_{i=1}^{N} y_i \log(y_i) \tag{11}$$

The total loss during training is divided into four parts: L_2 loss(L_R), LPIPS(L_P), message reconstruction loss for decoder training: Cross Entropy loss(L_M) and L_D loss. The joint loss function of our approach is defined as

$$L = \sum_{i=R,P,D,M} \lambda_i L_i \tag{12}$$

When training the network, the loss weights λ_R, λ_P, and λ_D are initially set to zero and then increase linearly so that the decoder training can achieve a higher accuracy. λ_M is set to 1, allowing 1 bit of information loss during model training to prevent overfitting.

3 Experiment and Discussion

3.1 Dataset and Experimental Setting

In this experiment of generating QR codes, Java language is used, modified on the basis of ZXing [18] open source library, the error correction level of QR codes is set to Error Correction Level. H, both width and height are 400, Version is 7. A total of 45000 black and white QR images are generated for training and 5000 images are used for testing.

The watermark contains 100 bits of information, which is sufficient for the traceability scenario of QR codes. This amount of information is enough to meet the needs of the application while minimizing the number of information bits, thereby reducing the burden on data transmission.

The image size is set as 400. In phase 1, we train the model with a batch size of 16 and a total step of 20000. The learning rate is set as $1e^{-4}$ with linear schedule. In phase 2, the learning rate is set as 1 and is fixed as constant. We train with a batch size of 16, a total step of 30000, and negative samples of 16. The experiments are conducted on a single RTX 3090 GPU.

3.2 Visual Quality Evaluation

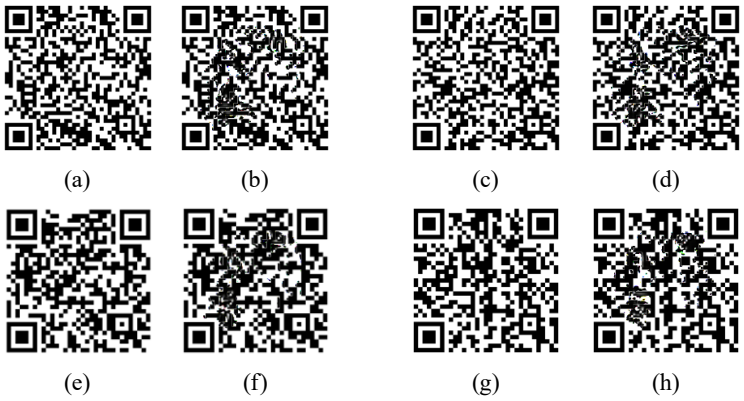

Fig. 4. Comparison of original images and watermark images.

Fig. 4 provides a visual comparison of the watermark and original images. We compared the performance of our model with several widely used networks, including UNet++ and UNet. The average values of SSIM, Bit Accuracy, and PSNR over the test set are summarized in Table 1. The results demonstrate a significant improvement in image quality and watermark extraction performance, confirming the effectiveness of our proposed network.

Table 1. Comparison with UNet and UNet++ model.

model	SSIM	PSNR	Bit Accuracy
UNet	0.931	0.90	24.15
UNet++	0.958	0.95	30.12
Ours	**0.977**	**1**	**36.27**

We compare with the existing advanced methods Tancik [7], Jia [9], and Fang [10]. As shown in Table 2, the PSNR value of our method can reach 36.27dB, indicating that our method is superior to the other three methods in the visual quality of QR code images.

3.3 Information Capacity on Visual Quality

To evaluate the impact of information capacity on visual quality, we retrained the model to encode 50, 100 and 200 bits of information. Table 3 shows the image quality and robustness test results at different capacities. By comparing the image performance under different capacities, we find that the image quality and robustness have a significant inverse relationship with the watermark embedding capacity. Specifically, as information capacity increased, image quality decreased.

Table 2. Comparison of image visual quality.

Metric	Tancik[7]	Jia[9]	Fang[10]	Ours
SSIM	0.922	0.936	0.977	**0.978**
PSNR	28.30	28.60	35.50	**36.27**

Table 3. Different inform capacities.

Metric	50bits	100bits	200bits
SSIM	0.978	0.977	0.963
PSNR	36.84	36.27	35.31
Bit Accuracy	1	1	0.95

To find an optimal balance between image quality, robustness and information capacity, we comprehensively considered the relative impact of the two. Through testing, we found that 100 bits of information capacity can maintain good robustness while ensuring the visual quality of the QR code, especially in the actual application of QR code traceability, which is enough to carry the required amount of information. Therefore, we chose 100 bits as the ideal capacity. This capacity can fully meet the needs for the practical application of QR codes, especially in the field of traceability and information tracking, without significantly affecting the recognition accuracy of the QR code and user experience.

3.4 Robustness Evaluation

When QR codes are transmitted in multiple media, various noises are inevitably introduced, such as shooting angle, camera sensor error, printing distortion, which can lead to the degradation of image quality, thus affecting the recovery of image information and the extraction of watermarking information. We will simulate different angles to shoot the QR code after adding the watermark and compare it with the previous methods.

We use 3D rotation matrix to restore the perspective rotated QR code when the image will inevitably lose part of the information, the larger the angle of rotation of the QR code image, the more information is lost, the image is restored, the left and right side of the image will appear blank image area, as shown in Fig. 5.

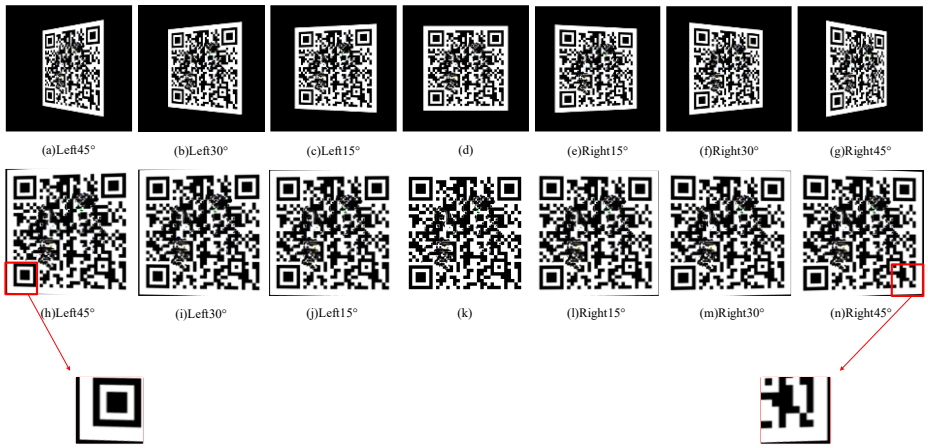

Fig. 5. Simulated and corrected images taken from different angles. (d) is the simulated frontal shot of the QR code image, (a) (b) (c) is the image of the QR code rotated to the left by 45°, 30°, 15° respectively, (e) (f) (g) is the image of the QR code rotated to the right by 15°, 30°, 45° respectively.

Table 4. Accuracy of watermark image recovery from different angles.

Model	Left45	Left30	Left15	0	Right15	Right30	Right45
Tancik[7]	97.3%	99.3%	99.5%	100%	98.4%	97.8%	94.4%
Fang[10]	98.9%	98.3%	96.6%	97.8%	98.4%	94.0%	94.8%
Jia[9]	**99.8%**	99.3%	99.5%	100%	99.3%	99.3%	98.7%
Ours	98.9%	**99.5%**	**99.8%**	**100%**	**99.8%**	**99.4%**	**98.8%**

Table 5. Ablation study

Methods	Network	Dilated convolution	Patch GAN	Distortion network	PSNR	SSIM
1	UNet	×	×	×	24.15	0.931
2	UNet++	×	×	×	30.12	0.938
3	UNet++	×	√	√	35.81	0.972
4	UNet++	√	√	×	30.32	0.939
5	UNet++	√	×	√	33.18	0.966
6	UNet++	√	√	√	**36.27**	**0.977**

As can be seen from Table 4, the accuracy of extracting watermark information changes at different angles, and as the angle increases, the accuracy of recovering the watermark information in the watermarked image becomes lower, and the watermark quality of the QR image generated by Tancik is poorer, the watermark in the white part of the QR code is more obvious, and the watermark information in the image can not be restored better when the QR code image is rotated to 45 degrees, and there exists a portion of the information Loss, our proposed method can better recover the watermark

information in all angles, and the accuracy of extraction is close to 99%, which is a good improvement for the robustness of QR code image.

3.5 Ablation Study

We conduct several ablation studies to analyze the performance of our proposed modules and strategies.

We remove the proposed dilated convolution, patchGAN discriminator and distortion network from the model respectively. Table 5 shows that the performance of the model decreases after removing all three modules, indicating that these modules improve the robustness of the model and the accuracy of watermark extraction.

4 Conclusion

In this work, we propose a QR code image watermark algorithm to hide secret information in QR images. To better adapt to the high-density encoding characteristics of QR codes, PatchGAN is added as a discriminator, so that the image loss and the loss of encoded watermark information between the image and the watermarked image are minimized. Based on the impact of transmitting and shooting QR images, a distortion network is introduced to simulate the environment of shooting QR codes from different angles. Experimental results demonstrate the proposed method achieved PSNR and SSIM values of 36.27 dB and 0.978 respectively, with better robustness and imperceptibility. Future work can expand our image watermarking approach to other types of cover signals, such as videos and point clouds.

Acknowledgments. This work is supported by the Fundamental Research Funds for the Central Universities(3282024009,20230051Z0114,20230050Z0114).

References

1. Panah, Arezou Soltani, et al.: On the properties of non-media digital watermarking: a review of state of the art techniques. IEEE Access 4, 2670-2704 (2016)
2. Tiwari, Sumit.: An introduction to QR code technology. 2016 international conference on information technology (ICIT). IEEE, 39-44 (2016)
3. Liu, Tao, Bin Yan, and Jeng-Shyang Pan.: Color visual secret sharing for QR code with perfect module reconstruction. Applied Sciences 9.21, 4670 (2019)
4. Kieseberg, Peter, et al.: QR code security. Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia, 430-435 (2010)
5. Pan, Jeng-Shyang, et al.: Digital watermarking with improved SMS applied for QR code. Engineering Applications of Artificial Intelligence 97, 104049 (2021)
6. Vongpradhip, Sartid, and Suppat Rungraungsilp.: QR code using invisible watermarking in frequency domain. 2011 Ninth International Conference on ICT and Knowledge Engineering. IEEE, 47-52 (2012)

7. Tancik, Matthew, Ben Mildenhall, and Ren Ng.: Stegastamp: Invisible hyperlinks in physical photographs. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2117-2126 (2020)
8. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox.: U-net: Convolutional networks for biomedical image segmentation. Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer international publishing, 234-241 (2015)
9. Jia, Jun, et al.: RIHOOP: Robust invisible hyperlinks in offline and online photographs. IEEE Transactions on Cybernetics, 7094-7106 (2020)
10. Fang, Han, et al.: Screen-shooting resilient watermarking. IEEE Transactions on Information Forensics and Security , 1403-1418 (2018)
11. Du, Qiaoqiao, et al. "FRFT Domain Watermarking Algorithm Based on GA Adaptive Optimization." International Conference on Intelligent Computing. Singapore: Springer Nature Singapore, 147-159 (2024)
12. Zhou, Zongwei, et al.: Unet++: A nested u-net architecture for medical image segmentation. Deep learning in medical image analysis and multimodal learning for clinical decision support: 4th international workshop, DLMIA 2018, and 8th international workshop, ML-CDS 2018, held in conjunction with MICCAI 2018, Granada, Spain, September 20, proceedings 4. Springer International Publishing, 3-11 (2018)
13. Isola, Phillip, et al.: Image-to-image translation with conditional adversarial networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 1125-1134 (2017)
14. Yu, Fisher, and Vladlen Koltun.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015).
15. Hummel, Robert A., B. Kimia, and Steven W. Zucker.: Deblurring gaussian blur. Computer Vision, Graphics, and Image Processing 38.1, 66-80 (1987)
16. Shin, Richard, and Dawn Song.: Jpeg-resistant adversarial images. NIPS 2017 workshop on machine learning and computer security, 8 (2017)
17. Talebi, Hossein, and Peyman Milanfar.: Learned perceptual image enhancement. 2018 IEEE international conference on computational photography (ICCP), 1-13 (2018)
18. Scheuermann, Constantin, et al.: Evaluation of barcode decoding performance using zxing library. Proceedings of the second workshop on smart mobile applications, SmartApps. (2012)