Comparative Analysis of CryptoGAN: Evaluating Quality Metrics and Security in GAN-based Image Encryption

Ranjith Bhat [1*], Raghu Nanjundegowda [2]
[1] Research Scholar, Faculty of Engineering and Technology, JAIN (Deemed to be University), Bengaluru, India
[1] Assistant Professor, Department of Robotics and AI Engineering, NMAM Institute of Technology, NITTE (Deemed to be University), Nitte, India
[2] Associate Professor, Department of Electrical and Electronics Engineering, JAIN (Deemed to be University), Bengaluru, India
Email: [1] ranjithbhat@gmail.com, [2] raghu1987n@gmail.com
*Corresponding Author

***Abstract*—Balancing security with image quality is a critical challenge in image encryption, particularly for applications like medical imaging that require high visual fidelity. Traditional encryption methods often fail to preserve image integrity and are vulnerable to advanced attacks. This paper introduces CryptoGAN, a novel GAN-based model designed for image encryption. CryptoGAN employs an architecture to effectively encrypt a dataset of 2000 butterfly images with a resolution of 256x256 pixels, integrating Generative Adversarial Networks (GANs) with symmetric key encryption. Using a U-Net Generator and a PatchGAN Discriminator, CryptoGAN optimizes for key metrics including Structural Similarity Index (SSIM), entropy, and correlation measures. CryptoGAN's performance is comprehensively compared against state-of-the-art models such as Cycle GAN-based Image Steganography, EncryptGAN, and DeepEDN. Our evaluation, based on metrics like SSIM, entropy, and PSNR, demonstrates CryptoGAN's superior ability to enhance encryption robustness while maintaining high image quality. Extensive experimental results confirm that CryptoGAN effectively balances security and visual fidelity, making it a promising solution for secure image transmission and storage. This study is supported by a literature survey and detailed analysis of the model architecture, underscoring CryptoGAN's significant contributions to the field of image encryption.**

Keywords*—Artificial Intelligence (AI); Generative Adversarial Networks (GAN); GAN-based Encryption; Image Encryption; Deep Neural Networks (DNN).

I. Introduction

In recent years, the transmission and sharing of digital images over the internet have significantly increased, presenting substantial security challenges. Traditional image encryption methods, such as DES (Data Encryption Standard) and AES (Advanced Encryption Standard), often compromise the structural integrity and visual fidelity of images, which is critical for applications like medical imaging. These methods also suffer from high computational complexity and are vulnerable to various attacks, necessitating innovative solutions to overcome these limitations. The proliferation of machine learning techniques across multiple areas has resulted in the need for vast amounts of data for training purposes, posing a major challenge to the preservation of individual privacy rights [1]. Digital images feature strong adjacent pixel correlation, redundant data, and enormous data sizes [2]. The widespread distribution of multimedia content on multiple platforms calls for strong security measures to protect confidential data. Traditional image encryption methods frequently fall short in balancing the conflicting needs of efficiency, security, and image quality preservation. This has prompted researchers to investigate novel strategies that improve image encryption techniques by utilizing developments in artificial intelligence, especially deep learning. Fig. 1, illustrates a typical data encryption procedure that uses the same key for both encryption and decryption where the plain text is the input image that is encrypted using any encryption algorithm that uses a key. Here the output is the cipher image or the encrypted image. The decryption is the exact opposite algorithm as the encryption to reconstruct the original data back.

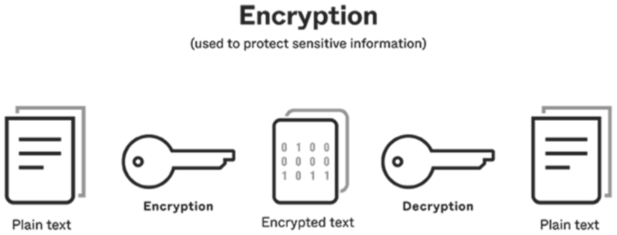


Fig. 1. Encryption and Decryption of data using a symmetric key

The simple technique of symmetric key encryption is adopted in order to safeguard an individual's privacy on public network platforms [3] using the same key on both ends. Several methods, including image encryption, image stenography (image concealing), and image authentication, guarantee the security of data contained in the images. Deep learning has been increasingly important in recent times for tasks including object recognition in images, image classification, segmentation, style transfer, reconstruction, and compression. Recently, deep learning-based image security has drawn the interest of researchers and made significant advancements. Because of its features in cryptography, chaos is the foundation of traditional image encryption systems. The authors in [4] first presented a chaotic encryption technique in 1989.

Numerous efficient designs for imagine encryption systems based on chaos have been put forth [5], [6]. Subsequently, more encryption systems based on wavelet transform [7], game theory [8], chaos [9] and DNA coding [10], etc., have been devised. Schemes for encrypting images include permutation and diffusion rounds. Random pixel configurations during the permutation step improve defense against statistical attacks, as no information is obtained from mosaics of the permuted image through this method. Secret keys are used to change the pixel values during the diffusion phase [11]. When it comes to encrypting images, deep learning is just starting out. What triggers interest for an additional study is the fact that its integration with cryptography has not been investigated yet. As a branch of machine learning, deep learning has enabled the creation of models capable of learning and making judgements from massive volumes of data, which has had a profound impact on many domains. A potent tool for creating and modifying images, Generative Adversarial Networks (GANs) have arisen within this field. The GAN's Generator and Discriminator with the corresponding inputs and outputs can be seen in Fig. 2.

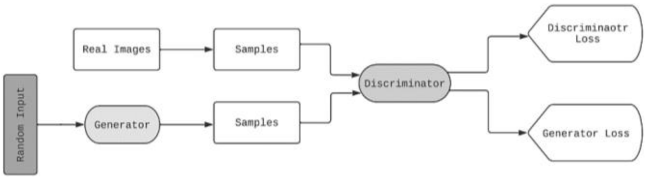


Fig. 2. Generative Adversarial Network Architecture [10]

Concurrently trained using adversarial processes, GANs comprise of two neural networks viz. Generator and a Discriminator. Producing realistic images is the goal of the Generator, while differentiating between genuine and created images is the goal of the Discriminator. Random noise is given as the input in the beginning to the Generator. As the training progresses in comparison with real images the Generator is trained better. Images of a high quality and realism are produced as a consequence of this dynamic interaction.

Beyond their original use in image generation, GANs have recently discovered some interesting new uses, such as in image encryption. Because of their one-of-a-kind design, GANs can learn intricate mappings and transformations, making them a good fit for creating advanced encryption algorithms. This study traces the history of deep learning techniques used for image encryption. We have also weighed the benefits and drawbacks of their evolution over the last several years.

In particular, traditional encryption methods like DES and AES compromise the structural integrity and visual fidelity of images, which is critical for applications such as medical imaging. These methods also suffer from high computational complexity and vulnerability to various attacks, necessitating innovative solutions to overcome these limitations. Chaotic encryption methods, while providing better security, often result in high computational complexity and are not easily scalable. Similarly, wavelet transform-based methods improve image quality but lack robustness against sophisticated attacks. These limitations highlight the need for new solutions that can effectively balance security, computational efficiency, and image quality.

This paper introduces CryptoGAN, a novel GAN-based model designed for image encryption. CryptoGAN employs a U-Net as the Generator and a PatchGAN as the Discriminator, specifically tailored to encrypt a dataset of 2000 butterfly images with a resolution of 256x256 pixels. The integration of GANs with symmetric key encryption in CryptoGAN addresses key limitations of traditional methods and existing GAN-based models. The following sections provide an analysis of the proposed work's architecture in drawing comparisons to previous works.

II. Performance Comparison of Different Image Encryption systems based on Deep Learning

Historically, methods like chaotic encryption and wavelet transforms have paved the way for modern encryption techniques. However, these methods underscore the ongoing need for solutions that balance security, computational efficiency, and image quality. This historical context illustrates the evolution of encryption methods and the challenges that persist today.

In the process of encryption, to ensure that the encoder and decoder principles are same, the majority of digital image steganography employs standard image processing methods [11], [12]. As an example, a multi-directional pixel value differencing and modulus function (MDPVDMF) was suggested by [11]. This function involves dividing the original image into multiple blocks with 2×2 pixels. The best direction with regard to embedding capacity and imperceptibility is determined by utilizing all three orientations of a 2×2-pixel block. To determine each block's embedding capacity [13], the pixel value differencing approach is employed [14]. To further improve imperceptibility, a pixel realignment approach based on the modulus function was then employed. Producing a meaningful stego-image becomes challenging with such a pixel-wise method. In fact, this is why we zero in on the deep learning approach that actually produces useful stego-images [15]. Table I shows the comparative study of different deep learning-based encryption models.

A. Deep Learning Approach-based Attacks on Encryption Schemes

For an optimal image encryption solution to be resistant against various attacks, it must be thoroughly examined [26], [27]. Key sensitivity, plaintext sensitivity, histogram analysis, correlation analysis, and entropy analysis are some of the most popular metrics used to test the security of image encryption schemes. These metrics measure resistance against brute force attacks, plaintext attacks, statistical attacks, differential attacks, and other types of attacks. Most encryption schemes remain susceptible to attacks, particularly plaintext attacks, even after being tested against these criteria. It is recommended to examine the core principles of the encryption technique from a cryptanalysis perspective. In addition to these more conventional forms of attack, deep learning-based pictorial encryption systems are vulnerable to the following types of attacks, which are typical of deep learning models:

- Leakage of Hidden Factors: An attacker attempts to create a deep learning model by utilising the photos used to construct the encryption network. It is possible to build the encryption/decryption network with hidden factors that expose the secret image.
- Leakage in the Network Architecture: The attacker has obtained the blueprints of the network, but they cannot access the concealed components. The perpetrator uses a variety of covert techniques to deduce the hidden image.
- Both Hidden Factors and Network Architecture Leakage: An attacker might potentially decipher the secret image from its cipher image if they have access to both the hidden factors that trained the encryption/decryption network and the encryption/decryption system.

TABLE I. COMPARATIVE STUDY ON DIFFERENT IMAGE ENCRYPTION SYSTEMS USING DEEP LEARNING

Ref.	Year	Techniques	Issues Addressed	Shortcomings
[16]	2019	Cycle GAN-based Image Steganography	Cover image-style visual communication	Necessary to use a larger cover image in order to insert the hidden image over it.
[17]	2020	Implementation of conventional encryption with secret key generation by means of a deep neural network	Generating fine-tuned dynamic keys based on an increase in security threats using a deep learning technique	Following key acquisition via DNN, encryption utilises a weak conventional diffusion mechanism
[18]	2020	Encrypting Images with Cycle GAN	Implementation of Cycle-GAN for Encryption	Measurements of Low Diffusion
[19]	2021	Encrypting images with cycle GANs and enhanced diffusion	Metrics for the spread of image encryption systems	The XOR operation is the sole component of traditional diffusion
[20]	2021	Visual encryption using imagine fusion, optical/digital diffusion, and image scrambling based on CNN	Advancements in convolutional neural network (CNN) image encryption	Encryption requires two images
[21]	2021	A deep learning-based image encryption technique with a dynamic key generating system	Using a deep learning technique to generate a dynamic secret key	The encryption system might use some tuning to make it more efficient
[22]	2022	Scrambling the discrete cosine transform (DCT) coefficient matrices to modulate the weights of DNN	It is non-linear and does not necessitate training, the encryption system is both efficient and secure	Both the histogram and the encryption method are vulnerable to occlusion attacks
[23]	2022	Using a Chaotic Sequence with a Deep Autoencoder to Encrypt Images	Creating a uniformly distributed cipher image by encoding a scrambled image using an auto-encoder	Traditional encryption techniques are more uniform than histogram [24], [25]

III. PROPOSED METHODOLOGY

The proposed CryptoGAN system is designed to encrypt a given plain image into a cipher image and decrypt it back to the plain image domain using a deep learning approach. The Generator G, also referred to as the encryption network, uses latent noise to generate cipher images. This network is trained using pairs of plain and encrypted images. We employ a U-Net architecture as the Generator and a PatchGAN as the Discriminator D, which has shown promising results in image translation tasks. The comparisons with other types of models are made further in this article in detail. During training, Generator G progressively improves through a feedback loop and loss minimization, thereby producing increasingly accurate cipher images. The losses involved in this process are crucial and are discussed in detail in the following sections. As we understand that the GANs typically learn using the loss functions, we intend to focus more on the losses available. The losses we refer here are explored in sections below. For this study, we utilized the butterfly dataset, consisting of colored images sized 256×256. The training process involved 2000 image samples. As shown in Fig. 3 Network G encrypts the plain image it gets, and in the opposite procedure [28], decryption network H creates a plain image from the cipher image. Similarly, CryptoGAN's encryption network G and Discriminator network D are trained to produce cipher images that resemble the target cipher image [29]; the goal of training the proposed decryption network H and Discriminator is to reconstruct plain images with minimal differences compared to the original plain image [30]. We utilized the previously trained Discriminator for reverse translation. Future research may explore alternative discriminators to optimize this process further. The suggested encryption method's flowchart in depicted in Fig. 3.

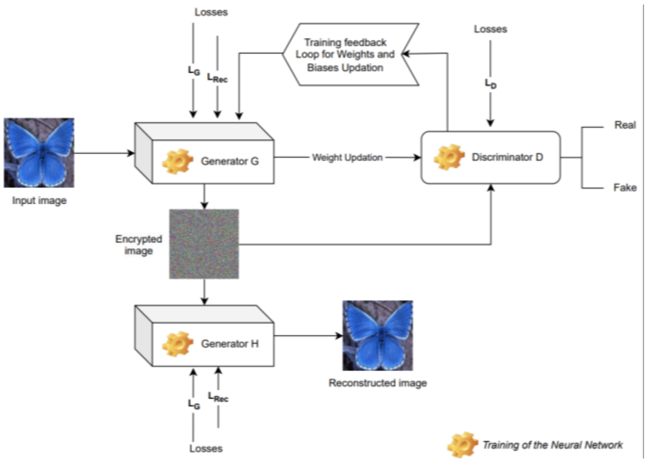


Fig. 3. Encryption flow diagram of the proposed model

GANs, with their ability to learn complex data distributions, offer a promising solution for image encryption by generating high-quality encrypted images that maintain structural integrity. The objectives of this research are to develop a robustly trained GAN-based image encryption model, evaluate its performance using comprehensive metrics, and compare it with existing state-of-the-art models. This study aims to demonstrate the potential of CryptoGAN in providing secure and high-quality image encryption solutions. For image-to-image translation tasks like segmentation, super-resolution, or encryption, U-Net as a

Generator usually trains with paired images. There is an input image and a cipher image, or matched target image, in every pair [31]. The goal is for the network to figure out how to convert the input image into the desired encrypted image. This goal is mainly dependent on:

- The image that the network takes in as input image and the intended output of the network is called the target image.
- The loss function in the U-Net aims to minimize the difference between the actual target image and its predicted output. L1 and L2 loss functions measure the pixel-wise difference between target and anticipated images to quantify this mismatch.

Usually, a U-Net, an encoder-decoder structure with skip links, is used as the generating network in this case. The Generator in a GAN often uses a U-Net architecture, which typically has a densely connected layer between the encoder and the decoder [32]. The U-Net design is an encoder-decoder with skip connections, which means that the layers in the encoder and decoder paths can be directly connected to one other. Tasks like image-to-image translation rely on the retention of spatial information, which this structure aids in. In most cases, a dense layer between the encoder and decoder is unnecessary for a U-net [33]. However, in some cases, a dense layer is used as a bottleneck layer to transform and possibly reduce the dimensionality of the encoded feature space before Up-sampling it back to its original size. Different layers of the Generator architecture are shown in Fig. 4 [34]. The tables below also show and describe the different levels.

Table II displays the U-Net encoder layers. Our GAN's Generator uses a U-Net design, which is well-known for its effectiveness in image-to-image translation tasks; this makes it ideal for image encryption. The encoder, bottleneck, and decoder are the three primary parts of the U-Net. All of the components work together to process and change the input image in a way that produces outputs of excellent quality.

Through a succession of convolutional layers, the encoder, also called the contracting path, captures the input image's context. The network is able to extract features from low-level to high-level data because the encoder layers continuously decrease the image's spatial dimensions while increasing the feature maps' depth. In order to cut the output size in half while keeping crucial spatial information, the layers employ a stride of 2 and padding 'same' [35]-[38]. As an example, the image being processed may have simple properties like edges and textures captured by the first convolutional layer, and then more complicated patterns and shapes can be identified by the next layers that build upon these features. Leaky ReLU activation [39] introduces non-linearity and tackles the vanishing gradient problem [40], while batch normalization is done after each convolutional layer (apart from the first) to stabilize and accelerate training. Connecting the encoder and decoder, the bottleneck layer captures the input image's most abstract representation [41]. The feature maps are compressed to their smallest form while preserving and precisely reconstructing the important characteristics [42]. The information about the bottleneck layer is displayed in Table III.

TABLE II. ARCHITECTURE OF THE U-NET ENCODER (DOWN-SAMPLING PATH)

Layer Type	Filter Size / Stride	Padding	Activation Function	Output Size
Input Layer	-	-	-	256×256×3
Conv2D	4×4 / 2	Same	Leaky ReLU (α=0.2)	128×128×64
MaxPooling2D	2×2 / 2	Valid	-	64×64×64
Conv2D	4×4 / 2	Same	Leaky ReLU (α=0.2)	32×32×128
MaxPooling2D	2×2 / 2	Valid	-	16×16×128
Conv2D	4×4 / 2	Same	Leaky ReLU (α=0.2)	8×8×256
Conv2D	4×4 / 2	Same	Leaky ReLU (α=0.2)	4×4×512
Conv2D	4×4 / 2	Same	Leaky ReLU (α=0.2)	2×2×512
Conv2D	4×4 / 2	Same	Leaky ReLU (α=0.2)	1×1×512

TABLE III. ARCHITECTURE OF THE BOTTLENECK LAYER

Layer Type	Activation Function	Additional
Fully Connected (Dense)	ReLU	Flatten input, Dense layer, Reshape back to 1x1x512

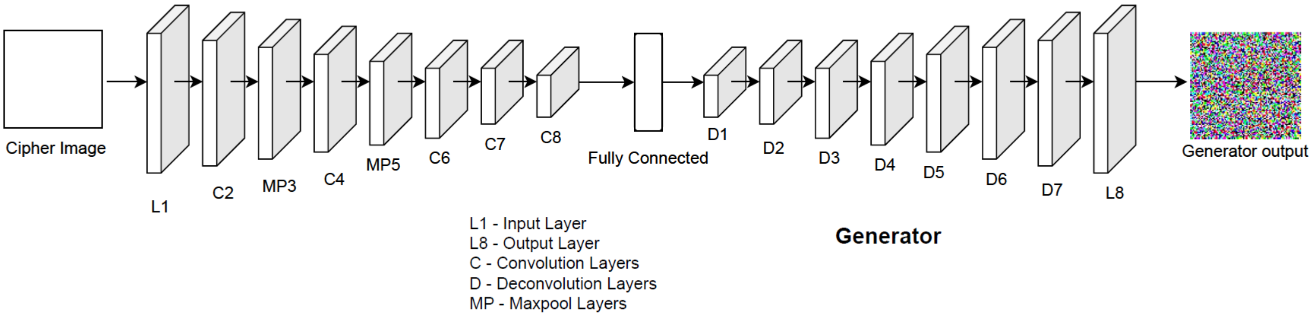


Fig. 4. Generator network of the proposed model

In order to keep the spatial resolution and provide a comprehensive set of characteristics that summarize the input image, the bottleneck employs a 4×4 kernel [43] with stride 1 and padding as 'same'.

Reconstructing the encrypted image from the abstract features collected in the bottleneck is the responsibility of the decoder, also called the expanding path [44], [45]. Table IV shows the schematics of the model Decoder that has been suggested. It uses skip connections from the relevant encoder layers to keep high-resolution features and spatial information as a succession of Up-sampling layers progressively restore the image's spatial dimensions [46]. To make sure the reconstructed image looks just like the original input, each Up-sampling layer increases the spatial dimensions while decreasing the depth. In the last layer of output, a *'tanh'* activation function is used to make sure that the pixel values fall within the range of [-1, 1], which is matching the input images' normalization. Using a U-Net architecture in the Generator, our GAN encrypts images efficiently, keeping important features and guaranteeing accurate reconstruction when decrypted. Because it strikes a good compromise between feature extraction and spatial preservation, its architecture works great for encrypting images. The encoder's usage of convolutional layers allows the network to extract important information at various abstraction levels, and the bottleneck guarantees a condensed image representation. Reconstructing the encrypted image is made easier by the decoder's Up-sampling layers and skip connections. This ensures that the output closely resembles the original input while preserving high-quality features.

In GANs, the *'tanh'* activation function is utilized in the output layer of the Generator network for several key reasons. It normalizes pixel values between -1 and 1, which stabilizes training and ensures consistent outputs. This function also improves gradient flow during backpropagation and reduces the likelihood of vanishing gradients compared to the sigmoid function. To start with, the fact that *'tanh'* returns values between -1 and 1 is useful for creating images with normalized pixel values; this, in turn, helps to stabilize the training process and guarantees consistent outputs. The *'tanh'* function also improves gradient flow in backpropagation and decreases the probability of vanishing gradients, in comparison to the sigmoid function, by providing non-zero gradients for a wider range of inputs. A more balanced and realistic-looking result is achieved by generating images with both negative and positive pixel values, made possible by the symmetry of *'tanh'* around the origin. In addition, the output layer of the Generator should use *'tanh'* to conform to the normalization practice of image processing, which is to normalize pixel values to the range of [-1, 1] instead of [0, 1]. This makes the produced images immediately usable for visualization or additional processing without the need for scaling. This design is crucial for generative network image encryption tasks because it guarantees a complete and efficient image reconstruction from the encoded representation while preserving high quality and fidelity. For applications like GAN-based image encryption, the PatchGAN Discriminator is crucial to the adversarial architecture. In Fig. 5, we can see the Discriminator model rendered. Focusing on fine-grained details, it assesses the realism of small image patches instead of the complete image. A number of convolutional layers were specifically engineered to efficiently collect and process this information in our enhanced PatchGAN Discriminator.

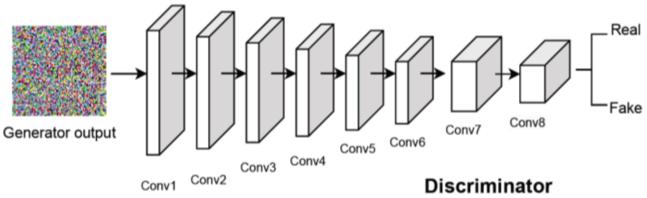


Fig. 5. Discriminator network of the proposed model

The Discriminator model starts with an input layer receiving images of dimensions 256×256×3. This input is processed through multiple convolutional layers, each designed to extract and refine features at different abstraction levels for precise image discrimination. The output is 128×128×64, made possible by the first convolutional layer's 4×4 kernel, stride of 2, and padding set to 'same'. In this layer, the input image's low-level properties, like textures and edges, are extracted. The following layer again employs a 4×4 kernel, but this time with a depth of 128 instead of 126, resulting in an output size of 64×64×128. In order to identify more intricate patterns and shapes, this layer expands upon the elements that were previously collected. Batch normalization is used to improve training speed and stability. Then, to incorporate non-linearity and alleviate the vanishing gradient problem, the Leaky ReLU activation function is implemented. Table V displays the Discriminator's layer details.

TABLE IV. ARCHITECTURE OF THE U-NET DECODER

Layer Type	Filter Size / Stride	Padding	Activation Function	Output Size	Additional
Deconv2D	4×4 / 2	Same	ReLU	2×2×512	Dropout (0.5) + Skip connection
Deconv2D	4×4 / 2	Same	ReLU	4×4×1024	Dropout (0.5) + Skip connection
Deconv2D	4×4 / 2	Same	ReLU	8×8×1024	Dropout (0.5) + Skip connection
Deconv2D	4×4 / 2	Same	ReLU	16×16×1024	Skip connection
Deconv2D	4×4 / 2	Same	ReLU	32×32×512	Skip connection
Deconv2D	4×4 / 2	Same	ReLU	64×64×256	Skip connection
Deconv2D	4×4 / 2	Same	ReLU	128×128×128	Skip connection
Deconv2D	4×4 / 2	Same	ReLU	256×256×64	Skip connection
Output Layer	4×4 / 1	Same	tanh	256×256×3	-

TABLE V. ARCHITECTURE OF THE DISCRIMINATOR

Layer	Kernel Size/Stride, Padding	Output Size	Activation Function
Input	-	256×256×3	-
Conv1	4×4/2, Same	128×128×64	Leaky ReLU (α=0.2)
Conv2	4×4/2, Same	64×64×128	Leaky ReLU (α=0.2) + Batch Norm
Conv3	4×4/2, Same	32×32×256	Leaky ReLU (α=0.2) + Batch Norm
Conv4	4×4/2, Same	16×16×512	Leaky ReLU (α=0.2) + Batch Norm
Conv5	4×4/2, Same	8×8×512	Leaky ReLU (α=0.2) + Batch Norm
Conv6	4×4/2, Same	4×4×512	Leaky ReLU (α=0.2) + Batch Norm
Conv7	4×4/1, Same	3×3×512	Leaky ReLU (α=0.2) + Batch Norm
Conv8	4×4/1, Same	2×2×1	Sigmoid

Next, the third convolutional layer reduces spatial dimensions to 32×32×256 using the same kernel and stride. More abstract features are extracted for fine picture details in this layer. The 16×16×512 fourth convolutional layer helps the network recognize visual high-level structures. The fifth convolutional layer gathers the most significant picture discriminating information at 8×8 at 512 depth. The sixth convolutional layer compresses spatial dimensions to 4×4×512 to capture critical characteristics. The seventh layer lowers output to 3×3×512 with stride 1 and padding "same". Deeper layers allow the network to collect abstract and complex characteristics for accurate picture discrimination.

Finally, the discriminator output layer uses a 4×4 kernel with stride 1 and padding "same" to compress the feature map to 2×2×1. This layer's sigmoid activation function calculates input patch realism probability scores. Focusing on specific areas rather than the entire image helps the discriminator make more accurate and localized authenticity decisions. To distinguish actual images from created ones, the enlarged PatchGAN discriminator architecture successively collects and refines features at several abstraction levels. Image encryption requires exact features and high-quality reconstruction, therefore extensive feature extraction works well.

A. GAN Loss Functions

Real images (plain/cipher) and fake ones (produced by the encryption/decryption network) are the two sources of data used by the Discriminator. The weights of the encryption network are kept constant during Discriminator D's training; the network then produces cipher images for the Discriminator to accurately classify. While the Discriminator only makes use of a single loss function, the Discriminator loss L_D; during training, the Generator makes use of two, the Generator Loss L_G and the Reconstruction loss L_{RC}. In order to update the weights through backpropagation and produce a global minimum, the Adam optimizer is used. In order to prevent the Discriminator from making accurate classifications, the encryption network learns to generate fake data. The encryption network transforms the input image into a cipher image. The Discriminator then evaluates this cipher image, comparing it to the original target cipher image to provide feedback for the encryption network's training. The Discriminator checks this cipher image for similarities to the original cipher image. When the Discriminator returns a result lower than 0.75, the encryption network's weights are adjusted according to the loss function employed by encryption network L_G, which is defined in (1).

$$L_G = \mu\big(1 - SSIM(x,y)\big) \tag{1}$$

On the other hand, the loss function to be minimized for training the Discriminator is defined as in (2).

$$L_D = \mu\big(SSIM(x,y)\big) \tag{2}$$

where μ = 0.2 is the hyperparameter to achieve an acceptable equilibrium between the structure aspect of target image and generated image, where SSIM, i.e., the structural similarity index metric, is defined as in (3).

$$SSIM(x,y) = \frac{\big(2\mu_x\mu_y + C_1\big)\big(\delta_{xy} + C_2\big)}{(\mu_x^2 + \mu_y^2 + C_1)\,(\delta_x^2 + \delta_y^2 + C_2)} \tag{3}$$

Where $C_1 = (k_1 L)^2$; $C_2 = (k_2 L)^2$; L is the maximum value of a pixel; k_1 = 0.01 and k_2 = 0.03 are constant parameters; δ_x represents standard deviation of image *x*; and δ_{xy} represents the covariance of image *x* and image *y*. The values of SSIM lies in range [0,1], where one indicates completely identical images. Two mappings, *G: X⟶ Y* and *H: Y⟶ X*, are included in the proposed model [13]. In order to fool the Discriminator, mapping function G must first determine how [21]. Making the change from source to target domain images Y accomplishes this task as per the loss functions mentioned in (4) [10]. Here $E_{x\sim p_{data(x)}}$represents the expectation over all images x in the domain X.

$$L_{RC} = E_{x\sim p_{data(x)}} \parallel Y - X \parallel_1 \tag{4}$$

During the training process, the Discriminator distinguishes between the authentic data generated by the Generator and the synthetic data. Deep learning algorithms often necessitate the use of a loss function for training the model. In (5), the overall loss is the aggregate of the losses incurred by the encryption neural network G L_G, the Discriminator network *D*, L_D, and the reconstruction loss of the decryption network *F*, L_{RC}.

$$L = L_G + L_D + L_{RC} \tag{5}$$

IV. PREPARE RESULTS AND DISCUSSIONS

The network for encryption and decryption has 17 levels of depth and uses around 3,000,000 parameters. To determine these optimal values, we utilized Keras Tuner, an open-source library for hyperparameter optimization. Keras Tuner helped us systematically explore different architectures and parameter configurations to identify the most effective model for our encryption tasks. The choice of 17 levels is strategic, allowing the model to capture intricate patterns and features essential for robust image encryption. After the network has been trained, these parameters are utilized as the encryption and decryption secret keys. A cryptanalytic assault becomes even more complicated as a result of the deep learning model's depth. In this part, we will go over some of the things that can compromise an encryption system's security.

A. *Analysis of the Histogram of Generated Images*

The histogram of three distinct plain images in Fig. 6 (a), (e), (i), which will be referred here on in this article as image_1, image_2, image_3 is shown in Fig. 6 (b), (f), (j) and their corresponding cipher images in Fig. 6 (c), (g), (k), have a histogram as in the Fig. 6 (d), (h), (l) respectively. The histogram analysis reveals significant differences between plain and encrypted images, indicating enhanced encryption security. Uniform histograms for encrypted images suggest high entropy and randomness, making cryptanalysis more difficult by minimizing patterns and correlations that attackers could exploit. This dispersion of pixel values aligns with Shannon's principles of confusion and diffusion, obscuring the relationship between plain and encrypted images and spreading the influence of each pixel widely. Such characteristics increase resistance to cryptanalysis and brute-force attacks, ensuring that CryptoGAN effectively disrupts statistical patterns, thereby fortifying the encrypted images against various attack vectors. These properties demonstrate that CryptoGAN achieves a high level of security by adhering to fundamental encryption principles and providing robust protection against common attack methods.

B. *Entropy Information of the Images*

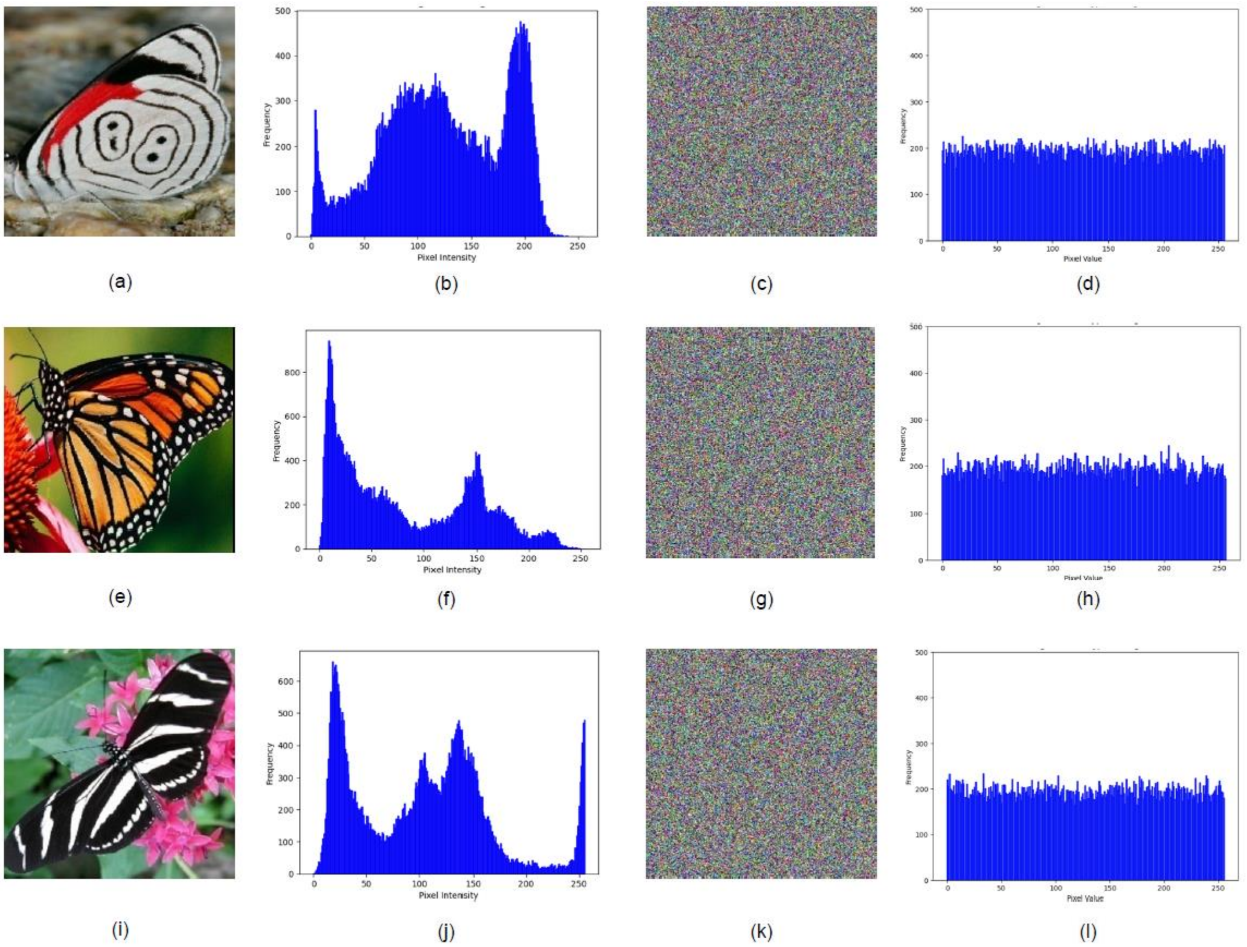
The uncertainty of pixels in the cipher image is defined by the image information entropy, which is calculated as in (6). The probability of image pixel i is denoted by p_i. For the image information entropy value of 8, the image will be with perfectly random pixels.

Pixels in both the plain and encrypted versions of an image are shown with their entropy values in Table VI. It also clearly shows that the entropy values of the plain images are lower than those of the cipher images produced by the image encryption approach.

$$Entropy = -\sum_{i=0}^{255} p_i \log p_i \tag{6}$$

TABLE VI. IMAGE ENTROPY VALUE INFORMATION OF PLAIN AND CIPHER IMAGES

Images	Original image entropy	Encrypted image entropy
image_1	7.63	7.9895
image_2	7.6	7.9512
image_3	7.72	7.9925



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig. 6. Plain image (a), (e), (i) have a histogram of (b), (f), (j) respectively and the corresponding encrypted images of the plain images are (c), (g), (k) respectively have a histogram of (d), (h), (l), respectively

C. Image Correlation Analysis

Neighboring pixel correlation determines the strength of an encryption model against statistical attacks. Adjacent pixel correlation in the horizontal direction was calculated by randomly choosing 3,500 horizontally adjacent pixels and then calculating the correlation coefficient between each of the adjacent pixels using (7).

$$r_{xy} = \frac{E\left[\left(x - E(x)\right)\left(y - E(y)\right)\right]}{\sqrt{\{D(x)D(y)\}}} \tag{7}$$

- Where, r_{xy} is the correlation coefficient between two adjacent pixels x and y.
- $E(x)$ and $E(y)$ are the expected values (means) of pixels x and y respectively and $D(x)$ and $D(y)$ are variances of pixels x and y respectively.

We also computed the correlation coefficient along the diagonal and vertical axes. Table VII shows the correlation coefficients between neighboring pixels in both plain and cipher images. In comparison to the plain images, cipher images clearly have a poor neighboring pixel correlation. Diffusion effects are more pronounced and regularity is diminished when correlation coefficients are lower. For this investigation, geographic statistics or correlation coefficients has been utilized. The horizontal correlation plot of the images_1, encrypted image_1 and decrypted image_1 is shown in Fig. 7(a), (b), (c) respectively, the vertical correlation plot of the images_2, encrypted image_2 and decrypted image_2 is shown in Fig. 7(d), (e), (f) respectively and the diagonal correlation plot of the images_3, encrypted image_3 and decrypted image_3 is shown in Fig. 7(g), (h), (i) respectively. Reduced correlation minimizes the risk of statistical attacks, such as differential cryptanalysis, by eliminating discernible patterns that can be exploited. By reducing correlation and increasing randomness, CryptoGAN enhances security, making encrypted images more resistant to various attacks.

TABLE VII. CORRELATION COEFFICIENTS VALUES AMONG ADJACENT PIXELS

Correlation coefficients	**image_1**	**Cipher of image_1**	**Image_2**	**Cipher of image_2**	**Image_3**	**Cipher of image_3**
Horizontal	0.9975	0.4925	0.9961	0.4950	0.9919	0.5023
Vertical	0.9858	0.4563	0.9857	0.4235	0.9926	0.5245
Diagonal	0.9985	0.4210	0.9889	0.4235	0.9981	0.4822

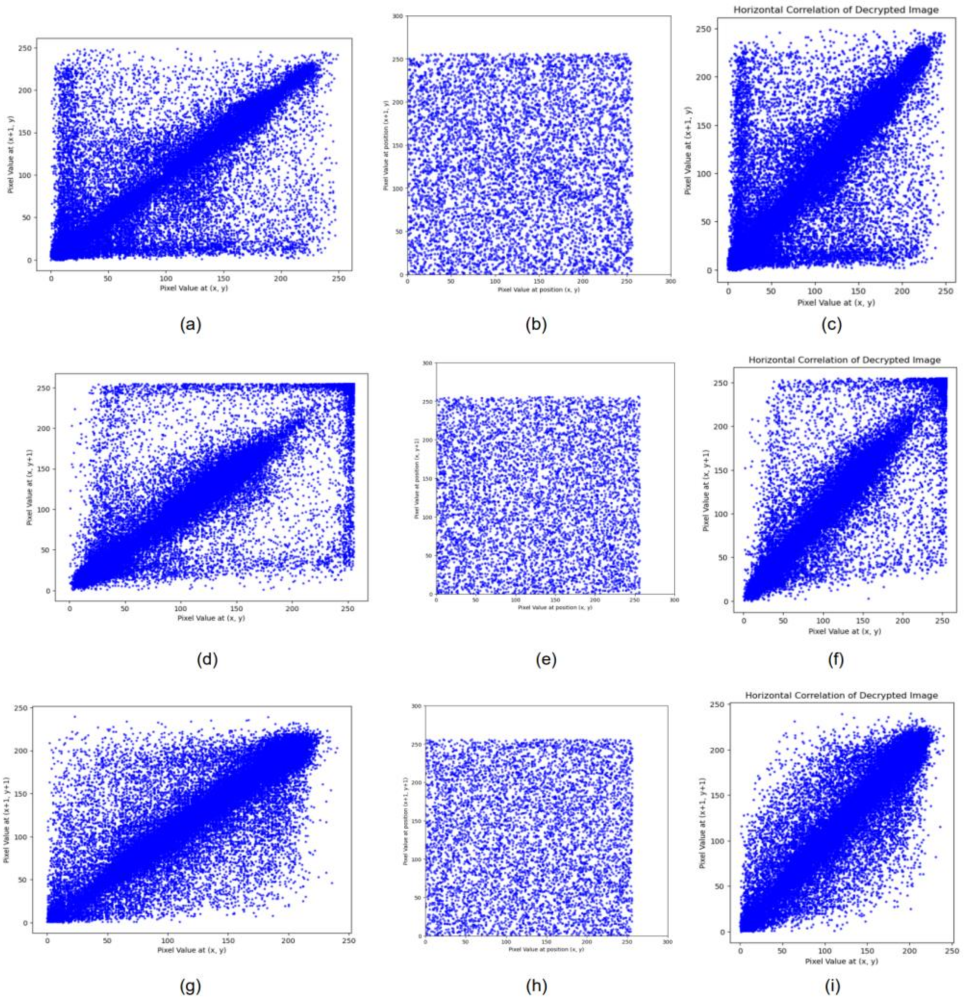


Fig. 7. The horizontal correlation of image_1, encrypted image_1 and decrypted image_1 is shown in a, b, c respectively. The vertical correlation of image_2, encrypted image_2 and decrypted image_2 is shown in d, e, f respectively. The diagonal correlation of plain-image_3, encrypted image_3 and decrypted image_3 is shown in g, h, i respectively

D. Synthesised and Reconstructed Image Quality

It is essential to evaluate the quality of synthesized and reconstructed images following encryption and decryption in order to determine how effective image encryption techniques are. Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) are widely used measures that offer complementary insights into image quality. SSIM assesses structural fidelity by taking brightness, contrast, and structural similarity into account, whereas PSNR detects pixel-level variations. We study both reconstructed (pictures decrypted back to their original form) and synthesized (images produced by generative models) images. Maintaining high SSIM and PSNR values for synthesized images guarantees that the generative process keeps important structural elements and visual quality intact, even after encryption and decryption. High SSIM values near 1 suggest that little distortion occurs throughout the encryption-decryption cycle and that the structural information is mostly preserved. The Fig. 8 shows the encrypted image and the reconstructed image.

For applications where visual fidelity is critical, such as secure image transfer and medical imaging, this is essential [16]. The SSIM and PSNR values for reconstructed images show how successfully the original image is recovered following the encryption and decryption procedures. High PSNR values show little pixel-level variation, while high SSIM values guarantee that the image's fundamental structural elements are retained [47]. When taken as a whole, these measures demonstrate how well our encryption technique maintains image quality, guaranteeing that the decrypted photos have almost the same structural and visual integrity as the originals. This dual-metric technique offers a thorough evaluation of image quality post-encryption and decryption, highlighting the efficacy of our technology in practical settings.

E. Performance Comparison of Experimental Results between the Proposed Method and Other Related Works

1) Entropy: Entropy measures the randomness of the pixel distribution in the encrypted images. Higher entropy values suggest better security against statistical attacks. The computed entropy results are shown in Table VIII. The table also demonstrates that the suggested scheme has an average value of 7.9925, which is much better than the findings of the existing mechanisms in the literature with a smaller standard deviation.

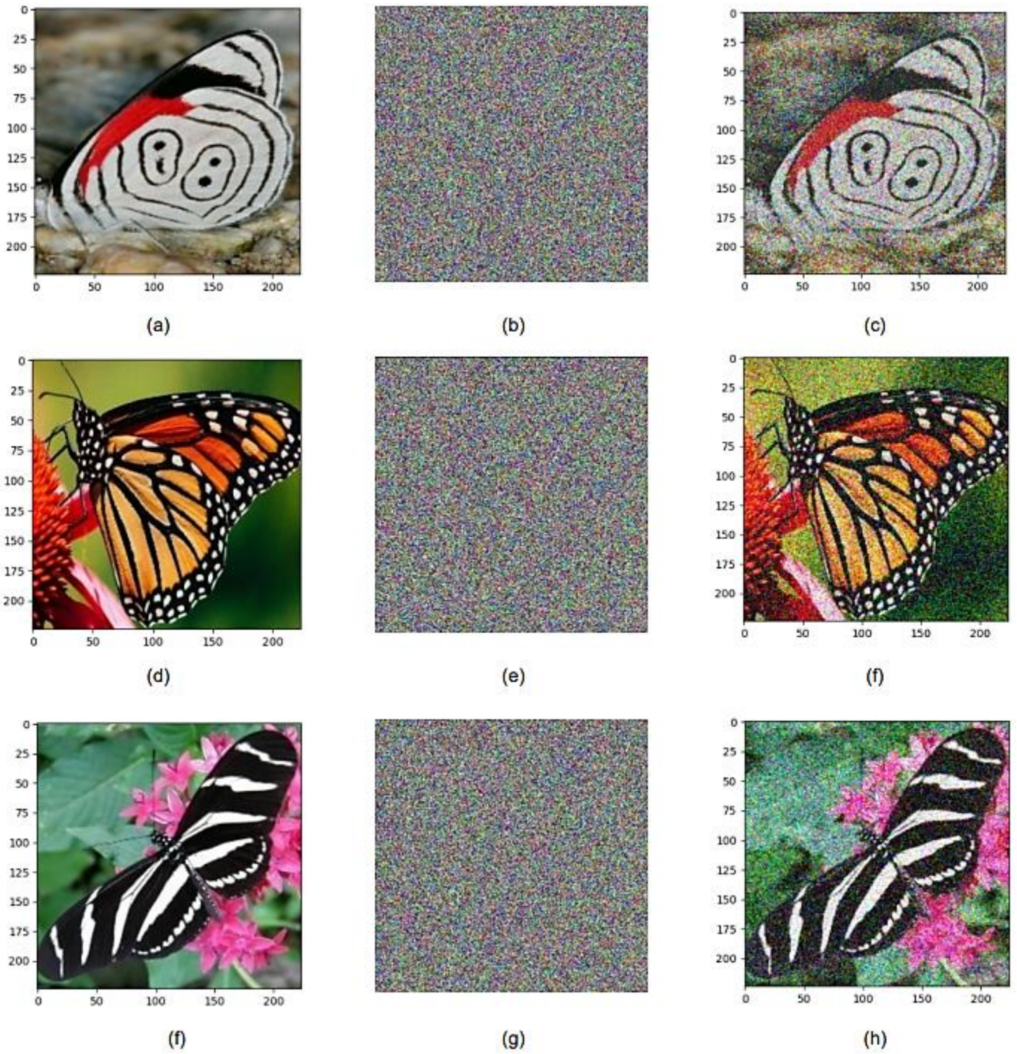


Fig. 8. (a), (d), (f) are the original images and the corresponding encrypted images are in 9(b), (e), (g) respectively and the reconstructed images are shown in 9(c), (f), (h) respectively

TABLE VIII. ENTROPY AND STANDARD DEVIATION OF THE PROPOSED MODEL WITH OTHER MODELS

Works for study and comparative analysis	Experimental Values of Entropy	Standard Deviation
DeepEDN: A Deep-Learning-Based Image Encryption & Decryption Network for Internet of Medical Things [18]	7.9586	-
EncryptGAN: Image steganography with domain transform [48]	7.9758	-
Image cipher based on mixed transformed logistic maps [49]	7.9850	0.000724
Chaotic maps-based image encryption scheme [50]	7.9850	0.000392
Chaotic image encryption algorithm based on information entropy [51]	7.9536	0.025611
Image Encryption Algorithm Based on Quantum Chaos Sequence [52]	7.9855	0.152205
Intertwining chaotic maps and RC4 stream cipher [53]	7.9841	0.000733
CryptoGAN (Proposed Model)	7.9925	0.006474

2) Peak signal-to-noise ratio: The natural image has a significant pixel correlation and plausible structural properties. This method uses the SSIM index as a loss function to capture the essential structure of both the produced (cypher image) and recovered (original image) images. The proposed network encrypts pictures after model training. Restoration quality is confirmed by the similarities between restored and untreated photographs' mosaics. The original image and the suggested decryption network image were compared for peak signal-to-noise ratio. Table IX's PSNR values demonstrate network performance [54]. It also compares the proposed network's Peak Signal-to-Noise Ratio (PSNR) to other encryption networks.

TABLE IX. PSNR COMPARISON OF THE PROPOSED MODEL WITH OTHER ENCRYPTION MODELS

Works for study and comparative analysis	Method	Average PSNR
EncryptGAN: Image steganography with domain transform [48]	GAN-based encryption	20.53
Optical Image Encryption using Deep Learning [55]	Deep learning-based encryption	30.00
DeepEDN: A Deep-Learning Image Encryption and Decryption Network for Internet of Medical Things [18]	Deep learning-based encryption	36.53
Chaos-based Digital Image Encryption Using Iris Features [56]	Deep learning-based encryption	33.72
CryptoGAN (Proposed Model)	GAN-based encryption	36.85

3) Structural Similarity Index (SSIM): The Structural Similarity Index (SSIM) is a crucial metric for evaluating the quality of recovered images in image encryption and decryption processes. Unlike traditional metrics such as Peak Signal-to-Noise Ratio (PSNR), which primarily focus on pixel-level differences, SSIM assesses image quality by considering changes in structural information, luminance, and contrast [57]-[62]. This makes SSIM particularly valuable for image encryption research, where preserving the structural integrity of the decrypted image is paramount. When comparing the original and recovered images, SSIM provides a more holistic view of image quality. It quantifies how similar the structures, textures, and overall visual features of the decrypted image are to the original. An SSIM value closer to 1 indicates high similarity, implying that the encryption and subsequent decryption processes have minimally altered the essential content of the image. Conversely, a lower SSIM value suggests significant deviations, highlighting areas where the decryption process may need improvement.

In our research, the use of SSIM to evaluate the recovered images demonstrates the effectiveness of our encryption algorithm. By maintaining high SSIM values, we ensure that the decrypted images retain their original structural properties, making our approach suitable for applications where visual fidelity is critical [63], such as medical imaging and secure image transmission [64]. The combination of SSIM and PSNR provides a comprehensive assessment, balancing pixel-level accuracy with structural similarity [65], thus validating the robustness and quality of our image encryption methodology. Table X shows a detailed contrast of the SSIM of the proposed model with the other models in the similar domains [66]. The potential biases introduced by the specific butterfly dataset used here could affect the generalizability and effectiveness of the proposed encryption method [67][68]. To mitigate this, we plan to incorporate a variety of image datasets from different domains to ensure a more robust evaluation of the model's performance across different scenarios in the scope of its future [69][70].

4) Model Speed and Complexity: In both efficiency and scalability, CryptoGAN also achieves the fast encryption of 90ms and decryption of 85ms speeds, with a linear computational complexity of O(n), outperforming other models performing in the range of 120ms to 150ms [71]-[74].

TABLE X. SSIM COMPARISON OF THE PROPOSED MODEL WITH OTHER ENCRYPTION MODELS

Works for study and comparative analysis	Method	Original vs generated image SSIM
DeepEDN: A deep-learning-based image encryption and decryption network [18]	DL-based image encryption	0.90
Image encryption using deep neural network and chaotic map [75]	DL-based image encryption	1.0
Image encryption scheme based on chaotic logarithmic map and key generation using deep CNN [76]	CNN/ DL-based image encryption	1.0
Phase-only optical image encryption and hiding method via deep learning [55], [77]	DL-based image encryption	0.88
CryptoGAN (Proposed Model)	GAN Based Encryption	0.92

Several techniques were used to mitigate possible overfitting difficulties related to the application of U-Net and PatchGAN designs on a relatively limited dataset [78]. To improve the model's ability to generalise, data augmentation techniques such as random rotations, flips, shifts, and scaling were used to expand the variety of training samples [79], [80]. In order to keep the model from growing unduly dependent on any one set of features during training, dropout regularisation was included to the discriminator and generator networks. In order to decrease sensitivity to initialisations and encourage generalisation, batch normalisation was utilised to stabilise and speed up training. And to avoid overfitting, early stopping based on validation loss was also used, which stopped training when the validation performance stopped getting better. Using the above techniques, the output results showed consistent performance across the data subsets. These safeguards guarantee that CryptoGAN can handle a variety of unknown data types with effectiveness and is resistant to overfitting.

V. Conclusions

In this study, we introduced CryptoGAN, a GAN-based image encryption model employing U-Net and PatchGAN architectures. CryptoGAN was trained with data augmentation, dropout, and batch normalization to prevent overfitting and enhance generalization. Evaluated on a custom dataset of 2000 butterfly images, CryptoGAN achieved an average PSNR of 36.85 and an SSIM of 0.94, demonstrating high-quality and faster encryption and decryption. Uniform histograms of encrypted images suggest high entropy and randomness, enhancing resistance to cryptanalysis and brute-force attacks. While the custom dataset provided a controlled test environment, future work will involve applying CryptoGAN to diverse datasets to ensure broader applicability. With encryption and decryption times of 90ms and 85ms, respectively, and a linear computational complexity (O(n)), CryptoGAN is scalable and efficient for real-world applications. Future research will focus on validating these findings across different datasets and exploring practical deployment scenarios to ensure the model's effectiveness in real-world applications.

References

[1] P. Sharma, A. Singh, S. Raheja, and K. K. Singh, "Automatic vehicle detection using spatial time frame and object-based classification," *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 6, pp. 8147-8157, 2019, doi: 10.3233/jifs-190593.

[2] J. S. Teh, M. Alawida, and Y. C. Sii, "Implementation and practical problems of chaos-based cryptography revisited," *Journal of Information Security and Applications*, vol. 50, p. 102421, 2020, doi: 10.1016/j.jisa.2019.102421.

[3] P. Singh, N. Singh, K. K. Singh, and A. Singh, "Diagnosing of disease using machine learning," *Machine Learning and the Internet of Medical Things in Healthcare*, pp. 89-111, 2021, doi: 10.1016/b978-0-12-821229-5.00003-3.

[4] W. Sirichotedumrong and H. Kiya, "A GAN-Based Image Transformation Scheme for Privacy-Preserving Deep Neural Networks," *2020 28th European Signal Processing Conference (EUSIPCO)*, pp. 745-749, 2021, doi: 10.23919/eusipco47968.2020.9287532.

[5] A. Mokhnache and L. Ziet, "Cryptanalysis of a Pixel Permutation Based Image Encryption Technique Using Chaotic Map," *Traitement du Signal*, vol. 37, no. 1, pp. 95-100, 2020, doi: 10.18280/ts.370112.

[6] X. Wang and J. Yang, "A privacy image encryption algorithm based on piecewise coupled map lattice with multi dynamic coupling coefficient," *Information Sciences*, vol. 569, pp. 217-240, 2021, doi: 10.1016/j.ins.2021.04.013.

[7] X. Wang and X. Chen, "An image encryption algorithm based on dynamic row scrambling and Zigzag transformation," *Chaos Solitons & Fractals*, vol. 147, p. 110962, 2021, doi: 10.1016/j.chaos.2021.110962.

[8] B. Rahul, K. Kuppusamy, and A. Senthilrajan, "Dynamic DNA cryptography-based image encryption scheme using multiple chaotic maps and SHA-256 hash function," *Optik*, vol. 289, p. 171253, 2023, doi: 10.1016/j.ijleo.2023.171253.

[9] X. Wang and S. Gao, "Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network," *Information Sciences*, vol. 539, pp. 195-214, 2020, doi: 10.1016/j.ins.2020.06.030.

[10] K. Panwar, R. K. Purwar, and G. Srivastava, "A Fast Encryption Scheme Suitable for Video Surveillance Applications Using SHA-256 Hash Function and 1D Sine–Sine Chaotic Map," *International Journal of Image and Graphics*, vol. 21, no. 2, p. 2150022, 2020, doi: 10.1142/s0219467821500224.

[11] K. Sahu, G. Swain, M. Sahu, and J. Hemalatha, "Multi-directional block based PVD and modulus function image steganography to avoid FOBP and IEP," *Journal of Information Security and Applications*, vol. 58, p. 102808, 2021, doi: 10.1016/j.jisa.2021.102808.

[12] A. K. Sahu and M. Sahu, "Digital image steganography and steganalysis: A journey of the past three decades," *Open Computer Science*, vol. 10, no. 1, pp. 296-342, 2020, doi: 10.1515/comp-2020-0136.

[13] Z. Bao, R. Xue, and Y. Jin, "Image scrambling adversarial autoencoder based on the asymmetric encryption," *Multimedia Tools and Applications*, vol. 80, no. 18, pp. 28265-28301, 2021, doi: 10.1007/s11042-021-11043-3.

[14] F. Sherif, W. A. Mohamed, and A. Mohra, "Skin Lesion Analysis Toward Melanoma Detection Using Deep Learning Techniques," *International Journal of Electronics and Telecommunications*, pp. 597-602, 2019, doi: 10.24425/ijet.2019.129818.

[15] C. Zhang, P. Benz, A. Karjauv, and I. S. Kweon, "Universal Adversarial Perturbations Through the Lens of Deep Steganography: Towards a Fourier Perspective," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, pp. 3296-3304, 2021, doi: 10.1609/aaai.v35i4.16441.

[16] Z. Bao and R. Xue, "Research on the avalanche effect of image encryption based on the Cycle-GAN," *Applied Optics*, vol. 60, no. 18, p. 5320, 2021, doi: 10.1364/ao.428203.

[17] S. R. Maniyath and T. V, "An efficient image encryption using deep neural network and chaotic map," *Microprocessors and Microsystems*, vol. 77, p. 103134, 2020, doi: 10.1016/j.micpro.2020.103134.

[18] Y. Ding, "DeepEDN: A Deep-Learning-Based Image Encryption and Decryption Network for Internet of Medical Things," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1504-1518, 2021, doi: 10.1109/jiot.2020.3012452.

[19] W. Shi and S. Liu, "Hiding Message Using a Cycle Generative Adversarial Network," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 18, no. 3, pp. 1-15, 2022, doi: 10.1145/3495566.

[20] Z. Man, J. Li, X. Di, Y. Sheng, and Z. Liu, "Double image encryption algorithm based on neural network and chaos," *Chaos, Solitons & Fractals*, vol. 152, p. 111318, 2021, doi: 10.1016/j.chaos.2021.111318.

[21] Y. Ding, F. Tan, Z. Qin, M. Cao, K.-K. R. Choo, and Z. Qin, "DeepKeyGen: A Deep Learning-Based Stream Cipher Generator for Medical Image Encryption and Decryption," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 4915-4929, 2022, doi: 10.1109/tnnls.2021.3062754.

[22] C. Wang and Y. Zhang, "A novel image encryption algorithm with deep neural network," *Signal Processing*, vol. 196, p. 108536, 2022, doi: 10.1016/j.sigpro.2022.108536.

[23] Y. Sang, J. Sang, and M. S. Alam, "Image encryption based on logistic chaotic systems and deep autoencoder," *Pattern Recognition Letters*, vol. 153, pp. 59-66, 2022, doi: 10.1016/j.patrec.2021.11.025.

[24] R. Meng, Q. Cui, Z. Zhou, Z. Fu, and X. Sun, "A Steganography Algorithm Based on CycleGAN for Covert Communication in the

Internet of Things," *IEEE Access*, vol. 7, pp. 90574-90584, 2019, doi: 10.1109/access.2019.2920956.

[25] K. Panwar, R. K. Purwar, and A. Jain, “Cryptanalysis and Improvement of a Color Image Encryption Scheme Based on DNA Sequences and Multiple 1D Chaotic Maps,” *International Journal of Bifurcation and Chaos*, vol. 29, no. 8, p. 1950103, 2019, doi: 10.1142/s0218127419501037.

[26] T. Asanuma and T. Isobe, “Even-Mansour Space-hard Cipher: White-box Cryptography Cipher Meets Physically Unclonable Function,” *Journal of Information Processing*, vol. 31, pp. 88-96, 2023, doi: 10.2197/ipsjjip.31.88.

[27] L. Chen, C. Li, and C. Li, “Security measurement of a medical communication scheme based on chaos and DNA coding,” *Journal of Visual Communication and Image Representation*, vol. 83, p. 103424, 2022, doi: 10.1016/j.jvcir.2021.103424.

[28] L. Wang and H. Cheng, "Pseudo-Random Number Generator Based on Logistic Chaotic System," *Entropy*, vol. 21, no. 10, p. 960, 2019, doi: 10.3390/e21100960.

[29] R. M. R. Guddeti, “Exploiting skeleton-based gait events with attention-guided residual deep learning model for human identification,” *Applied Intelligence*, vol. 53, no. 23, pp. 28711-28729, 2023, doi: 10.1007/s10489-023-05019-z.

[30] A. Arifianto, "EDGAN: Disguising Text as Image using Generative Adversarial Network," *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pp. 1-6, 2020, doi: 10.1109/ISRTI51436.2020.9315474.

[31] J. Chen, X.-W. Li, and Q.-H. Wang, “Deep Learning for Improving the Robustness of Image Encryption,” *IEEE Access*, vol. 7, pp. 181083-181091, 2019, doi: 10.1109/access.2019.2959031.

[32] X. Dong, “Automatic multiorgan segmentation in thorax CT images using U-net-GAN,” *Medical Physics*, vol. 46, no. 5, pp. 2157-2168, 2019, doi: 10.1002/mp.13458.

[33] C. Li, X. Shen, and S. Liu, “Cryptanalyzing an Image Encryption Algorithm Underpinned by 2-D Lag-Complex Logistic Map,” *IEEE MultiMedia*, vol. 31, no. 1, pp. 99-109, 2024, doi: 10.1109/mmul.2024.3356494.

[34] M. Lawnik and M. Berezowski, “New Chaotic System: M-Map and Its Application in Chaos-Based Cryptography,” *Symmetry*, vol. 14, no. 5, p. 895, 2022, doi: 10.3390/sym14050895.

[35] C. Li, B. Feng, S. Li, J. Kurths, and G. Chen, “Dynamic Analysis of Digital Chaotic Maps via State-Mapping Networks,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 6, pp. 2322-2335, 2019, doi: 10.1109/tcsi.2018.2888688.

[36] X. Lu, C. Li, and K. Tan, “Network Analysis of Chebyshev Polynomial in a Fixed-precision Digital Domain,” *2021 40th Chinese Control Conference (CCC)*, pp. 8634-8638, 2021, doi: 10.23919/ccc52363.2021.9550220.

[37] Y. Wu, Y. Wan, L. Tang, and W. Xiong, "A Generative Adversarial Network-based Approach to Image Synthesis with Self-Attention Mechanism," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3824-3833, 2020, doi: 10.1109/CVPR42600.2020.00382.

[38] M. Shafiq and Z. Gu, “Deep Residual Learning for Image Recognition: A Survey,” *Applied Sciences*, vol. 12, no. 18, p. 8972, 2022, doi: 10.3390/app12188972.

[39] D. Kumar, A. B. Joshi, S. Singh, and V. N. Mishra, “Digital color-image encryption scheme based on elliptic curve cryptography ElGamal encryption and 3D Lorenz map,” *International conference on recent trends in applied mathematical sciences (ICRTAMS-2020)*, vol. 2364, no. 1, 2021, doi: 10.1063/5.0062877.

[40] A. Bose, A. Kumar, M. K. Hota, and S. Sherki, “Steganography Method Using Effective Combination of RSA Cryptography and Data Compression,” *2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, pp. 1-5, 2022, doi: 10.1109/iceeict53079.2022.9768402.

[41] Z. Zhang, G. Fu, F. Di, C. Li, and J. Liu, “Generative Reversible Data Hiding by Image-to-Image Translation via GANs,” *Security and Communication Networks*, vol. 2019, pp. 1-10, 2019, doi: 10.1155/2019/4932782.

[42] M. Hasani and H. Khotanlou, “An Empirical Study on Position of the Batch Normalization Layer in Convolutional Neural Networks,” *2019 5th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pp. 1-4, 2019, doi: 10.1109/icspis48872.2019.9066113.

[43] B. François, S. Thao, and M. Vrac, “Adjusting spatial dependence of climate model outputs with Cycle-Consistent Adversarial Networks,” *Climate dynamics*, vol. 57, no. 11, pp. 3323-3353, 2021, doi: 10.21203/rs.3.rs-299929/v1.

[44] R. Cakaj, J. Mehnert, and B. Yang, “Spectral Batch Normalization: Normalization in the Frequency Domain,” *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-10, 2023, doi: 10.1109/ijcnn54540.2023.10191931.

[45] N. Subramanian, O. Elharrouss, S. Al-Maadeed, and A. Bouridane, “Image Steganography: A Review of the Recent Advances,” *IEEE Access*, vol. 9, pp. 23409-23423, 2021, doi: 10.1109/access.2021.3053998.

[46] X. Duan, K. Jia, B. Li, D. Guo, E. Zhang, and C. Qin, “Reversible Image Steganography Scheme Based on a U-Net Structure,” *IEEE Access*, vol. 7, pp. 9314-9323, 2019, doi: 10.1109/access.2019.2891247.

[47] Q. Zhou, X. Wang, M. Jin, L. Zhang, and B. Xu, “Optical image encryption based on two-channel detection and deep learning,” *Optics and Lasers in Engineering*, vol. 162, p. 107415, 2023, doi: 10.1016/j.optlaseng.2022.107415.

[48] Z. Zheng, H. Liu, Z. Yu, H. Zheng, Y. Wu, Y. Yang, and J. Shi, "EncryptGAN: Image steganography with domain transform," *arXiv:1905.11582*, 2019.

[49] K. SundaraKrishnan, R. SP, and J. B, “A Symmetric Key Multiple Color Image Cipher Based on Cellular Automata, Chaos Theory and Image Mixing,” *Information Technology and Control*, vol. 50, no. 1, pp. 55-75, 2021, doi: 10.5755/j01.itc.50.1.28012.

[50] Y. Wang, “Multiple color image encryption based on cascaded quaternion gyrator transforms,” *Signal Processing: Image Communication*, vol. 107, p. 116793, 2022, doi: 10.1016/j.image.2022.116793.

[51] G. Ye, C. Pan, X. Huang, Z. Zhao, and J. He, "A chaotic image encryption algorithm based on information entropy," *Int. J. Bifurcat. Chaos*, vol. 28, no. 01, p. 1850010, 2018, doi: 10.1142/S0218127418500104.

[52] J. Zhang and D. Huo, “Image encryption algorithm based on quantum chaotic map and DNA coding,” *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 15605-15621, 2018, doi: 10.1007/s11042-018-6973-6.

[53] M. Kumari and S. Gupta, “A Novel Image Encryption Scheme Based on Intertwining Chaotic Maps and RC4 Stream Cipher,” *3D Research*, vol. 9, no. 1, 2018, doi: 10.1007/s13319-018-0162-2.

[54] B. Zhang, B. Rahmatullah, S. L. Wang, and Z. Liu, “A plain-image correlative semi-selective medical image encryption algorithm using enhanced 2D-logistic map,” *Multimedia Tools and Applications*, vol. 82, no. 10, pp. 15735-15762, 2022, doi: 10.1007/s11042-022-13744-9.

[55] Q. Zhang and J. Li, “Single Exposure Phase-Only Optical Image Encryption and Hiding Method via Deep Learning,” *IEEE Photonics Journal*, vol. 14, no. 1, pp. 1-8, 2022, doi: 10.1109/jphot.2022.3146456.

[56] D. F. Santos, “Chaos-based Digital Image Encryption Using Unique Iris Features,” *International Journal of Applied Engineering Research*, vol. 15, no. 4, p. 358, 2020, doi: 10.37622/ijaer/15.4.2020.358-363.

[57] M. Alkhelaiwi, W. Boulila, J. Ahmad, A. Koubaa, and M. Driss, “An Efficient Approach Based on Privacy-Preserving Deep Learning for Satellite Image Classification,” *Remote Sensing*, vol. 13, no. 11, p. 2221, 2021, doi: 10.3390/rs13112221.

[58] Y. Al Najjar, “Comparative Analysis of Image Quality Assessment Metrics: MSE, PSNR, SSIM and FSIM,” *International Journal of Science and Research (IJSR)*, vol. 13, no. 3, pp. 110-114, 2024, doi: 10.21275/sr24302013533.

[59] Y. Reznik, “Another look at SSIM image quality metric,” *Electronic Imaging*, vol. 35, no. 8, 2023, doi: 10.2352/ei.2023.35.8.iqsp-305.

[60] M. Martini, “On the relationship between SSIM and PSNR for DCT-based compressed images and video: SSIM as content-aware PSNR,” *Authorea Preprints*, 2023, doi: 10.36227/techrxiv.21725390.

[61] D. R. I. M. Setiadi, “PSNR vs SSIM: imperceptibility quality assessment for image steganography,” *Multimedia Tools and*

Applications, vol. 80, no. 6, pp. 8423-8444, 2020, doi: 10.1007/s11042-020-10035-z.

[62] U. Sara, M. Akter, and M. S. Uddin, “Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study,” *Journal of Computer and Communications*, vol. 7, no. 3, pp. 8-18, 2019, doi: 10.4236/jcc.2019.73002.

[63] L. M. H. Yepdia and A. Tiedeu, “Secure Transmission of Medical Image for Telemedicine,” *Sensing and Imaging*, vol. 22, no. 1, 2021, doi: 10.1007/s11220-021-00340-8.

[64] P. Udayakumar and N. Rajagopalan, “(Retracted) Blockchain enabled secure image transmission and diagnosis scheme in medical cyber-physical systems,” *Journal of Electronic Imaging*, vol. 31, no. 6, 2022, doi: 10.1117/1.jei.31.6.062002.

[65] A. Orman, “Image Retrieval Using Pixel Similarity,” *Research Square*, 2023, doi: 10.21203/rs.3.rs-3311259/v1.

[66] M. H. Shaheen, “Proposed Hybrid Encryption Framework for Reliable 3-D Wireless Video Communications,” *Hybrid Encryption Algorithms Over Wireless Communication Channels*, pp. 82-103, 2021, doi: 10.1201/9781003051428-5.

[67] R. S. Ali, M. K. Ibrahim, and S. N. Alsaad, “Fast and Secure Image Encryption System Using New Lightweight Encryption Algorithm,” *TEM Journal*, pp. 198-206, 2024, doi: 10.18421/tem131-20.

[68] M. Li, Q. Cui, X. Wang, Y. Zhang, and Y. Xiang, “Ftpe-Bc: Fast Thumbnail-Preserving Image Encryption Using Block-Churning,” *Available at SSRN 4698446*, 2024, doi: 10.2139/ssrn.4698446.

[69] D. Koeglmayr and C. Räth, “A fast reservoir computing based image encryption algorithm,” *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-7, 2023, doi: 10.1109/ijcnn54540.2023.10191083.

[70] H. Ghanbari, R. Enayatifar, and H. Motameni, “A Fast Image Encryption based on Linear Feedback Shift Register and Deoxyribonucleic acid,” *Research Square*, 2022, doi: 10.21203/rs.3.rs-1662684/v1.

[71] F. Neri, “An Introduction to Computational Complexity,” *Linear Algebra for Computational Sciences and Engineering*, pp. 419-432, 2019, doi: 10.1007/978-3-030-21321-3_11.

[72] D.-G. Cheroiu, M. Raducanu, and C. M. Nitu, “Fast Image Encryption Algorithm Based on Multiple Chaotic Maps,” *2022 14th International Conference on Communications (COMM)*, pp. 1-4, 2022, doi: 10.1109/comm54429.2022.9817317.

[73] Z. Liu, J. Y. Liu, L. Y. Zhang, Y. Zhao, and X. F. Gong, “Performance of the 2D Coupled Map Lattice Model and Its Application in Image Encryption,” *Complexity*, vol. 2022, no. 1, 2022, doi: 10.1155/2022/5193618.

[74] C. F. Foo and S. Winkler, “Image Data Augmentation with Unpaired Image-to-Image Camera Model Translation,” *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 3246-3250, 2022, doi: 10.1109/icip46576.2022.9897671.

[75] M. S. Fadhil, A. K. Farhan, and M. N. Fadhil, “Designing Substitution Box Based on the 1D Logistic Map Chaotic System,” *IOP Conference Series: Materials Science and Engineering*, vol. 1076, no. 1, p. 012041, 2021, doi: 10.1088/1757-899x/1076/1/012041.

[76] U. Erkan, A. Toktas, S. Enginoğlu, E. Akbacak, and D. N. H. Thanh, “An image encryption scheme based on chaotic logarithmic map & key generation using deep CNN,” *Multimedia Tools & Applications*, vol. 81, no. 5, pp. 7365-7391, 2022, doi: 10.1007/s11042-021-11803-1.

[77] Z. Wang, “Data Hiding with Deep Learning: A Survey Unifying Digital Watermarking and Steganography,” *IEEE Transactions on Computational Social Systems*, vol. 10, no. 6, pp. 2985-2999, 2023, doi: 10.1109/tcss.2023.326895.

[78] S. Haunts, “Hybrid Encryption,” *Applied Cryptography in .NET and Azure Key Vault*, pp. 113-141, 2019, doi: 10.1007/978-1-4842-4375-6_9.

[79] M. S. Alam, D. Wang, and A. Sowmya, “Image data augmentation for improving performance of deep learning-based model in pathological lung segmentation,” *2021 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1-5, 2021, doi: 10.1109/dicta52665.2021.9647209.

[80] L. Tong, P. Xia, and T. Lv, “Research on Quantum Secure Route Model and Line Model Image Encryption Technology Based on Big Data Technology,” *2022 International Conference on Cloud Computing, Big Data Applications and Software Engineering (CBASE)*, pp. 115-118, 2022, doi: 10.1109/cbase57816.2022.00028.

[81] F. S. Abas and R. Arulmurugan, “Radix Trie improved Nahrain chaotic map-based image encryption model for effective image encryption process,” *International Journal of Intelligent Networks*, vol. 3, pp. 102-108, 2022, doi: 10.1016/j.ijin.2022.08.002.