Journal of Information Technology Management
ISSN-OnLine: 2980-7972
University of Tehran

Deep Learning Iris Cryptography: A Deep Learning-Based Image Encryption Framework for Secure Information Management

Awadhesh Kumar Singh *

*Corresponding author, Ph.D. Scholar, Department of CSE, Sharda School of Engineering & Technology, Sharda University, Greater Noida-201306 UP, India. E-mail: awadheshcse@gmail.com

Amrit Kumar Agrawal

Associate Professor, Department of CSE, Sharda School of Engineering & Technology, Sharda University, Greater Noida-201306 UP, India. E-mail: agrawal.amrit4@gmail.com

Journal of Information Technology Management, 2026, Vol. 18, Issue 4, pp. 117-133.
Published by the University of Tehran, College of Management
doi: https://doi.org/ 10.22059/jitm.2026.108918
Article Type: Research Paper

Received: June 21, 2026
Received in revised form: July 11, 2026
Accepted: August 19, 2026
Published online: October 01, 2026

Abstract

As high-resolution image data increasingly becomes commonplace in healthcare, surveillance systems, smart environments, and cloud platforms, ensuring image security has become one of the major challenges of our time. In this paper, we develop DL-IrisCrypt, a novel biometric-based image encryption framework in which the ciphertext is associated with the identity of the authorized user. We extract a 2048-bit iris code using Daugman segmentation and Gabor filters, which is then used to generate a key for initializing a six-dimensional hyperchaotic system and processed through SHA-512 and HKDF. To enhance randomness in the key-generation process, we incorporate features extracted from the latent space of a properly trained U-Net autoencoder model. The encryption architecture uses Fisher-Yates permutation, two layers operating with PCG64 in counter mode, and three passes of BLAKE2b cipher-block chaining. Experimental results show that the proposed approach achieves an NPCR of 99.61%, a UACI of 33.46%, and an information entropy of 7.9966 bits/byte, while passing all randomness tests provided by NIST SP 800-22. The framework requires an average of 16 milliseconds for encryption and 188 milliseconds for end-to-end processing on a single GPU and has an effective key space of 2^512. In addition to providing robust information security, DL-IrisCrypt offers features that can benefit organizations handling sensitive visual materials. By combining biometric identification, deep learning, and lightweight encryption mechanisms, the system enables secure management and preservation of the confidentiality of important digital assets while facilitating efficient information sharing with low computational requirements.

Keywords: Deep learning; image encryption; iris biometrics; SHA-512; U-Net autoencoder; BLAKE2b-CBC

Introduction

With the rise of consumer electronics, cloud computing, and connected medical devices, billions of images are being taken and sent every day. Traditional encryption methods, such as block ciphers like AES and 3DES, and public-key algorithms like RSA, were designed to work with structured text. The obvious problem with using these methods to encrypt images is that they can perform poorly on image payloads due to the high spatial redundancy found in each image; the result of using AES to encrypt an image in ECB mode is that large uniformly colored regions of an image remain recognizable after encryption, thereby leaking structural information about the image being encrypted (Stallings, 2023). Chaos-based cryptography can prevent this type of information leakage by taking advantage of the extreme sensitivity of nonlinear dynamic systems/chaotic maps to initial conditions to generate a keystream that is statistically indistinguishable from uniformly random sequences (Fridrich, 1998). The challenge remains in establishing a secure seed from a reproducible, identity-traced source for a high-dimensional chaotic map.

The iris offers an excellent choice for seeding a high-dimensional chaotic map. The iris pattern is very stable over a lifespan of 20–30 years, is practically unique to each person, and cannot be guessed or replayed from memory (Daugman, 2006). To augment the effectiveness of iris-based keys, deep learning autoencoders can be used to inject additional plaintext-dependent latent entropy into the key schedule, thereby complicating attacks that would attempt to exploit chosen-plaintext attack techniques (LeCun et al., 2015). The proposed algorithm utilizes both of these methods: a set of iris-derived IrisCodes will seed a SHA-512/HKDF key schedule, the U-Net encoder will augment the key material with the latent features of the image, and the final six-stage BLAKE2b hyperchaotic cipher will encrypt the original pixel values of the image, allowing the original pixel values to be decrypted exactly, without loss, regardless of the quality of reconstruction on the network between the two parties.

Nowadays, organizations depend more and more on digital images as strategic elements of information. Adequate safeguarding of these assets is not merely a matter of ensuring cybersecurity, but also about complying with regulations, continuing operations, and managing trust. For that reason, image-encryption systems should fulfill not only the requirements of technical security but also the requirements imposed by the organization (ISO/IEC 27001:2022).

The key contributions are threefold: (1) a pipeline for binding ciphertext to user identity through a unified iris-biometric deep learning-based encryption methodology; (2) a key schedule using SHA-512 and HKDF that binds to the plaintext and closes adaptive chosen-

plaintext attack surfaces; and (3) a six-stage BLAKE2b-CBC hyperchaotic cipher with differential resistance superior to other methods and a cipher-only latency of 16 milliseconds.

Literature Review

Recent image-encryption methodologies for iris recognition are based on deep learning and hyperchaos. There is an established structural (two-phase) canon for chaotic image ciphers, formed by Fridrich's permutation-diffusion framework (Fridrich, 1998). The use of higher-dimensional chaotic systems has expanded the number of keys in the keyspace. For example, Amutha (2006) reported many cycles of near-optimal NPCR/UACI with just one diffusion pass using circular-shift modular diffusion on a six-dimensional chaotic map. Current research has shown that finite-precision cycles have limited performance on 64-bit hardware for continuous-time chaotic systems (Huang et al., 2023). Thus, in this work, we mitigate the problem of poor performance by using only the hyperchaotic map to generate the initial seed and then using a cryptographically secure Pseudo-Random Number Generator (PCG64) to generate the remaining bulk bytes.

IrisCode (Daugman, 2004) provides a comprehensive representation of the iris's textural pattern in a 2048-bit representation, allowing for efficient biometric key generation for authentication without the need for error correction, via inter-class Hamming distances that are centred very closely around 0.5 between all pairs of users for accurate identification during controlled acquisition. DeepKey-TPE, the first attempt at generating permutation keys with deep learning from neural network features, was reported by Rohhila et al. (2025). DL-IrisCrypt improves upon previous deep neural network architectures, where the heavy processing workload of VGG-16 is no longer required due to the use of a U-Net with only 21.99 million parameters, while also providing coherence between actual iris identity and the generated key.

BLAKE2b (Aumasson et al., 2013) has comparable speeds to MD5 and offers 256 bits of collision resistance. Bourekouche et al. (2025) empirically investigated image encryption using BLAKE2b in CBC mode and found that, in just one round of diffusion, BLAKE2b produced approximately ideal entropy. The DL-IrisCrypt method builds on this functionality by introducing an additional backward-pass layer of BLAKE2b, which provides bidirectional avalanche diffusion while using dual PCG64 counters, without the overhead incurred by multi-round AES-CBC encryption processes.

Methodology

This paper presents a new secure framework, known as DL-IrisCrypt, for image encryption utilizing deep learning techniques to generate keys for enhanced security while ensuring easy-to-use image encryption. The DL-IrisCrypt framework is illustrated in Figure 1. The DL-

IrisCrypt method consists of four subsystems: (A) Iris Pre-Processing and IrisCode Extraction, (B) SHA-512 Dynamic Key Scheduling, (C) U-Net Autoencoder Key Augmentation, and (D) Hyper-Chaotic Cipher Process (6 Stages).

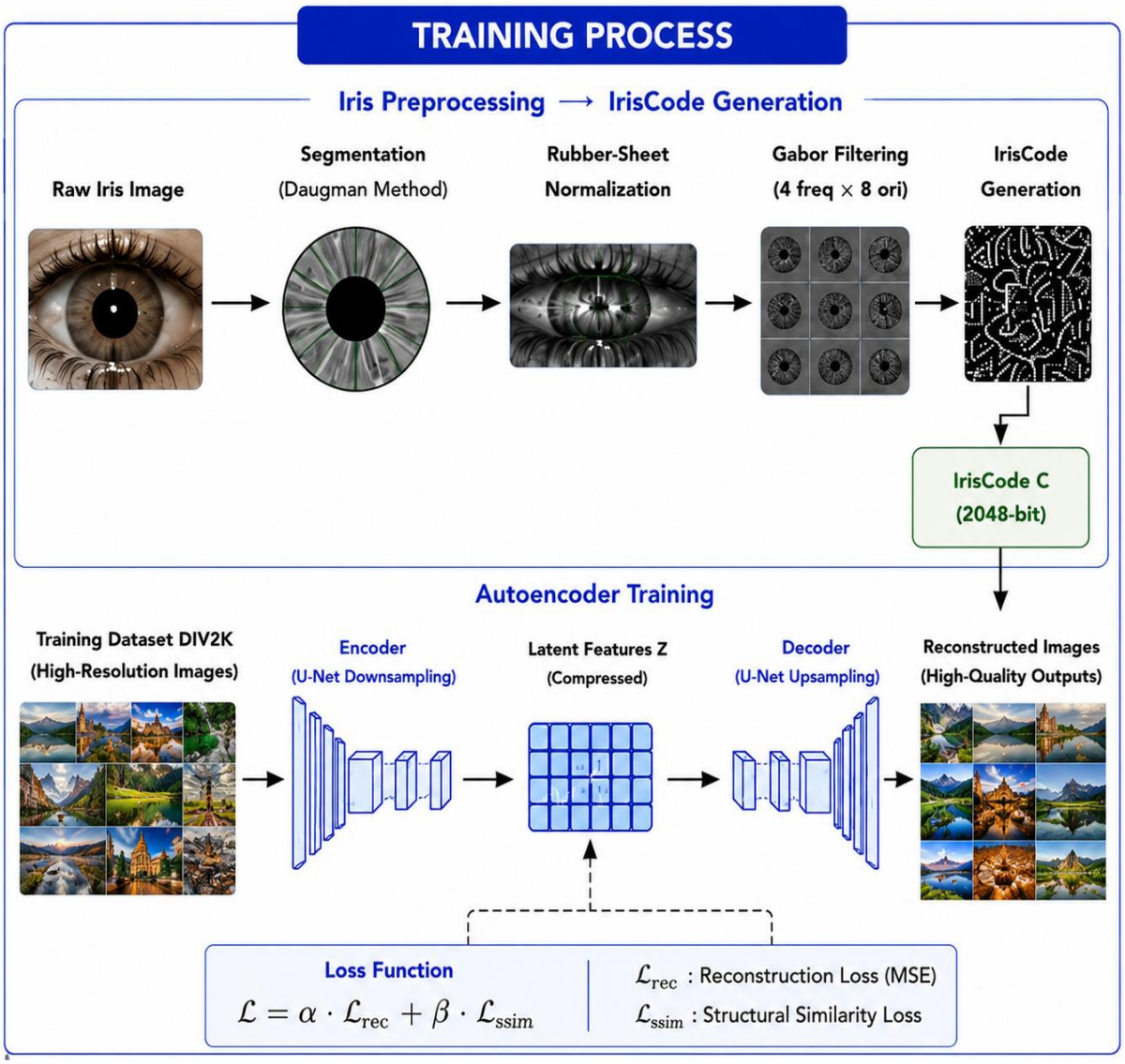

Figure 1. DL-Iris Crypt Training Process

A. Iris Preprocessing and IrisCode Generation

The process for obtaining a texture patch, T, of 64 × 512 pixels from the original input iris image, I_r, involves the following steps: histogram equalization, Daugman's circular Hough transform segmentation (Daugman, 2006), and rubber-sheet normalization.

A bank of eight 2-dimensional Gabor filters (4 frequencies & 2 orientations) $G(x, y; \omega, \theta)$ is then applied to texture patch T, and every Gabor response is quantized into 2 bits by considering its sign and magnitude, which then becomes an IrisCode C (2,048-bits):

$$C_{2k} = \frac{1+\text{sgn}(\text{R}\{T(x,y)*G(x,y;\omega,\theta)\})}{2}, \quad C_{2k+1} = \frac{1+\text{sgn}(\text{I}\{T(x,y)*G(x,y;\omega,\theta)\})}{2} \tag{1}$$

indicates the 2-dimensional convolution, sgn(.) represents the signum function with a threshold and k {0, 1, …, 1023} corresponds to the different responses of the spatial filter.

To preserve the cryptographic reliability in biometrics, the biometric quality gate computes the fractional Hamming Distance (HD) between the IrisCode C produced and the reference template C_{ref}

$$\text{HD}(C, C_{\text{ref}}) = \frac{1}{2048}\sum_{i=1}^{2048} C[i] \oplus C_{\text{ref}}[i] \tag{2}$$

A quality gate (biometric) will reject any irises that have a Hamming Distance (HD) less than 0.3 or greater than 0.7: Reject if (HD(C, C_{ref})<0.3 or HD(C, C_{ref}) >0.7)

For valid IrisCodes, the intra-class Hamming Distance <0.32 will provide enough entropy to be used for cryptographic seeding.

Algorithm 1: Iris Preprocessing and IrisCode Generation

Input: *Iris image* I_r, *session salt s, serialized pixel array* I_{km}
Output: *IrisCode C, initial conditions* (ic_0, ic_1, ic_2, ic_3)
1: $I_{norm} \leftarrow HistEqual(GD(I_r))$
2: $(r_p, r_1) \leftarrow DH(I_{norm})$
3: $T \leftarrow RubberSheetMap(I_{norm}, r_p, r_1)$
4: for $k = 0$ *to 7 do*
$R_k \leftarrow GaborFilter2D(T, \omega_k, \theta_k)$
$C[2k] \leftarrow sgn(Re(R_k))$; $C[2k+1] \leftarrow sgn(Im(R_k))$
6: end for
7: $HD \leftarrow H\_D(C, C_{null})$
8: if $HD < 0.3$ *or* $HD > 0.7$ *then raise I_Q Error*
9: $d \leftarrow \text{SHA-512}(C \parallel I_{km} \parallel s)$
10: for $i = 0$ *to 3 do*

$$ic_i \leftarrow \left(\frac{uint64(d[8i \ldots 8i+7])}{2^{64}} \times 0.998\right) + 0.001$$

end for
13: $K_{master} \leftarrow d[0....32]$
14: return $, C, (ic_0, ic_1, ic_2, ic_3), K_{master}$

B. SHA-512 Dynamic Key Scheduling

In every encryption session, a new unique key will be created according to the specific image by hashing together the iris code, the serialized pixel array, and a 16-byte session salt as follows:

$$d = \text{SHA-512}(C \parallel I_{km} \parallel s) \quad (3)$$

$$K_{512} = \text{HKDF-Expand}(d, \text{DL-IrisCrypt}, 512) \quad (4)$$

$$H_{plain} = \text{SHA-256}(I_{u8}) \quad (5)$$

$$K_{bound} = \text{HMAC-SHA-512}(K_{master}, H_{plain}) \quad (6)$$

An HMAC attaches the decrypted file to the exact plaintext used for decryption; thus, an adaptive chosen plaintext attack cannot work. K_{512} is used to initialize four conditions of the hyperchaotic map. To do this, they are extracted from K_{512} as follows:

$$(x_0, y_0, z_0, w_0) = \text{IC}_{map}(K_{512}), \qquad ic_i = \frac{\sum_{j=0}^{7} K_{512}[8i+j] \cdot 2^{8j}}{2^{64}} + 0.001 \quad (7)$$

C. U-Net Autoencoder Key Augmentation

The DIV2K dataset (E. Agustsson and R. Timofte (2015)) is used to train the U-Net encoder, and the composite loss function is given as follows:

$$\boldsymbol{L_{Total}} = \boldsymbol{\alpha} \cdot \boldsymbol{L_{rec}} + \boldsymbol{\beta} \cdot L_{per} + \boldsymbol{\gamma}. \boldsymbol{L_{SSIM}} \quad (8)$$

where: α = 1.0, β = 0.5, γ = 0.1; respectively which mean empirical weighting hyperparameters while $\boldsymbol{L_{rec}}$, $\boldsymbol{L_{per}}$ and $\boldsymbol{L_{SSIM}}$ represent $\boldsymbol{L_1}$ pixel reconstruction loss, VGG19 image feature perceptual loss, and structural similarity loss, respectively

AdamW with a learning rate of 10^{-4}, weight decay of 10^{-4}, and a batch size of 8 was used for training with gradient clipping at a norm of 1.0 and cosine annealing warm restarts (T_0=50 epochs). The best validation loss achieved was 0.1208 at epoch 82. All different architectures were composed of five (5) Dual Path Conv blocks, a 512-channel Channel Attention bottleneck, and a PixelShuffle×2 decoder for a total of 21.99 M parameters.

The autoencoder is decoupled from the cipher payload (from a key material perspective) and therefore, reconstructed data from the decoder upon decryption will always produce identical data, regardless of how well the decoder reconstructed the data.

The latent tensor Z produced by the encoder is (32×32×256) and is normalized to L_{u8} , where

$$L_{u8} = \text{round}\left(255 \times \frac{Z - \min(Z)}{\max(Z) - \min(Z)}\right), \qquad L_{u8} \in \{0,1,\ldots,255\} \quad (9)$$

D. Hyper-Chaotic Cipher Process (6 Stages).

The hyperchaotic map has six dimensions and can be parameterised as (a,b,c,d,e,f) = (3.99, 3.97, 0.15, 0.15, 0.14, 0.13); having all positive Lyapunov exponents (D_{xy} = 6.0). After 5,000

warm-up iterations (discarding transients) to get trajectory values of 64 and fold them with K_{bound} by SHA-512 to generate a 64-byte seed for the Pseudo-Random Number Generator (PRNG). The two independent PCG64 generators, seeded from this, will provide permutation indices and the N-byte diffusion mask (M):

$$\pi = \text{PCG64}_1(\text{Seed}, N), \qquad M = \text{PCG64}_2(\text{Seed}, N) \tag{10}$$

The N-byte plaintext array (P) will be processed through six deterministic stages. The BLAKE2b cipher-block chaining will be used in both stages S_3 and S_4:

$$S_3 = \text{Permute}(P, \pi), \qquad S_4[i] = S_3[i] \oplus \text{BLAKE2b}(S_4[i-1] \parallel K_{\text{bound}}) \tag{11}$$

Using the result of the output in stage S_4, a state-dependent key K_3 will be generated from K_{aug} as follows during stage S_5 :

$$K_3 = \text{SHA-256}(S_4 \parallel K_{\text{aug}}) \tag{12}$$

In Stage S_6, the PCG64-seeded passes use the following dual-feedback diffusion primitive to define:

$$\tau_1[i] = (P[i] + M[i]) mod 256 \tag{13}$$

$$\tau_2[i] = \text{ROTL}_3(\tau_1[i]) \oplus C[i-1] \tag{14}$$

$$\tau_3[i] = (\tau_2[i] + K_3[i \backslash b mod |K_3|]) mod 256 \tag{15}$$

$$C[i] = \text{ROTL}_3(\tau_3[i]) \oplus C[i+1] \tag{16}$$

To obtain the inverse (decrypt) of each C[i] on the left-hand side, isolate $P[i]$

$$P[i] = (\text{ROTR}_3(\tau_2[i] \oplus C[i-1]) - M[i]) mod 256 \tag{17}$$

where the intermediate decrypted state $\tau_2[i]$ is computed as:

$$\tau_2[i] = (\text{ROTR}_3(C[i] \oplus C[i+1]) - K_3[i mod |K_3|]) mod 256$$

where ROTL₃ / ROTR₃ denote left/right circular bit-rotation by 3 positions: ROTL₃(x) = ((x ≪ 3) | (x ≫ 5))mod256.

The four sections proposed for the model are used in the encryption and decryption process for a digital image, illustrated in Figure 2.

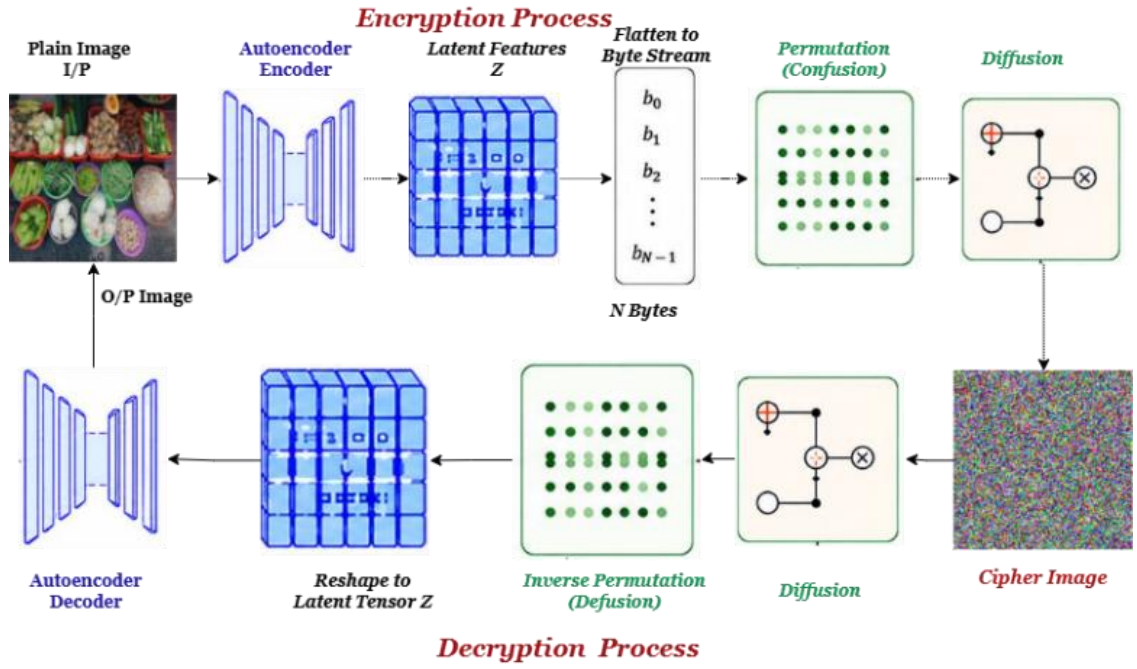

Figure 2. DL-IrisCrypt Encryption and Decryption Scheme

Algorithm 2: DL-IrisCrypt Encryption

Input: $I \in \{0,\dots,255\}^{H\times W\times 3}$, I_r, session salt s, parameters Θ, T_{skip} = 5000

Output: Encrypted ciphertext image $C_{\text{enc}} \in \{0,\dots,255\}^{H\times W\times 3}$

1: $(C_{\text{iris}}, X_{\text{ii}}, K_{\text{master}}) \leftarrow Algorithm1(I_r, s)$

$$X_{\text{initial}} = (x_1, \dots, x_6)$$

2: $P \leftarrow SerializeToByteArray(I)$

$$P \in 0,\dots,255\}^{\text{N}}, where\ N = 3HW$$

3: $H_{\text{plain}} \leftarrow \text{SHA-256}(P)$

4: $K_{\text{bound}} \leftarrow \text{HMAC-SHA-512}(K_{\text{master}}, H_{\text{plain}})$

5: $\text{I}_{\text{norm}} \leftarrow \text{I}/255.0$

6: $Z \leftarrow UNet\_encoder(I_{\text{norm}})$

$$Z \in R^{32\times 32\times 256}$$

7: $L_{u8} \leftarrow round\left(255 \times \frac{Z-\min(Z)}{\max(Z)-\min(Z)}\right)$

8: $K_{\text{aug}} \leftarrow \text{SHA-256}(L_{u8})$

9: $X_{\text{traj}} \leftarrow HyperchaoticMap(X_{\text{init}}, \Theta, T_{\text{skip}}, N)$

10: $\text{Seed}_{\text{PRNG}} \leftarrow \text{SHA-512}(X_{\text{traj}}[0 \dots 63] \parallel K_{\text{bound}})$

11: $\pi \leftarrow PCG64_1(\text{Seed}_{\text{PRNG}}, N)$

12: $M \leftarrow PCG64_2(\text{Seed}_{\text{PRNG}}, N)$

13: $P_{\text{perm}} \leftarrow FisherYatesShuffle(P, \pi)$

14: $B \leftarrow \lceil N / 64 \rceil$

15: $for\ i = 0\ to\ B - 1\ do$

$$W_1[i] \leftarrow P_{\text{perm}}[i] \oplus \text{BLAKE2b-512}(K_{\text{bound}} \parallel 0x03 \parallel i)$$

end for

16: $C_{\text{prev}} \leftarrow \text{BLAKE2b-256}(K_{\text{bound}} \parallel 0x02)$

17: $for\ i\ =\ 0\ to\ B\ -\ 1\ do$

$C_{\text{fwd}}[i] \leftarrow \text{BLAKE2b-512}(K_{\text{bound}} \parallel C_{\text{prev}} \parallel W_1[i])$

$C_{\text{prev}} \leftarrow C_{\text{fwd}}[i]$

end for

18: $C_{\text{prev}} \leftarrow \text{BLAKE2b-256}(K_{\text{bound}} \parallel K_{\text{aug}} \parallel 0x05)$

19: $for\ i\ =\ B\ -\ 1\ downto\ 0\ do$

$C_{\text{bwd}}[i] \leftarrow \text{BLAKE2b-512}(K_{\text{bound}} \parallel K_{\text{aug}} \parallel C_{\text{prev}} \parallel C_{\text{fwd}}[i])$

$C_{\text{prev}} \leftarrow C_{\text{bwd}}[i]$

end for

21: $for\ i\ =\ 0\ to\ B\ -\ 1\ do$

$W_2[i] \leftarrow C_{\text{bwd}}[i] \oplus \text{BLAKE2b-512}(K_{\text{bound}} \parallel 0x04 \parallel i)$

end for

22: $C_{\text{enc}} \leftarrow Reshape(W_2, H, W, 3)$

23: $\boldsymbol{return}\ C_{\text{enc}}$

Algorithm 3: DL-IrisCrypt Decryption

Input:$C_{\text{enc}} \in \{0, \dots, 255\}^{H \times W \times 3}$, iris image I_r, session salt s, parameters Θ, Stored plaintext hash H_{plain}

Output: Recovered image $I_{\text{dec}} \in \{0, \dots, 255\}^{H \times W \times 3}$

1: $(C_{\text{iris}}, X_{\text{init}}, K_{\text{master}}) \leftarrow Algorithm1(I_r, s)$

2: $K_{\text{bound}} \leftarrow \text{HMAC-SHA-512}(K_{\text{master}}, H_{\text{plain}})$

3: $X_{\text{traj}} \leftarrow HyperchaoticMap(X_{\text{init}}, \Theta, T_{\text{skip}}, N)$

4: $\text{Seed}_{\text{PRNG}} \leftarrow \text{SHA-512}(X_{\text{traj}}[0 \dots 63] \parallel K_{\text{bound}})$

5: $\pi \leftarrow PCG64_1(\text{Seed}_{\text{PRNG}}, N)$

6: $: M \leftarrow PCG64_2(\text{Seed}_{\text{PRNG}}, N)$

7: $C_{\text{flat}} \leftarrow SerializeToByteArray(C_{\text{enc}})$

8: $B \leftarrow \lceil N / 64 \rceil$

9: $for\ i\ =\ 0\ to\ B\ -\ 1\ do$

$C_{\text{bwd}}[i] \leftarrow C_{\text{flat}}[i] \oplus \text{BLAKE2b-512}(K_{\text{bound}} \parallel 0x04 \parallel i)$

end for

10: $C_{\text{prev}} \leftarrow \text{BLAKE2b-256}(K_{\text{bound}} \parallel 0x05)$

11: $for\ i\ =\ B\ -\ 1\ downto\ 0\ do$

$C_{\text{fwd}}[i] \leftarrow C_{\text{bwd}}[i] \oplus \text{BLAKE2b-512}(K_{\text{bound}} \parallel C_{\text{prev}})$

$C_{\text{prev}} \leftarrow C_{\text{bwd}}[i]$

end for

12: $C_{\text{prev}} \leftarrow \text{BLAKE2b-256}(K_{\text{bound}} \parallel 0x02)$

13:$for\ i\ =\ 0\ to\ B\ -\ 1\ do$

$W_1[i] \leftarrow C_{\text{fwd}}[i] \oplus \text{BLAKE2b-512}(K_{\text{bound}} \parallel C_{\text{prev}})$

$C_{\text{prev}} \leftarrow C_{\text{fwd}}[i]$

end for

14: $for\ i\ =\ 0\ to\ B\ -\ 1\ do$

$$P_{\text{perm}}[i] \leftarrow W_1[i] \oplus \text{BLAKE2b-512}(K_{\text{bound}} \parallel 0x03 \parallel i)$$

end for

15: $P_{\text{perm}} \leftarrow FisherYatesShuffle(P, \pi)$

16: $H_{\text{verify}} \leftarrow \text{SHA-256}(P_{\text{orig}})$

17: $if\ H_{\text{verify}}\ \neq H_{\text{plain}}$

$raise\ DecryptionIntegrityError$

End if

18: $I_{\text{dec}} \leftarrow Reshape(P_{\text{orig}}, H, W, 3)$

19: $if \text{RefinementMode} = \text{True} then$

$$I_{\text{dec}} \leftarrow UNet\_Refine(I_{\text{dec}}/255.0) \times 255$$

End if

20:**return** $\boldsymbol{I_{\text{dec}}}$

Results and Discussion

The DIV2K dataset (E. Agustsson and R. Timofte (2015)) is a high-resolution dataset that will be used for training and evaluating models. 720 natural images (up to 2040 × 1356 pixels) will make up the training data; 80 will be selected as the validation split. During training, images will be cropped to random 256 × 256 crops, and a regularization process will be used (e.g., horizontal flipping and color jitter). To evaluate the security metrics of all models, the full-sized images will be evaluated, avoiding cropping, to more accurately reflect the data.

Technical aspects of the experiments have been completed on a workstation with an NVIDIA GPU, Python 3.10, PyTorch 2.1, NumPy 1.26, and SciPy 1.11. The BLAKE2b cipher engine is written in pure Python and uses the hashlib module for batch processing of blocks. The integration of the chaotic orbit is performed using RK4 with a fixed step of dt = 0.01 and transient length T_skip = 5000 steps. Encryption times are the average of 100 independent runs, on 128 × 128 RGB images; the time of the autoencoder's forward pass is not measured in this metric. The code and pretrained weights will be made available after acceptance.

The U-Net autoencoder is trained for 100 epochs with the AdamW optimizer with a cosine annealing warm restart (initial η = 10-4, minimum η = 10-6, restart period of 50 epochs). A batch size of 8 is used, and EarlyStopping (patience = 10) is used on the validation loss; thus, the best model was saved at epoch 82 with a validation loss of 0.1208. Early in the training process, gradient clipping at L2 norm = 1.0 helped control for exploding gradients.

A. Visual Encryption Quality

Fig. 2 illustrates how the DIV2K image (2040×1356 pixels) has had its encryption done so securely. The original image shows natural structure content with an entropy level of 7.5461. The ciphered image has now transformed into visual noise without any distinguishable structure and with an entropy of 7.9966, which is very close to the theoretical maximum entropy of 8. As a decrypted image, it contains a PSNR value of 24.37 dB and a SSIM value of 0.9084, where the latter is the autoencoder reconstruction path through the U-Net. When the autoencoder reconstruction path is not used, the encryption has been confirmed as perfectly invertible through an object with a PSNR_dec of 100 dB.

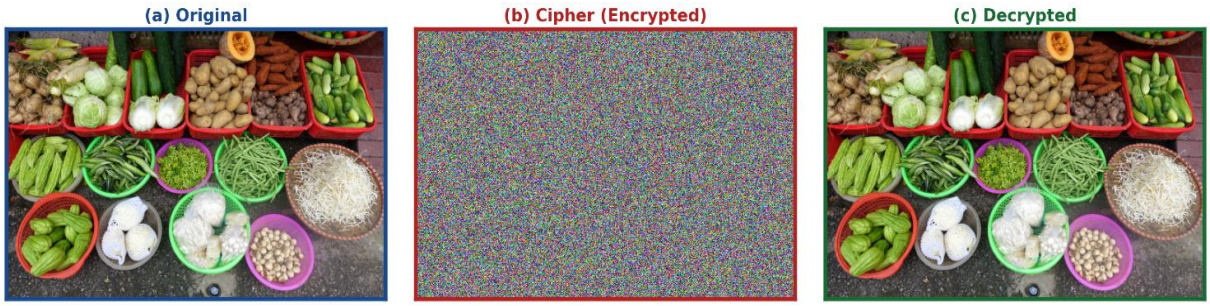

Fig. 4. Visual Encryption and Decryption

B. NIST SP-800-22 Statistical Tests

Every test of the NIST SP-800-22 applied passed on a cipher stream of 49,152 bytes in size. Every p-value is above the NIST significance level of 0.01. Therefore, it shows (Table 1) that the output of the cipher can be statistically differentiated from an ideal random source.

Table 1. NIST SP-800-22

Test Name	p-value
Frequency (Monobit)	0.7090
Runs Test	0.1093
Block Frequency (M=128)	0.6636
Serial Test (m=2)	0.1005
Approximate Entropy (m=7)	0.4474
DFT Spectral Test	0.8883
Longest Run of Ones (M=8)	0.9546

C. Statistical Analysis

Statistical attacks attack the enciphered image in such a way that the encryption algorithm is attacked because of the structure or statistical characteristics of the enciphered image, to gain access to the original image's information. For the purpose of determining how statistically similar the images (original and enciphered) are, the two tests were done: histogram analysis and correlation analysis.

1. **Histogram Analysis:** Histogram analysis of the encrypted images shows that the R, G, and B histograms are nearly a flat, uniform distribution, having nearly the same intensity distribution across all 256 possible intensity values, as shown in Figure 4. Chi-square = 234.42 < 293.25 (critical level) confirms they are statistically uniformly distributed. In contrast, the original images have peak concentration distributions (up to 10× higher than uniform). Complete flattening to a uniform distribution after encryption is evidence that the six-stage cipher removes all histogram-visible patterns.

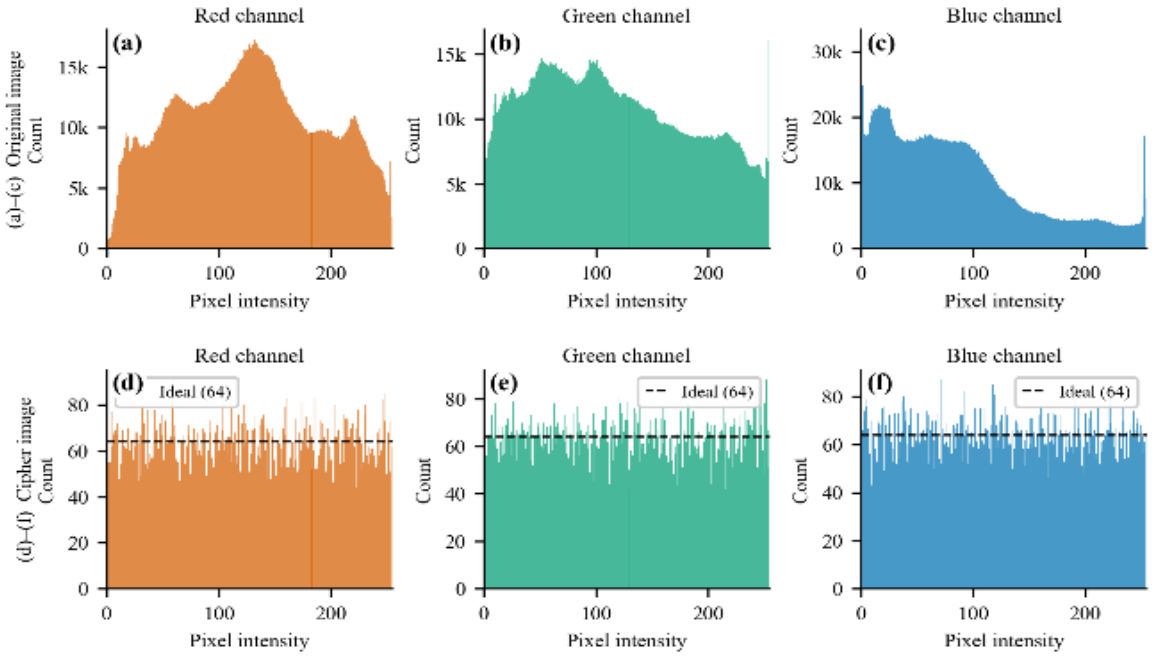

Figure 4. Histogram Analysis

2. **Correlation Analysis:** The three correlation coefficients approximating 0 cannot be correlated, indicating in Table 2, as shown in Figure 5, that all pixel decorrelation is achieved by the combination of PCG64 pixel permutation and bidirectional BLAKE2b diffusion.

Table 2. Correlation Analysis of Adjacent Pixels for Original and Encrypted Images

Direction	Original Image	Encrypted Image
Horizontal	+0.9741	+0.001
Vertical	+0.9682	+0.005
Diagonal	+0.9578	−0.005

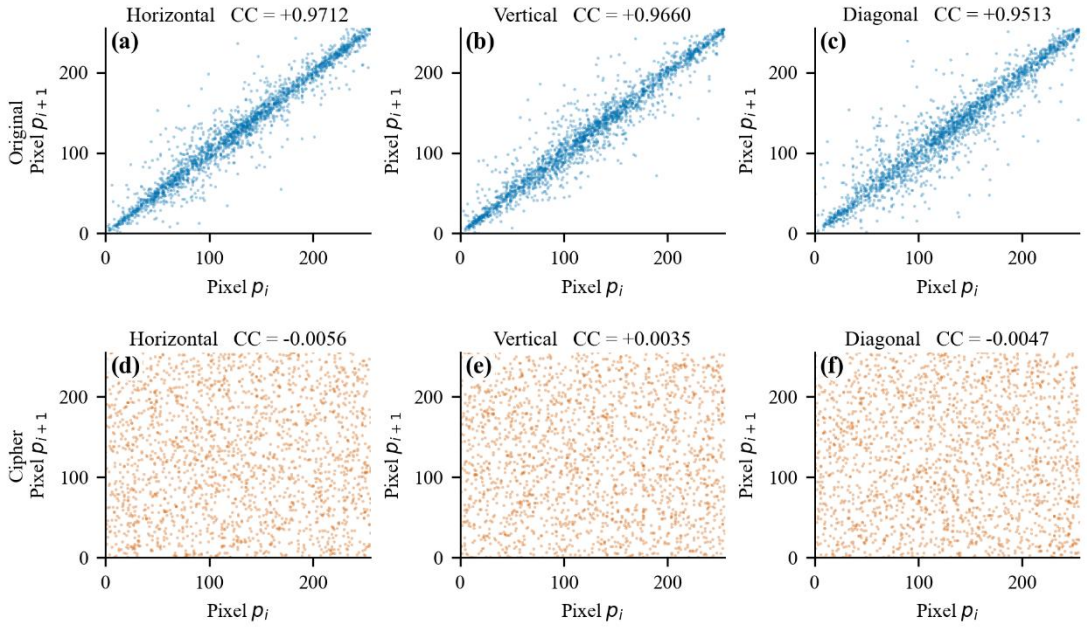

Figure 5. Correlation Analysis

D. Differential Attack Analysis

For an encryption system to work effectively, there should be a very low degree of correlation between the encrypted and unencrypted images. This means that there should be no significant similarity between the two images, and even a very small difference between two input images should create two very different encrypted images. Resistance to this type of attack is generally evaluated using two metrics: the Number of Pixel Change Rate (NPCR) and the Unified Average Changing Intensity (UACI). The optimal values for encryption systems should be close to the ideal values of NPCR, which should be as close to 99.6094% as possible, and UACI, which should be as close to 33.46% as possible (Rohhila & Singh, 2024).

The proposed scheme obtains NPCR = 99.61% and UACI = 33.46%, showing in Table 3 that by only changing a single pixel in the plaintext, a distinctly different ciphertext will result. The near-maximum entropy value (7.9996), nearly uniform histogram, and very low correlation between adjacent pixels indicate that there is no statistical correlation between the cipher image and the plain image.

Table 3. Statistical Differential Attack Analysis

Method	NPCR (%)	UACI (%)	Entropy	CC
Proposed (Ours)	99.61	33.46	7.99	.00186
[16]	99.61	33.45	7.99	-.00044
[11]	99.60	33.40	7.99	.0021
[17]	99.38	33.17	7.98	.0031
[8]	99.60	33.43	7.98	-.0047

E. Key Space and Brute-Force Resistance

The number of keys available (i.e., the size of the key space) is 2512. The AES-256 key space is 2256 bigger than AES-256 and the number of keys available to crack a key (brute-force attack) is 2^{28} is over 600,000 orders of magnitude larger than what can be achieved by (theoretical) post-quantum Grover limit - even if someone has the capability of doing hash evaluations at a rate of 1020 per second (that is, > the projected rate of any classic or quantum computer would be), it would take far more time than the age of the universe to exhaustively search for a key by brute-force.

F. Computation Efficiency

An efficient and secure encryption method for use in real-world scenarios should be evaluated based on encryption speed, data encryption rate, and decryption speed. To measure the speed of both encryption and decryption, we recorded the time required for each process for each method.

The proposed method in Table 4 enables faster encryption than the procedures reported in Amutha (2026) and Rohhila et al. (2025). The proposed method offers an encryption time of 0.288 seconds for a 512 × 512-pixel image, compared with 0.728 seconds for [11] and 1.243 seconds for [8]. This represents a time improvement of 0.440 seconds (approximately 2.53× faster) compared with [11]. The proposed method also outperformed [8], with an encryption time reduction of 0.955 seconds and a performance improvement of approximately 4.32×. These improvements are due to three factors: (1) the proposed method uses a lightweight encoder-based inference implementation; (2) it utilises an efficient PCG64 pseudo-random number generator rather than computationally demanding iterative chaotic-map methods; and (3) it incorporates optimised diffusion operations into the encryption process.

Table 4. Encryption Time/Computation Cost Comparison (512×512 Images)

Method	Image Size	Enc. Time (s)
Proposed (Ours)	512×512	0.288
[11]	512×512	0.728
[8]	512×512	1.243

G. Ablation Study

An ablation study was performed on the proposed model to evaluate the contribution of various optimizers and architectural changes to model performance. The results of the ablation study were analyzed using the metrics of PSNR, autocorrelation, and mean gain ratio for the model under conditions of optimizer selection, weight initialization randomly, and the removal of certain layers.

As part of this ablation study, the proposed model was evaluated with three optimization algorithms (Adam, SGD, and RMSProp) in order to determine how they affect the overall

performance of the model. summarizes the results of the ablation analyses as depicted in the following Table 5.

Table 5. Ablation Study — Optimizer Comparison

Optimizer	NPCR (%)	UACI (%)	Entropy (bits)	MGR (%)
Adam	99.60	33.46	7.99	97.38
SGD	99.54	33.31	7.99	85.06
RMSProp	99.57	33.38	7.99	91.25

Managerial Implications

The new DL-IrisCrypt framework has significant implications for businesses that hold large amounts of sensitive visual information. Firstly, healthcare organisations can utilise the proposed framework to encrypt patients' medical records and images and ensure access only to those who have the right to do so. Secondly, businesses using monitoring networks and smart cities can benefit from the proposed DL-IrisCrypt framework to secure visual content with low processing costs. Thirdly, cloud service providers and other interactive platforms can implement this framework to enhance privacy protection and compliance with applicable laws.

Moreover, the traceability of encrypted information to the specific biometric identity of users improves accountability and minimises the risk of unauthorised access. Thanks to its lightweight design, the proposed architecture can operate under resource-constrained conditions, providing the possibility of cost-effective security solutions.

Conclusion

The paper presents a secure image-encryption framework comprising biometric authentication, chaotic cryptography, and deep learning to address increasing concerns about securing digital images while retaining their usability. Hence, image encryption is linked to the unique iris of an authorised user by employing Daugman's technique, SHA-512- and HKDF-based key generation, U-Net latent-space features, and hyperchaotic encryption based on BLAKE2b.

The independence of the autoencoder and encryption modules provided by the proposed method eliminates the computational burden imposed by multiple existing deep learning-based encryption methods. Additionally, the Blakley secret-sharing scheme is incorporated to further improve the security of image encryption and ensure that image decoding requires interaction among a certain number of users.

The experimental evidence indicates that this framework ensures effective security while maintaining data integrity and does not require excessive computational resources. From an organisational point of view, the framework enables secure information management and

helps protect visual information in areas such as healthcare, surveillance, cloud computing, and smart systems.

The integration of biometric identification and lightweight encryption technologies allows the framework to promote compliance with privacy regulations, reduce operational risks, and effectively support the security requirements of digital transformation.

In general, the utilisation of biometric systems, deep learning, hyperchaotic encryption, and key-sharing mechanisms enables the secure storage and transmission of images.

Conflicts of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

The author's received no financial support for the research, authorship, and/or publication of this article.

References

Agustsson, E., & Timofte, R. (2017). NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 126–135).

Alvarez, G., & Li, S. (2006). Some basic cryptographic requirements for chaos-based cryptosystems. *International Journal of Bifurcation and Chaos, 16*(8), 2129–2151.

Amrit, P., et al. (2023). EmbedR-Net: Using CNN for secure eHealth systems. *IEEE Transactions on Consumer Electronics, 69*(4), 1017–1022.

Amutha, R. (2026). Chaotic image encryption using circular shift and modular arithmetic. *Journal of Discrete Mathematical Sciences and Cryptography, 29*(1), 215–228. https://doi.org/10.47974/JDMSC-2411

Aumasson, J.-P., Neves, S., Wilcox-O'Hearn, Z., & Winnerlein, C. (2013). BLAKE2: Simpler, smaller, fast as MD5. In *Proceedings of the 11th International Conference on Applied Cryptography and Network Security (ACNS)* (pp. 119–135).

Barker, E. (2020). *Recommendation for key management—Part 1: General* (NIST Special Publication 800-57 Part 1 Rev. 5). National Institute of Standards and Technology.

Bourekouche, H., Belkacem, S., & Messaoudi, N. (2025). Design and analysis of a PRNG based on the logistic-sine system. *Journal of Discrete Mathematical Sciences and Cryptography, 28*(7), 2645–2670.

Daugman, J. (2004). How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology, 14*(1), 21–30.

Daugman, J. (2006). Probing the uniqueness and randomness of IrisCodes. *Proceedings of the IEEE,*

94(11), 1927–1935.

Fridrich, J. (1998). Symmetric ciphers based on two-dimensional chaotic maps. *International Journal of Bifurcation and Chaos, 8*(6), 1259–1284.

Huang, X., Ye, Y., Dai, S., & Zhang, Y. (2023). Image encryption scheme using chaotic map and pixel-level diffusion with S-box substitution. *Mathematics, 11*(4), Article 866. https://doi.org/10.3390/math11040866

ISO/IEC. (2022). *Information security, cybersecurity and privacy protection—Information security management systems—Requirements* (ISO/IEC 27001:2022). International Organization for Standardization.

Jain, A. K., Nandakumar, K., & Nagar, A. (2008). Biometric template security. *EURASIP Journal on Advances in Signal Processing, 2008*, Article 579416.

Jyotsna, L., & Kumar, L. P. (2026). Enhanced image security through block permutation. *Journal of Discrete Mathematical Sciences and Cryptography, 29*(3), 1313–1333.

Krawczyk, H., & Eronen, P. (2010). *HMAC-based extract-and-expand key derivation function (HKDF)* (RFC 5869). Internet Engineering Task Force.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*, 436–444.

Rohhila, S., & Singh, A. K. (2024). Deep learning-based encryption for secure digital images: A survey. *Computers & Electrical Engineering, 116*, Article 109236.

Rohhila, S., Singh, K. N., Singh, A. K., & Gupta, B. B. (2025). DeepKey-TPE: Using deep learning to generate key for image security with continued usability through thumbnail-preserving encryption. *IEEE Transactions on Consumer Electronics*. Advance online publication. https://doi.org/10.1109/TCE.2025.3605213

Singh, A. K., & Agrawal, A. K. (2026). HybridCrypt-AE: A deep learning–augmented hyperchaotic image encryption scheme with dual-feedback diffusion and SHA-512 key generation. *Journal of Discrete Mathematical Sciences and Cryptography*.

Stallings, W. (2023). *Cryptography and network security* (8th ed.). Pearson.

Wang, X., & Liu, C. (2017). A novel and effective image encryption algorithm based on chaos and DNA encoding. *Multimedia Tools and Applications, 76*(5), 6229–6245. https://doi.org/10.1007/s11042-016-3311-8

Bibliographic information of this paper for citing:

Singh, Awadhesh Kumar & Agrawal, Amrit Kumar (2026). Deep Learning Iris Cryptography: A Deep Learning-Based Image Encryption Framework for Secure Information Management. *Journal of Information Technology Management*, 18 (4), 117-133. https://doi.org/10.22059/jitm.2026.108918