Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 29 (2026), No. 6, pp. 2521–2533
DOI : 10.47974/JDMSC-2761

An efficient CNN guided adaptive Rubik cube image encryption scheme

Prajakta Vijay Mankar [†]
Department of Applied Mathematics
Defence Institute of Advanced Technology (Deemed to be University)
Pune 411025
Maharashtra
India

Ruchi Telang Gode *
Department of Mathematics
National Defence Academy (NDA)
Pune 411023
Maharashtra
India

S. V. S. S. N. V. G. Krishna Murthy [§]
Department of Applied Mathematics
Defence Institute of Advanced Technology (Deemed to be University)
Pune 411025
Maharashtra
India

Abstract

In this work, we introduce a new method for image encryption that improves on the traditional Rubik cube based encryption model by integrating adaptive key generation using Convolutional Neural Networks (CNN) and Pseudo Random Number Generator (PRNG). Chaotic masking with Rubik cube based pixel permutation was employed in while effective, these masking employ static or random key vectors that are independent of the image content. This paper demonstrates that deep learning (CNN) guided adaptive key generation, coupled with robust permutation-diffusion architecture and non-chaotic (PRNG) masking, can deliver outstanding theoretical and empirical security in image encryption even without traditional chaotic maps. Our CNN guided enhancement dynamically generates permutation

[†] *E-mail:* **prajaktamankar8944@gmail.com**
[*] *E-mail:* **telang.ruchi82@gmail.com (Corresponding Author)**
[§] *E-mail:* **sgkmurthy@gmail.com**

vectors based on image features, introducing data adaptivity, larger key space, and enhanced security. This approach opens avenues for secure multimedia protection leveraging the frontier of AI driven cryptography and also reduces the time complexity. The proposed system achieves near ideal entropy, superior NPCR/UACI ratios, and high resilience to differential and statistical attacks while preserving computing efficiency suitable for real time systems. Experimental validation exhibits improvement over previous works in many instances.

Subject Classification: *94A60, 68P25, 68T07.*

Keywords: *Image encryption, Rubik cube, Pseudo random number generator, Convolution neural network, Statistical analysis, Differential attack.*

Introduction

Image encryption stands as one of the cornerstones of multimedia security in digital communication systems operating across increasingly distributed and vulnerable networks [1-3]. The encryption process fundamentally involves transforming plaintext visual information into un-intelligible ciphertext through deterministic mathematical operations and algorithmic transformations, rendering the image inaccessible to unauthorized parties lacking the appropriate cryptographic key for decryption. Digital images exhibit high spatial redundancy manifested as strong correlations between adjacent pixels, with typical correlation coefficients ranging from 0.90 to 0.98 on a scale where 0 indicates complete independence and 1 indicates perfect correlation. This extreme interdependence violates fundamental cryptographic principles requiring statistical independence between output symbols and input symbols, where changes in single input pixels should dramatically affect cipher text. Additionally, visual data exhibits characteristic frequency domain properties where certain frequency bands concentrate significant energy in lower frequencies for natural images, providing potential attack vectors through frequency analysis and Fourier based cryptanalysis. Consequently, the protection of digital images has become a prominent research focus [4, 5].

Survey of Modern Image Encryption Approaches

In order to produce pseudo random sequences for permutation and diffusion, early research mostly relied on chaotic maps. Tufillaro [6] began researching chaotic phenomena in wave dynamics as early as 1989. Because chaotic systems are sensitive to initial conditions, as well as their inherent randomness and deterministic behavior, chaos exhibits natural compatibility with cryptographic design. Most of the modern encryption schemes universally employ two stage confusion diffusion frameworks combining confusion stages that scramble pixel positions to disrupt spatial relationships with diffusion stages that distribute plain text information across entire cipher text through bitwise operations and mathematical transformations. However, one-dimensional maps suffer from limited key space and vulnerability to statistical attacks. Numerous

researchers have developed hybrid frameworks combining chaotic dynamics, neural networks, and pixel permutation-diffusion strategies [4, 5], including approaches based on Elliptic curves and bit-level permutation [7], DNA coding [8], Wavelet transforms [9], Game theory [10]. Surveys by [11, 12] systematically categorize chaos-based, neural-network-based, and hybrid encryption schemes. These surveys emphasize the advantages of deep learning in adaptive and real-time encryption for IoT environments.

CNN-Based Hybrid Models

Since the concept of neural networks was first introduced in the last century [13, 14], continuous advancements in computer hardware have greatly facilitated their practical implementation. As a result, numerous neural network models have been developed. Among these in 1980s and 1990s, research on Convolutional Neural Networks (CNNs) started, integrating Convolutional Neural Network (CNN) with chaos marks a significant advancement. CNNs [12, 15] are capable of extracting and learning features from target images, making them widely applicable in fields such as computer vision, natural language processing, image classification, image reconstruction, and various other domains. CNN generates chaotic pointers associated to plaintext. The CNN dynamically controls pixel scrambling by extracting characteristics from chaotic sequences used as convolution kernels, which makes the system extremely resilient to known-plaintext and chosen-plaintext attacks. For any robust image encryption system, achieving high security is a fundamental design requirement. However, many chaos-based image encryption algorithms have been shown to be vulnerable under cryptanalysis [16]. In particular, some schemes lack sufficient sensitivity to changes in plaintext, rendering them susceptible to known-plaintext and chosen-plaintext attacks, problems commonly observed in this domain. Recent studies further reveal that CNNs have been successfully used to compromise optical image encryption systems [12]. The integration of CNNs into a plaintext-dependent scrambling method is investigated in this study in order to overcome these security issues. Results from experiments show that the suggested picture encryption method linked to plaintext can effectively resist such attacks. Man et al. [17] introduced a double-image encryption algorithm combining CNN with a 5-D hyper-chaotic system. The chaotic sequence acts as CNN convolution kernels for scrambling, enhancing resistance to known-plaintext attacks. Ding et al. [18] proposed DLEDNet using a Cycle GAN architecture for medical IoMT applications in which medical images are transformed into a hidden domain and reconstructed perfectly (PSNR >40 dB), though at the cost of high computational complexity. Recent studies have explored the integration of CNNs with chaotic maps for image encryption. For example, fused logistic chaotic maps with a CNN-driven block encryption framework, while [19] utilized CNNs directly for image encryption and reported high NPCR values.

Rubik Cube based chaotic encryption

Chaos based and Rubik cube oriented encryption schemes have been widely studied for their strong confusion and diffusion properties. Recently [1, 2, 3] developed processes combining Rubik Cube pixel permutation with a chaotic masking technique to achieve robust encryption with high NPCR and UACI values. Their algorithms demonstrated security against differential and statistical threats but was limited by its use of fixed or random keys. Recent research has shifted toward intelligent cryptography, where neural networks especially CNNs can adaptively generate encryption keys or patterns to enhance unpredictability. Our work builds on this idea by embedding a CNN within the Rubik Cube encryption pipeline.

Pseudo-Random sequence

Pseudo-random sequence [20] is a deterministic sequence with random characteristics. A Pseudo-Random Number Generator (PRNG) is used in cryptography to create a series of key streams that guarantee communication security. Repeatability, extensibility, periodicity, and adaptability are characteristics of pseudo-random sequences that set it apart from random sequences.

Proposed Approach and System Architecture

To address the aforementioned security issues while ensuring strong protection and computational efficiency, we provide a newly developed dual-channel image encryption method that relies on adaptive neural-guided permutation and PRNG-driven diffusion. The proposed framework follows a structured permutation–diffusion architecture that integrates CNN-based key generation with Rubik cube-inspired permutation and pseudo-random diffusion.

The following is a summary of this work's primary contributions:

1. **Non-Chaotic Neural-Guided Key Generation:** The normalized input image is converted into two adaptive permutation vectors that correspond to row-wise and column-wise circular shifts using a lightweight Convolutional Neural Network (CNN). These vectors function as plaintext-dependent keys which ensure great sensitivity to picture content, significantly expand the effective key space, and fortify defences against brute-force and known-plaintext threats.
2. **Rubik Cube-Based Dual Permutation (Confusion):** Using the CNN-generated adaptive vectors, a Rubik cube-inspired permutation scheme performs circular shifts across rows and columns to disrupt spatial correlations among adjacent pixels. The permutation stage is executed twice using the same adaptive keys to further enhance confusion and eliminate residual correlations.
3. **PRNG-Based Diffusion Mechanism:** A Pseudo-Random Number Generator (PRNG) initialized with a fixed seed generates a diffusion mask.

The mask is XORed with the permuted image, followed by cascading neighborhood XOR operations to propagate pixel modifications throughout the image. This diffusion strategy ensures strong resistance to differential attacks and achieves high NPCR and UACI values while maintaining computational efficiency.

The encryption process operates on gray scale images or separately on RGB channels to prevent inter-channel correlation leakage. For each channel, a CNN generates adaptive row and column permutation vectors from the pixel intensity distribution, which are applied twice using a Rubik Cube–inspired permutation to achieve strong spatial confusion. Global pixel dependency is then enforced through PRNG-based diffusion with cascading XOR and random masking. Decryption follows the exact inverse sequence using the same adaptive keys, ensuring precise reconstruction of the original image.

The remainder of this paper is organized as follows: the *Proposed Method* section presents the detailed mathematical formulation of the method, the *Experimental Results and Comparative Analysis* section discusses experimental evaluation, and the *Conclusion* section summarizes the findings and outlines directions for further research.

Proposed Method

Proposed CNN-PRNG Rubik Cube Image Encryption Framework

This section introduces a hybrid image encryption framework combining CNN-based adaptive key generation with PRNG-driven diffusion under Rubik Cube permutation principles. By replacing static, image-independent chaotic keys with content-adaptive permutations, the effective key space is expanded from 2^{64}-2^{128} to approximately 2^{256}.

Notation and Framework Overview

Let I denote the input image of size $H \times W$, where pixel values lie in the range $[0, 255]$. For gray scale images, $I \in \mathbb{R}^{H \times W}$, whereas for color images, each RGB channel is processed independently.

The proposed encryption framework consists of three main stages: (i) CNN-based adaptive permutation key generation, (ii) Double Rubik cube permutation (confusion), and (iii) PRNG-based cascade diffusion.

Mathematical Notations: R_i and C_j denote the row and column circular shift values ($i = 0, \ldots, H-1;\ j = 0, \ldots, W-1$), K_{CNN} represents the CNN-generated adaptive key, P denotes the permuted image, M is the PRNG diffusion mask initialized with seed S, and I_e denotes the final encrypted image.

CNN-Based Adaptive Key Generation

A two-dimensional convolutional neural network (CNN) applies spatial convolution as $\mathbf{Y} = \mathbf{X} * \mathbf{W} + \mathbf{b}$,, where * denotes 2D convolution, followed by the

rectified linear unit $\mathrm{ReLU}(x) = max(0, x)$.

The CNN architecture is defined as

$$\mathrm{Conv2D}(1 \to 16, 3 \times 3) \xrightarrow{\mathrm{ReLU}} \mathrm{MaxPool}(2 \times 2) \xrightarrow{\mathrm{FC}} \mathbb{R}^{HW},$$

producing the adaptive key $K_{\mathrm{CNN}} * \mathrm{R}^{HW}$.

The permutation vectors are generated as $R_i = \lfloor |K_{\mathrm{CNN}}[i]| \rfloor \bmod W, i = 0, \ldots, H-1$, and $C_j = \lfloor |K_{\mathrm{CNN}}[H+j]| \rfloor \bmod H$, where $j = 0, \ldots, W-1$.

Rubik Cube Permutation Mechanism

CNN-generated keys drive circular row- and column-wise Rubik Cube permutations, effectively destroying spatial correlations.

PRNG-Enhanced Cascade Diffusion

Diffusion is performed using the Mersenne Twister PRNG (MT19937) seeded by both image statistics and CNN features.

Decryption and Reversibility

Decryption strictly follows the inverse order of encryption.

Algorithm 1: Encryption

Input: Plain image I (gray scale or color), **Output:** Encrypted image I_e and keys $\{R, C, S\}$.

Procedure:

Image Preprocessing:

- **If I is color:** Split into R, G, B channels: I_R, I_G, I_B
- **If I is grayscale:** Use as single channel: I_{gray}

2. **For each channel** I_c (color: 3 channels, grayscale: 1 channel):

(a) CNN Key Generation:

- Normalize: $\hat{I}_{c,i,j} = I_{c,i,j}/255$
- Pass through CNN: $K_{\mathrm{CNN}} = \mathrm{CNN}(\hat{I}_c)$
- Extract permutation vectors:

$R_i = \lfloor |K_{\mathrm{CNN},i}| \rfloor \bmod W$ for $i = 0, \ldots, H-1$,

$C_j = \lfloor |K_{\mathrm{CNN},H+j}| \rfloor \bmod H$ for $j = 0, \ldots, W-1$.

(b) First Rubik Cube Permutation:

Row-wise circular shift: $P_{i,j} = I_{c,i,(j-R_i) \bmod W}$

Column-wise circular shift: $P_{i,j} = P^{row}_{(i-C_j) \bmod H, j}$

(c) PRNG Seed Generation**:**

$$\boldsymbol{S = \left(\sum_{i=0}^{H-1} \sum_{j=0}^{W-1} I_{c,i,j} + \sum_{k=0}^{H+W-1} |K_{\mathrm{CNN},k}| \right) \times 10^6 \bmod 2^{32}}$$

(d) Cascade Diffusion:

$$\boldsymbol{E_{i,j}} = \begin{cases} \boldsymbol{P_{i,j}} \oplus \boldsymbol{M_{i,j}}, & \boldsymbol{i = 0,\ j = 0.} \\ \boldsymbol{P_{i,j}} \oplus \boldsymbol{M_{i,j}} \oplus \boldsymbol{E_{i-1,j}}, & \boldsymbol{i > 0,\ j = 0.} \\ \boldsymbol{P_{i,j}} \oplus \boldsymbol{M_{i,j}} \oplus \boldsymbol{E_{i,j-1}}, & \boldsymbol{i = 0,\ j > 0.} \\ \boldsymbol{P_{i,j}} \oplus \boldsymbol{M_{i,j}} \oplus \boldsymbol{E_{i-1,j}} \oplus \boldsymbol{E_{i,j-1}}, & \text{otherwise.} \end{cases}$$

(e) Second Rubik Cube Permutation**:**

$$\text{Row} - \text{wise circular shift} : \mathrm{C}^{\mathrm{row}}_{\mathrm{c,i,j}} = \mathrm{E}_{\mathrm{i,(j-R_i)modW}}$$

$$\text{Column} - \text{wise circular shift:}\ \mathrm{C}_{\mathrm{c,i,j}} = \mathrm{C}^{\mathrm{row}}_{\mathrm{c,(i-C_j)modH,j}}$$

(f) Store encrypted channel**:** $E_c = C_c$

3. **Final Output:**
   - **If color:** $I_e = \text{merge}(E_R, E_G, E_B)$
   - **If gray scale:** $I_e = E_{\text{gray}}$
   - **Keys:** $\{R_i, C_j, S\}$ **c**hannel

The overall framework of the encryption and decryption processes is illustrated in Figure 1.

Algorithm 2: Decryption

Input: Encrypted image I_e and keys$\{R_i, C_j, S\}$, **Output:** Decrypted image I_d

Procedure:

1. **Image Preprocessing:**
   - **If I_e is color:** Split into cipher channels C_R, C_G, C_B
   - **If I_e is grayscale:** Use as single cipher channel: C_{gray}
2. **For each cipher channel** C_c (color: 3 channels, grayscale: 1 channel):
   (a) **First Inverse Rubik Permutation**(inverse of first permutation):
   Column-wise inverse shift: $E^{col}_{i,j} = C_{c,(i+C_j)\ mod\ H,j}$
   Row-wise inverse shift: $E_{i,j} = E^{col}_{i,(j+R_i)\ mod\ W}$
   (b) **Inverse Cascade Diffusion:**
      i. Regenerate PRNG mask M using MT19937 with seed S
      ii. Apply inverse diffusion (reverse raster scan):

$$P_{i,j} = \begin{cases} E_{i,j} \oplus M_{i,j}, & i = 0,\ j = 0. \\ E_{i,j} \oplus M_{i,j} \oplus P_{i-1,j}, & i > 0,\ j = 0. \\ E_{i,j} \oplus M_{i,j} \oplus P_{i,j-1}, & i = 0,\ j > 0. \\ E_{i,j} \oplus M_{i,j} \oplus P_{i-1,j} \oplus P_{i,j-1}, & \text{otherwise.} \end{cases}$$

   (c) **Second Inverse Rubik Permutation** (Inverse of first permutation):
   Column-wise inverse shift: $P^{col}_{i,j} = P_{(i+C_j)\ mod\ H,j}$
   Row-wise inverse shift: $I_{c,i,j} = P^{col}_{i,(j+R_i) mod\ W}$
   (d) **Store decrypted channel:** $D_c = I_c$.

3. **Final Output:**
   - **If color:** Merge channels: $I_d = \text{merge}(D_R, D_G, D_B)$.
   - **If gray scale:** $I_d = D_{\text{gray}}$.

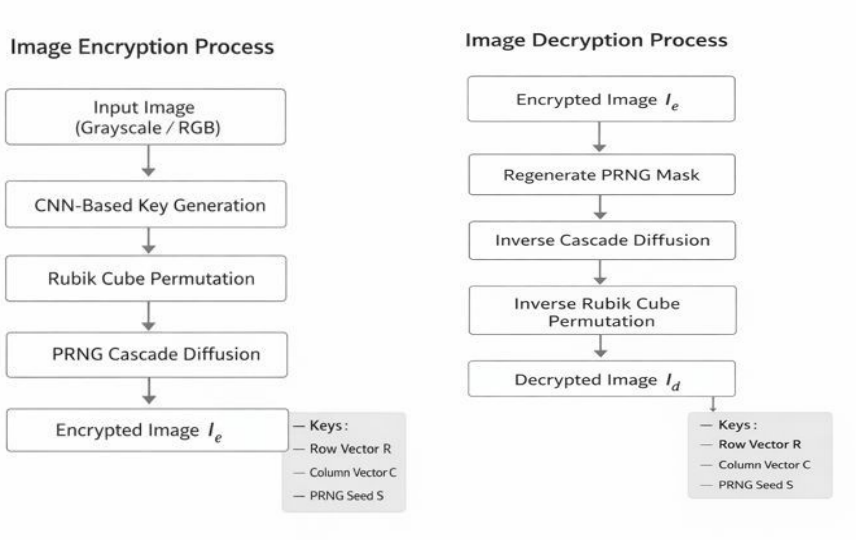

Figure 1
Workflow of (a) Encryption Process, (b) Decryption Process

Key Features of the Proposed CNN–PRNG Encryption Framework

- **Plaintext-Dependent Adaptive Key Generation:** Row and column permutation vectors are generated directly from the plaintext image using a CNN, ensuring that even visually similar images produce entirely different encryption keys and providing robust defence against chosen-plaintext and known-plaintext assaults.
- **Non-Chaotic yet Cryptographically Secure Design:** The scheme avoids continuous chaotic maps and their finite-precision and parameter-reconstruction vulnerabilities by integrating CNN-driven key adaptivity with a high-quality PRNG (MT19937), ensuring strong randomness and numerical stability.
- **Double-Stage Rubik Cube Permutation (Enhanced Confusion):** Two rounds of Rubik Cube–based permutation applied before and after diffusion effectively eliminate spatial correlations and prevent partial structural leakage in the cipher image.
- **PRNG-Based Cascade Diffusion Mechanism:** PRNG-generated masking combined with neighborhood-based cascading XOR procedures guarantees

that a single plaintext pixel change spreads over the whole cypher picture, producing high UACI and NPCR values as well as robust defence against differential assaults.

- **Large Effective Key Space:** The combined influence of CNN parameters, image-dependent feature activations, and PRNG seeds produces an effective key space exceeding 2^{256}, rendering brute-force attacks computationally infeasible.
- **High Statistical Security:** Strong resilience to statistical and entropy-based attacks is demonstrated by encrypted images with uniform histograms, near-zero correlation coefficients in all directions, and entropy levels around the theoretical maximum of 8.
- **Robustness against Noise and Data Loss:** The scheme maintains acceptable decryption quality under common channel impairments such as salt-and-pepper noise and image clipping, demonstrating suitability for real-world transmission environments.
- **Computational Efficiency and Practical Deployability:** With linear time complexity $O(n^2)$ and no computationally expensive chaotic iterations, the methodology fits well on CPU and GPU platforms for real-time and hardware-accelerated implementations.

Experimental Results and Comparative Analysis

Experimental Setup

All experiments were conducted using Google Colab, which provides a high-performance computing environment equipped with GPU acceleration, high-speed CPU, and ample RAM. The implementation was carried out using Python 3.9 with PyTorch 2.0 and Open CV 4.7 libraries. Performance evaluation was performed on standard gray scale (512 × 512) and color (256 × 256) test images.

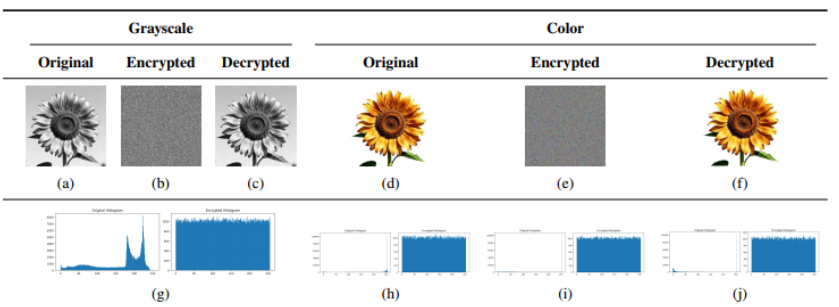

Figure 2

Encryption results and histogram analysis for the Flower image (512×512). (a)–(c) Grayscale original, encrypted, and decrypted images (d)–(f) Color original, encrypted, and decrypted images. (g) Grayscale histogram; (h)–(j) histograms of red, green, and blue channels

Visual Encryption Result

Figure 2 presents the encryption results and histogram analysis for the Flower image (512 × 512), including the original, encrypted, and decrypted grayscale images, the corresponding color images, and their respective histograms.

Variance Analysis

Variance measures the dispersion of pixel intensities around the mean, quantifying the deviation of pixel values from the average intensity. For encrypted images, low variance indicates non uniformity and potential security weaknesses, whereas values approaching the theoretical maximum imply strong encryption. The statistical security analysis results of encrypted grayscale and color images are summarized in Table 1.

For an $m \times n$ image with α-bit gray levels, the variance is computed as

$$\mathrm{Var}(I) = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(I_{i,j} - \mu\right)^2$$

Where, $I_{i,j}$ denotes the pixel intensity at position (i, j), and the mean pixel intensity is defined as

$$\mu = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{i,j}.$$

Table 1
Statistical security analysis of encrypted gray scale (512×512) and color (256×256) images

Metric	Cat Gray	Flower Gray	Cat R	Cat G	Cat B	Flower R	Flower G	Flower B
Variance	5453.2755	5452.1547	5486.3936	5477.0157	5411.3550	5460.2943	5463.6817	5459.4692
Entropy	7.9974	7.9993	7.9975	7.9973	7.9970	7.9992	7.9992	7.9993
Correlation (H)	0.005372	0.000093	0.002901	0.007158	0.006890	0.002612	0.001494	0.000388
Correlation (V)	0.001973	0.001741	0.006565	0.000886	0.002877	0.000795	0.002667	0.003301
Correlation (D)	0.001293	0.002748	0.000538	0.000186	0.003769	0.002676	0.000043	0.001801
NPCR (%)	99.6048	99.6067	99.6094	99.6216	99.6231	99.6094	99.5914	99.6193
UACI (%)	33.6030	33.4734	33.4519	33.4827	33.2996	33.4140	33.3717	33.4286

Table 2

Performance comparison of different encryption schemes on gray scale and color images

	Gray scale (512×512 – Flower)				Color (256×256 – Cat)			
Metric	**[7]**	**[18]**	**[8]**	**Proposed**	**[7]**	**[18]**	**[8]**	**Proposed**
Horizontal Correlation	NR	0.00225	0.0000	**0.000093**	NR	NR	NR	**0.00353**
Vertical Correlation	NR	-0.00125	-0.00023	**0.001741**	NR	NR	NR	**0.00353**
Diagonal Correlation	NR	0.0016	0.00067	**0.002748**	NR	NR	NR	**0.00353**
NPCR (%)	99.62	99.16	99.623	**99.6067**	99.62	NR	99.64	**99.6180**
UACI (%)	33.47	31.53	33.51	**33.4734**	33.47	NR	33.55	**33.4114**
Entropy	7.999	7.997	7.9994	**7.9993**	7.999	NR	7.9973	**7.9973**
Encryption Time (s)	0.27	NR	0.762	**0.7272**	0.27	NR	0.576	**0.1118**
Decryption Time (s)	0.27	NR	0.784	**0.0486**	0.27	NR	0.584	**0.0486**
Throughput (MB/s)	NR	NR	NR	**223.8**	NR	NR	NR	**223.8**
Key Space	$> 2^{26}$	NR	2^{128}	$> 2^{256}$	$> 2^{26}$	NR	2^{128}	$> 2^{256}$
Training Requirement	NR	NR	NR	**No**	NR	NR	NR	**No**
Real-Time Capability	NR	NR	NR	**Excellent**	NR	NR	NR	**Excellent**

Table 2 presents the performance comparison of the proposed encryption scheme with existing state-of-the-art methods.

Conclusion

This paper presented a CNN-guided adaptive Rubik Cube-based image encryption scheme that integrates machine learning-driven key generation with chaotic permutation [19, 20]. The proposed method achieves high randomness, strong sensitivity, and robustness against cryptographic attacks while maintaining low computational complexity. Extensive evaluation across six security and performance metrics confirms its effectiveness for real-time image encryption. The scheme is well-suited for applications in medical imaging, defense systems, and smart city surveillance, with future extensions toward parallel multi-image encryption and post-quantum security.

References

[1] L. Xu, Y. Chen, Y. Qin, and Z. Yang, "A mixed chaotic image encryption method based on parallel rotation scrambling in Rubik's cube space," *Entropy*, vol. 27, no. 6, p. 574 (2025), doi: 10.3390/e27060574.

[2] M. Palav, R. T. Gode, and S. V. S. S. N. V. G. Krishna Murthy, "An efficient image encryption scheme combining Rubik's cube principle with masking," in *Neural Networks, Machine Learning, and Image Processing*. Boca Raton, FL, USA: CRC Press (Taylor & Francis), pp. 181–198 (2022).

[3] S. Sharmila, R. S. Bhuvaneswaran, and D. Vaithiyanathan, "Secure image encryption using Rubik's cube-based scrambling with chaos-driven diffusion and circular shifts," *Optik,* vol. 339 (2025), doi: 10.1016/j.ijleo.2025.172533.

[4] C. S. Joice, S. H. Shanthini, K. Harsha, and S. Gopika, "A comprehensive review of image encryption algorithms: Techniques and applications," in *Proc. 5th Int. Conf. Data Intelligence and Cognitive Informatics (ICDICI)*, Tirunelveli, India, pp. 920–926 (2024), doi: 10.1109/ICDICI62993.2024.10810878.

[5] A. S. Sajitha and A. Shobha Rekh, "Review on various image encryption schemes," *Materials Today: Proceedings,* vol. 58, no. 1, pp. 529–534 (2022), doi: 10.1016/j.matpr.2022.03.058.

[6] N. Tufillaro, "Nonlinear and chaotic string vibrations," *American Journal of Physics,* vol. 57, no. 5, pp. 408–414 (1989).

[7] M. Palav, R. T. Gode, and S. V. S. S. N. V. G. Krishna Murthy, "A masking-based image encryption scheme using chaotic map and elliptic curve cryptography," in Mathematical Modeling, Computational Intelligence Techniques and Renewable Energy, vol. 1287, *Advances in Intelligent Systems and Computing*, Singapore: Springer, pp. 131–145 (2021).

[8] S. Zhu and C. Zhu, "Secure image encryption algorithm based on hyperchaos and dynamic DNA coding," *Entropy*, vol. 22, no. 7, p. 772 (2020), doi: 10.3390/e22070772.

[9] W. Hu, R. Zhou, S. Jiang, and G. Luo, "Quantum image encryption algorithm based on Arnold scrambling and wavelet transforms," *Quantum Information Processing,* vol. 19, no. 3, pp. 1–29 (2020).

[10] J. Zhou, J. Li, and X. Di, "A novel lossless medical image encryption scheme based on game theory with optimized ROI parameters and hidden ROI position," *IEEE Access*, vol. 8, pp. 122210–122228 (2020), doi: 10.1109/ACCESS.2020.3007550.

[11] M. Kaur and V. Kumar, "Comprehensive review on image encryption techniques," *Archives of Computational Methods in Engineering*, vol. 27, no. 1, pp. 15–43 (2020), doi: 10.1007/s11831-018-9298-8.

[12] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," arXiv preprint arXiv:1511.08458, (Nov. 2015). [Online]. Available: https://arxiv.org/abs/1511.08458.

[13] V. Kaur, D. P. Bhatt, P. K. Tiwari, and S. Tharewal, "Blockchain technology combined with the CNN and hashing algorithms enabled the secure storage of 3D biometric face and ear data," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 26, no. 3, pp. 729–738 (2023), doi: 10.47974/JDMSC-1745.

[14] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, no. 5, pp. 826–834 (Oct. 1983).

[15] S. N. Bhagirath, V. Bhatnagar, and L. Raja, "A comparative analysis of optimized CNN models using bio-inspired algorithms," *Journal of Information and Optimization Sciences*, vol. 46, no. 2, pp. 531–540 (2025), doi: 10.47974/JIOS-1957.

[16] B. Zhang and L. Liu, "Chaos-based image encryption: Review, application, and challenges," *Mathematics*, vol. 11, no. 11 (2021), doi: 10.3390/math11112585.

[17] 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 (2021), doi: 10.1016/j.chaos.2021.111318.

[18] Y. Ding, G. Wu, D. Chen, N. Zhang, L. Gong, and M. Cao, "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 (Feb. 2021).

[19] S. Kumaran, R. Aarthi V. Arthi, and G. Hemalatha, "Image encryption using convolutional neural network," *ITM Web of Conferences*, vol. 56 (2023), doi: 10.1051/itmconf/20235605005.

[20] L. Pasqualini and M. Parton, "Pseudo random number generation: A reinforcement learning approach," *Procedia Computer Science*, vol. 170, pp. 1122–1127 (2020), doi: 10.1016/j.procs.2020.03.057.

Received March, 2026