

Research Article

Performance Evaluation of Autoencoder and JPEG-Based Image Compression Schemes Integrated with AES and RSA Encryption

Zekeriya KATILMIŞ^{1a}, Şevval KOÇ^{2b}¹Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Kutahya Health Sciences University, Kütahya, Türkiye²Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Kutahya Health Sciences University, Kütahya, Türkiye

zekeriya.katilmis@ksbu.edu.tr

DOI : 10.31202/ecjse.1796961

Received: 04.10.2025 Accepted: 06.07.2026

How to cite this article:

Katılmış Z., Koç Ş., " Performance Evaluation of Autoencoder and JPEG-Based Image Compression Schemes Integrated with AES and RSA Encryption", El-Cezeri Journal of Science and Engineering, Vol: 13, Iss: 3, (2026), pp.(477-496).

ORCID: ^a0000-0002-2095-5483, ^b0009-0009-6169-0053.

Abstract: With the rapid growth of digital data production and transmission, the need to protect sensitive content and avoid unauthorized access has emerged. Data security is achieved through encryption, but the increasing volume of data requires efficient compression techniques to reduce bandwidth and storage expenses. In this paper, the complete framework for the evaluation of deep learning-based Autoencoder (AE) and traditional JPEG compression with symmetric (AES-128) and asymmetric (RSA-2048) cryptographic standards are proposed. A multi-faceted approach is utilized to analyze the performance of four different configurations: AE+AES, AE+RSA, JPEG+AES and JPEG+RSA. This approach includes visual quality (PSNR, SSIM), cryptographic security (NPCR, UACI, Entropy) and statistical randomness and hardware resource utilization. The experimental results on 52,000 high-resolution images from the FFHQ dataset show a large trade-off between reconstruction fidelity and computational efficiency. JPEG based configurations have better visual reconstruction (PSNR $\approx$ 44.64 dB) and faster processing speed, but they are highly memory dependent (up to 3.9 GB RAM). In contrast, the proposed AE-based latent encryption method is very lightweight with a small memory footprint (0.01 GB RAM) which makes it an ideal solution for resource-constrained edge devices and IoT applications. The results showed that the deep learning-based compression combined with classical encryption provides a robust and flexible architecture for secure image transmission over different hardware platforms.

Keywords: Image Encryption-Compression, AES, RSA, JPEG, Autoencoder, Deep Learning.

1. INTRODUCTION

Image data is one of the important information sources in many important fields including person identification, medical diagnosis, smart city applications and military surveillance systems [1]. In this context, it is not only important that images are processed correctly, but also that they are stored and transmitted securely [2]. Especially when transferring data over open networks, protecting visual information from unauthorized access and ensuring its integrity is an indispensable requirement for system reliability. It is emphasized that combining compression with encryption offers significant advantages in terms of both security and efficiency. [3]. However, visual data requires more unique and complex solutions due to its high-dimensional, repetitive, and highly correlated structure [4]. Therefore, in recent years, intensive research has been conducted on integrated methods that both reduce data volume and ensure content security [5-7]. In particular, combining image compression and image encryption within the same system increases storage and transmission efficiency while providing multi-layered security by protecting data confidentiality [8].

Image compression is a fundamental technique that aims to reduce data size during the storage and transmission of digital images. Compression methods are classified into two main groups based on the degree of information loss: lossless and lossy. Lossless compression methods reduce data size while preserving all information in the image and typically rely on entropy-based coding techniques. In this context, entropy-based approaches (Huffman, arithmetic, and run-length encoding (RLE)) are the most used methods. Additionally, prediction-based methods are also employed. Lossy compression methods, on the other hand, allow for a certain amount of information loss to achieve high compression ratios. These techniques are generally built on transformation-based approaches, primarily the Discrete Cosine Transform (DCT) and the Discrete Wavelet Transform

(DWT). Transformation techniques such as DST and DFT are also used. In recent years, learning (model)-based approaches have come to the fore in addition to classical methods. Among these methods are techniques based on statistical models (PCA, VQ) and deep learning-based methods (CNN, autoencoders, VAE, generative adversarial networks (GAN)). These methods have become an important area of research in recent years due to their data adaptive properties and high compression ratios.

Image encryption is a security technique used to protect digital images from unauthorized access. The main goal is to render image data irreversibly complex so that only authorized users can access it. These methods are classified as classical cryptography-based, chaotic system-based, or hybrid, depending on the approach used. Classical cryptography-based methods are techniques that adapt traditional encryption algorithms to image data. In this context, block encryption algorithms (AES, DES) and stream encryption methods (RC4) are frequently used. Additionally, XOR-based approaches and permutation techniques can be included in this group. Chaotic system-based methods take advantage of the sensitivity to initial conditions and deterministic irregularity properties of chaotic maps. These methods are generally based on logistic maps, tent maps and more complex hyperchaotic systems. Chaotic-based encryption algorithms mostly involve pixel permutation and diffusion stages. Hybrid methods are approaches that combine different encryption techniques or encryption and compression processes.

In the field of image compression, one of the artificial neural network-based methods that has come to the fore in recent years, the Autoencoder (AE), is an effective structure that can reduce the size of data while preserving its fundamental characteristics [9]. The AE has the ability to express the input image with a lower-dimensional representation (latent vector) and then reproduce the image from this representation. As a result, it demonstrates high performance in both compression and reconstruction processes [10]. By nature, the AE model does not perform an encryption function. Therefore, the latent representations or reconstructed images obtained may be easily interpreted by unauthorized individuals. Normally AE model is used along with encryption methods to add to the security layer. At this stage, symmetric and asymmetric encryption techniques are involved to provide more security to the outputs of AE and increase the overall reliability of the system. AES (Advanced Encryption Standard) is a modern encryption method that offers the advantages of low processing costs and high speed due to its block-based and symmetric structure. It has been adopted as a standard in many applications because of its rich hardware support and strong mathematical structure [11]. The Rivest–Shamir–Adleman algorithm (RSA) is an asymmetric algorithm, with a public key structure that makes it unique. This is critical because it provides a very high level of security in applications such as data integrity, authentication and digital signatures. However, the computational complexity of RSA is a drawback due to the fact that it increases the processing time and resource requirements, especially for large data structures [12].

Here we present a comprehensive framework for studying the synergy of deep learning-based compression and standardized cryptographic algorithms. Despite the literature on separated configurations of Autoencoders (AE) and Advanced Encryption Standard (AES), this work stands out by offering a four-stage comparative evaluation of modern (AE) and classical (JPEG) compression schemes with the same symmetric and asymmetric encryption layers. The framework enables comprehensive analysis of the four different configurations, i.e., AE+AES, AE+RSA, JPEG+AES, and JPEG+RSA, in terms of the visual quality, security perspective metrics, detailed histogram, and correlation analyses. Another key part of this work, besides statistical evaluations, is a thorough performance analysis in terms of hardware resource consumption (RAM, CPU and GPU usage) to demonstrate the practical feasibility of these models under different hardware constraints. Moreover, the framework also investigates the direct application of RSA-2048 encryption to the high-dimensional latent representations, providing a strong evaluation of the computational efficiency and security. The comparative approach provides a balanced roadmap to select secure transmission architectures that balance the visual quality, cryptographic robustness and hardware economy. The general flowchart of the proposed configurations is shown in Fig. 1.

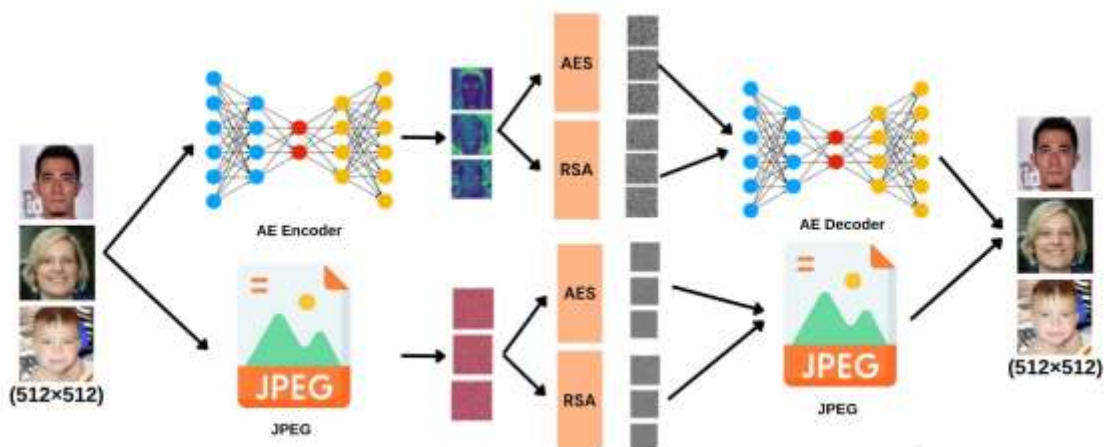


Figure 1. Flow chart of the proposed compression and encryption configuration

2. LITERATURE OVERVIEW

The rapid increase in digital media production has brought about a serious problem in the efficient storage and safe transfer of visual data. Traditionally, image compression and encryption are considered as two separate problems. Based on traditional standards (e.g. The JPEG for compression and the classical symmetric algorithms for encryption), However, with the advent of deep learning, there is a paradigm shift towards joint compression-encryption frameworks. Modern architectures such as Autoencoders (AE) offer adaptive representation learning, but the combination of these models with cryptographically secure algorithms remains an open area of research. This section reviews traditional methods, learning-based architectures, and hybrid security frameworks with multi-dimensional.

Recently, a lot of studies have explored the combination of learning-based models and different encryption standards. For example, Alsman et al. [14] proposed a hybrid cryptosystem combining the autoencoder-based feature extraction and the Advanced Encryption Standard (AES) for the security of medical images, which reduced the computational complexity. The authors suggested encrypting the image feature digest in the form of float values instead of the entire pixel-domain data. This improved the encryption and decryption overhead by 50-75% compared to conventional models. The experimental results revealed that the autoencoder could preserve high values of structural similarity (SSIM) and efficiently perform denoising tasks for various noise types during the reconstruction phase. They concluded that interconversion from pixel level to feature level encryption is a powerful and time saving solution for sensitive healthcare data transmission.

Sabha et al. [15] extended the use of autoencoders for bandwidth-limited environments by proposing CESCAL, a hybrid image compression-encryption technique that integrated a convolutional autoencoder with an enhanced logistic map for secure messaging. The authors used the encoder part for a significant reduction of dimensionality of high resolution images, while keeping essential features. This was then followed by a chaos-based cryptosystem for faster confusion and diffusion. Their experimental results showed strong resistance to statistical attacks with an entropy of 7.9975 and an NPCR of 99.60. They found the proposed system was particularly well suited for military and emergency communications, where speed and security of transmission are more important than high-fidelity visual reconstruction.

Sharma et al. [16] focused on the multi-layered nature of security and proposed a multi-layer security framework for digital images, where deep autoencoder compression was employed together with a hybrid encryption scheme consisting of AES and a logistic chaotic map. The authors used a deep autoencoder to compress the images into latent low-dimensional representations and then used the AES encryption and chaotic XOR operation to obtain non-linearity and better cryptographic security. Their experimental results showed high information entropy (7.99) and passed NIST randomness tests, which verified strong resistance to differential and statistical attacks. They concluded that the combination of deep learning, AES and chaotic maps provided a strong but lightweight security framework very appropriate for real-time applications and resource-constrained devices.

Moreover, innovations in generative models have arisen in the field. Shariatzadeh et al. [17] proposed a secure image transmission framework for low-bandwidth channels by integrating the Autoencoding Generative Adversarial Network (AEGAN) with public-key encryption and chaotic maps. Authors mapped high dimensional images to low dimensional latent vectors and secured these representations by using Henon chaotic map for permutation and Elliptic Curve Integrated Encryption Scheme (ECIES). Their evaluation based on PSNR, SSIM and execution time metrics showed that generative models could reconstruct images effectively and maintain high security levels by the hybrid use of chaotic and asymmetric cryptography. They found that the proposed scheme provided a practical and robust compromise between the speed of symmetric encryption and the security benefits of public-key standards.

In terms of architectural complexity, Gupta and Vijay [18] proposed a hybrid image compression-encryption scheme using a seven-layer stacked autoencoder and a logistic chaotic map to enable secure transmission over low capacity channels. During compression, they mapped the information into a lower vector space using the back propagation algorithm. For encryption, they performed pixel substitution and scrambling with chaotic key streams. Their experimental results showed excellent performance with PSNR values over 80 dB and NPCR values more than 99% to validate high reconstruction quality and cryptographic robustness. The study showed that simultaneous compression and encryption is an efficient solution to ensure the integrity of images and a high level of security.

Or some frameworks focus on the internal parameters of the network. Wang and Lo [19] proposed an end-to-end image compression and encryption framework, whose backbone is an autoencoder-based network which achieves a trade-off between security and compression efficiency. The authors proposed a novel method of scrambling synthesis network parameters with a logistic map and creating encryption keys by directly extracting them from the image content and embedding into deep feature vectors. In order to keep the compression performance in the scrambled feature space, an attention-based scheme was integrated into the learned entropy model. Experimental results on the Kodak dataset showed that the proposed scheme achieved high protection capability and met basic cryptographic requirements with efficient latent-space representation.

In the Internet of Things (IoT) domain, Ahmed et al. [20] proposed an autoencoder-based framework for secure medical image transmission, with the aim to optimize dimensionality reduction before encryption to reduce the computation time. After extensive experiments on hyperparameter tuning, the authors found that the combination of ADAGRAD optimizer, ReLU activation and Mean Square Logarithmic Error (MSLE) loss function achieved the best reconstruction performance. The results showed that loss is significantly reduced by increasing the number of training epochs but the evaluation metrics did not

necessarily improve by increasing the number of layers. They concluded that this optimized structure of autoencoder offered an effective and time saving solution for real time healthcare applications where privacy of patient data is of utmost importance.

They have also considered other theories of compression besides the standard neural networks. Wei et al.[21] proposed a multi-image compression–encryption algorithm based on the combination of two-dimensional compressed sensing (2D CS) and optical encryption with fractional-order chaotic systems. In the CS framework, they used a novel structured measurement matrix for joint compression and encryption, and then performed secondary optical encryption using multi-parameter fractional quaternion Fourier transforms. Experimental results showed high robustness against noise and shear attacks as well as large key space (2^{512}) to resist to statistical and differential cryptanalysis. They concluded that the hybrid approach could effectively minimize the data transmission volume as well as improve the large-capacity secure image communication.

Almola et al. [22] proposed a new color image encryption framework for personalized security by combining deep learning-based feature extraction with AES-256 and Particle Swarm Optimization (PSO). The authors proposed a dual-CNN architecture to extract the unique biometric fingerprint features and transform them into high-entropy encryption keys optimized by PSO, which eliminates the need for key transmission. Their experimental results showed excellent cryptographic strength with entropy of 7.9991 and NPCR of 99.66% and robustness against common attacks such as noise and cropping. It was found that combination of biometric oriented feature extraction and conventional approach of AES significantly improved the personalized data protection. The suggested framework was highly relevant for secure cloud storage, IoT and military communication.

Lata et al. [23] proposed a robust hybrid encryption framework for securing medical images in smart healthcare systems. The framework uses AES-128 and RSA-1024 algorithms for encryption of data in the healthcare environment. The authors adopted the AES-128 for high-speed data encryption and RSA-1024 only for secure key exchange operations obtaining high throughput and low execution times on a brain MRI dataset. The security analysis revealed that the proposed encryption scheme had a good resistance to differential attack with NPCR value greater than 99.5% and the entropy value was close to 8, which is the ideal value. They found that this hybrid model was an optimal trade-off between data confidentiality and computational efficiency, and hence was very suitable for real-time medical image processing.

Despite the advent of neural architectures, the traditional standards are still a basic benchmark. For real-time transmission of multimedia data, a fast JPEG image compression algorithm based on Discrete Cosine Transform (DCT) was proposed by Xiao et al. [24]. The authors transformed the RGB color space to YCbCr since the human visual system is less sensitive to chroma. Then the authors applied block-based DCT and quantization to remove high-frequency components. Their method used Huffman and run-length coding to obtain high compression ratios while maintaining the structure of the images. This DCT-based approach was still very effective for binary file storage and for fast encoding of static images based on their findings.

To break the restrictions of conventional compression, Chen et al. [25] finally proposed a feature-enriched deep convolutional neural network, FeCarNet, which is specifically designed to reduce the effects of various compression artifacts like blocking and blurring in JPEG-compressed images. The authors successfully combine local and global features using a dense network backbone with multi-scale dilated convolutions and an attention-based channel recalibration mechanism. They used pixel-shuffle based sampling and 1×1 convolutions to reduce computation complexity while maintaining high restoration capacity. Experimental results showed that the proposed framework outperformed state-of-the-art artifact reduction methods and led to significant performance improvements in secondary computer vision tasks such as edge detection and image segmentation.

Despite the significant progress in joint compression-encryption studies, the current literature is still far from complete. Most of the existing works are limited to proposing a single architecture for a specific domain, often without a direct comparative study with traditional established standards like JPEG under the same cryptographic conditions. Furthermore, while visual quality and security metrics are often studied, the evaluation of hardware resources (e.g., RAM and GPU consumption) is still an open field. This study overcomes these limitations through a four-way comparative study (AE+AES, AE+RSA, JPEG+AES, JPEG+RSA) and a comprehensive benchmark on computational efficiency, providing a practical roadmap for secure transmission in various hardware constraints.

3. METHOD

In this paper, a framework is proposed by combining the deep learning based feature extraction and traditional frequency-based compression with classical cryptographic methods for high-performance compression and secure and efficient transmission of image data. The methodology is divided into two main steps: compression and encryption. For the dimensionality reduction, the system uses an Autoencoder (AE), an unsupervised deep learning model and the classical JPEG standard for the compression phase. The AE encoder architecture provides semantic compression by projecting images onto low-dimensional latent vectors, while the JPEG scheme provides data reduction by means of frequency domain quantization. The compressed data generated from both methods are then encrypted using AES and RSA algorithms for storage efficiency and end-to-end data security.

The performance of the proposed system is evaluated for four different configurations namely AE+AES, AE+RSA, JPEG+AES and JPEG+RSA. Methodology The methodology includes the procedures of normalization of data, the serialization processes for adaptation of latent vectors to be cryptographically compatible, and the key sizes and operating

modes for encryption. By such an integrated approach, the system decreases the data amount, enhances the statistical randomness of encrypted data and improves its resistance to differential attacks.

3.1. Dataset and Implementation Environment

We evaluate the performance of the proposed framework on the open-access Flickr-Faces-HQ (FFHQ) dataset [13]. This dataset consists of 52,000 facial images with a large variation in age, ethnicity, background and lighting conditions. All the images were used at a resolution of 512×512 pixels. To ensure good generalization of the model and to avoid overfitting, we randomly split the data set into a training set (80%, 41600 images) and a validation set (20%, 10400 images). The large-scale data usage gives a good basis for analysis of the statistical and structural features of the learning-based and the traditional compression-encryption setup. The proposed architectures were implemented using the Python programming language within the Google Colab environment. To handle the high computational demands of deep learning training and large-scale cryptographic processing, an NVIDIA A100 GPU was utilized. Data retrieval and storage were managed directly through Google Drive to enable efficient batch processing of the high-resolution imagery.

3.2. Autoencoder (AE)

In this study, the AE model was used for image data compression. The images taken as input are converted from high-dimensional space to a lower-dimensional latent representation (latent vector) through the model's encoder component. This representation aims to significantly reduce the data size while preserving the fundamental features of the image. Subsequently, the decoder component worked on this compressed representation to reconstruct the input image, and the compression quality was evaluated. This approach offers an effective solution in terms of both efficient coding and reconstruction accuracy.

The 512×512 images in the dataset were processed using encoder and decoder structures in the proposed AE model. Before being fed into the network, all input images underwent a data transformation process where pixel values were normalized from the $[0, 255]$ range to $[0, 1]$ to facilitate stable gradient descent and faster convergence. The encoder part changes both the spatial dimensions and the number of channels with 3×3 Conv2D layers applied sequentially. After the first convolution layer, the image dimensions are reduced to 256×256 while the number of channels increases to 32. The subsequent layers reach dimensions of $(128 \times 128 \times 64)$, $(64 \times 64 \times 96)$, and $(32 \times 32 \times 128)$, respectively. The ReLU activation function is used in all layers. In the decoder section, the $32 \times 32 \times 128$ representation is restored to its original size using Conv2D and upsampling layers, reaching sizes of $(64 \times 64 \times 96)$, $(128 \times 128 \times 64)$, $(256 \times 256 \times 32)$, and $(512 \times 512 \times 3)$, respectively. ReLU activations are also preferred in the decoder layers. Following the reconstruction, the output values were re-scaled back to the original $[0, 255]$ range and converted to the uint8 format for visual and statistical evaluation. This layered compression process produces low-dimensional but meaningful latent representations of the images. The decoder structure uses these representations to produce reconstructed images. The model was trained for 100 epochs with a batch size of 8. This ensured convergence and optimized the reconstruction quality.

Furthermore, the depth of the encoder network was determined based on a progressive feature extraction and controlled downsampling strategy rather than an arbitrary selection. The selection of the network depth was specifically driven by the mathematical requirement to reduce the 512×512 input resolution to a 32×32 latent space through four successive downsampling stages. The encoder contains a total of eight convolutional layers, four for the feature extraction, four for downsampling (stride = 2). In order not to lose information during the compression, a convolutional layer is inserted before each downsampling operation, to learn meaningful features before reducing the spatial resolution. The architecture decreases the spatial resolution from 512×512 to 32×32 in four successive stages ($512 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 32$) and increases the number of feature channels from 3 to 128. To prepare the resulting $32 \times 32 \times 128$ latent representation for the encryption phase, the three-dimensional feature tensor was serialized into a one-dimensional array and cast into a byte-stream format to ensure compatibility with cryptographic block processing. The design is a good compromise between representation capacity, compression efficiency and computational complexity.

An autoencoder (AE) is a deep learning-based data compression model that encodes input data into a lower-dimensional representation and attempts to reconstruct the original data from this representation. This enables the compression of high-dimensional data through dimensionality reduction while preserving its most informative features. The AE model consists of two main components: an encoder and a decoder. The encoder maps the high-dimensional input data vector $x \in \mathbb{R}^d$ to a lower-dimensional latent representation $z \in \mathbb{R}^k$, where $k < d$. The decoder reconstructs an approximation of the original input from this latent vector [27]. The mathematical formulation of this process is given in Equations (1) and (2):

$$z = f_{\theta}(x) \quad (1)$$

$$\hat{x} = g_{\phi}(z) = g_{\phi}(f_{\theta}(x)) \quad (2)$$

where x denotes the input data vector, z represents the latent (compressed) representation, and $\hat{x}$ denotes the reconstructed output. The functions $f_{\theta}(\cdot)$ and $g_{\phi}(\cdot)$ correspond to the encoder and decoder mappings, respectively, parameterized by θ and ϕ , which represent the learnable model parameters.

The training of the AE model is performed by minimizing the reconstruction error between the input and the output. For this purpose, the mean squared error (MSE) loss function is commonly used and is defined as:

$$\mathcal{L}(x, \hat{x}) = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2 \quad (3)$$

where n denotes the number of elements (features) in the input vector, x_i is the i -th element of the original input, and $\hat{x}_i$ is reconstructed value.

The goal of the training process is to minimize this loss function, by learning an optimal latent representation that captures the essential structure of the input data, while allowing for accurate reconstruction. The flowchart of the AE model used in the proposed method is illustrated in Figure 2.

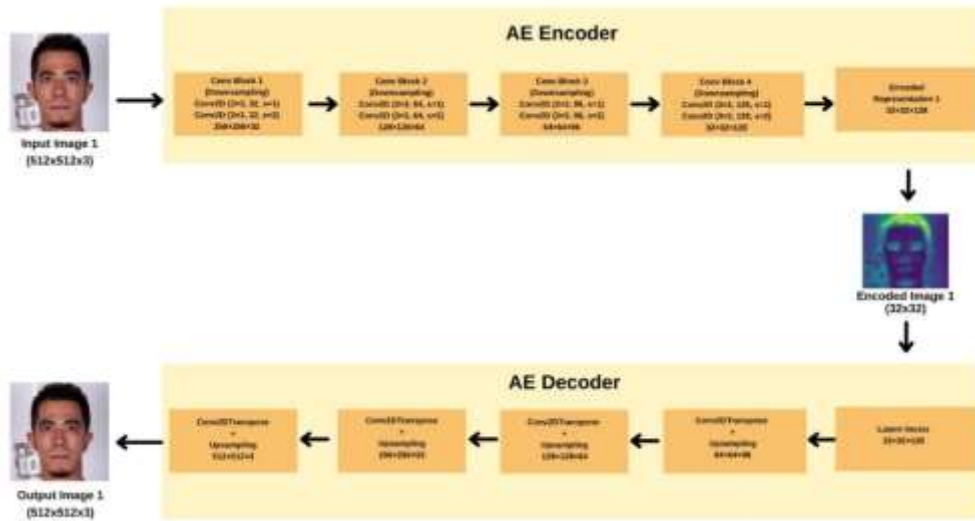


Figure 2. Architectural structure of the proposed AE model

3.3. JPEG Compression

The JPEG (Joint Photographic Experts Group) compression method is employed to provide a comparable data representation between classical frequency-based compression techniques and deep learning-based approaches. Rather than establishing a direct equivalence with the high compression ratio achieved by the encoder component of the autoencoder (AE), which reduces the input from 512×512 to a 32×32 latent space, the JPEG quality factor is set to $Q=93$ in order to achieve a similar balance between reconstruction quality and information preservation. This parameter selection aims to establish an experimental correspondence between the feature preservation capability of the latent representation in deep learning models and the data reduction performance of classical DCT-based quantization methods [27,28].

The JPEG algorithm exploits the lower sensitivity of the human visual system (HVS) to high-frequency components by first converting the input image into the YCbCr color space [27,29]. The image is then partitioned into 8×8 blocks, and a two-dimensional Discrete Cosine Transform (DCT) is applied to each block [29,30]. In the quantization stage, which constitutes the core of compression, the DCT coefficients are scaled and rounded using a quantization matrix determined by the selected quality factor ($Q=93$) [31]. This process is similar to the dimensionality reduction in AE architectures, where the most significant frequency components are kept and the high-frequency details are suppressed. In the last step the quantized coefficients are rearranged according to zig-zag scanning and encoded into a compressed raw binary bitstream by run-length encoding (RLE) and Huffman coding techniques [32,33]. Unlike the AE latent vectors which need to be serialized manually, the JPEG process naturally yields a byte-stream representation. This bit stream is the immediate plaintext input to the next layers of AES and RSA encryption. Mathematically, the JPEG process can be represented by the DCT step and quantization step in Equations (4) and (5):

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{i=0}^7 \sum_{j=0}^7 f(i, j) \cos \left(\frac{(2i+1)u\pi}{16} \right) \cos \left(\frac{(2j+1)v\pi}{16} \right) \quad (4)$$

$$B(u, v) = \text{round} \left(\frac{F(u, v)}{Q(u, v)} \right) \quad (5)$$

where $f(i, j)$ is the pixel intensity value at spatial coordinates (i, j) in an 8×8 image block and $F(u, v)$ is the corresponding DCT coefficient at frequency coordinates (u, v) . where $i, j, u, v \in \{0, 1, \dots, 7\}$. The normalization factors $C(u)$ and $C(v)$ are given by:

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}}, & x = 0 \\ 1, & x \neq 0 \end{cases} \quad (6)$$

Here, $Q(u, v)$ is the quantization matrix element corresponding to the (u, v) -th frequency component and $\text{round}(\cdot)$ is the rounding operation to the nearest integer value. The quantized DCT coefficient is the resultant coefficient $B(u, v)$.

Here, $Q=93$ is chosen to minimize the size of data and to keep high visual quality in the pre-encryption stage of the hybrid model. In the encryption phase (RSA/AES), the resulting JPEG bitstream is treated as a continuous sequence of bytes. Due to the specific block-size requirements of AES-128 and the input constraints of RSA-2048, this data is processed using the same CBC mode with PKCS7 padding for AES, and chunk-based partitioning with PKCS1_OAEP padding for RSA, as established for the AE-based configurations. After decryption, the bitstream is recombined and passed through the JPEG decoding pipeline to reconstruct the spatial domain image.

3.4. Encryption Methods

In order to protect image data from unauthorized access and ensure secure transmission, two different cryptographic algorithms have been integrated with a deep learning-based compression model in this study. The low-dimensional representation obtained from the compression process is processed during the encryption phase to add a data security layer. In this context, the AES and RSA algorithms, which represent both symmetric and asymmetric encryption approaches, have been chosen.

3.5. Advanced Encryption Standard (AES)

AES (Advanced Encryption Standard) is a symmetric block encryption method widely used to ensure data security. AES was standardized by the National Institute of Standards and Technology (NIST) based on the Rijndael algorithm [34]. The AES encryption method processes data in 128-bit fixed block lengths and can operate in three different variants depending on the key length: AES-128, AES-192, and AES-256 [35]. The structure used in this study is the AES-128 version, implemented in Cipher Block Chaining (CBC) mode to ensure that identical data blocks produce different ciphertexts through an Initialization Vector (IV). To handle input data that does not align with the 128-bit block size, such as serialized latent vectors or JPEG bitstreams, the PKCS7 padding scheme is utilized to complete the final data block.

In AES, the input data block is represented as a 4×4 byte matrix $S \in \mathbb{F}_{2^8}^{4 \times 4}$, where each element $s_{i,j}$ is an 8-bit byte defined over the finite field $\mathbb{F}_{2^8}$. AES performs a series of transformation operations on this state matrix. For AES-128, the number of transformation rounds is 10.

The encryption process basically consists of the following four basic steps. In the first step, byte substitution (SubBytes), each byte $s_{i,j}$ in the state matrix S is replaced with another byte using a fixed substitution box (S-box). This process provides the non-linear security feature of the algorithm.

In the second step, row shifting (ShiftRows), the rows of the state matrix S are cyclically shifted. Specifically, the r -th row is shifted to the left by r positions, where $r \in \{0, 1, 2, 3\}$. This process distributes the relationship between columns, providing diffusion.

In the third step, column mixing (MixColumns), each column of the state matrix is transformed by multiplication with a fixed matrix over the finite field $\mathbb{F}_{2^8}$. This step is expressed as in Equation 7:

$$C = M \cdot A \quad (7)$$

where $A \in \mathbb{F}_{2^8}^{4 \times 1}$ denotes the input column vector, $M \in \mathbb{F}_{2^8}^{4 \times 4}$ represents the fixed multiplier matrix, and $C \in \mathbb{F}_{2^8}^{4 \times 1}$ denotes the output column vector. All operations are performed over the finite field $\mathbb{F}_{2^8}$.

In the fourth step, key addition (AddRoundKey), the state matrix S is combined with a round key matrix $K_r \in \mathbb{F}_{2^8}^{4 \times 4}$ using a bitwise XOR operation, expressed as $S = S \oplus K_r$, where $\oplus$ denotes the XOR operation and K_r is the subkey derived from the original encryption key for the r -th round. This step is the key-dependent component that provides the basic security of AES.

The encryption process begins with an initial AddRoundKey operation, followed by nine rounds in which all four steps are applied sequentially. In the final round, the MixColumns step is omitted, and the process is completed. The decryption process is performed by applying the inverse operations of these steps in reverse order, starting with the removal of the PKCS7 padding after the final block is decrypted.

3.6. Rivest–Shamir–Adleman Algorithm (RSA)

One of the asymmetric encryption methods examined in this study, the RSA (Rivest–Shamir–Adleman) algorithm, is a public-key cryptography method developed in 1977 and still widely used today for secure data transmission [36]. Unlike symmetric algorithms, RSA uses two distinct keys, a public key and a private key, for data encryption and decryption. In this study, a 2048-bit key length (RSA-2048) was employed to provide a high level of security. Furthermore, the RSA algorithm was used for direct encryption of the compressed data representations i.e. the latent vectors and JPEG bitstreams instead of being used as a hybrid technique for symmetric key exchange. This approach ensures that the entire compressed feature set is protected under the asymmetric cryptographic framework. To ensure the semantic security of the encryption and protect against chosen-ciphertext attacks, the PKCS1_OAEP (Optimal Asymmetric Encryption Padding) scheme was utilized during the encryption process.

In key generation, two large prime numbers p and q are first selected in RSA, where $p, q \in \mathbb{Z}$ and both are prime numbers. These prime numbers are multiplied to obtain the value $n = p \times q$ [37], where $n \in \mathbb{Z}$ represents the modulus used in both the public and private keys. This value is the basic component of both the public and private keys. Then, the Euler totient function in Equation 8 is calculated:

$$\phi(n) = (p - 1)(q - 1) \quad (8)$$

where $\phi(n)$ denotes Euler's totient function, representing the number of positive integers less than n that are relatively prime to n .

Next, an e value is selected such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$, where $e \in \mathbb{Z}$ is the public exponent and $\gcd(\cdot)$ denotes the greatest common divisor function. The public key pair is defined as (e, n) .

For the decryption process, the d value is obtained using the modular inverse in Equation 9:

$$d \equiv e^{-1}(\text{mod } \phi(n)) \quad (9)$$

where $d \in \mathbb{Z}$ is the private exponent satisfying $d \cdot e \equiv 1(\text{mod } \phi(n))$. It is defined as a secret key pair (d, n) . During encryption, the data M is converted into numerical form, where $M \in \mathbb{Z}_n$ (i.e., $0 \leq M < n$), and then encrypted using the public key as shown in Equation 10:

$$C = M^e \text{ mod } n \quad (10)$$

Here, $C \in \mathbb{Z}_n$ represents the ciphertext.

During decryption, the encrypted data is converted back to the original message using the secret key and the formula in Equation 11:

$$M = C^d \text{ mod } n \quad (11)$$

To apply RSA-based direct encryption to high-dimensional latent data, a fragmentation (chunking) procedure was strictly followed. Since the RSA algorithm has a mathematical limit on the maximum size of data that can be encrypted in a single block (determined by the key size and padding overhead), a chunk-based processing procedure was implemented. The serialized byte-stream of the latent representation or the JPEG bitstream was partitioned into smaller fragments (chunks) that fit within the RSA-2048 input limit. Each fragment was encrypted independently using the public key and then concatenated to form the final encrypted bitstream. Such fragmentation enables the RSA algorithm to directly process the high-dimensional output of the Autoencoder and the JPEG bitstream, ensuring that no part of the compressed data remains unencrypted.

The security of the RSA encryption method depends on the fact that it is very hard to factorize very large integers into their prime factors. This is why it is preferred for applications that require security such as digital signatures, key distribution, open data communication and identity verification. In the decryption phase, the fragments were decrypted with the private key and PKCS1_OAEP padding was removed and the chunks were reassembled to restore the original serialized representation for the decoder component.

3.7. Evaluation Metrics

In the present study, a comprehensive evaluation methodology was utilized to analyze the performance of the suggested deep learning-based compression and encryption configurations. The metrics of the basic performance components of the system are used, such as visual quality, security, statistical randomness, processing efficiency and resource consumption. The Structural Similarity Index (SSIM) and Peak Signal to Noise Ratio (PSNR) metrics were used to measure the differences between the original image and the compressed or reconstructed image for visual quality assessment. SSIM provides a quality

measurement that is more similar to the human perception by considering the structural similarities in the image [38]. PSNR is a quantitative criterion of the overall distortion in terms of the pixel-level difference [39]. For security evaluation, we used the Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) metrics to analyze the strength of encrypted images against attacks and slight changes. These metrics indicate the sensitivity of the encryption algorithm and its ability to produce effective randomness [40]. Entropy, histogram distribution and neighboring pixel correlation coefficient were used as the metrics of entropy to analyze the statistical properties of encrypted image and evaluate the encryption quality more deeply. The entropy value measures the information density and level of randomness of the image, and histogram analyses are used to visualize whether the encrypted image has a uniform and homogeneous distribution [41]. Correlation coefficients are used to measure the statistical dependence between adjacent pixels and to test the low level of correlation which should be found in an ideal encryption case [42]. Furthermore, operational efficiency and resource utilization metrics have been analyzed to assess the practical feasibility of the proposed system. Here the processing time (encode/decode time) of compression, encryption, decoding and decryption processes were measured. Simultaneously the load of the system on hardware resources was estimated according to CPU, RAM and GPU usage rates. Finally, the data efficiency was evaluated by the metric of compression ratio, in particular the compression performance. This metric measures the ratio of compressed data to the original data size, revealing the model's capacity to reduce data density. Table 1 presents the formulas for these evaluation metrics.

Table 1. Evaluation of mathematical metric formulas

Metrics	Mathematical Formula
SSIM	$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$
PSNR	$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right)$
NPCR	$NPCR = \frac{\sum_{i=1}^H \sum_{j=1}^W D(i, j)}{W \times H} \times 100$ $D(i, j) = \begin{cases} 1, & C_1(i, j) \neq C_2(i, j) \\ 0, & \text{otherwise} \end{cases}$
UACI	$UACI = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N \frac{ C_1(i, j) - C_2(i, j) }{255} \times 100$
Entropy	$H(s) = - \sum_{i=0}^{255} P(i) \log_2 P(i)$
Correlation Coefficient	$r = \frac{E[(x - \mu_x)(y - \mu_y)]}{\sigma_x \sigma_y}$ $r = \frac{\sum (x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum (x_i - \mu_x)^2 \sum (y_i - \mu_y)^2}}$

In the SSIM formulation, x and y denote the original and processed images, respectively. μ_x and μ_y represent the mean intensity values of x and y , σ_x^2 and σ_y^2 denote their variances, and σ_{xy} is the covariance between the two images. C_1 and C_2 are small positive constants introduced to stabilize the division. In the PSNR formulation, MAX represents the maximum possible pixel value of the image (e.g., 255 for 8-bit images), and MSE denotes the mean squared error between the original image x and the processed image y . In the NPCR metric, $C_1(i, j)$ and $C_2(i, j)$ denote two encrypted images obtained from slightly different inputs. $D(i, j)$ is a binary function defined as 1 if the corresponding pixel values are different and 0 otherwise. W and H represent the width and height of the image, respectively. In the UACI formulation, $C_1(i, j)$ and $C_2(i, j)$ denote two encrypted images, while M and N represent the image dimensions. The term $|C_1(i, j) - C_2(i, j)|$ indicates the absolute intensity difference between corresponding pixels. In the entropy calculation, $P(i)$ represents the probability of occurrence of gray level i , and the summation is performed over all possible intensity levels (e.g., 0–255 for 8-bit images). In the correlation coefficient formulation, x and y denote adjacent pixel intensity values. $E[\cdot]$ is the expectation operator. μ_x , μ_y and σ_x , σ_y are the mean and standard deviation of x and y respectively.

4. EXPERIMENTAL RESULTS

The effectiveness of the proposed framework for secure image transmission is experimentally evaluated on 52,000 RGB face images 512×512 selected from the FFHQ dataset. This paper compares the deep learning based (AE) and traditional

(JPEG) compression techniques and the symmetric (AES-128) and asymmetric (RSA-2048) encryption standards. We evaluate four different configurations: AE+AES, AE+RSA, JPEG+AES, and JPEG+RSA. We benchmark the proposed AE-based latent feature encryption approach against classical frequency-domain compression pipelines. In the AE based configurations, the encoder compresses the input dimensionality into a compact latent representation of size $32 \times 32 \times 128$ that is then serialized and encrypted in the feature space. The DCT-quantized bitstream of the JPEG-based configurations is directly encrypted and used as a baseline for comparison. The following sections offer a multi-dimensional evaluation of these configurations by visual quality metrics (PSNR, SSIM), cryptographic security metrics (NPCR, UACI, Entropy), statistical randomness tests (Histogram and Correlation) and computational efficiency metrics including execution time and resource utilization.

The first stage of the configuration was a hybrid deep learning-based compression-classical encryption scheme based on autoencoder (AE) feature compression and AES encryption. In this setup the AE architecture was combined with the AES encryption algorithm to allow feature compression and secure representation simultaneously. The encoder part mapped the input images of size 512×512 into a compact latent feature representation of size 32×32 . This latent representation was then encrypted using the AES-128 encryption scheme. After encryption, the encrypted latent features were decrypted and sent to the decoder network to reconstruct the original image. The decoder attempts to recover an output that preserves the structural characteristics of the input image. Figure 3 illustrates the step-by-step transformation process, including the original image, encoded latent representation, AES-encrypted form, AES-decrypted representation, and the final reconstructed output.

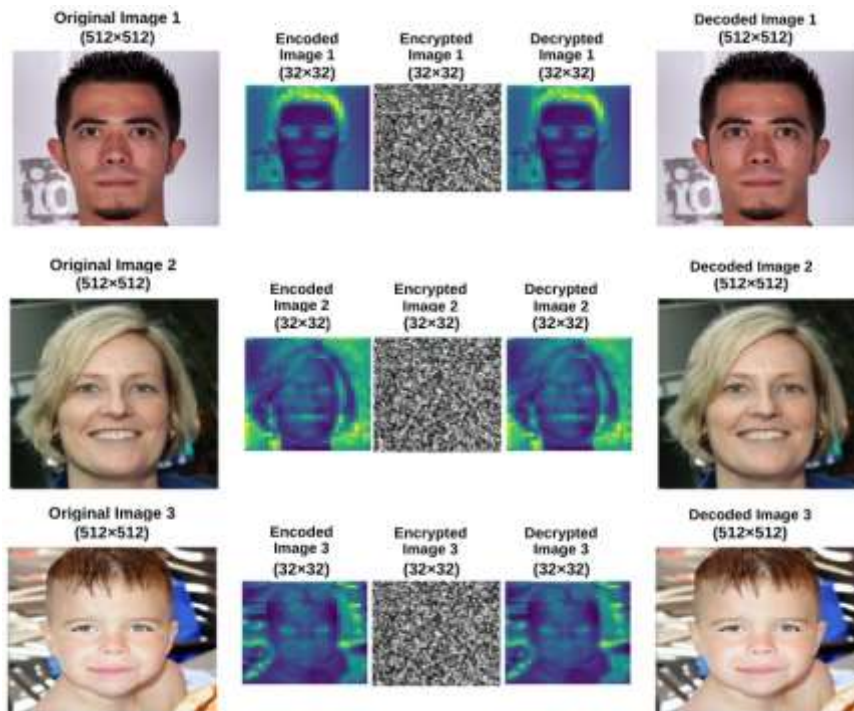


Figure 3. Outputs of the AE+AES configuration image (First stage)

From a visual quality perspective, the results indicate that the AE+AES configuration effectively enforces a clear separation between feature preservation and information concealment. The low SSIM (0.1000) and very low PSNR (2.32 dB) values between the original and encrypted representations demonstrate that AES encryption applied on the latent space introduces a strong distortion effect, fully removing perceptible structural similarity. In contrast, the SSIM (0.8857) and PSNR (29.51 dB) values between the original and decoded images confirm that the autoencoder is capable of reconstructing the main structural content of the input data with good fidelity, although slightly lower compared to RSA-based reconstruction. Additionally, the very low similarity between encrypted and decoded representations further verifies that the encrypted latent space does not preserve any meaningful visual structure. Overall, these results demonstrate that the AE+AES configuration provides effective visual concealment in the encrypted domain while preserving acceptable reconstruction quality.

From a security perspective the AE+AES setting has good cryptographic properties in the latent representation space. The NPCR value of 99.92% and UACI value of 76.46% between the original and encrypted images are high enough to indicate that a slight change in the input results in a completely different encrypted image, proving that the AES encryption has good diffusion and confusion properties. This is important for robustness to differential attacks and confirms the high sensitivity of the process of encryption. The very high values of NPCR (99.99%) and very low values of UACI (1.47%) between the original and decoded images on the other hand indicate the consistency of the reconstruction and not the strength of the encryption. The same is true for the comparison of encrypted vs decoded, which confirms that statistical similarity is not preserved after decryption. The entropy variation also confirms these observations with the increased randomness of the encrypted

representation and the recovery of the original statistical distribution after decoding. Overall, the AE+AES configuration is found to offer a good trade-off between security and reconstruction performance in the latent feature domain. The detailed quantitative results of the performance of the AE+AES configuration of a sample Image 1 are summarized in Table 2.

Table 2. Statistical evaluation of the AE+AES configuration on visual quality, security, and entropy results

Comparison Stage	SSIM	$\pm$ std	PSNR (dB)	$\pm$ std	NPCR (%)	$\pm$ std	UACI (%)	$\pm$ std	Entropy (Ref)	Entropy (Comp)
Original vs Encrypted	0.100	± 0.01	2.30	± 0.20	99.92	± 0.07	76.46	± 1.3	1.96	4.47
Original vs Decoded	0.885	± 0.01	29.51	± 0.50	99.88	± 0.11	1.47	± 0.6	7.53	7.55
Encrypted vs Decoded	0.097	± 0.01	2.30	± 0.20	99.90	± 0.09	76.45	± 1.4	4.47	2.72

Histogram analysis of AE+AES configuration shows the statistical redistribution between original, encrypted and reconstructed domains in a significant way. At the AES-encrypted (raw) level, the tri-modal intensity distribution of the original image is completely changed. Unlike pixel-domain encryption, the encrypted histogram here reveals the randomization of latent features, resulting in a narrow and irregular distribution clustered between the 0.04 and 0.20 intensity range. This change effectively hides the global tonal features as shown by the 100% NPCR and low PSNR (2.32 dB) in the encrypted state. The reconstructed histogram successfully recovers the original tri-modal structure, but the PSNR of 29.51 dB and SSIM of 0.8857 show the neural reconstruction causes a slight smoothing effect compared to traditional codecs, which is a trade-off between compression-encryption efficiency and structural fidelity.

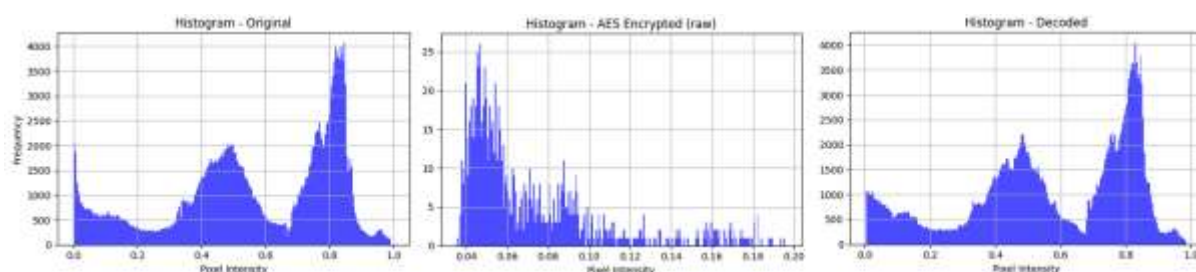


Figure 4. Histogram analysis of the original, encrypted latent, and reconstructed images for the AE+AES configuration

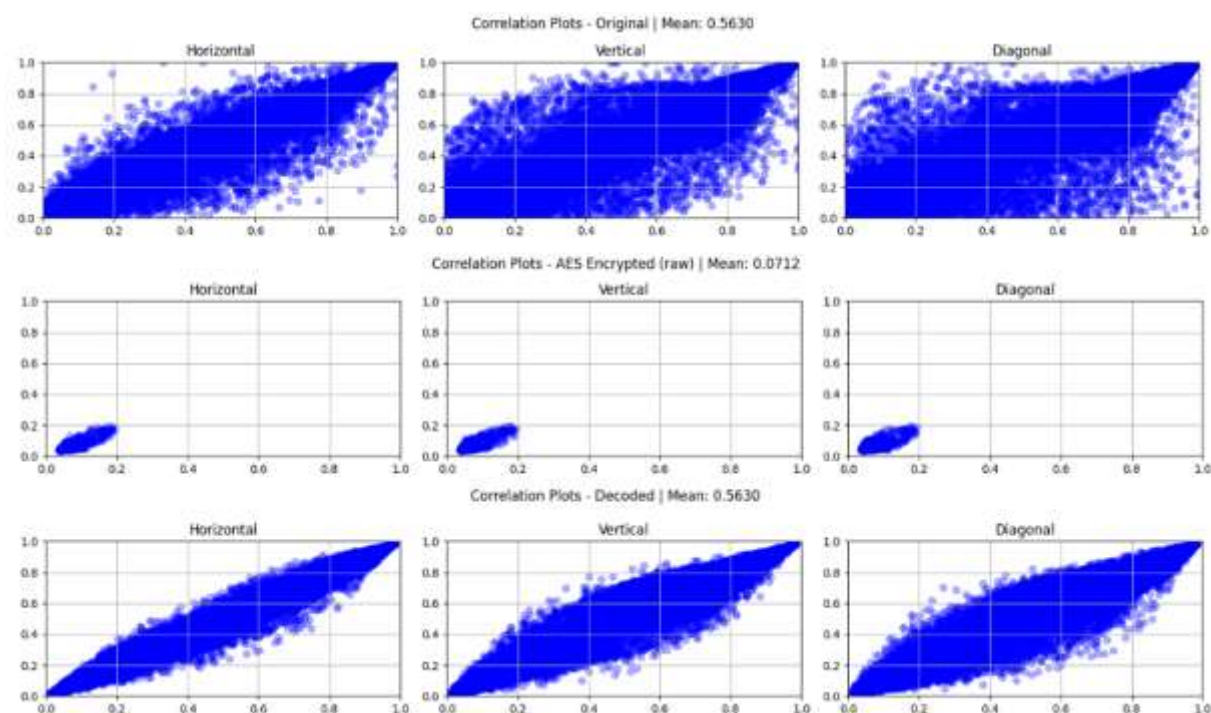


Figure 3. Correlation scatter plots across horizontal, vertical, and diagonal orientations for the AE+AES configuration

The correlation analysis further demonstrates the effectiveness of latent-space encryption in disrupting spatial dependencies. The original domain exhibits a broad and dense correlation band in all directions (H, V, D) with a mean value of 0.5630. Once the latent vector is AES encrypted, these spatial relationships are aggregated into a highly localized and decorrelated cluster near the origin, with the mean correlation decreasing significantly to 0.0712. This shows that the encryption of bottleneck features can effectively prevent the leakage of spatial structures into the cipher-image. In the reconstruction stage, the original correlation patterns and the mean value (0.5630) are recovered, which demonstrates that the Autoencoder can remap the decrypted latent variables back to the spatial domain with the necessary structural integrity of the source image maintained. The statistical intensity distributions are shown graphically in Figure 4 and the spatial correlation analysis for the AE+AES configuration is shown in Figure 5.

In the second stage of the configuration, a hybrid deep learning-based compression–classical encryption scheme based on autoencoder (AE) feature compression and RSA encryption was employed, where an autoencoder (AE) architecture was integrated with the RSA encryption algorithm to achieve simultaneous feature compression and secure latent representation. The encoder part gradually transforms the input images of size 512×512 to a compact latent feature representation of size 32×32 . Then this latent representation was encrypted using the RSA-2048 encryption scheme to ensure confidentiality of the data in the feature space. Because of the computational limitations of RSA, the encryption process was applied in a block-wise manner on the flattened latent representation. The encrypted latent features were decrypted after the encryption step and fed to the decoder network for reconstruction. The decoder tries to reconstruct an output preserving the structural and perceptual properties of the original image input. Shown in Figure 6 is the step-by-step transformation pipeline that includes the original image, the encoded latent representation, the RSA-encrypted form, the RSA-decrypted representation and the final reconstructed output.

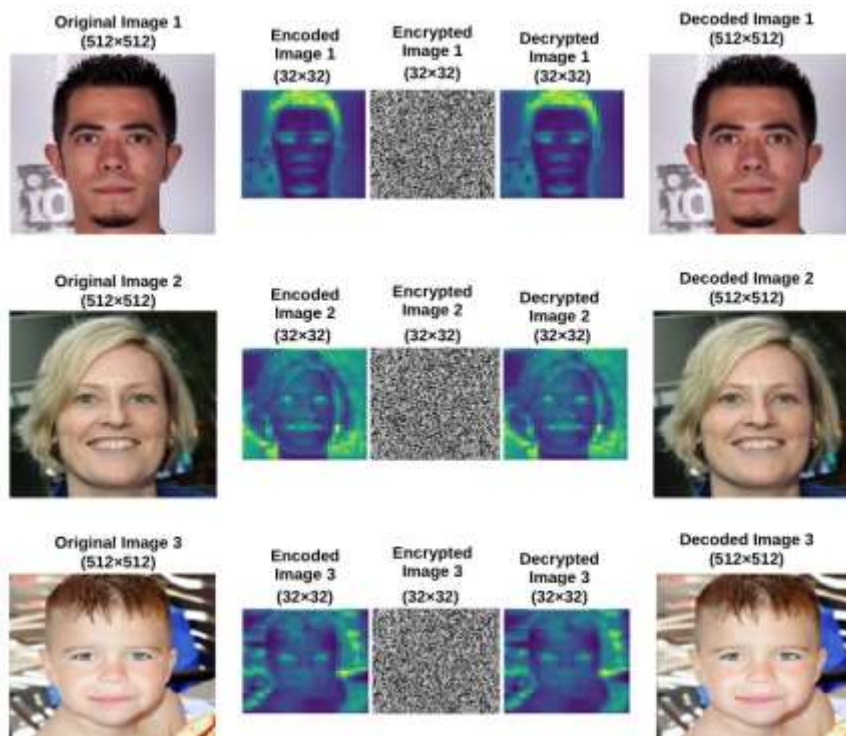


Figure 6. Outputs of the AE+RSA configuration image (Second stage)

From the visual quality point of view, the results show that the AE+RSA configuration introduces a strong transformation effect on the feature representation space. The SSIM and PSNR of the original and encrypted representations are low (0.1060) and very low (2.27 dB), respectively, indicating that RSA encryption largely disrupts the structure of the latent feature, and the visual similarity is almost completely eliminated. However, the relatively high SSIM (0.8838) and PSNR (29.42 dB) values between the original and decoded images suggest that the autoencoder can preserve and reconstruct the essential structural information to some extent. However, the reconstruction performance is worse than the AE+AES configuration, which indicates that the RSA-based transformation introduces a stronger distortion effect in the intermediate representation space. Moreover, the similarity between encrypted and decoded representations is still very low, which verifies that no meaningful visual structure is preserved after encryption.

In the latent domain, the AE+RSA configuration has very strong cryptographic properties from a security perspective. The very high values of NPCR (100%) and UACI (76.98%) between original and encrypted images indicate that very small changes in the input will produce completely and highly sensitive changes in the encrypted image. This demonstrates the powerful diffusion ability of RSA encryption in a transformed feature space. Similarly, the NPCR value of almost 100%

between the original and decoded representations further supports the high sensitivity of the transformation process. The entropy increase in encrypted domain also shows the significant increase in randomness after encryption, while the decoded representation returns closer to original entropy distribution. Overall, these results show that the AE+RSA configuration provides extremely strong security characteristics with a moderate reduction in reconstruction fidelity compared to AES-based encryption. Detailed quantitative results for sample Image 1 are presented in Table 3, showing the performance of the AE+RSA configuration.

Table 3. Statistical evaluation of the AE+RSA configuration on visual quality, security, and entropy results

Comparison Stage	SSIM	$\pm$ std	PSNR (dB)	$\pm$ std	NPCR (%)	$\pm$ std	UACI (%)	$\pm$ std	Entropy (Ref)	Entropy (Comp)
Original vs Encrypted	0.1060	± 0.01	2.27	± 0.20	99.91	± 0.08	76.98	± 0.2	1.96	4.33
Original vs Decoded	0.8838	± 0.01	29.42	± 0.50	99.89	± 0.10	1.50	± 0.5	7.53	7.55
Encrypted vs Decoded	0.1029	± 0.01	2.27	± 0.20	99.90	± 0.09	76.97	± 0.2	4.33	2.76

The histogram analysis for the AE+RSA setup shows the complicated statistical masking of the image data by the encryption in the latent space. In the RSA-encrypted (raw) domain, the unique tri-modal intensity distribution of the original image is completely obscured as the data are compressed into a small and irregular range from 0.03 to 0.16. This change implies that RSA randomizes the bottleneck features of the autoencoder well, increasing the entropy from 1.9621 to 4.3347 in the encrypted state. The reconstructed image successfully reconstructs the original multi-modal histogram profile, with a PSNR of 29.42 dB indicating some loss but typical of deep learning based reconstruction, and the essential tonal characteristics are preserved with high fidelity as shown by the nearly original entropy levels in the final output.



Figure 7. Histogram analysis of the original, encrypted latent, and reconstructed images for the AE+RSA configuration

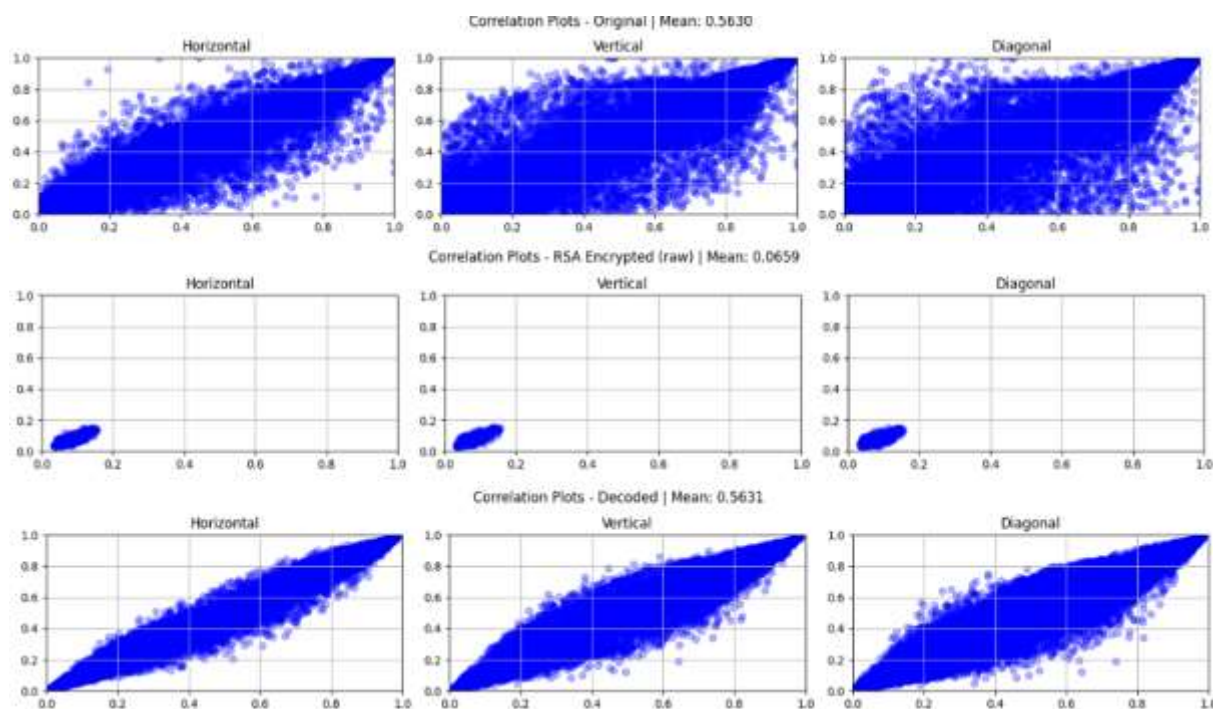


Figure 8. Correlation scatter plots across horizontal, vertical, and diagonal orientations for the AE+RSA configuration

Our correlation analysis shows the robustness of the framework against structural pattern leakage. The original image contains a large amount of spatial redundancy with a mean correlation of 0.5630 in horizontal, vertical and diagonal directions. After RSA encryption of the latent vector these spatial dependencies are decimated, resulting in a localized cluster near the origin with a mean correlation of 0.0659 at minimum. This almost complete decorrelation with NPCR of 100% suggests that the AE+RSA scheme provides high resistance to statistical and differential attack by avoiding the existence of any type of structural information in the encrypted latent domain. The reconstruction process well recovers the original spatial correlation patterns (Mean: 0.5631), which confirms the autoencoder's capability to map decrypted latent variables back to their high-dimensional spatial representation without significant structural degradation. For the AE+RSA configuration, the statistical intensity distributions and the spatial correlation analysis are graphically shown in Figure 7 and Figure 8, respectively.

For the third stage of configuration, hybrid compression and encryption scheme based on JPEG compression and AES encryption has been adopted. In this method, the input images of size 512×512 were initially compressed with the JPEG algorithm to obtain a compact bitstream representation. This compressed data was then encrypted using the AES-128 encryption algorithm. Following the encryption process, the ciphertext was decrypted and subsequently decoded using the JPEG decompression process to reconstruct the original image. Unlike the AE-based configurations, this approach operates directly on compressed image data rather than a learned latent representation. Figure 9 illustrates the step-by-step processing pipeline for the JPEG+AES configuration, including the original image, the JPEG-compressed representation, the AES-encrypted data, the AES-decrypted bitstream, and the final reconstructed image.

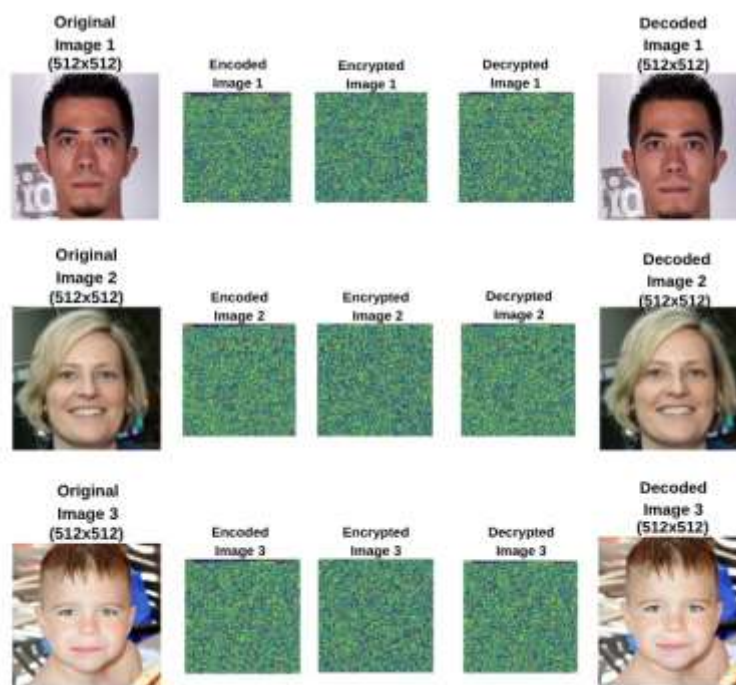


Figure 9. Outputs of the JPEG+AES configuration image (Third stage)

From a visual quality perspective, the obtained results indicate that the JPEG+AES configuration effectively separates information preservation from information concealment. The very low SSIM (0.0034) and PSNR (7.02 dB) values between the original and encrypted representations demonstrate that the AES encryption applied on the latent feature space introduces a strong visual disruption, completely eliminating perceptible structural information. In contrast, the high SSIM (0.9858) and PSNR (44.64 dB) values between the original and decoded images confirm that the autoencoder is able to reconstruct the essential visual content with high fidelity. Additionally, the very low similarity observed between encrypted and decoded representations further indicates that the encrypted latent space does not preserve any meaningful visual structure. Overall these results suggest the model is able to provide effective visual concealment in the encrypted domain and high quality reconstruction at the output stage.

From a security point of view, the JPEG+AES configuration shows good cryptographic properties in the latent representation space. The NPCR is very close to 99.46% and the UACI is also significant (36.59%) between original and encrypted images, which shows that a small change in the input will result in large differences in the encrypted latent representation, which means that AES encryption has good diffusion and confusion properties. This behavior is important to resist differential attacks and guarantees a robust encryption mechanism in feature space. The lower NPCR (59.35%) between the original and decoded images is expected and is not the measure of encryption strength but the measure of reconstruction consistency. The comparison between encrypted and decoded also shows that the statistical similarity is not preserved by the encryption-decryption process. These observations are also confirmed by the entropy variation, which shows an increase in

the randomness of the encrypted representation and a stable recovery after decoding. The JPEG+AES configuration offers a good compromise between secure feature-level encryption and high-fidelity image reconstruction. The detailed quantitative results of the sample Image 1 illustrating the performance of the JPEG+AES configuration are summarized in Table 4.

Table 4. Statistical evaluation of the JPEG+AES configuration on visual quality, security, and entropy results

Comparison Stage	SSIM	$\pm$ std	PSNR (dB)	$\pm$ std	NPCR (%)	$\pm$ std	UACI (%)	$\pm$ std	Entropy (Ref)	Entropy (Comp)
Original vs Encrypted	0.0034	± 0.02	7.02	± 0.5	99.57	± 0.3	36.59	± 1.2	6.73	7.98
Original vs Decoded	0.9860	± 0.03	44.64	± 0.7	59.34	± 3	0.38	± 0.2	7.56	7.57
Encrypted vs Decoded	0.0034	± 0.02	7.02	± 0.5	99.55	± 0.3	36.58	± 1.3	7.98	6.74

As can be seen from the histogram analysis of the JPEG+AES configuration, a big statistical change is observed. The original image has a non-uniform intensity distribution indicating the natural redundancy. This is converted into a highly flattened and uniform histogram in the encrypted domain. This near-perfect randomization is quantitatively confirmed by an entropy value of 7.9880, which is exceptionally close to the theoretical maximum of 8, indicating superior diffusion–confusion properties. The decoded image recovers a histogram profile that almost overlaps the original one. The PSNR of 44.64 dB and the SSIM of 0.9858 confirms that the JPEG-AES pipeline preserves the essential characteristics of intensity with negligible loss.

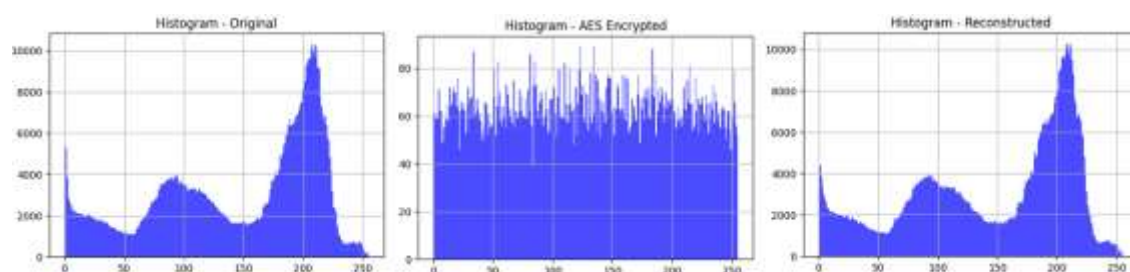


Figure 10. Histogram analysis of the original, encrypted representation, and reconstructed images for the JPEG+AES configuration

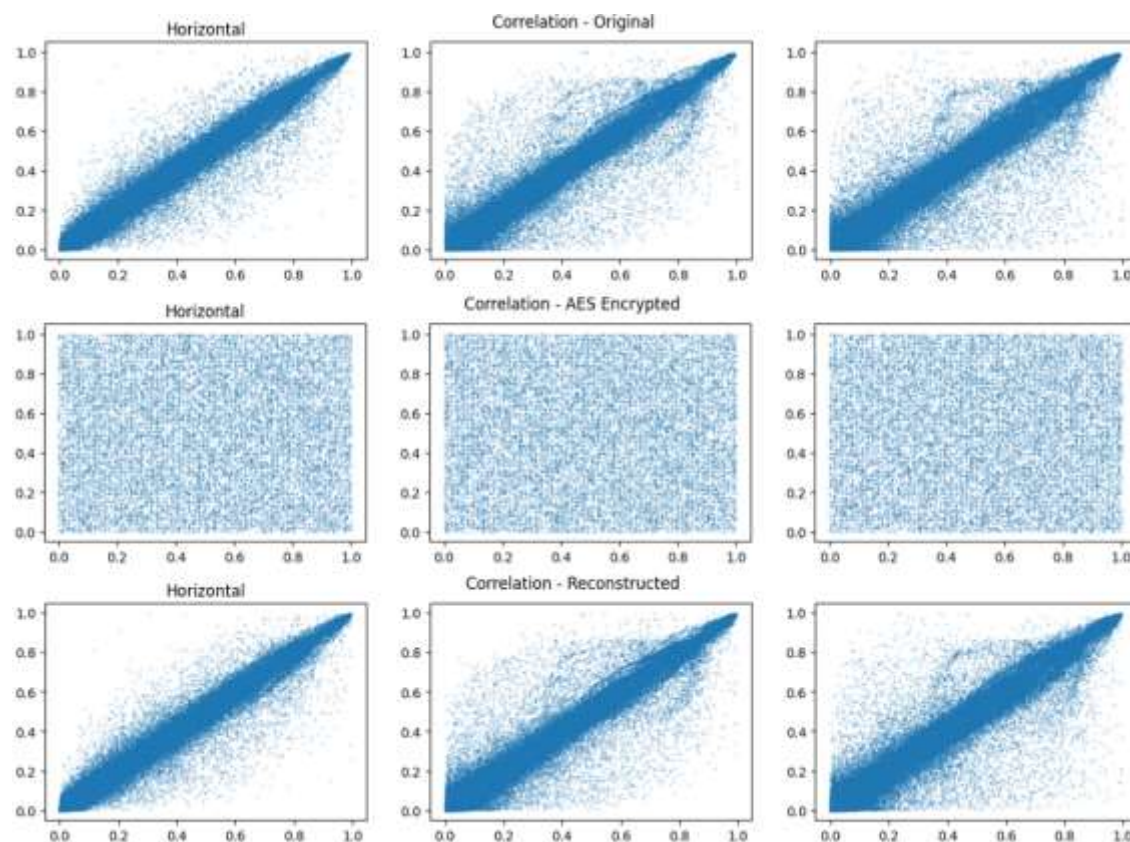


Figure 11. Correlation scatter plots across horizontal, vertical, and diagonal orientations for the JPEG+AES configuration

The correlation analysis also supports the security of the framework. The original image shows strong spatial dependencies in horizontal, vertical and diagonal directions whereas the AES-encrypted domain shows a complete loss of these patterns. The scatter plots are evenly distributed, which means the inter-pixel correlations are effectively removed in agreement with high NPCR (99.56%) and stable UACI (36.59%) values. This total decorrelation will provide a strong resistance to statistical and differential attacks. Finally, the decoded stage is able to recover the original correlation structures successfully, which demonstrates the proposed scheme is able to achieve an optimal tradeoff between high-level encrypted domain security and high-fidelity reconstruction. The statistical intensity distributions and the spatial correlation analysis of the JPEG+AES configuration are shown in Fig. 10 and Fig. 11 respectively.

In the fourth stage of the configuration, a sequential hybrid compression-encryption based on JPEG compression and RSA encryption is applied. In this context, the input images of size 512x512 are first compressed using the JPEG algorithm to get a compact bitstream representation. Then this bitstream is encrypted using the RSA encryption algorithm. Before encryption the compressed data are split into smaller portions as RSA cannot directly process large data blocks. After the encryption process, the data is decrypted and reconstructed in the image domain by JPEG decoding. Unlike AE-based methods, this approach operates directly on compressed data rather than on a learned latent representation space. Figure 12 displays the progressive processing pipeline for the JPEG+RSA setup with the original, JPEG-compressed, RSA-encrypted, RSA-decrypted bitstream and the reconstructed image.

From visual quality perspective, the obtained results reveal that JPEG+RSA configuration is performing very high in reconstruction. The high SSIM (0.9858) and PSNR (44.64 dB) between the original and decoded image demonstrate that the structural and perceptual information is preserved to a great extent in the JPEG compression–decompression process. As can be seen from the very low SSIM (0.0017) and low PSNR (6.05 dB) values between original image and encrypted data, the visual information in the encrypted data is totally distorted at the encryption stage. Likewise, low similarity values obtained between encrypted and decoded data validate that no meaningful visual structure is preserved in the encrypted domain. These results show that the method can achieve both high quality reconstruction and strong visual concealment.

From a security perspective, the JPEG+RSA configuration exhibits an effective transformation behavior. The NPCR and UACI between the original and encrypted data are 98.97% and 40.49%, respectively. This high NPCR and high UACI indicate that the encryption process changes the data significantly and that the differential sensitivity is strong. This property is desirable in the sense of resistance to differential attacks. On the other hand, the low UACI value (0.38%) between original and decoded images shows the accuracy of the reconstruction process. The increment of entropy values in the encrypted domain and their recovery after decryption demonstrate that the system has a balanced behavior in terms of randomness generation and information recovery. Overall, this configuration provides a good trade-off between secure encryption and high fidelity reconstruction performance. The detailed quantitative results obtained for sample Image 1 are summarized in Table 5, where the performance of the JPEG+RSA configuration is shown.

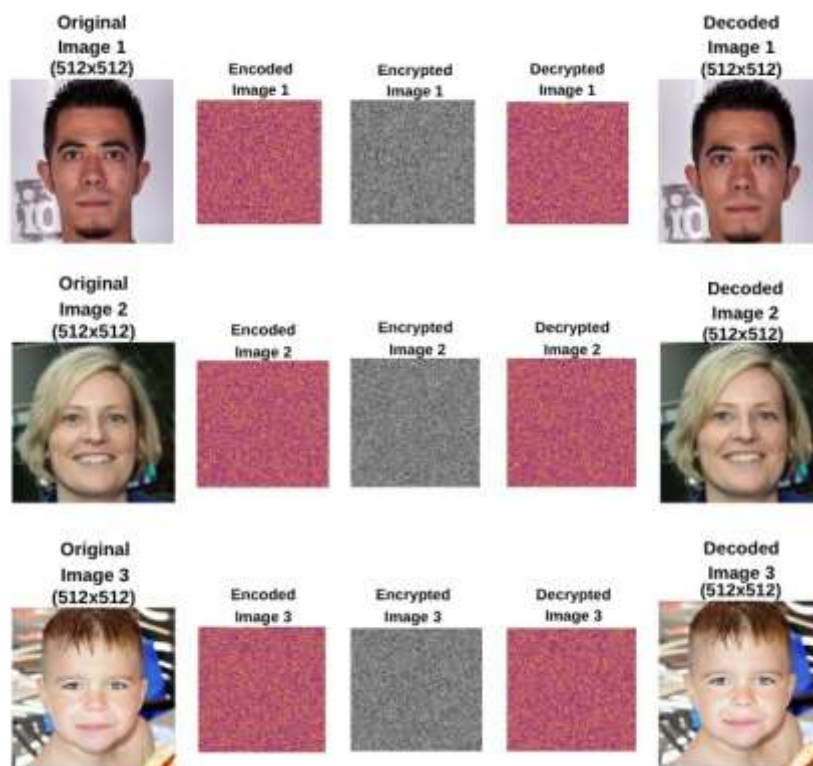
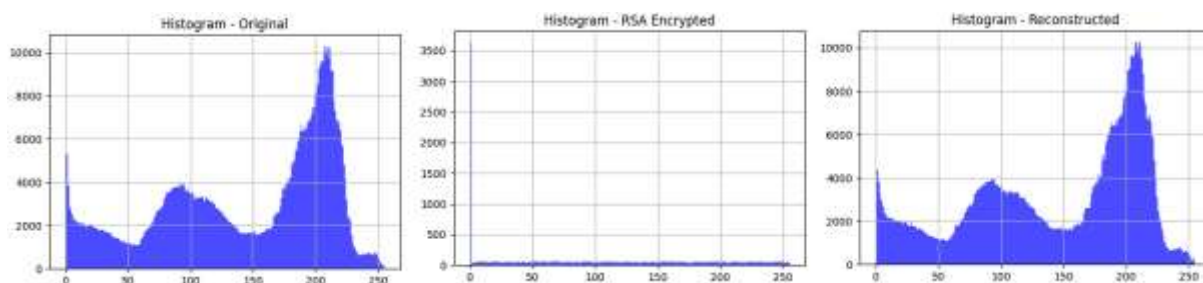
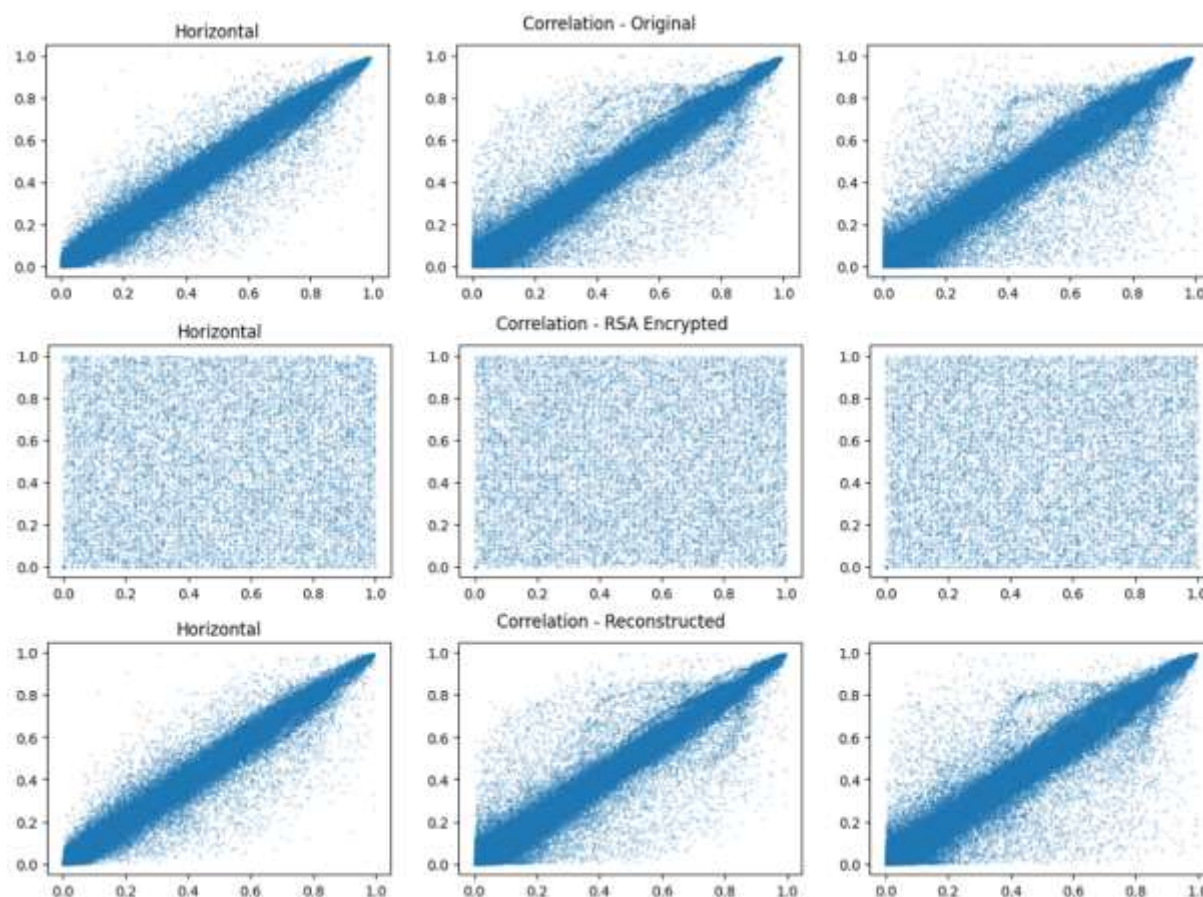


Figure 12. Outputs of the JPEG+RSA configuration image (Fourth stage)

Table 5. Statistical evaluation of the JPEG+RSA configuration on visual quality, security, and entropy results

Comparison Stage	SSIM	± std	PSNR (dB)	± std	NPCR (%)	± std	UACI (%)	± std	Entropy (Ref)	Entropy (Comp)
Original vs Encrypted	0.0017	± 0.02	6.05	± 0.5	98.58	± 0.3	40.48	± 1.2	6.73	6.97
Original vs Decoded	0.9858	± 0.03	44.64	± 0.7	59.34	± 3	0.38	± 0.2	7.56	7.57
Encrypted vs Decoded	0.0019	± 0.02	6.05	± 0.5	99.30	± 0.3	40.49	± 1.3	6.97	6.74

The histogram analysis of the JPEG+RSA configuration indicates a significant statistical change in the encryption process and a high fidelity in the reconstruction process. The histogram of the original image shows a clear multi-modal distribution of pixel intensities, which is effectively neutralized in the RSA-encrypted domain. The encrypted histogram is very uniformly distributed along the intensity range, with a significant increase in entropy (from 6.73 to 6.97) and a particular spike at the zero-intensity level, probably due to the structural padding or header handling of the RSA-JPEG hybrid process. The histogram profile of the reconstructed (decoded) image is almost identical to the original image, which is supported by the impressive PSNR of 44.64 dB and SSIM of 0.9858, indicating that the JPEG based compression with RSA encryption preserves almost all the tonal information.

**Figure 13. Histogram analysis of the original, encrypted representation, and reconstructed images for the JPEG+RSA configuration****Figure 14. Correlation scatter plots across horizontal, vertical, and diagonal orientations for the JPEG+RSA configuration**

The correlation analysis also confirms the security and reconstruction efficiency of the JPEG+RSA scheme. In the original domain, pixels are highly linearly dependent along the horizontal, vertical and diagonal axes, which is characteristic of the high spatial redundancy of natural images. With RSA encryption, these correlations are destroyed completely. This can be observed in the scatter plots which are uniformly distributed over the entire pixel space without any distinguishable patterns. This total decorrelation is proved by very low SSIM (0.0017) and high NPCR (98.96%) against original which denotes high resistance against statistical attacks. We find that the decoded image is able to recover the original spatial correlations and therefore the system is able to handle the non-linearities of the RSA encryption and achieve excellent structural integrity in the final output. Figure 13 and Figure 14 present the graphical results of the statistical intensity distributions and the spatial correlation analysis for the JPEG+RSA configuration, respectively.

The four configurations (AE+AES, AE+RSA, JPEG+AES, JPEG+RSA) can be described along two fundamental axes: (i) the compression method (AE versus JPEG) and (ii) the encryption algorithm (AES versus RSA). These two dimensions directly affect the systems' behavior in terms of the visual quality performance and security level. The reconstruction performance of JPEG-based methods is much higher in terms of visual quality metrics. Specifically, the JPEG+AES and JPEG+RSA configs achieve near-lossless reconstruction with SSIM $\approx$ 0.9858 and PSNR $\approx$ 44.64 dB between original and decoded images. AE-based systems, on the other hand, produced lower values (SSIM $\approx$ 0.883-0.886, PSNR $\approx$ 29.4-29.5 dB) indicating a lower reconstruction accuracy due to the higher compressed and abstract latent representations generated by the learning-based compression stage. However, such behavior can be explained as a consequence of the better representation learning ability of autoencoders which encode the input data into a more compact and high level feature space in the compression phase.

From security metrics view point, all the configurations have a good cryptographic behavior, where the values of NPCR are between about 98.58% to 99.92%, and UACI is between 36% to 77%, which shows a high diffusion and sensitivity to input change. In AE-based systems (AE+AES and AE+RSA), NPCR values near 100% imply that minor perturbations in the latent feature space result in considerable alterations in the encrypted representation, showing high resistance to differential attacks. On the contrary, JPEG-based systems have slightly more stable structural dependencies due to the nature of classical compression, thus leading to balanced UACI characteristics while still maintaining high NPCR values.

Overall, JPEG-based configurations (JPEG+AES and JPEG+RSA) have better visual reconstruction quality while AE-based configurations (AE+AES and AE+RSA) have better feature-level abstraction and representation transformation. All approaches guarantee high security robustness, but AE-based approaches pay more attention to deeper latent-space obfuscation while JPEG-based approaches follow a more traditional compression-encryption paradigm. Thus, this choice is a trade-off between high-fidelity reconstruction (JPEG-based) and deep feature-level security transformation (AE-based).

Table 6. Comparison of performance evaluation rates

Configuration	Encode Time (sec)	Encryption Time (sec)	Decryption Time (sn)	Decode Time (sec)	CPU Usage (%)	RAM Usage (GB)	GPU Usage (GB)
AE+AES	0.0830	0.00170	0.00160	0.07970	7.30	0.01	0.0
AE+RSA	0.0828	1.84250	4.75640	0.08070	7.85	0.01	0.0
JPEG+AES	0.0014	0.00026	0.00020	0.00174	10.0	3.90	1.2
JPEG+RSA	0.0014	0.24592	0.68027	0.00177	10.6	3.46	1.2

The performance analysis shows a basic difference between the transform-based (JPEG) and the neural network-based (AE) approaches in terms of execution speed and resource consumption. JPEG-based configurations offer much better temporal efficiency with encoding and decoding times almost 50 times faster than AE models, but at the cost of high hardware dependency, needing about 3.5–3.9 GB of RAM and 1.2 GB of GPU resources. Conversely, AE-based models demonstrate an extremely lightweight profile with a negligible memory footprint of 0.01 GB and no GPU requirement, making them highly suitable for resource-constrained edge devices despite their higher processing latency. It is worth noting that the RSA decryption process is much faster in the JPEG framework (0.68s) than in the AE framework (4.75s), which means that the latent features in AE are more computationally complex for asymmetric encryption. JPEG+AES is usually the fastest end-to-end pipeline for high-performance systems, but the AE configurations have a tactical advantage in hardware-economy-sensitive environments over raw execution speed. Table 6 Summary of the execution times, hardware resources utilization and compression ratios for the evaluated configurations.

5. CONCLUSION

In this paper, a comprehensive comparative framework for the secure image transmission based on deep learning based (AE) and traditional (JPEG) compression techniques with symmetric (AES) and asymmetric (RSA) encryption standards is proposed. The visual quality, the cryptographic security, and the computational efficiency of four different configurations, AE+AES, AE+RSA, JPEG+AES, JPEG+RSA, were thoroughly investigated. The experimental results indicate a distinct trade-off between reconstruction fidelity and representation abstraction. JPEG-based configurations (JPEG+AES/RSA) achieved superior visual quality with near-lossless reconstruction (PSNR $\approx$ 44.64 dB, SSIM $\approx$ 0.9858). On the contrary, the proposed AE based method resulted in more abstract latent representations and a moderate decrease in reconstruction fidelity (PSNR $\approx$ 29.51 dB). However, all configurations showed strong cryptographic performance from a security point of view with

NPCR values close to 99.9% and high entropy values confirming that the framework successfully hides both pixel-domain and latent-domain information against statistical and differential attacks. The operational efficiency and resource utilization of the models is an important finding of this study. JPEG-based schemes were almost 50 times faster in terms of execution speed, but had a high dependency on hardware resources (3.5–3.9 GB RAM). On the other hand, the AE-based configurations are characterized by an extremely lightweight profile, having a negligible memory footprint (0.01 GB RAM) and no requirement of GPU resources during inference. Hence, the AE-based latent encryption approach is well suited for resource-constrained edge devices and IoT applications, where hardware economy is preferred over raw processing speed.

In summary, the choice of configuration is based on the specific requirements of the application scenario: high-fidelity, high-speed needs are best met by JPEG-integrated models, while AE-integrated models provide a strategic advantage in secure, low-resource environments. In future work, we will aim at improving the reconstruction quality of AE models using generative architectures and will investigate the potential of combining chaotic systems and quantum resilient algorithms to increase the security of the latent feature space even further.

Acknowledgments

The authors would like to acknowledge all those who contributed indirectly to this study.

Authors' Contributions

In this study, authors contributed equally to the study.

Competing Interests

The authors declare that they have no conflict of interest.

References

- [1] Álvarez, G. and Li, S., Some basic cryptographic requirements for chaos-based cryptosystems, *International Journal of Bifurcation and Chaos*, 2006, 16(8), 2129–2151. <https://doi.org/10.1142/S0218127406015970>
- [2] Huang, Q. X., Yap, W. L., Chiu, M. Y. and Sun, H. M., Privacy-preserving deep learning with learnable image encryption on medical images. *IEEE Access*, 2022, 10, 66345–66355. [10.1109/ACCESS.2022.3185206](https://doi.org/10.1109/ACCESS.2022.3185206)
- [3] Zhang, B., Classification of encrypted text based on artificial intelligence. In *IOP Conference Series: Materials Science and Engineering* 2020, January, Vol. 740, No. 1, p. 012133, IOP Publishing. <https://doi.org/10.1088/1757-899X/740/1/01213>
- [4] Sang, Y., Sang, J. and Alam, M. S., Image encryption based on logistic chaotic systems and deep autoencoder, *Pattern Recognition Letters*, 2022, 153, 59–66. <https://doi.org/10.1016/j.patrec.2021.11.025>
- [5] Maung Maung, A. and Kiya, H., Generative model-based attack on learnable image encryption for privacy-preserving deep learning, 2023, arXiv:2303.05036. <https://doi.org/10.48550/arXiv.2303.05036>
- [6] Ahmed, F., Rehman, M. U., Ahmad, J., Khan, M. S., Boulila, W., Srivastava, G. and Buchanan, W. J. A., DNA based colour image encryption scheme using a convolutional autoencoder. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2023, 19(3s), 1–21. <https://doi.org/10.1145/3570165>
- [7] Al-Maadeed, T. A., Hussain, I., Anees, A. and Mustafa, M. T., An image encryption algorithm based on chaotic Lorenz system and novel primitive polynomial S-boxes. *Multimedia Tools and Applications*, 2021, 80, 24801–24822. <https://doi.org/10.1007/s11042-020-08821-w>
- [8] Sayood, K., *Introduction to data compression*. Morgan Kaufmann, 2017.
- [9] Naveen, A. K., Thunga, S., Murki, A., Kalale, M. and Anil, S., Autoencoded Image Compression for Secure and Fast Transmission. In *2024 IEEE International Conference on Computer Vision and Machine Intelligence (CVMi)*, 2024, October, (pp. 1–6). IEEE. DOI: [10.1109/CVMi61877.2024.10781629](https://doi.org/10.1109/CVMi61877.2024.10781629)
- [10] Hinton, G. E. and Salakhutdinov, R. R., Reducing the dimensionality of data with neural networks. *Science*, 2006, 313(5786), 504–507. <https://doi.org/10.1126/science.1127647>
- [11] Oswald, E., Daemen, J., & Rijmen, V., *The Design of Rijndael: AES - The Advanced Encryption Standard*, 2002, Springer Science & Business Media.
- [12] Rivest, R. L., Shamir, A., & Adleman, L., A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 1978, 21(2), 120–126. <https://doi.org/10.1145/358980.358990>
- [13] Karras, T., Laine, S., & Aila, T., A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (FFHQ Dataset)*, 2019, <https://www.kaggle.com/datasets/arnaud58/flickrfaceshq-dataset-ffhq>
- [14] Alslman, Y., Alnagi, E., Ahmad, A., AbuHour, Y., Younis, R., & Abu Al-haija, Q. (2022). Hybrid encryption scheme for medical imaging using autoencoder and advanced encryption standard. *Electronics*, 11(23), 3967.
- [15] Sabha, I., Parah, S. A., Sarosh, P., & Islam, M. O. U. (2024). CESCAL: A joint compression-encryption scheme based on convolutional autoencoder and logistic map. *Multimedia Tools and Applications*, 83(11), 32069–32098.
- [16] Sharma, V., Shrivastava, L., & Shrivastava, N., Multi-Layer Image Security Framework Using Autoencoder Compression and Chaotic-AES Hybrid Security. In *2025 International Conference on Electrical, Communication, and Computing Technologies (iCONECT)*, 2025, (pp. 1-5). IEEE.
- [17] Shariatzadeh, M., Eftekhari, M., & Rostami, M. J., A New Scheme for Image Compression and Encryption Using ECIES, Henon Map, and AEGAN, 2022, arXiv preprint arXiv:2208.07635.
- [18] Gupta, N., & Vijay, R., Hybrid image compression-encryption scheme based on multilayer stacked autoencoder and logistic map, *China Communications*, 2022, 19(1), 238–252.
- [19] Wang, B., & Lo, K. T., Autoencoder-based joint image compression and encryption, *Journal of Information Security and Applications*, 2024, 80, 103680.
- [20] Ahmed, A. M., Bouallegue, B., & Youness, H. A., Image Encryption and Compression Based on Auto Encoder for Real Time Using IoT Technique, *International Journal of Computer Science & Network Security*, 2025, 63–68.
- [21] Wei, J., Zhang, M., & Tong, X. (2022). Multi-image compression-encryption algorithm based on compressed sensing and optical encryption, *Entropy*, 24(6), 784.
- [22] Almola, S. A., Younis, H. A., & Khudeyer, R. S. A Robust Image Encryption Framework Using Deep Feature Extraction and AES Key Optimization, *Cryptography*, 2026, 10(2), 16.

- [23] Lata, K., Gupta, C., & Cenkeramaddi, L. R., A cryptographic framework for secure medical imaging in smart healthcare environments, *Results in Engineering*, 2025, 106780.
- [24] Xiao, W., Wan, N., Hong, A., & Chen, X., A fast JPEG image compression algorithm based on DCT. In 2020 IEEE International Conference on Smart Cloud (SmartCloud), 2020, (pp. 106-110). IEEE.
- [25] Chen, H., He, X., Yang, H., Qing, L., & Teng, Q., A feature-enriched deep convolutional neural network for JPEG image compression artifacts reduction and its applications. *IEEE Transactions on Neural Networks and Learning Systems*, 2021, 33(1), 430-444.
- [26] Berahmand, K., Daneshfar, F., Salehi, E. S., Li, Y., & Xu, Y., Autoencoders and their applications in machine learning: a survey. *Artificial Intelligence Review*, 2024, 57(2), 28. <https://doi.org/10.1007/s10462-023-10400-0>
- [27] Ozturk, E., & Mesut, A., Comparison of learned image compression methods and JPEG. In 2024 Innovations in Intelligent Systems and Applications Conference (ASYU), 2024, (pp. 1-6). IEEE.
- [28] Fraihat, S., & Al-Betar, M. A., A novel lossy image compression algorithm using multi-models stacked Auto Encoders, *Array*, 2023, 19, 100314.
- [29] Zhang, W., Zheng, X., Xing, M., Yang, J., Yu, H., & Zhu, Z., Chaos-Based Color Image Encryption with JPEG Compression: Balancing Security and Compression Efficiency, *Entropy*, 2025, 27(8), 838.
- [30] Salamah, A. H. A. M. (2026). JPEG-Inspired Encoding for Deep Learning.
- [31] Abdmouleh, M. K., Khalfallah, A., & Bouhleh, M. S., A novel selective encryption scheme for medical images transmission based-on JPEG compression algorithm, *Procedia computer science*, 2017, 112, 369-376.
- [32] Kurihara, K., Shiota, S., & Kiya, H. (2015). An encryption-then-compression system for jpeg standard. In *2015 Picture Coding Symposium (PCS)* (pp. 119-123). IEEE.
- [33] Lian, S., Sun, J., & Wang, Z., A novel image encryption scheme based-on JPEG encoding. In *Proceedings. Eighth International Conference on Information Visualisation, 2004. IV 2004, 2004*, (pp. 217-220). IEEE.
- [34] Elkabbany, G. F., Aslan, H. K., & Rasslan, M. N. (2015). A design of a fast parallel-pipelined implementation of AES: Advanced Encryption Standard. *arXiv:1501.01427*. <https://doi.org/10.48550/arXiv.1501.01427>
- [35] Smid, M. E., Development of the advanced encryption standard. *Journal of Research of the National Institute of Standards and Technology*, 2021, 126, 126024. <https://doi.org/10.6028/jres.126.024>
- [36] Kaliski, B., The mathematics of the RSA public-key cryptosystem. RSA Laboratories, 2006.
- [37] Nitaj, A., The mathematical cryptography of the RSA cryptosystem. In *Cryptography: Protocols, Design and Applications*, 2012, (pp. 159–186).
- [38] Renieblas, G. P., Nogués, A. T., González, A. M., Gómez-León, N., & Del Castillo, E. G., Structural similarity index family for image quality assessment in radiological images. *Journal of Medical Imaging*, 2017, 4(3), 035501. <https://doi.org/10.1117/1.JMI.4.3.035501>
- [39] Alsaraireh, S., Ahmad, A., & AbuHour, Y., New step in lightweight medical image encryption and authenticity. *Mathematics*, 2025, 13(11), 1799. <https://doi.org/10.3390/math13111799>
- [40] Susanto, A., Rachmawanto, E. H., Mulyono, I. U. W., Sari, C. A., Sarker, M. K., & Sazal, M. R., Triple layer image security using bit-shift, chaos, and stream encryption. *Bulletin of Electrical Engineering and Informatics*, 2020, 9(3), 980–998. <https://doi.org/10.11591/eei.v9i3.2492>
- [41] Hussein, M. K., Hassan, K. R., & Al-Mashhadi, H. M., The quality of image encryption techniques by reasoned logic. *TELKOMNIKA*, 2020, 18(6), 2992–2998. <https://doi.org/10.12928/TELKOMNIKA.v18i6.14114>
- [42] Shakhmetova, G., Barlybayev, A., Saukhanova, Z., Sharipbay, A., Raykul, S., & Khassenov, A., Enhancing visual data security: A novel FSM-based image encryption and decryption methodology. *Applied Sciences*, 2024, 14(11), 4341. <https://doi.org/10.3390/app14114341>