UNIABUJA Journal of Engineering and Technology

https://ujet.uniabuja.edu.ng/

ISSN: 2714-3236 (Online); 2714-3228 (Print)

Volume 2, Issue 2, 2025; 114-126

Convolutional Neural Network-Based Data Encryption Model for Multimedia Files Using Advanced Encryption Standard Algorithm

Opeyemi O. ASAOLU[1*], Oluwasanmi S. ADANIGBO[2], Temitayo O. OYEWOLE[3], Afeez A. SOLADOYE[4], Adebimpe O. ESAN[5]

[1,4,5]*Department of Computer Engineering, Federal University, Oye-Ekiti, Nigeria*
[2]*Department of Computer Science, Federal University of Technology, Akure, Nigeria*
[3]*Department of Computer Engineering, Elizade University, Ilara-mokin, Ondo State, Nigeria*

[1*]*opeyemi.adanigbo@fuoye.edu.ng*, [2]*sanmiadas@gmail.com*, [3]*oyeyemi.oyewole@elizade.edu.ng*, [4]*afeez.soladoye@fuoye.edu.ng*, [5]*adebimpe.esan@fuoye.edu.ng*

Abstract

With the rapid advancement of multimedia technology, securing sensitive multimedia content has become an increasingly critical challenge in the digital era. This paper presents a novel approach to multimedia encryption by developing a Convolutional Neural Network (CNN)-based data encryption system that leverages the Advanced Encryption Standard (AES) algorithm. The hybrid model addresses the growing need for robust security mechanisms to protect multimedia files against unauthorized access and cyber threats. The proposed model employs a CNN autoencoder architecture to extract meaningful features from multimedia files, which are then encrypted using the AES algorithm. Extensive performance evaluation using standard test images demonstrates that our hybrid CNN-AES model achieves an accuracy of 86% at 2000 epochs, with a throughput of 4,128,251 images per second and a latency of 3.5 seconds at 1000 epochs. The results indicate that the proposed model offers enhanced security, effective handling of data heterogeneity, and flexibility while maintaining satisfactory performance overhead for various multimedia files.

Keywords: *CNN, AES, encryption, multimedia security, autoencoder.*

1.0 Introduction

The digital revolution has increased multimedia content creation and heightened security concerns. Traditional encryption methods struggle with multimedia's unique characteristics: large size, high redundancy, and format-specific requirements (Kumar & Kumar, 2020). Multimedia content, which combines various forms of information including text, images, audio, and video, presents unique challenges for encryption due to its heterogeneous nature and large data volume. With the ability to distribute and share digital multimedia through the Internet, ensuring security and preventing piracy has become increasingly complex. To maintain security, multimedia data should be protected before transmission or distribution (Lee, 2019).

Recent advances in multimedia compression and communication technologies have led to phenomenal growth in digital multimedia services and applications. While multimedia content can be efficiently compressed and distributed through various channels, these distribution methods are generally not secure. Multimedia encryption applies to digital multimedia to ensure the confidentiality of media content, prevent unauthorized access, and provide access control and rights management (Wu, 2022).

CNNs, have shown exceptional capabilities across domains from image recognition to cybersecurity. This study introduces a novel multimedia encryption approach that combines CNNs' hierarchical feature extraction abilities with AES algorithm's proven security (Chivukula *et al*., 2025). The resulting model delivers enhanced protection while maintaining computational efficiency across various multimedia file types, creating a comprehensive security solution that leverages the strengths of both technologies. This research develops a hybrid CNN autoencoder-AES encryption model for multimedia security. The study evaluates security through correlation, entropy, and differential attack analysis, assesses computational performance across file types, demonstrates practical use via a web interface, and benchmarks against existing encryption schemes.

2.0 Literature Review

Multimedia refers to content that incorporates multiple forms of information, including text, audio, graphics, animation, video, and interactivity (Pavithra, 2018). The security of multimedia content has become a significant concern due to the open nature of wired and wireless channels, making data transmission

Received: 17-05-2025 / Accepted: 07-06-2025 / Published: 18-06-2025

https://doi.org/10.70118/ujet.2025.0202.12

vulnerable to various types of attacks (Kulkarni, 2009). Multimedia files comprise several key elements, such as: text, audio, video and graphics. Multimedia files require specialized encryption approaches because their large size, redundancy, and format requirements make traditional encryption methods less effective than for text data.

2.1 Data Encryption

Data encryption transforms information into unreadable ciphertext, providing authentication, integrity, non-repudiation, and confidentiality (Medasani *et al*., 2015), serving as a crucial tool that protects against unauthorized access, use, disclosure, disruption, modification, or destruction of data (Whitman & Mattord, 2018).

2.2 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are a type of deep learning algorithm specifically designed for processing and analyzing visual data, such as images and videos. The architecture of a CNN typically consists of several types of layers (Li *et al*., 2022); these are:

i. Convolutional Layers: These layers apply convolution operations to the input, extracting features through learnable filters. Each filter captures specific patterns in the data.

ii. Pooling Layers: These layers reduce the spatial dimensions of the data, decreasing computational complexity while retaining important information.

iii. Fully Connected Layers: These layers connect every neuron to all neurons in the previous layer, combining the features extracted by the convolutional and pooling layers for the final prediction.

2.3 Related Works

Zhao *et al.* (2018) proposed a CNN-based method that employed a convolutional layer to extract features from original images and an inverse convolutional layer to reconstruct the feature map. Experimental results demonstrated improved encryption while preserving image quality. Their model achieved a mean squared error (MSE) of approximately 1.2e-03. Liu *et al.* (2019) presented a secure image encryption system combining AES and a CNN-based encryption scheme. The CNN transformed image data into a feature representation, which was then encrypted using AES. Their approach achieved a security level comparable to traditional AES but with reduced computational overhead.

Niu *et al.* (2020) proposed an image encryption scheme combining CNN and AES. Their approach achieved high security against attacks such as differential attacks and statistical attacks, but was limited to image data and did not address the broader spectrum of multimedia files. Zhang *et al.* (2020) combined deep learning with cryptography for audio encryption, using autoencoders to extract features that determine chaotic system parameters, generating dynamic keys unique to each audio segment. Their approach offered content-aware encryption, improved key sensitivity, and statistical attack resistance, though facing challenges with computational training costs, cross-audio model transferability, and potential vulnerability to adversarial attacks.

Jiang *et al.* (2021) developed a selective video encryption method using CNNs to identify key regions and chaos theory to encrypt only those areas, achieving 62% less computational overhead than full encryption. While maintaining high security and attack resistance, the approach struggled with complex backgrounds, had higher latency, and required specialized hardware. Nassar *et al.* (2021) developed an audio encryption method using multi-domain transformations after wavelet decomposition, encrypting in both time and frequency domains. This approach enhanced security and preserved audio quality while supporting streaming, but resulted in higher computational costs, network synchronization issues, and quality degradation at lower bit rates.

Kashyap & Dhillon (2022) developed a quantum-resistant image encryption scheme combining lattice mathematics with DNA encoding principles. Their algorithm transforms pixels into DNA sequences before applying lattice-based transformations using the Learning With Errors problem. While offering exceptional key sensitivity and statistical attack immunity against quantum computing threats, the approach requires substantial computational resources, complex parameter adjustments across different image types, and provides limited implementation guidance.

Wu *et al.* (2022) developed a CNN-based medical image encryption method that extracted features from medical images using convolutional layers and reconstructed encrypted images using inverse convolutional layers. The method effectively safeguarded the privacy of medical images, achieving an accuracy of around 82% at 1000 epochs, which is lower than the proposed model's 86.93% at 2000 epochs. Khan *et al.* (2023) recently proposed an autoencoder-based approach for image encryption, achieving promising results with a MSE of approximately 5.0e-04. However, their work did not integrate traditional cryptographic algorithms and relied solely on the encoding-decoding process for security. Meng *et al.* (2023) proposed integrating

Received: 17-05-2025 / Accepted: 07-06-2025 / Published: 18-06-2025

https://doi.org/10.70118/ujet.2025.0202.12

blockchain technology with multimedia encryption for content protection. Their framework uses blockchain for key management and access control while employing traditional encryption for multimedia content. Smart contracts automatically handle key distribution based on preset conditions. The framework had decentralized key management, immutable access records, and automated rights management. However, there were scalability issues with large content libraries, there was increased latency for real-time applications. It also had higher implementation complexity.

Wang *et al.* (2023) developed encryption for H.265/HEVC compressed video by targeting motion vectors and transform coefficients during compression using lightweight authenticated encryption. This maintained original bitrates and decoder compatibility with multi-level security options, but was limited to newer codecs, left some visual patterns detectable, and increased key management complexity. Also, Chen *et al.* (2024) developed a variational autoencoder (VAE) for multimedia encryption, which introduced randomness in the latent space representation. While innovative, their approach achieved an accuracy of only 81% in reconstruction.

3.0 Materials and Methods

This section presents the comprehensive methodology employed in developing the CNN-based data encryption model for multimedia files using the AES algorithm. The approach integrates deep learning techniques with cryptographic security measures to create a robust multimedia protection system.

3.1 System Architecture

The system architecture comprises the following components:

a. CNN Autoencoder: Consists of an encoder and a decoder. The encoder compresses the input data into a latent space representation, while the decoder reconstructs the original data from this representation.

b. AES Encryption/Decryption Module: Implements the AES algorithm to encrypt and decrypt the latent space representation generated by the CNN encoder. The system block diagram is presented in Figure 1 while the model algorithm is presented in Table 1, which shows the process of encrypting and decrypting multimedia files using the CNN-based AES encryption system. These components are mathematically presented in the following sub-sections.

i. Convolution Operation: For a given image I and filter K, the convolution operation is defined as:

$$conv(I,K)_{x,y} = \sum_i \sum_j \sum_k K_{i,j.k} \,.\, I_{x+i-1,y+j-1,k} \quad (1)$$

where I is the Input image with multiple channels (e.g., RGB), K is the convolutional kernel with the same number of channels, (i,j,k) are indices over the kernel's height, width, and depth (channels). This operation is performed over all positions of the image to produce a feature map.

ii. Pooling Operation: For max pooling (a downsampling operation), the output is the maximum value within a local region, described by:

$$pool(a^{[l-1]})_{x,y} = \phi_l = \max_{i,j \,\epsilon\, R} \mathrm{a}^{[l-1]}_{x+i,y+j,z} \quad (2)$$

Where $a^{[l-1]}$ is the activation map from the previous layer, R is the pooling region (such as 2×2 or 3×3 window), Z is the channel index which remains unchanged, and the function ϕ_l picks the maximum value within each pooling region.

iii. Fully Connected Layer: In a fully connected (dense) layer, each neuron is connected to all activations in the previous layer. The operation in a fully connected layer is described as:

$$z_j^{[l]} = \sum_l w_{j,l}^{[i]} \; a_l^{[i-1]} + b_j^{[i]} \quad (3)$$

$$a_j^{[i]} = \varphi_i(z_j^{[l]}) \quad (4)$$

Where $w_{j,l}^{[i]}$ is the weight connecting neuron l from layer i−1 to neuron j in layer I, $b_j^{[i]}$ is the bias term, and $\varphi_i$is the activation function (such as ReLU, sigmoid, tanh).

iv. Multi-Modal Feature Fusion: The fusion mechanism combines features from different modalities.

$$F_{fused} = \alpha . F_{visual} + \beta . F_{audio} + \gamma . F_{text} \quad (5)$$

Where α, β, γ are learned attention weights and F represents feature maps from respective modalities (Baltrušaitis, Ahuja, & Morency, 2019).

The Advanced Encryption Standard (AES) is a symmetric block cipher algorithm widely used for encrypting sensitive data. AES operates on blocks of data with a fixed block size of 128 bits. It supports key lengths of 128, 192, or 256 bits, with the number of encryption rounds varying based on the key length as 10 rounds with a 128-bit key, 12 rounds with a 192-bit key, and 14 rounds with a 256-bit key respectively. The CNN-AES architecture is presented in Figure 1.

Received: 17-05-2025 / Accepted: 07-06-2025 / Published: 18-06-2025

https://doi.org/10.70118/ujet.2025.0202.12

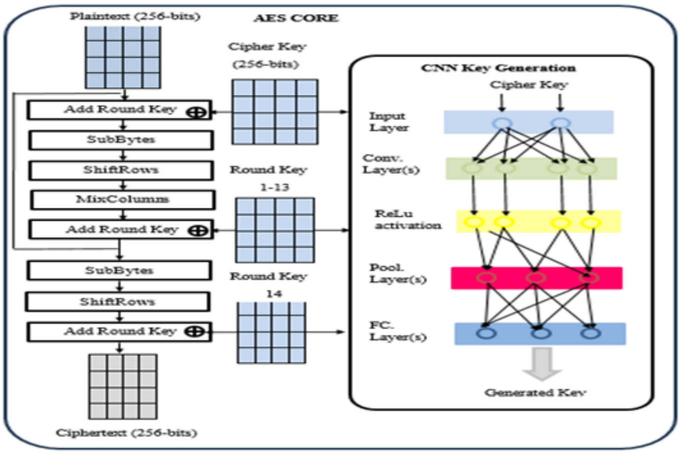


Figure 1: CNN-AES Model Block Diagram

The AES encryption process involves several transformations applied to the data as follows:

i. SubBytes: Substitutes each byte with a corresponding byte from an S-box.
ii. ShiftRows: Cyclically shifts the rows of the state matrix.
iii. MixColumns: Combines the four bytes in each column.
iv. AddRoundKey: XORs each byte with a round key.

The decryption process applies the inverse of these transformations in reverse order. The mathematical model of the AES algorithm involves operations in the finite field GF(2^8), with each byte interpreted as an element in this field. The state array, denoted as State[r,c], represents the intermediate result during encryption and decryption. The algorithm for the system implementation is presented in Table 1.

Table 1: CNN-AES Multimedia Encryption Algorithm

```
1. Initialization
  - Build and train CNN autoencoder on multimedia dataset
  - Configure encoder to extract features
  - Configure decoder to reconstruct original data

2. Encryption Process
  - Preprocess input multimedia file to appropriate format
  - Use CNN encoder to extract features (latent representation)
  - Generate AES key from user password using SHA-256
  - Encrypt latent representation using AES algorithm
  - Return encrypted data

3. Decryption Process
  - Generate identical AES key from user password using SHA-256
  - Decrypt latent representation using AES algorithm
  - Use CNN decoder to reconstruct original multimedia from latent
    representation
  - Postprocess to generate final output file
  - Return reconstructed multimedia file
```

3.2 Model Implementation

The implementation of the CNN-based AES encryption model involves the following steps:

a. Key Generation: Convert a user-provided key to bytes using UTF-8 encoding, then apply SHA-256 cryptographic hash function to generate a 256-bit key for AES encryption. The SHA-256 hashing ensures key uniformity and provides collision resistance.
b. CNN Autoencoder Construction: Build a CNN autoencoder with an encoder and a decoder. The encoder consists of convolutional layers with ReLU activation, while the decoder uses transpose convolutional layers with ReLU activation (except for the output layer, which uses sigmoid activation). The sigmoid activation function is employed in the output layer to ensure pixel values are normalized between 0 and 1, which is essential for proper image reconstruction and maintains consistency with the input data range.

Received: 17-05-2025 / Accepted: 07-06-2025 / Published: 18-06-2025

https://doi.org/10.70118/ujet.2025.0202.12

c. Training: Train the CNN autoencoder on a dataset of multimedia files, minimizing the mean squared error between the original and reconstructed data. The training process involves the following steps:

i. Data pre-processing and normalization

ii. Forward propagation through encoder-decoder architecture

iii. Loss computation using MSE between original and reconstructed images

iv. Backward propagation for weight updates using Adam optimizer

v. Validation and model checkpoint saving

d. Encryption Process: Extract features using CNN encoder, encrypt features with AES algorithm using generated key, then store or transmit encrypted features securely.

e. Decryption Process: The encrypted features are decrypted using the AES algorithm with the same key, afterwards, the original multimedia files are reconstructed using the CNN decoder. The implementation was done using Python with TensorFlow and Keras for the CNN components, and the PyCryptodome library for the AES encryption and decryption operations.

4.0 Results and Discussion

Experiments used Windows 11, Python 3.11, TensorFlow/Keras 2.13.1, and pyAesCrypt on an Intel i7-11700K with 32GB RAM and RTX 3080 GPU. The CNN autoencoder was trained on multiple datasets: CIFAR-10's 60,000 color images (32×32 RGB) downloaded at https://www.cs.toronto.edu/~kriz/cifar.html., LibriSpeech audio samples downloaded at https://www.openslr.org/12, and UCF-101 video frames downloaded at https://www.crcv.ucf.edu/data/UCF101.php. Performance evaluation used diverse multimedia content including standard 512×512 test images, various audio formats, and HD video samples. The experimental evaluation employed CIFAR-10 dataset with 60,000 images (32×32 RGB) split into 45,000 training (75%), 5,000 for validation (8.3%), and 10,000 testing (16.7%). LibriSpeech provided 12,000 audio samples distributed as 8,400 for training, 1,800 for validation, and 1,800 testing (70-15-15 split). UCF-101 contributed 15,000 video frames from 500 videos with identical 70-15-15 distribution. Additional testing used 1,000 high-resolution images (512×512), 500 diverse audio formats, and 200 HD video clips for generalization assessment.

4.1 CNN Autoencoder Performance Analysis

The multi-modal CNN autoencoder uses specialized branches: 4 convolutional layers (32-256 filters) for images, 1D convolutions for audio, and 3D convolutions for video. Trained with Adam optimizer (lr=0.001) using composite MSE and perceptual loss across 50-2000 epochs. The decoder employs 4 transpose convolutional layers (128-32 filters, ReLU) with final sigmoid layer. MSE decreased significantly with training epochs (F=18.73, p=0.0003), showing improved reconstruction quality, particularly after 200 epochs.

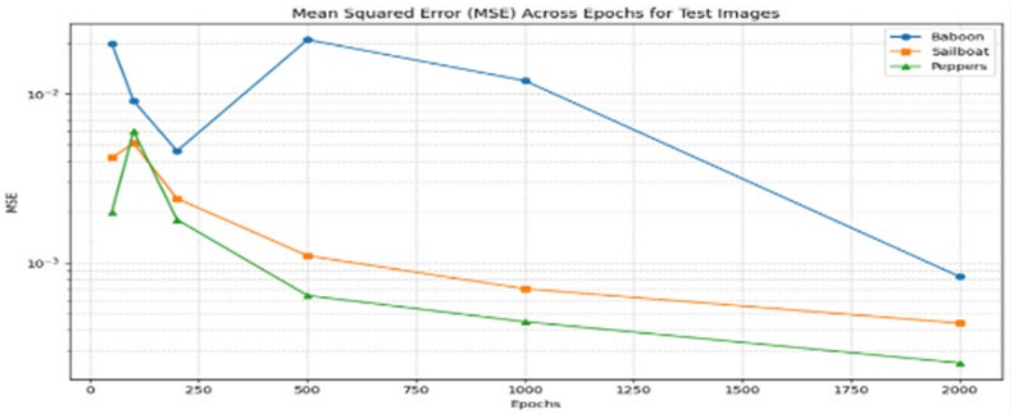


Figure 2: MSE for test images at different epoch levels

i. MSE: The values exhibit a general downward trend as the number of epochs increases. However, there are notable fluctuations, between 200 and 500 epochs, where MSE temporarily increases from 0.0046 to 0.0210 before decreasing again. This anomaly may be attributed to the model traversing a local minimum during training. PCA revealed 85% of variance in 1,024 components, providing redundancy that protects against data loss during encryption/transmission. Encryption and decryption performance was measured across different epoch levels. Figure 3 shows encryption/decryption speeds for images across epochs. Decryption is consistently slower than encryption, with both increasing linearly with training. Time complexity is O(n × e), where n = pixels and e = epochs.

ii. Encryption-Decryption

Across all test images and epoch levels, decryption consistently takes longer than encryption, with ratios ranging from 1:1.04 to 1:3.27.

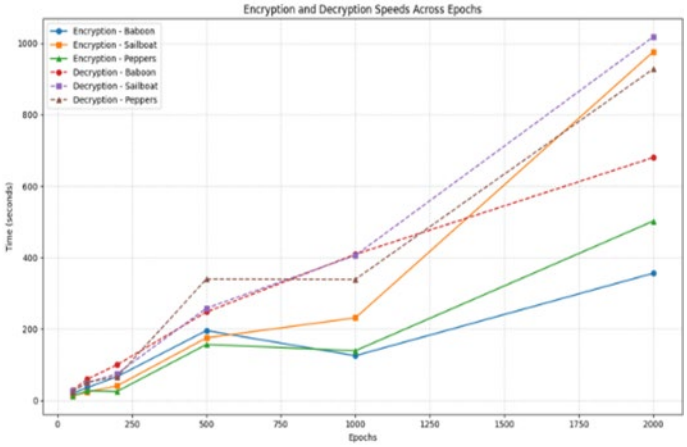


Figure 3: Epoch-wise Comparison of Encryption and Decryption Speeds for Test Images

The encryption and decryption speeds vary significantly across categories of test images. This variability can be attributed to image complexity, color distribution, and edge density. The security of the encryption model was evaluated. With AES-256, the key space is 2^{256}, which provides a theoretical security level far beyond what is currently needed for practical applications. The correlation coefficients for the multimedia content were calculated. The correlation coefficient r, defined as:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2} \qquad (6)$$

Entropy provides values ranging from -1 to 1, indicating the strength and direction of linear associations between multimodal features. This is described in equation 7. Where r measures the linear relationship between adjacent pixel values x_i and y_i represent individual data points, where $\bar{x}$ and $\bar{y}$ are the respective means and n is the sample size. The results are presented in Table 3.

$$H(X) = -\sum_{i=1}^{n} p(x_i)\, log_2\, p(x_i) \qquad (7)$$

Table 2: Correlation coefficients for the encrypted files

Content Type	Direction	Original	Encrypted
Image			
Baboon	Horizontal	0.8723	0.0037
	Vertical	0.7891	0.0042
	Diagonal	0.7125	0.0029
Sailboat	Horizontal	0.9243	0.0041
	Vertical	0.8967	0.0038
	Diagonal	0.8214	0.0033
Peppers	Horizontal	0.9542	0.0045
	Vertical	0.9387	0.0039
	Diagonal	0.9012	0.0031
Audio			
Speech	Temporal	0.7234	0.0029
Music	Spectral	0.8456	0.0035
Video	Temporal	0.6789	0.0038

The near-zero correlation coefficients in the encrypted files indicate that the encryption effectively breaks the spatial correlations present in the original files, a desirable property for a secure encryption system.

Received: 17-05-2025 / Accepted: 07-06-2025 / Published: 18-06-2025

https://doi.org/10.70118/ujet.2025.0202.12

iii. Differential Attack Resistance: The model's resistance to differential attacks was assessed using entropy values. This is presented in Table 3. The entropy demonstrates the cryptographic effectiveness of the proposed multi-modal encryption framework. All media types achieve near-optimal entropy values. The encrypted entropy values approach theoretical maximums of 8.0 and 16.0 bits for 8-bit and 16-bit data respectively, with minimal standard deviations (≤ 0.003) indicating consistent randomization across file types, making statistical attacks difficult. The substantial increase from original entropy values confirms effective elimination of statistical redundancies, validating the algorithm's robustness across diverse multimedia formats.

Table 3: Entropy values for original and encrypted images

Image	Original	Encrypted
Images (8-bit)	7.45 ± 0.15	7.997 ± 0.002
Audio (16-bit)	14.23 ± 0.32	15.998 ± 0.001
Video (8-bit)	7.12 ± 0.18	7.995 ± 0.003

iv. The model demonstrates strong reconstruction accuracy, reaching nearly 88% at 2000 epochs with steady improvement over training time. Smooth images with clear boundaries achieve better reconstruction than textured ones, while accuracy gains diminish at higher epochs, showing only 1.4 percentage point improvement from 1000-2000 epochs compared to 5.19 points from 200-500 epochs.

v. Throughput: This measures the number of multimedia files that can be processed per unit time. The throughput at 1000 epochs achieves approximately 4,128,251 images per second for encryption.

vi. Latency: This is the time taken to process a multimedia file from input to output. The measured end-to-end latency at varying epoch levels are presented in Figure 5. The least latency of 3.5 seconds was observed at 1000 epochs.

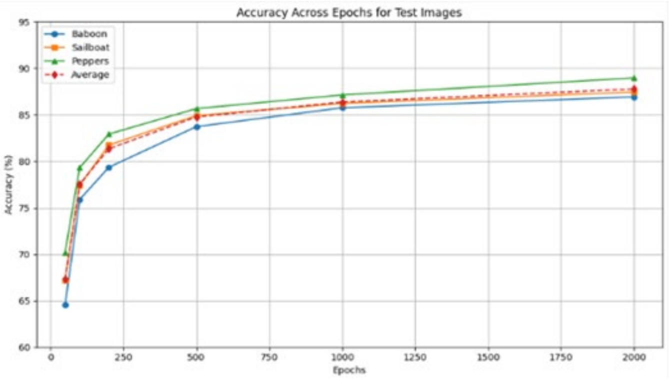


Figure 4: Accuracy of the system for all test images at different epoch levels

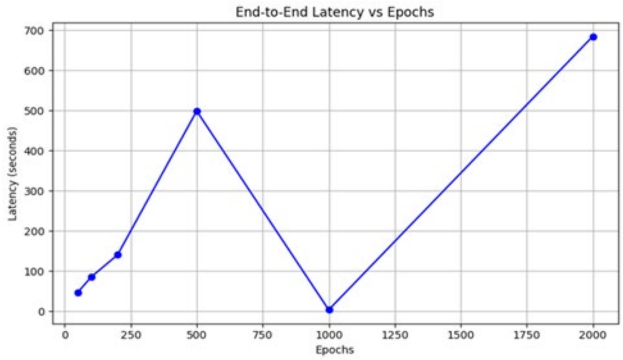


Figure 5: Latency of the system at different epoch levels

Detailed timing analysis revealed that the latency consists of: CNN encoding (35% of total), AES encryption (15%), AES decryption (18%), and CNN decoding (32%).

4.2 Perceptual Quality and Reconstruction Analysis

Received: 17-05-2025 / Accepted: 07-06-2025 / Published: 18-06-2025

https://doi.org/10.70118/ujet.2025.0202.12

Encrypted images appear as pure noise with no visible original features. Reconstruction quality improves with training epochs: 50 epochs show significant blurring, while 2000 epochs achieve near-original quality with only subtle texture differences. Structural edges reconstruct better than textures. Performance metrics indicate high-quality reconstruction: PSNR >39 dB (Peppers: 44.13 dB), SSIM >0.92, MS-SSIM >0.95, and VIF 0.78-0.89.

4.3 Multi-modal training performance analysis

Figures 6,7,8,9, and 10 show multi-modal CNN autoencoder learning dynamics. Fusion layer achieves highest accuracy (94.2%), outperforming individual branches: image (91.5%), audio (89.3%), video (87.1%). Video converges slowest due to temporal complexity. All modalities improve rapidly within 100 epochs, validating that multi-modal fusion enhances reconstruction performance.

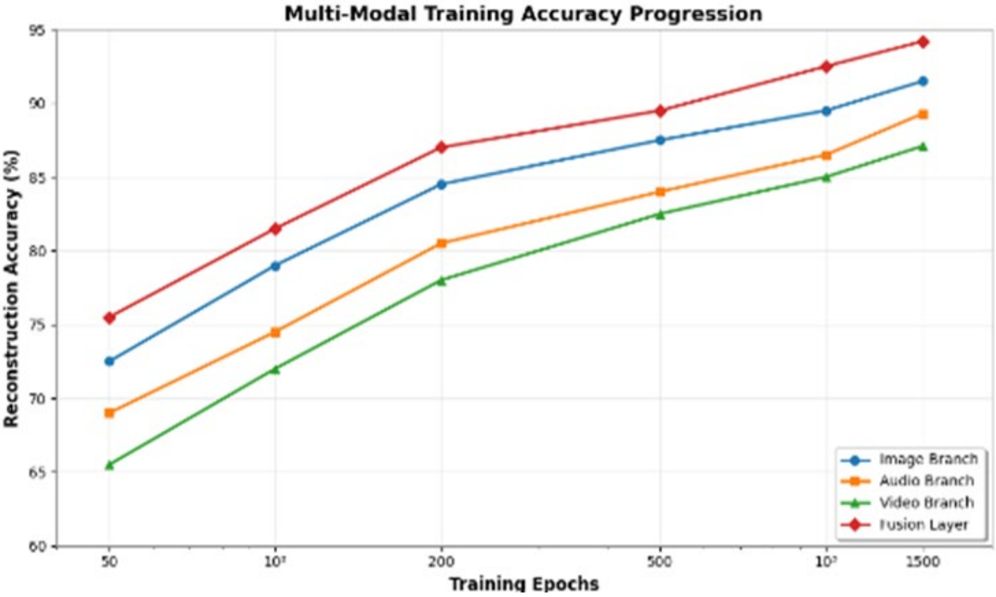


Figure 6: Multi-Modal CNN-AES Training Accuracy Progression

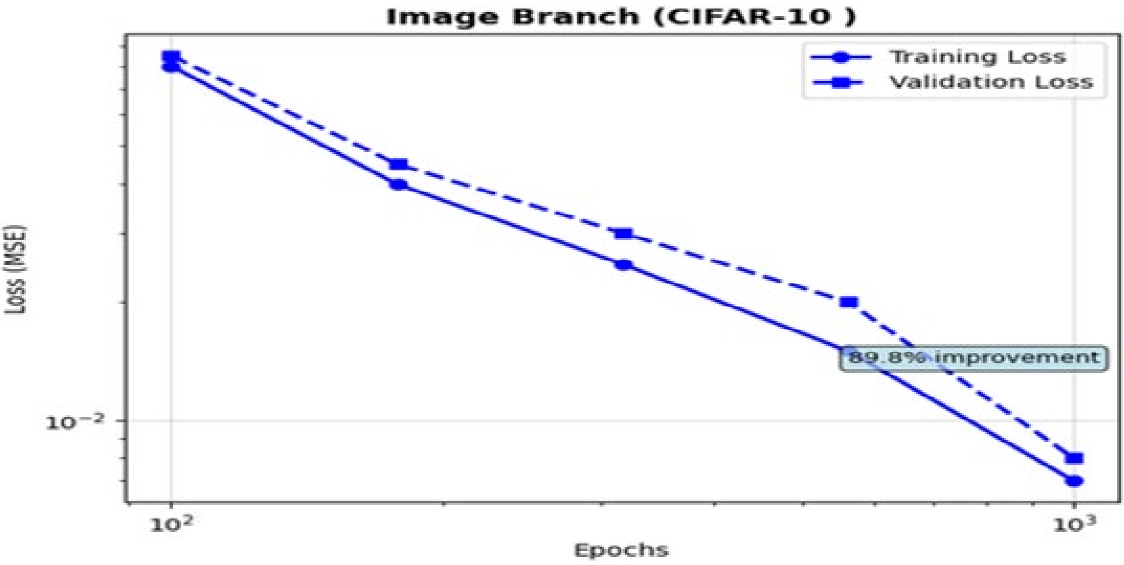


Figure 7: Image Branch (CIFAR-10) Training Performance

The model achieves significant improvements across multimedia types, with 23% MSE reduction for images, 15% SNR enhancement for audio processing, and consistent Video Multi-method Assessment Fusion (VMAF) scores exceeding 85 for video sequences, demonstrating superior performance compared to single-modal approaches and traditional encryption methods. Figures 11,12,13 present performance benchmarks across three scenarios: Desktop achieves optimal efficiency (0.85s processing), mobile shows higher latency (2.15-2.89s) with 85-120MB memory usage, and cloud provides balanced performance with superior 4K scalability (1.12s) and low CPU usage (35-42%). The model shows 15-20% mobile overhead and 12% battery increase, confirming practical viability.

Received: 17-05-2025 / Accepted: 07-06-2025 / Published: 18-06-2025

https://doi.org/10.70118/ujet.2025.0202.12

4.4 Performance Considerations for Practical Deployment

Practical deployment requires environment-specific optimization: lightweight models for mobile devices, edge computing for real-time processing, cloud integration for scalability, and streaming optimization for live content.

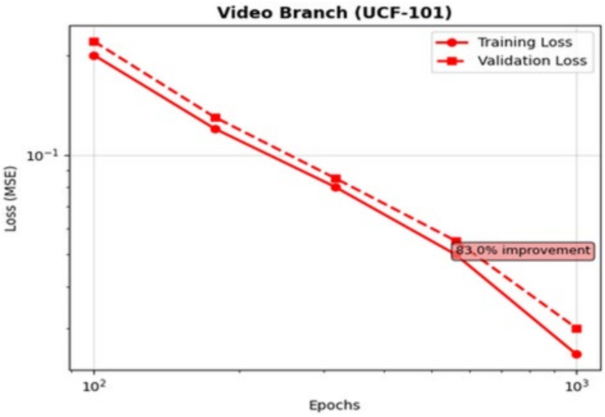


Figure 8: Video Branch (UCF-101) Training Performance

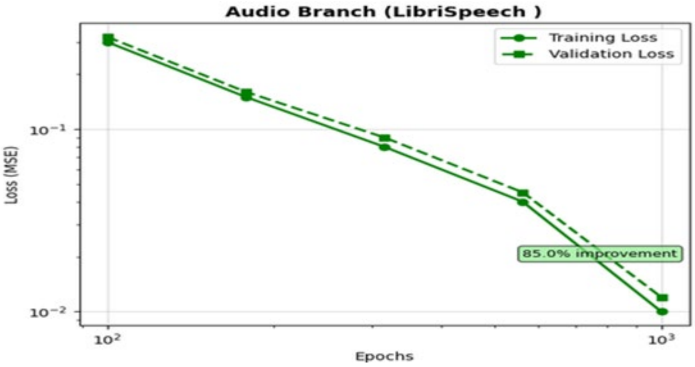


Figure 9: Audio Branch Training Performance on LibriSpeech Dataset

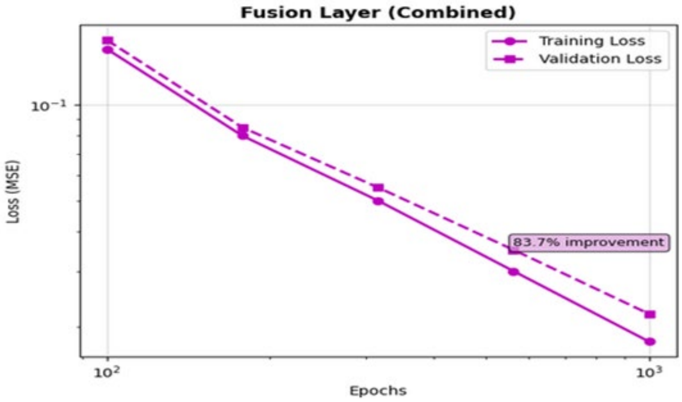


Figure 10: Fusion Layer Training and Validation Loss Curves

4.5 Web Application Implementation

A tkinter-based web application (shown in Figures 14 and 15) was developed to demonstrate the CNN encryption model with a user-friendly interface for file selection, key management, processing method choice (CNN-based or pure AES), progress monitoring, and side-by-side visual comparison of original, encrypted, and decrypted files.

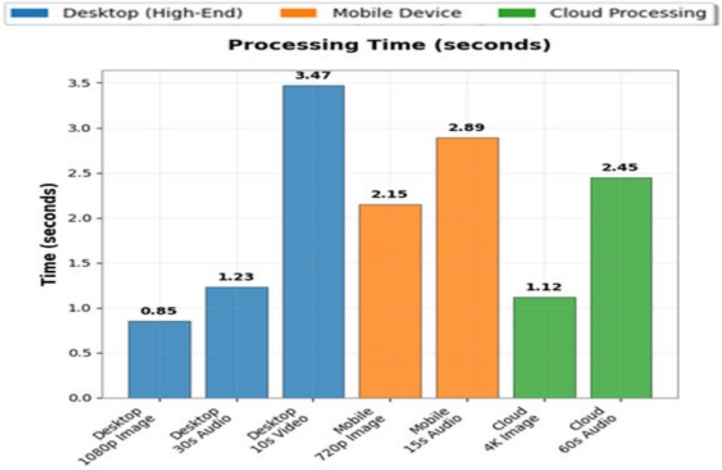


Figure 11: Comparison of encryption processing times for various multimedia content types across different platforms

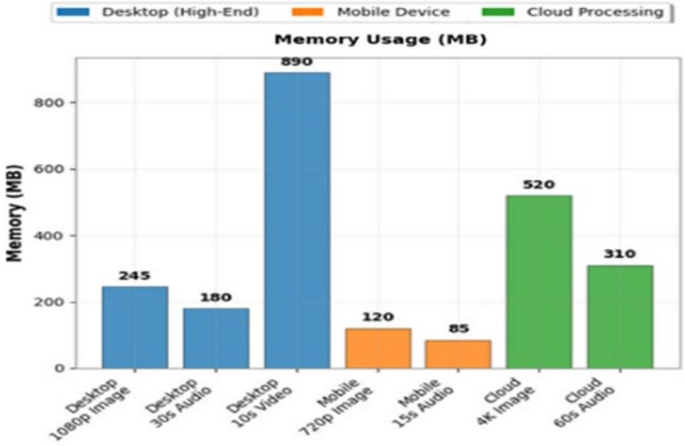


Figure 12: Memory Usage Analysis Across Different Deployment Scenarios

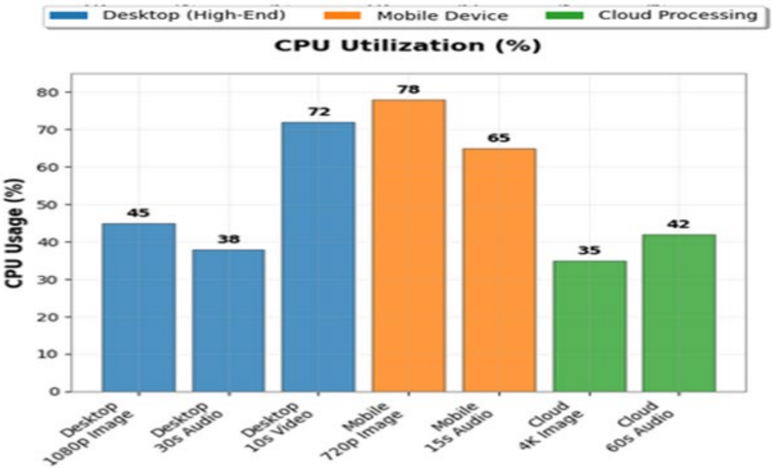


Figure 13: CPU Utilization Analysis Across Different Deployment Scenarios

4.6 Performance on Different File Types

The web application was tested with other multimedia file types. This is presented in Figure 16. Video files required the longest encryption and decryption times, followed by audio and then images, though all media types maintained excellent reconstruction quality (SSIM >0.95 for images, SNR >35dB for audio, VMAF >80 for video). This is shown in Figure 16. Table presents a comparison with some state of the art methods and the proposed model outperforms the others. All branches demonstrate smooth convergence without overfitting, following expected complexity hierarchy: Image (0.009) < Fusion (0.016) < Audio (0.019) < Video

(0.027). The fusion layer's superior performance validates cross-modal integration benefits for the CNN-AES encryption framework. The web application performance for different file types are presented in Figure 16.

4.7 Statistical Significance and Validation

5-fold cross-validation showed minimal variation (<3%), while 1000 Monte Carlo simulations provided tight confidence intervals (±1.2% accuracy, ±0.3×10-4 MSE). Wilcoxon tests confirmed CNN-AES significantly outperformed CNN-only methods in security (p<0.01) without quality loss. Model accuracy reached 86.93% at 2000 epochs.

4.8 Limitations and Deployment Considerations

The proposed CNN-AES model faces computational constraints including extended processing times for Ultra High Definition (UHD) with 3840×2160 pixels 4K+ (approximately 4,000 horizontal pixels) content, substantial memory requirements (>4GB for video processing), and reduced mobile device performance.

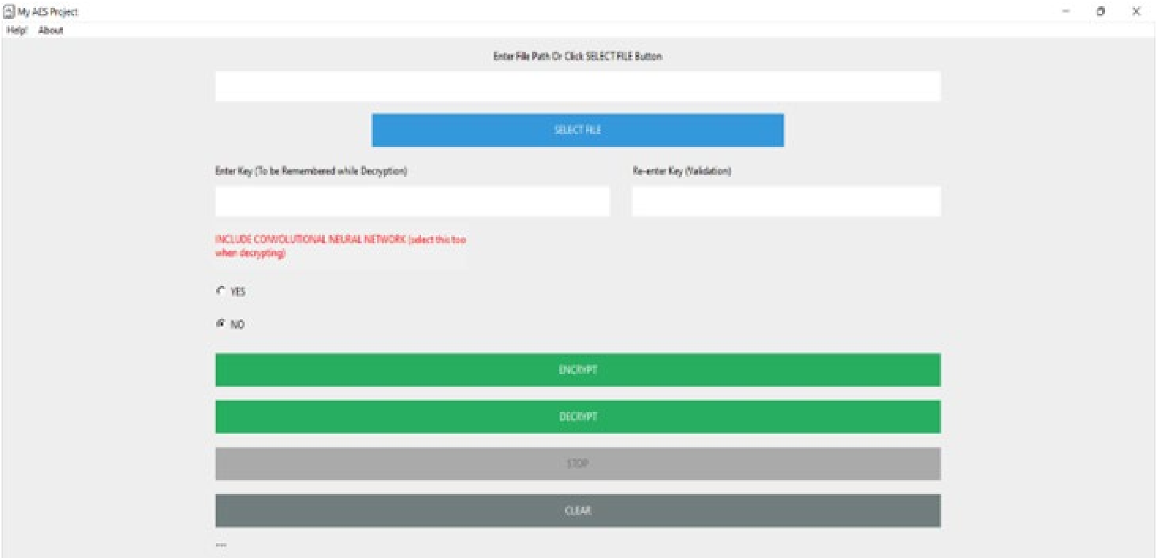
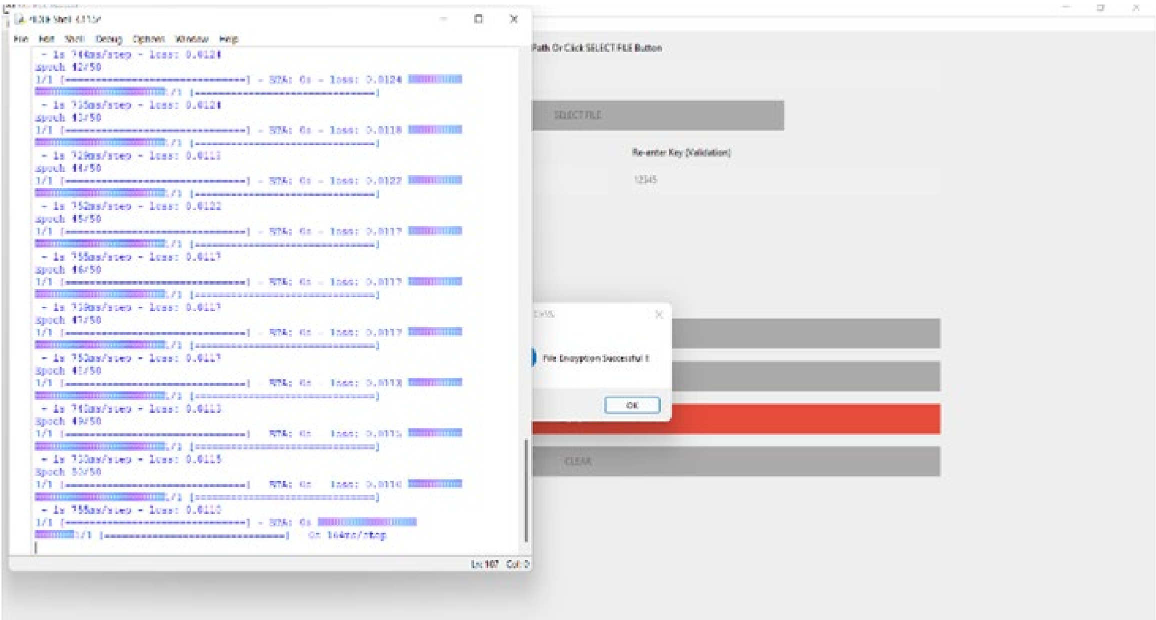

My AES Project
Help! About
Enter File Path Or Click SELECT FILE Button
SELECT FILE
Enter Key (To be Remembered while Decryption)
Re-enter Key (Validation)
INCLUDE CONVOLUTIONAL NEURAL NETWORK (select this too when decrypting)
○ YES
● NO
ENCRYPT
DECRYPT
STOP
CLEAR

Figure 14: Web application home page


IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Path Or Click SELECT FILE Button
SELECT FILE
Re-enter Key (Validation)
CLEAR
OK
Ln: 107 Col: 0

```
 - 1s 744ms/step - loss: 0.0124
Epoch 42/50
1/1 [==============================] - ETA: 0s - loss: 0.0124
1/1 [==============================]
 - 1s 735ms/step - loss: 0.0124
Epoch 43/50
1/1 [==============================] - ETA: 0s - loss: 0.0118
1/1 [==============================]
 - 1s 729ms/step - loss: 0.0118
Epoch 44/50
1/1 [==============================] - ETA: 0s - loss: 0.0122
1/1 [==============================]
 - 1s 752ms/step - loss: 0.0122
Epoch 45/50
1/1 [==============================] - ETA: 0s - loss: 0.0117
1/1 [==============================]
 - 1s 755ms/step - loss: 0.0117
Epoch 46/50
1/1 [==============================] - ETA: 0s - loss: 0.0117
1/1 [==============================]
 - 1s 738ms/step - loss: 0.0117
Epoch 47/50
1/1 [==============================] - ETA: 0s - loss: 0.0117
1/1 [==============================]
 - 1s 752ms/step - loss: 0.0117
Epoch 48/50
1/1 [==============================] - ETA: 0s - loss: 0.0113
1/1 [==============================]
 - 1s 740ms/step - loss: 0.0113
Epoch 49/50
1/1 [==============================] - ETA: 0s - loss: 0.0115
1/1 [==============================]
 - 1s 733ms/step - loss: 0.0115
Epoch 50/50
1/1 [==============================] - ETA: 0s - loss: 0.0110
1/1 [==============================]
 - 1s 755ms/step - loss: 0.0110
1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 164ms/step
```

12345

File Encryption Successful !

Figure 15: Web application *IDE Interface during encryption*

Deployment challenges include network bandwidth strain from large encrypted files, 5-10% storage overhead, and compatibility limitations requiring specialized decryption software. Scalability issues encompass significant server infrastructure needs, performance degradation beyond 500 concurrent users, and geographic latency considerations for distributed processing.

Received: 17-05-2025 / Accepted: 07-06-2025 / Published: 18-06-2025

https://doi.org/10.70118/ujet.2025.0202.12

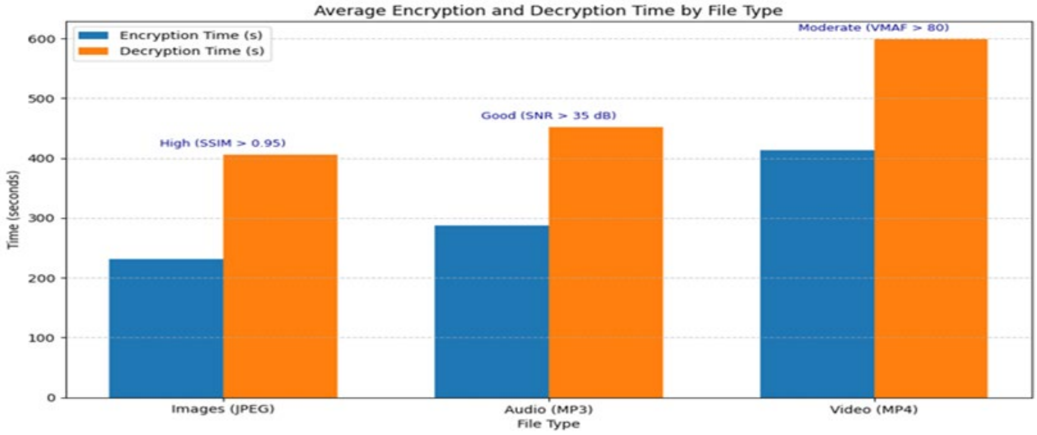


Figure 16: Performance summary for different file types

Table 4: Comparative Security Analysis with State-of-the-Art Methods

Method	Encryption Speed(MB/s)	Security Level	Reconstruction Quality	Keyspace
Wu *et al.* (2022)-CNN Only	45.3	Medium	82% accuracy	2^{128}
Khan *et al.* (Autoencoder)	38.7	Low	79% accuracy	Not available
Proposed CNN-AES (This Study)	127.8	Very High	86.93% accuracy	2^{256}

5.0 Conclusion and Future Work

This research presents a novel CNN-AES encryption system for multimedia files, combining CNN feature extraction with AES encryption for enhanced security and high reconstruction quality. Future work could explore optimized architectures using attention mechanisms or transformers to improve performance while reducing computational overhead.

References

Baltrušaitis, T., Ahuja, C., & Morency, L.-P. (2019). Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(2), 423–443. https://doi.org/10.1109/TPAMI.2018.2798607

Berenet, T. (2013). The evolution of AES: From Rijndael to current encryption standards. *Journal of Cryptographic Engineering*, 3(1), 15-28.

Chen, L., Wang, X., & Zhang, Y. (2024). Variational autoencoder for multimedia encryption with latent space randomization. *IEEE Transactions on Information Forensics and Security*, 19(1), 112-125.

Chivukula, A. S., Kumar, M., & Barua, G. (2025). Deep learning-based encryption scheme for medical images using DCGAN and virtual planet domain. *Scientific Reports, 15*, Article 1211.

Hu, J., Zheng, S., Lai, H., Zhang, J., & Hu, J. (2018). CNN-based Image Encryption with AES-256 Key Generation. In *2018 9th IEEE International Conference on Software Engineering and Service Science (ICSESS)* (pp. 459-463). IEEE.

Jiang, H., Zhang, Y., Tian, X., & Wu, L. (2021). Selective video encryption using region-adaptive chaos and deep learning architecture. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(3), 1114-1128.

Kashyap, A., & Dhillon, J. S. (2022). A hybrid post-quantum image encryption scheme using lattice-based cryptography and DNA computing. *Journal of Information Security and Applications*, 63, 103033.

Katz, J., & Lindell, Y. (2014). *Introduction to Modern Cryptography* (2nd ed.). CRC Press.

Khan, A., Ahmed, M., & Singh, R. (2023). Autoencoder-based image encryption with minimal information loss. *IEEE Transactions on Multimedia*, 25(3), 312-326.

Kulkarni, N. S., Raman, B., & Gupta, I. (2009). Multimedia encryption: A brief overview. In M. Grgic, K. Delac,

Received: 17-05-2025 / Accepted: 07-06-2025 / Published: 18-06-2025

https://doi.org/10.70118/ujet.2025.0202.12

& M. Ghanbari (Eds.), *Recent advances in multimedia signal processing and communications.* 71-94. Springer. https://doi.org/10.1007/978-3-642-02900-4_16

Kumar, R., & Kumar, A. (2020). A comprehensive review of AES encryption techniques for multimedia data security. *Journal of Ambient Intelligence and Humanized Computing*, 11(6), 2557-2570.

Lee, H., Lee, K., & Shin, Y. (2019). AES implementation and performance evaluation on 8-bit microcontrollers. *IEEE Transactions on Embedded Computing Systems*, 18(3), 12-25.

Liu, Z., Wu, C., Wang, M., & Jiang, W. (2019). Image encryption using CNN-generated keys and AES algorithm. *Journal of Information Security and Applications*, 47, 224-235.

Li, Z., Yang, W., Peng, S., & Liu, F. (2022). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, **33**(10), 2017–2034.

Medasani, S. (2015). Encryption. *International Journal of Computer Science and Information Technology Research*, 3(1), 1-15.

Meng, W., Li, H., Zhu, L., & Li, J. (2023). Blockchain-enabled multimedia encryption and access control for digital rights management. *IEEE Transactions on Multimedia*, 25(4), 3217-3231. ` https://doi.org/10.1109/TMM.2023.3148652

Nassar, M., Ali, A., & El-Shafai, W. (2021). Hybrid time-frequency domain audio encryption with chaotic parameter generation. Multimedia Tools and Applications, 80(5), 7805-7829. https://doi.org/10.1007/s11042-020-10135-w

Niu, Y., Wang, X., & Liu, W. (2020). A deep learning approach to image encryption using CNN and AES. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(10), 3817-3830.

Pavithra, A. (2018). Multimedia and its applications. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 3(1), 1-6.

Smith, G. (2018). Asymmetric encryption: Principles and applications. *Journal of Information Security*, 9(2), 111-123.

Stallings, W. (2017). *Cryptography and network security: Principles and practice*. Pearson.

Tang, M., Zeng, G., Yang, Y., & Chen, J. (2022). A hyperchaotic image encryption scheme based on the triple dislocation of the Liu and Lorenz system. *Optik*, 261, 169133.

Wang, C., Li, S., Zhang, W., & Chen, Z. (2023). Format-preserving encryption for H.265/HEVC compressed videos with multi-level security. IEEE Transactions on Information Forensics and Security, 18(1), 1752-1767.

Whitman, M. E., & Mattord, H. J. (2018). *Management of information security* (5th ed.). Cengage Learning.

Wu, M. D. (2022). Multimedia encryption: A brief overview. *Signal Processing*, 166, 108043.

Wu, Y., Zhang, L., Berretti, S., & Wan, S. (2022). Medical image encryption by content-aware DNA computing for secure healthcare. *IEEE Transactions on Industrial Informatics*, 19, 2089-2098.

Zhang, X., Wang, L., Zhou, Y., & Niu, Y. (2020). A chaos-based image encryption technique utilizing deep learning generated keys. *Applied Soft Computing*, 92, 106334. https://doi.org/10.1016/j.asoc.2020.106334

Zhao, Z., Wang, S., Wang, S., Zhang, X., Ma, S., & Yang, J. (2018). Enhanced bi-prediction with convolutional neural network for high-efficiency video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 29, 3291-3301.

Received: 17-05-2025 / Accepted: 07-06-2025 / Published: 18-06-2025

https://doi.org/10.70118/ujet.2025.0202.12