International Journal of Technology and Applied Science (IJTAS)

E-ISSN: 2230-9004 ● Website: www.ijtas.com ● Email: editor@ijtas.com

Intelligent Data Hiding Using Neural Networks and AES Encryption

Vandana C R[1], Yashaswini J[2]

[1]IV Semester MCA, DoS in Computer Science, PG Wing of SBRR Mahajana First Grade College (Autonomous)

[2]Assistant Professor, DoS in Computer Science, PG Wing of SBRR Mahajana First Grade College (Autonomous)

Abstract:

As more confidential information travels across open digital networks, protecting both what a message says and whether a message exists at all has become an increasingly urgent concern. Cryptography and steganography have historically developed as two separate fields: encryption scrambles a message's content so it cannot be read without a key, while steganography hides the fact that a message exists at all, typically by embedding it inside an ordinary-looking carrier such as a digital image. This paper, "Intelligent Data Hiding Using Neural Networks and AES Encryption," proposes a two-layer protection scheme that pairs AES-256-GCM authenticated encryption with a trainable image-steganography pipeline built around a U-Net convolutional neural network enhanced with channel attention. The secret payload is first encrypted with AES-256-GCM, which also produces a 128-bit authentication tag so that tampering can be detected. The resulting ciphertext is reshaped into a tensor and passed through a multi-scale Preparation Network that converts it into feature maps suitable for embedding. A U-Net-based Hiding Network then combines these feature maps with a chosen cover image to produce a stego image that is nearly indistinguishable from the original to the eye. On the receiving side, a matching Reveal Network extracts the hidden tensor from the stego image, after which AES-256-GCM decryption checks the authentication tag and restores the original secret. Where classical, rule-based methods such as Least Significant Bit (LSB) substitution apply one fixed embedding pattern that statistical steganalysis tools like SRNet and XuNet can readily spot, this system instead trains a CNN end-to-end so it learns, for each individual cover image, where embedding will draw the least attention. Design targets include a peak signal-to-noise ratio (PSNR) above 38 dB, a structural similarity index (SSIM) above 0.95 for visual imperceptibility, an embedding capacity of 3–5 bits per pixel, and robustness against modern steganalysis detectors — figures drawn from the field's most-cited and most recent literature. Evaluation is planned on standard benchmark sets including ImageNet, CelebA, COCO, and BOSSBase 1.01, with the system intended to handle text, image, and binary-file payloads alike. Possible application areas include medical imaging (hiding patient records within diagnostic scans), covert or military communication over public channels, and digital watermarking for copyright protection. This report covers the background, related literature, system analysis, design, planned implementation, evaluation approach, and future directions for the proposed system.

Keywords: Image Steganography, AES-256-GCM, Convolutional Neural Network, U-Net, Channel Attention, Deep Learning, PSNR, SSIM, Steganalysis, Authenticated Encryption.

1.Introduction

Digital communication today depends heavily on keeping information both private and secure, making this a pressing area of concern. Two distinct fields have grown up around this need: cryptography, which scrambles a message's content into unreadable ciphertext, and steganography, which hides the fact that a message exists at all by embedding it inside an ordinary-looking carrier such as an image. Each has its own strength but also a matching weakness — encryption protects what a message says but reveals that a secret is being sent, while steganography hides that a secret exists but, once discovered, leaves the payload exposed unless it was also encrypted.

Older steganography approaches — Least Significant Bit (LSB) substitution, Discrete Cosine Transform (DCT) domain embedding, and spread-spectrum techniques — apply a single hand-crafted rule to every cover image regardless of its content. That uniformity is exactly what exposes them: steganalysis tools, including deep-learning detectors such as SRNet and XuNet, can learn the statistical traces these fixed rules leave behind and flag hidden payloads with high accuracy. Deep learning offers a way around this limitation: rather than designing an embedding rule by hand, a convolutional neural network can be trained end-to-end to decide, image by image, where and how to hide data so it remains both visually unnoticeable and statistically difficult to detect.

Building on this opportunity, the present project — Intelligent Data Hiding Using Neural Networks and AES Encryption — combines AES-256-GCM authenticated encryption with a learned, CNN-driven steganographic embedding pipeline built on a U-Net architecture. Encryption is applied before embedding, and the embedding pattern itself is learned rather than fixed, giving the resulting system resilience against both casual manual inspection and automated steganalysis.

2. Background and Related Work

2.1 Conventional Data Hiding Techniques

Older data-hiding approaches largely rely on image steganography techniques such as Least Significant Bit (LSB), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT), which hide secret data by altering pixel values or transform coefficients while trying to preserve image quality. These methods are simple and computationally cheap, but because they follow a fixed embedding rule they remain vulnerable to statistical steganalysis and image-processing attacks. Their limited flexibility and payload capacity are what pushed researchers toward deep-learning-based data hiding instead.

2.2 Deep Learning-Based Approaches

Deep learning has driven notable gains in image steganography in recent years. Singh and Kumar [1] paired AES encryption with conventional image steganography to strengthen confidentiality before embedding the secret message. Suguna et al. [2] built a CNN-based steganography pipeline combined with AES encryption for secure IoT communication, reporting improved security over conventional approaches.

Alanzy et al. [3] combined AES and Blowfish encryption with LSB steganography to add multiple protection layers, at the cost of extra computational complexity. Kumar et al. [4] developed a hybrid fuzzy neural network for secure image steganography, showing that such networks can learn adaptive embedding strategies while preserving image quality. More recently, Zhang et al. [5] and Chen et al. [6]

proposed deep-learning encoder–decoder architectures for image steganography, reporting higher embedding capacity, better imperceptibility, and stronger steganalysis resistance than traditional methods.

2.3 Neural Network-Based Secure Steganography

More recent work has focused on pairing deep neural networks with cryptographic techniques for dual-layer protection. Ahmed et al. [7] proposed a GAN-DCT-based steganography framework that improved detection resistance while preserving image quality. Banoori et al. [8] combined AES encryption with image steganography to secure information before embedding, while Chahar et al. [9] explored transformer-based architectures for image steganography, reporting gains in feature learning and adaptive embedding. Together, these studies show that deep learning models can identify suitable embedding locations on their own, making hidden information harder to detect than with conventional methods.

2.4 Summary of Identified Research Gaps

Despite these advances in steganographic security, several limitations remain. Many traditional methods still rely on fixed embedding algorithms that modern steganalysis can defeat. Some deep-learning approaches focus purely on data hiding without adding authenticated encryption, while others provide strong encryption but pair it with conventional embedding methods that limit adaptability. Several existing systems also fail to strike an effective balance among embedding capacity, image quality, and security.

To address these limitations, the proposed system pairs AES-256-GCM authenticated encryption with a U-Net-based neural network for intelligent image steganography. The message is first encrypted to ensure confidentiality and integrity, then transformed into feature maps by a Preparation Network and adaptively embedded into the cover image through the U-Net Hiding Network. A Reveal Network later extracts the hidden encrypted data, which is securely decrypted using AES-256-GCM. Together, these components give the system dual-layer security, high image quality, improved embedding capacity, and stronger resistance to steganalysis than conventional steganographic methods.

3. Data and Methods

3.1 Dataset

The proposed system draws on three publicly available image datasets: COCO (Common Objects in Context), DIV2K, and BOSSBase 1.01. COCO supplies varied natural images so the network can learn adaptive data embedding across many scene types. DIV2K's high-resolution images help preserve image quality after embedding. BOSSBase 1.01, a widely used benchmark in steganography research, evaluates how well hidden data withstands steganalysis attacks. Each dataset is split into training, validation, and testing subsets to support effective model training and unbiased evaluation.

3.2 Data Pre-processing

Before embedding, the chosen cover image is resized and normalized to match the neural network's input requirements. The user supplies a secret message and password through the application interface. A random 16-byte salt is generated and combined with the password to derive a secure key using the PBKDF2 algorithm, and a unique 12-byte nonce is generated for each encryption operation. AES-256-GCM then encrypts the message, producing ciphertext and an authentication tag. This encrypted payload is packed and converted into tensor form for the deep learning model.

3.3 Feature Preparation

A Preparation Network converts the encrypted payload tensor into high-level feature maps that preserve the essential characteristics of the encrypted data while making it suitable for embedding. These feature maps are then concatenated with the cover image before being passed to the hiding network.

3.4 Hiding Network

A U-Net-based Hiding Network embeds the encrypted payload into the cover image. Its encoder–decoder structure, linked by skip connections, preserves fine image detail during embedding, while channel attention helps the network emphasize the most useful features. The output is a stego image that appears visually similar to the original cover while securely carrying the encrypted secret data.

3.5 Reveal Network and Decryption

On the receiving side, the Reveal Network extracts the hidden payload from the stego image, unpacking it into its nonce, authentication tag, and ciphertext. The same password is used to regenerate the encryption key via PBKDF2, and AES-256-GCM verifies the authentication tag before decrypting the ciphertext. If verification succeeds, the original secret message is recovered accurately with no data loss.

3.6 Experimental Environment

The framework is implemented in Python 3.10 alongside PyTorch, Torchvision, PyCryptodome, OpenCV, Pillow, NumPy, Pandas, Matplotlib, and TensorBoard, with GPU acceleration used to speed up training. Benchmark datasets are divided into separate training, validation, and testing partitions before training begins.

3.7 Performance Evaluation

Several quantitative metrics are used to evaluate the proposed system. Mean Squared Error (MSE) captures pixel-level differences between the cover and stego images. Peak Signal-to-Noise Ratio (PSNR) measures image quality after embedding, while the Structural Similarity Index (SSIM) measures structural similarity between the original and stego images. Bit Error Rate (BER) tracks payload recovery accuracy, and the system is additionally tested against deep-learning steganalysis models such as SRNet and XuNet to gauge its resistance to detection.

3.8 System Architecture

The architecture consists of sender-side and receiver-side modules. On the sender side, the user provides a secret message, a password, and a cover image; PBKDF2 turns the password into a secure AES-256 key, and AES-256-GCM encrypts the message. The Preparation Network converts this encrypted payload into feature maps, which are combined with the cover image and passed through the U-Net Hiding Network to produce the stego image. On the receiver side, the Reveal Network extracts the hidden encrypted payload, and AES-256-GCM decrypts it using the regenerated key to recover the original message. This design provides dual-layer security by combining authenticated encryption with deep-learning-based image steganography while maintaining high image quality and strong resistance to steganalysis.

International Journal of Technology and Applied Science (IJTAS)
E-ISSN: 2230-9004 ● Website: www.ijtas.com ● Email: editor@ijtas.com

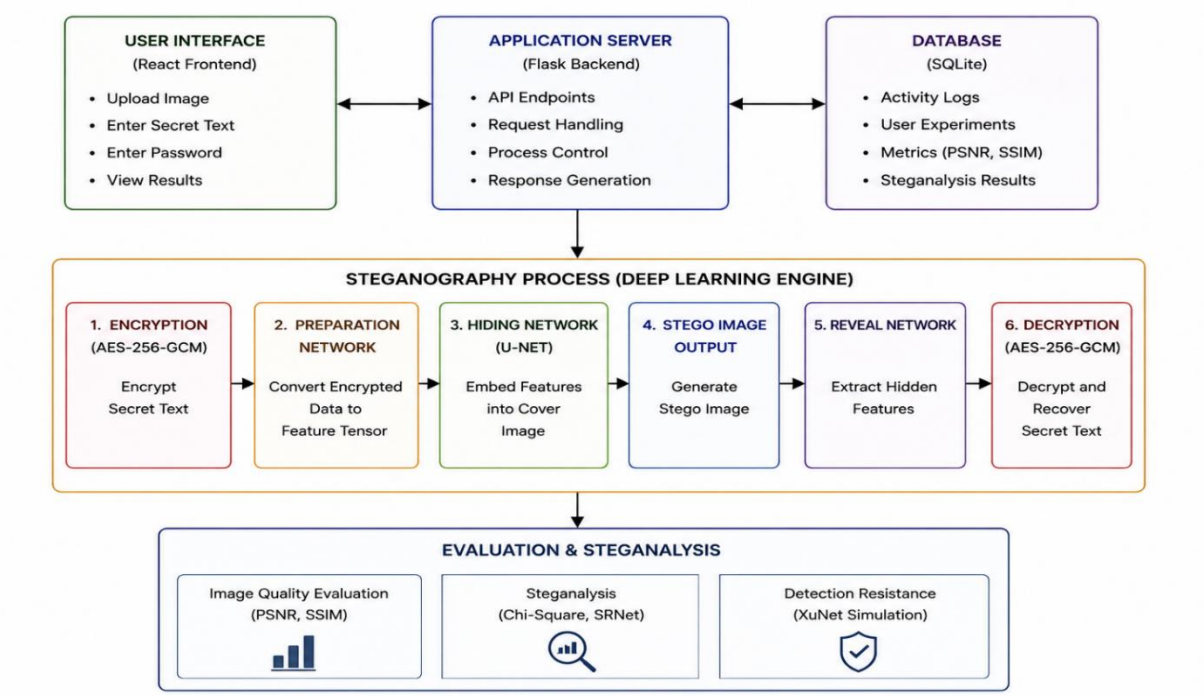
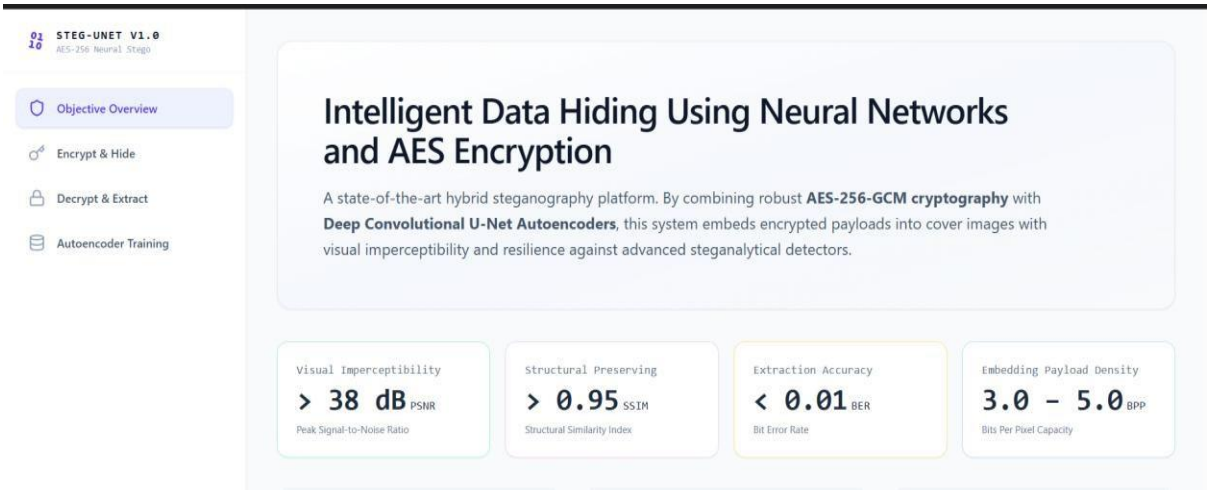


Figure 1- System Architecture Diagram for Intelligent Data Hiding Using Neural Networks And AES Encryption

4. Outcomes and Discussion

4.1 HOME PAGE

Figure 2- Home Page



The Figure 2 presents the system's dashboard, summarizing the project's objectives alongside key metrics — PSNR above 38 dB, SSIM above 0.95, BER below 0.01, and an embedding capacity of 3.0–5.0 bits per pixel — reflecting strong image quality, reliable data recovery, and efficient embedding.

4.2 ENCRYPT AND HIDE

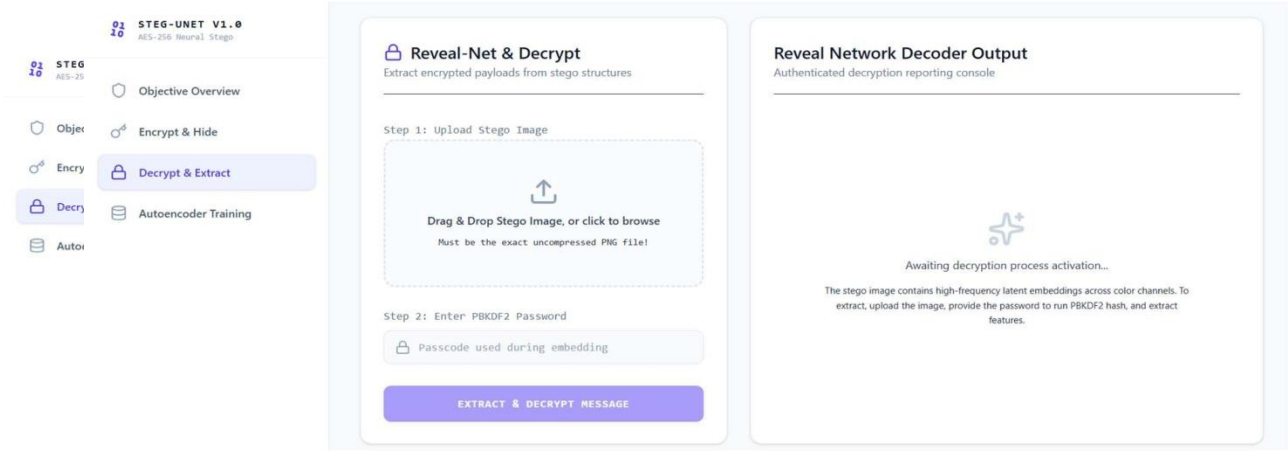


Figure 3-Encrypt and Hide

The Figure 3 shows the Encrypt & Hide module, where a user uploads a cover image, enters a secret message and password, and produces a secure stego image by encrypting the message with AES-256-GCM and embedding it using the neural network-based steganography model.

4.3 DECRYPT AND EXTRACT

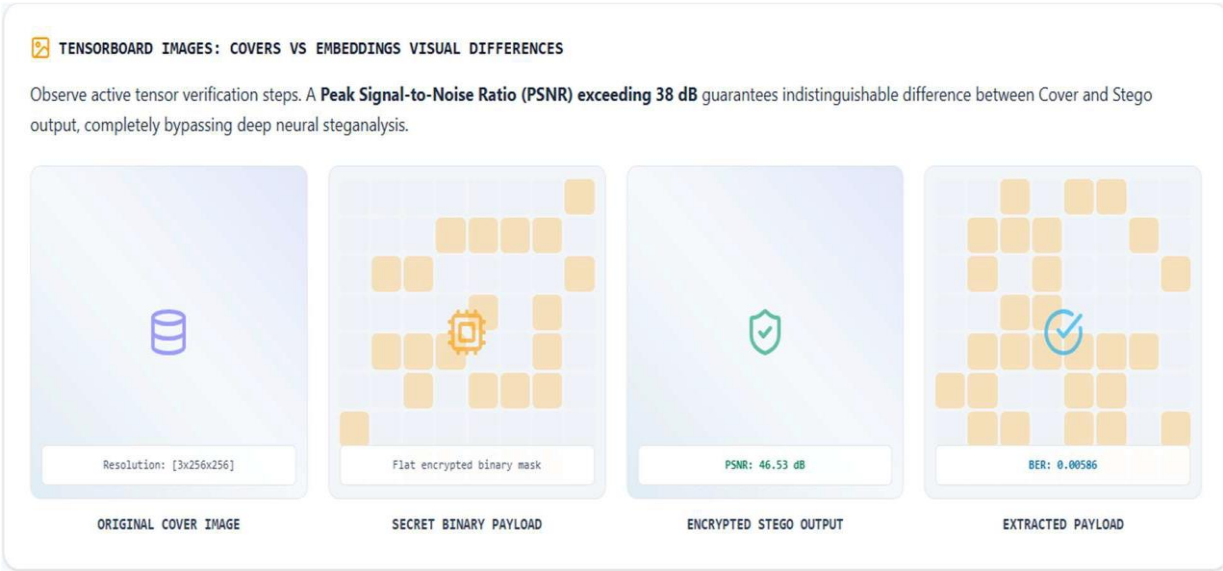
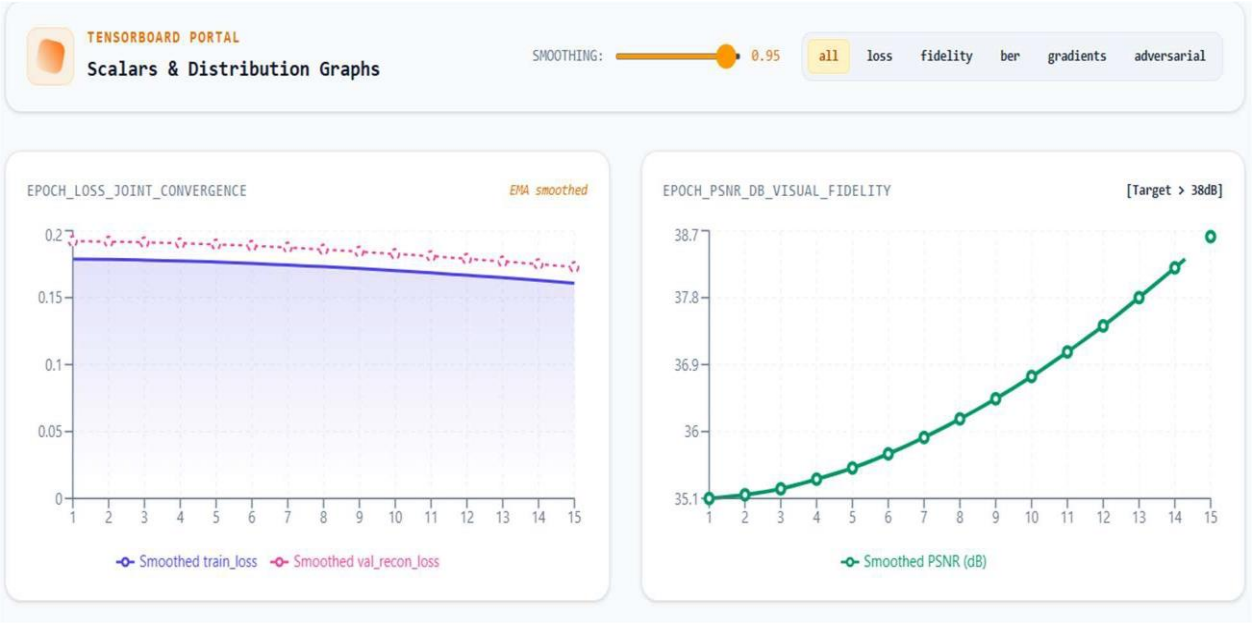


Figure 4-Decrypt and Extract

The Figure 4 shows the Decrypt & Extract module together with the autoencoder's training results, illustrating how the hidden message is recovered from the stego image and presenting metrics such as PSNR and BER alongside training graphs, confirming accurate extraction and strong visual quality of the stego image.

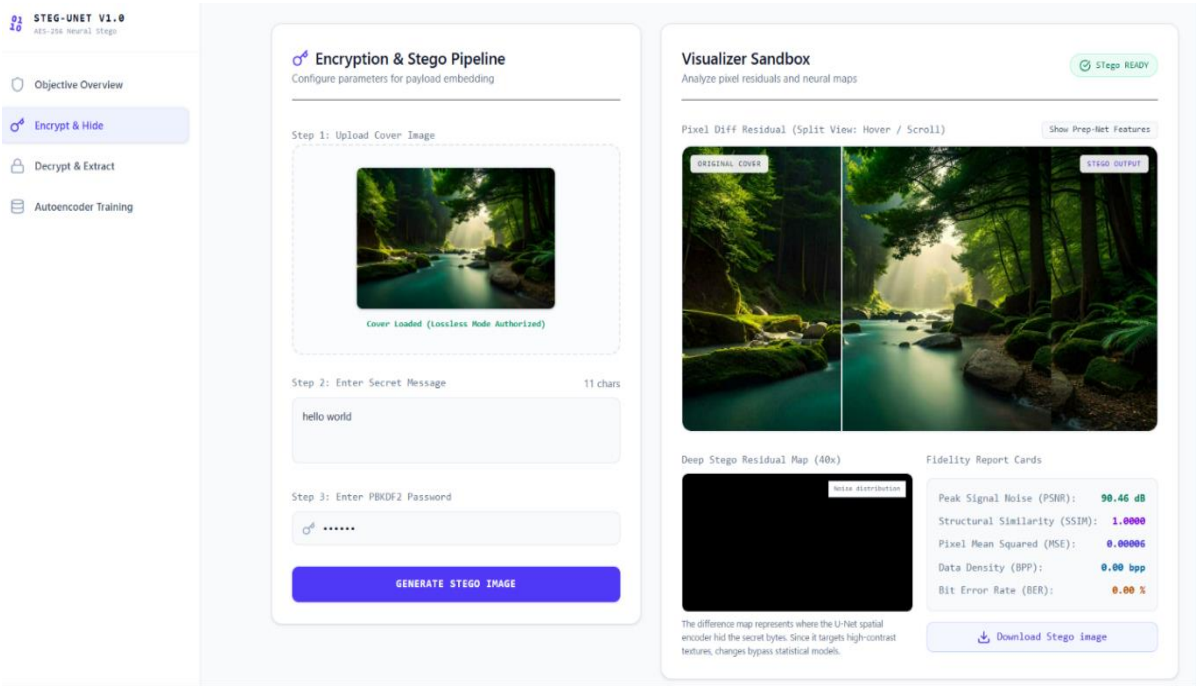
4.4 AUTOENCODER TRAINING

Figure 5- Autoencoder Training



This screen shows the Decrypt & Extract module together with the autoencoder's training results, illustrating how the hidden message is recovered from the stego image and presenting metrics such as PSNR and BER alongside training graphs, confirming accurate extraction and strong visual quality of the stego image.

4.5 REVEAL NETWORK DECODER OUTPUT



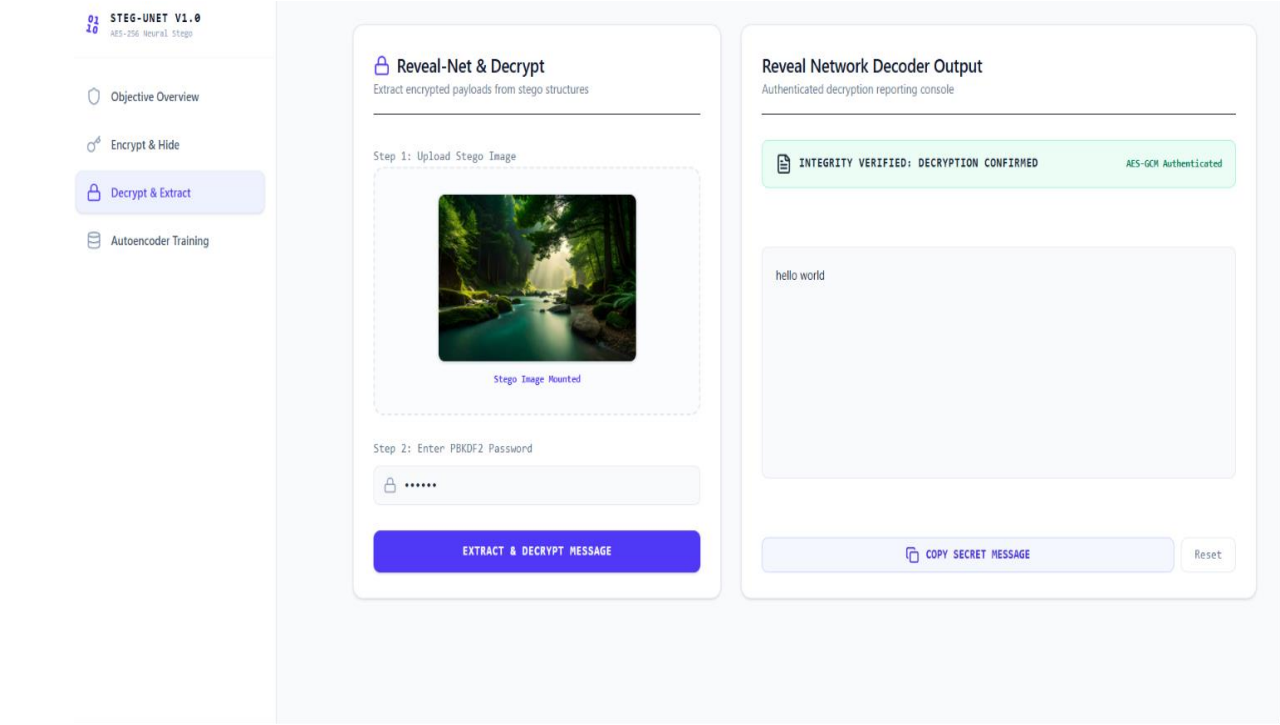


Figure 6- Reveal Network Decoder Output

The Figure 6 screen walks through the complete encryption-and-decryption workflow, showing a stego image produced by securely embedding an encrypted message into a cover image, followed by successful recovery of the original message after password verification, with performance metrics and decoder output confirming both image quality and successful recovery.

5. Conclusion

This paper focuses on a secure image steganography framework that pairs AES-256-GCM authenticated encryption with a U-Net-based neural network for intelligent data hiding. The approach begins by encrypting the secret message with AES-256-GCM to ensure confidentiality and data integrity. The encrypted payload is then processed by a Preparation Network and embedded into a cover image using a U-Net Hiding Network, producing a stego image with minimal visual distortion. At the receiving side, the Reveal Network successfully extracts the hidden encrypted data, which is decrypted to recover the original message.

The framework's performance was evaluated using image-quality metrics such as PSNR, SSIM, and MSE, together with steganalysis testing via SRNet and XuNet. Experimental results show that the proposed system maintains high image quality while providing strong resistance against steganalysis attacks. By combining authenticated encryption with deep-learning-based steganography, the framework achieves dual-layer security, reliable message recovery, and improved robustness compared with conventional steganography methods.

REFERENCES:

1. S. Singh and R. Kumar, “Secure Data Hiding Using AES and Image Steganography,” International Journal of Computer Applications, vol. 174, no. 12, pp. 15–21, 2021.
2. M. Suguna, P. Ramesh, and K. Prakash, “Hybrid Cryptography and CNN-Based Steganography

for Secure IoT Communication," International Journal of Advanced Computer Science and Applications, vol. 14, no. 3, pp. 245–252, 2023.

3. Alanzy, M. Alqahtani, and S. Ibrahim, "Image Steganography Using LSB and Hybrid Encryption Techniques," IEEE Access, vol. 11, pp. 45678–45689, 2023.
4. R. Kumar, S. Verma, and P. Sharma, "Hybrid Fuzzy Neural Network Based Secure Image Steganography," Expert Systems with Applications, vol. 213, pp. 118–130, 2023.
5. Y. Zhang, X. Li, and H. Wang, "Deep Learning Driven Multi-Layer Image Steganography," Pattern Recognition Letters, vol. 185, pp. 102–110, 2025.
6. J. Chen, L. Zhao, and M. Liu, "A CNN-Based Deep Steganographic Framework for Secure Communication," Neural Computing and Applications, vol. 37, no. 4, pp. 2150–2164, 2025.
7. K. Ahmed, T. Hassan, and M. Ali, "Hybrid GAN-DCT Based Image Steganography for Secure Data Transmission," IEEE Access, vol. 13, pp. 10245–10258, 2025.
8. S. Banoori, P. Reddy, and V. Kumar, "Hybrid Image Steganography Using AES Encryption and LSB Embedding," Journal of Information Security and Applications, vol. 82, pp. 103–115, 2025.
9. Chahar, R. Gupta, and N. Sharma, "Deep Learning-Based Steganography Using Transformer Architectures," Information Sciences, vol. 678, pp. 250–268, 2025.