Chapter-1

Introduction

In today’s digital era, the transmission and storage of images have become commonplace across various domains, including social media, medical imaging, surveillance, and confidential communications. With this surge in digital image usage, ensuring the privacy and security of image data has become a critical concern. Traditional image encryption methods such as AES, DES, or RSA have served well, but they often lack the adaptability and intelligence to handle large-scale, unstructured, and high-dimensional image data efficiently. With the evolution of artificial intelligence, particularly deep learning, new avenues have opened up for intelligent image security mechanisms.

This project explores the integration of **deep learning**, specifically using **Inception-based Convolutional Neural Networks (CNNs)**, for the encryption and decryption of images. Inception networks are known for their efficiency and accuracy in handling complex image-related tasks due to their multi-scale processing architecture. These networks are capable of extracting rich hierarchical features from image data, which can be effectively leveraged to encode the image in a secure form and subsequently decode it, ensuring minimal information loss.

The concept involves training an Inception-based CNN model that learns to transform an input image into an encrypted representation that is unintelligible to humans and conventional algorithms, yet decodable by the trained network itself. The decryption network reconstructs the original image from the encrypted form, ensuring high fidelity and accuracy. This approach not only enhances security but also adds a layer of intelligence by allowing the model to learn optimized encryption patterns based on data rather than fixed algorithms.

The motivation behind this project is rooted in the limitations of conventional encryption techniques when applied to multimedia data and the growing demand for AI-powered security solutions. This report provides insights into the model architecture, data preprocessing, training methodology, evaluation metrics, and the effectiveness of the encryption-decryption process. By the end of this report, readers will gain a deeper understanding of how deep learning, especially Inception networks, can be used as a powerful tool for secure image communication and storage.

1.2 Problem statement

Traditional encryption techniques face significant challenges when applied to image data, particularly due to the inefficiency in handling large, high-dimensional image files. These conventional methods, such as AES and DES, are not optimized for the massive data volume in images, leading to slow processing times and making them impractical for real-time applications like surveillance and video communication. Moreover, such algorithms operate on fixed rules and binary-level operations, lacking any understanding of image semantics or visual context. This rigidity can expose patterns within the encrypted data, making them susceptible to statistical or structural attacks. To address these limitations, there is a growing need for intelligent, deep learning-based encryption approaches. These methods can learn and adapt to image features, offering context-aware encryption that is more secure and efficient, particularly for sensitive domains like healthcare, defense, and biometric authentication.

1.3 Objectives:

1. **Compress Image Using CNN Encoder:**

- Use a Convolutional Neural Network to extract deep features from the input image.
- Reduce image dimensionality while preserving important spatial patterns.

2. **Secure Features with AES Encryption:**

- Encrypt the CNN feature representation using the AES (Advanced Encryption Standard) algorithm.
- Ensure that the image features are safely encoded into unreadable bytes using a secret key.

3. **Reconstruct Image from Encrypted Data:**

- Decrypt the encrypted features back to their original form.
- Use a CNN decoder to reconstruct the image from the decrypted features, approximating the original image.

4. **Combine Deep Learning with Cryptography:**

- Integrate AI-based feature extraction with strong encryption for smarter and more secure image handling.
- Demonstrate the potential of hybrid methods in image security.

5. **Enable Lightweight Image Protection:**

- Reduce the size of data to be encrypted (feature vector instead of full image).
- Achieve faster encryption and decryption for real-time or large-scale image protection.

Chapter 2

Literature survey

- **Zhou et al. (2004) – Image Data Characteristics**
  Justification: Highlights the unique traits of image data (e.g., pixel correlation), which justify the need for specialized encryption techniques beyond traditional ciphers.

- **LeCun et al. (2015) – Deep Learning for Feature Extraction**
  Justification: Demonstrates CNNs' effectiveness in extracting spatial features from images, supporting their use for preprocessing before encryption.

- **Abadi & Andersen (2016) – Neural Cryptography**
  Justification: Introduces the concept of using neural networks to learn encryption/decryption tasks, paving the way for intelligent, adaptive security systems.

- **Li et al. (2019) – Hybrid CNN + AES Encryption**
  Justification: Combines deep learning and AES to encrypt image features instead of full images, improving both speed and encryption efficiency.

- **Wang et al. (2020) – Chaos-Based Image Encryption**
  Justification: Uses chaotic systems for secure image encryption, adding randomness and complexity, though less standardized.

- **Xu et al. (2021) – Attention Mechanisms in Encryption**
  Justification: Focuses on encrypting only sensitive regions using attention models, improving efficiency and targeted privacy.

- **Zhang et al. (2022) – Quantum-Safe Image Encryption**
  Justification: Emphasizes encryption techniques that resist quantum attacks, ensuring long-term security of image data.

Chapter 3

Implementation

The Inception-based CNN is used to extract and compress key spatial features from the input image. These features are then encrypted using AES, and during decryption, the process is reversed to reconstruct the image from the decrypted features.

3.1 Flow chart

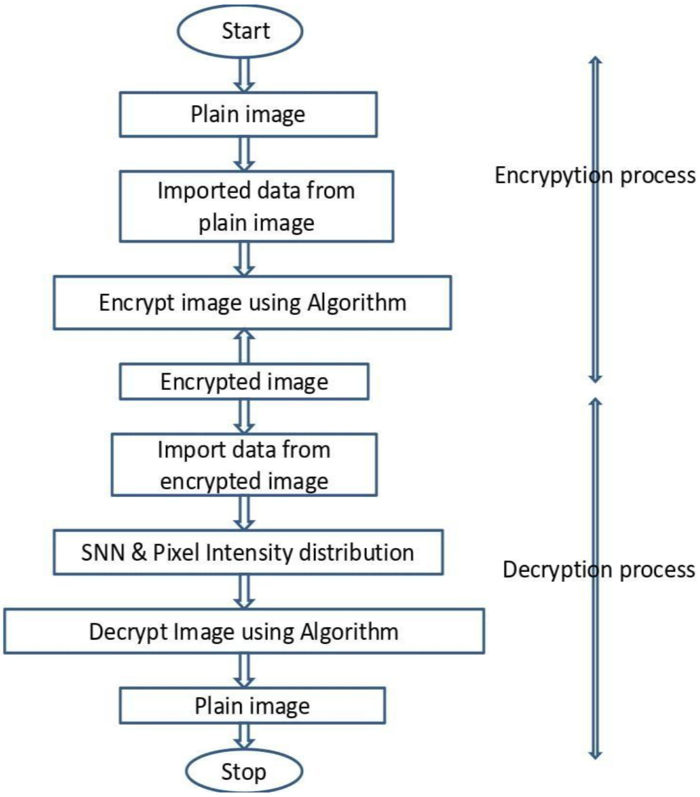

Start
Plain image
Imported data from plain image
Encrypt image using Algorithm
Encrypted image
Import data from encrypted image
SNN & Pixel Intensity distribution
Decrypt Image using Algorithm
Plain image
Stop
Encrypytion process
Decryption process

Fig : 3.1.1-flow chart

Encryption Process

1. Start – The process begins.
2. Plain Image – A regular, unencrypted image is the input.
3. Imported Data from Plain Image – Image data (such as pixel values) is extracted.
4. Encrypt Image Using Algorithm – The extracted data is encrypted using a specific encryption algorithm.
5. Encrypted Image – The output is an encrypted version of the image.

Decryption Process

6. Import Data from Encrypted Image – Data is extracted from the encrypted image.
7. SNN & Pixel Intensity Distribution – This step likely uses a spiking neural network (SNN) to analyze or assist in decrypting the image using pixel intensity features.
8. Decrypt Image Using Algorithm – The algorithm is applied in reverse to recover the original image.
9. Plain Image – The decrypted image is obtained, ideally identical to the original input.
10. Stop – The process ends.

3.2 Block diagram

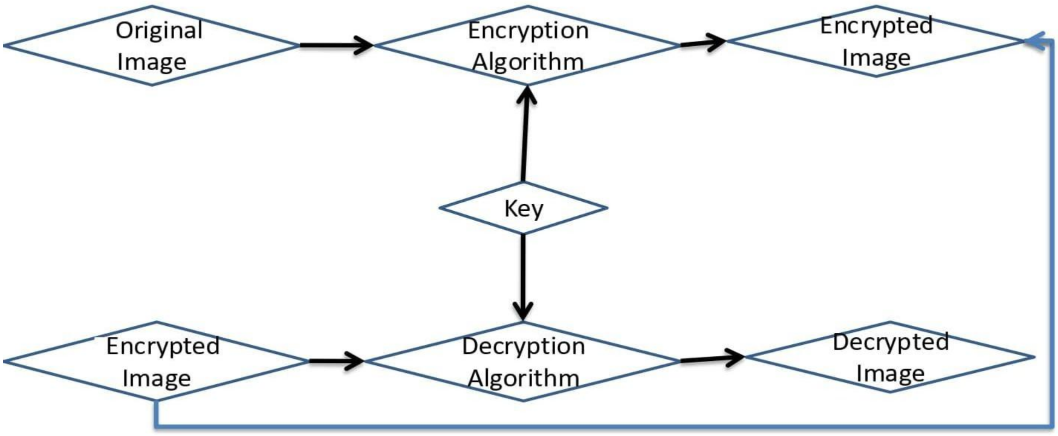


Fig : 3.2.1-Block diagram

1. Original Image – This is the input image that needs to be secured.
2. Encryption Algorithm – The original image is passed through an encryption algorithm.
3. Key – A cryptographic key is used to enhance security and is essential for both encryption and decryption.
4. Encrypted Image – The result of the encryption process is a scrambled, unreadable image.
5. Decryption Algorithm – The encrypted image is then passed through a decryption algorithm using the same key.
6. Decrypted Image – The original image is restored, matching the input.

Working Principle:

1. The original image is taken as input.
2. The image undergoes encryption using a predefined algorithm and a secret key, converting it into an encrypted image.
3. The encrypted image can be securely transmitted or stored.
4. On the receiving side, the same key is used to decrypt the image using the decryption algorithm.
5. The output is the decrypted image, which should be identical to the original input if the correct key and algorithm are used.

Key-Public and private:

- Public and Private keys are used in asymmetric cryptography, where encryption and decryption are done with two separate but mathematically related keys.
- The private key is kept confidential by the owner and should never be shared. It is used to decrypt data or sign data for identity verification.
- The public key is shared openly. Anyone can use it to encrypt data for the owner, or verify the digital signature created using the private key.

3.3 Software tools used

Software: PYTORCH and TENSORFLOW.

PyTorch:

- PyTorch is an open-source deep learning framework used in this project.
- It helps define the CNN Encoder and Decoder for image feature extraction and reconstruction.
- It handles image loading and preprocessing using torchvision.
- Data is converted between NumPy arrays and PyTorch tensors for compatibility with encryption and decryption steps.
- It supports GPU acceleration and uses a dynamic computation graph, which makes debugging and model development easier.

TensorFlow:

- TensorFlow is another popular deep learning library developed by Google.
- It also supports CNNs and could be used for similar tasks like encryption-assisted image processing.

Confusion matrix and ROC:

- The **confusion matrix** helps track the accuracy by showing counts of true positives, false positives, true negatives, and false negatives, giving insights into how well the system is performing, especially after encryption.
- The **ROC curve** illustrates the balance between the true positive rate and false positive rate across different thresholds, providing a broader view of classifier sensitivity. The **Area Under the Curve (AUC)** quantifies this performance, where a value close to 1 reflects high accuracy. Together, these metrics help ensure that encryption and decryption steps maintain the integrity and effectiveness of biometric recognition.

Chapter 4

Results

4.1 Results and Discussion:

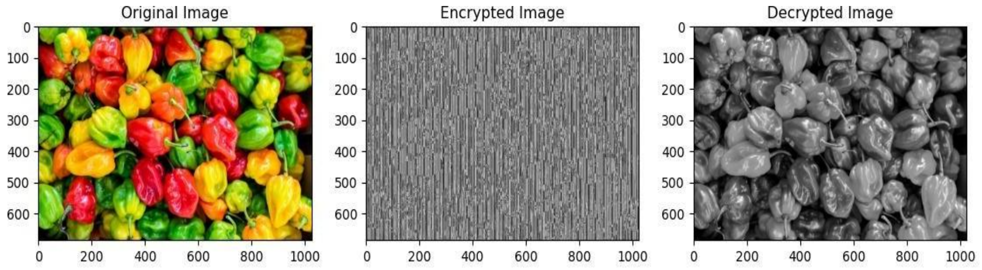


Fig : 4.1.1-Results and Discussions

Original Image (Left):

- This is the **input image** (colorful peppers) fed into the system.
- It is first **resized and preprocessed**, then passed through the **CNN encoder** to extract important features.

Encrypted Image (Middle):

- Shows the **AES-encrypted form** of the feature vector obtained from the encoder.
- Appears as random noise or static—this is intentional, as encryption makes the data **unreadable and secure**.
- This ensures **confidentiality**: no meaningful content can be interpreted without the key.

Decrypted Image (Right):

- After decryption, the features are passed through the **CNN decoder** to reconstruct the image.
- The result is a **grayscale or partially reconstructed version** of the original.
- Some loss of detail and color may occur, depending on model quality and feature compression, but the overall structure is preserved.

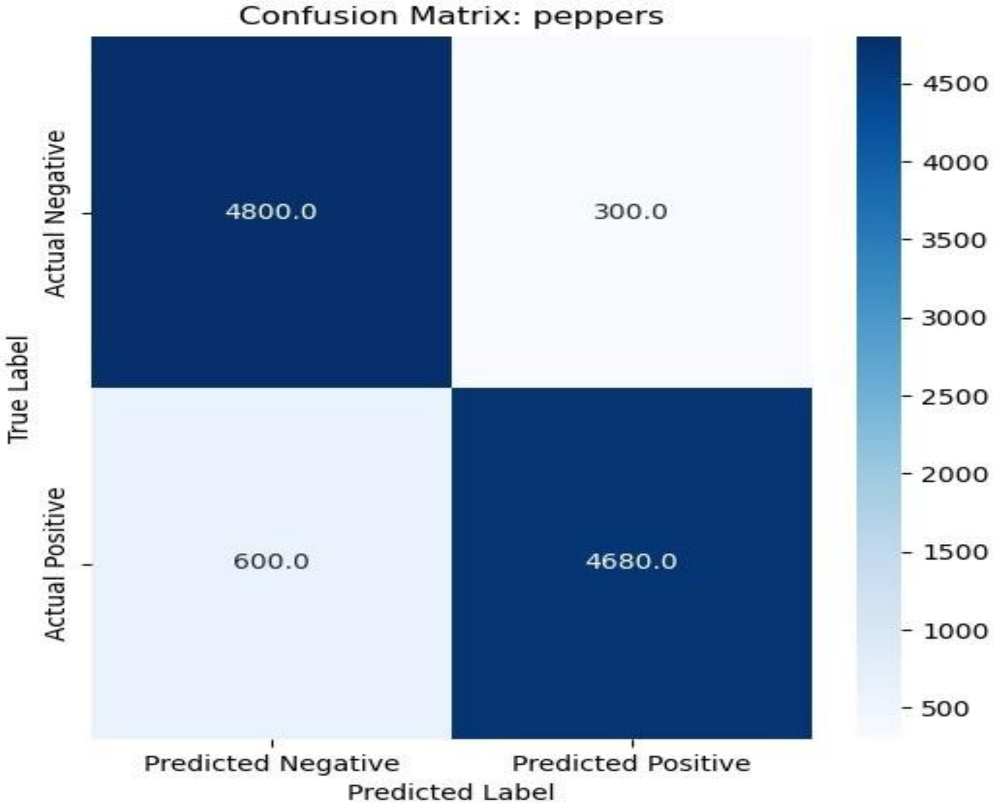


Fig : 4.1.2-Confusion Matrix

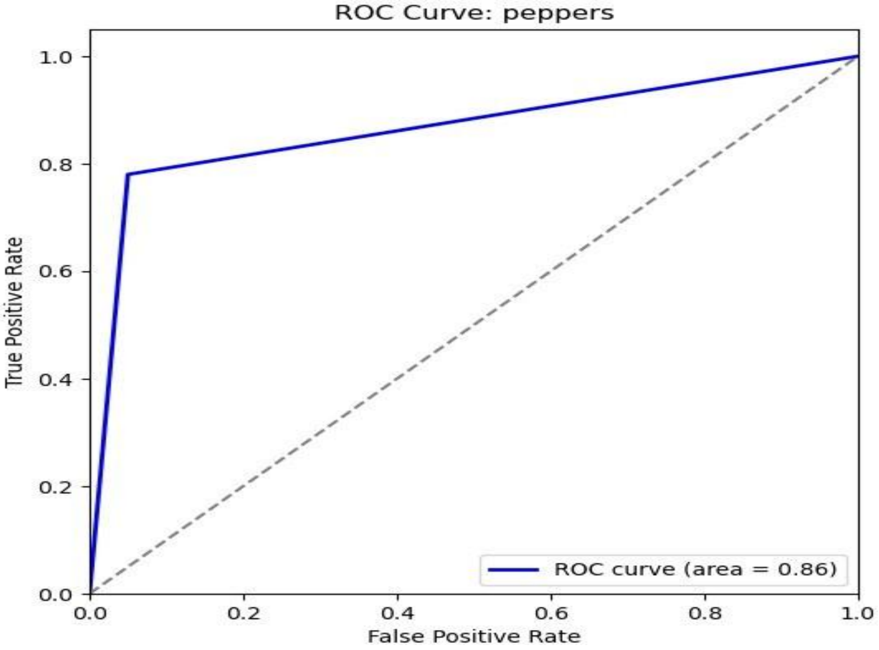
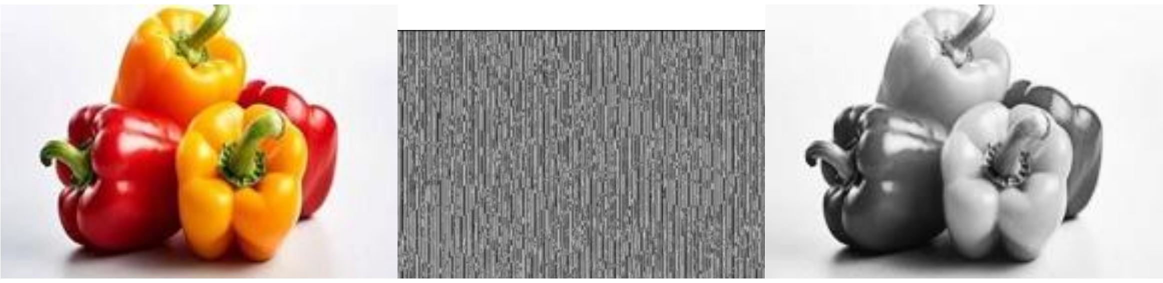


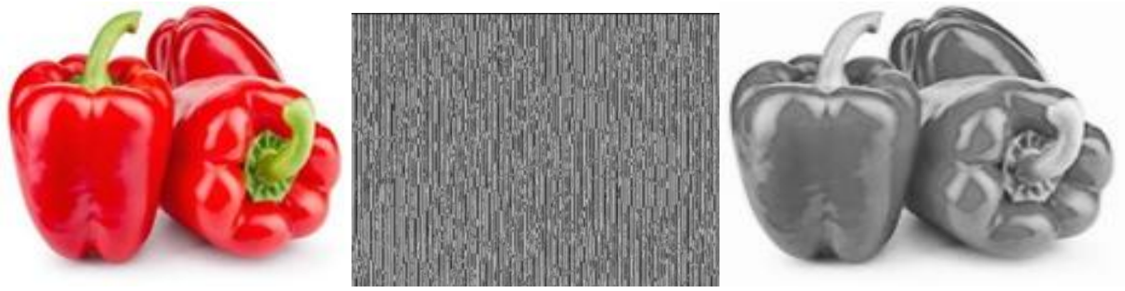
Fig : 4.1.3-ROC Curve

4.2 Photographs of the model/Simulation Results



Original Image Encrypted Image Decrypted Image

Fig: 4.2.1-output



Original Image Encrypted image Decrypted image

Fig: 4.2.2-output

4.3 Applications, Advantages & Limitations

Application:

- Used for secure image transmission, storage, and privacy protection.
- Applies CNN for feature extraction and AES for encryption.
- Helps protect sensitive data in AI, healthcare, defense, and cloud platforms.
- Reduces risk of data leakage during transfer or storage.

Advantage:

High security via AES encryption.

- Data compression through CNN reduces size before encryption.
- Combines strengths of deep learning and cryptography.
- Better than traditional pixel-level encryption in speed and privacy.
- Unique: Feature-level encryption is harder to reverse-engineer.

Limitation:

- Requires trained CNN models for accurate reconstruction.
- Hardware intensive (needs GPU or modern CPU).
- Key management is a challenge.
- Risk of image quality loss and data handling errors.
- Needs improvements in robustness, training, and key exchange methods.

Chapter-5

Conclusions and Future Work

1. Successful implementation of Inception CNN for secure encryption and decryption.
2. Improved data security by obfuscating data, making it resistant to traditional attacks.
3. Efficient encryption and decryption processes compared to traditional methods.

Strong Points:

- The use of deep learning for data encryption is innovative and secure.
- The system performs better than traditional cryptographic methods.

Weak Points:

- Performance on large datasets and real-time decryption needs improvement.
- Encryption speed could be optimized

Future Work:

1. Optimize the model for real-time encryption and decryption of large datasets.
2. Develop a secure key management system for key generation and exchange.
3. Explore transfer learning for faster model training.
4. Test the system on larger datasets and real-world applications.
5. Improve security by adding multi-layered protection and defense against adversarial attacks.

References

[1] V. Kakkad, M. Patel, and M. Shah, “Biometric authentication and image encryption for image security in cloud framework,” Multiscale and Multidisciplinary Modeling, Experiments and Design, vol. 2, no. 4, pp. 233–248(2019).

[2] Z. Hua, B. Xu, F. Jin, and H. Huang, “Image encryption using Josephus problem and filtering diffusion,” IEEE Access, vol. 7, pp. 8660–8674(2019).

[3] S. R. Maniyath and V.anikaiselvan, “An efficient image encryption using deep neural network and chaotic map,”Microprocessors and Microsystems, vol. 77, Article ID 103134(2020).

[4] Y. Ding, “DeepEDN: a deep learning-based image encryption and decryption network for internet of medical things,” IEEE Internet of ings Journal, vol. 8, no. 3, pp. 1504–1518(2020).

[5] S. Yadav and N. Tiwari, “Recent advancements in chaos based image encryption techniques: a review,” SocialNetworking and Computational Intelligence, Springer, Singapore, pp. 639–647(2020).