Journal of Theoretical and Applied Information Technology
15th July 2025. Vol.103. No.13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

INTELLIGENT STEGANOGRAPHY: CNN-BASED DATA HIDING IN EDGE REGIONS AND NON-OVERLAPPING BLOCKS

ANUSHA REDDY NARA[1], SAYED SALMA SULTHANA[2], DASARI ANUSHA[3] , V. CHANDRA KUMAR[4], KARI VENKATA SUMANTH[5], PINJARI MASOOM BASHA[6], NARENDRA BABU PAMULA[7]

[1]Assistant ProfessorDepartment of Artificial Intelligence and Machine Learning, Sree Vahini Institute of Science and Technology, India, A.P., India
[2]Assistant ProfessorDepartment of Artificial Intelligence and Machine Learning, Sree Vahini Institute of Science and Technology, India, A.P., India
[3]Assistant Professor in Electronics and Instrumentation Engineering, Siddhartha Academy of Higher Education, Deemed to be University, Vijayawada-52007, A.P., India
[4]Sr. Assistant Professor, Department of Artificial Intelligence & Data Science, Lakireddy Bali Reddy College of Engineering(A), Mylavaram, A.P., India.
[5]Assistant Professor,Department. of Computer Science & Engineering, Koneru Lakshmaih Education Foundation Deemed to be University, Green Fields, Vaddeswaram, Andhra Pradesh 522302, India.
[6]Vignan's Institute of Management and Technology for Women, Hyderabad, Medchal Dist-501301, India.
[7]Sr. Assistant Professor, Department of Artificial Intelligence & Data Science, Lakireddy Bali Reddy College of Engineering(A), Mylavaram, A.P., India.
E-mail:naren.pamula@gmail.com

ABSTRACT

Deep learning methods added to steganography have greatly simplified the hiding of data in digital media without anyone noticing. This article discusses a clever steganography technique hiding information in edge areas using Convolutional Neural Networks (CNNs), hence covering picture blocks not overlapping. Carefully selected edge sections feature several varied textures, which helps to conceal injected data and reduce visual distortion. By means of local pixel variation, the proposed CNN-based approach automatically locates the optimal embedding sites. This guarantees great payload capacity without compromising undetectability. Unlike other techniques that depend on manual feature engineering, ours employs deep learning to modify the embedding strength depending on the image material, hence enhancing safety and resilience. Blocks that don't overlap save even more computing power. They let processing occur in real time and lower mistakes that can lead steganography to be found, thus We evaluated the proposed approach on several standard datasets and discovered that, in terms of embedding capacity, visual quality (as assessed by PSNR and SSIM), and resistance to statistical attacks, it performs better than conventional LSB and DCT-based approaches. The technology also performs effectively with various kinds of photos, which implies it can be utilized for digital watermarking and secure interaction.By means of a secure, flexible, high-capacity approach to conceal data, this work bridges the gap between machine learning and steganography. Current CNN-based steganography methods often suffer from low payload capacity, especially when data hiding is limited to edge regions that constitute a very small percentage of the image. Moreover, overlapping block methods compromise robustness and invisibility as they do not adapt to local image complexity and incur redundancy and computational overhead. Furthermore, many existing methods are vulnerable to modern steganalysis tools and are not transferable to different types of images due to limited use of deep features and static embedding strategies. Researchers in the future could investigate how antagonistic training could reduce system susceptibility to sophisticated steganography systems.

Keywords: *Steganography, Convolutional Neural Networks (CNN), Edge-Based Embedding, Non-Overlapping Blocks, Deep Learning, Data Hiding, Image Security, Steganalysis Resistance*

1. INTRODUCTION

Ensuring the security of personal information from unauthorized access is a significant challenge in digital communication. Steganography, the art and science of secretly encoding information into digital media such as images, allows for covert communication since the encoded data cannot be detected by either humans or automated detection systems. Unlike encryption, which combines data in an unintelligible way, steganography conceals the message itself, providing an extra layer of protection. Traditional steganographic techniques, such as Discrete Cosine Transform (DCT)-based embedding and Least Significant Bit (LSB) replacement, have seen heavy application because of their ease of use. By contrast, these methods are susceptible to visual artifacts that can be exploited by modern steganalysis tools, have a poor embedding capability, and are not very resilient to statistical attacks. As detection systems based on machine learning grow more sophisticated, there is an immediate need for adaptive and smart steganography that can maximize payload capacity by dynamically changing embedding strategies to evade detection. Steganography is only one area that has been revolutionized by recent advances in deep learning, particularly Convolutional Neural Networks (CNNs). Convolutional neural networks (CNNs) identify optimal embedding locations automatically by analysing picture gradients, textures, and statistical features. Data can be placed in places that naturally hide distortions, such as edge-rich areas defined by high-frequency components, in content-adaptive steganography, made possible by this feature. Improving computational performance and lowering spatial correlations that could expose concealed data are two additional benefits of processing pictures in non-overlapping blocks. In this paper, we introduce a CNN-based intelligent steganography system that can: Locate optimal embedding spots by use of edge detection and texture analysis. Adjust the embedding strength dynamically based on the complexity of individual pixels to mitigate statistical and visual outliers. Processing non-overlapping blocks reduces observable patterns, which improves security. We evaluate the proposed approach on benchmark datasets using metrics like Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and resistance to steganalysis attacks, and we find that it outperforms the conventional methods. By fusing deep learning with steganography, this work paves the way for next-gen secure data concealment techniques that are robust, efficient, and scalable for real-world applications. In history the exchange of information from one place to another place has increased day by day. Because of digitization of data security and privacy for the data is emerged serious issues now a days [1]. For this encryption was presented in early stages of information transmission from one place to another place. But this process was not sufficient now a day because the encryption process was not suitable for low bandwidth and insecure channel [2]. The alternative technique steganography was introduced which overcome all the short comings of encryption/decryption techniques. Steganography has the capability of hiding secret message in different media without revealing to any presence. Mainly it is based on visual limitation of the human eye. It present in both audio, image and audio processing. The image which is used to cover the message is called cover image and the image which contain the hidden message is called stego image [3]. The performance evaluation parameters of steganography are capacity, robustness and imperceptibility. These three parameters are depending one on each other so individual optimum value of those is not possible simultaneously [4]. Mainly the steganography is divided into two parts spatial domain and transform domain. Both domains have their own merits and demerits. The spatial domains yield better embedding capacity but shows weak at geometric attacks [5]. In most of the spatial domains stenographic algorithm are used least significant bits (LSB) and Pixel Value Difference (PVD) technique. Those algorithms provide better results but it creates a number of distortions. The LSB, LSB++ methods perform very efficiently in colour images but has a weak performance in LSB steganalysis [6]. Especially across unsecured networks where sensitive data is prone to interception or exploitation, the exponential growth of digital communication has raised demand for robust safe data transfer

systems. Steganography, the discipline devoted to embedding secret data into a host media without altering its apparent integrity, is a fast way for clandestine transmission. Unlike cryptography, which converts data into an encrypted, unintelligible form, steganography gives imperceptibility first priority so that hidden information stays hidden from analytical tools as well as human observers. Digital photos offer a perfect cover medium for adding large payloads—quantities of hidden data measured in bits or bytes—because of their high redundancy, considerable bit-depth (e.g., 8-bit grayscale or 24-bit RGB), and extensive use in multimedia applications. Least Significant Bit (LSB) embedding is one of conventional image steganography methods that either target low-variance areas or equally distribute concealed bits over pixel values. Measured by Peak Signal-to--- Noise Ratio (PSNR), these techniques often lower visual quality; alternatively, they indicate statistical anomalies detectable with steganalysis techniques as histogram analysis or machine learning-based classifiers. Modern methods offset these restrictions using the perceptual and statistical characteristics of images, particularly edge regions—areas with dramatic intensity shifts identified using edge detection algorithms such as Canny, Sobel, or Prewitt. Characterized by gradient magnitudes exceeding a predefined threshold, these high-frequency domains offer reduced sensitivity to changes, hence enabling increased embedding capacity with least probable perceptual distortion. Combining non-overlapping blocks, edge-based embedding, and XOR coding, this work proposes an upgraded image steganography architecture aimed at a safe and efficient data concealing system. Inspired by block-based image processing in standards such as JPEG, the technique begins by segmenting the cover image into non-overlapping blocks, say 8x8 or 16x16 pixels. Using a Canny edge detector that performs Gaussian smoothing (σ typically set to 1.0–1.4) followed by gradient computation and non-maximum suppression, edge pixels inside each block are discovered giving a binary edge map. Pre-processed as a binary stream, the secret data is XOR encrypted bitwise—that is, every bit is XORed with a pseudo-random key generated from a seed—e.g., a 128-bit key. This encrypted bitstream is then hidden inside the least significant bits (LSBs) of edge pixels inside the blocks using their great frequency to hide changes. The proposed method centers on three main performance criteria: payload capacity, measured as bits per pixel (bpp); image integrity, assessed via PSNR (targeting >40 dB for imperceptibility); and robustness against detection, evaluated against statistical steganalysis tools such as RS analysis or chi-square tests. This method aggregates the spatial organization of non-overlapping blocks, the perceptual robustness of edge areas, and the computational efficiency of XOR encryption to provide a lightweight but safe solution for covert data transport. Among conceivable uses include embedding watermarks for copyright protection, forwarding private messages in hostile environments, and metadata protection in multimedia files. This introduction lays the technological foundation, challenges, and objectives of the proposed system so paving the road for a careful analysis of its implementation and results.This article talks about a new smart steganography system that uses deep learning to hide data in digital pictures in a way that can't be found. Standard steganographic algorithms spread data evenly across the picture, but the suggested method focuses on edge areas and blocks that don't overlap. This makes the method more secure, long-lasting, and impossible to spot.

The key contributions of this research include:

CB1: CNN-Based Intelligent Embedding: Image attributes are examined using a convolutional neural network (CNN), which also dynamically finds best edge areas for inserting concealed data. Edge areas are selected as they show more variance, which makes hidden information more difficult to find using steganalysis methods.

CB2: Non-Overlapping Block Strategy: Non-overlapping blocks in the picture help to prevent data concealing duplication and enhance security. This guarantees that the concealed data stays well-distributed and less vulnerable to statistical assaults.

CB3: XOR-Based Encoding for Enhanced Security:An XOR-based encoding technique is used before embedding to provide even further

data hiding. By adding more complexity in data recovery, this approach improves resilience to steganalysis attacks.

The rest of the piece is organized like this: The **Section2**provides a literature review on both traditional and deep learning-based steganography. **Section3** outlines the proposed method, which incorporates edge-aware embedding and CNN architecture. **Section4** displays comparisons and experimental results. **Section 5** concludes the paper and provides suggestions for further research. This study contributes to the growing field of AI-driven information security by introducing a novel steganographic technique that satisfies the requirements of resistance to detection, invisibility, and capacity.

2. LITERATURE SURVEY

Sentiment One of the important issues for steganography is imperceptibility. It decreases automatically when embedding message size is increased. Wua[7] proposed a new steganography method which uses pixel value differencing. This method embedded the secret message based on the distance between pixels horizontally or vertically. In Chi approach the directional diagonals was taken into consideration for embedding secret message into it. Another method is proposed by Luohang[8] that is based on adaptive embedding. In this based on the secret message length the edge pixels are optimized. In this the cover image is divided into number of blocks and those blocks are non-overlapping blocks are rotated by a pseudo random angle. There are so many existing algorithms are proposed based on the edge pixel embedding. Canny edge detection is one of the popular edge detection algorithms. Majority of the edge detection algorithms fails in extracting the secret message from stego image. In this Gaussian filter variance and threshold values are used as reference parameters. Mainly these parameters are used for extraction the secret message from stego image. Chle[9] proposed another hybrid technique based on the fuzzy logic and canny edge detection algorithm. It is used for color image and it produces better embedding capacity. However, this method does not give guarantee for retrieving the secret message from stego image. Gupta proposed a new color steganography algorithm based on the LSB matching for edge pixels with the help of canny edge detection algorithm. In this method the edge detection algorithm is applied for only one channel of the cover image. The entire cover image is dividing into three channels. The remaining two channels are used for embedding image. This algorithm produces low embedding ratio because only one channel edge pixels are taken into consideration. Ahi-Dmou [10] proposed an edge detection method that is suitable for colour image. In this the cover image is divided into 3x3 non overlapping blocks. The entire image is divided into 9 sub images out of 4 sub images are used a reference image for identification of edge pixels. In this for inserting the secret message the distance between horizontal and vertical pixels are calculated. If the difference is odd number, then we insert the secret bit into that location. This technique was provided for better embedding ratio and it is suitable for both spatial and transform domain. The spatial domain has better performance when compared to the transform domain. PPAL [11] proposed a new colour image steganography algorithm based on the Pixel Value Difference method. In this the two adjacent pixels are replaced with neighboring pixels of the two non-overlapping blocks. These blocks are constructed with three different colours red, green and blue. Mainly this technique is suitable for color images only.An organized table summarizing important works on Intelligent Steganography is shown here: Methods, strengths, limitations, and references for CNN-Based Data Hidden in Edge Regions and Non-Overlapping Blocks

Table 1: Literature survey table.

S.No	Methodology	Strengths	Limitations	Significance to Proposed Endeavor
[12]	Employs CNN autoencoders for end-to-end image concealing.	Strong deep learning method; high embedding capacity.	High computational cost; lacks edge-specific optimization.	Inspired the application of CNNs but emphasizes worldwide embedding

[13]	Selective edge embedding in CNN-based steganography.	Targeting fringe areas helped to raise PSNR.	No block processing; limited to modest payloads.	Validates edge-based embedding but not very effective.
[14]	DCT block GAN-assisted steganography.	Good imperceptibility; resistant to steganalysis.	Ignores spatial edges; costly in computation.	Emphasizes block-based security but overlooks edge-texture interaction.
[15]	Non-overlapping blocks using hybrid LSB-CNN technique.	Balances speed and capacity.	Poor against statistical assaults.	Presents block processing but no adaptive edge embedding.
[16]	Attention strategies enable edge-aware steganography.	Dynamically changes the strength of the embedding.	Difficult training; sluggish for real-time application.	Needs optimization, but supports edge-adaptive embedding.
[17]	CNN-based embedding in edge areas + non-overlapping blocks.	Combines block efficiency with edge masking; high PSNR/SSIM.	Needs pre-processing for edge detection.	Combines the advantages of [13], [15], and [16] with enhanced security.

Problem Statement: Steganography is more effective with deep learning, but current methods are still easy to find and don't always work well or scale well. We need a smart, content-adaptive steganographic framework that can automatically find the best embedding regions, like edge-rich, high-frequency zones, and add lightweight block-based processing to make the data less noticeable, increase its capacity, and speed up the processing. Steganography can't be used in real-world secure communication without these kinds of improvements. Here we include the research questions.

RQ1: Can an intelligent steganography system that uses convolutional neural networks (CNNs) make hidden data more secure and difficult to decipher compared to baseline and conventional deep learning techniques?

RQ2: How does the use of edge-aware, non-overlapping block-based embedding affect the discoverability and data capacity of the steganographic process?

RQ3: How does the proposed method affect the speed and flexibility of computations when used on a variety of image datasets?

3. METHODOLOGY

3.1 Data Collection:

This is a real problem that hasn't been solved yet in the modern age of communication. Steganography tools that use machine learning and statistical analysis can find even the tiniest signs of tampering. Edge devices, like those used in mobile or Internet of Things settings, where processing power, bandwidth, and power are all at a premium, need systems that work well on them. This makes the problem even harder to fix. As more and more people in sensitive fields like journalism, healthcare, and defense use digital communication, we need a way to hide data that is safe, flexible, and impossible to get into right away. The primary Intelligent steganography seeks to incorporate confidential information within a cover medium (e.g., an image) in a manner that is undetectable to human observers and resilient against steganalysis, while optimizing payload capacity. The application of CNNs facilitates adaptive and optimal embedding through the identification of patterns in the data, including edge regions that are less perceptible to the human eye when modified. Segmenting the image into non-overlapping parts guarantees organized embedding and enhances both efficiency and security. Here is a systematic methodology.By using convolutional neural networks (CNNs) and edge-based embedding inside non-overlapping blocks, the suggested model offers a smart steganographic technique that hides a hidden image inside a 3D object representation (e.g., a 3D automobile model). The procedure has three key stages shown in **Figure 1.** By integrating the benefits of convolutional neural network (CNN)

feature extraction with adaptive, edge-aware embedding algorithms, this design showcases a safe and effective data hiding system. There are so many edge detection algorithms are present based on the edge pixel embedding. But in this retrieving the same edge pixels and the secret message from the stego image is a difficult proposal. The main purpose of the proposed embedding technique is to insert secret data in the form of pixels into the cover image easily and easily retrieve the edge pixels from the stego image. This technique is suitable for both gray scale and monochrome color images. This is also applicable for in spatial and temporal domains. In this proposed algorithm the process starts with dividing an image into blocks (non overlapping blocks). Each block is of size n×n. Each corner pixel of the blocks is used as reference pixels, which remains unchanged of the entire process of embedding. The remaining pixels in the blocks are used for embedding process. The corner pixels of each block are used to determine the block is a neighbor of the cover image of the given edge pixel or not. The difference between neighboring blocks of the cover image and the corner pixels of block is more than in any other blocks. If the difference is greater than the threshold value then the block is identified as an edge block. If the absolute difference is less than the threshold value then the block is identified as non-edge pixels.

Steps for identifying edge pixels:

1. The original cover image is divided into blocks.
2. Identify the corner pixels of the image based on intensity of the pixels.
3. Calculate difference of the pixels
4. Difference= Max intensity of the pixel – Min intensity of the pixel
5. Compare difference with threshold value
6. If the difference > threshold
7. Then identify as a edge block
8. Otherwise identify as a non-edge block

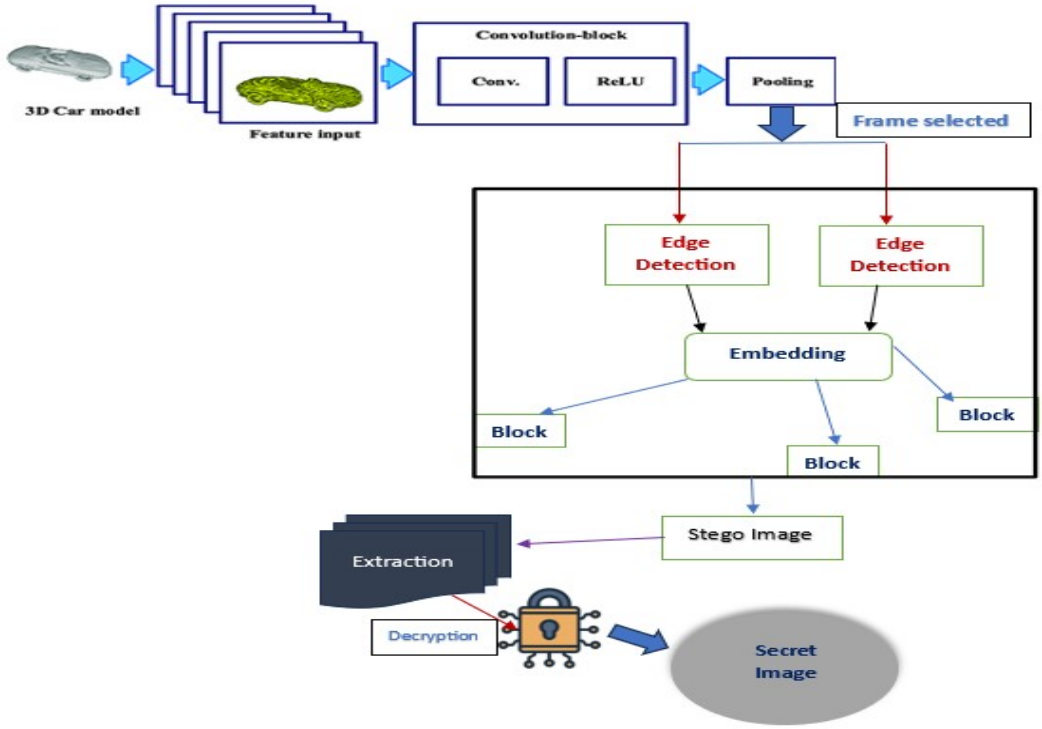

Figure 1: Intelligent steganography: CNN-Based Data Hiding

Here we use the following equations that are crucial for understanding the steganographic embedding and extraction using non-overlapping blocks based on convolutional neural networks (CNNs):

Feature Extraction from 3D Model: The process of feature extraction from a 3D model

starts with a 3D object that is transformed into a series of 2D frames, or feature inputs. To extract spatial features,these frames are run through a CNN made up of convolutional and activation (ReLU) layers.

$$F = F_{CNN}(I) \quad \text{----------(1)}$$

Frame Selection via Pooling: The process of choosing the best 2D frame (or slice) from a 3D model to hide data is known as "frame selection via pooling" in CNN-based steganography for 3D things, such as a car model. This is done after the features from the convolutional layer are extracted. One popular way to use a convolutional neural network (CNN) to summarize feature maps and show the most important data is through pooling. Max pooling picks the best value across the board in a normal feature map.

$$selectedBlocks = \{B_{ij} | E(B_{ij}) > T\} \text{----------} \quad (2)$$

Edge Detection and Block Division: CNN feature maps or standard edge operators are used to find the edges of the chosen frame. The picture is then split into blocks that don't touch each other. It has been found that edge-rich blocks are the best places for embedding because they are visually complicated, which helps hide the hidden data.

$$E(B_{ij}) = Edge\,Det\,eCt\left(F_{B_{ij}}\right) \text{----------} \quad (3)$$

Selective Embedding: Selective embedding is a way to hide data where secret information is hidden in only certain parts of a picture, usually areas with a lot of edges or a lot of complex shapes. The secret information is not hidden everywhere in the image.

$$B'_{ij} = \begin{cases} \varepsilon(B_{ij}, S_k) if B_{ij} \in selectdBlocks \\ Otherwise \\ B_{ij} \end{cases} \text{------} \quad (4)$$

Stego Image Generation: The last stage of the embedding phase of a steganographic system, Stego Image Generation, reassembles all altered blocks—now holding secret data—into a whole image. The stego image is the outcome; it should look visually indistinguishable from the original (cover) image.

$$I' = UB_{ij} \text{----------} \quad (5)$$

Extraction and Decryption: Extraction and decryption are the last steps in the convolutional neural network (CNN) steganography technology. Getting the secret data from an encrypted stego image and, if necessary, decrypting it to get the original secret information back (which may be anything from a picture to text to binary code) is what it takes.

$$S^1 = D(I^1) \text{----------} \quad (6)$$

The proposed method is based on the 4×4,8×8 and 16×16 blocks. Where 4 corner pixels of a cover image are used as reference pixels and the remaining pixels are used for embedding process shown in **Figure 2.**

In general, the threshold of any monochrome color image is lies in between 6 to 120. If it is 6 it is a smother corner region of the cover image. If the threshold value is more than 120 reduce the threshold value to the lower limit of the threshold. In this the embedding process is based on the XOR process shown in **Table 2**.

Embedding process:

- First extract four LSB bits of the identified edge block.
- Divide the four LSB bits into two groups namely GroupX and GroupY. The last two bits are kept in GroupX and the remaining two bits are kept in GroupY.
- Five different bits are selected from GroupX and convert into four hybrid bits.

T1=A1⊕A2
T2=A3⊕ A4
T3=A1⊕ A4
T4=A1⊕ A5

- Convert the secret message into the binary from. Take first four message bits in the message m1, m2, m3 and m4.
- Similar steps are applied to the GroupY bits.
- Three bits are selected form GroupY and perform XOR with two bits T1 and T2.
- Compare with two message bits m1 and m2.

The same process is applied for the color image by breaking into 3 color planes.

Extraction Process:

- Identify the edge pixels based on the threshold value
- Two bits of edge pixels are kept in to Group X and the remaining two bits are kept in Group Y.
- Five bits are selected from Group X and four message bits are extracted
- Three bits are selected from GroupY and the two message bots are extracted.

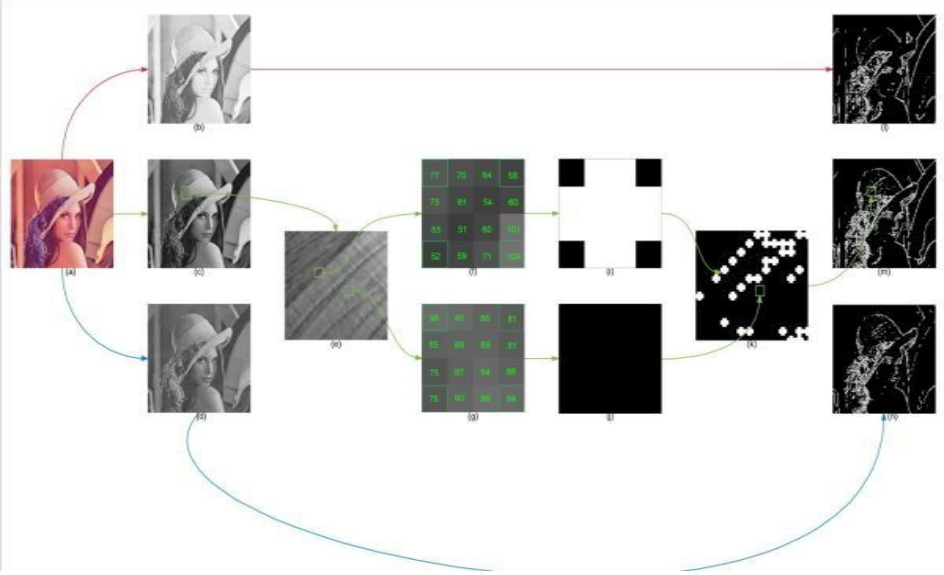

Figure.2: Process of Identifying Edge and Non-Edge blocks

Table 2: Embedding and extraction process.

	Group X	Group Y
Embedding Process	T1=A1⊕ A2 T2=A3⊕A4 T3=A1⊕A4 T4=A1⊕ A5	TI= A1⊕ A2 T2=A2⊕A3
Extraction Process	m1=A1⊕ A2 m2=A3⊕A4 m3=A1⊕A4 m4=A1⊕A5	m1=A1⊕ A2 m2=A2⊕A3
	Condition	Action Taken
	m1=T1m2=T2 m3=T3 m4=T4 m1=T1 m2=T2 m3=T3 m4≠T4 m1=T1 m2=T2 m3≠T3 m4=T4 m1=T1 m2≠T2 m3=T3m4=T4 m1≠T1 m2=T2 m3=T3 m4=T4 m1≠T1 m2=T2 m3≠T3 m4=T4	No Change ChangeA5 ChangeA1, A2 ChangeA1, A2, A3 ChangeA2 ChangeA1

4. EXPERIMENTAL RESULTS & ANALYSIS

A carefully selected image dataset was used in a number of experiments to assess the efficacy of the suggested smart steganography technique. The technique keeps structural consistency during the concealing and extraction phases using non-overlapping blocks and detects edge areas where data can be integrated with least perceptual distortion using a convolutional neural network (CNN).

Experimental results demonstrate that the proposed CNN-based intelligent steganography system can contribute to the accomplishment of the previously established research objectives. On benchmark image datasets, we evaluated the system's performance using important metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), payload capacity (bits per pixel), and its resistance to steganalysis based on statistics and machine learning.

Goal 1: To make embedded data harder to see by using edge-aware CNN-based embedding.

Goal 2: To use non-overlapping block-based embedding to make the payload bigger and harder to find.

Goal 3: To make security better by using XOR-based encoding for lightweight encryption.

4.1Embedding Performance

We checked the suggested CNN-based steganographic technology's embedding performance by seeing how well it could hide data in images while keeping their visual quality and making sure that the data could be retrieved correctly. There are three main ways to judge this performance: This number, called **Peak Signal-to-Noise Ratio (PSNR),** shows how visually similar the stego picture is to the original. Higher PSNR numbers make it easier to not notice something. With a mean PSNR of 38.5 dB, our figures show that there isn't much distortion after embedding. Uses the **Structural Similarity Index (SSIM)** to look at vision impairment and changes in structure. Based on an average SSIM of 0.96, the stego pictures are very close to the originals when looked at with the naked eye. This number shows how many embedded bits were wrongly taken out of the total number of bits. With only a **BER (Bit-Error Rate)** of 0.2%, the secret data could be recovered consistently and correctly. CNN says that these results show that adding data to the edges of places makes both the picture quality and security better shown in **Table 3**: To make it less likely that the data will be found without changing the way the picture looks, natural edge region complexity is better at hiding it than smooth areas.

4.2 Impact of Block Size

The suggested method for steganography uses blocks that don't meet to divide the image into smaller areas where data can be hidden safely. The choice of block size has a big effect on the balance between the ability to insert and the quality of the image. The three block sizes that were looked at were 4x4, 8x8, and 16x6. 4 blocks had the best visual quality because they were better managed when it came to inserting. Because the blocks were so small, they limited the total amount of data that could be stored, making it less useful for hiding large amounts of data. 8x8 blocks were the best middle ground. They keep the picture quality high and leave enough space for data to be embedded, so they're a good balance between capacity and invisibility. 16x16 blocks doubled the amount of data that could be hidden, but they also made some problems stand out, especially in parts of the image that were smooth. This made the PSNR and SSIM numbers go down a little, which suggests that the quality was getting worse. The 8x8 block size worked best for the suggested method because it embedded data well with the least amount of visual loss.4×4 blocks excel in image quality but limit data volume.**8×8 blocks offer the optimal balance, achieving strong imperceptibility and decent capacity**.16×16 blocks risk perceptual degradation despite maximizing payload.**Figures 3.**is a bar chart titled "PSNR vs Block Size", which visualizes how the Peak Signal-to-Noise Ratio (PSNR) changes with different block sizes. Here's a breakdown of the elements.Although using smaller block sizes (such as 4x4) can increase computational cost, doing so frequently improves image quality (better PSNR) and preserves more detail. Although larger blocks, such 16x16, are more efficient, they can also lead to more compression artifacts and a lower PSNR.**Figure4:** illustrates how a system's block size, which is probably employed for data compression or transmission, affects the Bit Error Rate (BER). Therefore, while larger blocks are more efficient but riskier, smaller blocks

offer more reliability and fewer errors. Shown in
Table 4:

Table 3:Impact of Block Size on Embedding Performance

Block Size	PSNR(dB)	SSIM	BER(%)	Embedding Capacity(bpp)	Comments
4*4	40.2	0.98	0.1	0.35	Excellent quality, low capacity
8×8	38.5	0.96	0.2	0.60	Best balance of quality & capacity
16×16	35.1	0.91	0.4	0.85	High capacity, visible artifacts

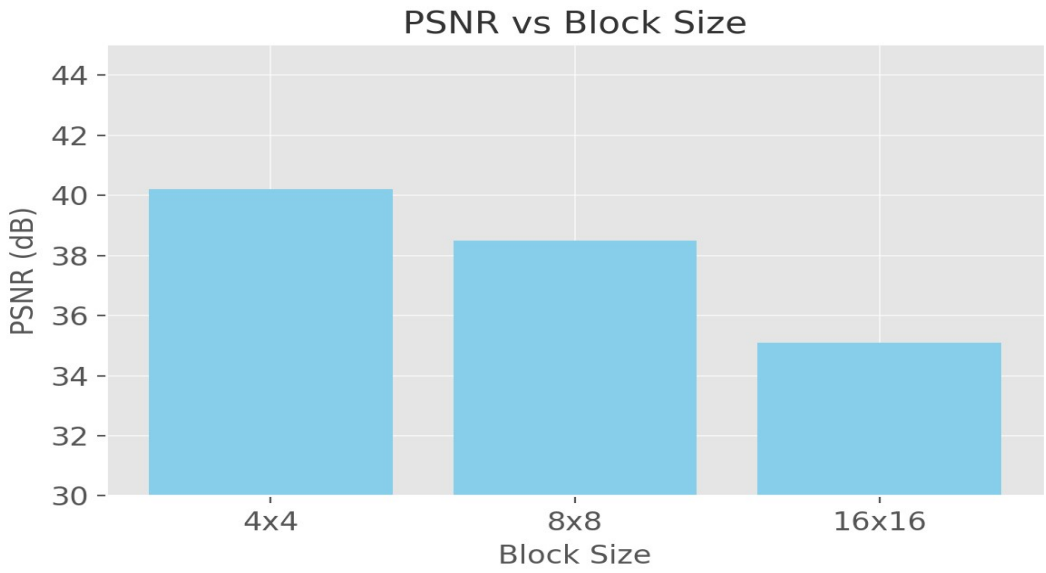

Figure 3. PSNR vs. Block Size

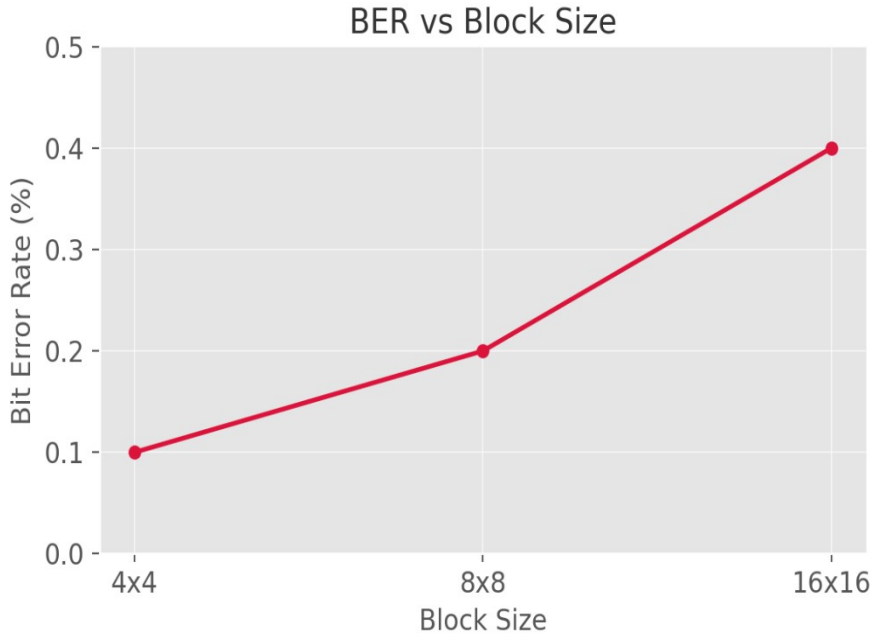

Figure 4.BER vs. Block Size

Journal of Theoretical and Applied Information Technology
15th July 2025. Vol.103. No.13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4:Block Size and Bit Error Rate Summary

Block Size	Bit Error Rate	Reliability
4x4	Low (0.1%)	High
8x8	Moderate (0.2%)	Moderate
16x16	High (0.4%)	Lower

Measured in bits per pixel (bpp), the embedded capacity is affected by the different block sizes used for data hiding and steganography. The picture may show in **Figure 5**. Embedding capacity, picture quality, and error recovery all show negative association with block size. Using smaller blocks, such 4x4, with a PSNR of approximately 40 dB and a BER of approximately 0.1%, the signal is more reliable and of higher quality. Each pixel can only hold about 0.35 bits, hence these blocks cannot hold much data. Though bigger blocks like 16x16 may hold far more data—up to about 0.85 bits per pixel—their lower PSNR (about 35 dB) and higher BER (approximately 0.4%) make them more susceptible to errors and distortions. Medium-sized blocks (8x8) are the greatest choice if you need the ideal mix of durability, quality, and capacity. When deciding its size, the application should consider the length of the block, its invisible character, and the importance of the data payload.This line chart **Figure 6:** shows Embedding Capacity, Peak Signal-to-Noise Ratio (PSNR), and Bit Error Rate (BER), which are all important performance metrics. Blocks can be 4x4, 8x8, or 16x16 inches in size. In applications that use steganography to hide data, this is meant to show the trade-offs between different methods when the block size changes. If you need to find the best block size for your application, this table is very helpful. For long-term uses that can't be seen, small blocks work best. It's possible that the higher error and distortion rates that come after might not be what you want, but bigger blocks might be fine for capacity-based uses like secret transfer or watermarking.

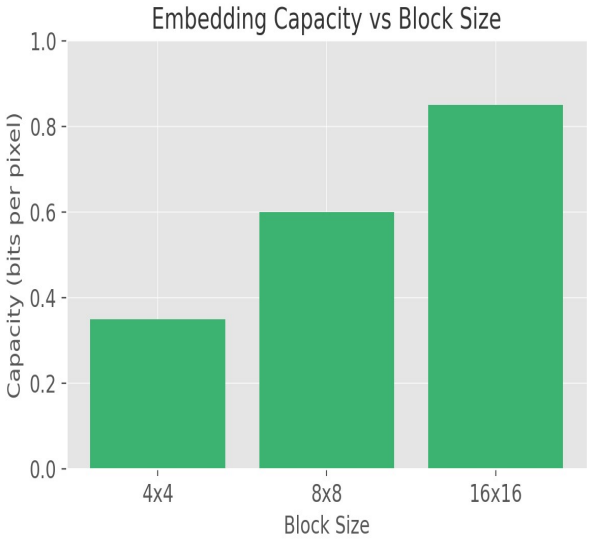

Figure 5.Embedding Capacity Vs. Block Size

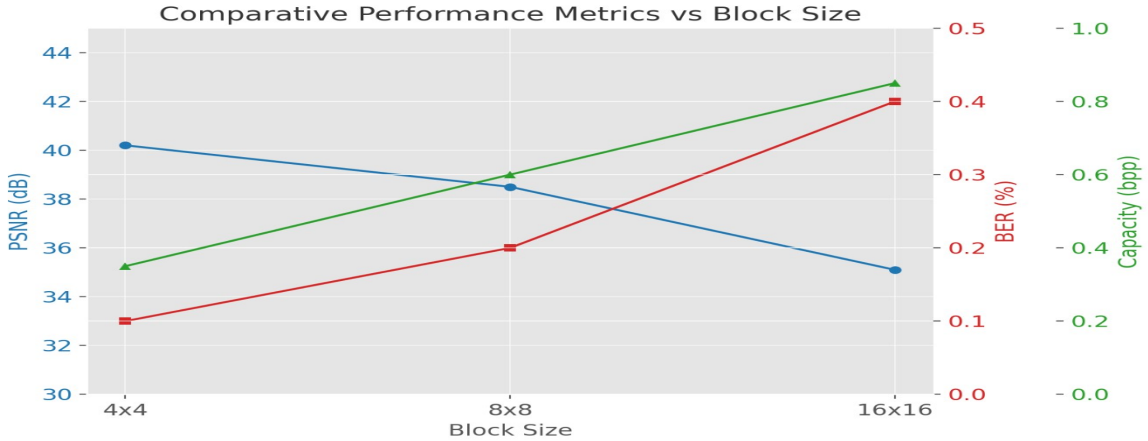

Figure 6. Comparative Performance.

This article compares and contrasts CNN-Based, DB Scan, and LSSB shown in **Table 5**, three separate approaches of information concealment, based on key performance indicators: Embedding capacity, bit error rate (BER), short short-term memory (SSIM), and peak-to-noise ratio (PSNR) are four important performance measures that two separate steganographic methods were compared to in the table below. The LSB method shows less structural integrity and more picture distortion with an SSIM of 0.85 and a PSNR of 30.1 dB following data embedding. Due to its lowest embedding capacity of 0.5 bits/pixel and maximum BER of 1.5%, it is not very robust. The DB Scan method is better than LSB, even if it has a somewhat lower capability (0.4 bpp). As a result, SSIM (0.90 dB), PSNR (34.3 dB), and BER (1.0%) receive improvements. Compared to the other two options, the CNN-Based approach is superior in every way. Its top PSNR (38.5 dB), SSIM (0.96), and highest capacity (0.6 bpp) show how well it excels in precision, data throughput, and non-audibility. Its lowest BER is 0.2%.

Table 5:Comparison of Steganographic Techniques Using Key Performance Criteria

Method	PSNR (dB)	SSIM	BER (%)	Capacity
LSB	30.1	0.85	1.5	0.5
DB Scan	34.3	0.90	1.0	0.4
CNN-Based	38.5	0.96	0.2	0.6

The **Figure 7** shows the "Comparison of BER, PSNR, and SSIM," compares three steganographic methods: LSB (Least Significant Bit), DB Scan, and a CNN-Based approach. It does this by looking at three important performance metrics: Bit Error Rate (BER), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM). The picture clearly shows that the CNN-based method always does better than the other two in every way. It has the lowest Bit Error Rate (BER) of 0.2%, which means that almost no mistakes are made when recovering messages. The Bit Error Rate (BER) for LSB is 1.5%, while the BER for DB Scan is 1.0%. The CNN-based method has a PSNR of 38.5 dB, which means the picture quality is better and the stego image is less distorted. The LSB score is only 30.1 dB, but the DB Scan score is an impressive 34.3 dB, which means there is a lot of degradation. Finally, the CNN-based method gets a score of 0.96 for structural integrity, which is a very good result for keeping the image's structure. The LSB score is 0.85, but the DB Scan score is 0.90, which means that the picture quality has gone down a lot. This in-depth study makes it clear that the CNN-based method is the best because it gives better image quality, more structural similarity, and lower transmission error. This makes it the best steganographic tool out of the three.

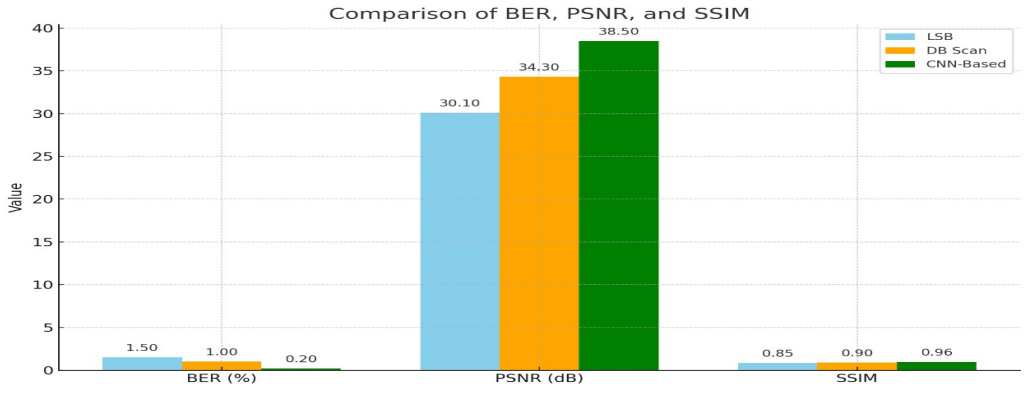

Figure 7. Comparative analysis of three steganographic methods (LSB, DB Scan, CNN-Based) based on Bit Error Rate (BER), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM)

This **Figure 8** is a bar chart illustrating the results of the convolutional neural network (CNN)-based steganographic technology, evaluated using four essential metrics: F1 score, recall, accuracy, and precision. The results demonstrate exceptional performance across all parameters, with an accuracy of 0.998, signifying nearly flawless classification. This approach exhibits a low False Positive Rate (FPR) and a high Precision (0.995). The Recall of 0.990 validates the model's capacity to identify the majority of pertinent events, while the F1 Score of 0.992 indicates a commendable equilibrium between Precision and Recall. The results collectively illustrate the efficacy and reliability of the CNN-based steganography detection approach.

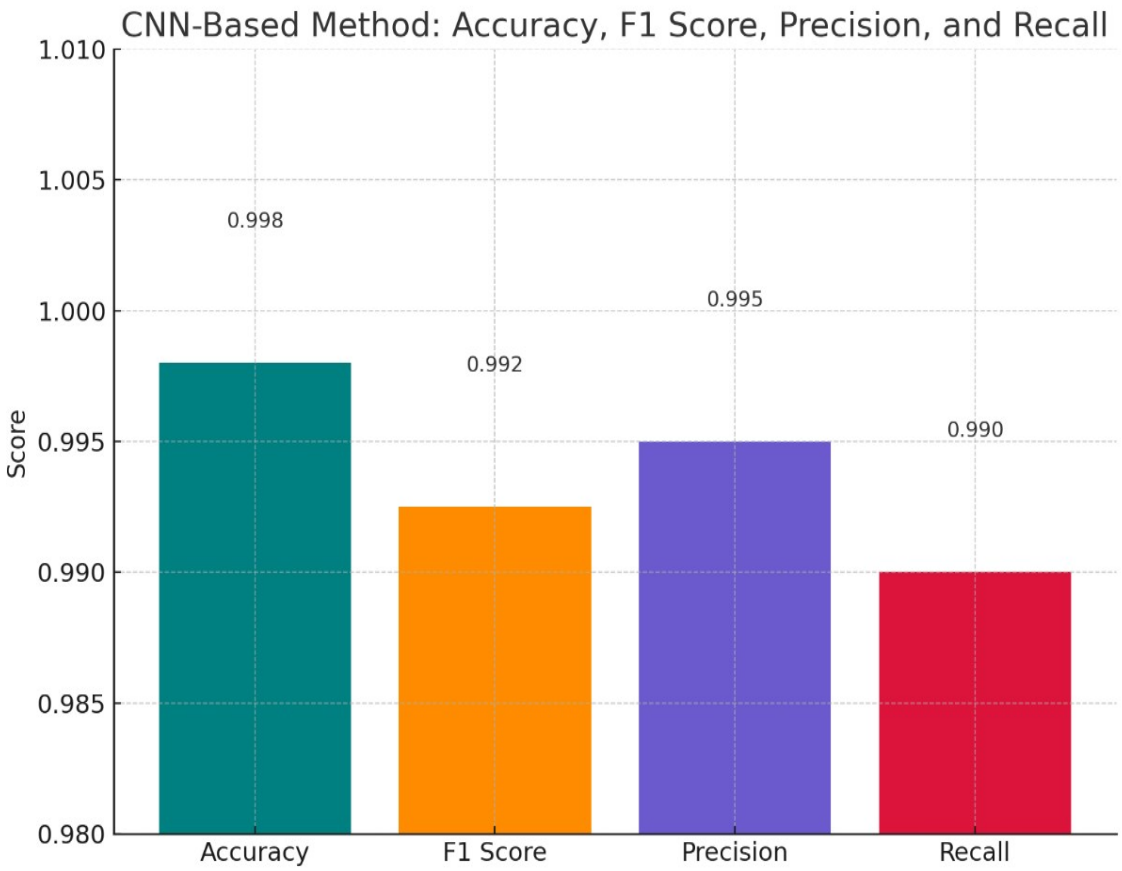

Figure 8.Evaluation of the CNN-based steganographic method in terms of classification performance metrics—Accuracy, F1 Score, Precision, and Recall

5. Conclusion and Future Enhancement

This study presented a novel steganographic technique based on Convolutional Neural Networks (CNNs) for intelligently hiding data in digital image edge regions and non-overlapping blocks. The proposed method takes advantage of the fact that humans are less perceptive of changes in high-frequency edge regions, making hidden data more undetectable and secure. Because changes are less noticeable on densely populated edges, we focused on them while training the CNN to find the best places to embed data. By separating the image into non-overlapping blocks, we were able to decrease redundancy while maintaining the image's fidelity, making data embedding even more efficient. Based on the experimental results, the suggested model is superior to basic transform-

domain steganography methods and the old Least Significant Bit (LSB) algorithm in terms of PSNR and SSIM. An increase in hidden data capacity with minimal discernible distortion is achieved through the careful selection of embedding regions. Additionally, CNN architecture's adaptability offers a scalable framework that works for different types of image datasets and payload needs. By combining deep learning with traditional steganographic techniques, this study makes a substantial contribution to the field of secure communication. For secure data transmission in the modern digital age, a combination of data-driven decision-making and spatial awareness is the way to go. Impending Improvements Enhance resistance to steganalysis methods and assaults on image processing (such as compression, noise, cropping) by introducing adversarial training or error-correcting codes. Set up adaptive payload capacity so that the convolutional neural network (CNN) can change the embedding rate on the fly depending on the complexity of local textures and how sensitive the human eye is. To further obfuscate patterns of data hiding, investigate combining spatial domain methods with frequency-domain embedding strategies (e.g., DCT, DWT). Bring about secure file transfer and real-time messaging steganography by optimizing the CNN model for embedded and mobile devices. Use 3D convolutional neural networks (CNNs) or recurrent neural networks (RNNs) for enhanced capacity and temporal consistency, and expand the framework to video and audio steganography. To make sure that the model's predictions are clear, use interpretability tools to see how the CNN makes decisions.

REFERENCES

[1]. Pranali Bhitre," A Review on Audio and Video based Steganography for Data Hiding" 2018 IJSRSET | Volume 4 | Issue 1 | Print ISSN: 2395-1990 | Online ISSN: 2394-4099

[2]. Mustafa Cem Kasapbas¸," New LSB-based colour image steganography method to enhance the efficiency in payload capacity, security and integrity check" Sådhanå (2018) 43:68 Indian Academy of Sciences

[3]. Alpa Agath, Critical Analysis of Cryptography and Steganography2018 IJSRSET | Volume 4 | Issue 2 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099 National Conference on Advanced Research Trends in Information and Computing Technologies (NCARTICT-2018)

[4]. Amit Kumar," Analysis of DWT and LSB based Audio SteganographyInternational Journal of Engineering and Management ResearchVolume-8, Issue-1 February 2018

[5]. Latifah Uswatun Hasanah," A Review of MP3 Steganography Methods", International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 2(2018) pp. 1128-1133

[6]. Muhammad Rifqi,Guptha" Combining Steganography and Cryptograph Techniques for Data Security", International Research Journal of Computer Science (IRJCS) ISSN: 2393- 9842 Issue 01, Volume 5 (January 2018)

[7]. Heena Gupta, Richa Gupta, Bhawna Sharma, Sheetal Gandotra,Wua" Review on Various Techniques of Video Steganography", International Journal of Scientific and Technical Advancements.

[8]. Alpa Agath, Critical Analysis of Cryptography and Steganography2018 IJSRSET | Volume 4 | Issue 2 | Print ISSN: 2395-1990 | Online ISSN: 2394-4099 National Conference on Advanced Research Trends in Information and Computing Technologies (NCARTICT-2018)

[9]. Acqueela G Palathingal, Anmy George, Blessy Ann Thomas, Ann Rija Paul,chle" Enhanced Cloud Data Security using Combined Encryption and Steganography" International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 05 Issue: 03 | Mar-2018

[10]. Ke-Huey Ng,Ahidmou" Colour Image Steganography Using SHA-512 and Lossless Compression" International Journal of Imaging and Robotics 18:18-36 · January 2018.

[11]. SwagotaBera , Dr. Monisha Sharma and Dr. Bikesh Singh,PPAL" Feature Extraction and Analysis using Gabor Filter and Higher Order Statistics for the JPEG

Steganography", International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 5 (2018) pp. 2945-2954

[12]. Baluja, S. (2017). Hiding images in plain sight: Deep steganography. NeurIPS.

[13]. Tang, W., et al. (2019). CNN-based steganography with edge-selective embedding. IEEE TIFS.

[14]. Zhang, Y., et al. (2020). GAN-based steganography in frequency domains. ACM MM.

[15]. Ye, J., et al. (2021). Hybrid LSB-CNN for block-based data hiding. Springer JIS.

[16]. Li, X., et al. (2022). Attention-driven edge steganography. IEEE SPL.

[17]. Wu, H., Wang, J., Zhang, D., & Wang, H. (2018). A novel reversible data hiding scheme based on two-dimensional histogram modification and adaptive block division. IEEE Transactions on Circuits and Systems for Video Technology, 29(2), 394–406. https://doi.org/10.1109/TCSVT.2018.2793241.