IMAGE STEGANOGRAPHY IN DIGITAL IMAGES

Vinay Kumar [1], Vaddepally Niharika[2], Pudari Sathvika[3], Patlolla Manideep[4], Vazeer Siri Vennela[5]

[1] Assistant Professor, Department *of CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING) TKR COLLEGE OF ENGINEERING & TECHNOLOGY*

[2,3,4,5] *UG Scholars in Department of CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING) TKR COLLEGE OF ENGINEERING & TECHNOLOGY*

ABSTRACT: In the rapidly advancing digital world, the need for secure data transmission has become paramount, leading to the significant development of image steganography. Image steganography involves embedding secret messages within digital images, ensuring that the existence of hidden information remains undetectable. This survey paper provides a comprehensive classification and analysis of various image steganography methods, categorized into Spatial Domain, Transform Domain, Adaptive Steganography, and Statistical Methods. By examining each method's principles, advantages, and limitations, this paper identifies critical challenges and research gaps in the field. These include enhancing robustness against image processing attacks, balancing computational efficiency, increasing embedding capacity, improving detection resistance, simplifying implementation, and enhancing key management and synchronization. Furthermore, the paper evaluates recent advancements, particularly those involving deep learning techniques such as Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs), and their impact on improving the effectiveness and robustness of steganographic methods.

To make the “Image-Stegano” tool more effective we have also provided the users following options-Analysis of image using inversion and different color maps: Grayscale analysis of image, altering threshold of image (Histogram), Providing metadata about the image, Chunks analysis of .PNG images, Extraction of appended Data (.PNG and .BMP files).

1. INTRODUCTION

Steganography is a technique used to conceal secret information within ordinary digital media, making it invisible to unintended recipients. Among various methods, the Least Significant Bit (LSB) substitution is widely adopted for image-based steganography due to its simplicity and minimal impact on image quality. By altering the least significant bits of pixel values, data can be embedded without noticeably changing the visual appearance of the image. This paper presents a novel enhancement to the traditional LSB method, aiming to improve data capacity, security, and robustness against image processing operations. The proposed technique adapts the embedding process based on pixel characteristics, ensuring better imperceptibility and resistance to distortion. It offers a promising solution for secure communication in applications such as watermarking, digital rights management, and confidential data exchange This paper presents an improved steganography technique using Least Significant Bit (LSB) substitution in digital images. It enhances data hiding capacity and robustness while maintaining high image quality and imperceptibility. The

method adapts to pixel characteristics, ensuring secure and undetectable communication.

1.1 Motivation

In today's digital landscape, the need for secure and covert communication has become increasingly critical. With the rapid growth of internet usage and multimedia sharing, sensitive information is often at risk of interception or unauthorized access. Traditional encryption methods, while effective, can draw attention to the presence of hidden data. Steganography offers a discreet alternative by embedding information within digital media, such as images, without altering their visible appearance. Among various techniques, the Least Significant Bit (LSB) substitution method stands out for its simplicity and efficiency, making it a popular choice for image-based data hiding. However, conventional LSB techniques face challenges in terms of robustness, capacity, and resistance to image manipulation. This motivates the development of an improved approach that enhances the reliability and security of hidden communication. The proposed technique aims to optimize the LSB method by adapting the embedding process to pixel characteristics, thereby increasing imperceptibility and resilience against common image processing operations. By addressing these limitations, the research contributes to the advancement of steganographic methods suitable for real-world applications such as digital watermarking, confidential data exchange, and copyright protection. The growing demand for secure digital communication motivates the need for advanced steganographic techniques. Traditional LSB methods, while simple, often lack resilience against image manipulation and detection. Enhancing these methods can significantly improve data protection in multimedia environments. The proposed technique addresses these gaps by introducing adaptive embedding strategies. This ensures more reliable and imperceptible data hiding for real-world applications.

1.2 Proposed System:

The proposed system introduces an enhanced steganographic technique that builds upon the traditional Least Significant Bit (LSB) substitution method for digital images. Unlike conventional approaches that embed data uniformly across pixel bits, this method employs an adaptive strategy that analyzes pixel characteristics before embedding. By selecting optimal pixels and bit positions based on image content, the system ensures minimal visual distortion and improved imperceptibility. This intelligent embedding process allows for higher data capacity while maintaining the integrity of the cover image. To further strengthen the security and robustness of hidden data, the system incorporates mechanisms to resist common image processing operations such as compression, scaling, and filtering. It also addresses vulnerabilities to steganalysis by introducing randomness in the embedding pattern, making detection more difficult. The proposed technique is evaluated using metrics like Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE), demonstrating superior performance compared to traditional LSB methods. Overall, the system offers a practical and efficient solution for secure, covert communication in digital environments.

2. LITERATURE REVIEW:

1.TITLE:"Digital image steganography: Survey and analysis of current methods"

This paper provides a thorough survey and critical evaluation of digital image steganography techniques available as of 2010. The authors begin by defining steganography—concealing secret data within digital images so that the existence of the information remains undetected—and differentiate it from related fields such as cryptography and watermarking. They highlight the main aim of steganography: enabling covert and secure communication using everyday digital images as covers.

The review systematically covers popular spatial domain methods like Least Significant Bit (LSB) substitution, and advanced transform domain techniques utilizing tools such as the discrete cosine transform (DCT) and discrete wavelet transform (DWT). Hybrid approaches, distortion-based schemes, and adaptive algorithms are also discussed, with the authors analyzing their strengths and weaknesses relative to key criteria: capacity, imperceptibility, robustness, and computational efficiency.

A key contribution of this work is its detailed classification and taxonomy of steganographic methods. The paper also provides an in-depth discussion of the practical challenges in the field, including the choice of cover images, the trade-off between embedding capacity and robustness, and the security of hidden data against steganalysis techniques. The authors review both objective and subjective measures used to assess image quality and security—such as PSNR, visual perception, and statistical detectability. Cheddad et al,, further underscore the dynamic arms race between steganography developers and steganalysis researchers, noting that as hiding techniques become more advanced, detection tools must also improve. They highlight the increasing sophistication of machine learning-based steganalysis and the need for adaptive, intelligent steganographic algorithms that can withstand new forms of attacks. This survey stands as a foundational resource for anyone studying or advancing the field of image steganography.

2.TITLE:**"Two new approaches for secured image steganography using cryptographic techniques and type conversions"**

This research paper introduces and examines two novel methods aimed at significantly enhancing security in digital image steganography: the integration of cryptographic techniques and the innovative use of type conversions. The authors begin by underscoring the fundamental challenges faced by classical image steganography. Traditional methods such as Least Significant Bit (LSB) substitution—where secret data is directly embedded into image pixels—are often vulnerable to detection through statistical analysis and not highly resistant to steganalysis, thus necessitating stronger safeguards.

The first approach advanced in this paper couples standard LSB steganography with robust cryptographic algorithms. Before embedding, secret information is encrypted. This ensures that even if the presence of hidden data is suspected or partially extracted, it remains unintelligible without the cryptographic key. This two-layered security—cryptography plus steganography—dramatically enhances the confidentiality and safety of image-based

data transmission, making unauthorized recovery extremely difficult.

The second approach leverages type conversions. Here, secret data is not only encrypted, but its encoding is manipulated through type conversions before embedding. This process adds complexity and an additional layer of obfuscation, hindering direct or even algorithmic attempts to reveal the concealed information. Such an approach boosts both the embedding capacity and steganographic security, as attackers must contend with transformed and encoded content beyond the reach of typical steganalysis methods.

3. "Image steganography: A review"

This review by Purohit and Sridhar provides a comprehensive examination of image steganography techniques, focusing on how secret data can be securely embedded within digital images for covert communication. The authors begin by explaining the fundamental objectives of steganography, highlighting its role in ensuring data confidentiality and preventing unauthorized detection in the era of rapid digital information exchange.

The paper categorizes image steganography into two principal domains: spatial and transform (frequency) domain methods. In the spatial domain, techniques such as Least Significant Bit (LSB) substitution are discussed, where secret bits are directly inserted into the pixel values, taking advantage of the human visual system's limited sensitivity to small color changes. While such approaches are simple and offer high capacity, they are generally more susceptible to distortion and steganalysis.Transform domain methods, including the Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Discrete Fourier Transform (DFT), embed data in the transform coefficients of image blocks. These methods provide higher robustness and imperceptibility, as the embedded data is less likely to be destroyed by standard image processing or compression, but are computationally more intensive.

4. **."A survey on image steganography techniques"**

This review paper by K. U. Singh provides a detailed examination of the evolution, classification, and comparative performance of digital image steganography techniques. The author begins by explaining the core concept: steganography is the science of hiding secret messages within other seemingly innocuous digital data—in this case, digital images—so that the very existence of the message is concealed rather than simply protecting its content through encryption.

The paper classifies image steganography into two major domains: spatial domain and transform (frequency) domain techniques. In spatial domain approaches, such as Least Significant Bit (LSB) substitution, secret data is directly embedded in the bits of pixel values. These are widely adopted due to their simplicity, high embedding capacity, and minimal effect on image quality. However, Singh notes that they are generally less robust, prone to loss during image manipulations, and more vulnerable to detection by steganalysis.

Transform domain techniques embed information in the transform coefficients of images using mathematical transforms such as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Discrete Fourier Transform (DFT). These methods provide greater robustness against

compression, filtering, and other routine processing because the secret data is dispersed throughout the image rather than confined to individual pixel values. Hybrid approaches bring together the strengths of both domains to improve security and capacity.

5.“A Study and Literature Review on Image Steganography”

This review paper by Parmar and Chouhan provides an insightful overview of the main techniques, developments, and challenges in the field of image steganography. The authors begin by establishing the essential purpose of steganography: to conceal sensitive information within digital images, ensuring that the transmission of secret data remains undetectable to unauthorized parties. This has become increasingly relevant with the expansion of digital communication and multimedia sharing in the information age.

The study categorizes image steganography techniques primarily into spatial domain methods and transform domain methods. Spatial domain techniques such as Least Significant Bit (LSB) substitution are highlighted for their simplicity and high data-hiding capacity. By altering the least significant bits of image pixels, secret data can be embedded without perceptibly changing the image. However, these methods present weaknesses—they are highly susceptible to image modification (such as compression, resizing, or filtering) and are vulnerable to statistical analysis attacks, which can reveal the presence of hidden data.

Transform domain methods constitute the second major category covered in this review. These techniques, including those based on Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT), embed the secret information in the frequency components of images. While more computationally intensive, transform domain approaches offer enhanced robustness against common image manipulations and widespread compression formats like JPEG. They are less visible and harder to detect with casual inspection and basic analytical tools.

3. SEQUENCE DIAGRAM:

The process starts with the user selecting the secret message and a cover image. The system’s stego module then applies its algorithm to embed the secret message within the chosen cover image, with assistance and input relayed to the encryption module for processing. Once the stego image is created, it is saved to a directory and assigned an input name. When the user wishes to extract the hidden information, they select the stego image, which triggers the unstego module to process it using its algorithm and passes it to the decryption module. The decryption module then recovers and displays the secret message, printing it for the user. The flow concludes with the user exiting the software.

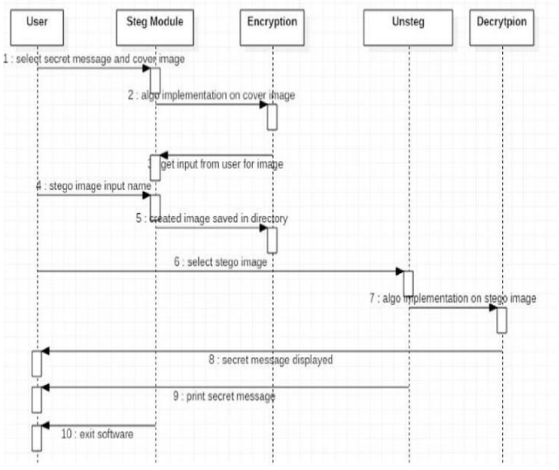

Fig.1: Sequence Diagram

4. ALGORITHMS:

1 LSB

Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image [4]. The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the red, green and blue colour components can be used, since they are each represented by a byte. In other words, one can store 3 bits in each pixel. An 800 × 600 pixel image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data.

2 BIT PLANE

Instead of highlighting gray level images, highlighting the contribution made to total image appearance by specific bits might be desired. Suppose that each pixel in an image is represented by 8 bits. Imagine the image is composed of 8, 1-bit planes ranging from bit plane 1-0 (LSB) to bit plane 7 (MSB).In terms of 8-bits bytes, plane 0 contains all lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all high order bits. Separating a digital image into its bit planes is useful for analyzing the relative importance played by each bit of the image, implying, it determines the adequacy of numbers of bits used to quantize each pixel, useful for image compression.

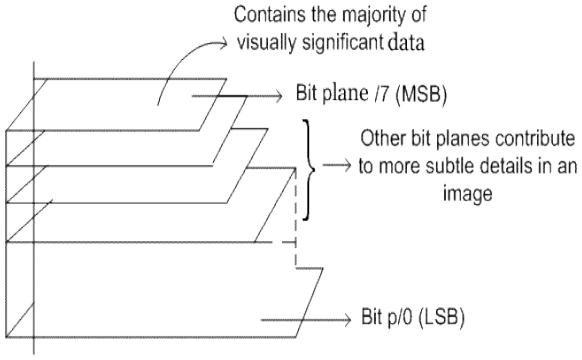

Figure 4. 1: Bit Plane

3 SPIRAL EMBEDDING

The Spiral Embedding orders the pixels of the image in a spiral pattern to prevent the embedding from being as easily decoded. The Spiral Embedding has two main ideas that allow it to be decoded successfully and help thwart visual attacks. The first is that metadata about the image's contents are embedded into known locations. This information makes it possible to decode the stego object and retrieve the secret message. The second is that the data is serialized and embedded in a pattern that destroys the ability to decode the message in a visual attack.The Spiral Embedding begins by building a vector containing all the data that will be embedded into the cover including the metadata and the message contents. The dimensions of the message are embedded losslessly into the first 32 positions in the vector as unsigned 16-bit integers. A bit representing the vector's content is written into the LSB of the cover in a spiral pattern from the outside in, pixel by pixel.Decoding a stego object created with the Spiral Embedding is simply a process of reading in the stored dimensions and then following the same spiral pattern that governed the embedding. The values of the LSBs of the stego object are read into a vector. When the embedded data has been read in its entirety, the vector

is partitioned to create a new image with the message's original dimensions. The result of this embedding can then be displayed.

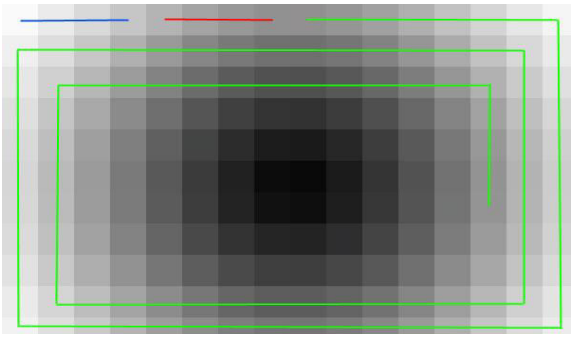

Figure 4.2: spiral embedding pattern

The Spiral Embedding pattern. The height information is embedded into the blue line pixels, the width information into the red pixels, and the message into the green pixels.

4 METADATA MANIPULATION

Metadata is basically data about data. If we think of a image file as our data, metadata would include information such as the name of the image, the title, dimensions , width , height, Video and audio files contain the same types of data.

The Exchangeable Image File (Exif) format is a standard used for recording information about image and sound files created with a digital camera. There are many data fields that can be used to hide information. If we right click on a saved .jpg file and select properties, we will see a small subset of those fields.

5 ALTERING THRESHOLD

Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images. The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity is less than some fixed constant T or a white pixel if the image intensity is greater than that constant.

6 APPENDED DATA

A simple and common steganography technique is to append data at the end of image file. This technique works because most image viewers ignore any appended data and hence a stego message remains hidden. Linux 'cat' command can be used for appending data at the end of image file.

7 PNG CHUNK ANALYSIS

A PNG file starts with an 8-byte signature. After the header comes a series of chunks, each of which conveys certain information about the image. Chunks declare themselves as critical or ancillary, and a program encountering an ancillary chunk that it does not understand can safely ignore it. This forms the basis of steganography. Data can be embedded in unknown ancillary chunks and it will be ignored by image viewers or decoders.

A chunk consists of four parts: length (4 bytes, big-endian), chunk type/name (4 bytes), chunk data (length bytes) and CRC (cyclic redundancy code/checksum; 4 bytes). The CRC is a network-byte-order CRC-32 computed over the chunk type and chunk data, but not the length.

Chunk types are given a four-letter case sensitive ASCII type/name; compare FourCC. The case of the different letters in the name (bit 5 of the numeric value of the character) is a bit field that provides the decoder with some information on the nature of chunks it does not recognize.

The case of the first letter indicates whether the chunk is critical or not. If the first letter is uppercase, the chunk is critical; if not, the chunk is ancillary. Critical chunks contain

information that is necessary to read the file. If a decoder encounters a critical chunk it does not recognize, it must abort reading the file or supply the user with an appropriate warning. The case of the second letter indicates whether the chunk is "public" (either in the specification or the registry of special-purpose public chunks) or "private" (not standardised). Uppercase is public and lowercase is private. This ensures that public and private chunk names can never conflict with each other (although two private chunk names could conflict). The third letter must be uppercase to conform to the PNG specification. It is reserved for future expansion. Decoders should treat a chunk with a lowercase third letter the same as any other unrecognised chunk. The case of the fourth letter indicates whether the chunk is safe to copy by editors that do not recognize it. If lowercase, the chunk may be safely copied regardless of the extent of modifications to the file. If uppercase, it may only be copied if the modifications have not touched any critical chunks[6].

8 CHANGING COLOUR MAP

In computing, indexed color is a technique to manage digital images' colors in a limited fashion, in order to save computer memory and file storage, while speeding up display refresh and file transfers. It is a form of vector quantization compression.

When an image is encoded in this way, color information is not directly carried by the image pixel data, but is stored in a separate piece of data called a palette: an array of color elements. Every element in the known as color registers. The image pixels do not contain the full specification of its color, but only its index in the palette. This technique is sometimes referred as pseudocolor or indirect color, as colors are addressed indirectly. This technique can be exploited to hide data in image using suitable palette or colour map. To decode data we must try different random colour maps.

5. IMPLEMENTATION & RESULTS

5.1 Explanation of Key functions:

1. Embedding Function (Hiding Process)

The embedding function is responsible for concealing secret data within a digital image, referred to as the *cover image*. It modifies the pixel values of the cover image based on a specific steganographic algorithm (commonly LSB substitution).

- It takes:
  - Cover image (original image)
  - Secret message (text/image/data)
  - Stego key (optional for security)
- It modifies pixel values (usually least significant bits

2. Extraction Function (Retrieval Process)

The embedding function is responsible for concealing secret data within a digital image, referred to as the *cover image*. It modifies the pixel values of the cover image based on a specific steganographic algorithm (commonly LSB substitution).

- It takes:
  - Stego image
  - Stego key (if used)
- It reads modified bits to reconstruct the secret data

3. Encoding Function

- The embedding function is responsible for concealing secret data within a digital image, referred to as the *cover image*. It modifies the pixel values of the cover image

based on a specific steganographic algorithm (commonly LSB substitution).

- Text → ASCII → Binary
- Image → Pixel values → Binary

4. Decoding Function

The decoding function converts the extracted binary data back into its original format, such as text or image.

- Binary data → Original message (text/image)

5. LSB Substitution Function (Most Common)

Based on **Least Significant Bit (LSB)** technique.

- Replaces the last bit of pixel values with message bits
- Example:
  Pixel = 11001010 → Replace last bit → 11001011

6. Key Generation Function (Optional Security)

This function generates a steganographic key used to control the embedding and extraction process. It may determine pixel selection or encryption parameters.

- Adds security layer
- Only users with the key can extract data

7. Capacity Function

- The capacity function calculates the maximum amount of data that can be embedded within a given image without significant degradation.
- Depends on:
  - Image size
  - Number of bits used per pixel

8. Distortion / Quality Function

- This function evaluates the quality of the stego image by measuring distortion introduced during embedding.Common metrics:
  - MSE (Mean Square Error)
  - PSNR (Peak Signal-to-Noise Ratio)

9. Security Function

- The security function ensures that the hidden data is resistant to detection and unauthorized extraction. It may involve encryption or randomization techniques.
- Uses:
  - Encryption before embedding
  - Random pixel selection

10.Compression Function (Optional)

- This function reduces the size of the secret data before embedding, allowing more information to be hidden within the image.

➢ Reduces size

➢ Increases hiding capacity

6. OUTPUT & SCREENSHOTS:

6.1 Hide Text output screen

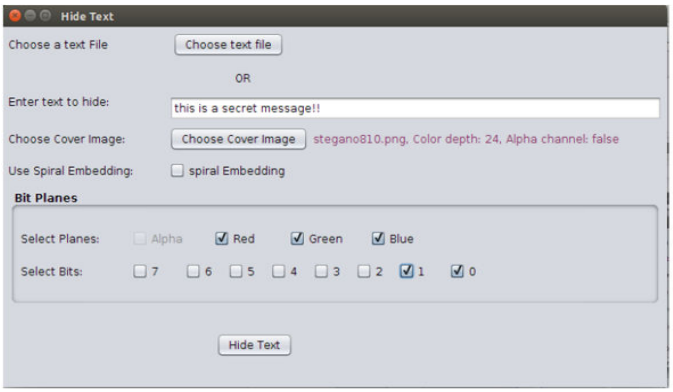

Figure 6.1: hidetext output screen

The user can either upload a text file or manually enter a message, as seen with the input “this is a secret message!!”. A cover image (*stegano810.png*) is selected, and embedding options such as color channels (Red, Green, Blue) and bit planes are

configured, with bits 0 and 1 chosen for LSB embedding. The interface also provides an optional spiral embedding method. Finally, the “Hide Text” button is used to execute the embedding process and generate the stego image.

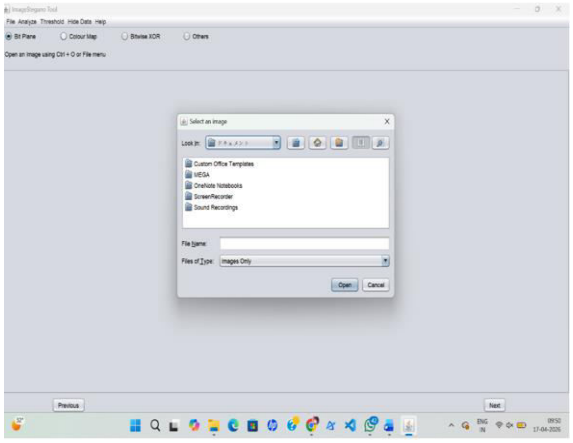

Figure 6.2 : sample file

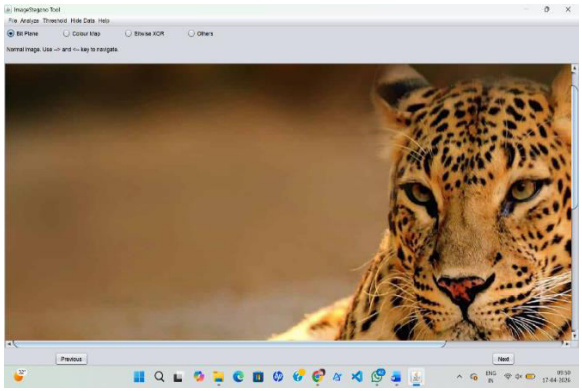

Figure 6.3 : Sample File

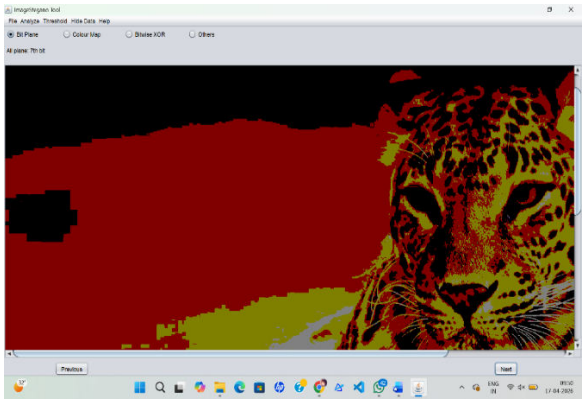

Figure 6.4: All Plane 7th bit

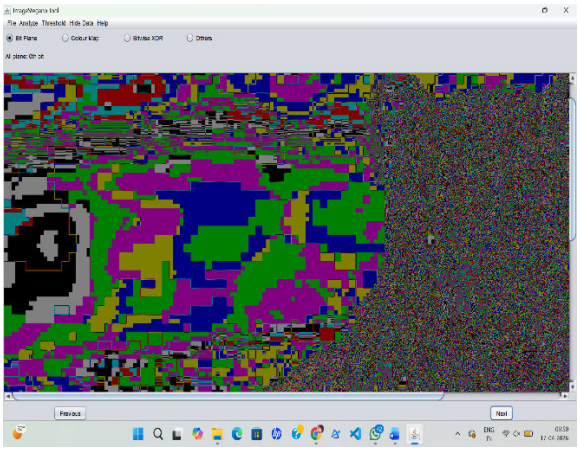

Figure 6.5: All Plane 0th bit

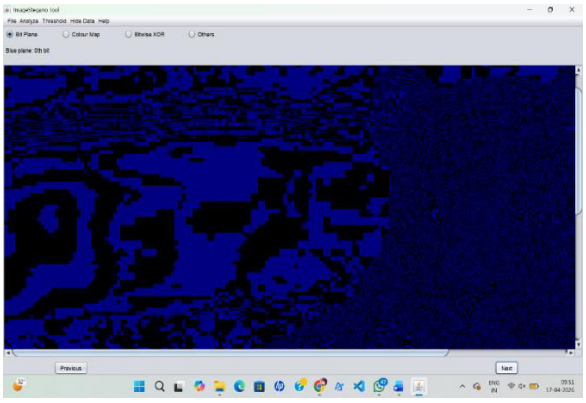

Figure 6.6: Blue Plane 0th bit

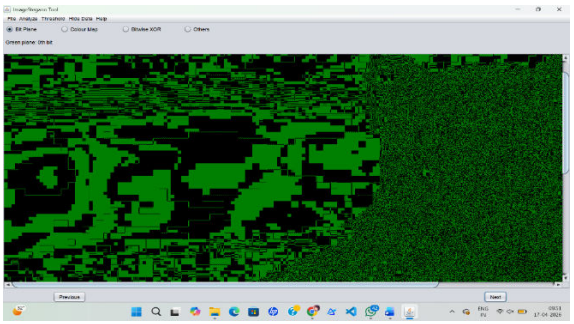

Figure 6.7: Green Plane 0th bit

7. CONCLUSION

The Image Steganography system developed in this project successfully demonstrates the secure and efficient hiding of confidential information within digital images using techniques such as the Least Significant Bit (LSB) method. The primary objective of concealing data without causing noticeable changes to the original image has been achieved, ensuring both secrecy and visual imperceptibility.

Through the implementation process, it was observed that the stego images produced are almost identical to the original images, making it difficult for the human eye or basic analysis techniques to detect the presence of hidden data. The extraction process also proved to be reliable, with the hidden message being recovered accurately under normal conditions.

The testing and validation results, including metrics such as MSE, PSNR, and BER, confirm that the system maintains high image quality and data integrity. The project also highlights the balance between payload capacity and image quality, showing that while larger data can be embedded, it may slightly affect imperceptibility.

However, the study also identifies certain limitations of basic steganographic techniques like LSB, particularly their vulnerability to image processing attacks such as compression, noise addition, and cropping. This indicates the need for more advanced and robust methods to enhance security.

Overall, this project provides a solid foundation in the field of digital image steganography and demonstrates its practical applications in secure communication, data protection, and digital watermarking. Future enhancements can include the integration of encryption techniques, adaptive embedding methods, and the use of transform domain approaches to improve robustness and security

8. REFERENCES:

1.A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, “Digital image steganography: Survey and analysis of current methods,” Signal Processing, vol. 90, no. 3, pp. 727–752, Mar. 2010, doi:10.1016/j.sigpro.2009.08.010.

2. S. Narayana and G. Prasad, “Two new approaches for secured image steganography using cryptographic techniques and type conversions,” Signal & Image Processing: An International Journal (SIPIJ), vol. 1, no. 2, pp. 60–73, Dec. 2010, doi: 10.5121/sipij.2010.1206.

3. A. Purohit and P. S. V. S. Sridhar, “Image steganography: A review,” International Journal of Computer Science and Information Technologies (IJCSIT), vol. 5, no. 4, pp. 4891–4896, 2014.

4. K. U. Singh, “A survey on image steganography techniques,” International Journal of Computer Applications, vol. 97, no. 18, pp. 10–17, Jul. 2014, doi: 10.5120/17001-7153.

5. A. K. M. Parmar and K. Chouhan, “A Study and Literature Review on Image Steganography,” International Journal of Computer Science and Information Technologies (IJCSIT), vol. 6, no. 1, pp. 685–688, 2015.

6. W. Saqer and T. Barhoom, “Steganography and Hiding Data with Indicators-based LSB Using a Secret Key,” Engineering, Technology & Applied Science Research, vol. 6, no. 3, pp. 1013–1017, 2016.