

IMAGE AUTHENTICATION AND TAMPERED DETECTION USING STEGANOGRAPHY APPROACH

REPORT SUBMITTED

BY

ARINDAM GOSWAMI(11500219015)

PRIYANSHU KASHYAP(11500219037)

ANUSHKA GUPTA(11500219038)

RISHABH KUMAR CHAUDHARY(11500219047)

Academic Year 2022-23

UNDER THE GUIDANCE OF

Ms. Sudipta Roy

DEPARTMENT OF INFORMATION TECHNOLOGY

B.P. PODDAR INSTITUTE OF MANAGEMENT AND TECHNOLOGY

FOR THE AWARD OF THE DEGREE OF

Bachelor of Technology

In

Information Technology



DEPARTMENT OF INFORMATION TECHNOLOGY

B.P. PODDAR INSTITUTE OF MANAGEMENT AND TECHNOLOGY

[Affiliated to West Bengal University of Technology]

137, V.I.P. ROAD, PODDAR VIHAR, KOLKATA-700052

IMAGE AUTHENTICATION AND TAMPERED DETECTION USING STEGANOGRAPHY APPROACH

*Arindam Goswami
Priyanshu Kashyap
Anushka Gupta
Rishabh Kumar Chaudhary*

CERTIFICATE

This is to certify that the Project Report entitled, **Image Authentication and Tampering detection using Steganography approach** submitted by **Mr. Arindam Goswami, Mr. Priyanshu Kashyap, Ms. Anushka Gupta and Mr. Rishabh Kumar Chaudhary** of B. P. Poddar Institute of Management and Technology, is a record of Project work carried out by them under my supervision and guidance and is worthy of consideration for the award of the degree of Bachelor of Technology in Information Technology of the Institute.

.....

[Ms. Sudipta Roy]

Assistant Professor, Dept. of Information Technology

B.P. PODDAR INSTITUTE OF MANAGEMENT & TECHNOLOGY

Counter signed by

.....

[Dr. Sabnam Sengupta]

Head of Dept. of Information Technology

B.P. PODDAR INSTITUTE OF MANAGEMENT & TECHNOLOGY

CONTENTS

1. ACKNOWLEDGEMENT
2. ABSTRACT
3. OBJECTIVE
4. INTRODUCTION
5. LITERATURE REVIEW
6. ABOUT PROJECT
7. PROJECT - ANALYSIS
 - EXISTING WORK
 - PROPOSED WORK
 - PURPOSE OF THE PROJECT
8. SYSTEM REQUIREMENTS
9. PROPOSED ALGORITHM
10. SOURCE CODE
11. RESULT ANALYSIS
12. FEATURES
13. ADVANTAGES
14. FUTURE SCOPE
15. CONCLUSION
16. REFERENCE

1. ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our mentor, **Ms. Sudipta Roy**, for her invaluable guidance and support throughout the development of the proposed embedding algorithm and Tampering detection and analysis for secure image steganography. Her expertise and insightful suggestions have been instrumental in shaping the direction and execution of this research. **Ms. Sudipta Roy** profound knowledge in the field of steganography and cryptography has greatly influenced our understanding of the subject matter. Her guidance helped us navigate through complex concepts and methodologies, ensuring that our algorithm adhered to the highest standards of security and efficiency.

We are grateful for his constant availability and willingness to provide constructive feedback and advice. Her meticulous attention to detail and rigorous review of our work have significantly enhanced the quality and validity of our research findings. Furthermore, **Ms. Sudipta Roy** ability to inspire and motivate us throughout the research process has been truly remarkable. Her passion for the subject and dedication to our growth as researchers have been a constant source of inspiration and encouragement.

We would also like to acknowledge her role in fostering a collaborative and conducive research environment. His open-mindedness and willingness to listen to our ideas have facilitated meaningful discussions and critical thinking, contributing to the overall success of this project. We extend our deepest appreciation to **Ms. Sudipta Roy** for her unwavering support, mentor-ship, and guidance. Her expertise, patience, and encouragement have been indispensable to our research journey, and we are sincerely grateful for the opportunity to work under her supervision.

2. ABSTRACT

The validity of digital photos has recently come into doubt due to the quick development of digital image altering technologies. A system called picture tampering detection uses the inherent image regularities to identify manipulated photographs. Existing intrinsic image regularities, however, are made for a certain class of tampering procedures. picture tampering detection accuracy is significantly reduced when many types of tampering activities are utilized to manipulate a digital picture. The re-normalized histogram of noise and noise difference, which is calculated as the histogram to peak-value ratio, is used in this study to identify a new class of intrinsic picture regularities. The re-normalized histogram would rise when the peak value of this histogram fell when the image was altered through image tampering. Making use of the re-normalized the experimental findings demonstrate that the suggested technique can detect a variety of tamper activities, including single type and many kinds of tamper operations, without knowing the tampering operation sequence, type, or parameter beforehand.

To detect tampering, the original cover image is first inputted, and the watermark image is hidden using DCT and PN sequence. Tampering is then applied to the stego image during transmission, incorporating noise. At the receiver's end, the tampered image with noise is received, and the watermark image is extracted from the stego image. The detection process involves comparing the pixel byte elements and computing ratios and average values for specific sub-block images between the original and extracted watermark images. Threshold values are used to determine the distortion and tampering effects in the extracted watermark image.

By evaluating the ratios and threshold values, the overall distortion scenario in the extracted watermark image can be observed, providing insights into the extent of tampering and distortion caused during the embedding and extraction processes.

3. OBJECTIVE

This project aims to develop a steganographic algorithm that encrypts an 32X32 bitmap (.bmp) image within another image (512X512) and develop a tampering detection algorithm by doing pixel and variance operation and doing analysis of the manual tampering and noise to provide security and authentication to prevent any third party from tampering with the document or image. The project focuses on application-based authentication protocol. This project also focuses on accurately embedding the hidden image in the image by reducing the noise as much as possible. The objective of this research is to develop a novel embedding algorithm and tamper detection algorithm for secure image transmission that combines the DCT technique, PN Sequence operation, and public key encryption. The algorithm aims to achieve the following:

- 1. Security:** Ensure the confidentiality and integrity of the hidden information within the stego (watermarked) image by implementing robust encryption techniques and resistance against data security methods.
- 2. Visual Quality Preservation:** Minimize the impact on the visual quality of the cover image during the embedding process, ensuring that any alterations made to the image are imperceptible to the human eye.
- 3. High Embedding Capacity:** Accommodate a significant amount of hidden data within the cover image while maintaining an acceptable level of visual quality and ensuring efficient utilization of the available embedding space.
- 4. Practical Applicability:** Develop an algorithm that is computationally efficient and compatible with various image formats, making it suitable for real-time applications and integration with existing steganographic systems and analyzing the noise and tampering by third party.
- 5. Comparative Performance:** Evaluate the proposed algorithm against existing embedding techniques and tampering detection and analysis in terms of security, embedding capacity, imperceptibility, and resistance to watermarking techniques to demonstrate its superiority and effectiveness.

By achieving these objectives, the research aims to contribute to the advancement of secure image transmission, providing a reliable and efficient method for confidential data transmission within digital images.

4. INTRODUCTION

In the field of digital image processing, ensuring the security and integrity of images has become increasingly important. One approach to address this concern is the embedding of a secret image within a cover image using techniques such as Discrete Cosine Transform (DCT) and Pseudo-Noise (PN) sequences. This report focuses on the process of embedding a 32x32 bitmap (.bmp) image into a 512x512 cover image, as well as the subsequent extraction process involving tampering detection and noise analysis.

The embedding algorithm begins by converting the original RGB image into the YCbCr color space. The Y-channel is then extracted for watermark embedding purposes. To enhance the security of the embedding process, a pseudo-random sequence known as the PN sequence is generated using a security key. This PN sequence is utilized to permute the Y-channel, introducing an additional layer of protection. The permutation is then reversed, and the individual color channels are recombined to form the RGB watermarked image.

The extraction process involves a similar approach. The Y-channel is extracted from the watermarked image, and DCT is applied to 8x8 blocks. The difference between the pixel locations where the watermark was embedded is calculated. A negative difference corresponds to foreground pixels, while a positive difference corresponds to the background pixels of the watermark. This difference serves as a basis for extracting the secret image from the watermarked image.

In addition to the embedding and extraction processes, this report also focuses on tampering detection and noise analysis. Tampering detection is achieved by introducing intentional modifications to the watermarked image during transmission, simulating potential attacks or alterations. Incorporating noise into the image during tampering further enhances the realism of the analysis. At the receiver's end, the tampered image with integrated noise is processed to extract the watermark image. By tracking the differences between the original and extracted watermark images, evaluating the average values of sub-block elements, and comparing them with predefined threshold values, the distortion and tampering effects on the extracted watermark image can be assessed.

Through a comprehensive analysis of the quantified values obtained from the tampering detection and noise analysis processes, this report aims to provide insights into the distortion and tampering scenarios in the extracted watermark image. This information can contribute to the development of robust techniques for secure image embedding, extraction, and tampering detection, ensuring the integrity and authenticity of digital images in various applications.

STEGANOGRAPHY:

Steganography is the practice of concealing secret information within a cover medium, such as an image, audio file, video, or text, without arousing suspicion from unintended recipients. It aims to hide the existence of the embedded data, making it difficult to detect.

There are several types of steganography techniques, including:

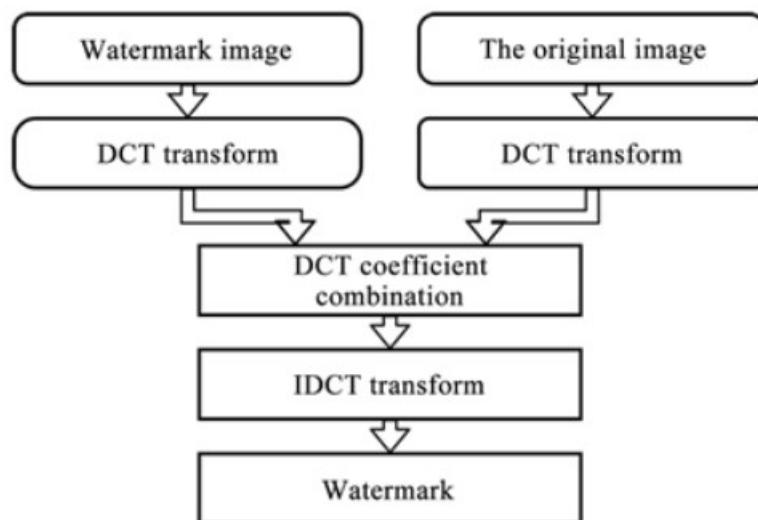
1. **Image Steganography:** Concealing data within digital images by modifying the pixel values. It can involve techniques like DCT,DWT,DFT,least significant bit (LSB) substitution, where the least significant bits of pixel values are replaced with secret data.
2. **Audio Steganography:** Embedding hidden information within audio files by exploiting imperceptible modifications. It can be done by modifying the amplitude or phase of audio samples.
3. **Video Steganography:** Hiding data within video files by manipulating frames or video components. Techniques include modifying motion vectors or adding imperceptible changes to the video frames.
4. **Text Steganography:** Concealing data within the structure or content of a text document. This can involve techniques like using invisible ink, special character encoding, or formatting changes.
5. **Network Steganography:** Embedding secret data within network protocols or communication channels. It can involve techniques like modifying packet timing, using unused header fields, or manipulating network traffic patterns.
6. **Digital Watermarking:** Embedding hidden information, usually in the form of a unique identifier or copyright mark, within multimedia content. Watermarks are often designed to be robust against various attacks and can be used for copyright protection or content authentication.

These are just a few examples of steganography techniques, and there are many variations and hybrid methods used in practice. The goal of steganography is to ensure that the embedded information remains covert and resistant to detection by unintended recipients or adversaries.

WATERMARKING:

Image digital watermarking is a technique used to embed hidden information or a digital signature within a digital image. There are two main types of digital watermarks: visible and invisible. Visible watermarks are usually overlaid on top of the image and are easily visible, often consisting of text or a logo. They are commonly used to indicate ownership or copyright information.

Invisible watermarks, on the other hand, are embedded within the image itself and are not easily noticeable to the human eye. They are designed to be robust against various image processing operations and are difficult to remove without specialized knowledge or software. Invisible watermarks can be detected and extracted using specific algorithms or software tools, allowing the image owner to verify its authenticity or trace unauthorized use.



TYPES OF WATERMARKING:

- **Visible Watermarking:**

Visible watermarking involves embedding visible information or logos onto an image or video. These watermarks are easily noticeable and serve as a form of copyright protection or branding. They are typically added in a semi-transparent manner, allowing the underlying content to remain visible. Visible watermarks are commonly used in digital media to deter unauthorized usage or redistribution.

- **Invisible Watermarking:**

Invisible watermarking, also known as digital or imperceptible watermarking, involves embedding a watermark that is not visually apparent in the content. The watermark is imperceptible to human observers, but it can be detected and extracted using appropriate techniques. Invisible watermarks are primarily used for copyright protection, content authentication, and tamper detection.

- **Spatial Domain Watermarking:**

Spatial domain watermarking refers to the embedding of the watermark directly into the spatial domain of the host image or video. The watermark bits are typically embedded by modifying the pixel values or the least significant bits (LSBs) of the image. Spatial domain techniques are simple and computationally efficient but can be vulnerable to attacks such as image cropping or resizing.

- **Frequency Domain Watermarking:**

Frequency domain watermarking involves embedding the watermark in the frequency domain of the host image or video. Commonly used techniques include Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), or Discrete Wavelet Transform (DWT). Frequency domain watermarking provides robustness against common signal processing operations and offers better resistance to attacks compared to spatial domain techniques.

- **Spread Spectrum Watermarking:**

Spread Spectrum watermarking techniques involve spreading the watermark information across the entire host signal using a pseudo random (PN) noise sequence. This approach provides robustness against attacks and ensures that the watermark remains detectable even if parts of the host signal are altered or removed. Spread Spectrum watermarking is commonly used in audio and image watermarking applications.

TYPES OF TRANSFORMATION DOMAIN:

- **Discrete Cosine Transform (DCT):**

DCT is a widely employed transform domain for watermarking applications. It converts the spatial domain representation of an image into frequency domain coefficients. The DCT coefficients capture the image's energy distribution and are suitable for embedding watermarks due to their energy compaction properties. DCT-based watermarking techniques offer good robustness against common attacks such as compression and filtering.

- **Discrete Wavelet Transform (DWT):**

DWT is a versatile transform domain that provides both spatial and frequency localization. It decomposes an image into different frequency subbands, capturing both high-frequency details and low-frequency components. Watermarking in the DWT domain enables embedding the watermark selectively in specific subbands, providing robustness against various signal processing operations. DWT-based techniques are well-suited for scalable and robust watermarking applications.

- **Discrete Fourier Transform (DFT):**

DFT is a transform domain that converts an image from the spatial domain to the frequency domain. It represents the image using complex numbers, where the magnitude and phase spectra carry important frequency information. DFT-based watermarking techniques exploit the frequency components of the image to embed and extract watermarks. They are commonly used in audio and speech watermarking applications.

- **Discrete Radon Transform (DRT):**

DRT is a transform domain used for watermarking applications, particularly in medical imaging. It is based on the Radon transform, which captures the image's line integrals or projections from different angles. Watermark embedding and extraction in the DRT domain involve modifying the projection data to embed and retrieve the watermark information.

- **Singular Value Decomposition (SVD):**

SVD is a matrix decomposition technique used for watermarking applications, primarily in image and video watermarking. It decomposes an image or video into three matrices: the left singular vectors, singular values, and right singular vectors. Watermark embedding and extraction in the SVD domain involve modifying the singular values to embed and retrieve the watermark. SVD-based watermarking techniques offer good robustness against geometric transformations and compression.

PERFORMANCE METRICS:

The following performance metrics to evaluate the effectiveness of the proposed embedding and extraction processes:

Mean Square Measurement Error (MSME):

MSME measures the average square difference between the original watermark image and the extracted watermark image. It quantifies the overall distortion or error introduced during the embedding and extraction process. A lower MSME value indicates better accuracy and fidelity in extracting the watermark.

$$MSE = 1/n * \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Peak Signal-to-Noise Ratio (PSNR):

PSNR is a widely used metric to assess the quality of the extracted watermark image compared to the original watermark image. It measures the ratio between the maximum possible power of the image and the power of the error introduced during embedding and extraction. A higher PSNR value indicates better preservation of the watermark image's quality and integrity.

$$PSNR = 10 * \log_{10}(MAX^2 / MSE)$$

Structural Similarity Index (SSI):

SSI evaluates the structural similarity between the original watermark image and the extracted watermark image. It considers not only pixel-wise differences but also the perception of structure and texture. SSI provides a measure of how well the extracted image retains the structural characteristics of the original watermark. A higher SSI value indicates better preservation of structural similarity.

$$CC = \sum[(O - \bar{O}) * (E - \bar{E})] / [\sqrt{\sum(O - \bar{O})^2} * \sqrt{\sum(E - \bar{E})^2}]$$

Correlation Coefficient (CC):

CC measures the linear relationship between the original watermark image and the extracted watermark image. It quantifies the level of correlation between the two images, indicating how accurately the extracted watermark represents the original watermark. A higher CC value indicates a stronger correlation and better fidelity in the extraction process.

$$SSI = [SSIM(O, E)]^\alpha * [MSSIM(O, E)]^{(1-\alpha)}$$

These performance metrics provide quantitative measures to evaluate the success of the embedding and extraction algorithms in preserving the quality, accuracy, and structural characteristics of the watermark image. By analyzing these metrics, you can assess the effectiveness and robustness of the proposed approach in embedding and extracting the secret image, as well as detect any potential distortions or tampering.

TYPES OF IMAGE TAMPERING TECHNIQUES:

Image tampering techniques can be broadly categorized into two main types: image manipulation techniques and image forgery techniques. Here's an overview of each type:

1. Image Manipulation Techniques:

- a. Cropping and Resizing: Cropping or resizing an image to remove or alter certain elements.
- b. Copy-Move: Copying a region from one part of the image and pasting it onto another area to conceal or duplicate objects.

2. Image Forgery Techniques:

- a. Splicing: Combining parts of multiple images to create a composite image.
- b. Cloning: Duplicating objects or regions within an image to make it appear as if they exist in multiple locations.
- c. Object Removal: Removing unwanted objects or elements from an image while attempting to maintain the overall visual consistency.
- d. Text and Logo Insertion: Adding or overlaying text, logos, or watermarks onto an image to alter its context or claim ownership.
- e. Image Noise Addition: Introducing random noise or artifacts to the image to degrade its quality and make manipulation harder to detect.

It's important to note that these techniques can be used individually or in combination to achieve more image tampering. The field of image forensics focuses on developing methods to detect and analyze such tampering to ensure the authenticity and integrity of digital images.

NOISE:

In image detection techniques, "noise" refers to unwanted and random variations or disturbances that can occur in an image. It can obscure the underlying information and affect the accuracy and quality of image analysis, recognition, or processing algorithms. Noise can be introduced during image acquisition, transmission, or storage processes.

Various factors can contribute to the presence of noise in images, such as electronic interference, sensor limitations, compression artifacts, environmental conditions, and inherent properties of the imaging system. Different types of noise can occur, including:

Gaussian Noise: Also known as additive white Gaussian noise (AWGN), it is a type of random noise that follows a Gaussian distribution. It is characterized by small fluctuations in pixel values and can be caused by electronic interference or sensor noise.

Salt and Pepper Noise: It is a type of random noise that manifests as randomly occurring white and black pixels scattered throughout the image. It can be caused by errors in data transmission or faults in image acquisition devices.

Poisson Noise: It is a type of noise that occurs in images with low light levels, such as medical or astronomical images. Poisson noise is characterized by random variations in pixel intensities and follows a Poisson distribution.

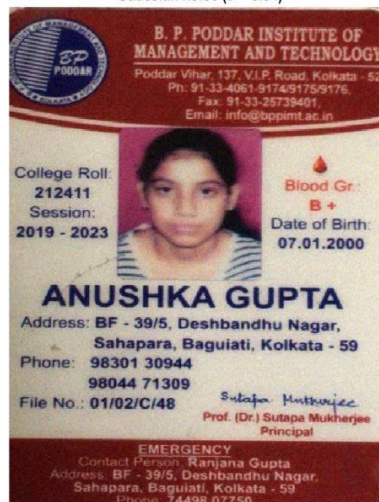
Speckle Noise: It is a type of multiplicative noise that appears as granular patterns in an image. Speckle noise is commonly found in ultrasound or synthetic aperture radar (SAR) images. It is caused by interference patterns or random fluctuations in the imaging process.

Quantization Noise: It occurs when an analog signal is digitized, resulting in the loss of information due to limited bit depth. Quantization noise appears as a low-level noise that introduces small errors in pixel values.



Stego Image without Tampering

Gaussian noise ($\sigma = 0.04$)



Tampered Image with 4% Gaussian noise

Salt and pepper noise (density = 0.10)



Tampered image with 10% salt and pepper noise

5. LITERATURE REVIEW

Deepika Sharma et al. Pointed in “**Digital Image Tampering – A Threat to Security Management**” that the ability to transmit digital images using imaging technologies like digital cameras, scanners, photo-editing, and software packages has significantly increased over the past few years as a result of significant advancements in computing and network technologies as well as the availability of better bandwidths. The application of this technique, however, also extends to the manipulation of digital photos and the production of convincing photocopies. As a result of the easy availability of digital photos and the availability of free image-altering tools, there is a growing challenge in determining the authenticity of digital photographs.

Detection techniques, according to **Minati Mishra et al.** Proposed a method in “**Digital Image Tamper Detection Techniques - A Comprehensive Study**” in the International Journal of Computer Science and Business Informatics 2013 which is primarily based on Both active and passive tamper detection methods have been the subject of much research, and there is still much work being done globally to effectively detect tampering in digital photographs. Splicing and cloning, two commonly employed passive detection approaches, have been discussed in this study. However, it's noteworthy to note that both tamper detection and tampering strategies are developing. Th issue is becoming more difficult as newer iterations of detection-resistant tampering methods are growing in response to the introduction of a new picture authentication and tamper detection approach.

Arshad Jamal et al. Demonstrated in “**Digital Watermarking Techniques And its Application Digital Halal Certificate: A Survey**” where he discussed the primary basic concepts of watermarking techniques in order to build a more reliable, high authenticity, and high-security watermarking, a number of studies and research. In the area of digital image processing, a variety of watermarking techniques and algorithms are available.

Rajkumar Ramasamy et al. discussed about different algorithms in, “**Digital watermarking —A tutorial**” in IEEE Conference 7 July 2022 that In order to build more reliable, high authenticity, and high-security watermarking, a number of studies and research that has presented various aspects of digital watermarking, including the principles, requirements, various domains with basic algorithms, and comparison parameters. Also, they have tried to give complete basic information about digital watermarking, which will help new researchers to get the maximum knowledge in this domain.

Mahbuba Begum et al. in their review “**Digital Image Watermarking Techniques: A review**” **Information 2020, 11, 110; doi:10.3390/info11020110**” stated about the use of the DWT, DCT, DFT, and LSB algorithms has been used to analyze watermarking in medical photographs. This study's objective is to compare the outcomes of several methods used to calculate PSNR and SR values on medical pictures. Using DCT, DWT, LSB, and DFT approaches, logo pictures are injected values on medical pictures. Using DCT, DWT, LSB, and DFT approaches, logo pictures are injected into medical imaging (MR) in this study, and several attacks are connected to the resulting images. The three MR pictures utilized are distinct. The DFT approach was determined to have the highest PSNR (48, 1430). DCT, DFT, and LSB values are essentially the same. Their PSNR scores are in the 35 range. DFT SR values are the poorest when comparing the SR values of the DCT, DFT, and DWT approaches. embedding utilizing frequency in MR images as a result.

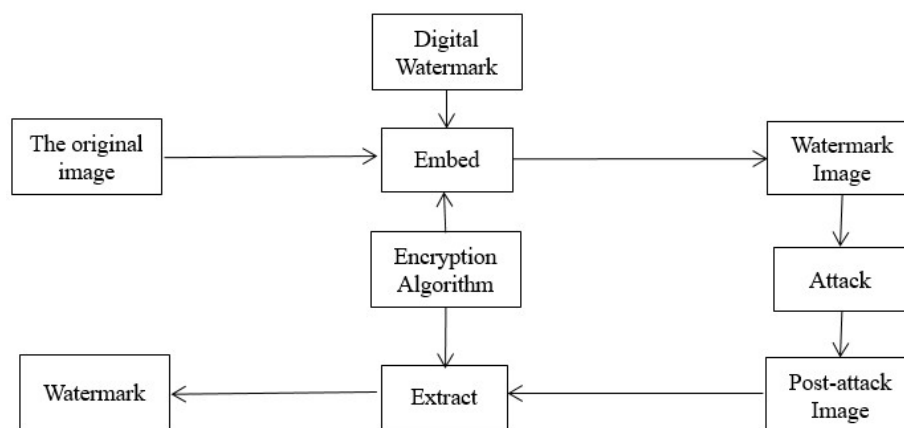
Yashovardhan Kelkar et al. proposed an algorithm on “**Analysis of Robustness of Digital Watermarking techniques under various attacks**”. In this paper they proposed three algorithms based on Least Significant Bit(LSB), Discrete Cosine Transformation(DCT) and Discrete Wavelet Transformation and then compared their results with one another based on the PSNR values and MSE values. LSB algorithm is straight forward and embeds the watermark image in the LSB plain of the original image. DCT algorithm uses the concept of transforming the cover image into three frequency sub-bands and inserting the watermark in the mid frequency range. DWT splits the original image into four frequency halves LL, LH, HL, HH and inserting the watermark in the LH and HL regions. Experimental results show high PSNR and MSE values of watermarked images under different attacks for DWT transformation compared to LSB and DCT transformations.

Swanirbhar Majumder et al. presents a robust and imperceptible algorithm on “**DWT and SVD based image watermarking scheme using Noise Visibility and Contrast Sensitivity**”. The robustness is brought about by hiding the watermark in the Eigen values after computing the Singular Value Decomposition(SVD) on low frequency sub-bands after DWT of the original image. While for imperceptibility the Contrast Sensitivity Function(CSF) along with Noise Visibility Function(NVF) is used here. In this paper DWT and SVD mainly takes care of the robustness part and NVF and CSF takes care of the imperceptibility part. The main novelty of the paper lies in the fact that here the SVD has been used with CDS9/7 wavelets, NVF and CSF. Also experimental results on the Lena image show good outcome and high PSNR value of nearly 41 and NC of 0.998 has been reported under different attacks.

6. ABOUT PROJECT

The project focuses on developing an image watermarking algorithm that embeds 32x32 bitmap watermark into images to protect their ownership, authenticity, or copyright. Watermarking involves adding a visible or invisible mark or pattern to the image that can serve as a proof of ownership or provide additional information about the image. By implementing robust watermarking techniques, the project aims to provide a reliable mechanism for protecting image assets from unauthorized use, distribution, or alteration.

Then an Image tampering algorithm is developed to tampered a watermarked image. Image tampering can involve activities such as image forgery, manipulation, splicing, or content alteration, which can potentially compromise the integrity and authenticity of the images. By implementing advanced techniques and algorithms, the project aims to provide a reliable mechanism for detecting tampered images and ensuring the integrity of visual data.



7. PROJECT ANALYSIS

● EXISTING WORK:

A strategy to embed 24 binary pictures in the DCT domain and 12 binary images in the spatial domain was described in 2010 by K. Ganesan and Tarun Kumar Guptha, one of the more recent studies in this topic. The spatial domain was replaced using LSB. Experimental findings, however, indicated that introducing six binary pictures in the spatial domain would have produced better outcomes than doing so with nine or twelve. A innovative method of picture watermarking based on the embedding of numerous watermarks in various areas of the image was proposed by SamehOueslati et al. in 2013. For the geographical domain, segmentation via fuzzy c-means clustering was applied. In the middle frequency band, DCT coefficients based on the quantization JPEG table were computed. The suggested method was proven to be resistant to various asynchronous assaults and jpeg compression since it incorporated numerous domains. In 2014, ShampaChakraverty et al. created a multiple watermarking system in which data was saved as a relational database with a number of properties. K denotes the employment of a clustering procedure to get the data. The method was discovered to be resistant to tuple addition and tuple alteration assaults, and in certain instances the full watermark was retrieved. Mehdi Alirezanejad and colleagues published a spatial domain watermarking method that year that made use of high-boost filtering to produce better-quality extracted images. The information in the watermark was enlarged by the filtering procedure. 2015 saw the proposal of a blind watermarking scheme by K. Karthik and Dr. M. A. DoraiRangaswamy.

● PROPOSED WORK:

In this Project, the proposed work focuses on embedding a 32x32 bitmap (.bmp) image into a 512x512 cover image using Discrete Cosine Transform (DCT) and Pseudo-Noise (PN) sequence. The goal is to achieve secure watermarking and develop tampering detection mechanisms along with noise analysis.

The embedding process starts by converting the original RGB image to the YCbCr color space, where the Y-channel is extracted for watermark embedding. A PN sequence is generated using a security key to ensure secure embedding. The Y-channel is permuted using the PN sequence, and then divided into non-overlapping 8x8 blocks. DCT is applied to each block, and 1024 randomly selected blocks are used to embed the watermark pixels with a strength of 0.1. Inverse DCT (IDCT) is applied to each block, and the blocks are combined to reconstruct the original dimensions. The permutation is then removed, and the RGB channels are combined to form the watermarked image.

For extraction, the algorithm retrieves the Y-channel, applies DCT to 8x8 blocks, and calculates the difference at the pixel locations where the watermark is embedded. Positive differences correspond to background pixels, while negative differences correspond to foreground pixels of the watermark.

The tampering detection process involves several steps. First, the original cover image is inputted at the client-side. The watermark image is hidden in the cover image using DCT and the PN sequence. Tampering is then applied to the stego image during transmission, incorporating noise. At the receiver's end, the tampered image is received, and the watermark image is extracted from the stego image.

To detect tampering and distortion in the extracted watermark image, several analysis techniques are applied. The differences between similar color pixel byte elements within specific sub-block images of the original and extracted watermarks are tracked. Ratios of counter variables related to these differences are computed based on a specific threshold value. The average values of all sub-block elements in the concerned sub-block images are also computed, and ratios of counter variables related to these average differences are evaluated. Additionally, the similarity aspect between the original and extracted watermark images is analyzed based on standard parameters. By comparing the quantified values and threshold values, the distortion and tampering effects on the extracted watermark image can be determined.

Overall, the proposed work aims to embed a bitmap image into a cover image using DCT and PN sequence, while also incorporating tampering detection mechanisms and noise analysis. The tampering detection process involves tracking differences, computing ratios, and evaluating distortion scenarios in the extracted watermark image.

● PURPOSE OF THE PROJECT:

1. Preprocessing:

- Convert the original RGB image to YCbCr color space.
- Extract the Y-channel from the YCbCr image.

2. Watermark Embedding:

- Generate a pseudo-random PN sequence using a security key.
- Permute the Y-channel using the PN sequence.
- Divide the permuted Y-channel into 8x8 non-overlapping blocks.
- Apply Discrete Cosine Transform (DCT) to each block.
- Select 1024 random blocks to embed the watermark pixels with a strength of 0.1.
- Apply Inverse DCT (IDCT) to each block.
- Combine the blocks to reconstruct the original dimensions.
- Remove the permutation.
- Combine the YCbCr channels to form the RGB watermarked image.

3. Watermark Extraction:

- Extract the Y-channel from the watermarked image.
- Divide the Y-channel into 8x8 blocks.
- Apply DCT to each block.
- Compute the difference at the pixel location where the watermark was embedded.
- Negative differences indicate foreground pixels, and positive differences indicate background pixels.

4. Tampering Detection:

- Client Side:
 - Input the original cover image.
 - Hide the watermark image in the cover image using DCT and PN sequence.
- Sender Side:
 - Apply tampering to the stego image.
 - Incorporate noise during transmission.
- Receiver Side:
 - Receive the tampered stego image.
 - Extract the watermark image from the stego image.
 - Compare the original watermark image and the extracted watermark image.

5. Tampering Detection Metrics:

- Difference Ratio:
 - Track the difference between similar color pixel byte elements within sub-block images.
 - Compute the ratio of two counter variables based on a specific threshold value.
- Average Ratio:
 - Compute the average value of all sub-block elements for the concerned sub-block images.
 - Compute the ratio of two counter variables based on a specific threshold value.
- Similarity and Distortion Analysis:
 - Compute the similarity aspect of the original and extracted watermark images.
 - Analyze the distortion aspect of the extracted watermark image based on a specific threshold value.

6. Evaluation:

- Analyze the quantified values from the difference ratio, average ratio, and similarity analysis.
- Determine the distortion and tampering effect in the extracted watermark image.

By following the proposed work, the 32x32 bitmap .bmp image can be embedded within the 512x512 cover image using DCT and PN sequence. The extraction process includes tampering detection and noise analysis to evaluate the distortion and tampering effect in the extracted watermark image.

8. SYSTEM REQUIREMENT

SOFTWARE REQUIREMENT:

- MATLAB editor R2023a
- IRFAN View
- Simulink

HARDWARE REQUIREMENT:

- OS: 32-bit or 64-bit: Vista, Win 7, Win 8 or above version
- Processor: Intel Core 2 Duo, 2.4 GHz / AMD Athlon X2, 2.4 GHz or above version
- Memory: At least 512 MB RAM
- Graphics: At least 512 MB virtual memory

9. PROPOSED ALGORITHM

WATERMARKING ALGORITHM:

The provided code implements a blind watermarking algorithm using the Discrete Cosine Transform (DCT). The algorithm takes an original image `x`, a watermark image, and a key as input, and it returns watermarked image `y`.

Here is a step-by-step explanation of the algorithm:

1. Convert the original image `x` from the RGB color space to the YCbCr color space using the `rgb2ycbcr` function.
2. Extract the individual channels (Y, Cb, and Cr) from the YCbCr image.
3. Generate a pseudo-random noise (PN) sequence for permutation. The key is used to initialize the random number generator (`rng`) to ensure reproducibility.
4. Perform permutation on the Y channel of the image. Each pixel in the Y channel is permuted based on the PN sequence as follows:
 - If the corresponding PN value is 1, the pixel value is replaced with the product of the PN value and the current pixel value, and the pixel value at the corresponding position (j, i) is replaced with the complement of the PN value multiplied by the pixel value at position (i, j).
 - If the PN value is 0, the pixel value remains unchanged.
5. Resize the watermark image to a target size (32x32) while maintaining the aspect ratio using the `imresize` function.
6. Convert the resized watermark image to binary using the `imbinarize` function.
7. Reshape the binary watermark image into a vector (`W_vec`) by concatenating its columns.
8. Divide the permuted Y channel into 8x8 blocks and apply the DCT (Discrete Cosine Transform) to each block. The resulting DCT coefficients are stored in the `blocks` cell array.

9. Select a random set of blocks from the 'blocks' array to embed the watermark. The selection is based on the key provided. The number of selected blocks is equal to the number of elements in the watermark vector.

10. Perform the embedding stage by modifying the DCT coefficient at the position (4, 2) in each selected block. If the corresponding watermark bit is 0, decrease the DCT coefficient by a small value ('alpha'). If the watermark bit is 1, increase the DCT coefficient by 'alpha'.

11. Combine the modified blocks back to their original dimensions and apply the inverse DCT (IDCT) to each block. The resulting blocks form the watermarked Y channel ('Y_watermarked').

12. Reverse the permutation applied in step 4 to retrieve the original Y channel. Each pixel in the Y channel is reversed according to the PN sequence as follows:

- If the corresponding PN value is 1, the pixel value is replaced with the product of the PN value and the original pixel value, and the pixel value at the corresponding position (j, i) is replaced with the complement of the PN value multiplied by the watermarked pixel value at position (i, j).

- If the PN value is 0, the pixel value remains unchanged.

13. Combine the modified Y channel with the original Cb and Cr channels to form the watermarked YCbCr image.

14. Convert the watermarked YCbCr image back to the RGB color space using the 'ycbcr2rgb' function to obtain the final watermarked image 'y'.

The algorithm aims to embed the watermark in a perceptually invisible manner by modifying the DCT coefficients of selected blocks in the Y channel while considering the pseudo random permutation to enhance security.

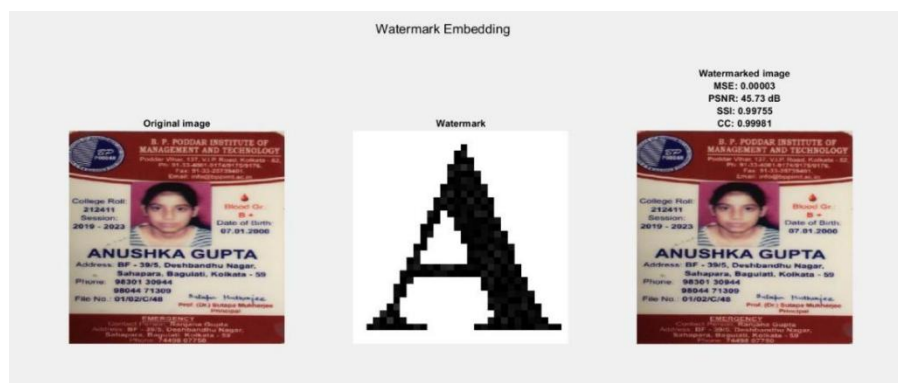


Fig : Watermark Embedding

EXTRACTION ALGORITHM:

The provided code implements the extraction stage of a blind watermarking algorithm that uses the Discrete Cosine Transform (DCT). The algorithm takes an original image `x`, a watermarked image `y`, and a key as input, and it returns the original image without the watermark (`x`) and the extracted watermark.

Here is a step-by-step explanation of the algorithm:

1. Convert both the original image `x` and the watermarked image `y` from the RGB color space to the YCbCr color space using the `rgb2ycbcr` function.
2. Extract the Y-channel from both the original and watermarked YCbCr images.
3. Divide both the original and watermarked Y-channels into 8x8 blocks and apply the DCT (Discrete Cosine Transform) to each block. The resulting DCT coefficients are stored in separate cell arrays `x_blocks` and `y_blocks`.
4. Calculate the number of blocks (`num_blocks`) based on the size of the Y-channel.
5. Select a random set of blocks from the `x_blocks` and `y_blocks` arrays to extract the watermark. The selection is based on the key provided. The number of selected blocks is fixed at 1024.
6. Perform the extraction stage by comparing the DCT coefficient at position (4, 2) in each selected watermarked block with the corresponding coefficient in the original block. If difference (`diff`) is negative, assign the corresponding watermark bit as 0. If the difference is positive, assign the watermark bit as 1.
7. Reshape the extracted watermark bits into a 32x32 matrix to reconstruct the watermark image.
8. Return the original image without the watermark (`x`) and the extracted watermark image.

The algorithm aims to extract the watermark by comparing the DCT coefficients of selected blocks in the watermarked image with the corresponding coefficients in the original image. The extracted watermark is then reconstructed as a binary image based on the differences between the coefficients.

OUTPUT:

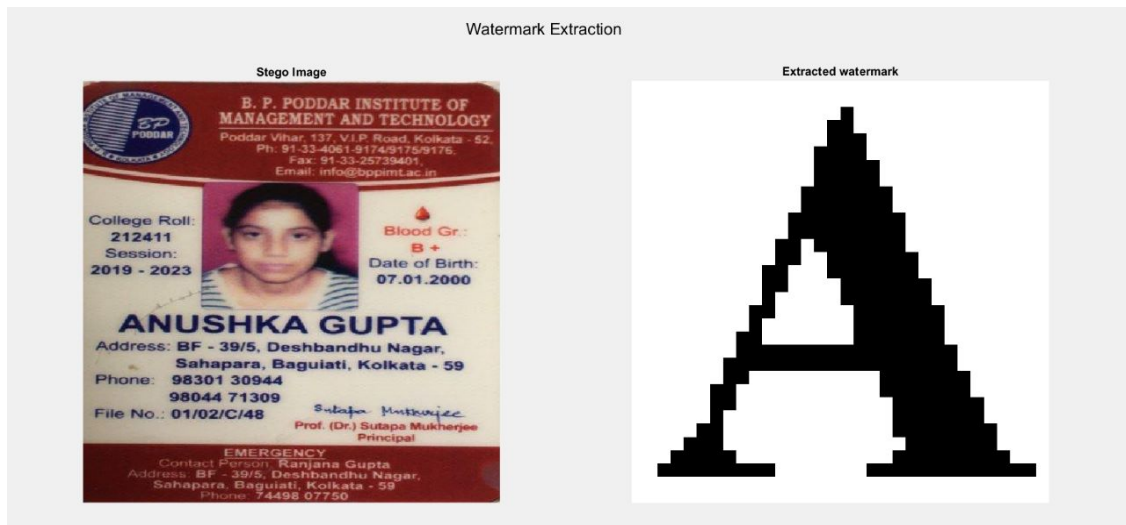


Fig2(a): Watermark extraction without tampering

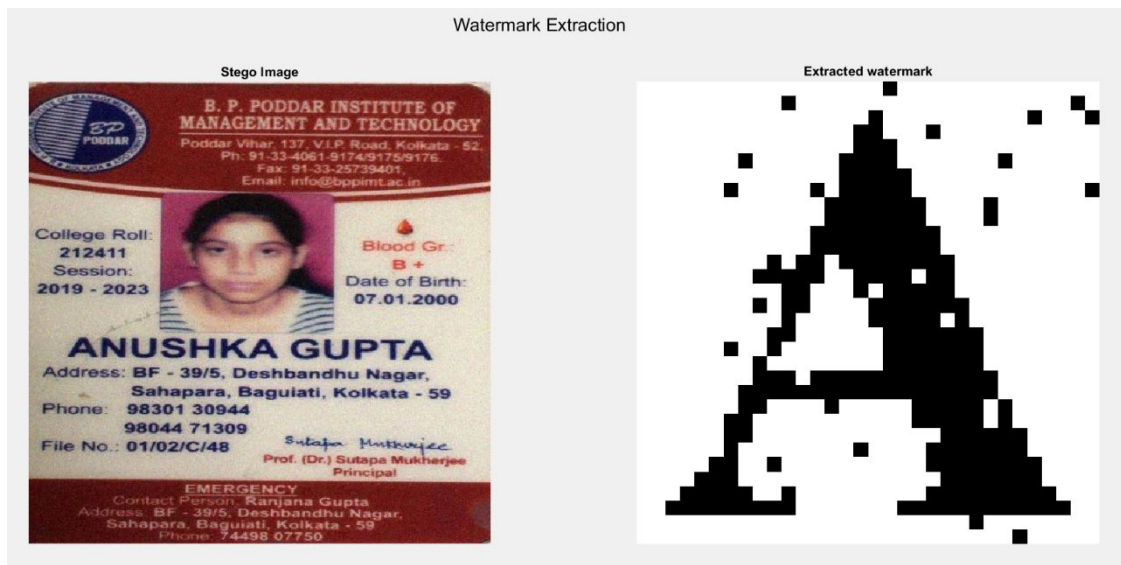


Fig 2(b): Watermark Extraction with tampering with 4% Gaussian Noise

TAMPERING DETECTION ALGORITHM:

Step 1: At the Client Side: Input the Original cover image.

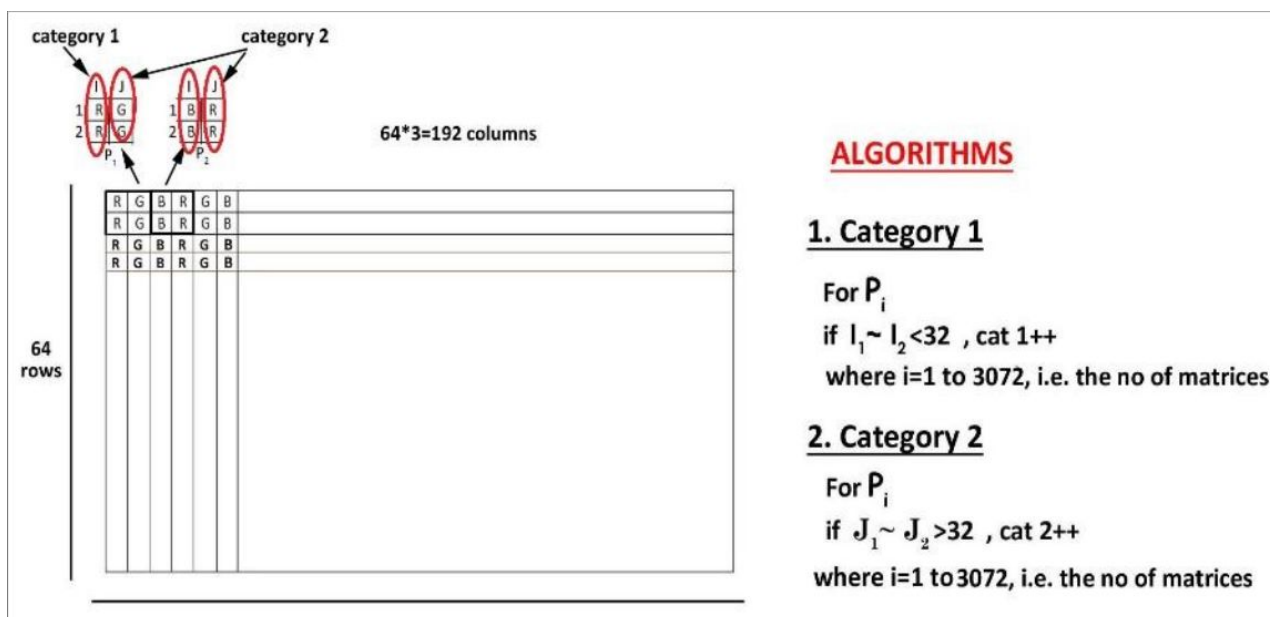
Step 2: Hide the watermark image in the cover image using DCT and PN sequence.

Step 3: Apply tampering on the Stego image and send it to the receiver.

Step 4: During transmission, noise is incorporated (tampering will occur).

Step 5: At the receiver, the image is tampered with noise is integrated and the receiver extracts the watermark image from the stego image.

Step 6: For specific watermark image: Track the difference between similar color pixel byte elements within the concerned sub-block image for both the original watermark and extracted watermark. Now, compute the ratio of two counter variables related to this difference based on a specific threshold value.



Step 7: For specific watermark image: Compute the average value of all the sub-block elements for the concerned sub-block image for both the original watermark and extracted watermark image. Now, compute the ratio of two counter variables related to this difference based on a specific threshold value.

Step 8: For specific watermark image: Compute the similarity aspect of both the original and extracted watermark image in terms of standard parameters and based on specific threshold value the distortion aspect of the extracted watermark image can be judged.

Step 9: Now, by evaluating the ratios found in Step 6, and Step 7 and the threshold value found in Step 8, the overall distortion scenario in the specific watermark image can be observed.

Hence, by analyzing the quantified values tracked from **Step 6**, **Step 7**, and **Step 8** the distortion cum tampering effect in the corresponding extracted watermark image can be decided.

10. SOURCE CODE

EMBEDDING CODE:

```
function y = embed(x, watermark, key)
% watermarking algorithm using Discrete Cosine Transform
% ARGUMENTS
% x: Original image to be watermarked
% watermark: Watermark image
% key: Key (any positive integer) for PN sequence generator
% RETURNS
% y: Watermarked image

% Convert image to YCbCr color space
x_YCbCr = rgb2ycbcr(x);

% Extract individual channels
Y = x_YCbCr(:,:,1);
Cb = x_YCbCr(:,:,2);
Cr = x_YCbCr(:,:,3);

% Generate PN sequence for permutation
rng(key);
PN = randi([0 1],size(Y,1));

% Permutation
I = zeros(size(Y));
for i = 1:size(Y,1)
    for j = 1:size(Y,2)
        if PN(i,j) == 1
            I(i,j) = PN(i,j)*Y(i,j) + imcomplement(PN(i,j))*Y(j,i);
        else
            I(i,j) = Y(i,j);
        end
    end
end

% Resize the watermark while maintaining the aspect ratio
targetSize = [32, 32];
watermark = imresize(watermark, targetSize);

% Resize the watermark, convert it to binary and finally, reshape to a vector
%if size(watermark) ~= [32, 32]
%watermark = imresize(watermark, [32, 32]);
%end
watermark = imbinarize(watermark);
W_vec = reshape(watermark, 1, numel(watermark));

% Transform Y-channel to 8x8 blocks and apply DCT
r = 1;
c = 1;
b_size = 8;
num_blocks = size(Y,1)*b_size;
for k = 1:num_blocks
    blocks{k} = dct2(I(c:c+b_size-1, r:r+b_size-1));
% Update blocks
    if r + b_size >= size(Y, 2)
        r = 1;
        c = c + b_size;
    else
        r = r + b_size;
    end
end
```

```

c = c + b_size;
else
r = r + b_size;
end
end

% Select a 1024-element random blocks to embed the watermark
rng(key);
n = randperm(numel(blocks),numel(W_vec));

% Embedding stage
alpha = 0.1;
for k = 1 : numel(watermark)
if W_vec(k) == 0
blocks{n(k)}(4,2) = blocks{n(k)}(4,2) - alpha;
else
blocks{n(k)}(4,2) = blocks{n(k)}(4,2) + alpha;
end
end

% Combine the blocks back to original dimensions and apply IDCT
r = 1;
c = 1;
for k = 1 : numel(blocks)
Y_watermarked(c:c+b_size-1, r:r+b_size-1) = idct2(blocks{k});
% Update blocks
if r + b_size >= size(Y, 2)
r = 1;
c = c + b_size;
else
r = r + b_size;
end
end

% Remove permutation
I = zeros(size(Y));
for i = 1:size(Y,1)
for j = 1:size(Y,2)
if PN(i,j) == 1
I(i,j) = PN(i,j)*Y(i,j) + imcomplement(PN(i,j))*Y_watermarked(i,j);
else
I(i,j) = Y_watermarked(i,j);
end
end
end

% Combine the channels and convert back to RGB map
y = cat(3, I, Cb, Cr);
y = ycbcr2rgb(y);

end

```

EXTRACTION CODE:

```
function watermark = extract(x, y, key)
% watermarking algorithm using Discrete Cosine Transform
% ARGUMENTS
% x: Original image
% y: Watermarked image
% key: Key (any positive integer) to decode PN sequence
% RETURNS
% y: Original image without watermark
% y: Extracted watermark

% Convert the images to YCbCr color space
x_YCbCr = rgb2ycbcr(x);
y_YCbCr = rgb2ycbcr(y);

% Extract Y-channel
x_Y = x_YCbCr(:,:,1);
y_Y = y_YCbCr(:,:,1);

% Transform Y-channel to 8x8 blocks and apply DCT
r = 1;
c = 1;
block_size = 8;
num_blocks = size(x_Y,1)*block_size;
for k = 1:num_blocks
x_blocks{k} = dct2(x_Y(c:c+block_size-1, r:r+block_size-1));
y_blocks{k} = dct2(y_Y(c:c+block_size-1, r:r+block_size-1));

% Update blocks
if r + block_size >= size(x_Y, 2)
r = 1;
c = c + block_size;
else
r = r + block_size;
end
end

% Get 1024 watermarked blocks
rng(key);
n = randperm(numel(x_blocks), 1024);

% Extraction stage
for k = 1 : 1024
diff(k) = y_blocks{n(k)}(4,2) - x_blocks{n(k)}(4,2);
if diff(k) < 0
watermark(k) = 0;
elseif diff(k) > 0
watermark(k) = 1;
end
end

% Convert to original dimensions
watermark = reshape(watermark, 32, 32);

end
```


TAMPERING DETECTION CODE:

```
% Input parameters
[filename, filepath] = uigetfile('*.jpg', 'Select the Original watermark image');
path = fullfile(filepath, filename);
originalImage = im2double(imread(path));
[filename, filepath] = uigetfile('*.jpg', 'Select the Extracted watermark image');
path = fullfile(filepath, filename);
extractedImage = im2double(imread(path));

threshold = 32;
cat1 = 0;
cat2 = 0;

for i = 1:size(originalImage, 1)-1
    for j = 1:size(originalImage, 2)-1
        a = originalImage(i, j);
        c = originalImage(i + 1, j);
        if abs(a - c) < threshold
            cat1 = cat1 + 1;
        end
    end
end

for k = 1:size(extractedImage, 1)-1
    for l = 1:size(extractedImage, 2)-1
        b = extractedImage(k, l);
        d = extractedImage(k + 1, l);
        if abs(b - d) < threshold
            cat2 = cat2 + 1;
        end
    end
end

% Evaluation metrics
MSE = immse(extractedImage, originalImage);
PSNR = psnr(extractedImage, originalImage);
SSIM = ssim(extractedImage, originalImage);
CC = corr2(extractedImage, originalImage);

X1 = mean(originalImage(:));
X2 = mean(extractedImage(:));

similarity = abs(X1 - X2);
distortionThreshold = 0.1;
ratio=cat1/cat2;

ratio1 = cat1 / ((size(originalImage, 1)-1) * (size(originalImage, 2)-1));
ratio2 = cat2 / ((size(originalImage, 1)-1) * (size(originalImage, 2)-1));

%if ratio > 1 || similarity > distortionThreshold
%disp('Watermark tampering detected');
%else
%disp('No watermark tampering detected');
%end

figure('Name', 'Tampering Detection');
```

```

% Subplot 1
subplot(2,2,1);
imshow(originalImage);
title('Original Watermark');

% Subplot 2
subplot(2,2,2);
imshow(extractedImage);
subtitle = {'Extracted Watermark', ...
' (PSNR: ', num2str(PSNR, '%.2f'), ' dB)', ...
' [CC: ', num2str(CC, '%.5f')]'};
title(subtitle);

% Plotting the graph
x = 1:2; % x-axis values
y1 = [cat1 cat2]; % y-axis values for cat1 and cat2
y2 = [X1 X2]; % y-axis values for X1 and X2

subplot(2,2,3);
bar(x, y1);
xlabel('Category');
ylabel('Count');
title('Comparison of cat1 and cat2');
set(gca, 'XTickLabel', {'cat1', 'cat2'});

subplot(2,2,4);
bar(x, y2);
xlabel('Category');
ylabel('Count');
title('Comparison of X1 and X2');
set(gca, 'XTickLabel', {'X1', 'X2'});

% Perform tampering detection
difference = imabsdiff(originalImage, extractedImage);
tamperedPixels = sum(difference(:) > 0);

% Calculate noise percentage
totalPixels = numel(originalImage);
noisePercentage = (tamperedPixels / totalPixels) * 100;


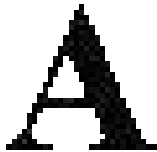


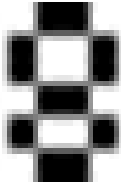

% Calculate tampering percentage
tamperingPercentage = (tamperedPixels / numel(extractedImage)) * 100;

% Display results
fprintf('Noise Percentage: %.2f%%\n', noisePercentage);
fprintf('Tampering Percentage: %.2f%%\n', tamperingPercentage);



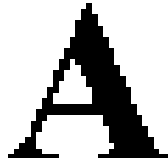


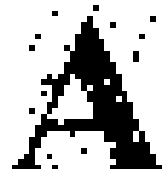



% Determine if tampering or noise is detected
if tamperingPercentage > 20
disp('Tampering detected.');
```

11. RESULT ANALYSIS

Embedding:

Cover Image	Secret Image	Stego Image	Performance (our Algorithm)	Normal Results
			MSME=0.00003 PSNR=45.73 db SSI=0.99755 CC=0.99981	PSNR<30
			MSME=0.00002 PSNR=46.48 db SSI=0.99663 CC=0.99989	PSNR<30

Tampering Analysis:

Noise	Stego-Image	Tampered Stego-Image	Extracted Secret Image	Tampering Analysis
No Noise				16.99%
4% Gaussian Noise				19.82%
10% Salt & Pepper noise				45.12%

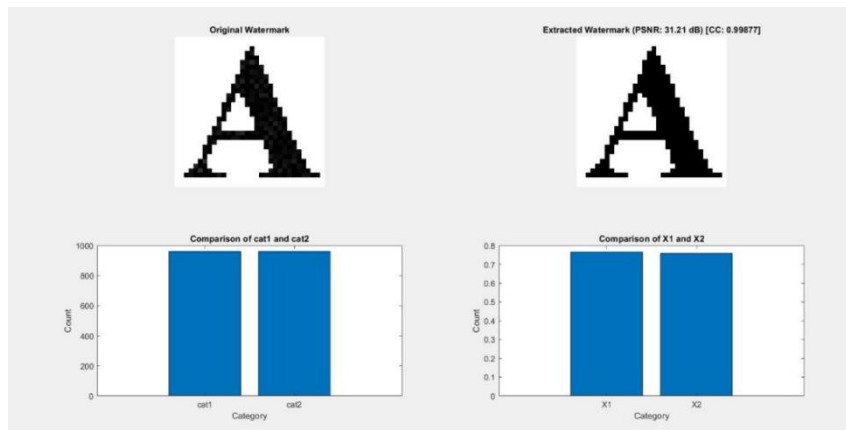


Fig: without tampering

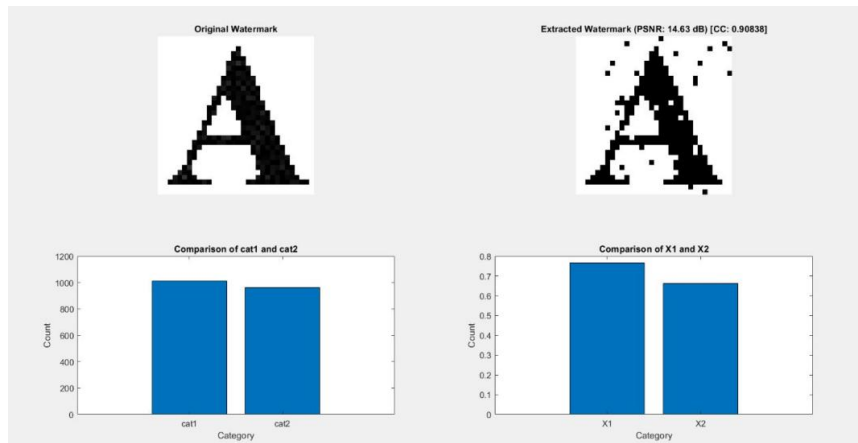


Fig: tampering with 4% Gaussian noise

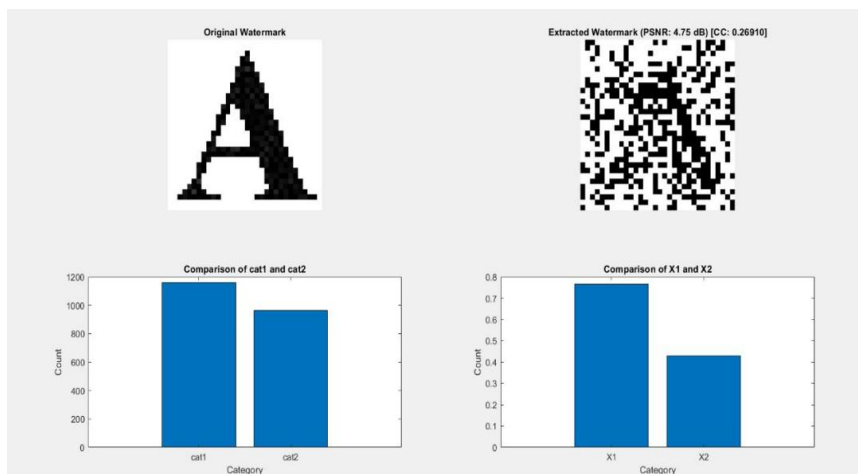


Fig: tampering with 10% salt and pepper noise

12. FEATURES

Encryption: Encrypting image data helps to ensure its confidentiality by converting it into unreadable format using encryption algorithms. Only authorized parties with the decryption key can access the original image.

Access Control: Implementing access controls ensures that only authorized individuals or systems can view, modify, or delete the image data. Access control mechanisms may include user authentication, role-based access control (RBAC), and permissions management.

Secure Storage: Storing image data in secure environments is crucial. This involves protecting the storage infrastructure against physical threats, implementing secure protocols for data transmission, and employing secure storage solutions with proper access controls.

Data Integrity: Maintaining data integrity ensures that the image data remains unaltered and free from unauthorized modifications. Techniques such as digital signatures, checksums, or hash functions can be used to verify the integrity of image data.

Backup and Disaster Recovery: Regularly backing up image data and having a robust disaster recovery plan in place are essential. Backups help in restoring image data in case of accidental deletion, data corruption, or system failures.

Audit Logs and Monitoring: Monitoring and logging activities related to image data access, modifications, and transfers provide a way to track and investigate any suspicious or unauthorized actions. Audit logs can assist in identifying potential security breaches or policy violations.

Secure Transmission: When transmitting image data over networks or sharing it externally, it is important to use secure protocols such as HTTPS or encrypted file transfer methods to prevent unauthorized interception or tampering.

Overall, the features of this project lie in its secure and robust watermark embedding technique, tampering detection capability, and the ability to assess the impact of noise on the watermark's integrity. These features make it suitable for applications where data confidentiality, authentication, and tampering detection are essential, such as digital rights management, copyright protection, and forensic analysis.

13. ADVANTAGES

- 1. Security:** The use of the PN sequence and security key ensures a high level of security in embedding the watermark. The pseudo-random sequence adds randomness and makes it difficult for unauthorized parties to extract or tamper with the embedded image.
- 2. High capacity:** By using DCT and selecting 1024 random blocks for embedding, a significant amount of data can be hidden within the cover image. This allows for a large amount of information to be stored while maintaining the visual quality of the cover image.
- 3. Robustness:** The DCT-based embedding technique provides robustness against various image processing operations such as compression, filtering, and scaling. The watermark remains intact even after common image manipulations, ensuring the integrity of the embedded information.
- 4. Tampering detection:** The tampering detection algorithm allows for the identification of any modifications made to the embedded image. By comparing the original watermark image with the extracted watermark image, any differences can be detected, indicating tampering or unauthorized modifications.
- 5. Noise analysis:** The incorporation of noise during transmission helps simulate real-world scenarios where images may be distorted or corrupted. By analyzing the noise-integrated image and comparing it with the extracted watermark, the impact of noise on the watermark's quality and integrity can be assessed.
- 6. Extraction accuracy:** The use of DCT and pixel difference computation enhances the accuracy of extracting the embedded watermark. By utilizing these techniques, the extraction process can precisely identify the foreground and background pixels of the watermark, ensuring a reliable retrieval of the hidden information.
- 7. Non-invasive:** The embedding process does not require any significant alteration to the cover image, preserving its original content and quality. The watermark can be added without perceptible changes to the visual appearance of the cover image, making it suitable for applications where image aesthetics are crucial.
- 8. Compatibility:** The embedding and extraction algorithms can be implemented on various image formats, including BMP, without requiring major modifications. This ensures compatibility with existing image processing systems and allows for seamless integration into different environments.

14. FUTURE SCOPE

1. **Enhancing Embedding Efficiency:** Explore techniques to improve the embedding efficiency by optimizing the number of blocks used for embedding and watermarking strength. This could involve adaptive selection of blocks based on their significance in the image or implementing intelligent algorithms for determining the optimal strength of watermarking.
2. **Robustness against Attacks:** Investigate the robustness of the proposed embedding and extraction algorithm against various attacks, such as compression, cropping, rotation, and filtering. Develop strategies to enhance the system's resilience to these attacks and evaluate its performance under different scenarios.
3. **Security Enhancement:** Evaluate and enhance the security aspects of the algorithm, such as key generation, PN sequence generation, and permutation techniques. Implement more sophisticated encryption and decryption mechanisms to ensure a higher level of security in the embedding and extraction process.
4. **Large-Scale Testing:** Conduct extensive testing and analysis on a larger dataset comprising diverse cover images and secret images. Evaluate the algorithm's performance, including its embedding capacity, extraction accuracy, and tampering detection capability, on a wide range of images to ensure its reliability and generalizability.
5. **Noise Analysis:** Perform in-depth noise analysis to understand the impact of different types and levels of noise on the embedding and extraction process. Investigate noise reduction techniques or propose methods to enhance the algorithm's robustness against noise, ensuring reliable extraction even in noisy environments.
6. **Integration with Existing Systems:** Explore the integration of the proposed algorithm with existing image processing and security systems. Investigate how the embedding and extraction process can be seamlessly incorporated into larger-scale applications, such as digital rights management, copyright protection, or data authentication systems.
7. **User Interface and User Experience:** Develop an intuitive and user-friendly interface for the embedding and extraction process, making it accessible to users with varying levels of technical expertise. Focus on improving the overall user experience, including ease of use, system feedback, and visual representations of the embedding and extraction results.

By addressing these future directions, the proposed embedding, tampering detection, and noise analysis algorithm can be further enhanced, leading to improved performance, security, and usability in various practical applications involving image watermarking and content protection.

15. CONCLUSION

In conclusion, this report presented a method for embedding a 32x32 bitmap image within a 512x512 cover image using Discrete Cosine Transform (DCT) and a Pseudo-Noise (PN) sequence. The embedding process involved converting the original image to the YCbCr color space, generating a PN sequence using a security key, and permuting the Y-channel of the cover image. DCT was applied to 8x8 blocks of the permuted frame, and a selected number of random blocks were used to embed the watermark pixels. The process was reversed to extract the embedded image from the cover image.

Additionally, tampering detection was incorporated into the algorithm to ensure the integrity of the embedded image. The tampering detection process involved tracking differences between the original watermark and the extracted watermark, computing ratios of these differences, and comparing them to predefined threshold values. The average values of sub-block elements in both images were also compared using counter variables and threshold values. Finally, the distortion and tampering effect in the extracted watermark image were evaluated based on the calculated ratios and thresholds.

The proposed method provides a secure and robust approach for embedding and extracting a secret image while detecting tampering and noise. By analyzing the quantified values obtained through the tampering detection process, the distortion and tampering effects in the extracted watermark image can be accurately determined. This technique holds promise for applications where data integrity and confidentiality are paramount, such as digital watermarking, copyright protection, and secure communication.

Only two algorithms have been used to calculate the pixel variations. In future more effective algorithms can be used to attain more accuracy and precision. Such algorithms could more efficiently detect even the smallest of tampering. That means we may be able to detect the position of manual tampering more precisely and can pin point the location. Also we can use more complex algorithm for encoding and decoding to increase the security and confirm authenticity.

16. REFERENCES

- [1]Deepika Sharma , Pawanesh Abrol, “Digital Image Tampering – A Threat to Security Management”, International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 10, October 2013
- [2]Minati Mishra , Flt. Lt. Dr. M. C. Adhikary , “Digital Image Tamper Detection Techniques - A Comprehensive Study” in International Journal of Computer Science and Business Informatics 2013
- [3]Arshad Jamal, Mohammed Hazim Alkawaz , Mariam-Aisha Fatima, Mohd Shukri Ab Yajid , “Digital Watermarking” Techniques And its Application Digital Halal Certificate: A Survey” 2019 IEEE on Systems, Process and Control (ICSPC 2019), 13–14 December 2019, Melaka, Malaysia.
- [4]Rajkumar Ramasamy and Vasuki Arumugam, ”Digital watermarking —A tutorial” in IEEE Conference 7 July 2022
- [5]Mahbuba Begum ,* and Mohammad Shorif Uddin, “Digital Image Watermarking Techniques: A review” Information 2020, 11, 110; doi:10.3390/info11020110
- [6]Ziyue Xiang and Daniel E Acuna "Scientific Image Tampering Detection Based On Noise Inconsistencies: A Method And Datasets"
- [7]A. Mohanarathinam, S. Kamalraj, G. K. D. Prasanna Venkatesan, Renjith V. Ravi, C. S. Manikandababu , “Digital watermarking techniques for image security: a review” Received: 9 April 2019 / Accepted: 12 September 2019 / Published online: 20 September 2019 © Springer-Verlag GmbH Germany, part of Springer Nature 2019
- [8]Drian Morales-Ortega and Manuel Cedillo-Hernandez , “Ownership Authentication and Tamper Detection in Digital Images via Zero-Watermarking” 2022 45th International Conference on Telecommunications and Signal Processing (TSP) | 978-1-6654-6948-7/22/\$31.00 ©2022 IEEE | DOI: 10.1109/TSP55681.2022.9851253
- [9]Olkan Kaya, Ersin Elbasi, “Robust Medical Image Watermarking Using Frequency Domain and Least Significant Bits Algorithms” in IEEE Conference 2022
- [10]Xiaoqiang zhang and Xuesong wang, (November 2018), “Digital Image Encryption Algorithm Based on Elliptic Curve Public Cryptosystem.” IEEE Access, vol.6
- [11]A Study on Image Forgery Detection Techniques Shijo Easowa*, Dr. L. C. Manikandanb
- [12]A. Makandar and B. Halalli, (2015), “A review on preprocessing techniques for digital mammography images.” International Journal of computer applications
- [13]Parameswaran L, Anbumani K. Content-based watermarking for image authentication using independent component analysis. INFORMATICA.2008;32(3).
- [14]Sun L, Xu J, Zhang X, Tian Y. An Image Watermarking Scheme Using Arnold Transform and Fuzzy Smooth Support Vector Machine. MATHPROBL ENG. 2015 Oct 11;2015.
- [15]Saeed, F., Zahedan, M. G., & Azimi, M. A Blind Watermarking Algorithm Based On DCT-DWT And Arnold Transform. Department Of Telecommunications, IJCSE, 2(06).