Digital Image Steganography: A Frequency Domain Approach

Shreyas Wankhede
ai21btech11028@iith.ac.in

Abhishek Kumar
ai21btech11003@iith.ac.in

Pradeep Mundlik
ai21btech11022@iith.ac.in

Sathvika Marri
ai21btech11020@iith.ac.in

Jacky Meena
ee21btech11025@iith.ac.in

Abstract

This project delves into digital image steganography, a vital component of secure communication in today's digital landscape. Steganography conceals sensitive information within seemingly innocuous cover media, offering a covert alternative to encryption. Focusing on the frequency domain approach, mainly using techniques like the Discrete Fourier Transform (DFT), this study explores embedding secret data within digital images. By manipulating the frequency components, imperceptible alterations are made to the digital images, ensuring the confidentiality and integrity of the hidden information. The paper provides a comprehensive overview of digital image steganography, discussing its methodologies and applications.

1. Introduction

In today's digital age, the importance of secure communication and data transmission cannot be overstated. With the proliferation of digital platforms and the vast exchange of information over networks, ensuring the confidentiality and integrity of sensitive data has become paramount. This has led to the emergence of various techniques and methodologies aimed at concealing information within digital media, one of which is steganography.

Steganography, derived from the Greek words "steganos" (meaning covered or concealed) and "graphein" (meaning writing or drawing), is the art and science of hiding secret information within innocuous cover media in such a way that the existence of the hidden message remains undetectable to unintended recipients. Unlike cryptography, which focuses on encrypting the content of a message to render it unintelligible to unauthorized parties, steganography ensures that the existence of the hidden information itself is not apparent.

Steganography techniques can be broadly categorized into various types based on the cover media, such as text, audio, video, and images. In this project, we specifically focus on digital image steganography, leveraging the frequency domain approach facilitated by techniques like Discrete Fourier Transform (DFT).

Digital image steganography involves embedding secret data into digital images, exploiting the imperceptible alterations that can be made to the pixel values or the frequency components of the image. By utilizing transformations like DFT, which represent images in terms of their frequency components, we can embed information in a manner that is robust against typical image processing operations while minimizing perceptual distortion.

By employing the frequency domain approach, we aim to achieve efficient and imperceptible hiding of information within images, thereby ensuring both the secrecy and integrity of the concealed data.

2. Problem Statement

Developing a robust digital image steganography method that enhances the imperceptibility and capacity of hidden information within images. By leveraging the Fourier Transform, this technique will seek to embed data without compromising the image's visual integrity, ensuring that the embedded message remains undetectable to both visual and algorithmic detection methods while maintaining resistance to common image processing operations.

3. Literature Review

Old popular data hiding techniques have generally revolved around modifying or patternizing the least significant bit (LSB) of the pixel values of an image. Modifying low bit planes of an image often overwrites visual structures that exist to some degree in all of the image's bit layers. Hidden messages within the low bit planes of an image can potentially be deciphered visually. Thus, stego systems modifying the LSB often exhibit vulnerability to visual detection methods. The importance of Fourier Transformation in data hiding techniques has been seen rarely in the literature. In early work, researchers have used it by encoding the hidden message in the coefficients of the DFT magnitude. This

paper presents a similar idea, overcoming the shortcomings of old techniques by exploiting the spectral properties of Fourier Transformation. It improves three different aspects: capacity, 3-layer security, and robustness of hiding data.

An important fact that the paper builds on is that as long as the Fourier phase of an image remains intact, the original appearance remains intact with very minute visible artifacts if the Fourier magnitude of the image is slightly modified. Experimental results show that, in general, the hidden message may be as large as half the size of the carrier image for an unnoticeable amount of noise artifacts in the stego image.

Experimental evidence has established that the human visual system is much more sensitive to brightness than to color. This can be reasoned out because high-frequency information, i.e., fine details and edges, etc., mainly come from the brightness/luminance channels of an image. Most importantly, we use the color and brightness separation strategy and avoid altering the luminance channel. Experiments have revealed that altering the chrominance channels of an image has very few visible artifacts (almost not detectable by the human eye). Thus, we exploit not only the spectral properties of the image but also the chrominance and achromatic (luminance) properties of the image. In total, we embed the hidden message into the Fourier magnitude spectrum of the chrominance channels of the image.

Without significantly altering the visual quality of the image, the paper presents a technique of transforming the image into the L*a*b* space nonlinearly, following the color/brightness separation strategy. This transformation specifies the color of an image in a way that is independent of the characteristics of any particular imaging device. Let's call the image I=(R,G,B). The nonlinear operation converts the tuple (R,G,B) into the tuple (L,a,b), where L is the luminance channel and a and b are the two chrominance channels. Thus, this technique has almost double the hidden data carrying capacity than previous techniques. Now, we take the DFT of the chrominance channels a and b to embed the secret data. We separate the magnitude and phases in each of the chromatic channels. Most specifically, we embed the hidden message by replacing high-frequency areas of the chrominance channels of the Fourier magnitude of the chrominance channels. This type of embedding prevents aliasing of the hidden message when extracted. Aliasing refers to a phenomenon where high-frequency components in a signal are incorrectly represented or distorted in a reconstructed version of the signal due to insufficient sampling. In our case, high-frequency components appear as mirroring of parts of the hidden image from one side onto the opposite side and thus cause data loss. We partition the secret information into channel a and the rest in channel b. We hide the secret message through a for-loop which later acts as a security key for recovering the message. Without knowing the correct embedding loop, it becomes a guessing game for successfully recovering the hidden information. After modifying the Fourier magnitude of the a and b chrominance channels, we then combine them with their corresponding phases. We then apply the inverse FFT to the modified DFTs of the a and b channels to get the modified color/brightness separated image tuple (L,a',b'), where a' is the modified chromatic channel a and b' is the modified chromatic channel b. Then we transform (L,a',b') to the modified image tuple (R',G',B'). Here we get our stego image (image carrying the hidden information) S=(R',G',B').

Now let's look at how to recover the hidden information. The extraction of the secret message is just the process that is the reverse of hiding it. First, we convert (R',G',B') to (L,a',b') by the nonlinear transformation that was used during the hiding phase. Then we take the DFT of the chromatic channels a' and b'. Finally, we apply the security key for-loop to recover the hidden message. The first part of the hidden message is extracted from the high-frequency areas in the Fourier chrominance-a magnitude spectrum, and the second hidden image is extracted from the high-frequency areas in the Fourier chrominance-b magnitude spectrum, making sure that the for-loop used to extract the hidden images is the same for-loop that was used to embed these hidden images. Thus, we encode and decode the secret message, compromising the minor color artifacts in the stego image. The interesting fact about this technique is that it provides a 3-layered security measure because the resultant image has the effect of scattering the hidden image (info) three times across all pixels of the carrier image. The first scattering is when we multiply the modified chromatic channel Fourier magnitude with its unmodified phase. The second scattering is when we do the inverse FFT of the DFT of the chromatic channel, and the third scattering is when we convert (L,a',b') to (R',G',B'). This scattering leads to robustness to stego medium tampering. The most powerful fact about this technique is that because of the above scattering, as long as 40 or more of the stego image data remains intact, we can extract the hidden message image with reasonable integrity. The paper then demonstrates the robustness of this image-hiding technique to tampering (manipulating) degradations in the stego image by simulating different tampering effects like cutting parts of the image, repainting the image, rotating the stego image, etc., and still being able to recover the message in a legitimate, understandable amount.

In general, a technique qualifies as a steganography technique broadly based on:

1. Transparency: How much information you can hide in the cover image (stego hidden) without distorting the original appearance of the image much so that it is undetectable by any visual attacks.
2. Robustness/Tamper resistance: The ability of the hidden

message to remain undamaged even if the stego image undergoes some sort of transformation like filtering (linear and blurring), blurring, cropping, repainting.

We can see clearly that the technique mentioned in the paper satisfies the above properties. In total, the paper presents a very secure, high-capacity, tamper-resistant technique to hide some secrecy exploiting the color/brightness of the image and spectral properties of the Fourier-transformed image.

4. Methodologies

4.1. LSB matching

Least Significant Bit (LSB) matching is a steganographic technique used to embed hidden information within the least significant bits of pixel values in an image. The least significant bit is the rightmost bit in a binary representation of a number and holds the smallest value. By modifying the LSB of pixel values, one can encode hidden information with minimal impact on the visual appearance of the image.

LSB Modification

In the context of an image, each pixel's color intensity is typically represented by an 8-bit binary number, ranging from 0 to 255. The LSB of each pixel value is the bit that contributes the least to the pixel's overall value, making it an ideal target for embedding hidden data without significantly altering the image.

Consider an 8-bit pixel value $P = b_7b_6b_5b_4b_3b_2b_1b_0$, where b_7 is the Most Significant Bit (MSB) and b_0 is the Least Significant Bit (LSB). Modifying the LSB (b_0) changes the pixel value by ± 1, while modifying a higher-order bit b_k ($k > 0$) changes it by $\pm 2^k$. The percentage change due to modifying b_k is $\frac{2^k}{P} \times 100\%$, which increases exponentially with k. Hence, modifying the LSB results in the minimal percentage change compared to any other bit.

Consider an 8-bit image where each pixel value, P, can range from 0 to 255. The Least Significant Bit (LSB) of a pixel value can be either 0 or 1, corresponding to the decimal values 0 and 1, respectively.

Let P' be the pixel value after modifying the LSB. The change in pixel value is either an increment of 1 (if the original LSB was 0) or a decrement of 1 (if the original LSB was 1). Therefore, the absolute difference $|P' - P|$ is always 1.

The percentage change in pixel value can be calculated as:

$$\text{Percentage Change} = \frac{|P' - P|}{P} \times 100\%$$

Since $|P' - P| = 1$, the percentage change simplifies to:

$$\text{Percentage Change} = \frac{1}{P} \times 100\%$$

For an 8-bit image:

- The minimum non-zero pixel value is $P = 1$, resulting in:

$$\text{Percentage Change} = \frac{1}{1} \times 100\% = 100\%$$

- The maximum pixel value is $P = 255$, resulting in:

$$\text{Percentage Change} = \frac{1}{255} \times 100\% \approx 0.39\%$$

Therefore, the percentage change in pixel value due to modifying the LSB ranges from 100% for the smallest non-zero pixel value to approximately 0.39% for the largest pixel value. This demonstrates that the impact of modifying the LSB on the pixel value is minimal, especially for larger pixel values.

LSB Matching for Image Embedding

The LSB matching technique extends the basic LSB modification process to embed a sequence of bits (representing the hidden information) across multiple pixels in an image. The process involves the following steps:

1. For each pixel value of hidden image (information):
   (a) Convert the pixel value to its 8-bit binary representation.
   (b) Every bit of the 8 bit binary representation will be embedded in the one unique pixel of the original image.
   (c) If the bit matches with the LSB of the pixel value, the pixel value is unchanged .
   (d) If the LSB does not match, increment or decrement the pixel's value by 1. If the pixel's value is at its maximum (255), decrement it; if it's at its minimum (0), increment it.
2. Repeat the process for each pixel in the hidden Image (information) until all bits are embedded.
3. The resulting image is the stego image, which contains the hidden information which is very similar visually to the original image (no observable difference).

This technique ensures that the changes to the pixel values are minimal, preserving the overall appearance of the image while successfully embedding the hidden information. However, the method imposes certain constraints on the ratio of hidden image size and the original image.The original image size should be atleast 8 times the size of hidden image .

Retrieving Embedded Information

To extract the hidden information from an image that has undergone LSB matching:

Figure 1. Original Image

Figure 2. Stego Image

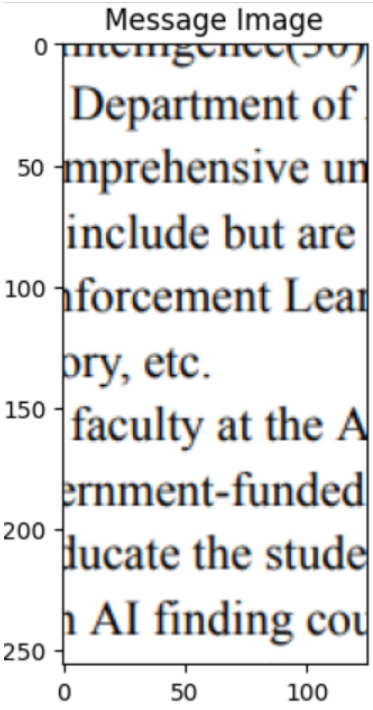

Figure 3. Hidden Image

1. By observing the difference of stego image and original image, we can figure out the pixel loactions where LSB has been modified.
2. Accordingly the difference image will be remodified after comparing with the LSB of the original image at the same pixel locations
3. Finally concatenate the modified bits in a binary sequence of 8 bits and convert this back to decimal pixel value.
4. Store these pixel values sequentially in an array of appropriate dimension.
5. These pixel values will correspond to the hidden image pixels and thus the hidden message will be recovered.

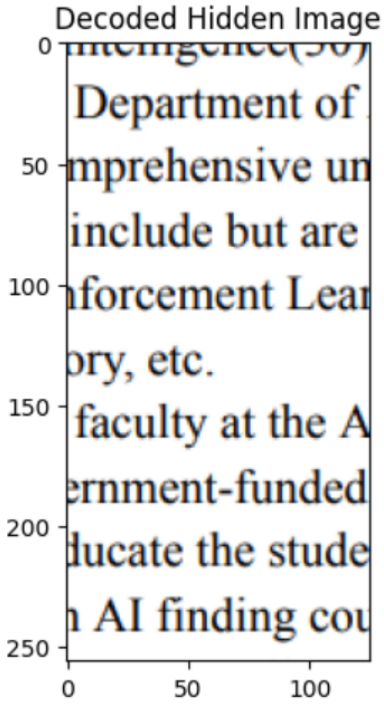

Figure 4. Recovered Hidden Image

By following this process, one can retrieve the embedded information with minimal error, provided that the stego image has not been significantly altered after the embedding process.

4.2. DFT Steganography

Let's consider three images A, B, and C , where A is the original image, B is an image created using the phase information of the Discrete Fourier Transform (DFT) of A, and C is an image created using the magnitude information of the DFT of A

Given an image I, we compute its Discrete Fourier Transform (DFT) to obtain the magnitude M_a and the phase angle P_a:

$$(M_a, P_a) = \text{DFT}(I)$$

Now, keeping the phase P_a intact, we create an image B by performing the Inverse Discrete Fourier Transform (IDFT) using a matrix U of all ones and the original phase P_a:

$$B = \text{IDFT}(U, P_a)$$

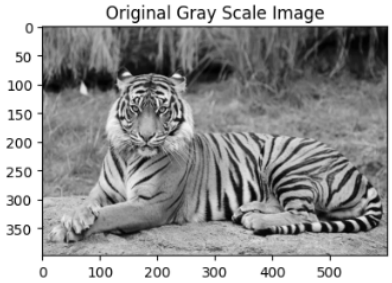

Figure 5. Original Image

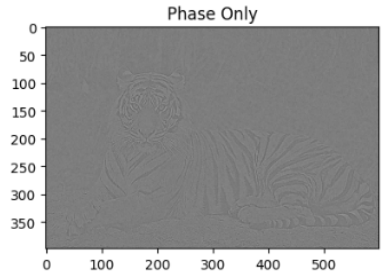

Figure 6. Phase Image

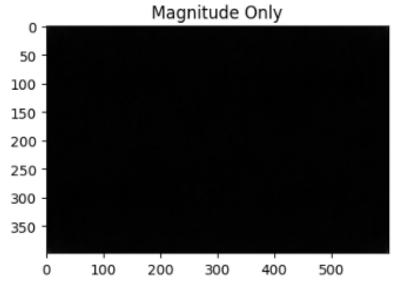

Figure 7. Magnitude Image

where U is a matrix with all entries equal to 1, defined as:

$$U_{m\times n} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

We also create an image C by performing the IDFT using the original magnitude M_a and a null matrix O:

$$C = \text{IDFT}(M_a, O)$$

where O is a matrix with all entries equal to 0.

We observe that image B is similar to the original image A, while image C is not even close to A.

Based on this observation, we can devise a message hiding technique that utilizes the phase information of the DFT to embed a message into an image while maintaining its visual appearance.

Message Hiding Technique with High Capacity

The message hiding technique has the capacity to hide information content that can be up to 50% of the information content of the carrier (host) image (RGB).

Encoding

Figure 8. RGB Image

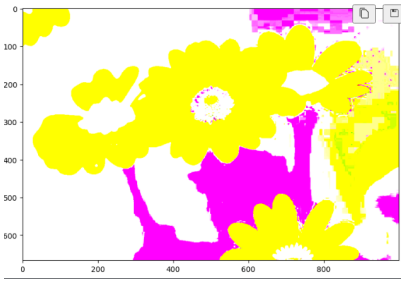

Figure 9. L*a*b* Image

Perceptual experimental evidence has established that the human visual system has a much higher sensitivity to changes in brightness than to color. Therefore, we adopt the following chrominance-luminance separation strategy:

$$(R, G, B) \rightarrow (L, a, b)$$

The $L^*a^*b^*$ color space is a nonlinear transformation of RGB space that specifies color in terms of human perception in a way that is independent of the characteristics of any particular imaging device.

CIE standards, based on experiments related to human perception and receptors, present the conversion (map) as follows:

In case of 8-bit and 16-bit images, R, G, and B are converted to the floating-point format and scaled to fit the 0 to 1 range. The conversion is performed as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \leftarrow \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$X \leftarrow \frac{X}{X_n}, \text{ where } X_n = 0.950456$$

$$Z \leftarrow \frac{Z}{Z_n}, \text{ where } Z_n = 1.088754$$

$$L^* = \begin{cases} 116 \cdot f(Y/Y_n)^{1/3} - 16 & \text{for } Y > 0.008856 \\ 903.3 \cdot Y & \text{for } Y \leq 0.008856 \end{cases}$$

$$a^* \leftarrow 500(f(X) - f(Y)) + \delta$$

$$b^* \leftarrow 200(f(Y) - f(Z)) + \delta$$

where

$$f(t) = \begin{cases} t^{1/3} & \text{for } t > 0.008856 \\ 7.787t + \frac{16}{116} & \text{for } t \leq 0.008856 \end{cases}$$

and

$$\delta = \begin{cases} 128 & \text{for 8-bit images} \\ 0 & \text{for floating-point images} \end{cases}$$

This outputs $0 \leq L^* \leq 100$, $-127 \leq a^* \leq 127$, $-127 \leq b^* \leq 127$. The values are then converted to the destination data type:

- 8-bit images: $L^* \leftarrow L^*/255 \cdot 100$, $a^* \leftarrow a^* + 128$, $b^* \leftarrow b^* + 128$
- 16-bit images: (currently not supported)
- 32-bit images: L^*, a^*, and b^* are left as is

Proving the Transformation as a Bijective Map

Let's denote the matrix used in the linear transformation (matrix-vector product) as T. We establish that T is invertible because the determinant of T is nonzero (specifically, $\det(T) = 0.2070563$), which implies that the mapping is invertible.

Given that X_n and Z_n are greater than zero, the inverse operations for division by X_n and Z_n are simply the multiplication by X_n and Z_n respectively. Thus, these operations are bijective.

Furthermore, the thresholds ($Y > 0.008856$ and $t > 0.008856$) divide the real line into mutually exclusive and exhaustive sets ($L > 7.99959199306$ and $f(t) > 0.20689303442$). Consequently, the conversions from $Y \to L$ and $t \to f(t)$ are invertible and hence establish a bijective mapping.

With the above procedures, we derive (L, a, b), where L indicates brightness, and the axes a and b represent the green-red and blue-yellow opponent colors, respectively.

Message Hiding Technique

Given the L*a*b* components, we calculate the Discrete Fourier Transforms (DFTs) of the chrominance channels:

$$M_a, P_a = \text{DFT}(a), \quad M_b, P_b = \text{DFT}(b)$$

where M_a, M_b are the magnitudes and P_a, P_b are the phase angles.

We retain L, P_a, and P_b unchanged. For the security key, we randomly sample locations from the chrominance channels a and b with a uniform distribution. The magnitudes M_a and M_b at these locations are replaced with the values of our message, embedding the first part into the chrominance a and the second part into the chrominance b. Additional metadata about the message dimensions is also included in the key.

The modified magnitudes M'_a and M'_b lead us to the modified channels a' and b' as follows:

$$a' = \text{IDFT}(M'_a \cdot \exp(1j \cdot P_a))$$

$$b' = \text{IDFT}(M'_b \cdot \exp(1j \cdot P_b))$$

where $\cdot$ denotes element-wise multiplication.

Thus, we obtain the components (L, a', b') after the message embedding process. The luminance component L remains unchanged, while a' and b' are the modified chrominance channels that now contain the hidden message.

Conversion from Lab to RGB

The conversion from L*a*b* to RGB is performed as follows:

$$y = \begin{cases} \left(\frac{L^*+16}{116}\right)^3, & \text{if } \left(\frac{L^*+16}{116}\right)^3 > 0.008856 \\ \frac{L^*}{903.3}, & \text{otherwise} \end{cases}$$

$$\Delta = 128$$

$$f_Y = \frac{a^* - \Delta}{500} + f_Y$$

$$f_Z = \frac{b^* + \Delta}{200} + f_Y$$

$$f_Y = \begin{cases} y^{1/3}, & \text{if } y > 0.008856 \\ 7.787y + \frac{16}{116}, & \text{if } y \leq 0.008856 \end{cases}$$

$$x = \begin{cases} (f_X)^3, & \text{if } f_X > 0.2068930342 \\ \left(\frac{f_X - 0.1379310345}{7.787}\right), & \text{if } f_X \leq 0.2068930342 \end{cases}$$

$$z = \begin{cases} (f_Z)^3, & \text{if } f_Z > 0.2068930342 \\ \left(\frac{f_Z - 0.1379310345}{7.787}\right), & \text{if } f_Z \leq 0.2068930342 \end{cases}$$

Now, compute the inverse transformation from XYZ to RGB

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Upon completing the conversion process, we obtain the stego image represented by the color components (R', G', B').

Message Extraction

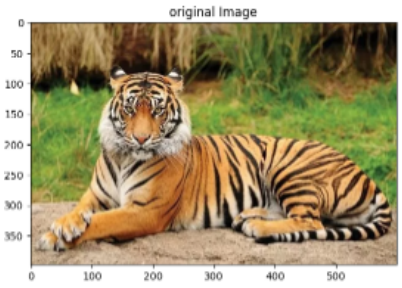

Figure 10. Original Image

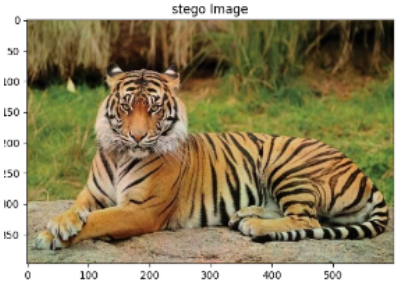

Figure 11. Stego Image

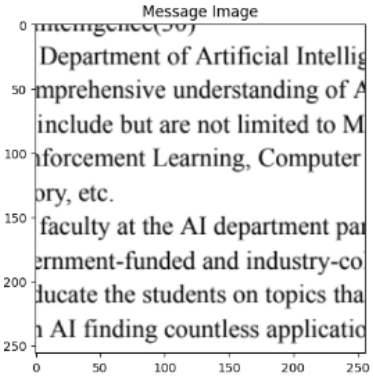

Figure 12. Hidden Message

Decoding the Hidden Message

The decoding process involves reversing the steps taken during encoding to retrieve the hidden message. Starting with the stego image with components (R', G', B'), we first convert it back into the $L^*a^*b^*$ color space:

$$(L, a', b') = \text{RGB to LAB}(R', G', B')$$

Next, we perform the Discrete Fourier Transform on the chrominance channels a' and b':

$$M'_a, P'_a = \text{DFT}(a'), \quad M'_b, P'_b = \text{DFT}(b')$$

Using the security key obtained from the for loop during the encoding process, we sample the same locations in the magnitude spectra M'_a and M'_b to recover the portions of the hidden message embedded within.

Finally, the recovered portions are assembled to reconstruct the hidden image:

$$\text{Recovered Image} = \text{Assemble}(M'_a, M'_b)$$

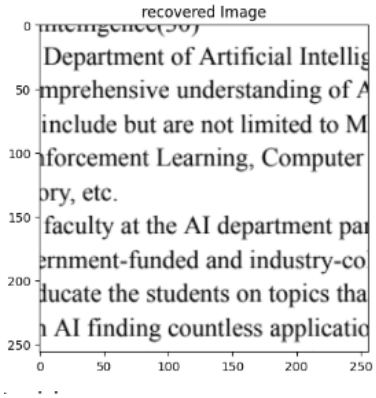

Figure 13. Recovered Hidden Image

4.3. Precision

A critical factor in making the above process practically feasible was dealing with the precision of floating-point numbers. To ensure accuracy and maintain the integrity of the hidden information throughout the transformations, we utilized the highest precision data types available for floating-point numbers at every step.

5. Future Works

1. **Transparency:** Enhancing the imperceptibility of the hidden message within the host image to ensure it remains undetectable to the human eye.
2. **Robustness:** Improving the resilience of the embedded message against various image processing operations and attacks.
3. **Capacity:** Increasing the amount of information that can be securely hidden within the host image without compromising its quality.
4. **Security:** Strengthening the security measures to protect the hidden message from unauthorized access and detection.

5. **Efficiency:** Implementing the Discrete Fourier Transform using the Fast Fourier Transform (FFT) algorithm to optimize the computational efficiency of the process and aslo trying different techniques like DCT, DWT etc.

References

[1] Sos S. Agaian, Khaled M. Mstafa, and Khaled Elleithy. Frequency domain approach of image steganography. https://www.researchgate.net/publication/309425281_Frequency_Domain_Approach_of_Image_Steganography, 2016.

[2] CIE Commission. Cie homepage. http://www.cie.co.at, 2021.

[3] Danielle Lerch. Image steganography with lsb matching. https://daniellerch.me/image-stego-lsbm/, 2021.

[4] OpenCV. Color conversions. https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html, 2021.

[5] Stack Overflow. Convert lab color to rgb. https://stackoverflow.com/questions/7880264/convert-lab-color-to-rgb, 2011.

[6] Tamer Rabie. Digital image steganography: An fft approach. https://www.researchgate.net/publication/269705199_Digital_Image_Steganography_An_FFT_Approach, 2018.

[7] Wikipedia. Cielab color space. https://en.wikipedia.org/wiki/CIELAB_color_space, 2021.

[8] Wikipedia. Discrete fourier transform. https://en.wikipedia.org/wiki/Discrete_Fourier_transform, 2021.

[9] Wikipedia. Fast fourier transform. https://en.wikipedia.org/wiki/Fast_Fourier_transform, 2021.

[10] Wikipedia. Steganography. https://en.wikipedia.org/wiki/Steganography, 2021.

[11] YouTube. Digital image steganography. https://www.youtube.com/watch?v=u_luy52v7z4, 2021.