Image Steganography using Wavelet Transform

JV Aditya (12140840),G Hemanth(12140660), N Childvias (12141120)

ABSTRACT

This project employs domain transfer methods, specifically utilizing the Wavelet Transform, for image steganography. Our experimental analysis demonstrates an enhanced tradeoff between capacity and quality. Various image quality parameters are evaluated to validate the observed improvements. The technique is tested on a larger dataset, and the results are compiled for evaluation. Additionally, a user-friendly graphical user interface (GUI) has been integrated into the steganography system to enhance ease of use. Users will be able to provide custom input and encode the payload image into a cover image using different hyperparameters.

WAVELET TRANSFORM

The Wavelet Transform is a method employed to convert the spatial domain into the frequency domain, effectively separating the high-frequency and low-frequency components of an image. In this approach, the filter bank algorithm is typically utilized. Initially, the image undergoes convolution with both a High Pass Filter and a Low Pass Filter. This convolution yields the high-pass and low-pass components pixel by pixel. Subsequently, each of these low-frequency and high-frequency components undergoes another round of convolution, this time with Low Pass Filters and High Pass Filters. This process ultimately yields the 2D Wavelet Transform of the image. In this project we have implemented Image steganography using 2D discreate Wavelet Transform.

IMAGE STEGANOGRAPHY

The algorithm that we have used to HCSSD encoder is as follows:

1. **Pre-Processing:**
   - Image input is provided, and each color component is separated.
   - Normalize each RGB component.
   - Apply Discrete Wavelet Transform (DWT) on these normalized components, completing the pre-processing.
2. **Wavelet Fusion:**
   - Pre-process both the Cover image and Payload image.
   - Merge these two images using a hyperparameter multiplied with the payload image to conceal it within the cover image.
3. **Compute Inverse DWT (IDWT):**

- Compute the IDWT for all the fused Red, Blue, and Green components.

4. **Combine Components:**
   - Combine the resulting Red, Blue, and Green components to form the fused color image.

Similarly, the algorithm that we have used to HCSSD decoder is as follows:

1. **Separate Components:**
   - Separate Red, Blue, and Green components of both the cover and stego color images.
2. **Normalize Components:**
   - Normalize all three components in both cases.
3. **Preprocessing:**
   - Preprocessing is done on both groups of image components.
4. **Compute 2D DWT:**
   - Compute 2D Discrete Wavelet Transform (DWT) of RGB components of the cover and stego images using Haar Wavelet.
5. **Subtract DWT Coefficients:**
   - Subtract the DWT coefficients of the corresponding components (RGB) of the stego and cover images.
6. **Compute IDWT:**
   - Compute IDWT of all the subtracted Red, Blue, and Green components.
7. **Combine Components:**
   - Combine the Red, Blue, and Green components obtained to form the fused color image.

Metrics for Stego Image Quality

Signal to Noise Ratio (SNR):

- Measures the quality of the stego image.
- $SNR = 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_\varepsilon^2}\right)$, where σ_x is the mean square value of the input image, and σ_ε is the mean square difference between the cover and stego images.
- Higher SNR values indicate better stego image quality.

Peak Signal-to-Noise Ratio (PSNR):

- Quantifies image quality based on Mean Square Error (MSE).
- $MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$, where I is the input image, K is the stego image, m and n are the number of pixels.
- $PSNR = 10\log_{10}\left(\frac{\text{MAX}_I^2}{MSE}\right)$, where MAX_I is the maximum pixel value of the image.
- Higher PSNR values indicate better image quality, with typical values between 30 and 50 dB.

Weighted Peak Signal-To-Noise Ratio (WPSNR):

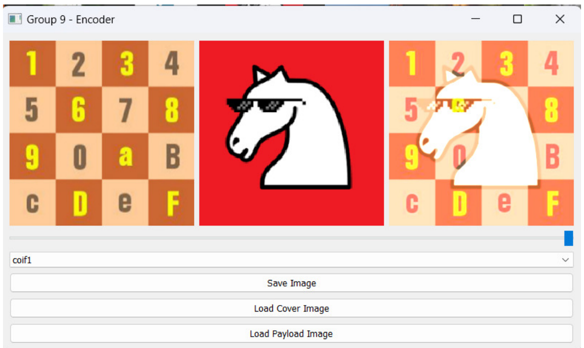
- Adjusts PSNR by considering perceptual differences between blocks using Normalized Video Fidelity (NVF).
- $WPSNR = 20\log_{10}\left(\frac{255}{\sqrt{\text{MSE}\times\text{NVF}}}\right)$.
- $NVF = \text{NORM}\left(\frac{1}{1+i_{\text{block}}^2}\right)$, where i_{block} is a simplified function.
- Higher WPSNR values indicate improved stego image quality.

Structural Content (SC):

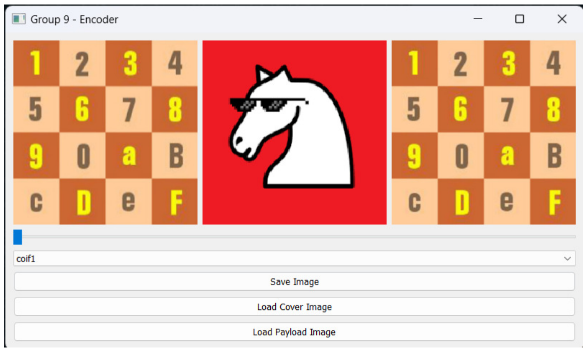
- Measures the similarity content between two images (cover and stego).
- Formulated as $SSIM(x,y) = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)}$, where μ_x, μ_y, σ_{xy}, σ_x^2, and σ_y^2 represent averages and covariances, and c_1, c_2 are constants.
- Default values: $c_1 = (0.01L)^2$, $c_2 = (0.03L)^2$, L is the dynamic range of pixel values.

GUI

The graphical user interface (GUI) is a pivotal aspect of our project, providing an intuitive and user-friendly platform for users to effortlessly discover the optimal alpha value. The interface simplifies the process by incorporating a scale, allowing users to conveniently adjust the alpha value. This dynamic feature not only enhances user interaction but also facilitates the exploration of different alpha values, enabling users to find the most effective setting for their specific needs. The GUI's intuitive design contributes significantly to the accessibility and usability of our project, empowering users to navigate and optimize the steganography process seamlessly.



(a) GUI Example with high alpha



(b) GUI example with low alpha

Figure 1: Histogram of the Cover image.

Conclusion

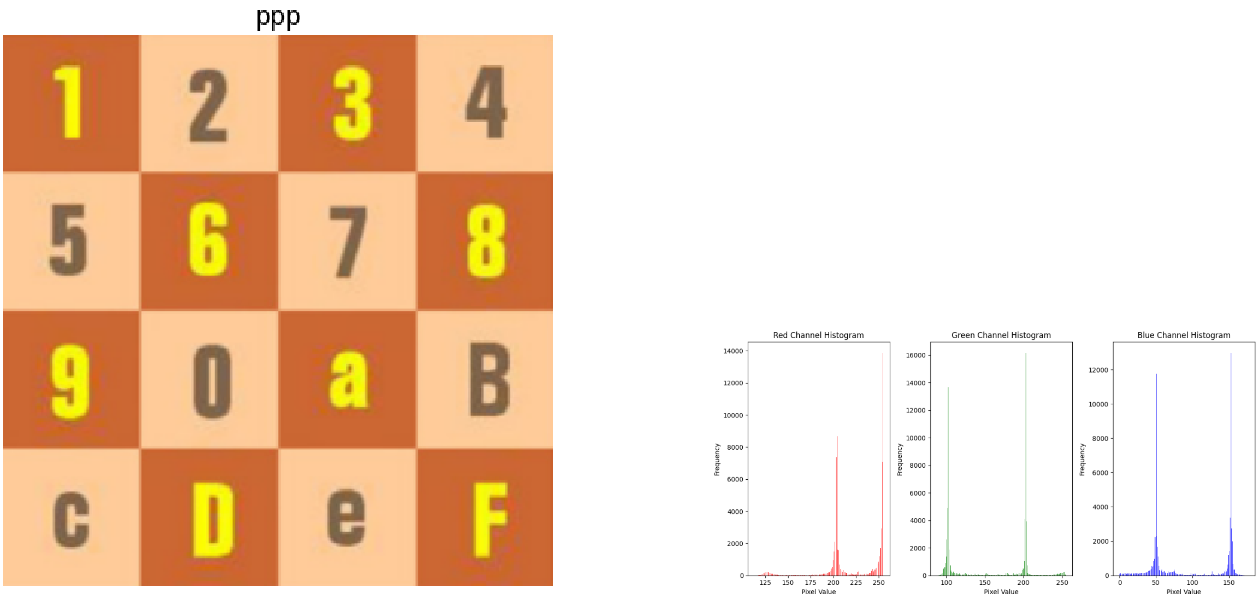
In summar, our project successfully hides information in images using a technique called steganography, where we transfer data between different image domains, mainly using something called the Wavelet Transform. We use a specific kind of this technique, the 2D discrete Wavelet Transform, in both the encoding and decoding steps, showing that it works well. We've made it user-friendly with a graphical interface to make it easy for people to use. Our tests, especially with the Haar wavelet, suggest that our method creates stego images with better quality, as seen in higher PSNR and WPSNR values. We've used various measures like SNR, PSNR, WPSNR, and SC to thoroughly evaluate our system's performance. In a nutshell, our project provides a strong foundation for image steganography, balancing the amount of hidden information with the quality of the resulting image.

RESULTS

We are taking the value of $\alpha = 0.2$ and plotting the cover image 2a and stego image 3a. From these images we can clearly see that the difference in between the RGB components of the payload and stego image, the stego image has similar distribution as cover image. The image quality using different wavelet's continues the trend as given in the research paper.The Haar wavelet has performed better than compared to other wavelets.

Table 1: Image quality parameters for $\alpha = 0.2$.

Wavelet	MSE	PSNR (dB)	WPSNR (dB)	SC
db1	798.5621	17.4856	18.2023	0.9064
coif1	674.2314	18.8538	18.3535	0.9054
sym2	760.8475	19.7234	19.8444	0.9019
Dmey	800.8471	19.7556	20.2048	0.8733
Haar	584.2612	20.4647	21.1343	0.9710

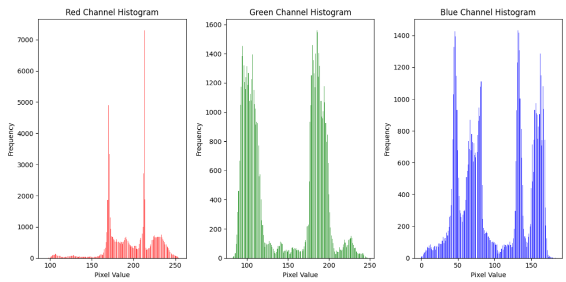


(a) Cover image $\alpha = 0.2$.

(b) RGB distribution of the cover image.

Figure 2: Histogram of the Cover image.

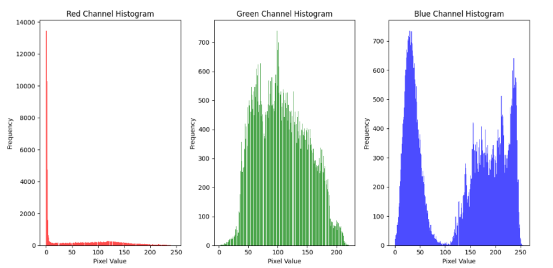
(a) Stego image $\alpha = 0.2$.



(b) RGB distribution of the Stego image.

Figure 3: Histogram of the stego image.

(a) Payload image.



(b) RGB distribution of the Payload image.

Figure 4: Histogram of the payload image.