Security & Forensic Lab-3 Report

Sunday, 21.02.2021

P Rahul Bharadwaj, 17CS02003

Problem Statement

To embed a watermark image in the original image by designing a new frequency domain algorithm that will ensure that the visual quality of the career image remains good in terms of PSNR after the watermarking.

Approach:

In digital watermarking, the watermark is an identification code that carries information about the copyright owner, inventor of the work, legal customer, etc, the cover image is decomposed into different frequency sub bands using wavelet decomposition(DWT) and block based DCT is applied.The logo of watermark is embedded on the high and middle {HL, LH, HH} frequency sub bands of the wavelet transformed cover Image.

Security is ensured with a private key such that water mark coefficients are manipulated with the random values generated by the secret key. Also the embedded watermark is scrambled at the start by arnold transform and recovered by using inverse arnold transform.

Software Requirements

- **open-cv**
- **Python**
- **Numpy**
- **Scipy**
- **pywavelet**

Watermark embedding

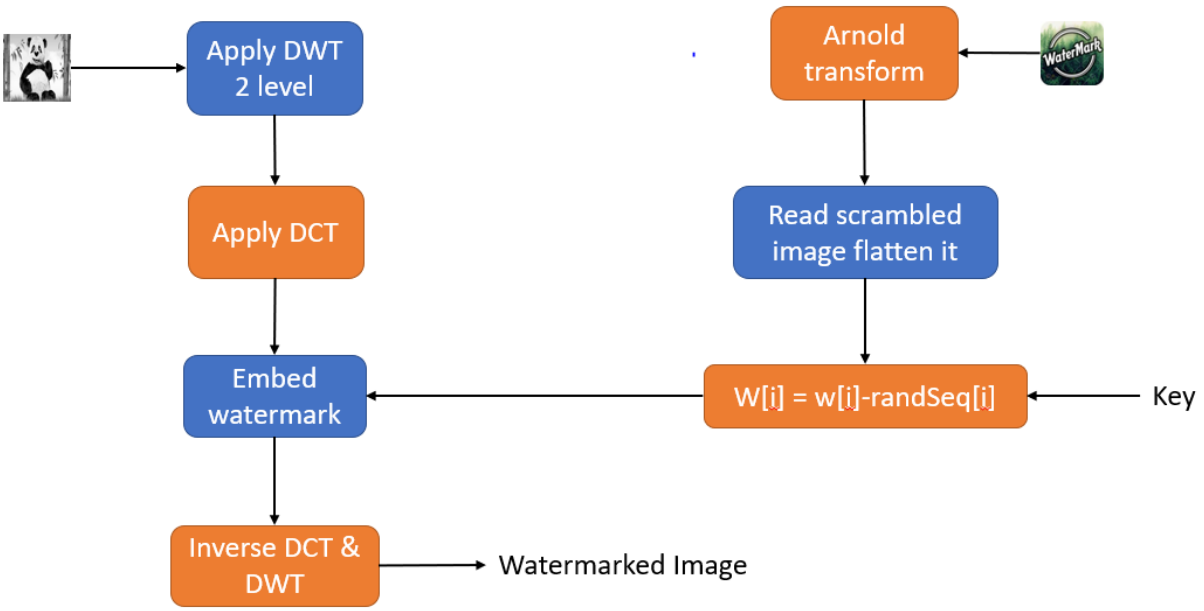


Fig.1 Water Embedding Algorithm

At the start a coverImage is taken and a discrete wavelet transform is performed on it, i.e 2D-DWT is applied on grayscale image. It transforms an image into sub-bands such that the wavelet coefficients in the lower level sub-bands typically contain more energy than those in higher level sub-bands. The first stage of the DWT divides an image into four sub-bands by applying low-pass and high pass filters.For the second level of decomposition, DWT further divides the lowest sub-band using the same filtering method as above. The Discrete Wavelet Transform analyzes the signal at different frequency bands with different resolutions by decomposing the signal into an approximation and detailed information. And then dct is applied on these bands, the watermark image is scrambled using Arnold transform and normalised. By providing a key as seed to a random generator we manipulate watermark image coefficients and embed them in blocks of cover image. After that we apply inverse dct and inverse (2-level) dwt to reconstruct the image.

Finally we write the reconstructed image in to a file.

Extracting Watermark

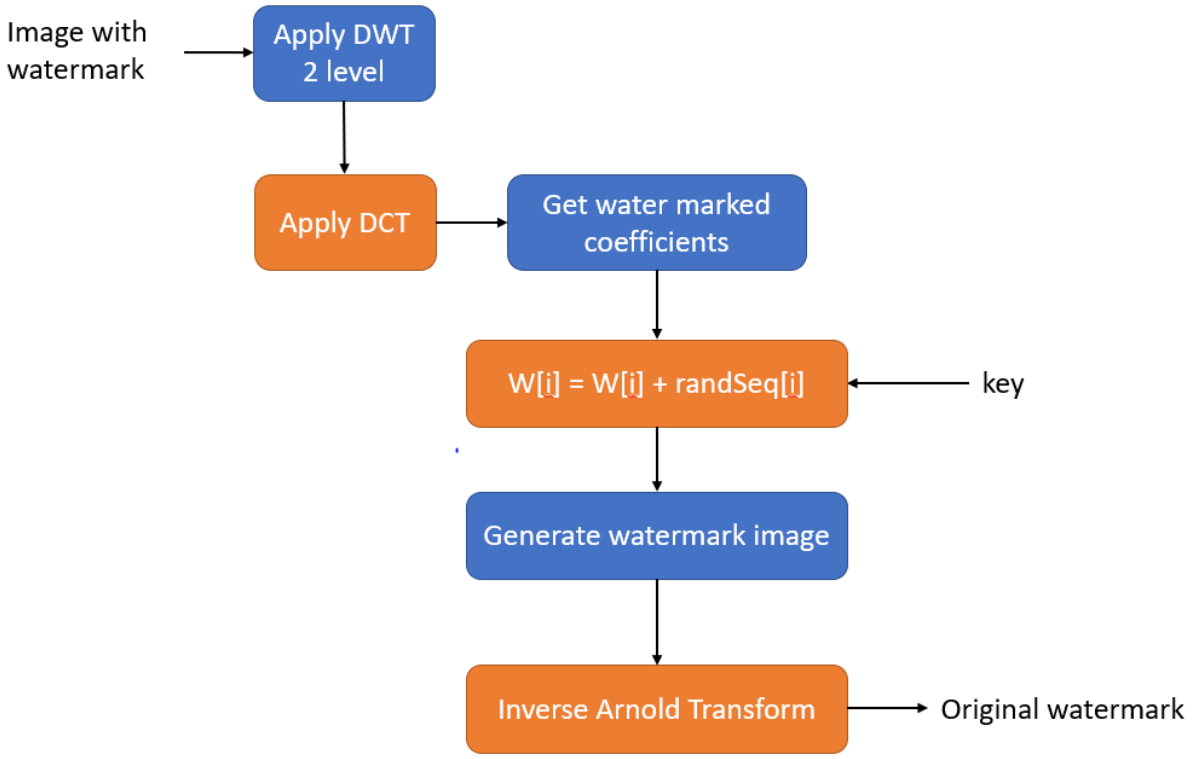


Fig.2 Watermark extraction Algorithm

To recover the watermark from the watermarked image, we first apply 2 level dwt as we have done while embedding and then perform dct on the image with watermark to obtain the watermark coefficients. Then with the help of a secret key we generate a random sequence of numbers and manipulate the watermark coefficients to get back our watermark image. Then we reconstruct the watermark image by these coefficients. After this the extracted watermark undergoes inverse arnold transform to recover the original watermark.

Arnold Transform:

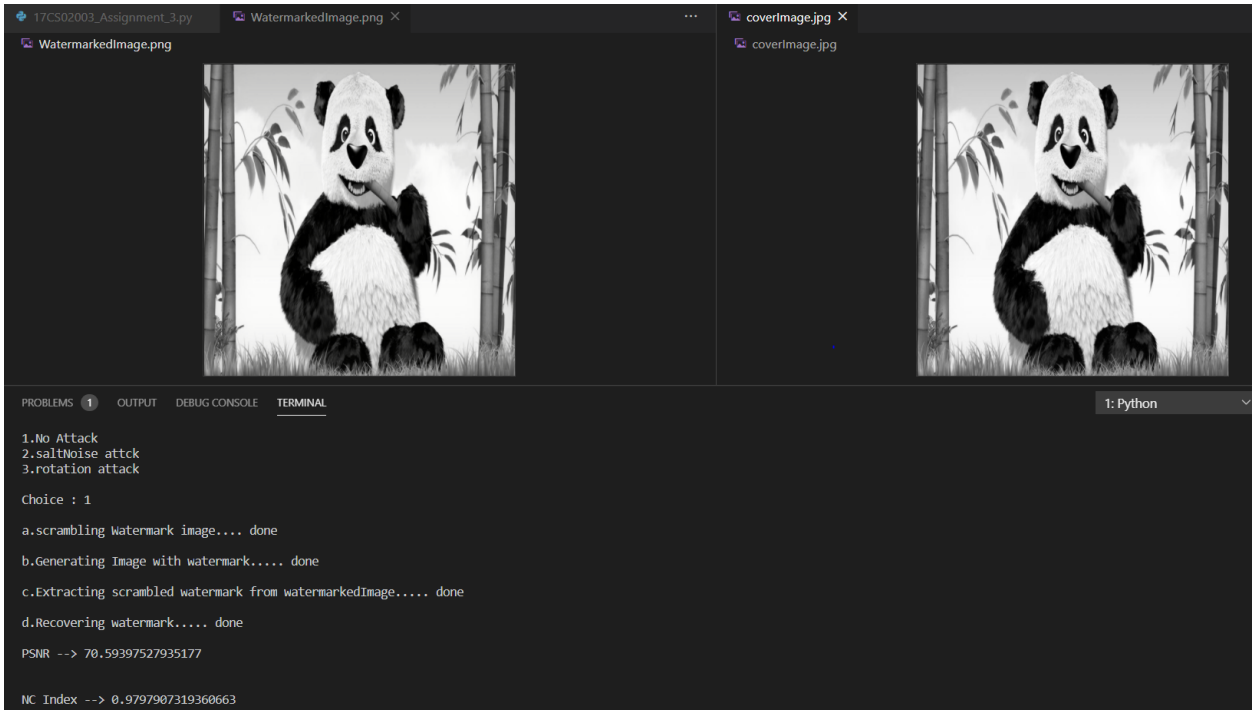
$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & b \\ a & ab+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} mod(N)$$

Inverse Arnold Transform:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} ab+1 & -b \\ -a & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} mod(N)$$

The two-dimensional Arnold transformation and inverse transformation of an M*N digital image is defined as above. For simplicity I have chosen a=b=1.

Output:



And the quality of embedding is measured by computing psnr score.

Quality of watermark is measured by normalised cross correlation index (NC Index).

Cover Image & Watermark Image

Fig.3 Cover Image

Fig.4 Image with Watermark

Fig.5 watermark

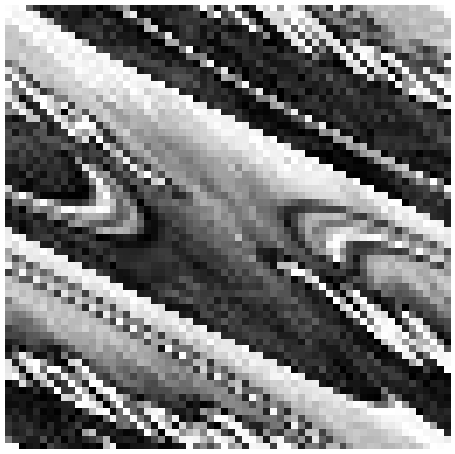


Fig.6 Scrambled watermark (Arnold)

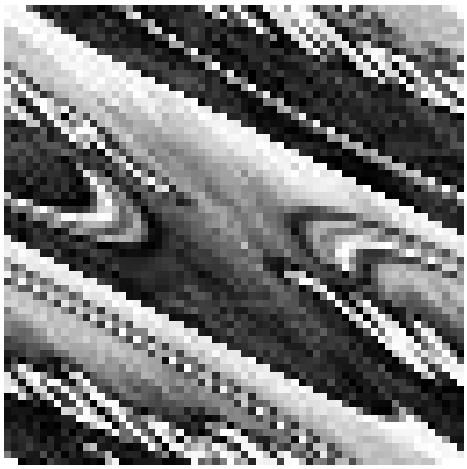


Fig.7 extracted watermark

Fig.6 recovered watermark (Arnold)

References

- DCT (How it works?)
- DWT (How it works ?)
- Robust and Invisible Image Watermarking in RGB Color space using SVD by D.Vaishnavia,* , T.S.Subashinib