SF Lab - 2

Assignment - 3

Watermark Using DCT based method

Anuj M. Choure
19CS02010

Algorithm:

DCT based watermarking algorithm has been used.

1. First we calculate the block size which is suitable for hiding the watermark in the image.
2. The block size and a random number will make the key. The key will be the output of the program, which will be used while extraction of the watermark.
3. We use a pseudorandom number generator to generate the indices of the block where the watermark info should be hidden.
4. The watermark image is flattened and converted to binary string. Each of these bits are stored in one block.
5. On each of the index of the block, we perform DCT and on the (0,0) element we perform following operation:
    a. If the current watermark bit is 0 then 10s place of the DCT coefficient will be rounded to the nearest even number.
    b. If the bit is 1 then 10s place of the DCT coefficient will be rounded to the nearest odd number.
    c. Everytime the one's place will be made 5, so that after IDCT rounding of the integer will not change the watermark data which is stored in the tens place of DCT coefficient.
    d. After that IDCT is performed on the block.
6. The above step is performed until all the watermark bits are stored.

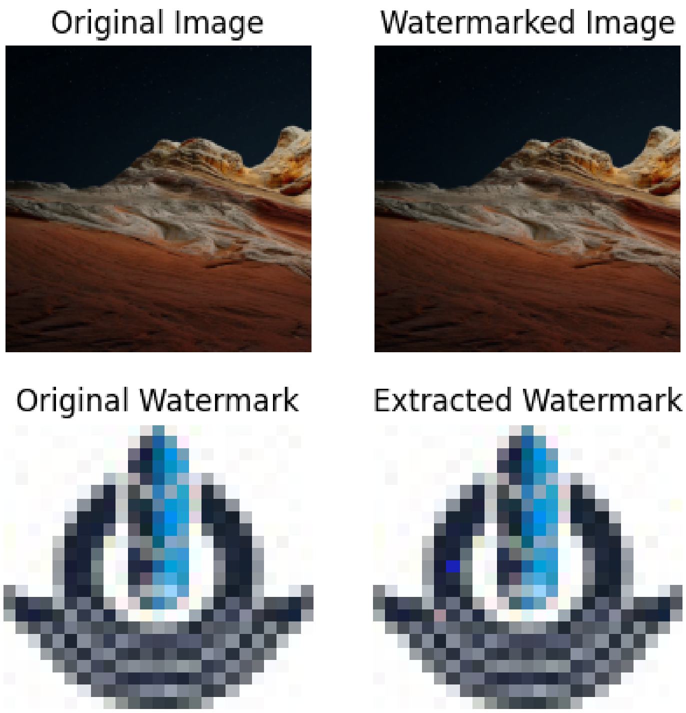
To extract the watermark following steps are followed:

1. The key is needed to get the block size and the initial number for the pseudorandom number generator.
2. Step 5 from above is performed and the tens place digit is stored in a string.
3. From the string integers are generated using 8 consecutive bits.
4. Original watermark size is required and using this size the string is reshaped to appropriate size.

The result without any attacks are as follows:

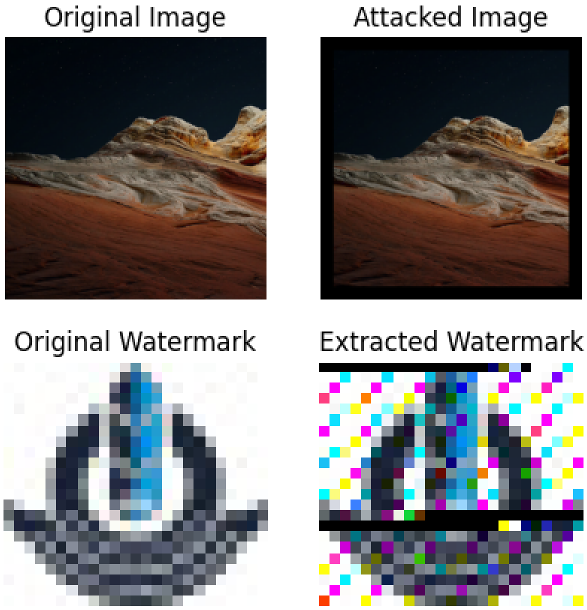
PSNR - 50.30

NCC - 0.99



Attacks used to check robustness of the algorithm.

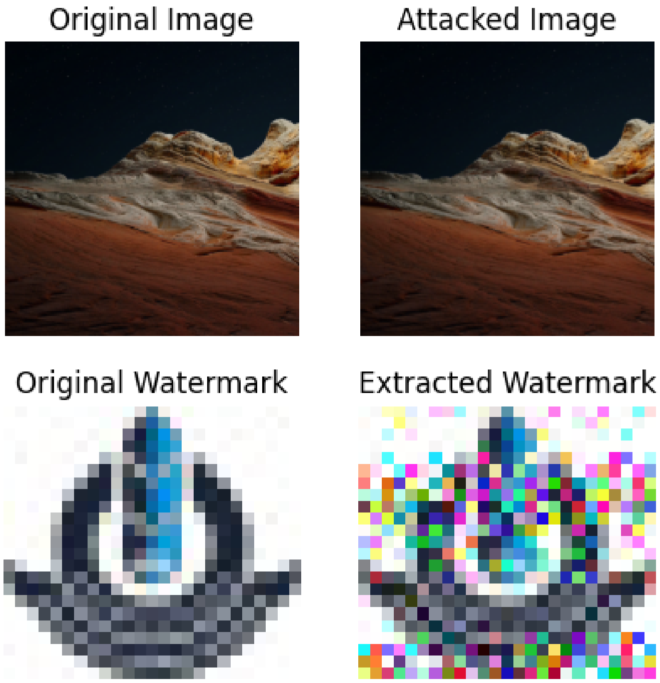
1. **Cropping Attack** (Geometric attack)
   5% of the boundary of the image is converted completely black and then the extraction algorithm is performed.
   Result



NCC - 0.91

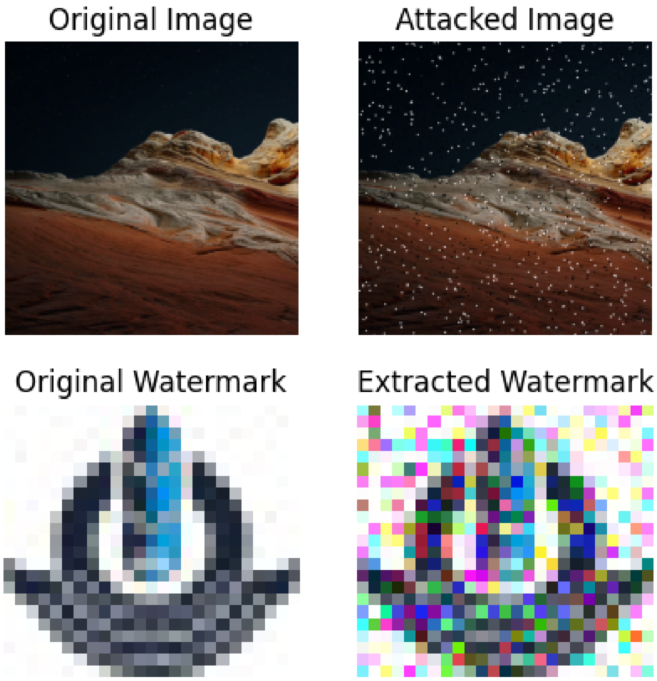
2. **Scaling Attack** (Geometric attack)
   The watermarked image was scaled down to 85% and then resized to the original image. This process will affect the quality of the image.
   Result:

NCC - 0.94



3. **Salt and Pepper Noise Attack** (Signal Processing Attack)
   In this attack we make the pixel intensity of random locations either highest (255) or lowest (0). This will add random white and black dots throughout the image.
   Result

NCC - 0.96



4. **Median filtering noise attack** (Signal Processing Attack)
   Median filtering is a signal processing attack that is used to smooth an image by replacing each pixel with the median value of the pixels in a surrounding neighborhood.
   Result

NCC - 0.90