

Tartu University  
Faculty of Science and Technology  
Institute of Technology

Georg Reintam

**Wavelet based digital art protection**

Master's thesis (30 EAP)  
Robotics and Computer Engineering

Supervisor(s):

Prof. Gholamreza Anbarjafari  
Msc Rain Eric Haamer

Tartu 2021

# Resümee/Abstract

## **Wavelet based digital art protection**

This thesis objective is to provide a robust watermarking algorithm to protect digital images. The proposed algorithm is using wavelet-based watermarking in which we are investigating how embedding in high-frequency subbands and low-frequency subbands would affect the robustness of the watermark while facing typical signal processing attacks.

Additionally, the proposed algorithm uses linear algebraic factorization methods, SVD and QR decomposition, to further secure the embedded information. Since the embedded watermark images in real-world application may differ a lot, then an additional objective is to investigate the effect of symmetry of the watermark on the introduced algorithm.

**CERCS:** T111 Image processing

**Keywords:** Watermark, QR Decomposition, SVD, Non-blind

## **Lainiku põhine digitaalse kunsti kaitsmine**

Selle töö eesmärk on välja pakkuda robustne digitaalne vesimärki sisse panev ning välja võttev algoritm kaitsmaks digitaalseid pilte. Välja pakutud algoritm kasutab lainiku põhise vesimärki ning kasutab QR-lagundamist ning singulaarse väärtuse dekompositsiooni, et saavutada parem kaitse väliste rünnakute vastu. Lisaks on testitud erinevaid vesimärke - sümmetrilised ja mitte-sümmetrilised - et näha, kuidas need vastu peavad rünnakutele.

**CERCS:** T111 Pilditöötlus

**Märksõnad:** Lainik, Vesivärv, Singulaarse väärtuse dekompositsioon

# Contents

<b>Resümee/Abstract</b>	<b>2</b>
<b>List of Figures</b>	<b>4</b>
<b>List of Tables</b>	<b>5</b>
<b>Abbreviations. Constants. Generic Terms</b>	<b>6</b>
<b>1 Introduction</b>	<b>7</b>
<b>2 Literature review</b>	<b>8</b>
2.1 Watermark . . . . .	8
2.2 Discrete Wavelet transform . . . . .	10
2.3 QR and Singular Value Decomposition . . . . .	10
<b>3 Methodology</b>	<b>12</b>
3.1 Watermark Embedding . . . . .	12
3.2 Detailed description of watermark embedding . . . . .	12
3.3 Watermark Extracting . . . . .	14
3.4 Detailed description of watermark extracting . . . . .	14
3.5 Attacks for the image . . . . .	18
3.6 Quality Measurements . . . . .	18
<b>4 Experimental results</b>	<b>20</b>
4.1 Discussion . . . . .	21
<b>5 Conclusion and Future work</b>	<b>27</b>
5.1 Future work . . . . .	27
<b>Bibliography</b>	<b>29</b>
<b>Non-exclusive license</b>	<b>31</b>

# List of Figures

2.1	Block scheme showing general algorithm for watermark (colored bubbles) with white boxes indicating different methods available at that stage. . . . .	9
3.1	Block diagram of the watermark embedding . . . . .	15
3.2	Block diagram of watermark extraction . . . . .	17
3.3	Various attacks on images . . . . .	19
4.1	Host images . . . . .	20
4.2	Embedded Watermarks . . . . .	21

# List of Tables

4.1	Symmetrical watermark with $K = 0.5$ . A = Attack name. B = Attacked image in HH. C = Recieved watermark in HH. D = Attacked image in LL. E = Recieved watermark in LL . . . . .	22
4.2	Non-symmetrical watermark with $K = 0.5$ . A = Attack name. B = Attacked image in HH. C = Recieved watermark in HH. D = Attacked image in LL. E = Recieved watermark in LL . . . . .	23
4.3	Non-symmetrical watermark with $K = 50$ . A = Attack name. B = Attacked image in HH. C = Recieved watermark in HH. D = Attacked image in LL. E = Recieved watermark in LL . . . . .	24
4.4	Non-symmetrical watermark with $K = 50$ . A = Attack name. B = Attacked image in HH. C = Recieved watermark in HH. D = Attacked image in LL. E = Recieved watermark in LL . . . . .	25
4.5	PSNR, SSIM, MSE value of watermark image when embedding strength coefficient is $K = 0.5$ . . . . .	26
4.6	PSNR, SSIM, MSE value of watermark image when embedding strength coefficient is $K = 50$ . . . . .	26

# Abbreviations. Constants. Generic Terms

**HH** - HIGH-HIGH subband

**LL** - LOW-LOW subband

**HL** - HIGH-LOW subband

**LH** - LOW-HIGH subband

**SVD** - Singular value decomposition

**DWT** - Discrete wavelet transform

**PSNR** - Peak signal-to-noise ratio

**SSIM** - The structural similarity index measure

**MSE** - Mean Square Error

# 1 Introduction

Digital art is becoming very important. Since there are more ways of expressing oneself with new technological ways (hardware, types of digital art [1]) then there is more sophisticated art in the market. On the other side, the consumption of multimedia is rising with each day making the demand for digital art even greater. Therefore it raises the problem of defending one content copyright and a way to trace the original owner. It is common for an artist to sign their canvas corner with their signature to prove its ownership, but in digital art this kind of method is vulnerable. It is easy to copy the image from somewhere by a third party and use it for their interest. So the problem is, how to find out who has the right to the image and if it is original or not [2].

One way to fix that problem is to insert hidden information or a watermark inside of the image. Inserted watermark has to be able to be extracted and withstand various attacks, such as image copying, lossy compression, filtering, that may be done intentionally to make it claim its ownership. On the flip side, the author of the image does not want to corrupt the image with the visible watermark or modified image pixel values so that it can ruin the experience for the buyer. Therefore a balance between the strength of the embedding watermark and the quality of the watermarked image has to be found.

To combat the copyright problem with inserting the watermark, then there are various ways of implementing it where all are taking into account the needs of the embedding. These are for example format of the image, robustness against certain attacks only, embedded watermark to be seen on the image and so on. Mainly the algorithms are separated in terms of the processing domain: spatial and transform domain. When the first one is easier to perform in terms of processing power and mathematical complexity, then they lack robustness towards attacks.

This thesis objective is to provide a robust watermarking algorithm to give a solution to the aforementioned balance. The proposed algorithm is using wavelet-based watermarking in which we are investigating how embedding in high-frequency subbands and low-frequency subbands would affect the robustness of the watermark while facing typical signal processing attacks. Additionally, the proposed algorithm uses linear algebraic factorization methods to further secure the embedded information. Since the embedded watermark images in real-world application may differ a lot, then an additional objective is to investigate the effect of symmetry of the watermark on the introduced algorithm.

This thesis is structured so that the opening sections give a detailed overview of the proposed algorithms in the literature and how they differ from each other. The chapter ends with a more detailed overview of the methods used in the proposed algorithm. Chapter 3 is giving a detailed overview of the proposed algorithm and explaining methods to evaluate it. The final section of this work reviews the results that were obtained from the experiments.

## 2 Literature review

### 2.1 Watermark

A successful digital watermark is a piece of code embedded in a multimedia file (audio, image, video) with the goal of providing copyright information. This piece of code that is embedded can be a digital signature or a hidden secret message (steganography). The beauty of the watermark lies in its secrecy it has the ability to add a digital signature in a way that is invisible to the common eye. Without that artists have to advertise their work by downsampling, modifying or other various ways to insure against theft.

The common algorithm that is used in the literature divides the process into two parts: embedding and extraction. Watermark embedding has the goal of embedding the information into a selected multimedia file. The second part extraction is the process of receiving the embedded information. This process is executed successfully if the received information from the unsecured communication line does not have any defects. Those defects can come from various attacks that are aimed to either delete or manipulate the embedded information.

Successful extraction is done, if [3]

- To be able to determine whether an image has been altered or not;
- Robustness - To be able to locate any alteration made on the image
- To be able to integrate embedding data with host image rather than as a separate fail
- Imperceptibility - The embedded authentication information is invisible under normal viewing conditions
- To allow the watermarked image to be stored in lossy-compression format
- Computational complexity the computational load of the algorithm should have

Watermarking algorithms in the literature are divided into many different categories [4]: Watermark type, host data, domain, perceptivity, robustness, data extraction. All the mentioned categories have their purpose and when designing an algorithm one has to take them into account. Figure 2.1 illustrates the categories. The foremost one has to know is the host data image, text, audio, video. Also here it matters what kind of data type is represented (RGB, 3D [5], grayscale). Watermark type means what is the embedded information: image or noise.

Perceptivity means if you do want your embedded watermark to be visible in the image or you prefer to keep that hidden. Robustness is where there are robust and fragile watermark techniques, where in the first one the watermark is very strongly embedded to remain resilient to



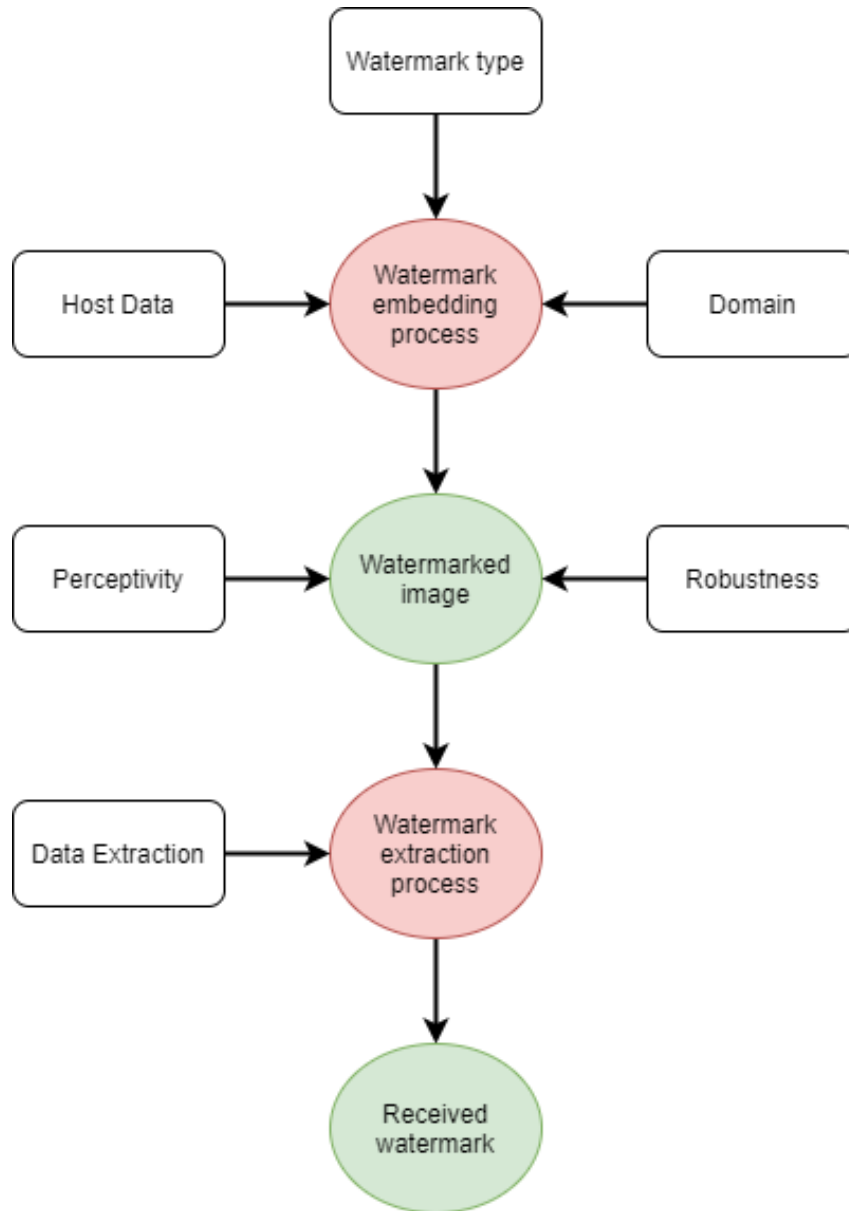


Figure 2.1: Block scheme showing general algorithm for watermark (colored bubbles) with white boxes indicating different methods available at that stage.

attacks and are therefore used mostly for copyright protection. Fragile watermarking schemes are very easily manipulated [6]. Also, there are semi-fragile algorithms available, where they can resist certain types of attacks. In the watermark extraction process, it is also important to know whether you need it to be blind, semi-blind or non-blind [7]. In the blind algorithms, a secret key is needed to extract the embedded watermark. In semi-blind algorithms, there is a need for a secret key and the originally inserted watermark. And lastly, in the blind version, a secret key, the original signal and watermark embedding sequence is needed

Finally, and most importantly watermark algorithms generally are grouped into spatial and frequency (transform) domain algorithms. The spatial algorithms embed the watermark into the digital content by pixel modification [8, 9]. The most commonly used algorithm is the least significant bit (LSB) method. This method is used to add secret information in the lowest bit in a series of numbers in binary. It is enough to embed into 1 to 4 least bits (half of the 8-bit

image) because the watermark quality is low. These changes are enough to hide the necessary information but to be unseeable to the human visibility system [10]. This way of inserting a watermark has very low computational complexity, but in a case of attack where pixel values are changed by a third party, then the extracted watermark is easily corrupted. This is making it not imperceptible or robust.

## 2.2 Discrete Wavelet transform

To overcome spatial domain shortcomings then an alternative way is to use wavelets. The frequency-domain uses Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT) in order to convert the pixel values to a set of correlated values which leaves a deeper impact over a certain region of values within the image [11]. The DWT has the advantage over other algorithms, that it takes into account the local image characteristics at different resolution levels.

The DWT produces a time-frequency representation of a signal, which is computed using successive high and low pass filters of a discrete time-domain signal. DWT decomposes an input signal into four bands of data resulting in four different frequency subbands: Low-Low (LL), Low-high (LH), High-Low (HL), High-High (HH ). If the input signal is an image, then this transformation reads images as vectors in the vector space of all images using 2-dimensional functions. This extracts hidden information from the image that can be used in future data processing.

## 2.3 QR and Singular Value Decomposition

QR decomposition is a procedure of decomposing a matrix  $A$  of  $m \times n$  into a product  $A = Q \times R$ , where  $Q$  is an orthogonal matrix and  $R$  is an upper triangular matrix. The properties of the  $R$  matrix is that when the columns in the matrix  $A$  have correlation with each other, then absolute values of the elements in the first row of the  $R$  matrix are greater than those in other rows. Greater the matrix element in the first row of  $R$  is, the bigger the quantization step and bigger the quantization remainder is. Greater quantization remainder is, the greater the allowed modification range is [12]. In image processing this can be reflected in defining the important components of the image [13, 14], which can be used afterwards with different transformations in order to transform them into a usable space.

The singular value decomposition (SVD) purpose is to rotate the data so that the first vector directions have the most data variance and this will continue in a declining order. This gives a way to factorize the matrix into a product of three matrices. If we have given a matrix  $A$  with dimension  $m \times n$  then it results in 3 new matrices: Two of the matrices  $U$  and  $V$  are a unitary matrix and matrix  $S$  is a diagonal matrix. The diagonal elements of  $S$  are called singular values of  $A$ . SVD decomposes the image into different parts and indicates the degree of the significance of each decomposed part. The data in the three matrices are sorted by how much it contributes to the matrix  $A$  product. This will give an approximation by using only the most important parts of the matrices [15].

SVD is heavily used in large data augmentation and in image compression algorithms [16] where its goal is to reduce high-dimensional data into fewer dimensions and only retain impor-

tant information. Singular matrix obtained from SVD is used to identify the most significant (i.e, stable) components (i.e., eigenvalues) of an image [17–20]. A good watermarking algorithm aims to insert the hidden message into stable components of an image so that the attacks will have minimal impact on retrieving them in the extraction stage.

# 3 Methodology

## 3.1 Watermark Embedding

Watermark embedding starts with reading in the grayscale image. Image is then divided into four blocks and entropy of each block is calculated as a threshold. Average of all blocks is found and the following is done only for blocks that have an entropy value lower than calculated threshold. Block is decomposed into four frequency bands using two-level DWT. After applying orthogonal-triangular decomposition on the outcome of the previous step, the diagonal matrix is calculated. This diagonal matrix is used in SVD and singular values of a cover image are added with singular values of watermark image. After that, inverse SVD, QR decomposition and DWT are used to get a watermarked block. Modified blocks are added together with higher entropy blocks and a watermarked image is obtained.

## 3.2 Detailed description of watermark embedding

Watermark embedding scheme is presented in FIGURE and explained in the following. Convert the image into grayscale and divide  $m \times n$  into  $\alpha \times \beta$  blocks, where  $\beta$  divides  $m$  and  $\beta$  divides  $n$ . Let  $M = \frac{m}{\alpha}$  and  $N = \frac{n}{\beta}$ . Then each block can be described as in equation 3.1.

$$B_{mn} \quad m \in \{1 \dots M\}, n \in \{1 \dots N\} \quad (3.1)$$

Calculate entropy value for each individual block, where the entropy value is designated as  $E$ . Calculate the average of all entropy values  $E$  from all blocks and denote the outcome as the threshold  $T$ . This can be calculated as given in equation 3.2.

$$T = \sum_{m=1}^M \sum_{n=1}^N \frac{E(B_{mn})}{M \times N} \quad (3.2)$$

Use two-level Discrete wavelet transformation on each block with entropy value  $E$  less than calculated threshold  $T$  to decompose it into four sub-bands as given in equation 3.3.

$$[LL, LH, HL, HH] = DWT(B_{mn}), \forall (B_{mn}) \in \{B_{mn} : E(B_{mn}) < T\} \quad (3.3)$$

Apply QR decomposition to matrix  $LL$  or  $HH$  (depending on which sub-band the watermark is embedded) to calculate the diagonal matrix as given in equation 3.4.

$$\begin{aligned} [Q] &= QR(LL) \\ D1 &= \text{diag}(R) \\ D &= \text{zeros}(R) \\ D &= D1 \end{aligned} \quad (3.4)$$

Apply SVD to diagonal matrix  $D$  from equation 3.4 to further decompose it as shown in equation 3.5.

$$[U \ S \ V] = SVD(D) \quad (3.5)$$

Apply SVD to watermark image  $W$  and decompose it as shown in equation 3.6.

$$[U_1 \ S_1 \ V_1] = SVD(W) \quad (3.6)$$

Calculate new singular values by adding original images decomposed singular values to watermark images singular values multiplied by scaling factor  $K$  that is for controlling the strength of the added watermark. This is shown in equation 3.7.

$$S_2 = S + K \times S_1 \quad (3.7)$$

Combine unitary matrices  $U$  and  $V$  from the decomposed original image with new singular values calculated in equation 3.7 as shown in equation 3.8 .

$$D2 = U \times S_2 \times V^T \quad (3.8)$$

Replace upper-triangular matrix  $R$  diagonal values with modified diagonal matrix  $D2$  as shown in equation 3.9.

$$R = D2 \quad (3.9)$$

Combine unitary matrix  $Q$  with modified upper-triangular matrix  $R$  as shown in equation 10

$$C2 = Q \times R \quad (3.10)$$

Calculate inverse DWT to get a watermarked image block as shown in the equation 3.11. Use modified  $LL$  subband  $C2$  and  $LH, HL$  and  $HH$  are the ones acquired in equation 3. Similar is in the case of inserting watermark into  $HH$  subband where it is replaced with the  $LL$  values in previous equations and in the inverse DWT modified  $HH$  subband  $C2$  is inserted and  $LL, HL, LH$  are the same as in equation 3.3.

$$I = IDWT(C2 LH HL HH) \quad (3.11)$$

Add together modified low entropy blocks with high entropy blocks. This will result in a watermarked grayscale image and the whole process is visualised in 3.1.

### 3.3 Watermark Extracting

Watermark extraction aims to find the embedded watermark without any corruption. For that reason in our method, the original image and watermarked image is divided into four blocks and entropy of each original images block is found together with average entropy value. Following is done for all original images blocks that have entropy lower than calculated threshold and for corresponding watermarked images blocks. Two-level DWT is applied to both images block and low-frequency bands are found. Thereafter QR decomposition is applied on either  $LL$  or  $HH$  on those results. Calculated original images diagonal matrix from orthogonal-triangular decomposition and watermarked image diagonal matrix are used with SVD to get both image singular values. Original image singular values are subtracted from watermarked image singular values to get watermark singular values. SVD is also applied to watermark images to get unitary matrices. Using singular values from subtraction and unitary matrices obtained from watermark image, inverse SVD is used to get extracted watermark image.

### 3.4 Detailed description of watermark extracting

Watermark extraction scheme is presented in Figure 3.2 and explained in the following. Read in the original grayscale image. divide  $m \times n$  into  $\alpha \times \beta$  blocks, where  $\beta$  divides  $m$  and  $\beta$  divides  $n$ . Let  $M = \frac{m}{\alpha}$  and  $N = \frac{n}{\beta}$ . Then each block can be described as in equation 3.12

$$B'_{mn} \quad m \in \{1 \dots M\}, n \in \{1 \dots N\} \quad (3.12)$$

Divide watermarked image corresponding colour channels similarly into  $\alpha \times \beta$  blocks, where each block can be described as in equation 3.13 .

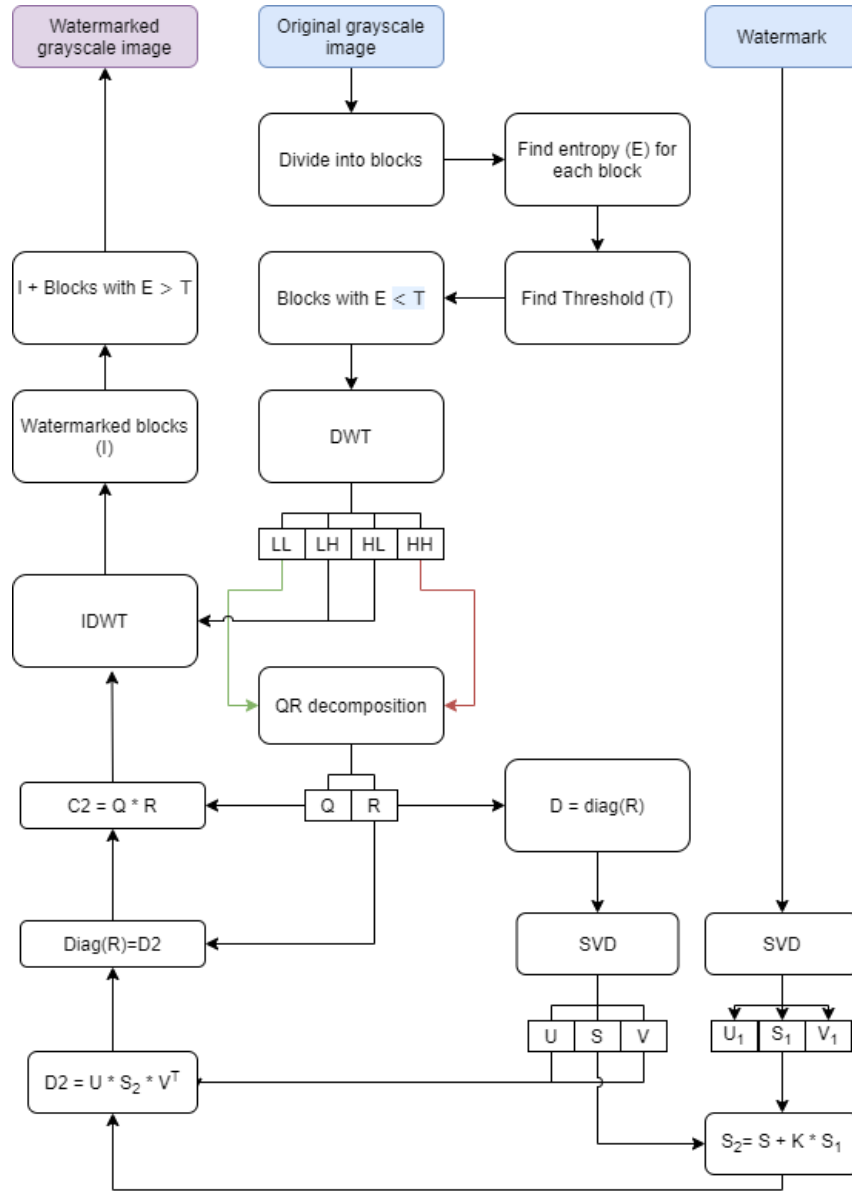


Figure 3.1: Block diagram of the watermark embedding

$$T = \sum_{m=1}^M \sum_{n=1}^N \frac{E(B_{mn})}{M \times N} \quad (3.13)$$

Calculate entropy value for each block of the original image, where the entropy value is designated as  $E$ . Then calculate the average of all entropy values  $E$  for all blocks of the original image and denote the outcome as the threshold  $T$ .  $T$  can be calculated as given in equation 3.14 .

$$LL \ LH \ HL \ HH = DWT(B_{mn}), \forall (B_{mn}) \in \{B_{mn} : E(B_{mn}) < T\} \quad (3.14)$$

Use two-level DWT on each original image block with entropy value  $E$  less than calculated threshold  $T$  to decompose it into four subbands as given in equation (3.15).

$$LL' LH' HL' HH' = DWT(B'_{mn}), \forall (B'_{mn}) \in \{B'_{mn} : E(B_{mn}) < T\} \quad (3.15)$$

Apply QR decomposition to matrix LL (or if embedding to HH, then HH to calculate diagonal matrix as given in equation 3.16.

$$\begin{aligned} [QR] &= QR(LL) \\ D1 &= \text{diag}(R) \\ D &= \text{zeros}(R) \\ D &= D1 \end{aligned} \quad (3.16)$$

Apply QR decomposition to matrix LL (or if embedding to HH, then HH to calculate diagonal matrix as given in equation 3.17.

$$\begin{aligned} [Q'R'] &= QR(LL') \\ D1' &= \text{diag}(R') \\ D' &= \text{zeros}(R') \\ D' &= D1' \end{aligned} \quad (3.17)$$

Apply to diagonal matrix D from equation 3.17 to further decompose it as shown in equation 3.18.

$$[U \ S \ V] = SVD(D) \quad (3.18)$$

Apply SVD to diagonal matrix D from equation 3.18 to further decompose it as shown in equation 3.19.

$$[U' \ S' \ V'] = SVD(D') \quad (3.19)$$

Apply SVD to watermark image W and decompose it as shown in equation 3.20.

$$[U_1 \ S_1 \ V_1] = SVD(W) \quad (3.20)$$

Subtract singular values of the original images block from singular values of watermarked image block and divide the outcome by scaling factor K to get singular values of extracted watermark image as shown in equation 3.21.



$$S_1 = \frac{S' - S}{K} \quad (3.21)$$

Combine unitary matrices  $U_1$  and  $V_1$  from watermark image with extracted singular values calculated in equation 3.21 to get extracted watermark for each block as shown in equation 3.22.

$$WI = U_1 \times S_1 \times V_1^T \quad (3.22)$$

In Figure 3.2 is extraction algorithm visualised.

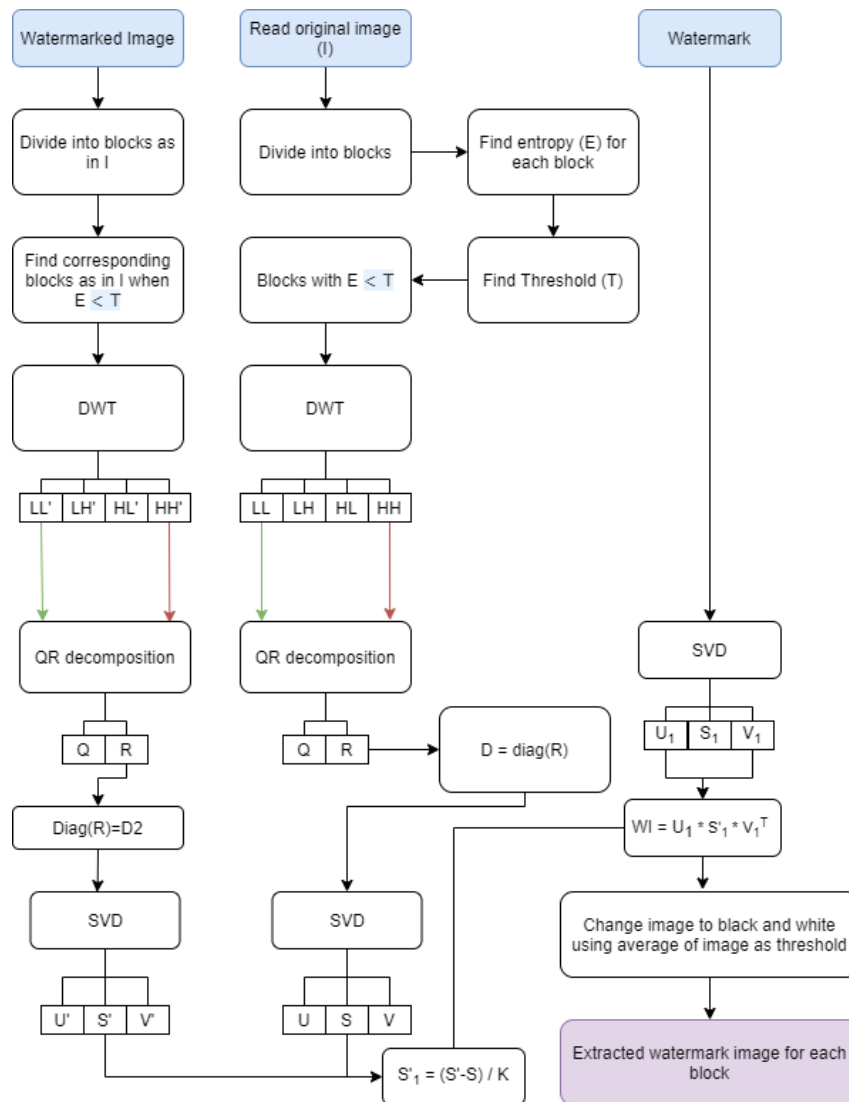


Figure 3.2: Block diagram of watermark extraction

### 3.5 Attacks for the image

Attacks for the image are taking place after it has left from the author to the world. From the watermark point of view, it is happening after the extraction and before the embedding. In the literature [21] there are many categories of attacks: removal attacks, geometric attacks, cryptographic attacks, protocol attacks.

The removal attack's goal is to remove the inserted watermark from the image without knowing the algorithm keys and its goal is to disturb the watermark information to the degree that it's hard to prove its belonging. In this work, we attack embedded images with Gaussian - and Salt Pepper noise. Gaussian noise is adding a noise signal to an image to corrupt the image [22]. The noise signal is using Gaussian probability distribution function 3.23 to generate random numbers. Salt and Pepper noise differs from the Gaussian noise with its probability distribution function. In image it represents randomly occurring white and black pixels in an image.

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{z^2}{2\sigma^2}} \quad (3.23)$$

Geometrical attacks [23] are aiming to distort the image with a displacement of its pixels. In this work, we are using image modification, image rotation, flip and resize. In an image modification attack, a randomly sized black box is added to the image. In a rotation attack, an image is rotated 45 degrees bilinearly interpolated pixels and resized to match the original image. In a flip attack, an image is just turned upside-down and resize attack is where the image is down-scaled.

Cryptographic and protocol attacks are aiming to manipulate the embedding algorithm. Cryptographic attack aims to crack the security in schemes (with a huge watermarked image dataset) and using that information to remove the watermark. Protocol attack's [24] goal is to add attackers' own watermark into the image to question the true owner of the image. These attacks are complicated and out of the reach of this thesis.

### 3.6 Quality Measurements

A good watermark has good robustness and imperceptibility. The robustness of the algorithm means its performance against the intentional or unintentional removal or degradation (attacks). Imperceptibility is a way to measure the quality of the watermark.

Four different metrics are used to measure degradation: Peak signal-to-noise ratio (PSNR), Structural similarity index (SSIM), Mean-squared error (MSE), Visually. The PSNR defines the similarity between an original image and the reconstructed image in decibels. The higher the PSNR value the closer it is to an actual image. It can be calculated by using the following equation 3.24, with logarithm base 10, MAX is the maximum possible pixel value of the image (255 in eight bits per pixel) and MSE is a mean-squared error.

$$PSNR_{db} = 10\log\left(\frac{MAX^2}{MSE}\right) \quad (3.24)$$

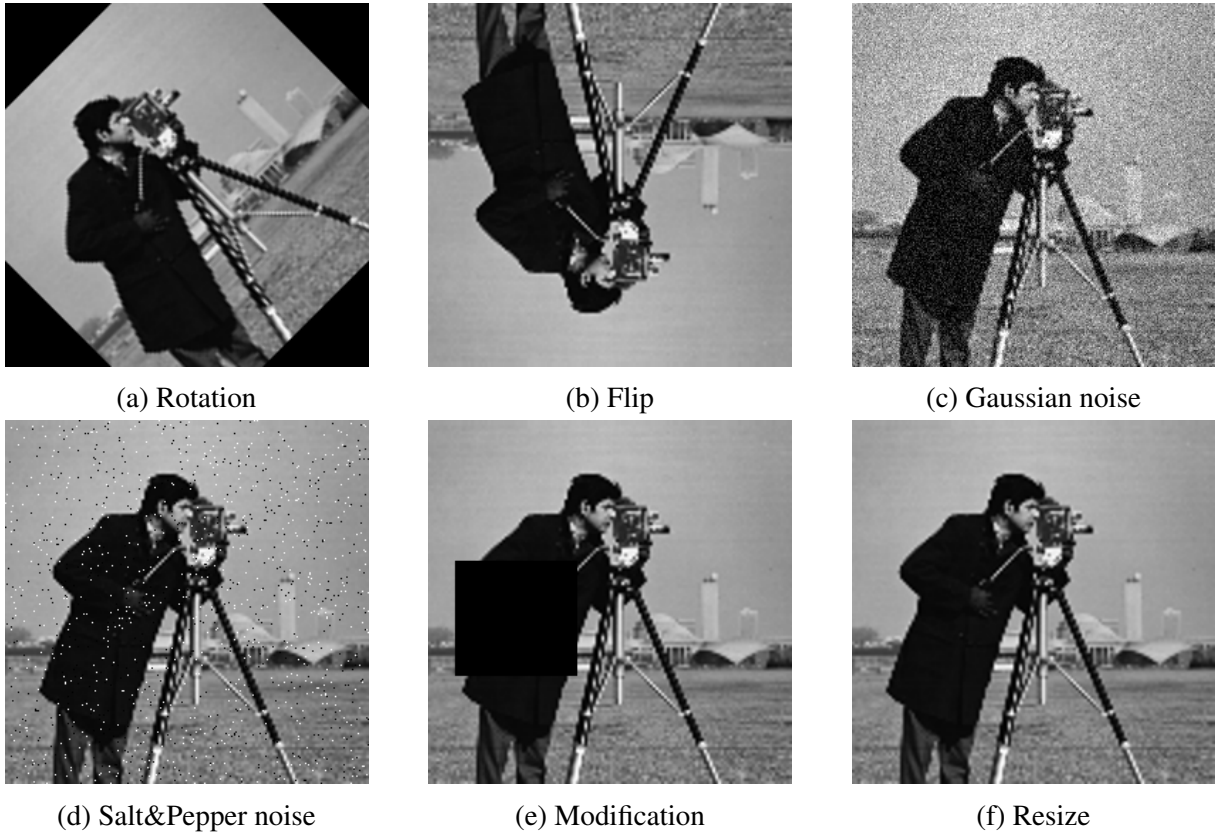


Figure 3.3: Various attacks on images

SSIM is based on the computation of three parameters: luminance, contrast, structural. The overall index is a multiplicative combination of these three parameters. The closer the SSIM value to 1 the higher the similarity between the two images is. It is calculated with equation 3.25 , where  $\mu_x$  is the mean of image x,  $\sigma_x$  is the standard deviation of the image x and  $C_x$  is the standard constant to avoid 0/0.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3.25)$$

The MSE is the average of the pixel difference between two images. It is calculated with equation 3.26, where I(S) is original and W(S) is the watermarked image.

$$MSE = \frac{1}{S} \sum_1^S (I(S) - W(S))^2 \quad (3.26)$$

These three metrics help us to both evaluate the robustness and imperceptibility. The fourth Visual metric is using human sight to tell the difference between the two images. It is more objective but can tell the difference between the two images.

## 4 Experimental results

The program was fully programmed and run in MATLAB 2021a and all the related add-ins. For the signal processing (2DWT, IDWT, CZT) a dedicated Signal Processing Toolbox was used. All of the attacks and other image modifications are using the Image Processing Toolbox library and its offered functions.

Multiple experiments were conducted using three  $256 \times 256$  gray scale images. Selected host images are taken from the image processing community which are used the most for watermark image testing Cameraman, baboon, Serrano. They can be seen in Figure 4.1.

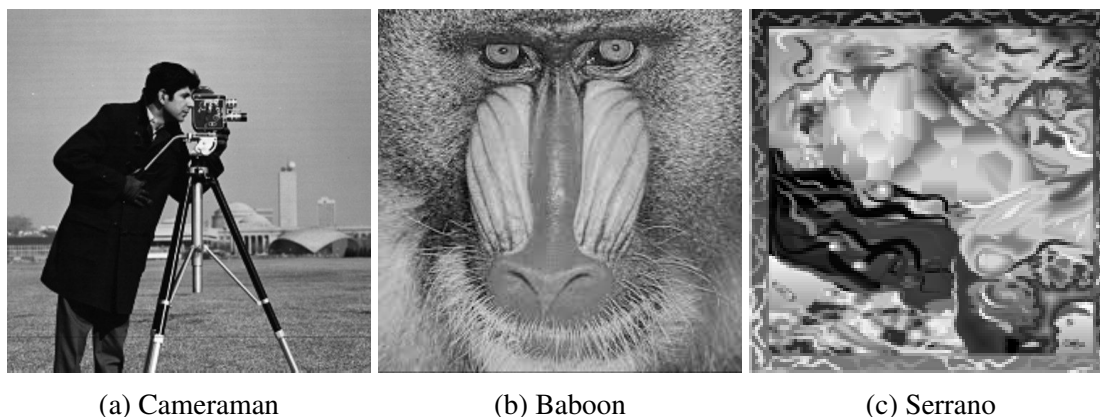


Figure 4.1: Host images

In the experiment two  $128 \times 128$  sized grayscale watermark images were used. Watermarks fall into two categories: symmetrical and non-symmetrical. Symmetrical watermarks are images that both sides (vertical) of the image hold equal values. Non-Symmetrical watermarks are when both sides are not equal. Two watermarks were used to study the result of symmetry of the watermark. Used watermarks are shown in figure 4.2.

Proposed algorithm was experimented with different watermark embedding strengths  $K$ : 0.5 and 50. Both watermarks were embedded into LL and HH subbands of the host image. The visual quality of the watermarked image with  $K$  value 0.5 on the symmetrical case is in table 4.1 and non-symmetrical in Table 4.2. Same table with  $K$  value 50 is represented in Table 4.3 and Table 4.4. Quantitative results - PSNR, SSIM, MSE - are presented in Table 4.5 and Table 4.5.

In 4.1, we see different signal processing attacks visually: 45 degree rotation, flip, Gaussian noise with 0.02 variance, Salt and Pepper noise with 0.02 variance, modification, resize on watermarked image in which watermark is symmetrical. As it can be seen from column C (representing extracted watermark while embedded in HH subband) and column E (representing

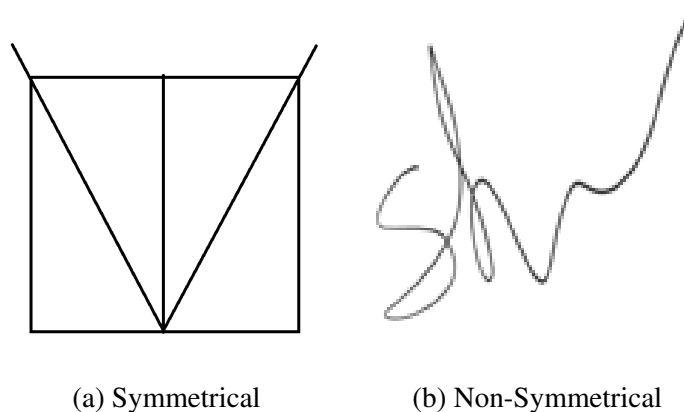


Figure 4.2: Embedded Watermarks

extracted watermark while embedding in LL subband) that for no attack and image rotation attack embedding in LL subband is more robust than embedding in HH while for flip, gaussian noise, salt and pepper noise, modification and resize attacks embedding in HH subband is more robust. In order to investigate the effect of asymmetrical watermark, 4.2 has been created in which an asymmetrical watermark has been embedded. All the aforementioned signal processing attacks have been applied and as can be seen from column C and E, embedding in the HH subband is more robust than embedding in LL subband. As this result is in line with the symmetrical watermark, we can see our proposed algorithm is performing well regardless of the existence of symmetry in the watermark.

In Tables 4.3 and 4.4 similar attacks are applied to watermarked images. For this measurement K value - which is embedding strength coefficient - is increased from 0.5 to 50. As seen from both - 4.3 and 4.4 - embedding watermarks with higher strength coefficient will in return corrupt the image in LL subband. As it can be seen in column E, no attack and rotation attacks are more robust in LL. That is similar to lower K values.

In order to introduce some quantitative results, PSNR, SSIM and MSE between the original (without embedded watermark) image and watermarked image have been calculated when the embedding factor, K, is varying and the embedding is happening in different subbands. 4.5 is showing the results for K=0.5 and Table 4.5 is showing results for the K = 50.

## 4.1 Discussion

The main objective of this thesis was being investigated using aforementioned experiments. Throughout those experiments the impact of wavelets and the introduced updated watermarking scheme was studied. The experimental results showed that embedding watermark in HH subband will result in a more robust extraction of watermark when signal processing attacks have been employed. This is due to the fact that high frequency components lay repetitively in various parts of the image enabling the watermark to be extracted very well. However, the watermark embedding in LL can be very useful if the aim of the process is to have a ghost view of the watermark in the digital image. Moreover throughout the experimental results, as can be seen in table 4.1 and 4.2, embedding watermarks in HH subbands preserve the visual quality of

Table 4.1: Symmetrical watermark with  $K = 0.5$ . A = Attack name. B = Attacked image in HH. C = Recieved watermark in HH. D = Attacked image in LL. E = Recieved watermark in LL


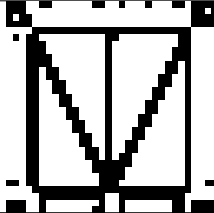

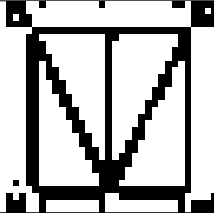

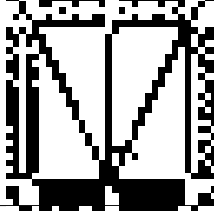

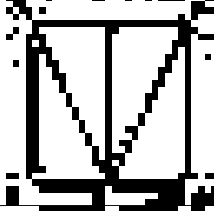

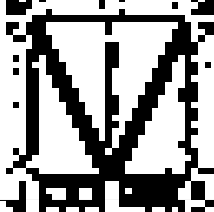



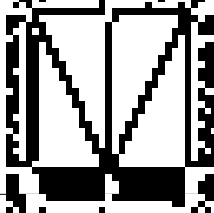

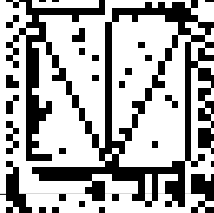

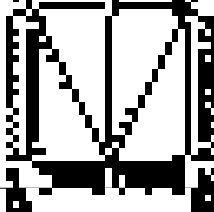

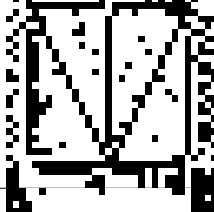

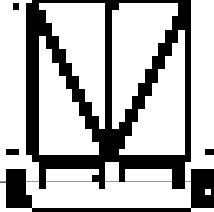

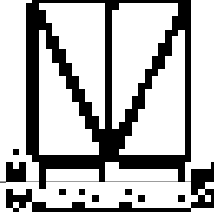

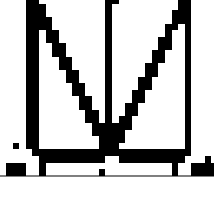

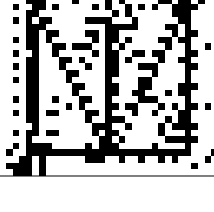











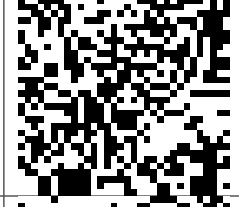











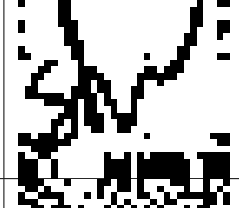




A	B	C	D	E
No attack				
Rotation				
Flip				
Gaussian noise				
Salt&Pepper noise				
Modification				
Resize				

image.

Proposed algorithm can be useful for anyone that wishes to copyright their image on the world wide web by adding a watermark inside their picture. This gives an artist a way of showing their work without doing serious downsampling, which in return corrupts the image and can

Table 4.2: Non-symmetrical watermark with  $K = 0.5$ . A = Attack name. B = Attacked image in HH. C = Recieved watermark in HH. D = Attacked image in LL. E = Recieved watermark in LL

A	B	C	D	E
No attack				
Rotation				
Flip				
Gaussian noise				
Salt&Pepper noise				
Modification				
Resize				

therefore ruin the experience for the seller.

Table 4.3: Non-symmetrical watermark with  $K = 50$ . A = Attack name. B = Attacked image in HH. C = Recieved watermark in HH. D = Attacked image in LL. E = Recieved watermark in LL


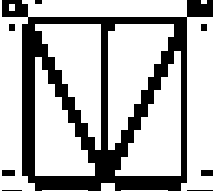

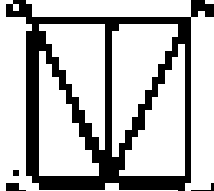

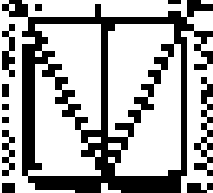

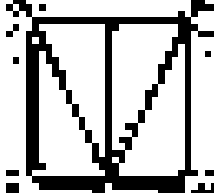

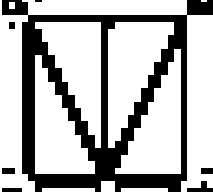

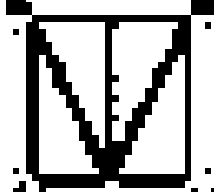

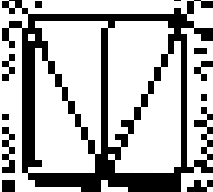

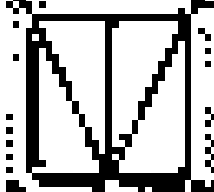

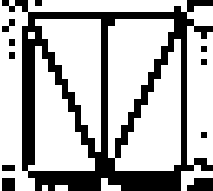

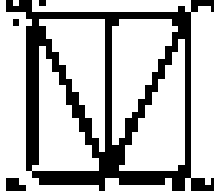

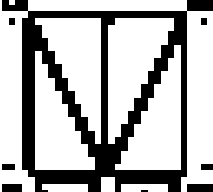

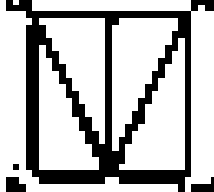

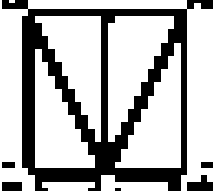

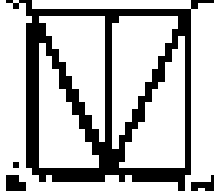
A	B	C	D	E
No attack				
Rotation				
Flip				
Gaussian noise				
Salt&Pepper noise				
Modification				
Resize				



Table 4.4: Non-symmetrical watermark with  $K = 50$ . A = Attack name. B = Attacked image in HH. C = Recieved watermark in HH. D = Attacked image in LL. E = Recieved watermark in LL





























A	B	C	D	E
No attack				
Rotation				
Flip				
Gaussian noise				
Salt&Pepper noise				
Modification				
Resize				

Table 4.5: PSNR, SSIM, MSE value of watermark image when embedding strength coefficient is  $K = 0.5$

<b>Cameraman</b>	<b>Image</b>	<b>PSNR</b>	<b>SSIM</b>	<b>MSE</b>
	Symmetrical WM in HH	61.1157	0.9996	0.0258
	Non-Symmetrical WM in HH	60.9681	0.9996	0.0266
	Symmetrical WM in LL	61.1594	0.9996	0.0388
	Non-Symmetrical WM in LL	61.0090	0.9996	0.0408
<b>Baboon</b>				
	Symmetrical WM in HH	59.3928	0.9999	0.0373
	Non-Symmetrical WM in HH	59.3733	0.9999	0.0379
	Symmetrical WM in LL	59.2834	0.9999	0.0573
	Non-Symmetrical WM in LL	59.2834	0.9999	0.0596
<b>Serrano</b>				
	Symmetrical WM in HH	60.0911	0.9999	0.0320
	Non-Symmetrical WM in HH	60.2050	0.9999	0.0362
	Symmetrical WM in LL	60.2146	0.9999	0.0487
	Non-Symmetrical WM in LL	60.0974	0.9999	0.0508

Table 4.6: PSNR, SSIM, MSE value of watermark image when embedding strength coefficient is  $K = 50$

<b>Cameraman</b>	<b>Image</b>	<b>PSNR</b>	<b>SSIM</b>	<b>MSE</b>
	Symmetrical WM in HH	27.1929	0.7415	11.6226
	Non-Symmetrical WM in HH	26.9928	0.7543	11.6709
	Symmetrical WM in LL	25.2837	0.7636	14.0056
	Non-Symmetrical WM in LL	25.1454	0.7722	13.9989
<b>Baboon</b>				
	Symmetrical WM in HH	25.3767	0.8841	17.9482
	Non-Symmetrical WM in HH	25.1580	0.8828	18.0025
	Symmetrical WM in LL	23.1051	0.8814	22.7696
	Non-Symmetrical WM in LL	22.8796	0.8804	22.8035
<b>Serrano</b>				
	Symmetrical WM in HH	26.0218	0.9109	13.8812
	Non-Symmetrical WM in HH	25.7937	0.9093	13.8499
	Symmetrical WM in LL	25.4749	0.9017	19.3104
	Non-Symmetrical WM in LL	25.2546	0.9008	19.2680

## 5 Conclusion and Future work

This thesis proposes a non-blind watermarking technique that is done in the frequency domain using DWT and linear algebra factorizations SVD and QR to further improve the embedding procedure. Moreover, it has the ability to change the strength of watermark embedding. This has the goal that when higher embedding strength coefficient values, especially in LL subband, then the watermark will be easier to be detected and on the other hand, the artefact and ghost image is visible in the host image.

In order to see the robustness of the proposed pipeline, it was tested with multiple images that had different characteristics and were embedded with both symmetrical and non-symmetrical watermarks resulting in the watermarked image. Since the proposed algorithm uses DWT to decompose an input image into four different subbands then it was tested in which subband LL or HH is more robust and imperceptible. Watermarked images were then attacked with various methods with the goal of eliminating the watermark from that image.

From the tests, we can conclude that embedding watermarks in the HH are more robust to the attacks due to the fact that there are more high-frequency components distributed all along with the image than LL subbands. While doing the attacks on both of the subbands embedded images results that LL is more sensitive to some attacks. Increasing the watermark embedding strength coefficient resulted in having more robust extracted watermarks with the cost of ghost view of the watermark within the host image.

### 5.1 Future work

Due to advancement in utilisation of deep neural networks, in the future work it would be good to utilise more recent deep neural networks in order to conduct embedding and extracting of watermarks. Moreover, it will make sense to build on top of this thesis and conduct investigation on the impact of wavelets in deep neural network based watermarking schemes such as [24] and [25].

# Acknowledgements

I would like to thank my supervisor Prof. Gholamreza Anbarjafari for encouraging me throughout the thesis process on a hard time of covid-19 lockdowns as well as teaching me all the things that were needed for this thesis. Also, I want to thank Rain Eric Haamer for explaining things in plain Estonian language and sometimes being a interpreter between me and the professor.

# Bibliography

- [1] The Rise of Digital Art <https://www.artdictionmagazine.com/the-rise-of-digital-art/19.05.2021>, 10:15 (UTC).
- [2] Artists report discovering their work is being stolen and sold as NFTs <https://www.abc.net.au/news/science/2021-03-16/nfts-artists-report-their-work-is-being-stolen-and-sold/1324940818.05.2021>, 19:15 (UTC).
- [3] Min Wu and Bede Liu, "Watermarking for image authentication," Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269), 1998, pp. 437-441 vol.2, doi: 10.1109/ICIP.1998.723413.
- [4] Saini, Lalit Shrivastava, Vishal. (2014). A Survey of Digital Watermarking Techniques and its Applications.
- [5] Kim, Wook-Hyung Hou, Jong-Uk Jang, Han-Ui Lee, Heung-Kyu. (2018). Robust Template-Based Watermarking for DIBR 3D Images. Applied Sciences. 8. 911. 10.3390/app8060911.
- [6] Bongirwar, Apeksha Fazeel, A Zama, Fazeel. (2016). A Survey on Robust Watermarking in Non Blind Method.
- [7] Arya, Ranjan Singh, Shalu Saharan, Ravi. (2015). A Secure Non-blind Block Based Digital Image Watermarking Technique Using DWT and DCT. 10.1109/ICACCI.2015.7275917.
- [8] H. H. Larijani and G. R. Rad, "A new spatial domain algorithm for gray scale images watermarking," 2008 International Conference on Computer and Communication Engineering, 2008, pp. 157-161, doi: 10.1109/ICCCE.2008.4580587.
- [9] Frattolillo, Franco. (2014). Watermarking Protocols: Problems, Challenges and a Possible Solution. The Computer Journal. 58. 10.1093/comjnl/bxu015.
- [10] A. Bamatraf, R. Ibrahim and M. N. B. M. Salleh, "Digital watermarking algorithm using LSB," 2010 International Conference on Computer Applications and Industrial Electronics, 2010, pp. 155-159, doi: 10.1109/ICCAIE.2010.5735066.
- [11] Abraham, Jobin Paul, Varghese. (2016). An Imperceptible Spatial Domain Color Image Watermarking Scheme. Journal of King Saud University - Computer and Information Sciences. 31. 10.1016/j.jksuci.2016.12.004.
- [12] Jia, S. Zhou, Q. Zhou, H.. (2017). A novel color image watermarking scheme based on DWT and QR decomposition. Journal of Applied Science and Engineering. 20. 193-200. 10.6180/jase.2017.20.2.07.

- [13] Su, Qingtang Niu, Yugang Wang, Gang Jia, Shaoli Yue, Jun. (2014). Color image blind watermarking scheme based on QR decomposition. *Signal Processing*. 94. 219-235. 10.1016/j.sigpro.2013.06.025.
- [14] Su, Qingtang et al. Color image blind watermarking scheme based on QR decomposition. *Signal Process*. 94 (2014): 219-235.
- [15] Timbaumann, SVD image compression demo. <http://timbaumann.info/svd-image-compression-demo/> 18.05.2021, 13:14 (UTC).
- [16] Anbarjafari, Gholamreza Rufai, Awwal Demirel, Hasan. (2013). Lossy Image Compression Using Singular Value Decomposition and Wavelet Difference Reduction. *Digital Signal Processing*. 10.1016/j.dsp.2013.09.008.
- [17] P. Mitra, R. Gunjan and M. S. Gaur, "A multi-resolution watermarking based on contourlet transform using SVD and QR decomposition," 2012 International Conference on Recent Advances in Computing and Software Systems, 2012, pp. 135-140, doi: 10.1109/RACSS.2012.6212712.
- [18] Jia, Shao-li. (2014). A novel blind color images watermarking based on SVD. *Optik - International Journal for Light and Electron Optics*. 125. 28682874. 10.1016/j.ijleo.2014.01.002.
- [19] Poonam, Arora, Shaifali. (2018). A DWT-SVD based Robust Digital Watermarking for Digital Images. *Procedia Computer Science*. 132. 1441-1448. 10.1016/j.procs.2018.05.076.
- [20] M.S., Silja Kp, Soman. (2009). A Watermarking Algorithm Based on Contour let Transform and Nonnegative Matrix Factorization. 279-281. 10.1109/ARTCom.2009.198.
- [21] S. Voloshynovskiy, S. Pereira, T. Pun, J. J. Eggers and J. K. Su, "Attacks on digital watermarks: classification, estimation based attacks, and benchmarks," in *IEEE Communications Magazine*, vol. 39, no. 8, pp. 118-126, Aug. 2001, doi: 10.1109/35.940053.
- [22] C. Song, S. Sudirman, M. Merabti and D. Llewellyn-Jones, "Analysis of Digital Image Watermark Attacks," 2010 7th IEEE Consumer Communications and Networking Conference, 2010, pp. 1-5, doi: 10.1109/CCNC.2010.5421631.
- [23] Licks, Vinicius Jordan, Ramiro. (2005). Geometric Attacks on Image Watermarking Systems. *Multimedia, IEEE*. 12. 68 - 78. 10.1109/MMUL.2005.46.
- [24] Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M.P., Huang, H. and Molloy, I., 2018, May. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security* (pp. 159-172).
- [25] Kandi, Haribabu Mishra, Deepak Sai Subrahmanyam, Gorthi. (2016). Exploring the Learning Capabilities of Convolutional Neural Networks for Robust Image Watermarking. *Computers Security*. 65. 10.1016/j.cose.2016.11.016.

# Non-exclusive licence to reproduce thesis and make thesis public

I, Georg Reintam

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

## **“Wavelet based digital art protection”**

supervised by Prof. Gholamreza Anbarjafari and MSC Rain Eric Haamer

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.
3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.
4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

*Georg Reintam*  
**20.05.2021**