Exploiting QR Code Error Correction for Digital Image Watermarking

Yang-Wai Chow, Willy Susilo*, Joonsang Baek, Jongkil Kim, and Wei Zong
Institute of Cybersecurity and Cryptology,
School of Computing and Informatino Technology,
University of Wollongong, Australia
{caseyc, wsusilo, baek, jongkil, wzong}@uow.edu.au

Received: November 3, 2020; Accepted: January 22, 2021; Published: June 30, 2021

Abstract

The growing use of online digital media has made the task of protecting copyright and intellectual property increasingly more challenging. Data hiding techniques, such as digital watermarking, can be used to embed data within a signal for various purposes, e.g., for digital rights management. To achieve this, there are a number of desirable properties for digital watermarking, namely, imperceptibility, capacity, robustness and security. The research presented in this paper investigates a QR code based watermarking technique for digital images. The utilization of QR codes for this purpose has a number of advantages due to the QR code structure's inherent properties. As such, the aim of this work is to exploit the inbuilt error correction mechanism and to capitalize on the high data capacity of the QR code structure. This paper examines the proposed QR code watermarking technique and demonstrates its robustness and security against common digital image attacks that may be carried out by an adversary on the watermarked image.

Keywords: Data Hiding, Error Correction, QR Code, Watermarking

1 Introduction

With the extensive and widespread use, exchange and sharing of online digital media content, it has become increasingly difficult to protect copyright and intellectual property. Of the various techniques that can be used to achieve this, data hiding techniques, such as digital watermarking, can be used for the purpose of digital rights management.

Digital watermarking is a widespread field that has been studied over many decades [7]. The idea behind watermarking is to embed additional data within a signal and be able to extract this data when required [11]. This embedding of additional data within the signal must be performed in a way that does not interfere with normal usage of the signal. Furthermore, a successful watermark should be robust against signal alteration, up to a point at which the signal is damaged and loses its commercial value [21]. In light of this, there are four key properties that affect any watermarking system; namely, invisibility, capacity, robustness and security [7, 19]. These are described as follows:

- Invisibility, or imperceptibility, is the property whereby a human cannot perceive the difference between the original and watermarked signal;
- Capacity, or payload, is the amount of data that can be embedded by the watermarking scheme;

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), 12(2):114-140, June 2021
DOI:*10.22667/JOWUA.2021.06.30.114*

*Corresponding author: School of Computing and Information Technology, University of Wollongong, Northfields Avenue, Wollongong, NSW 2522, Australia, Tel: +61-2-4221-5535

- Robustness is the scheme's ability to withstand distortions to the watermarked signal;
- Security refers to a watermark scheme's resistance against intentional attacks by an adversary to impair the watermark.

The research presented in this paper, which investigates a QR code watermarking technique for digital images, extends the work previously presented in Chow et al. [4]. The purpose of the proposed QR code watermarking approach is to capitalize on the inherent error correction properties of the QR code structure, along with its high data capacity. The QR code error correction mechanism allows a QR code to be correctly decoded despite the presence of slight errors in the QR code, as long as the amount of error does not exceed the QR code's error correction capacity. Therefore, by embedding a QR code as a watermark within a digital image, the watermark can potentially withstand distortions to the signal, provided the QR code can be adequately reconstructed through the watermark extraction process.

There are two primary methods for embedding watermark data within digital images in an imperceptible manner. This can be done via the spatial domain or the frequency domain. There are a number of advantages of modifying coefficients in the frequency domain, for example, it incorporates features of the human visual system more effectively, it provides the ability to spread the embedded signal in the frequency domain, and it operates in the compressed domain which is also used by most compression standards [12]. Therefore, to make the watermark imperceptible, the proposed approach uses the Discrete Wavelet Transform (DWT) technique.

The aim of the proposed QR code watermarking approach, is to embed a QR code symbol within one of the DWT sub-bands of a digital image. Within the frequency domain, the strength of the embedded watermark can be adjusted based on the desired tradeoff between imperceptibility and robustness. This paper presents the proposed technique and examines its features with respect to the key watermarking properties. In addition, the paper demonstrates the robustness and security of the proposed QR code watermarking technique against common digital image attacks, like image compression, noise, cropping, sharpening and blurring, that may be carried out by an adversary.

1.1 Contribution

This paper presents a QR code based digital watermarking technique. The aim of this approach is to insert a QR code in a digital image by embedding it within one the image's DWT sub-bands. The purpose of embedding a QR code is to capitalize on the inherent error correction properties of the QR code structure. This results in a watermarking technique that is robust against common distortions, as a QR code can be correctly decoded as long as the error in the reconstructed QR code, obtained as a result of the watermark extraction process, does not exceed its error correction capacity. The proposed technique is examine in terms of key watermarking properties; namely, imperceptibility, capacity, robustness and security. In addition, this paper demonstrates the robustness and security of the proposed technique against common digital image distortions that may result from attacks conducted by an adversary on the watermarked image.

1.2 Paper Structure

The rest of this paper is organized as follows. Section 2 presents a background to key concepts related to the proposed watermarking technique. Related work on data hiding and watermarking using QR codes is discussed in Section 3. The proposed watermarking technique using QR codes is presented in Section 5. Section 6 shows experiment results demonstrating the effectiveness of the proposed technique. Finally, Section 7 summarizes and concludes the paper.

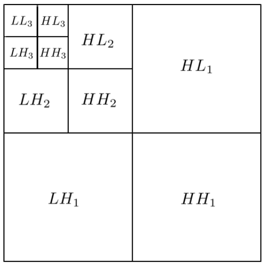


Figure 1: DWT at level 3.

2 Background

A brief background to some of the key concepts and techniques in this research is presented in this section.

2.1 The QR Code

The Quick Response code (QR code) is a two-dimensional (2D) barcode that was invented by the company Denso Wave [9] in 1994 for the automotive industry in Japan. Since its inception, the QR code has seen widespread adoption for various purposes in a variety of different applications. Its popularity has been attributed to its ease of use, robustness, fast decoding, high data capacity and so on.

A QR code symbol consists of a 2D array of light and dark squares, known as modules [13]. The QR code structure contains modules for encoding data and for function patterns. Function patterns consist of finder patterns, separators, timing patterns and alignment patterns. For example, there are three identical finder patterns located at the upper left and right, and lower left corner of the symbol. The finder patterns are for a QR code reader to recognize a QR code symbol and to determine its orientation.

In the QR code structure, a collection of 8 data modules form a codeword. The layout of the modules that form the codewords is determined by the QR code version. In addition, the QR code structure has an inherent error correction mechanism that allows data to be recovered even if a certain number of codewords have been corrupted. The QR code version together with the error correction level govern the data capacity of a QR code. There are forty different QR code versions and four error correction levels; namely, L (low), M (medium), Q (quartile) and H (high). Each error correction level corresponds to an error recovery threshold of approximately 7%, 15%, 25% and 30%, respectively. This means that if the amount of codeword error is within the threshold, the QR code can be decoded correctly.

2.2 Discrete Wavelet Transform (DWT)

The Discrete Wavelet Transform (DWT) is a technique that is widely used in image and signal processing. For digital images, the DWT technique involves the decomposition of an image into frequency channels of constant bandwidth on a logarithmic scale [15, 17].

When applying the DWT technique to a 2D image, the image is decomposed into four sub-bands, which are denoted as LL (low-low), LH (low-high), HL (high-low) and HH (high-high). Each sub-band can in turn be further decomposed at the next level, and this process can continue until the desired number of levels is reached. In view of the fact that the human visual system is more sensitive to the LL sub-band (i.e. the low frequency component), to maintain visual quality of the image despite the embedded watermark, information is typically embedded within one or more of the other three sub-bands [15]. Fig.

1 gives a depiction of how the DWT can decompose an image into sub-bands at 3 levels. For experiments in this paper, the watermark was embedded within the HH_3 sub-band.

2.3 Arnold Transform

The Arnold transform is an invertible transform that can be used for scrambling the pixels in a digital image. The transform scrambles the pixels within an image to disrupt the correlation between adjacent pixels. As such, the Arnold transform is commonly used as part of many watermarking schemes, as it distributes the pixels over the entire image [14]. The reason for doing this is so that any error introduced by distorting a watermarked image will be scattered over the image, and the watermark can still potentially be recovered despite the error. Eq. 1 shows the typical form of the Arnold transform used in watermarking techniques for digital images, where N is the size of the image. Applying the transform over a number of iterations will convert the original image into a chaotic image. The transform is invertible because the original image can be recovered from the chaotic image, by applying the same transform to the chaotic image for a number of iterations.

The purpose of employing this transform for the technique proposed in this paper, is due to the fact that adjacent pixels in image data have strong correlation to each other. By using the Arnold transform, this high pixel correlation will be disrupted. The Arnold transform is shown in Eq. 1 [10], where p and q are positive integers, $\det(A) = 1$, and (x', y') are the new coordinates of the pixel after Arnold transform is applied to a pixel at position (x, y). The period of the Arnold transform depends on p, q and the size N of the image. After several iterations of applying the transform, the correlation among adjacent pixels can be disturbed completely.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \mod N \tag{1}$$

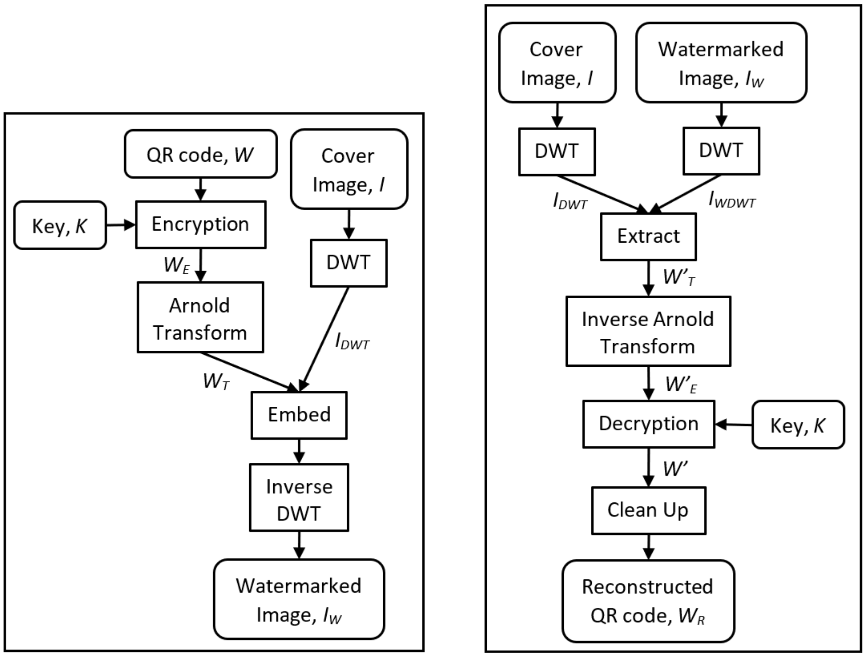
The Arnold Transform has also been used for image encryption [10]. The underlying notion for this is that the shuffling the pixels in the spatial domain confuses the relationship between the cipher image and the plain image. For image encryption, the parameters p, q and the number of iterations of applying the transform, can all be used as the secret keys.

3 Related Work

This section presents related work on the use of QR codes for data hiding and watermarking.

There have been a variety of different uses of QR codes in the area of computer security. In previous work, Chow et al. [6] proposed the use of QR codes for watermarking using two techniques in the frequency domain. Their proposed approach combined the use of the DWT with the Discrete Cosine Transform (DCT) for QR code watermarking. In other work on QR code watermarking, an authentication method for medical images using a QR code based on zero watermarking scheme was proposed [22]. In the scheme, a patient's identification details and a link their data was encoded in the form of a QR code which served as the watermark. Similarly, in recent work the use of QR codes to store injury information of patients and to embed it into medical images was proposed [2].

Kang et al. [14] proposed a watermarking approach based on the combination of DCT, QR codes and chaotic theory. In their approach, a QR code image is encrypted with a chaotic system to enhance the security of the watermark, before embedding it within DCT blocks after undergoing block based scrambling. In related work, a digital rights management method for protecting documents by repeatedly inserting a QR code into the DWT sub-band of a document was investigated [3]. The embedding of QR codes as watermarks for copyright protection of digital images has also been proposed in various other



(a) (b)

Figure 2: Overview of the QR code watermarking processes; (a) embedding process; (b) extraction process.

studies [1, 8, 18]. In addition, others have proposed QR code watermarking methods which use different approaches, such as, the incorporation of an attack detection feature to detect malicious interference by an attacker [23], or the embedding of QR code watermarks using a just noticeable difference model to increase imperceptibility [16]. Furthermore, unlike the common approach of using the RGB color space, watermarking can also be embedded using the YCbCr color space [20].

In other related work on QR codes for information security, Tkachenko et al. [24] described a modified QR code that could contain two storage levels. They called this a two-level QR code, as it had a public and a private storage level. The purpose of the two-level QR code was for document authentication. In addition, QR codes have also been used for secret sharing [5]. In this work, a method of distributing shares by embedding them into cover QR codes was proposed. These QR codes contained both public and private information, which allowed for the shares to be transmitted over public channels. The public information in the QR codes could be access by anyone, whereas only authorized individuals would be able to obtain the private information.

4 Proposed QR Code Watermarking Technique

The aim of the QR code watermarking technique proposed in this paper is to embed a QR code watermark within a cover image, and to be able to correctly extract the watermark. Fig. 3 depicts the processes involved in the embedding and extraction processes. Details of the processes will be described in the respective subsections to follow.

4.1 Embedding Process

An overview of the embedding process is depicted in Fig. 3(a). It can be seen from the figure that the embedding process accepts three inputs; a QR code, W, which is the watermark image; a key, K, for encryption; and a cover image, I. The output of the embedding process is a watermarked image, I_W.

It should be noted that K is a random bit string, which is used to encrypt and decrypt the watermark. The purpose of doing this is to ensure that even if an adversary can extract W, the adversary will not be able to obtain information about the contents of the watermark. The bits in K are to be XORed with the light and dark modules of W. As such, the length of the bit string must match the number of modules in W.

For experiment results presented in this paper, I was converted to DWT level 3 and the encrypted and scrambled watermark, W_T, was embedded within the HH_3 sub-band. The purpose of embedding information within the HH sub-band is due to the fact the human visual system is less sensitive to perturbations in this sub-band. The DWT coefficients C were modified based on Eq. 3 for the x and y pixels in W_T, where $W_{T,(x,y)} \in \pm 1$. The λ parameter can be adjusted to balance between watermark imperceptibility and robustness.

$$C'_{(x,y)} = C_{(x,y)} + \lambda W_{T,(x,y)} |C_{(x,y)}| \tag{2}$$

Prior to embedding the watermark, bits in the encrypted watermark are scrambled using Arnold transform. The reason for this is to distribute the watermark data across the entire image. In practice, this effectively reduces localized errors in the extracted watermark, which may result from distortions introduced to I_W by an adversary. For grayscale images, there is only one channel for watermarking. Whereas for color images, there are three channels, namely, the red (R), green (G), and blue (B) color channels, and the watermark is embedded into each channel. Algorithm 3 provides details of the steps involved in the embedding process.

Algorithm 1 Embedding algorithm

```
Input: A QR code, $W$, a cover image, $I$, and a key, $K$.
Output: A watermarked image, $I_W$

Step 1. Encrypt information in $W$ by XORing the random bits in $K$ with the modules in $W$ to produce $W_E$.
Step 2. Generate a chaotic image $W_T$ by scrambling the bits of the encrypted watermark $W_E$ using Arnold transform over a number of iterations.
Step 3. Convert $I$ to $I_{DWT}$ by performing DWT to the desired level.
Step 4. Embed $W_T$ in a $I_{DWT}$ sub-band.
Step 5. Generate the watermarked image $I_W$ by inversing DWT.
```

4.2 Extraction Process

Fig. 3(b) gives an overview of the extraction process, which is very much the reverse of the embedding process. To extract the watermark image, the extraction algorithm requires the original cover image, I; the watermarked image, I_W; and the key, K, for decryption. The output of the algorithm is the reconstructed watermark, i.e. a reconstructed QR code, W_R.

It should be noted that if I_W is distorted from attacks by an adversary, W' will result in a noisy image. Hence, a clean-up stage is required to restore the QR code. This is possible as long as certain information about the QR code is known; namely, the QR code version, error correction level, masking pattern and

Algorithm 2 Extraction algorithm

```
Input: The original cover image, $I$, the watermarked image, $I_W$, and and the key, $K$.
Output: A reconstructed QR code, $W_R$

Step 1. Convert $I$ to $I_{DWT}$, and $I_W$ to $I_{WDWT}$, by performing DWT on the cover image and watermarked image respectively.
Step 2. Extract $W'_T$ from differences in the specific sub-band ($HH_3$ in the experiments) of $I_{DWT}$ and $I_{WDWT}$.
Step 3. Generate $W'_E$ by inversing the Arnold transform.
Step 4. Decrypt $W'_E$ using $K$ to produce the extracted watermark image $W'$.
Step 5. Clean-up the $W'$ and restore the QR code function patterns to produce $W_R$.
```

number of pixels per module. With this information, restoring the modules involves counting the total number of black and white bits for every module in W'. If there are more white bits, set the module color to white, and vice versa. Also, to ensure that the QR code is decodable, the QR code function patterns, which may have been corrupted, are restored to produce the reconstructed QR code, W_R. For color images, since the watermark is embedded within each of the color channels, W' is extracted from the R, G, and B channels respectively. This results in W'_c, where $c \in \{r,g,b\}$. Thus, when reconstructing a module for color images, all module bits from W'_c are counted when determining the value for the restored module, i.e. considers the total number of black and white bits from all the color channels.

Any QR code reader should be able to decode W_R, as long as the error in W_R is below the error correction threshold of the QR code. Note that it is possible to only embed the data modules of W in I_W, since the function patterns are restored during the clean-up stage. However, in our experiments, we chose to embed the entire QR code because it provides information on the amount of noise in W', which results from distortions made to I_W. The steps involved in the extraction algorithm are provided in Algorithm 4.

5 Proposed QR Code Watermarking Technique

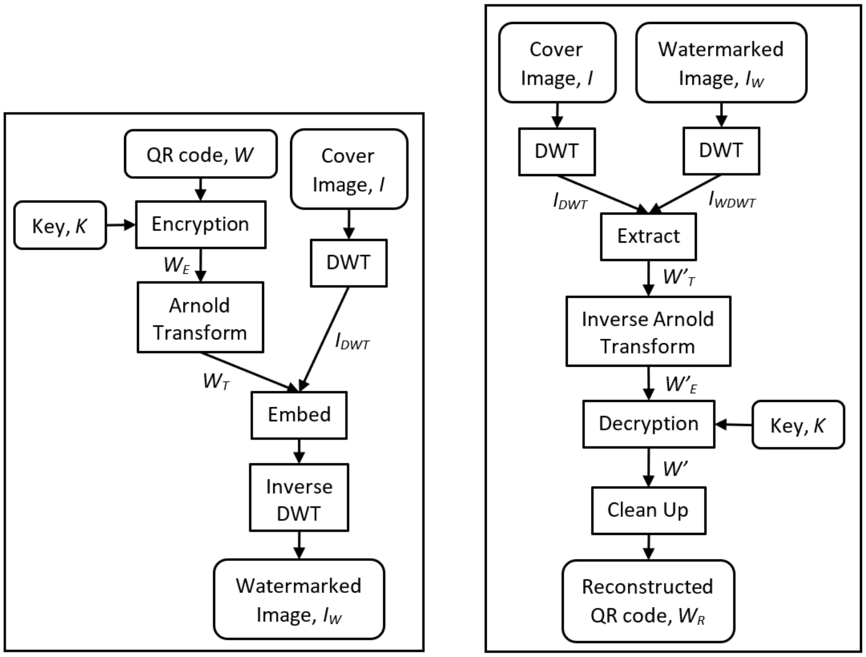
The aim of the QR code watermarking technique proposed in this paper is to embed a QR code watermark within a cover image, and to be able to correctly extract the watermark. Fig. 3 depicts the processes involved in the embedding and extraction processes. Details of the processes will be described in the respective subsections to follow.

5.1 Embedding Process

An overview of the embedding process is depicted in Fig. 3(a). It can be seen from the figure that the embedding process accepts three inputs; a QR code, W, which is the watermark image; a key, K, for encryption; and a cover image, I. The output of the embedding process is a watermarked image, I_W.

It should be noted that K is a random bit string, which is used to encrypt and decrypt the watermark. The purpose of doing this is to ensure that even if an adversary can extract W, the adversary will not be able to obtain information about the contents of the watermark. The bits in K are to be XORed with the light and dark modules of W. As such, the length of the bit string must match the number of modules in W.

For experiment results presented in this paper, I was converted to DWT level 3 and the encrypted and scrambled watermark, W_T, was embedded within the HH_3 sub-band. The purpose of embedding information within the HH sub-band is due to the fact the human visual system is less sensitive to perturbations in this sub-band. The DWT coefficients C were modified based on Eq. 3 for the x and



(a) (b)

Figure 3: Overview of the QR code watermarking processes; (a) embedding process; (b) extraction process.

y pixels in W_T, where $W_{T,(x,y)} \in \pm 1$. The λ parameter can be adjusted to balance between watermark imperceptibility and robustness.

$$C'_{(x,y)} = C_{(x,y)} + \lambda W_{T,(x,y)} |C_{(x,y)}| \tag{3}$$

Prior to embedding the watermark, bits in the encrypted watermark are scrambled using Arnold transform. The reason for this is to distribute the watermark data across the entire image. In practice, this effectively reduces localized errors in the extracted watermark, which may result from distortions introduced to I_W by an adversary. For grayscale images, there is only one channel for watermarking. Whereas for color images, there are three channels, namely, the red (R), green (G), and blue (B) color channels, and the watermark is embedded into each channel. Algorithm 3 provides details of the steps involved in the embedding process.

Algorithm 3 Embedding algorithm

```
Input: A QR code, $W$, a cover image, $I$, and a key, $K$.
Output: A watermarked image, $I_W$

Step 1. Encrypt information in $W$ by XORing the random bits in $K$ with the modules in $W$ to produce $W_E$.
Step 2. Generate a chaotic image $W_T$ by scrambling the bits of the encrypted watermark $W_E$ using Arnold transform over a number of iterations.
Step 3. Convert $I$ to $I_{DWT}$ by performing DWT to the desired level.
Step 4. Embed $W_T$ in a $I_{DWT}$ sub-band.
Step 5. Generate the watermarked image $I_W$ by inversing DWT.
```

5.2 Extraction Process

Fig. 3(b) gives an overview of the extraction process, which is very much the reverse of the embedding process. To extract the watermark image, the extraction algorithm requires the original cover image, I; the watermarked image, I_W; and the key, K, for decryption. The output of the algorithm is the reconstructed watermark, i.e. a reconstructed QR code, W_R.

Algorithm 4 Extraction algorithm

```
Input: The original cover image, $I$, the watermarked image, $I_W$, and and the key, $K$.
Output: A reconstructed QR code, $W_R$

Step 1. Convert $I$ to $I_{DWT}$, and $I_W$ to $I_{WDWT}$, by performing DWT on the cover image and watermarked image respectively.
Step 2. Extract $W'_T$ from differences in the specific sub-band ($HH_3$ in the experiments) of $I_{DWT}$ and $I_{WDWT}$.
Step 3. Generate $W'_E$ by inversing the Arnold transform.
Step 4. Decrypt $W'_E$ using $K$ to produce the extracted watermark image $W'$.
Step 5. Clean-up the $W'$ and restore the QR code function patterns to produce $W_R$.
```

It should be noted that if I_W is distorted from attacks by an adversary, W' will result in a noisy image. Hence, a clean-up stage is required to restore the QR code. This is possible as long as certain information about the QR code is known; namely, the QR code version, error correction level, masking pattern and number of pixels per module. With this information, restoring the modules involves counting the total number of black and white bits for every module in W'. If there are more white bits, set the module color to white, and vice versa. Also, to ensure that the QR code is decodable, the QR code function patterns, which may have been corrupted, are restored to produce the reconstructed QR code, W_R. For color images, since the watermark is embedded within each of the color channels, W' is extracted from the R, G, and B channels respectively. This results in W'_c, where $c \in \{r,g,b\}$. Thus, when reconstructing a module for color images, all module bits from W'_c are counted when determining the value for the restored module, i.e. considers the total number of black and white bits from all the color channels.

Any QR code reader should be able to decode W_R, as long as the error in W_R is below the error correction threshold of the QR code. Note that it is possible to only embed the data modules of W in I_W, since the function patterns are restored during the clean-up stage. However, in our experiments, we chose to embed the entire QR code because it provides information on the amount of noise in W', which results from distortions made to I_W. The steps involved in the extraction algorithm are provided in Algorithm 4.

6 Experiment Results and Discussion

This section presents results from experiments conducted to evaluate the proposed QR code watermarking technique. The experiments were performed using Matlab 2019b. 70 standard test images were used to evaluate the performance of the proposed watermarking technique. The color and grayscale images that were used in the experiments are shown in Fig. 4 and Fig. 5, respectively. The first 3 images, namely, the Lena, Peppers and Mandrill images, were specifically used to demonstrate results for individual images in the discussion to follow.

All images were 512 × 512 pixels in dimension. The color images contained 3 channels, i.e. red, green and blue, while grayscale images contained only a single intensity channel. A QR code version 1 with error correction level H was used in the experiments. This QR code version is made up of 21 × 21 modules. Since the HH_3 sub-band of a 512 × 512 image has a 64 × 64 resolution, each module in the

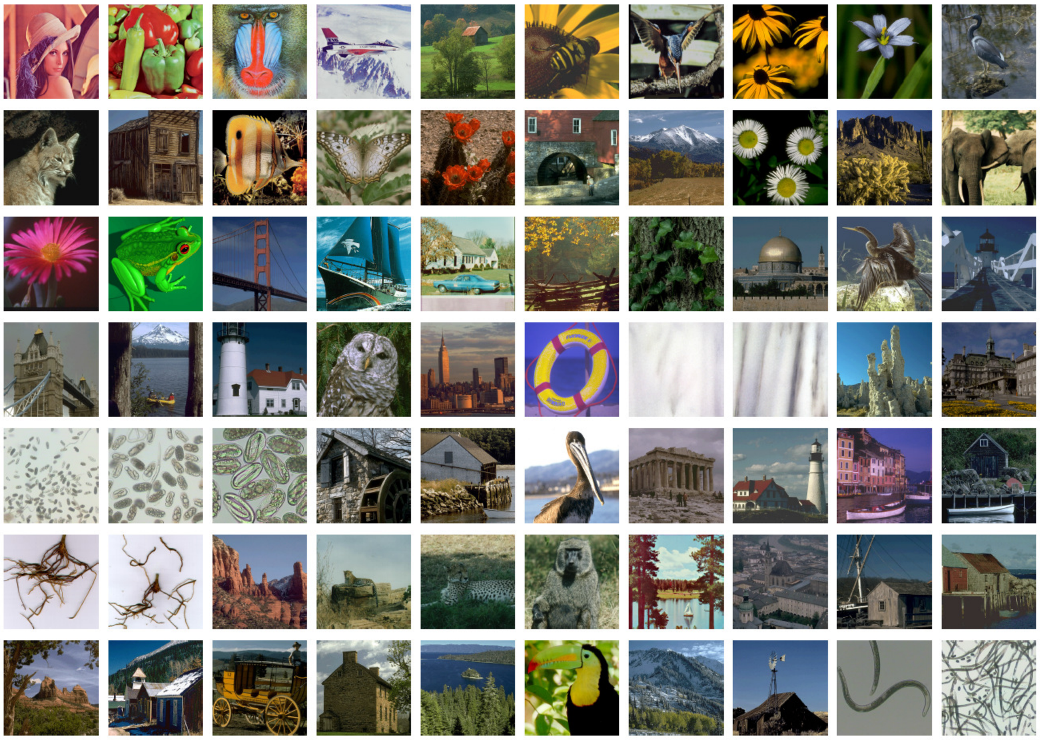


Figure 4: 70 color images that were used in the experiment.

Figure 5: 70 grayscale images that were used in the experiment.

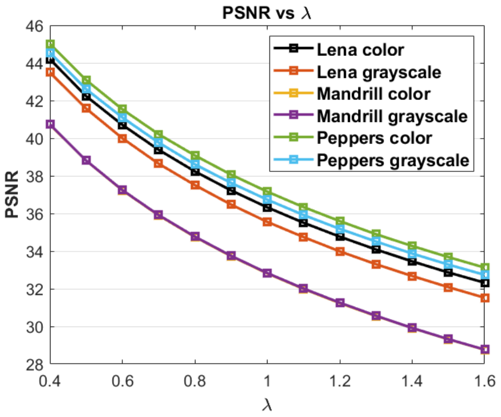


Figure 6: PSNR values vs watermarking strength λ.

QR code was converted to consist of 3 × 3 pixels, resulting in a total QR code size of 63 × 63 pixels. It should be noted that the watermark was embedded repeatedly into 3 channels for color images, while for grayscale images the watermark was only embedded in the single channel.

6.1 Imperceptibility

Imperceptibility is the degree at which a human cannot perceive the difference between the original and watermarked signals. The Peak Signal-to-Noise Ratio (PSNR) metric was used as a measure of image quality and to indicate the imperceptibility of distortions resulting from embedding a watermark within a cover image. PSNR is a commonly used metric to ascertain signal strength.

Fig. 6 shows a graph of the PSNR values that were obtained by varying the value of λ for the color and grayscale Lena, Peppers and Mandrill images. Greater PSNR values indicate less difference between I and I_W. At low λ values, the human visual system is less sensitive to distortions cause by embedding the watermark. However, increasing the value of λ increases the distortion in the resulting images. When the distortion is clearly visible in I_W, the image looses its commercial value and usefulness. Note that in Fig. 6, the data plots for the grayscale and color Mandrill images are overlapping, as the PSNR results for both images are almost the same.

6.2 Capacity

Capacity, or payload, is the amount of data that can be embedded by a watermarking scheme. The data capacity of the proposed watermarking technique is based on the capacity of the QR code version and error correction level of W. For a given QR code version, the higher the error correction level, the lower the data capacity, but the more robust the resulting watermark will be to errors. Hence, there is a trade-off between data capacity and watermark robustness.

In addition, the size of W, is also governed by the size of I, since the watermark is to be embedded within a DWT sub-band of I. The higher the number of module in W, the more data the QR code can encode. However, this also means that for the watermark to be able to fit within a DWT sub-band, less pixels may have to be used to encode each module. The lower the number of pixels per module, the less robust the watermark, because there is a higher potential for the pixels per module to be corrupted by distortions made to the watermarked image.

6.3 Robustness and Security

Robustness and security refer to a watermarking scheme's ability to withstand distortions to the watermarked signal. In the case of security, these distortions are intentional attacks by an adversary to impair the watermark [7, 19]. The robustness and security of the proposed technique were examined by applying various attacks to the watermarked images; namely, JPEG compression, sharpening, blurring, salt-and-pepper noise, and cropping. These are attacks that are typically used to evaluate watermarking techniques.

The distortions introduced to the watermarked images were performed by varying the parameters of the attacks within a certain range of values. For the JPEG compression attack, the images were compressed by varying the compression quality between 90 and 10. For the sharpening attack, unsharp masking was used with its strength ranging from 0.2 to 2.0. Gaussian filtering was performed for the blurring attack, with the value of the σ parameter varied from 0.2 to 2.0. In the salt-and-pepper noise attack, pixels in an image were randomly overwritten. The amount of pixels that were randomly overwritten were varied from 1% to 9% of the total number of pixels in the image. Two versions of cropping attacks were implemented, which were named the cropping V1 attack and cropping V2 attack. In the cropping V1 attack, the image center was removed, while in the cropping V2 attack, the corners of the image were removed. In both cropping attacks, the amount of cropping ranged from 5% to 40% of the image size.

To evaluate the amount of error in the extracted watermark and the reconstructed QR code, the Bit Error Rate (BER), Module Error Rate (MER) and Codeword Error Rate (CER) metrics were used. The BER refers to the percentage of bits that were in error in the extracted watermark, W', whereas the MER is the percentage of incorrect QR code modules in the reconstructed QR code, W_R. CER on the other hand, refers to the percentage of erroneous codewords in the QR code. Note that a codeword is composed of 8 modules. In a QR code with the error correction level H, the data in the QR code can potentially be recovered despite approximately 30% erroneous codewords. This means that the reconstructed QR code can be correctly decoded as long as the CER is less than the error recovery threshold.

To illustrate the various attacks, Tables 1, 2, 3, 4, 5 and 6 present results from the experiments on the grayscale and color versions of the Lena, Peppers and Mandrill images, respectively. These results were obtained using $\lambda = 1.0$. The values of the attack parameter that was used for a particular image are shown in the parentheses following the attack name in the tables. For example, the compression attack results shown in the tables used a JPEG quality of 50. For each test image and attack, the tables show the extracted watermark image with the BER, as well as the reconstructed QR code with the MER and CER. As described in Section 5.2, the reconstructed QR code, W_R, was obtained after cleaning up the noise in W'. In addition, gray modules in the reconstructed QR code depict the modules that were incorrectly recovered. It should be noted that color images contain 3 channels: R, G and B, where 3 versions of the watermark can be extracted. As such, the information from all 3 versions of the watermark are combined when reconstructing the QR code.

Two observations can be made from examining the results in Tables 1, 2, 3, 4, 5 and 6. First, watermarked color images were more robust against attacks compare with the same watermark embedded within the corresponding grayscale images. This can be deduced from the fact that the reconstructed QR code from the watermarked color images have lower CER values. The reason for this is because the watermark is embedded in the 3 separate color channels of color images, while it is only embedded within the single intensity channel of the grayscale images. Thus, this results in a higher chance for the QR code to be reconstructed correctly, since there are 3 copies of the information.

The second observation is that watermarks in images with higher entropy are more robust than watermarks in images with less entropy. This can be seen from the lower CER values resulting from both the grayscale and color images of the Mandrill image, as compared with results of the Lena and Peppers

Table 1: Results on the grayscale Lena image.

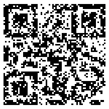
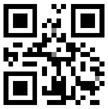
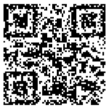
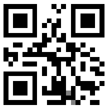
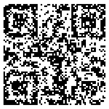
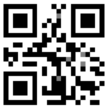
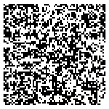
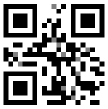
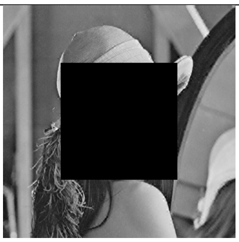
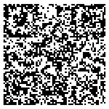
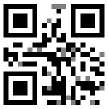
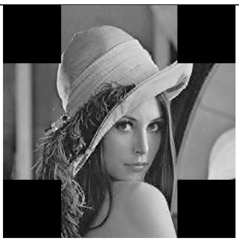
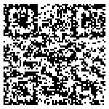
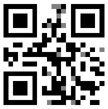
Attack	**Attacked Image**	**Extracted Watermark, W'**	**Reconstructed QR Code, W_R**	**Metrics**
Compression (50)				BER: 19.43% MER: 1.44% CER: 11.54%
Sharpening (1.0)				BER: 19.25% MER: 0.96% CER: 3.85%
Blurring (1.0)				BER: 21.59% MER: 2.40% CER: 15.38%
Noise (2%)				BER: 30.64% MER: 5.77% CER: 38.46%
CroppingV1 (25%)				BER: 28.92% MER: 10.10% CER: 61.54%
CroppingV2 (25%)				BER: 25.65% MER: 6.73% CER: 46.15%

Table 2: Results on the color Lena image.

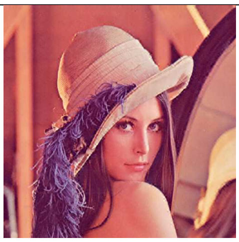
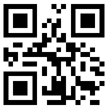
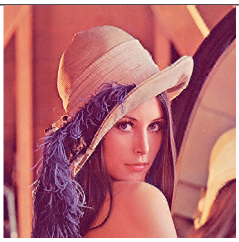
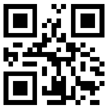
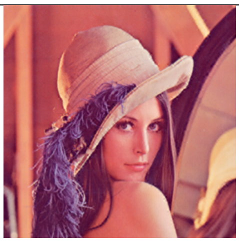
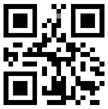
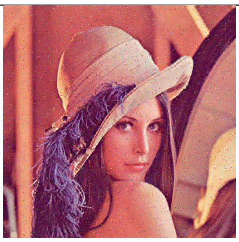
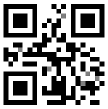
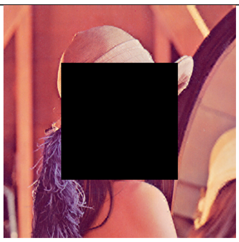
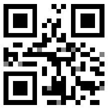
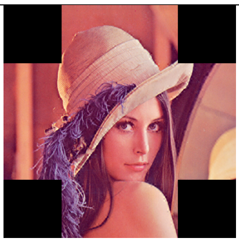
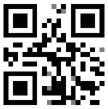
Attack	**Attacked Image**	**Extracted Watermark, W'**	**Reconstructed QR Code, W_R**	**Metrics**
Compression (50)		R: G: B:		BER$_R$:22.07% BER$_G$:18.09% BER$_B$:23.41% MER: 0.00% CER: 0.00%
Sharpening (1.0)		R: G: B:		BER$_R$:14.41% BER$_G$:13.61% BER$_B$:10.61% MER: 0.00% CER: 0.00%
Blurring (1.0)		R: G: B:		BER$_R$:22.83% BER$_G$:20.38% BER$_B$:21.67% MER: 0.48% CER: 3.85%
Noise (2%)		R: G: B:		BER$_R$:32.07% BER$_G$:28.87% BER$_B$:29.38% MER: 2.88% CER: 23.08%
CroppingV1 (25%)		R: G: B:		BER$_R$:29.98% BER$_G$:27.51% BER$_B$:27.21% MER: 4.81% CER: 38.46%
CroppingV2 (25%)		R: G: B:		BER$_R$:26.63% BER$_G$:24.72% BER$_B$:24.69% MER: 3.85% CER: 26.92%

Table 3: Results on the grayscale Peppers image.

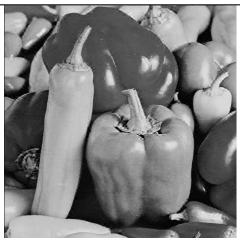
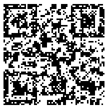
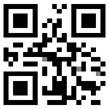
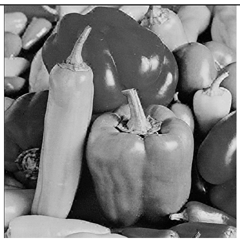
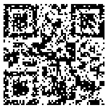
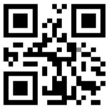
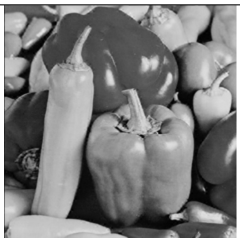
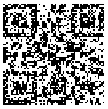
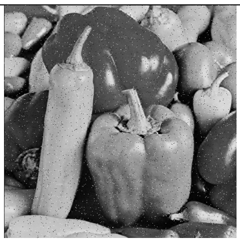
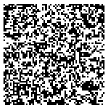
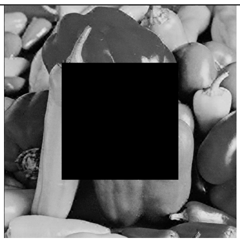
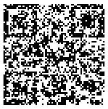
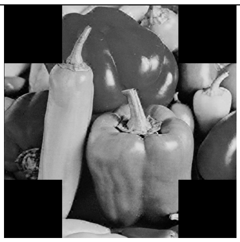
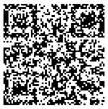
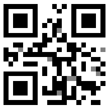
Attack	**Attacked Image**	**Extracted Watermark, W'**	**Reconstructed QR Code, W_R**	**Metrics**
Compression (50)				BER: 19.38% MER: 1.44% CER: 11.54%
Sharpening (1.0)				BER: 18.01% MER: 0.48% CER: 3.85%
Blurring (1.0)				BER: 22.90% MER: 1.44% CER: 11.54%
Noise (2%)				BER: 31.65% MER: 5.29% CER: 38.46%
CroppingV1 (25%)				BER: 25.98% MER: 5.29% CER: 34.62%
CroppingV2 (25%)				BER: 25.45% MER: 4.33% CER: 30.77%

Table 4: Results on the color Peppers image.

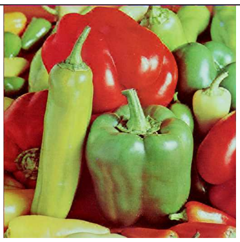
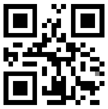
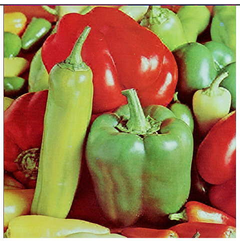
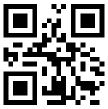
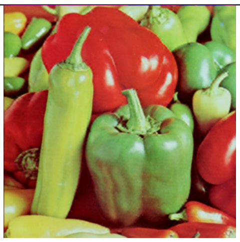
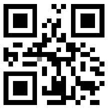
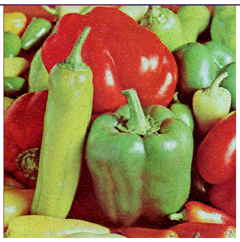
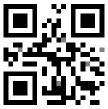
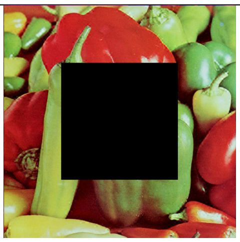
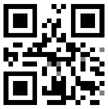
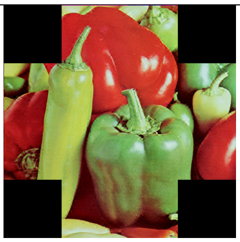
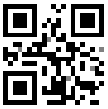
Attack	**Attacked Image**	**Extracted Watermark, W'**	**Reconstructed QR Code, W_R**	**Metrics**
Compression (50)		R: G: B:		BER_R:25.32% BER_G:19.55% BER_B:24.67 MER: 0.48% CER: 3.85%
Sharpening (1.0)		R: G: B:		BER_R:12.93% BER_G:13.88% BER_B:10.96% MER: 0.00% CER: 0.00%
Blurring (1.0)		R: G: B:		BER_R:22.55% BER_G:22.80% BER_B:21.57% MER: 0.48% CER: 3.85%
Noise (2%)		R: G: B:		BER_R:32.05% BER_G:29.35% BER_B:30.16% MER: 2.40% CER: 19.23%
CroppingV1 (25%)		R: G: B:		BER_R:26.53% BER_G:25.30% BER_B:25.85% MER: 2.88% CER: 23.08%
CroppingV2 (25%)		R: G: B:		BER_R:24.67% BER_G:25.07% BER_B:24.69% MER: 2.40% CER: 19.23%

Table 5: Results on the grayscale Mandrill image.

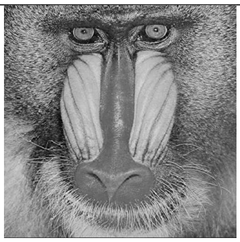
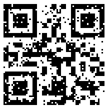
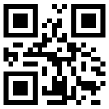
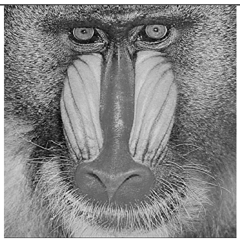
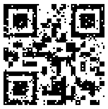
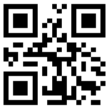
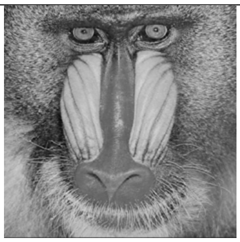
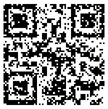
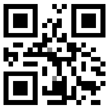
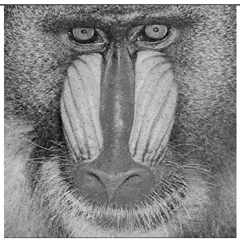
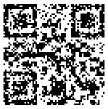
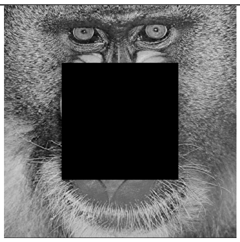
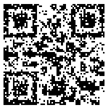
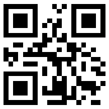
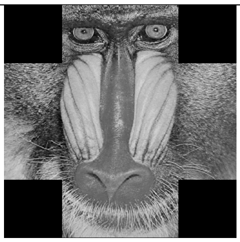
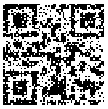
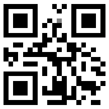
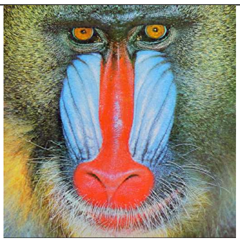
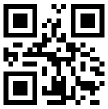
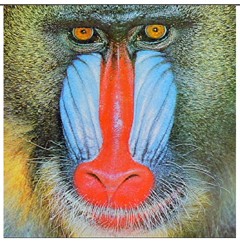
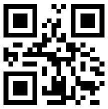
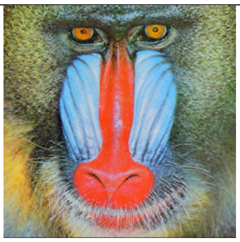
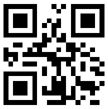
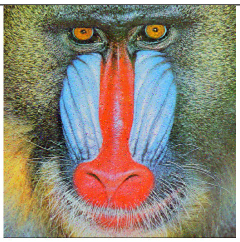
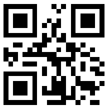
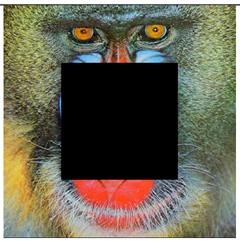
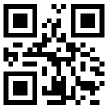
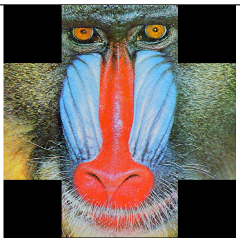
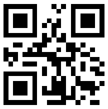
Attack	**Attacked Image**	**Extracted Watermark, W'**	**Reconstructed QR Code, W_R**	**Metrics**
Compression (50)				BER: 7.28% MER: 0.00% CER: 0.00%
Sharpening (1.0)				BER: 7.21% MER: 0.00% CER: 0.00%
Blurring (1.0)				BER: 16.00% MER: 0.96% CER: 7.69%
Noise (2%)				BER: 17.18% MER: 1.44% CER: 11.54%
CroppingV1 (25%)				BER: 16.28% MER: 0.00% CER: 0.00%
CroppingV2 (25%)				BER: 16.43% MER: 0.48% CER: 3.85%

Table 6: Results on the color Mandrill image.

Attack	**Attacked Image**	**Extracted Watermark, W'**	**Reconstructed QR Code, W_R**	**Metrics**
Compression (50)		R: G: B:		BER_R:14.01% BER_G:6.20% BER_B:17.81% MER: 0.00% CER: 0.00%
Sharpening (1.0)		R: G: B:		BER_R:5.64% BER_G:5.42% BER_B:3.96% MER: 0.00% CER: 0.00%
Blurring (1.0)		R: G: B:		BER_R:16.55% BER_G:15.44% BER_B:16.83% MER: 0.00% CER: 0.00%
Noise (2%)		R: G: B:		BER_R:18.87% BER_G:16.07% BER_B:15.34% MER: 0.00% CER: 0.00%
CroppingV1 (25%)		R: G: B:		BER_R:17.56% BER_G:16.02% BER_B:15.97% MER: 0.00% CER: 0.00%
CroppingV2 (25%)		R: G: B:		BER_R:17.38% BER_G:16.02% BER_B:16.60% MER: 0.00% CER: 0.00%

images. The reason for this is because images with higher entropy contain more variations in the high frequency components and this condition is favorable for the proposed method. More specifically, the proposed method embeds watermarks within the DWT high frequency sub-band, i.e. HH_3. Thus, the robustness of watermarks can be increased if the HH_3 sub-band contains a greater degree of variations. Conversely, if the coefficients in the HH_3 are all zeros, watermarks cannot be embedded at all based on Eq. 3.

To further investigate the accuracy of the two observations, Fig. 7 shows a comparison between the robustness of using grayscale and color images of the Lena, Peppers and Mandrill against various attacks. An exception occurs for the blurring attack, in which the lower entropy images, i.e. the Lena and Peppers images, may be more robust than the higher entropy Mandrill image. However, this exception is due to the characteristic of the blurring attack, which imposes more damage to images with higher entropy. For the other attacks, i.e. the JPEG attack, sharpening attack, etc., the two observations are shown to be true in general.

From Fig. 7, it can be seen that the proposed method is most robust against image sharpening attacks. This is because all CER values are within the recovery threshold, with the majority of them being 0. On the other hand, noise attacks severely damage the recovered watermark. Watermarks in the grayscale Peppers image cannot even withstand 1% of noise made to the image.

Another interesting finding from Fig. 7 is that watermarks in the grayscale and color Lena are more robust to the cropping V1 attack than the cropping V2 attack. An explanation to this is due to the fact that the Lena image contains higher entropy in the center of the image, compare with the four image corners. Thus, cropping the center part of the image would damage the watermark more severely than cropping the corners of the image. This supports the previously discussed observation that the proposed method favors images with higher entropy.

From the discussion, the following four characteristics were observed about the proposed method:

C1: the watermarks were more robust in color images than in grayscale images;
C2: the watermarks were more robust in images with higher entropy;
C3: the watermarks were most robust against the sharpening attack;
C4: the watermarks were least robust against the noise attack.

As observations about characteristics C1 and C4 were made from results of 3 test images, the experiments were extended to all 70 of the grayscale and color images, previously shown in Fig. 4 and Fig. 5, to see whether these observations still held to be true. The minimum, average and maximum CER values of the reconstructed QR codes resulting from the watermarked images that were tested using the various attacks are shown in Fig. 8 and Fig. 9, for grayscale and color images respectively.

The four characteristics, C1 to C4, are discussed based on the average CER results. It can be seen that C1 holds true since all average CER values in the results for color images are lower than the value for the corresponding grayscale images. To confirm whether C2 holds true, the 10 most robust and 10 least robust cover images, in which the watermarks were the most robust and least robust, are shown in Table 7. These images were obtained by building two lists of images based on their robustness to the various types of attacks. As shown in Table 7, the 10 most robust color and grayscale cover images contain higher entropy as compare with the 10 least robust cover images. This suggests that the observation of C2 is valid.

The accuracy of C3 holds true as it can be seen from Fig. 8 and Fig. 9 that the average CER values of the grayscale and color cover images for the sharpening attack are consistently below 20% as compare with other attacks. The observation of C4 is valid for grayscale cover images, as the average CER value of the grayscale cover images is over 30% even for a 1% noise attack. This means that on average, the QR code cannot be correctly decoded for noise attack. However, the observation of C4 is not obvious for color cover images. As such, it was concluded that C4 may only be partially true under certain

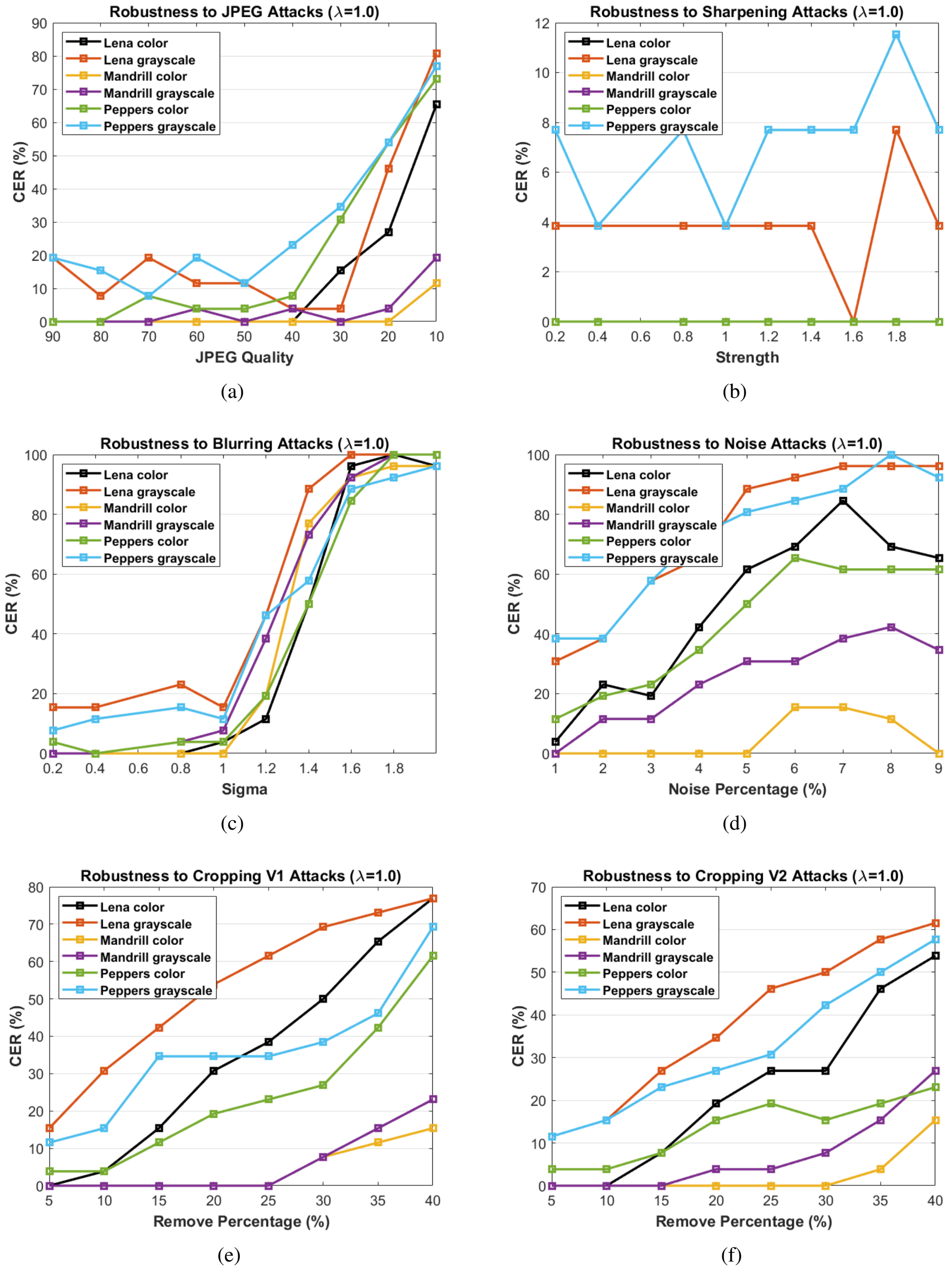
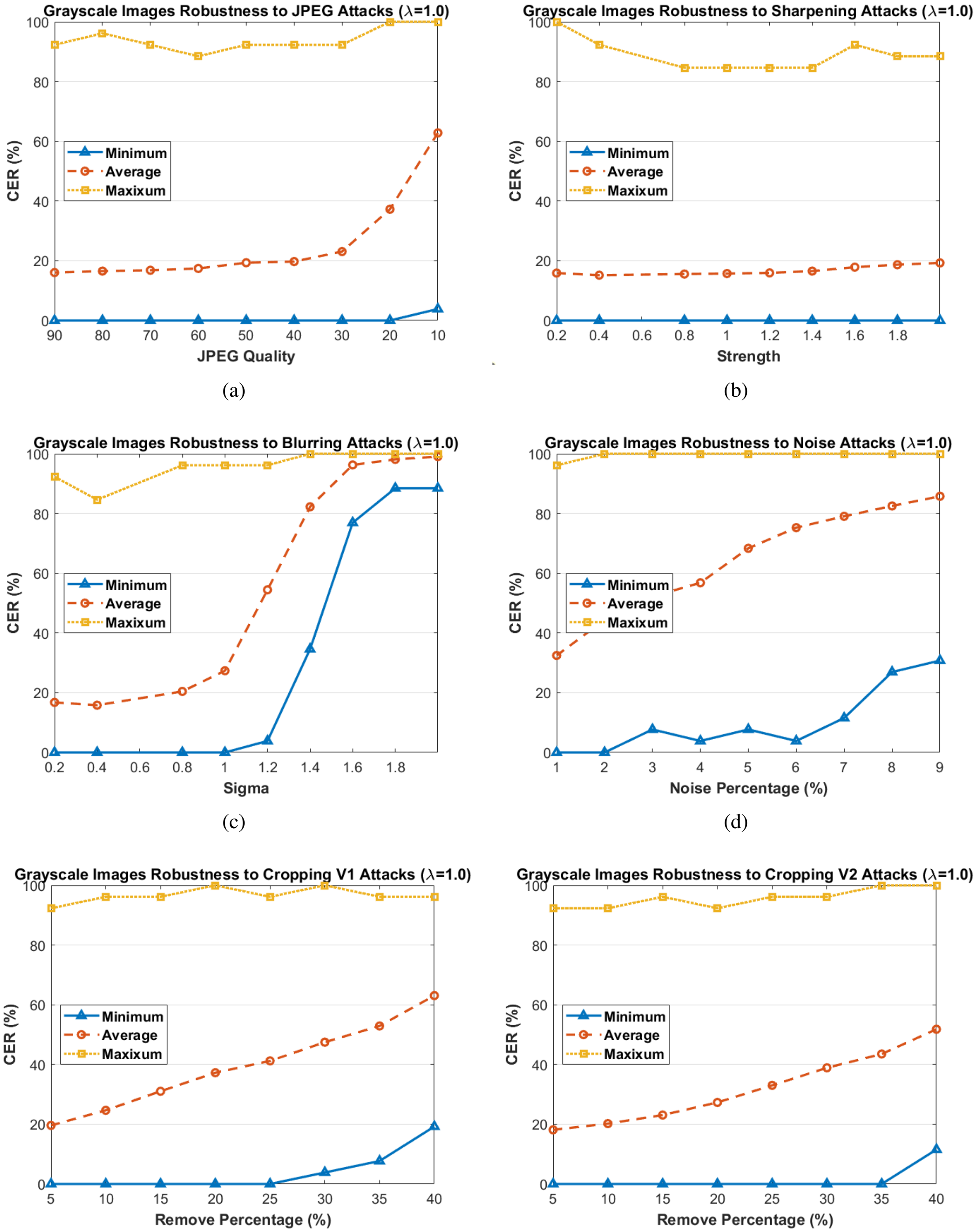


Figure 7: Comparisons of the robustness of watermarks between different images; (a) JPEG compression attack; (b) sharpening attack; (c) blurring attack; (d) noise attack; (e) cropping V1 attack; (f) cropping V2 attack.



(a)

(b)

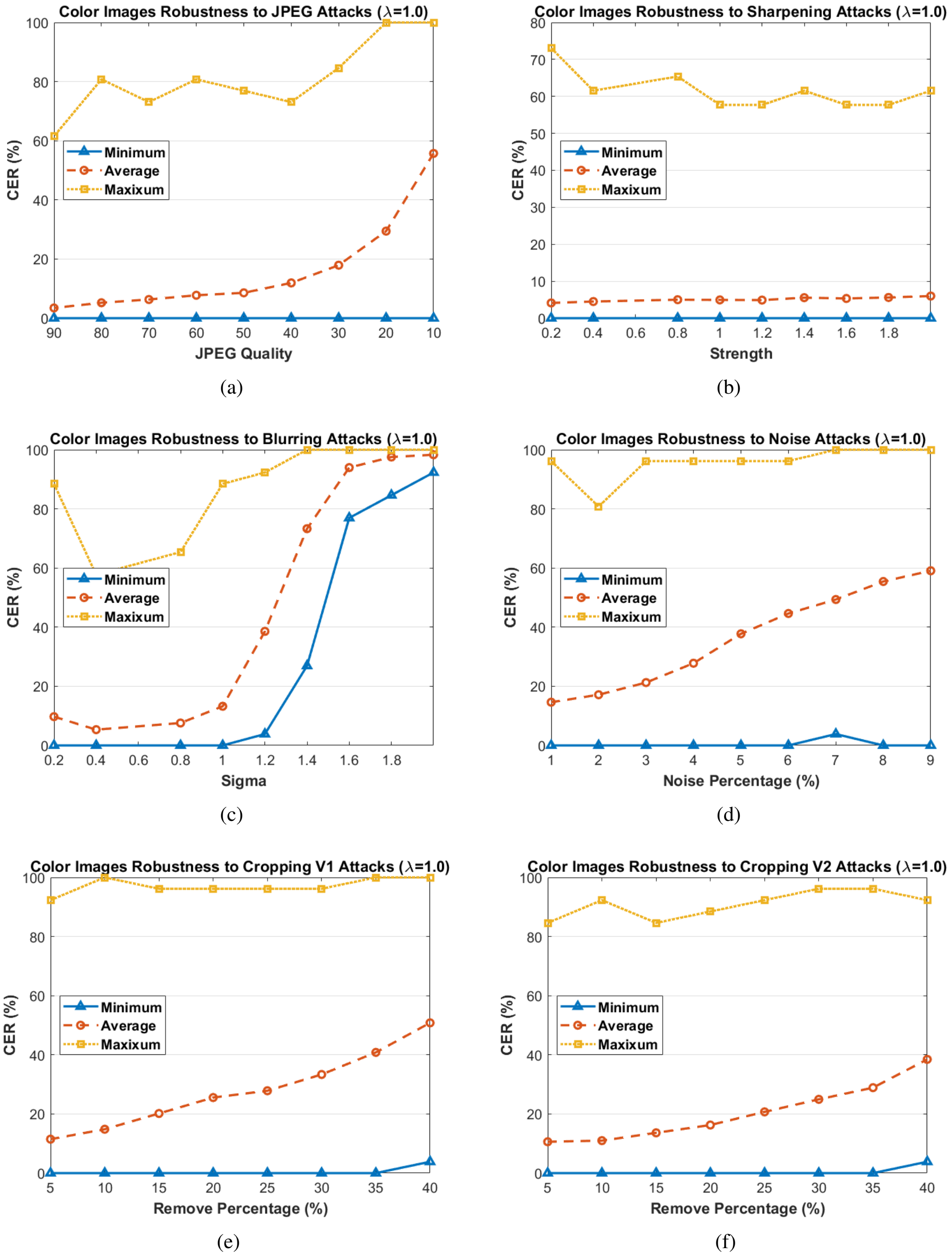
(c)

(d)

(e)

(f)

Figure 8: Codeword error rates resulting from attacks against the 70 grayscale images; (a) JPEG compression attack; (b) sharpening attack; (c) blurring attack; (d) noise attack; (e) cropping V1 attack; (f) cropping V2 attack.



(a)

(b)

(c)

(d)

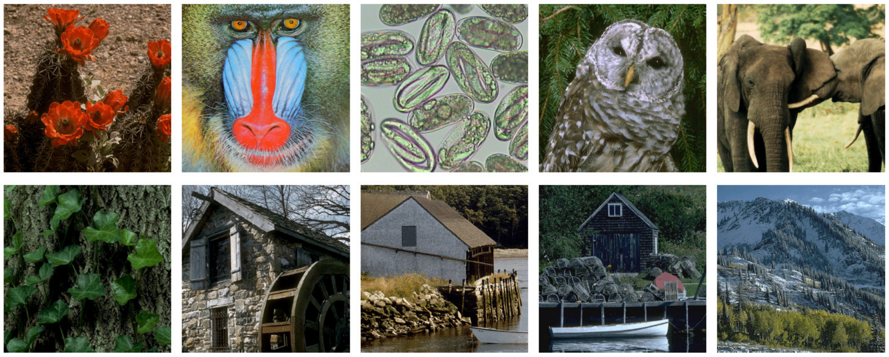
(e)

(f)

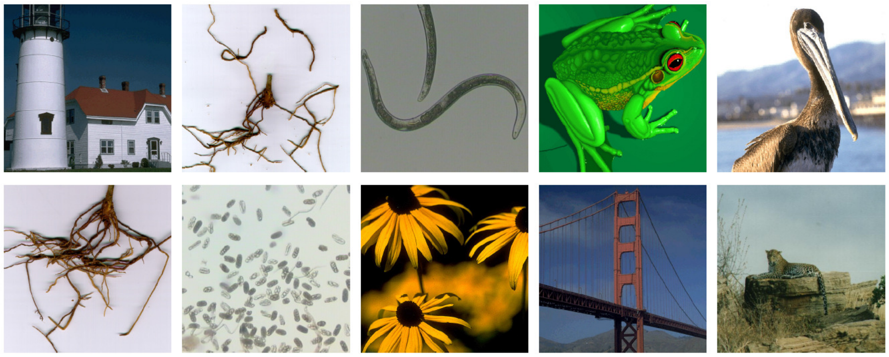
Figure 9: Codeword error rates resulting from attacks against the 70 color images; (a) JPEG compression attack; (b) sharpening attack; (c) blurring attack; (d) noise attack; (e) cropping V1 attack; (f) cropping V2 attack.

Table 7: The 10 most robust and least robust cover images.

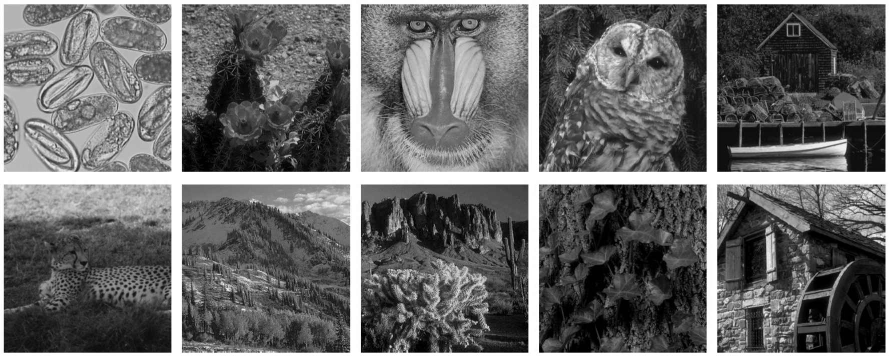
10 most robust color cover images:



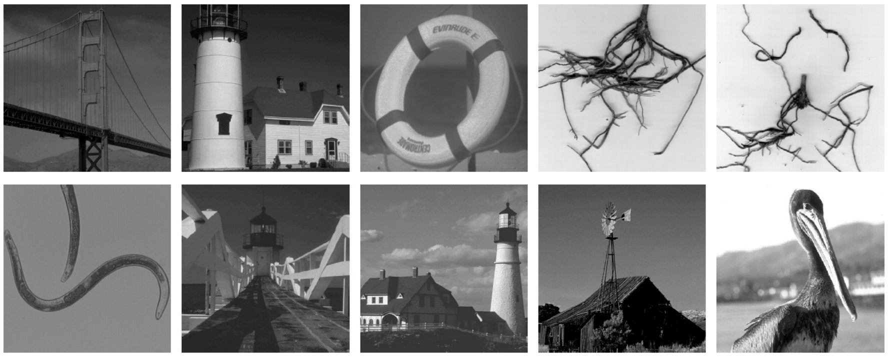
10 least robust color cover images:



10 most robust grayscale cover images:



10 least robust grayscale cover images:



circumstances.

It can also be seen from the results that there are consistent gaps between the maximum and minimum CER values for all grayscale and color cover images. This suggests that the performance of the proposed method differs significantly based on the visual characteristics of the cover images. This observation affirms the validity of C2.

6.4 Limitations

As previously discussed, a limitation of the proposed method is that its performance relies on characteristics of the cover images, which is not uncommon for watermarking techniques. For example, if the cover image has low entropy, the CER value of reconstructed QR code may exceed the error correction threshold even for weak attacks. Consequently, cover images with high entropy are favorable in the proposed method. Another potential limitation is the robustness against noise attacks using grayscale cover images. To overcome this, watermarks can be embedded into the corresponding color images, or the embedding strength of the watermarks can be increased, i.e. using a larger λ value.

7 Conclusion

This paper presents a QR code watermarking technique for digital images. The objective of the proposed watermarking technique is to embed a QR code within a cover image in an imperceptible manner. This was achieved by embedding a QR code within one of the cover image's high frequency DWT sub-bands. The reason for using a QR code as a watermark is because the QR code structure incorporates an error correction mechanism that allows it to be correctly decoded even if it contains some error. In this paper, the properties and characteristics of the proposed watermarking technique were discussed. In addition, this paper demonstrated the robustness of the proposed technique against common attacks that may be conducted by an adversary. The experiment results showed that the proposed method are more robust when using color cover images compare with grayscale images. Furthermore, cover images with high entropy images produced in better robustness results.

Acknowledgments

The authors would like to thank the anonymous reviewers for their invaluable feedback and suggestions on improving this paper.

References

[1] E. Avila-Domenech and A. Soria-Lorente. Watermarking based on krawtchouk moments for handwritten document images. In *Proc. of the 2018 International Workshop on Artificial Intelligence and Pattern Recognition (IWAIPR'18), Havana, Cuba*, volume 11047 of *Lecture Notes in Computer Science*, pages 122–129. Springer, Cham, September 2018.

[2] A. Boonyapalanant, M. Ketcham, and M. Piyaneeranart. Hiding patient injury information in medical images with QR code. In *Proc. of the 2019 International Conference on Computing and Information Technology (IC2IT'19), Bangkok, Thailand*, volume 936 of *Advances in Intelligent Systems and Computing*, pages 258–267. Springer, Cham, July 2019.

[3] N. Cardamone and F. d'Amore. DWT and QR code based watermarking for document DRM. In *Prof. of the 17th International Workshop on Digital Forensics and Watermarking (IWDW'18), Jeju Island, Korea*, volume 11378 of *Lecture Notes in Computer Science*, pages 137–150. Springer, Cham, October 2018.

[4] Y. Chow, W. Susilo, J. Baek, and J. Kim. QR code watermarking for digital images. In *Proc. of the 20th World Conference on Information Security Applications (WISA'19), Jeju Island, Korea*, volume 11897 of *Lecture Notes in Computer Science*, pages 25–37. Springer, Cham, August 2020.

[5] Y. Chow, W. Susilo, J. Tonien, E. Vlahu-Gjorgievska, and G. Yang. Cooperative secret sharing using QR codes and symmetric keys. *Symmetry*, 10(4):95, 2018.

[6] Y. Chow, W. Susilo, J. Tonien, and W. Zong. A QR code watermarking approach based on the DWT-DCT technique. In *Proc. of the 22nd Australasian Conference on Information Security and Privacy (ACISP'17), Auckland, New Zealand*, volume 10343 of *Lecture Notes in Computer Science*, pages 314–331. Springer, Cham, July 2017.

[7] I. J. Cox and M. L. Miller. Electronic watermarking: the first 50 years. In *Proc. of the 4th IEEE Workshop on Multimedia Signal Processing (Cat. No.01TH8564), Cannes, France*. IEEE, October 2001.

[8] Q. B. Dang, K. Louisa, M. Coustaty, M. M. Luqman, and J.-M. Ogier. A blind document image watermarking approach based on discrete wavelet transform and QR code embedding. In *Proc. of the 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW'19), Sydney, NSW, Australia*. IEEE, September 2019.

[9] Denso Wave Incorporated. QRcode.com. `http://www.qrcode.com/en/` [Online; Accessed on June 17, 2021].

[10] Z.-H. Guan, F. Huang, and W. Guan. Chaos-based image encryption algorithm. *Physics Letters A*, 346(1–3):153–157, 2005.

[11] F. Hartung and M. Kutter. Multimedia watermarking techniques. *Proceedings of the IEEE*, 87(7):1079–1107, July 1999.

[12] J. Huang, Y. Q. Shi, and Y. Shi. Embedding image watermarks in dc components. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(6):974–979, September 2000.

[13] International Organization for Standardization. Information technology — automatic identification and data capture techniques — QR code 2005 bar code symbology specification. ISO/IEC 18004:2006, 2006. `http://www.iso.org/` [Online; Accessed on June 17, 2021].

[14] Q. Kang, K. Li, and J. Yang. A digital watermarking approach based on DCT domain combining QR code and chaotic theory. In *Proc. of the 11th International Conference on Wireless and Optical Communications Networks (WOCN'14), Vijayawada, India*. IEEE, September 2014.

[15] C. C. Lai and C. C. Tsai. Digital image watermarking using discrete wavelet transform and singular value decomposition. *IEEE Transactions on Instrumentation and Measurement*, 59(11):3060–3063, November 2010.

[16] H.-C. Lee, C.-R. Dong, and T.-M. Lin. Digital watermarking based on JND model and QR code features. In *Advances in Intelligent Systems and Applications – Volume 2*, pages 141–148. Springer, 2013.

[17] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[18] Z. Meng, T. Morizumi, S. Miyata, and H. Kinoshita. Design scheme of copyright management system based on digital watermarking and blockchain. In *Proc. of the 42nd IEEE Annual Computer Software and Applications Conference (COMPSAC'18), Tokyo, Japan*, volume 2, pages 359–364. IEEE, July 2018.

[19] A. S. Panah, R. V. Schyndel, T. Sellis, and E. Bertino. On the properties of non-media digital watermarking: A review of state of the art techniques. *IEEE Access*, 4:2670–2704, 2016.

[20] C. Patvardhan, P. Kumar, and C. V. Lakshmi. Effective color image watermarking scheme using YCbCr color space and QR code. *Multimedia Tools and Applications*, 77(10):12655–12677, 2018.

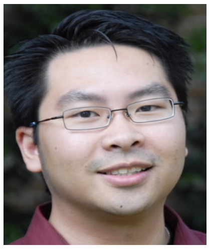
[21] C. I. Podilchuk and E. J. Delp. Digital watermarking: algorithms and applications. *IEEE Signal Processing Magazine*, 18(4):33–46, July 2001.

[22] V. Seenivasagam and R. Velumani. A QR code based zero-watermarking scheme for authentication of medical images in teleradiology cloud. *Computational and Mathematical Methods in Medicine*, 2013:516465:1–516465:16, 2013.

[23] P. P. Thulasidharan and M. S. Nair. QR code based blind digital image watermarking with attack detection code. *AEU - International Journal of Electronics and Communications*, 69(7):1074–1084, 2015.

[24] I. Tkachenko, W. Puech, C. Destruel, O. Strauss, J. Gaudin, and C. Guichard. Two-level QR code for private message sharing and document authentication. *IEEE Transactions on Information Forensics and Security*, 11(3):571–583, 2016.

Author Biography



Yang-Wai Chow received his Ph.D. in computer science from Monash University, Australia. He is currently an Associate Professor in the School of Computing and Information Technology, at the University of Wollongong, Australia, and a member of the Institute of Cybersecurity and Cryptology. His main research interests include virtual reality, multimedia security and cyber security. His work has been published in many international conferences and journals. He serves as a reviewer for several international journals and conferences.

Willy Susilo obtained his Bachelor Degree in Computer Science from Universitas Surabaya, Indonesia with a "Summa Cum Laude" predicate. He received his Master and Doctor of Philosophy degrees from the University of Wollongong (UOW). His main research interest include cryptography and cyber security. He received a prestigious ARC Future Fellowship from the Australian Research Council. He also received the UOW Researcher of the Year 2016 due to his research excellence. He is the Director of Institute of Cybersecurity and Cryptology, UOW.

Joonsang Baek received his Ph.D. degree in computer science from Monash University, Australia, in 2004. He received the M.S. in computer engineering from Korea Advanced Institute of Science and Technology-IT Convergence Campus (KIAST-ICC), Korea, in 2000, and received the B.S in mathematics from Pohang University of Science and Technology (POSTECH), Korea, in 1998. He is currently a senior lecturer at School of Computing and Information Technology, University of Wollongong, Australia. His current research interests are in the field of applied cybersecurity and cryptography. Joonsang has published his work in a number of reputable journals and conference proceedings and has served as chairs and program committee members for a number of renowned conferences and workshops.

Jongkil Kim received his PhD in Computer Science from the University of Wollongong (UOW), Wollongong, Australia in 2016. He is currently a lecturer in the School of Computing and Information Technology and a member of Institute of Cybersecurity and Cryptology in UOW. His main research interest is in the area of applied cryptography and cybersecurity including public key cryptography and security protocols. He has authored multiple publications in prestigious conferences and journals in an information security area.

Wei Zong is a PhD student in the Institute of Cybersecurity and Cryptology at the University of Wollongong, Australia. His research interests include machine learning, adversarial attacks, network intrusion detection and visualization. He has published several papers in these areas.