Physical Anti-copying Semi-robust Random Watermarking for QR Code

Jiale Chen[1], Li Dong[1,2(✉)], Rangding Wang[1], Diqun Yan[1], Weiwei Sun[3], and Hang-Yu Fan[3]

[1] Department of Computer Science, Ningbo University, Zhejiang, China
{2111082075,dongli,wangrangding,yandiqun}@nbu.edu.cn
[2] The Key Lab of Mobile Network Application Technology of Zhejiang Province, Zhejiang, China
[3] Alibaba Group, Zhejiang, China
{sunweiwei.sww,hangyu.fhy}@alibaba-inc.com

Abstract. Recently, QR code has been applied in anti-counterfeiting scenarios, where a unique QR code is attached for a specific item. However, such a QR code-based anti-counterfeiting solution cannot resolve the physical illegal copying issue. The genuine QR code can be physically replicated by scanning and printing. In this work, we propose a physical anti-copying semi-robust randomly watermarking system for QR code. Specifically, the authentic and counterfeit channels a QR code experiences are investigated first. By exploiting the distortion characteristics between two channels, we devise a randomly watermark embedding system, where the watermark bit is embedded via modulating the relationship between two carefully selected transformed coefficients. Finally, to obtain a valid and recognizable binary QR code image, a random binarization procedure is applied, and the regions originally belonging to the white module are erased. The final resultant watermark appears as *white-dot pattern* resides the black module of QR code, which is robust to the authentic print-scan but fragile to the physically illegal copying. Experimental results demonstrate the effectiveness of the proposed watermarking system. This work makes the first step towards exploring semi-robust watermarking for combating physically illegal copying.

Keywords: QR code · Semi-robust watermark · Physical anti-copying

1 Introduction

Counterfeiting is a criminal offense that involves the fraudulent production and distribution of an item similar to a genuine product. The production, distribution, and sale of counterfeit items not only defrauds those buying the items but also steals profits from the owners and distributors of the genuine articles. To combat the widespread counterfeiting issue, anti-counterfeiting marks can be attached to the genuine product as accessories or printing on the package surface. Traditional anti-counterfeiting countermeasures including micro-text [2], special color

X. Zhao et al. (Eds.): IWDW 2022, LNCS 13825, pp. 131–146, 2023.
https://doi.org/10.1007/978-3-031-25115-3_9

bar printing [15], thermal ink [8], RFID tags [17] or NFC tags [1]. However, in practice, for common consumers, such countermeasures are still far behind satisfactory due to the lack of professional anti-counterfeiting detection tools or operations.

Recently, QR code has been widely adopted in anti-counterfeiting because it is cheap and easy to use. One popular QR code anti-counterfeiting scheme is the *One Item, One QR code* solution [7]. As shown in Fig. 1, this solution generates a unique QR code and then prints or pastes it on the authentic product. End-user can use mobile devices to scan and decode the QR code, and then verify the authenticity of products by through an online anti-counterfeiting system. However, there is a flaw for *One Item, One QR code* solution. Considering that the authentic QR code is printed and published, malicious counterfeiter can scan, restore and the print a counterfeit QR code, which can pass the authentication as well. This physically illegal copying (IC) attack violates the unique QR code for one item principle, and poses great threat to the QR code based anti-counterfeiting. One approach to mitigate this issue is covering up part or the entire QR code. Consumers can uncover the QR code for verification after purchase. Unfortunately, this remedy is also flawed because consumers cannot verify the authenticity before purchase.

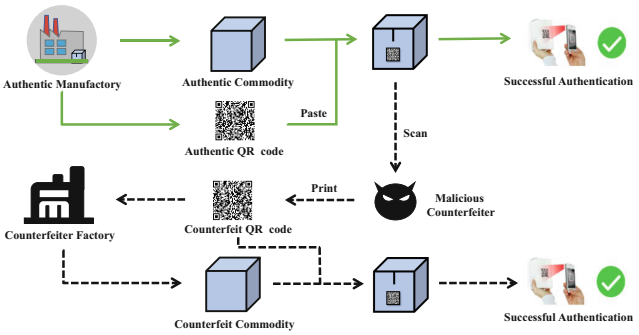


Fig. 1. The widely-deployed *One Item, One QR code* anti-counterfeiting solution cannot resist physically illegal copying. The end-user will wrongly authenticate the counterfeiting commodity when the malicious counterfeiter replicates the authentic code by scanning and printing.

In this work, we propose a semi-robust random QR code watermarking scheme for solving the physical illegal copying issue. Specifically, the authentic and counterfeit channels a QR code experiences are analyzed, based on which the watermarking-based physical anti-copying solution is formally formulated. A random embedding is devised, where the watermark bit is embedded via modulating the relationship of a paired transformed coefficient. We then apply a random image binarization procedure to obtain a valid binary QR code image. Finally, the regions originally belonging to the white module are erased, maintaining the QR code recognition. The final resultant watermark appears as *white-dot pattern* resides the black module of the QR code, which is robust

over the authentic channel while fragile to the counterfeit channel. Experimental results validate the effectiveness of the proposed watermarking system. The contributions of this work can be summarized as follows,

- We propose a physical anti-copying semi-robust random watermarking scheme for QR codes. For the first time, the semi-robust watermarking technique is introduced for solving the physical illegal copying of QR codes.
- We suggest a transform-domain watermark embedding algorithm and explore its applicability in the semi-robust watermarking context for a discrete binary image.
- A prototype mobile application is developed. Experimental results demonstrate that the proposed watermarking system could achieve authenticity verification of a physical QR code and watermark communication simultaneously.

The rest of this work is organized as follows. Section 2 briefly reviews the related work. In Sect. 3, we analyze and characterize the authentic and counterfeit channels that QR code experiences, based on which Section presents the physical anti-copying semi-robust watermarking system. Experimental results are provided in Sect. 4, and finally, Sect. 5 concludes this work.

2 Related Work

2.1 Physical Anti-copying

Physical anti-copying (PAC) methods attempt to extract discriminate features that will deviate significantly when a QR code undergoes different communication channels. Pichard *et al.* [12] proposed a dense and random noise pattern, termed Copy Detection Pattern (CDP), for document copying authentication. They then applied CDP to the QR code for product certification [13]. CDP is generated according to the maximum entropy principle, and its high-density randomness ensures its irreversibility. The physically illegal copying makes CDP blurred, which can be easily distinguished from the original CDP. Nguyen *et al.* [11] proposed a reliable performance index of the certification system based on the Neyman Pearson hypothesis test. Recently, Chen *et al.* [4] 2020 proposed a binary classifier-based scheme. The features from both spatial and frequency domains are extracted to train a two-class classifier, which can be used for distinguishing counterfeit barcodes from authentic ones. However, all the aforementioned schemes lack the capability to carry additional watermark bits. To embed data into a QR code, Tkachenko *et al.* [16] proposed a Two-Level QR code. This method replaces the black module of the QR code with a specially designed texture module that encodes data. Thus, the generated QR code can be divided into public and private levels. The standard QR code decoder can be used for the public level to decode it. The private data decodes by maximizing the correlation between the texture module and the candidate template texture modules. Although this scheme carries additional data, its texture module, which is sensitive to the printing and scanning process, is empirically designed and has poor transparency.

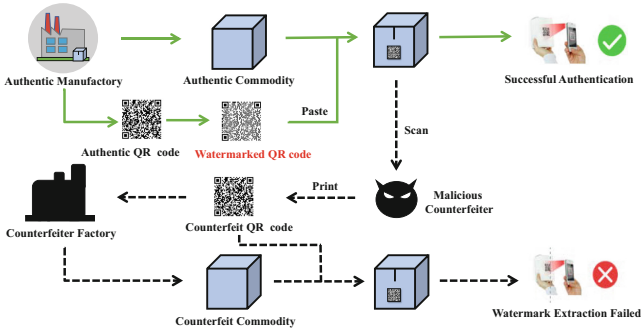


Fig. 2. Application scenario for the proposed physical anti-copying semi-robust watermarking system.

2.2 Watermarking for QR Code

Image watermarking aims at embedding data (*i.e.,* watermark) into the cover image. It has been successfully applied in many fields, such as copyright protection. Conventionally, image watermarking is vastly discussed in the digital world. However, QR code is often printed and entered into the physical world, then captured and decoded. Thus, when the image watermarking meets the QR codes, one has to consider the robustness of the watermark against printing and capture. There are some robust watermarking schemes developed for resisting printing distortion, *e.g.,* [5,6,10,14]. In addition, some semi-fragile watermarking is only robust to certain types of distortion. Bao *et al.* [3] proposed a watermarking scheme that operates in the transform wavelet domain, which is robust to JPEG compression but sensitive to malicious filtering and random noise. This scheme can be used for image authentication but can not resist printing distortion. Xie *et al.* [19] proposed an anti-counterfeiting watermarking algorithm for QR codes. Still, the watermark can only resist print-and-capture distortion and cannot be against physically illegal copying. In 2021, Xie *et al.* [18] devised an anti-copying 2D barcode by exploiting channel noise characteristics, where the authentication data were stored by exploiting the QR code error-tolerance limit. An authentication decision is made by checking whether the 2D barcode can be correctly decoded. Applying watermarking to physically Illegal Copying (IC) QR codes is quite challenging. As stated in [18], *"... to the best of our knowledge, there is no public report in which a digital watermarking technique has been used against IC attacks."*

In this work, we make the first step towards applying semi-robust randomly watermarking to physically illegal copying. The application scenario is shown in Fig. 2. The watermark bits are embedded into the authentic QR code image and then attached to the package for distribution. This watermark can be correctly extracted when it undergoes an authentic print-and-capture channel, and at the same time, it can not be extracted when the QR code is physically copied. In this next section, we dive into the proposed watermarking system.

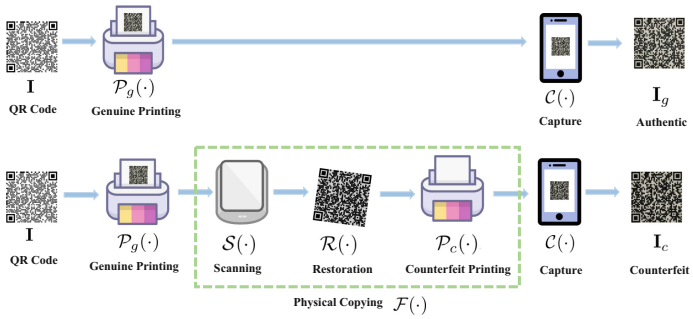


Fig. 3. Comparison of the authentic channel and counterfeit channel. Top: The authentic channel (*i.e.,* Print-Capture channel) a QR code experiences. Bottom: The counterfeit channel (*i.e.,* Print-Scan-Print-Capture channel). The key difference lies in additional physical copying action in the counterfeit channel.

3 Proposed Physical Anti-copying Watermarking System

In this section, we first investigate the authentic and counterfeit channels that a QR code would undergo, and then model the distortion for these two channels. By exploiting the distortion characteristics between two channels, we propose a physical anti-copying watermarking system.

3.1 Model for Authentic and Counterfeit Channels

Remind that this work aims to design an effective semi-robust watermark, which could survive when communicating for the print-then-capture channel (*i.e.*, the authentic channel) while degrading or even invalid for the physical-copying-then-capture (*i.e.*, the counterfeit channel). Therefore, we shall first investigate these two channels and carefully identify and exploit their differences. Based on several previous non-watermark anti-copying schemes [4,9,20], the authentic and counterfeit channels can be modeled as follows.

Authentic Channel: As shown in Fig. 3, the authentic channel consists of two critical operations, *i.e.,* printing and scanning, which can be formally expressed as

$$\mathbf{I}_g = \mathrm{AutCh}(\mathbf{I}) \triangleq \mathcal{C}\left(\mathcal{P}_g\left(\mathbf{I}\right)\right), \tag{1}$$

where $\mathrm{AutCh}(\cdot)$ represents the authentic channel, and $\mathbf{I}$ and $\mathbf{I}_g$ are the original digital QR code image, and the captured image by the end-user, respectively. $\mathcal{P}_g(\cdot)$ denotes the genuine printing performed by authentic manufacturer, and $\mathcal{C}(\cdot)$ is the capture process for QR code image. As noted in [9,20], the printing process can be modeled as a linear function, and the capture process can be modeled as low-pass filtering and then re-sampling.

Counterfeit Channel: As shown in Fig. 3, a counterfeiter first obtains the printed authentic QR code, and then physically replicate it for fooling consumers.

Thus, the counterfeit channel CtfCh$(\cdot)$ can be expressed as

$$\mathbf{I}_c = \mathrm{CtfCh}(\mathbf{I}) \triangleq \mathcal{C}\left(\mathcal{F}\left(\mathcal{P}_g\left(\mathbf{I}\right)\right)\right), \tag{2}$$

where $\mathcal{F}(\cdot)$ denotes the physical replication operation for the printed authentic QR code by a counterfeiter. Clearly, the counterfeit channel shares the printing and capture process of the authentic channel. With a thorough examination, the physical replication $\mathcal{F}(\cdot)$ can be further decomposed into three successive operations, *i.e.,* QR code scanning $\mathcal{S}(\cdot)$, restoration $\mathcal{R}(\cdot)$[1], and printing $\mathcal{P}_c(\cdot)$, which can formally written by

$$\mathcal{F}(\mathbf{I}) = \mathcal{P}_c\left(\mathcal{R}\left(\mathcal{S}(\mathbf{I})\right)\right). \tag{3}$$

The goal of a counterfeiter is to make the counterfeiting physical QR code $\mathbf{I}_c$ as same as possible to the authentic one $\mathbf{I}_g$, *i.e.,* $\mathbf{I}_c \approx \mathbf{I}_g$. From (1) and (2), one can notice the key difference between authentic and counterfeit channels lies in $\mathcal{F}(\cdot)$. We next analyze the distortion difference between these two channels. First, for a smart counterfeiter, the counterfeiting printing $\mathcal{P}_c(\cdot)$ can be similar to that of the authentic one $\mathcal{P}_g(\cdot)$, by employing similar printing equipment. Second, the aim of the restoration process $\mathcal{R}(\cdot)$ is to mitigate the difference between the captured QR image and the authentic one, using certain restoration techniques such as image binarization. Finally, the scanning operation $\mathcal{S}(\cdot)$ uses a high-resolution scanner (if possible) to scan the physical QR code. Essentially, scanning is a low-pass filtering and re-sampling process similar to the capture process $\mathcal{C}(\cdot)$.

In summary, the dominating distortion over the counterfeit channel stems from the additional scanning operation, suggesting that the distortion of the counterfeit channel suffers additional low-pass filtering and re-sampling distortion. It is worth noting that the distortion of the capture process also incurs low-pass filtering and re-sampling distortion. This requires that an effective physical anti-copying watermarking be semi-robust to low-pass filtering and re-sampling distortion. More specifically, the distortion incurred by an authentic channel requires the anti-copying watermark to be robust. In contrast, the distortion introduced by counterfeit channels requires the anti-copying watermark to be fragile. Thus, we shall carefully design a semi-robust watermarking system, striking the sweet point between fragility and robustness.

Before diving into the proposed watermarking system, we define the physical anti-copying semi-robust watermarking problem formally. Let $\mathbf{w}$ be the watermark bitstream, the watermark embedding process can be expressed as

$$\mathbf{I}^w = \mathrm{Emb}\left(\mathbf{I}, \mathbf{w}\right), \tag{4}$$

where $\mathrm{Emb}(\cdot,\cdot)$ is the watermarking function, embedding watermark $\mathbf{w}$ into the cover QR image $\mathbf{I}$; $\mathbf{I}^w$ is the resultant watermarked image. Upon receiving $\mathbf{I}^w$,

[1] The restoration aims at restoring the captured QR code, including denoising, histogram equalization, and binarization *etc.*. This operation is often optional.

the watermark extraction procedure $\mathrm{Ext}(\cdot)$ is performed as follows

$$\mathbf{w} = \mathrm{Ext}\left(\mathbf{I}^w\right). \tag{5}$$

Then, when the watermarked image $\mathbf{I}^w$ communicates over authentic or counterfeit channel, we have

$$\mathbf{w}^g = \mathrm{Ext}\left(\mathrm{AutCh}\left(\mathbf{I}^w\right)\right), \quad \mathbf{w}^c = \mathrm{Ext}\left(\mathrm{CtfCh}\left(\mathbf{I}^w\right)\right), \tag{6}$$

where $\mathbf{w}^g$ and $\mathbf{w}^c$ are the extracted watermark bitstream under authentic or counterfeit channel, respectively. Note that some of the extracted watermark bits would be incorrect. To measure the extraction accuracy, the number of correctly extracted bits can be evaluated by

$$e^g = \sum_i \mathbb{I}\left(w_i^g = w_i\right), \quad e^c = \sum_i \mathbb{I}\left(w_i^c = w_i\right). \tag{7}$$

where e^g and e^c are the number of correctly extracted bits for $\mathbf{w}^g$ and $\mathbf{w}^c$, respectively. $\mathbb{I}(\cdot)$ denotes the indicator function. The goal of the proposed physical anti-copying semi-robust watermarking system are two-fold. First, when the QR code communicates over authentic channel, the extracted watermark shall be *correctly* extracted; and when the QR code communicates over counterfeit channel, the extracted watermark shall be *wrongly* extracted. Thus, the physical anti-copying semi-robust watermarking, consisting of $\mathrm{Emb}(\cdot,\cdot)$ and $\mathrm{Ext}(\cdot)$, should maximize e^g and minimize e^c simultaneously, *i.e.,*

$$\arg\max_{\mathrm{Emb}(\cdot,\cdot),\mathrm{Ext}(\cdot)} (e^g - e^c). \tag{8}$$

In the next, we present the proposed physical anti-copying semi-robust random watermarking system, attempting to maximize (8).

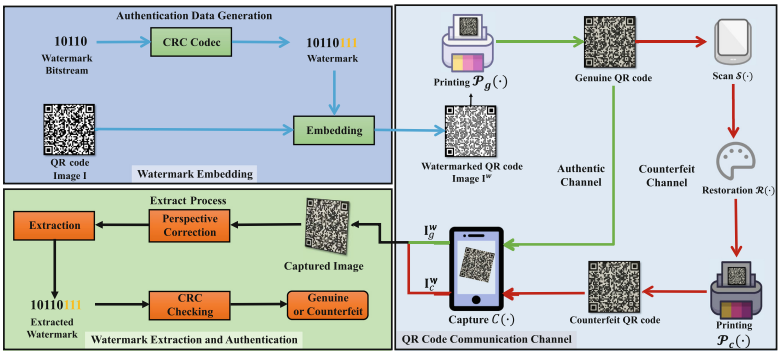


Fig. 4. Workflow of the proposed physical anti-copying semi-robust watermarking system.

3.2 Watermark Embedding

As shown in Fig. 4, the proposed watermark embedding consists of two steps. First, the authentication checksum data used for error detection is generated and appended to the watermark bitstream. Then, the watermark is embedded into the cover QR code image. Let us first discuss the crucial watermark embedding procedure.

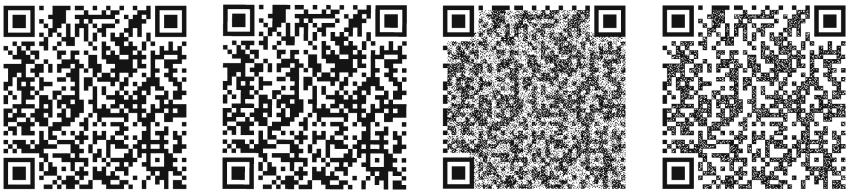
The key idea of the proposed embedding scheme is to design a robustness-controllable embedding algorithm. Note that here the robustness refers to the robustness against low-pass filtering and re-sampling. To implement a robustness-controllable embedding algorithm, we in this work suggest embedding one watermark bit by modulating the relationship between paired transformed coefficients.

More specifically, the original QR code is first divided into overlapping blocks of size $N \times N$. For each image block, the 2D discrete cosine transform (DCT) is then applied, and one can obtain $N \times N$ DCT coefficient matrix $\mathbf{M}$. A pair of coefficients c_1, c_2 are selected from the low or middle frequency bands, *e.g.*, $c_1 = \mathbf{M}(12, 19)$ and $c_2 = \mathbf{M}(19, 12)$. Then, the embedding procedure can be formulated as

$$\begin{cases} \hat{c}_1 = \max(c_1, c_2) + \Delta, \hat{c}_2 = \min(c_1, c_2) - \Delta, & \text{if } w = 0 \\ \hat{c}_1 = \min(c_1, c_2) - \Delta, \hat{c}_2 = \max(c_1, c_2) + \Delta, & \text{if } w = 1 \end{cases} \tag{9}$$

where $\hat{c}_1$ and $\hat{c}_2$ are the resultant embedded coefficients. $w \in \{0, 1\}$ is the watermark bit to be embedded, and Δ is the embedding strength parameter, aiming to enlarge the differences between $\hat{c}_1$ and $\hat{c}_2$. More importantly, Δ controls the strength of the modification, and thus in fact is the critical parameter to control the robustness. One can obtain the intermediate embedded image $\tilde{\mathbf{I}}^w$ by Inverse-DCT, where each pixel of $\tilde{\mathbf{I}}^w$ is a real value. That is, $\tilde{\mathbf{I}}^w(i, j) \in \mathbb{R}$, where (i, j) are the indices for the i-th row and j-th column pixel. Considering that a valid QR code shall be a binary-valued image, we need to randomly binarize the real-value image $\tilde{\mathbf{I}}^w$ into a binary image.

Specifically, suppose the discrete dynamic set for binary QR code is $\{0, 255\}$, where pixel values for the black and white are 0 and 255, respectively. After performing the watermark embedding (9), the pixel value of the intermediate embedded image could be larger, equal, or smaller than that of the original image pixel value. We discuss these three types of relationships, based on which to design a binarization rule. First, for these pixels that enjoy no changes after embedding, we can safely leave them alone, without further action. Second, the pixel value of the intermediate embedded image may overflow or underflow the valid set $\{0, 255\}$. For this case, we shall clip the pixel value into the valid set $\{0, 255\}$. For instance, for pixel $\mathbf{I}(i, j) = 255$, it may become $\tilde{\mathbf{I}}^w(i, j) = 256$ after embedding. Thus, one shall clip this value to 255. Similarly, for these pixels $\tilde{\mathbf{I}}^w(i, j) < 0$ whose original pixel values are zeros, one has to clip them to 0. Third, the pixel value of the intermediate embedded image may slightly change, but still, escape the valid set $\{0, 255\}$. For this case, we would like to pull the pixel value to 0 or 255. As a concrete example, suppose the pixel $\mathbf{I}(i, j) = 0$, it



(a) $\mathbf{I}$ (b) $\tilde{\mathbf{I}}^w$ (c) $\mathbf{I}_b^w$ (d) $\mathbf{I}^w$

Fig. 5. An example of the embedding process is the randomly embedding strength $p = 0.5$. (a) Original QR code image $\mathbf{I}$. (b) Intermediate embedded QR code image $\tilde{\mathbf{I}}^w$. Note that the $\tilde{\mathbf{I}}^w$ is a real-value watermarked image (normalized in $[0, 255]$ for better visualization), which is quite similar to the original image; zoom in for better comparison. (c) Binarized image $\mathbf{I}_b^w$ according to (10). (d) The final watermarked binary QR image $\mathbf{I}^w$, by erasing the regions that originally belongs to the white module of $\mathbf{I}$.

may become $\tilde{\mathbf{I}}^w(i,j) = 3$ after embedding. Thus, we propose to lift this value to 255 with probability p. Similarly, for these pixels $\tilde{\mathbf{I}}^w(i,j) < 255$, whose original pixel values are 255, we suggest downgrading it as 0 with probability p. We called p randomly embedding strength. Mathematically, let $\mathbf{I}_b^w = \mathbf{I}$, we summarize the aforementioned binarization operation on the intermediate embedded image $\tilde{\mathbf{I}}^w$ as follows

$$\mathbf{I}_b^w(i,j) = \begin{cases} 255 & \text{if } \tilde{\mathbf{I}}^w(i,j) > 0,\ \mathbf{I}(i,j) = 0,\ p_{ij} \le p \\ 0 & \text{if } \tilde{\mathbf{I}}^w(i,j) < 255,\ \mathbf{I}(i,j) = 255,\ p_{ij} \le p \end{cases} \tag{10}$$

where $p_{ij} \sim U[0,1]$ and $\mathbf{I}_b^w$ denotes the binarized watermarked image. Finally, to maintain a valid QR code recognition, we propose to erase these regions that are originally belonging to white. This erasion can be expressed by

$$\mathbf{I}^w(i,j) = \begin{cases} \mathbf{I}_b^w(i,j) & \text{if } \mathbf{I}(i,j) = 0 \\ 255 & \text{if } \mathbf{I}(i,j) = 255. \end{cases} \tag{11}$$

In the experiment, we also found that, when the entire QR code image is used for embedding, the QR recognition effectiveness will degrade. This is because the proposed watermark embedding scheme injects specific white-dots into the black module, which could deteriorate the recognition effectiveness of the position detection pattern. To resolve this issue, we suggest excluding the position detection pattern (*i.e.,* the three black squares) and the boundary.

Until now, we obtain the final watermarked QR code image $\mathbf{I}^w$. To intuitively illustrate the proposed embedding procedure, we in Fig. 5 provide an exemplar watermarking process, where the algorithmic parameters are set the same as Sect. 4.1. One can see from Fig. 5-(d) that the semi-robust watermark is rendered as the white-dots in the black module of the QR code. As will be demonstrated shortly, such a white-dot pattern could survive over the authentic channel, while it will be significantly eroded under the counterfeit channel.

Let us go back to the authentication data generation procedure. The goal of this work is to use a watermark to verify the authenticity of the QR code. Thus, we shall verify the correctness of the extracted watermark bits. In this work, we employ the widely-used Cyclic Redundancy Check (CRC) code for checking. Before embedding the watermark into the QR code image, the bitstream encoded with CRC is first obtained as the checksum for the given watermark.

Remarks: It is worth noting that the embedding strategy (9) was successfully practiced in several robust watermarking schemes, *e.g.,* [5]. However, none of these works explored the applicability of (9) in the semi-robust watermarking context for a binary image such as QR code.

3.3 Watermark Extraction and Authentication

As illustrated in Fig. 4, the general watermark extraction and authentication procedure contains three steps. First, the captured QR code image is prescriptively-corrected, and then the watermark is extracted. Finally, the extracted data is verified through CRC checking.

First, QR code recognition is performed. The standard QR code recognition algorithm includes scanning, image binarization, perspective, geometric correction, and decoding *et al.*. Due to the error-tolerance design of the QR code, the incurred distortion by the embedded watermark does not interfere with the decoding procedure. However, one shall successfully locate and perspective-correct the captured QR code image to facilitate the watermark extraction. Luckily, many off-the-shelf QR codecs are equipped with an efficient automatic positioning and correction algorithm. Thus, in experiments, we employ the QR code localization procedure `QRCodeDetector` provided by OpenCV for locating and perspective correction.

After perspective correction, the perspective-corrected QR code image is divided into non-overlapping blocks, similar to the block division of the embedding procedure. Then for each image block, the DCT transform is applied. The DCT coefficient pairs $\tilde{c}_1$ and $\tilde{c}_2$ extracted from the same coefficient bands used in the embedding procedure. The watermark bit can be extracted by

$$\hat{w} = \begin{cases} 1 & \text{if } \tilde{c}_1 \geq \tilde{c}_2 \\ 0 & \text{if } \tilde{c}_1 < \tilde{c}_2 \end{cases}. \tag{12}$$

Upon extracting the watermark bitstream, CRC checking is conducted. If CRC checking passes, the decision for an authentic QR code is made. Otherwise, excessive erroneous watermark bits are extracted, making the decision counterfeit.

4 Experimental Result

4.1 Experimental Setup

The size of the QR code image is 246×246 of version 6, and the image block size for embedding one bit is 30×30. The length of the randomly-generated

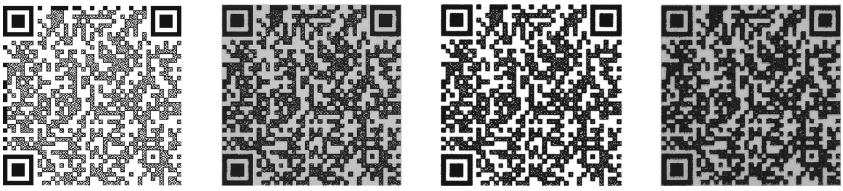
(a) Counterfeit QR code (b) Authentic QR code

Fig. 6. Handheld authentication using prototype mobile app. (a) and (b) are the authentication for the physical counterfeiting QR code, and the authentic QR code, respectively. The authentication results notify on the screen.

Table 1. Experimental settings for three constructed datasets. Note that it is unnecessary for an authentic manufacturer to use a scanner to replicate QR code; thus, the cell is noted as NaN.

Producer	Authentic manufacturer	Counterfeiter I	Counterfeiter II
Printer	Brother MFC-T4500DW (all with 1200 dpi)	Brother MFC-T4500DW (all with 1200 dpi)	RICOH Aficio MP 7500 PCL (all with 1200 dpi)
Scanner	NaN	Brother MFC-T4500DW (all with 1200 ppi)	Brother MFC-T4500DW (all with 1200 ppi)
Camera	HUAWEI Nova 8 Pro	HUAWEI Nova 8 Pro	One Plus 8 Pro
Dataset size	770	770	770
Print size (cm)	1.5	1.5	1.5

watermark bitstream is 59 bits. The embedding coefficient pairs used in this experiment are $\mathbf{M}(19, 12)$ and $\mathbf{M}(12, 19)$. $\Delta = 50$ and random embedding strength $p = 1.0$. Considering that no publicly available physical anti-copying watermarking datasets. This work constructed three datasets, including one authentic QR code dataset and its two counterfeiting QR code counterparts. The printing size of the QR code image is 1.5 cm $\times$ 1.5 cm. The detailed experimental equipment settings for these three datasets are tabulated in Table 1, where each dataset contains 770 samples. The printing resolution is 1200 dpi, and the scanning resolution is 1200 PPI, which is the maximum-available setting provided by the tested equipment. In addition, to verify the practical usage of the proposed method, we have developed a prototype mobile application (see Fig. 6). The size of each captured frame is fixed as 786×672 for all tested mobile phones.



(a) $\mathbf{I}^w$ (b) $\mathbf{I}_g^w$ (c) $\mathbf{I}_r^w$ (d) $\mathbf{I}_c^w$

Fig. 7. Comparison of the authentic QR code with counterfeited QR code. (a) The digital watermarked QR code image $\mathbf{I}^w$, (b) Captured authentic QR code image that experiences authentic PC channel $\mathbf{I}_g^w = \text{AutCh}(\mathbf{I}^w)$. (c) The counterfeited QR code image is obtained by restoring the scanned physical authentic QR code, $\mathbf{I}_r^w = \mathcal{R}\left(\mathcal{S}\left(\mathcal{P}_g\left(\mathbf{I}^w\right)\right)\right)$. (d) Capture the counterfeit QR code image printed by a counterfeiter $\mathbf{I}_c^w = \text{CtfCh}(\mathbf{I}^w)$.

4.2 Comparison of the Authentic and Counterfeited QR Code

As shown in Fig. 7-(a)(b), for the authentic channel, the watermarked QR code image $\mathbf{I}^w$ is authentically printed by the Printer Brother MFC-T4500DW and then captured by the mobile camera of the Huawei Nova 8 PRO. For the counterfeit channel, the authentic QR code image is first scanned by Brother MFC-T4500DW under 1200 dpi, and then the counterfeit QR code image is obtained by printing the scanned QR code image with Brother MFC-T4500DW. Note that we here deliberately use the same printing equipment for both authentic manufacturers and counterfeiters. The reason for this setting is to push the counterfeiting ability to the limit,*i.e.,* the counterfeiter could replicate the QR code using the same equipment as the authentic manufacturer.

By carefully observing the four QR code images from Fig. 7 (a) to (b), one can notice that the number of white-dots in the black module of QR code (*i.e.,* the watermark) is decreasing. This suggests that the embedded watermark erodes gradually. Quantitatively, we test the watermark extraction under 10 trials. For the captured authentic QR code image that experiences PC channel, *i.e.,* $\mathbf{I}_g^w = \text{AutCh}(\mathbf{I}^w)$, the watermark can still be extracted in a low erroneous bit level. The average number of erroneous bits is 0.9, meaning that less than 1 bit goes wrong out of a total of 59 watermark bits. In contrast, for the capture of the counterfeit QR code image printed by a counterfeiter, *i.e.,* $\mathbf{I}_c^w = \text{CtfCh}(\mathbf{I}^w)$, the average erroneous bit is 28.9, closing to the 29.5 erroneous bits of the random-guessing watermark extraction.

4.3 Printing Size v.s. Erroneous Bits

In general, the printing size of the anti-copying QR code depends on the printing equipment. Considering that the printing resolution of a printer is limited, a QR code can hardly be printed faithfully as its digital version. Thus the watermark cannot be extracted correctly when the printing size is too small. Therefore, finding the relationship between the printing size and the number of erroneous bits is important. To this end, we print different QR codes of various sizes,

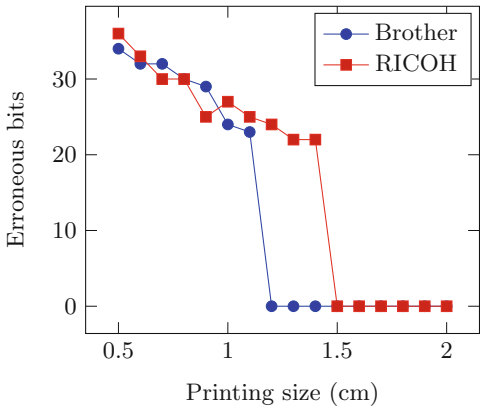


Fig. 8. The printing size (cm) versus the number of erroneous bits.

ranging from 0.5 cm × 0.5 cm to 2.5 cm × 2.5 cm, and then record the number of erroneous bits. Each QR code is recognized for 30 attempts. The minimum number of erroneous bits among these attempts is recorded as the final result. As shown in Fig. 8, the number of watermark erroneous bits decreases w.r.t. the increment of the printing size. The printing size of the QR code is negatively correlated with the number of error bits. In this light, one can observe the lower bound of the printing size, where the watermark cannot be correctly extracted for the printing size when smaller than this lower bound.

Table 2. The range of test printing size for authentic manufacturer and counterfeiter.

Printer	Authentic	Counterfeit I	Counterfeit II
Upper bound (cm)	$+\infty$	1.8	2.3
Lower bound (cm)	1.2	0.0	0.0

4.4 Printing Size v.s. Anti-copying Capability

In practice, physical anti-copying watermarking supports smaller printing size is preferred, which can be attributed to two reasons. First, smaller printed QR code can find more application scenarios, *e.g.,* delicate package. Second, acquiring a high-resolution image for smaller printed QR code is costly, and thus small printing size barriers the counterfeiting; when the QR code printed large enough, it can be forged counterfeited even using a low-resolution scanner or printer. We in this section aim to empirically find the feasible printing size for support reasonably good anti-copying capability of QR code.

To find a feasible range of the printing size, we have printed QR codes of different sizes at an interval of 0.1 cm, and counterfeit them with different printers.

As aforementioned in Sect. 4.3, larger(smaller) printed QR code often leads to fewer(more) the erroneous bits. In other words, when the printing size greater than a threshold, the extracted watermark is error-free; and when the printing size is less than a threshold for counterfeit QR code, errors would occur during watermark extraction. Therefore, we can take the minimum printing size of authentic QR codes that can be extracted correctly as the lower bound for the anti-copying printing size range. The maximum printing size of counterfeit that cannot be extracted correctly as the upper bound for the anti-copying printing size range.

The results are provided in Table 2. It can be seen that, for authentic manufacturer, when the printing size of the authentic QR code exceeds 1.2 cm × 1.2 cm, the watermark can be extracted correctly, *i.e.,* the printing size range of which the watermark can be extracted correctly is $\mathbb{A} = [1.2, +\infty)$. For Counterfeiter I, when the printing size of the counterfeit QR code is less than 1.8 cm × 1.8 cm, the watermark cannot be extracted correctly, *i.e.,* the printing size range the watermark cannot be extracted is $\mathbb{F}_1 = (0, 1.8]$. Similarly, for Counterfeiter II, the printing size range the watermark cannot be extracted is $\mathbb{F}_2 = (0, 2.3]$. Therefore, a feasible printing size range of anti-copying should be $\mathbb{A} \bigcap \mathbb{F}_1 \bigcap \mathbb{F}_2 = [1.2, 1.8]$.

Table 3. Performance Comparison with Chen *et al.* [4]. The best results highlighted in bold.

Method	FAR	FRR	NACC	AUC
Chen *et al.* [4]	0.00%	2.50%	98.75%	0.9958
Proposed	0.00%	**0.52%**	**99.74%**	**0.9974**

4.5 Comparison of Authentication Performance

To the best of our knowledge, few works realize the physical anti-copying function from the watermarking perspective. To this end, we compare the most recent and relevant work Chen *et al.* [4]. They employed spatial and frequency features to train a two-class classier to distinguish the authentic QR from the counterfeit ones. Instead, we report the authenticity of QR based on the success or failure of semi-robust watermark extraction. False Acceptance Rate (FAR, the percentage of counterfeit samples that have been falsely accepted as authentic), False Rejection Rate (FRR, the percentage of genuine samples that have been falsely accepted as counterfeit), and Normalized ACCuracy (NACC) are employed as performance metrics. The NACC is defined as follows,

$$\text{NACC} = 1 - (\text{FAR} + \text{FRR})/2 \tag{13}$$

Experiments were conducted on the datasets shown in Table 1. Experimental results are given in Table 3. Compared with Chen *et al.* [4]. The proposed method always achieves superior performance under all the metrics. Specifically, our

proposed semi-robust watermarking solution shows advantages in anti-copying performance, with a higher accuracy rate of 99.74% and lower FRR of 0.52%. Despite the superior performance, our method provides an additional communication channel via semi-robust watermarking, while Chen *et al.* [4] merely made a binary decision without the capability of carrying additional information.

5 Conclusion

In this work, we made the first step toward implementing a physical anti-copying semi-robust watermarking for QR codes. We devised a random watermark embedding procedure by exploiting the distortion characteristics between the authentic and counterfeit channels. The resultant semi-watermark appears as irregular *white-dot pattern* resides the black module of QR code, which is robust to the authentic print-scan but fragile to the physically illegal copying. Compared with existing physical anti-copying approaches, the proposed scheme requires no training data to train classifiers. More importantly, the proposed method provides the verification of authenticity for a QR code and additional communication capability for transmitting watermarks simultaneously. Experimental results demonstrate the effectiveness of the proposed watermarking system. We also developed a prototype mobile app to verify the practical usage of the proposed method. We would like to extend the proposed scheme to color barcode cases for future work.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (61901237, 62171244), Alibaba Innovative Research Program. Ningbo Natural Science Foundation- Young Doctoral Innovation Research Project (Grant No. 2022J080).

References

1. Alzahrani, N., Bulusu, N.: Securing pharmaceutical and high-value products against tag reapplication attacks using NFC tags. In: 2016 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 1–6. IEEE (2016)
2. Baldini, G., Fovino, I.N., Satta, R.: Survey of techniques for fight against counterfeit goods and intellectual property rights (IPR) infringing (2015)
3. Bao, P., Ma, X.: Image adaptive watermarking using wavelet domain singular value decomposition. IEEE Trans. Circuits Syst. Video Technol. **15**(1), 96–102 (2005)
4. Chen, C., Li, M., Ferreira, A., Huang, J., Cai, R.: A copy-proof scheme based on the spectral and spatial barcoding channel models. IEEE Trans. Inf. Forensics Secur. **15**, 1056–1071 (2020)
5. Fang, H., Zhang, W., Zhou, H., Cui, H., Yu, N.: Screen-shooting resilient watermarking. IEEE Trans. Inf. Forensics Secur. **14**(6), 1403–1418 (2019)
6. Kang, X., Huang, J., Zeng, W.: Efficient general print-scanning resilient data hiding based on uniform log-polar mapping. IEEE Trans. Inf. Forensics Secur. **5**(1), 1–12 (2010)
7. Keni, H., Earle, M., Min, M.: Product authentication using hash chains and printed QR codes (2017)

8. Lehtonen, M.O., Michahelles, F., Fleisch, E.: Trust and security in RFID-based product authentication systems. IEEE Syst. J. **1**(2), 129–144 (2007)
9. Malvido, A., Pérez-González, F., Cousiño, A.: A novel model for the print-and-capture channel in 2D bar codes. In: Gunsel, B., Jain, A.K., Tekalp, A.M., Sankur, B. (eds.) MRCS 2006. LNCS, vol. 4105, pp. 627–634. Springer, Heidelberg (2006). https://doi.org/10.1007/11848035_83
10. Nakamura, T., Katayama, A., Kitahara, R., Nakazawa, K.: A fast and robust digital watermark detection scheme for cellular phones. NTT Tech. Rev. **4**, 57–63 (2006)
11. Nguyen, H.P., Retraint, F., Morain-Nicolier, F., Delahaies, A.: A watermarking technique to secure printed matrix barcode-application for anti-counterfeit packaging. IEEE Access **7**, 131839–131850 (2019)
12. Picard, J.: Digital authentication with copy-detection patterns. Electron Imaging **5310**, 176–783 (2004)
13. Picard, J., Landry, P., Bolay, M.: Counterfeit detection with QR codes. In: Proceedings of the 21st ACM Symposium on Document Engineering. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3469096.3474924
14. Pramila, A., Keskinarkaus, A., Seppänen, T.: Toward an interactive poster using digital watermarking and a mobile phone camera. SIViP **6**, 211–222 (2012)
15. Song, B., Mitchell, C.J.: RFID authentication protocol for low-cost tags. In: Proceedings of the first ACM conference on Wireless network security, pp. 140–147 (2008)
16. Tkachenko, I., Puech, W., Destruel, C., Strauss, O., Gaudin, J.M., Guichard, C.: Two-level QR code for private message sharing and document authentication. IEEE Trans. Inf. Forensics Secur. **11**(3), 571–583 (2016)
17. Turcu, C.E., Turcu, C.O., Cerlinca, M., Cerlinca, T., Prodan, R., Popa, V.: An RFID-based system for product authentication. In: Eurocon 2013, pp. 32–39. IEEE (2013)
18. Xie, N., Zhang, Q., Chen, Y., Hu, J., Luo, G., Chen, C.: Low-cost anti-copying 2D barcode by exploiting channel noise characteristics. IEEE Trans. Multimedia **23**, 3752–3767 (2021). https://doi.org/10.1109/TMM.2020.3031083
19. Xie, R., Hong, C., Zhu, S., Tao, D.: Anti-counterfeiting digital watermarking algorithm for printed QR barcode. Neurocomputing **167**, 625–635 (2015)
20. Zhang, L., Chen, C., Mow, W.H.: Accurate modeling and efficient estimation of the print-capture channel with application in barcoding. IEEE Trans. Image Process. **28**(1), 464–478 (2019)