

Print to Scan Resistant Watermarking based on Green Noise Diffusion Method using Machine Learning

Yoshi Michael Cho*, Hiroyuki Imada*, Naoto Kawamura**

Hyunho Kang***, Keiichi Iwamura*

* Tokyo University of Science, Tokyo, Japan

** Kawamura Techno.Lab, Tokyo, Japan

*** National Institute of Technology, Tokyo college, Tokyo, Japan

cho_michael_yoshi@sec.ee.kagu.tus.ac.jp, imada_hiroyuki@sec.ee.kagu.tus.ac.jp, kawamura.7010@gmail.com

kang@tokyo-ct.ac.jp, iwamura@ee.kagu.tus.ac.jp

Abstract— The green-noise diffused watermarking method is resistant to printing and scanning, and the embedded pattern is not easily visible. We extended this method to enable the extraction of watermark information from cropped images by embedding marker patterns. We used machine learning and embedded five green-noise patterns, including four value patterns and one marker pattern. Correct watermark extraction rate of over 95% are obtained from the printed images. We also embedded sub-information in the marker pattern to obtain block synchronization from the cropped images. Furthermore, the introduction of RS codes allowed us to obtain an extraction correctness rate close to 100% from both electronic and printed images.

Keywords— Watermarking, Data Hiding, Copyright Protection

I. INTRODUCTION

In today's digital information society, the reproduction of information has enabled many people to share information, which has greatly advanced society. However, this convenience has led to the illegal reproduction and distribution of personal work, which has caused problems such as copyright infringement. In the case of printed images, the high image quality of imaging devices in recent years has made it easy to obtain reproductions that look exactly the same as the originals, which has encouraged not only illegal copying that infringes copyrights but also malicious criminal acts such as counterfeiting banknotes and securities.

Digital watermarking technology has been attracting attention as a countermeasure against such fraudulent activities. Unlike encryption and digital rights management, digital watermarking embeds information directly into documents and images, thus providing the ultimate security technology for protecting copyrights and tracking.

The functions and requirements for such watermarking technologies depend on their applications. In the case of copyright protection, it is necessary to track and monitor not only electronic data but also printed materials, because high-quality photographs are available in printed form. For this

reason, watermarking technology with print and scan resilience is required in addition to electronic data.

Various methods have been proposed for printing and scan-resilience watermarking. On digital halftoning of printers, embedding information using the dot pattern method, error diffusion method, and dither method [1][2] have been proposed. These methods are not intended to be used in combination with electronic data because they are directly embedded in printed materials and are used to embed messages in printed images.

A method for embedding a two-dimensional pattern has also been proposed [3]. Because the embedding is in real space (pixel space), it enables fast embedding and can be used in smartphones, etc. However, the pattern and its boundaries are easily visible as artifacts.

In this paper, we propose a print/scan-resistant watermarking method using the green noise diffusion method [4]. This method divides an image into blocks, embeds watermark information in each block, and has the characteristics of print resistance.

Usually, copyright protection applications deal with large-sized images as target images. Therefore, it is necessary to extract the watermark from the cropped image of the image.

In this study, we investigate a green-noise diffusion method that embeds marker patterns and can be applied to cropped images. For this purpose, identifier based on machine learning are used to detect watermark patterns, including marker patterns with high accuracy. To obtain block synchronization, sub-information is introduced and we investigated block synchronization from cropped images. Furthermore, we aimed to create a highly secure watermarking method using camouflage patterns.

In the following sections, we describe the operational forms for copyright protection applications in Section II, the algorithm of the Green-Noise Diffused Watermarking method in Section III, machine learning in Section IV, implementation and experiments in Section V, evaluation and discussion in Section VI, and finally, a summary.

II. OPERATIONAL FORMS OF WATERMARKING

The following requirements are listed for the watermarking method for copyright protection as shown in Figure 1.

1. Any image viewer can read the embedded information without a key using publicly available watermark extraction software. This excludes watermarking methods that cannot be read without a key.
2. The purchaser can remove/add/rewrite the watermark using the "key". In other words, the watermark cannot be removed without a key. For this purpose, the watermark must be reversible using a key.
3. Electronic and printed data can be traceable and detected for unauthorized use. This requires a watermarking method that possesses print/scan resistance.
4. The loss of the key can be minimized. In other words, the key should be identified for each image, and the lost/stolen key should not have the ability to remove the watermark from other images.

III. ALGORITHM OF GREEN-NOISE DIFFUSION METHOD

A. Green-Noise Pattern

Let us assume that the dot distribution is dispersed in $R \times R$ pixels (R is a power of 2). Let the area ratio of dot 0 (black dot) be g ($0 \leq g \leq 1$), and the dot pattern at point (x, y) be $p(x, y) \in \{0, 1\}$. A dot pattern with an intermediate area ratio ($g = 1/2$) was obtained for embedding a digital watermark.

- Step 1:** First, let $p(x, y)$ be a random dot of $R^2/2$. The initial value is generated by a random number generator and shows a white noise spectrum.
- Step 2:** Next, perform a two-dimensional Fourier transform of the dot pattern to obtain $P(f_x, f_y)$.
- Step 3:** For $P(f_x, f_y)$, filter $D(f_x, f_y)$ which has a constant value ($=1$) in the band of $f_{x, \min} < f_x < f_{x, \max}$, $f_{y, \min} < f_y < f_{y, \max}$ are filtered to obtain a new $P'(f_x, f_y)$.
- Step 4:** Perform an inverse Fourier transform on $P'(f_x, f_y)$ to obtain a multi-level dot pattern $p'(x, y)$.
- Step 5:** Calculate the error function $e(x, y) = p'(x, y) - p(x, y)$, and invert the same number of white and black dots in descending order of the error values.
- Step 6:** Repeat Step 2~5 to obtain the dot pattern with $g=1/2$.

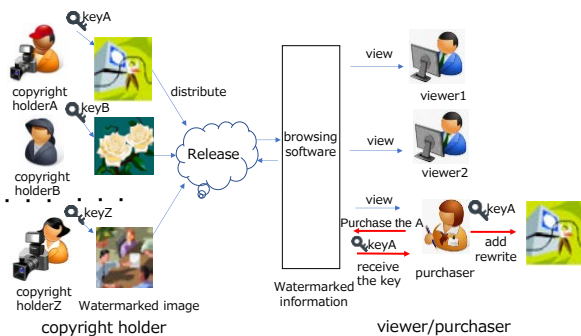


Figure 1. Operational form of digital watermarking for copyright protection applications

B. Generating ID Patterns

Anisotropy are introduced for the watermark embedding pattern and a four-level embedding pattern p_i ($i=0,1,2,3$) and a marker pattern H are generated by filtering. At this time, a random number pattern with the embedder's ID as the seed value is used as the initial value for filtering [5]. The obtained pattern p_i shows the green noise characteristics shown in Figure 2, and different patterns can be obtained by changing the seed value. The ID pattern can be regarded as a two-dimensional pattern, such as a fingerprint unique to the copyright holder. The pattern is stored as a secret key 'Key (M)', and the watermark can be removed, appended, or rewritten as required. 4-level patterns are mapped to an embedding code, which is a Gray Code with a Hamming distance of 1.

C. Introduction of Camouflage Patterns

A collusion attack is an attack in which a large number of users collude to remove the watermark [6]. In the case of block-wise watermarking, the attack can be easily carried out by cutting out the embedded pattern for each block from a single image and analysing the pattern without the collusion of many users. In order to avoid such attacks and to obtain higher security, "Camouflage patterns" are introduced, which show the same spectrum but have different dot patterns. The camouflage pattern consisted of an inversion pattern and a phase-shift pattern.

For the inversion pattern, eight patterns are generated for one ID pattern, and the i -th pattern $p_{i, \sigma_i}(x, y)$ are as follows:

- σ_0 : Pattern of $p_i(x, y)$ (basic ID pattern).
- σ_1 : Vertical reversal pattern of $p_i(x, y)$.
- σ_2 : Horizontal reversal pattern of $p_i(x, y)$.
- σ_3 : Vertical and horizontal reversal patterns of $p_i(x, y)$.
- σ_4 : Negative pattern of $p_i(x, y)$.
- σ_5 : Negative pattern of vertical reversal of $p_i(x, y)$.
- σ_6 : Negative pattern of the horizontal reversal term of $p_i(x, y)$.
- σ_7 : Negative vertical and horizontal reversal patterns of $p_i(x, y)$.

The vertical and horizontal directions are relative to the major and minor axes of the ellipse, respectively. The negative pattern is a pattern in which the white and black dots are reversed. Because the numbers of white and black dots are the same, the number of dots does not change even if they are reversed. Figure 3(a) shows the inverted pattern for $R=32$. The pattern σ_i was generated by a random number generator and randomized for each block.

Next, the phase shift pattern is introduced in which the starting position of the pattern is shifted. The shift amount was determined randomly by generating random numbers in the blocks. Now, when the shift amount is $s_k = (x_k, y_k)$ ($0 \leq x_k, y_k < R$), the shifted pattern is as follows:

$$p_{i, \sigma_j, s_k}(x, y) = p_{i, \sigma_j}((x + x_k) \bmod R, (y + y_k) \bmod R) \quad (1)$$

Here, $\bmod R$ denotes the remainder, with R being the modulo. Figure 3(b) shows a phase shift pattern. For each (i, σ_i) pattern, R^2 patterns can be generated. Because of the randomness of the dots, it is difficult to see the boundary line even after the phase shift. The same spectral shape is obtained.

pattern	pattern 0	pattern 1	pattern 2	pattern 3	pattern 4
Embedded code	(00)	(01)	(11)	(10)	(H)
Dot pattern 32x32					
Spectrum					

Figure 2. Embedding pattern and spectral characteristics (32x32)

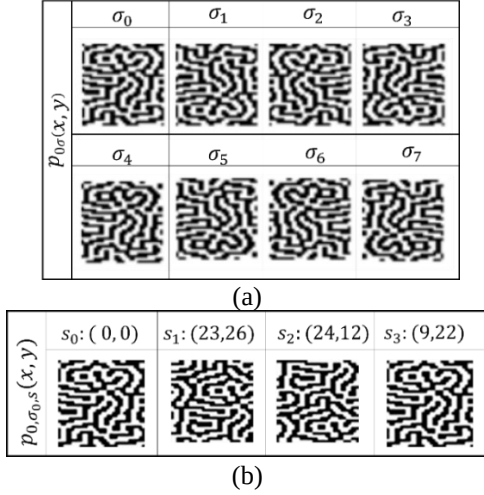


Figure 3. Camouflage patterns: (a) inversion pattern, (b) phase shift pattern

D. Introduction of Sub Information Patterns

The synchronization marker pattern was superimposed and embedded as secondary information. The pattern of the secondary information $m(x,y)$ should be set such that its spectral distribution is in the high-frequency range that does not overlap with the spectrum of the main information, that is, the green noise pattern. For example, if checkerboard pattern are used for every other pixel, its spectrum will be the Nyquist frequency, and there will be no crosstalk with the spectrum of the green noise.

The sub-information is extracted by convolution with the embedded image in pixel space (real space) using the filters M_0 and M_1 shown in Figure4(a).

$$U_0 = W(x,y) * M_0$$

$$U_1 = W(x,y) * M_1 \quad (2)$$

Here, $*$ denotes convolution. The output $M_{out}(x,y)$ is represented by Eq. (3) as follows:

$$M_{out}(x,y) = \alpha |U_0 - U_1| \quad (3)$$

Here, α is the standardization coefficient, and $\alpha = 1$. Figure4(b) shows the marker pattern with the embedded sub-information, its spectrum, and the extraction sub-information.

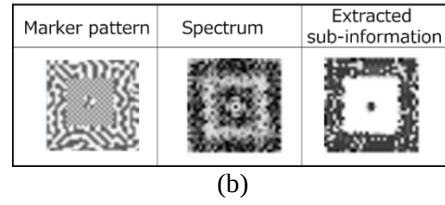
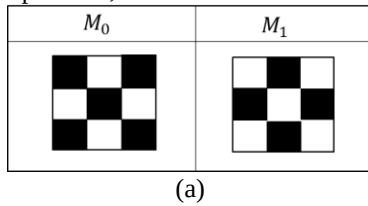


Figure 4. Sub information: (a) Sub information identification filter, (b) Marker pattern and its spectrum, extraction sub information

E. Watermark embedding

The watermark information is embedded in real space (pixel space) in units of $R \times R$ blocks. Let $i_k(x,y)$ be the image data of the k th block, gain be the embedding strength, and $w_k(x,y)$ be the image data in which the watermark is embedded. The watermark data to be embedded are decomposed into a 1D bit sequence and $w_m \in (0,1)$. The embedding is performed by extracting two bits from w_m for each block, and the embedding into the k th block is expressed by equations (4) and (5)

$$w_k(x,y) = i_k(x,y) + \text{gain} \cdot p_{i,\sigma_k,s_k}(x,y) \quad (4)$$

$$\text{When } w_m = (00), \quad p_{i,\sigma_k,s_k}(x,y) = p_{0,\sigma_k,s_k}(x,y)$$

$$\text{When } w_m = (01), \quad p_{i,\sigma_k,s_k}(x,y) = p_{1,\sigma_k,s_k}(x,y)$$

$$\text{When } w_m = (11), \quad p_{i,\sigma_k,s_k}(x,y) = p_{2,\sigma_k,s_k}(x,y) \quad (5)$$

$$\text{When } w_m = (10), \quad p_{i,\sigma_k,s_k}(x,y) = p_{3,\sigma_k,s_k}(x,y)$$

To maintain the average luminance, $p(x,y) \in (-1/2, 1/2)$. (σ_k, s_k) was randomly determined using a random number generator. If the seed value is the password of the embedder, security is maintained even if the key is lost, as described later.

To avoid overflow/underflow of pixel data by embedding, the following linear transformation is applied before embedding:

$$i' = (1 - \text{gain}) \times i(x,y) + \text{gain}/2 \quad (6)$$

F. Extracting and Viewing Watermarks

The extraction of the watermark is achieved using extraction software, which is independent of the ID and camouflage pattern, and anyone can read and see the embedded information without a key.

The image in the k th block is extracted and transformed into spectral information W_k by fast Fourier transform (FFT). The luminance distribution was then normalized by histogram equalization. Subsequently, patterns are identified and classified, and 2-bit data of the embedded information are extracted from the identified patterns. Pattern identification is performed using the machine learning discriminator described below.

G. Remove, Add, Rewrite Watermark

A key and password are required to remove, add, or rewrite the watermark. Removal can be performed by subtracting the watermark information to return to the original image, and appending or rewriting can be performed. In other words, the arithmetic operation of Equation (7) is performed.

$$i_k(x,y) = w_k(x,y) - \text{gain} \times p_{i,\sigma_k,s_k}(x,y) \quad (7)$$

If a linear transformation is performed to avoid overflow/underflow, inverse transformation is performed.

IV.MACHINE LEARNING

A. Network Configuration

In images with high-frequency, the conventional identifier has many judgment errors. Therefore, discriminator based on neural network machine learning are used.

The structure of the NN is illustrated in Figure 5. The number of neurons in the input layer was 256, the hidden layer (middle layer) was 50, and the output layer was five. The sigmoid function is used as the activation function, and the confidence information is obtained by Softmax in the final stage. The dataset consists of 1k×1k images of mandrill images. This is because mandrill images have a high spatial frequency and are difficult to identify; therefore, images that are as difficult to discriminate as possible are more effective for training. Of the total 5000 labelled patterns, 80% were used for training and 20% for testing.

V. IMPLEMENTATION AND EXPERIMENTS

A. Implementation

The watermarking program was performed in the VC#. For this reason, the network parameters resulting from machine learning in Python were incorporated into the watermarking program as CSV data. The array size of the network parameters is W1 [256,50] , W2 [50,5], b1 [50], and b2 [5].

B. Experiments

The experiment was conducted on six images (512×512), as shown in Figure 6. A total of 2048 bits (256 ASCII characters) were embedded in a block size of 32×32 with a gain of 0.125. Embedding was performed by decomposing the image data into Y, Cb, and Cr, and embedding them into the luminance data. Table 1 shows the correct response rates for electronic data reading. This was > 98.8%.

C. Read from Printed, Scanned Images

The watermark was embedded with a camouflage pattern using $gain=0.1875$, and the image was printed using a Canon TS5030 inkjet printer at 175 ppi (*pixels per inch*) (print size 74.3 mm). The image was scanned with a scanner built into the printer at 600dpi. After reading, the images were cropped and formatted to 512×512 pixels, and watermark extraction was performed. Table 2 shows the results of watermark identification. All the results showed a high rate of correct answers (greater than 95%).

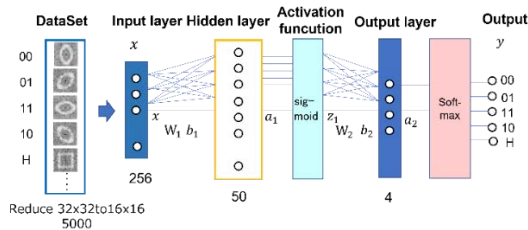


Figure 5. Configuration of a neural network

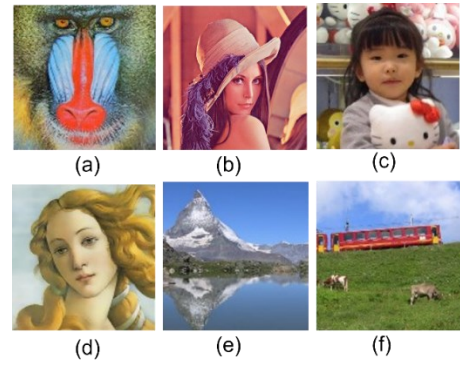


Figure 6. Evaluation image (512×512)

VI.EVALUATION AND DISCUSSION

A. Evaluation of Embedded Image Quality

The green noise pattern is a band-limited pattern with spatial frequency f restricted to $f_{min} < f < f_{max}$, which is set to be less responsive for the human visual system and responsive to the printer. In our method, the watermark is embedded block by block, but the boundary between the two is hardly noticeable because of the random dots.

PSNR (Peak Signal to Noise Ratio) for embedded image quality is

$$PSNR = -20 \log_{10} \left(\frac{gain}{2} \right) \quad (8)$$

which is a function of $gain$ only. The PSNR value does not take into account visual characteristics and does not reflect actual vision, so it must be evaluated using the visual transfer function (VTF) of the visual system.

The VTF is expressed as follows [6].

$$VTF = 2.72e^{-0.138x} (1.3 - e^{-0.1x}) [cycles/degree] \quad (9)$$

We re-evaluate the PSNR by including the VTF response and call it the visual PSNR. The visual PSNR of a printed image placed at a clear distance ($L=300$ mm) improves by 5~7 dB when VTF is considered. When print resistance is required, $gain$ is about 0.125~0.1875, and the visual PSNR is about 26~28 dB.

When only electronic data is required for operation, $gain$ should be less than 0.0625 and the PSNR should be more than 30dB. Furthermore, embedding the artifacts in the blue component (B channel) of the image data will further reduce the visibility of the artifacts.

TABLE 1. PERCENTAGE OF CORRECT READINGS (ELECTRONIC DATA)

Image	(a)	(b)	(c)	(d)	(e)	(f)
Correct answer rate	0.988	1.0	1.0	1.0	1.0	1.0

gain=0.125

TABLE 2. CORRECT IDENTIFICATION RATE (PRINT/SCANNED DATA) 175PPI

Image	(a)	(b)	(c)	(d)	(e)	(f)
Correct answer rate	0.961	0.996	0.988	0.9960	0.996	0.973

B. Evaluation of Image Editing and Processing Resistance

Figure 7 shows the percentage of correct responses to various attacks by image editing and processing of Lena images (electronic data) with embedded watermarks. For processing that changes the image position or size, pixel interpolation was performed from the four edges of the image using bilinear transformation to create a corrected image of a predetermined size, after which the watermark was extracted.

The results show that the watermark is robust to image processing that causes luminance changes, but the watermark disappears and the correct rate decreases when the cluster dot pattern is easily lost when applied strong smoothing etc.

C. Evaluating the resistance of sub-information

The sub-information could be detected in the range of 0.75- to 1.5 times image enlargement. For image rotation, the sub-information could be detected in the range of 0° to 5°. However, the pattern was lost during the smoothing process and could not be detected. This is because a checkerboard pattern of one pixel each was used as the sub-information. This indicates that the sub-information is not print-resistant.

Although a checkerboard pattern with a width of two pixels can be used to achieve print resistance, it causes crosstalk with the main information (watermark information). Therefore, it is necessary to embed the sub-information only in the marker so that it does not affect the main information (watermark information).

D. Evaluation and discussion of camouflage characteristics

Let the image size be $W \times H$, and the block size be B . Because there are eight inversion patterns and $(B-1)^2$ phase shift patterns, the occurrence rate of the same pattern η is expressed by the following equation:

$$\eta = \frac{WH}{4 \times 8B^2(B-1)^2} \quad (10)$$

When $B=32$, $\eta=0.0083$, which indicates that the frequency of appearance of the same pattern is extremely low.

Attack and intensity	Attacked image	Correct answer rate	Attack and intensity	Attacked image	Correct answer rate
Embedded image (No attack)		1.	Q=50		1.0
Smoothing	2 pixels	1.	Q=30		0.973
	3 pixels	0.844	Q=20		0.863
	Gaussian 5 x 5, $\sigma=2$	0.910	Resizing	$\times 1.5$	1.
Shading	Amplitude = 32/255	1.		$\times 0.75$	1.
	Amplitude = 64/255	1.	Rotation	10°	0.996
Gaussian Noise	$\sigma=10$ (3 σ)	1.		45°	1.
	$\sigma=30$	0.996			

Figure 7. Correct response rate for various types of image editing

In the case of the block average attack, we classify the blocks into $p_i(x,y)$ ($i=0,\dots,3$), add them up, and obtain the average value $C_i(x,y)$ using the following formula:

$$C_i(x,y) = \frac{1}{N} \sum W_n(x,y) = \frac{1}{N} \sum_n i_n(x,y) + gain \times \frac{1}{N} \sum_n W_n(x,y) \quad (11)$$

Σ is the sum of the pixel units in each block, and N is the number of patterns that appear. If there is no camouflage pattern, then,

$$\lim_{N \rightarrow \infty} C_i(x,y) = gain \cdot p_i(x,y) + const, \quad (12)$$

and the pattern was detected. On the other hand, when the camouflage pattern is used, $p_{i,k}(x,y)$ is random, then,

$$\lim_{N \rightarrow \infty} C_i(x,y) = const, \quad (13)$$

and no embedded patterns were extracted. It can be stated that the introduction of camouflage patterns makes it difficult to decipher these patterns. Therefore, the frequency of appearance of the same pattern is quite low, and block-averaging attacks become difficult.

Figure 8 shows the extraction of patterns by block averaging operation when a Lena image is embedded with $R=32$, and it can be observed that patterns are not extracted when camouflage is present.

E. Resistance evaluation by image cropping

A synchronization marker was embedded into an HD image (1920×1080) to evaluate its performance against cropping. The image of HD size was divided into 640×256 regions, and 318 bits (39 ASCII characters) were embedded into each region. The embedded area was identified by detecting the marker position. Figure 9 shows the extraction of watermark information from the marker embedded image. Figure 9(a) shows the original image (1920×1080), (b) is the image divided into 640×256 and embedded, (c) is 950×512 by cropping, and (d) is the image obtained from the image in (c).

In (e), the embedded area was extracted from the position of the marker, and in (f), the watermark information was extracted, and the correct answer rate was 0.970.

F. Evaluation and discussion of key size and security

The embedding pattern was changed for each image. The unique ID of the copyright holder is used as the seed value to obtain its own ID pattern, and each image has its own SubID to change the amount of phase shift. In other words, in taking the following example:

Image a of copyright holder A: $ID_A + SubID_a$

Image b: $ID_A + SubID_b$

...

Image p of copyright holder B: $ID_B + SubID_p$

Image q: $ID_B + SubID_q$

...

Suppose that ID_A is Mr. A's unique pattern (such as his fingerprint), and by using a pattern with different phase shifts

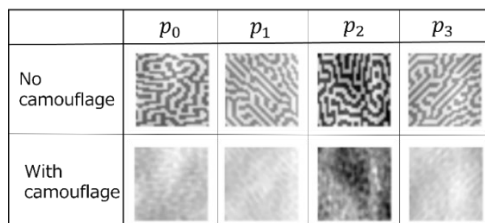


Figure 8. Block averaging operation with/without camouflage

for each image, the same pattern will not be used, and regardless of whether the key is stolen, other images cannot be accessed using the key.

Furthermore, by using the seed value of the random number (there are two types: inversion pattern and phase shift pattern) as a password, even if the key is stolen, the image cannot be accessed without the password.

The green noise pattern is obtained differently depending on the initial random number pattern (the initial pattern indicates white noise) at the time of pattern generation. The number of patterns can be approximately $R^2 C_{R^2/2}$ patterns if the block size is R , and can be almost infinitely generated.

G. Evaluation of tolerance by introducing error correcting codes

We introduced Reed-Solomon (RS) codes as error correcting codes in the embedded information. Using Irreducible polynomial $x^8 + x^4 + x^3 + x + 1$ in expanded Galois field $GF(2^8)$ consisting of 2^8 elements, RS(16,6) encoder/decoder, that can correct up to 3 words in 16 words, was implemented. In this case, the generation polynomial is $x^6 + a^{166}x^4 + x^4 + a^{134}x^3 + a^5x^2 + a^{176}x + a^{15}$. As a result of embedding watermark information using this RS codes, total amount of information that can be embedded is reduced to 10/16 compared to the case without error correction codes, but the correct extraction rate is increased close to 100% for both printed and electronic images.

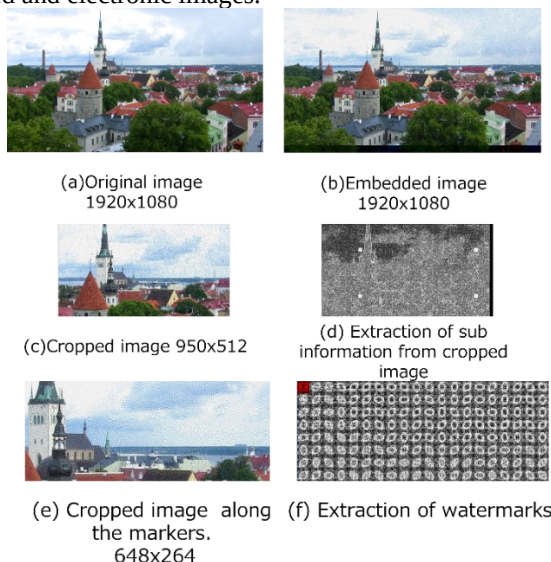


Figure 9. Watermark extraction conducted by Sub-Information

VII. CONCLUSIONS

The green-noise diffused watermarking method is resistant to printing and scanning, and the embedded pattern is not easily visible. We extended this method to extract watermark information from cropped images by embedding markers. We used machine learning to improve the identifier accuracy and correct watermark extraction rate of over 95% are obtained from printed images. Furthermore, we embedded sub-information in the marker pattern to obtain accurate block synchronization from the cropped images.

REFERENCES

- [1] T. Kitamura and M. Nakashizuka, "Data Embedding Image Halftoning by Block Error Diffusion," *IEEEJ*, Vol. 31, No. 4, pp. 584-593, 2002. [in Japanese]
- [2] N. Kawamura, "Digital Halftoning by Stochastic Dither Screen Mask Representing Green to Blue Noise Characteristics," *IEEEJ Trans. on IEVC*, Vol. 5, No. 1, pp. 34-41, 2017.
- [3] K. Yamanaka, M. Iwata and K. Kise, "A consideration of Digital Watermarking Method to Printed Images," *The 30th Symposium on Cryptography and Information Security(SCIS)*, 2013. [in Japanese]
- [4] N. Kawamura, H. Kang and K. Iwamura, "Secure Multi-level Green-Noise Diffused Watermarking with Printing Durability," *IEEEJ*, Vol. 49, No. 2, pp. 110-118, 2019. [in Japanese]
- [5] S. Kuse and K. Iwamura, "A method to have a robustness against average value attack to digital watermark," *IPSI SIG Technical Report*, Vol.2010-CSEC-51, Vol. 2, 2012. [in Japanese]
- [6] R. P. Dooley, "Prediction brightness appearance at edges using linear and non-linear visual describing functions," *SPSE Annual Meeting*, 1975.

Yoshi Michael Cho is pursuing Bachelor of Engineering in field of electrical engineering with Tokyo University of science, Japan.

His main research interests include watermarking and deep learning.

Hiroyuki Imada received B.E. and M.S. in field of electrical engineering, from Tokyo University of Science, Japan, in 2019 and 2021.

His main research interests include watermarking and deep learning.

Naoto Kawamura received M.S degree from Kyushu University in 1971, Ph.D. from the University of Tokyo in 1990. He has been working in Canon Inc. R&D division engaged in development of digital devices such as laser beam printer, color laser coping machine and research of image processing technologies for various imaging devices.

He is currently working as a technical consultant at Kawamura Techno Lab.

Hyunho Kang is currently an Associate Professor in the Department of Electronic Engineering at National Institute of Technology, Tokyo College, Japan; he has held this position since April 2017. He received his Ph.D. from the University of Electro-Communications, Tokyo, in 2008. From September 2010 to March 2013, he was an AIST Postdoctoral Researcher at the National Institute of Advanced Industrial Science and Technology (AIST), Japan, where his research work focused mainly on the evaluation of physical unclonable functions. From April 2013 to March 2017, he was an Assistant Professor in the Department of Electrical Engineering at Tokyo University of Science, Japan.

His main interests are machine learning, deep learning, information security applications, multimedia security (steganography, digital watermarking), biometric security and physical unclonable functions.

Dr. Kang is a senior member of IEICE and a member of IPSJ.

Keiichi Iwamura received B.S. and M.S. degrees in Information Engineering from Kyushu University in 1980 and 1982, respectively. During 1982–2006, he was with Canon Inc. He received a Ph.D. from Tokyo University. He is now a Professor at the Tokyo University of Science.

His subjects are coding theory, information security, and digital watermarking.

Dr. Iwamura is a fellow of the Information Processing Society of Japan and a fellow of the Institute of Electronics, Information and Communication Engineers