ICIC Express Letters
Part B: Applications
Volume **12**, Number **7**, July **2021**

ICIC International ©2021 ISSN 2185-2766
pp. **659–668**

DATA EMBEDDING METHOD FOR PRINTED IMAGES USING MULTILAYER NEURAL NETWORKS WITH PARTIALLY FIXED WEIGHTS

Hideaki Orii[1,*], Takaharu Kouda[1] and Hideaki Kawano[2]

[1]Department of Electrical Engineering
Faculty of Engineering
Fukuoka University
8-19-1 Nanakuma, Jonan-ku, Fukuoka 814-0180, Japan
[*]Corresponding author: oriih@fukuoka-u.ac.jp

[2]Department of Electrical Engineering and Electronics
Faculty of Engineering
Kyushu Institute of Technology
1-1 Sensui-cho, Tobata-ku, Kitakyushu, Fukuoka 804-8550, Japan

Received November 2020; accepted February 2021

Abstract. *A technique for data embedding in images, also known as watermarking and steganography, is used in various applications such as copyright protection and security. Recently, watermarking has been applied not only to electronic media but also to printed textures and other patterns for information presentation. In some cases, a digital watermarking method using the discrete cosine transform (DCT) has been employed. In this paper, we propose a novel data embedding method for printed images using neural networks. In the proposed method, the process of embedding and restoring data is represented by a neural network, and the process of embedding data in DCT coefficients is optimized by the learning of the neural network. In the experiments, we applied the proposed method to images with various textures and confirmed its effectiveness.*
Keywords: Data embedding, Neural networks, DCT

1. **Introduction.** In recent years, it has become common to use two-dimensional barcodes (QR codes) to represent arbitrary information on printed materials, such as posters and leaflets, and to add them to objects to present additional information. However, QR codes are black-and-white and have a random texture. Therefore, when they are added to posters or other objects, even if they are displayed in a small size, they may create a strong visual impression and spoil the design. A technique for data embedding in images, also known as watermarking and steganography, has been traditionally studied for the purpose of embedding copyright information [1]. Recently, watermarking has been applied not only to electronic media but also to printed textures to solve the foregoing problems. Various methods have been proposed to embed information in printed images without spoiling their appearance, by applying conventional watermarking technology to printed images [2, 3, 4]. In these methods, for example, the image is converted to a frequency representation, and the information is embedded in the coefficients of the frequency representation to provide robustness against printing and capturing. However, there are some problems such as the limitation on the amount of information that can be embedded and the existence of parameters that must be set manually.

A data embedding method for information presentation needs to be resistant to image deterioration due to printing and capturing. To be specific, it needs to be robust against changes in the brightness of the entire image due to printing and geometric changes due to

DOI: 10.24507/icicelb.12.07.659

photography. To make the method robust, a digital watermarking method using the discrete cosine transform (DCT) has been used [2]. In the conventional method, a watermark is embedded in the coefficients of the DCT component of an image, and the watermark is detected by obtaining the coefficient from the image after printing and capturing. At this time, there is a degree of freedom in the position of the DCT coefficients when embedding the information, but an optimum determination method has been insufficiently studied.

On the other hand, neural networks have shown remarkable performance in fields such as image recognition and speech recognition in recent years [5, 6, 7, 8]. We have also proposed an image conversion method using the framework of neural networks, and achieved good results [9]. This is achieved by increasing the expressiveness of the input/output characteristics of the network by deepening the layer of the neural network, and the relationship between the input and output of the training data with its powerful expressive power.

In this paper, we propose a novel method that uses neural networks to embed information in printed images while preserving the appearance of the target image. In the proposed method, the "information embedding/extraction" process and the "image geometric transformation" that occur during the printing/capturing process are explicitly represented as network weight parameters in the neural network (NN) model, so that the embedding/extraction parameters are optimized during the training process.

The rest of this paper is organized as follows. In Section 2, we give an overview of the data embedding process with DCT and the details of the proposed method. Section 3 presents the experimental results and discussions. Finally, the conclusions are given in Section 4.

2. **Proposed Method.**

In the conventional information embedding method using DCT, embedding is performed by changing the DCT coefficient of the original image in accordance with the information expressed in binary. In this process, the coefficients on the diagonal of the DCT image are changed so as not to be affected by affine transformation due to the capture of printing. Further, to prevent interference when composing, the embedding positions are arranged at equal intervals discretely on the diagonal line. However, these embedding positions are experimentally determined, and it is not known whether they are optimal.

In the proposed method, the information embedding position in the DCT coefficient is optimized in the learning process of the neural network. To be specific, we describe a sequence consisting of DCT transformation, embedding processing, inverse DCT transformation, geometric transformation, DCT transformation, and decoding of images as weights of neural networks. We optimize the embedding process parameters based on the mechanism of the backpropagation algorithm in neural networks.

An overview of conventional data embedding with DCT is given in Section 2.1, and an overview of neural networks and their learning process is given in Section 2.2. The details of the optimization method for data embedding in an image using the multilayer neural network with partially fixed weights and the entire algorithm of the proposed method are described in Section 2.3.

2.1. **Data embedding with DCT.**

T. Mizumoto and K. Matsui proposed a watermarking method for the DCT area that has robustness against the printing and capturing processes [3]. In this method, first, the original image is converted into CMYK format, and two-dimensional DCT is performed on each color plane. Next, for each plane after conversion, information is embedded with the following procedure.

Let N be the length of the embedded bit sequence and $x[n]$ be the bit sequence. As the frequency band to be used for embedding, a straight line extending from the DC component in the direction of the angle θ with respect to the *u axis* is set as an embedding

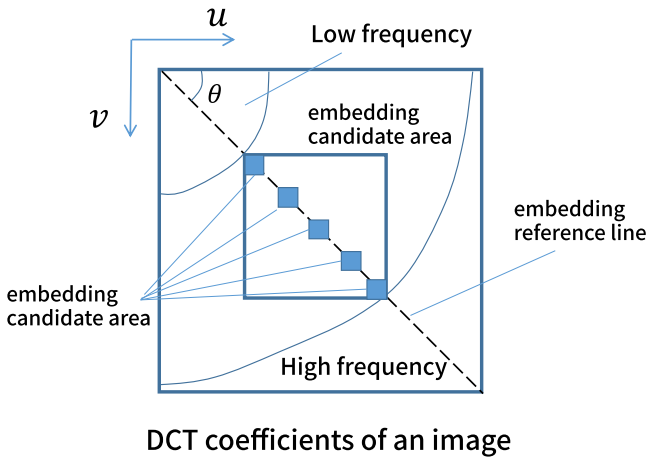


FIGURE 1. DCT coefficients of an image and embedding area

reference line, and an embedding candidate area is defined around the embedded reference line as shown in Figure 1. This makes it possible to avoid the influence of image rotation, by embedding with this line as the center. The high-frequency-component area is avoided to prevent the influence of scaling and sharing. The candidate region for embedding is also divided into $a \times a$. An area of $b \times b$ is provided between the areas to prevent elements from interfering due to aliasing.

A point close to the DC coefficient among the intersection points of the embedding reference line and the embedding candidate region is set as the embedding start point P, and E_p is a watermark pointer. For each area, the embedding is performed as follows:

1) $E_p = 0$.
2) For DCT coefficients in the embedding area, if $x[E_p \bmod N] == 0$, then V_0 is embedded, else if $x[E_p \bmod N] == 1$, then V_1 is multiplied by the sign of the original coefficient and embedded.
3) $E_p = E_p + 1$, and the point P is moved by $a + b$ along the embedded reference line.

In the detection, the DCT areas in which data are embedded are examined in the same order that they were embedded, the absolute values of the DCT coefficients in each area are taken, and their maximum values are stored in $X[E_p]$. The value of $X[E_p]$ is determined by the values of V_0 and V_1 used for embedding. Thus, by setting the appropriate V_0 and V_1, it is possible to distinguish the two groups by the threshold value.

2.2. **Neural networks.** Neural network models are essentially simple mathematical models defining a function $g : \boldsymbol{X} \rightarrow \boldsymbol{Y}$. In a neural network, the j-th neuron in a layer calculates and outputs a value y_j as follows:

$$y_j = f(\Sigma_i x_i \cdot w_{ij} + b_j). \tag{1}$$

Here, x_i is the i-th input value for a layer. w_{ij} is the weighted value of the j-th neuron for the i-th input value. b_j is the bias value of the j-th neuron. $f(\cdot)$ is the transfer function of the j-th neuron. Therefore, when the number of neurons in a layer is M, an output vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_M]^T$ of this layer with respect to an input vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]^T$ can be obtained as follows:

$$\boldsymbol{y} = f\left(\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & \cdots & w_{MN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_M \end{bmatrix}\right) \tag{2}$$

$$= f\left(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}\right). \tag{3}$$

As mentioned earlier, the relationship between an input vector and an output vector in a layer is defined by the weighted value matrix $\boldsymbol{W}$, bias vector $\boldsymbol{b}$, and transfer function $f(\cdot)$. In a multilayer neural network, the neurons in each layer calculate the output vector based on the previous layer output and send the vector as the next layer input. Therefore, an input vector of a multilayer neural network is mapped a number of times from the input layer to the output layer. The training of a neural network is generally supervised with a training data set consisting of many pairs of an input vector and its desirable output vector. The weighted value matrix $\boldsymbol{W}$ and bias vector $\boldsymbol{b}$ of each layer are updated based on the error of the output vector. This is called the backpropagation algorithm. To use this algorithm, the transfer functions of each layer must be differentiable.

2.3. **Optimization method for data embedding in images using multilayer neural networks.** Neural networks can express various processes by connecting layers capable of expressing various mappings, as mentioned in the previous section. In the proposed method, we design a neural network that can perform a sequence consisting of DCT transformation, embedding processing, inverse DCT transformation, geometric transformation, DCT transformation, and decoding of images by connecting layers capable of expressing various mappings. Thus, embedding process parameters can be optimized based on the mechanism of the backpropagation algorithm in neural networks.

DCT as described in Section 2.1 is defined by the following equation:

$$X(i,j) = \alpha_i \alpha_j \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left\{ I(m,n) \cos \frac{\pi(2m+1)i}{2M} \cos \frac{\pi(2n+1)j}{2N} \right\}. \tag{4}$$

Here, $I(m,n)$ is an image; $X(i,j)$ is the DCT image; M and N are the height and width of the image, respectively; and α_i and α_j are expressed as follows:

$$\alpha_i = \begin{cases} \dfrac{1}{\sqrt{M}} & i = 0 \\ \sqrt{\dfrac{2}{M}} & i = 1, 2, \ldots, M-1 \end{cases}, \qquad \alpha_j = \begin{cases} \dfrac{1}{\sqrt{N}} & j = 0 \\ \sqrt{\dfrac{2}{N}} & j = 1, 2, \ldots, N-1 \end{cases}. \tag{5}$$

From the mathematical expression, the DCT image of an image can be calculated as a weighted sum of all the pixel values of the original image. This is nothing but a linear mapping of $(M \times N) \to (M \times N)$, and it can be expressed as a layer of a neural network. The inverse DCT transformation, affine transformation, image blur, etc., are also similar.

Figure 2 shows the architecture of the multilayer neural network in the proposed method. "fc1", "fw1", "fvw1", "fw2", and "fc2" are functional blocks with one or more layers. The network of each layer has a weighted matrix corresponding to each process in data embedding. When DCT image data are input to this network, an information-embedded image and a bit string decoded from the image are output. Therefore, by using the original image and the desired bit string as the teacher data of the neural network, each layer of the network is optimized for data embedding. In this training, by fixing the weights of the layers corresponding to the inverse DCT (IDCT) transformation and the DCT transformation, the parameters in fc1 and fc2 are trained assuming DCT processing. "fvw1" is a layer that simulates affine transformations of images, such as parallel translation, rotation, and image blurring, and its parameter is randomly set in learning iterations. This makes it possible to optimize data embedding processing that is robust against affine transformation of the embedded image.

Figure 3 shows data embedding and decryption with the trained neural network. It can be achieved by partially using the trained neural network.

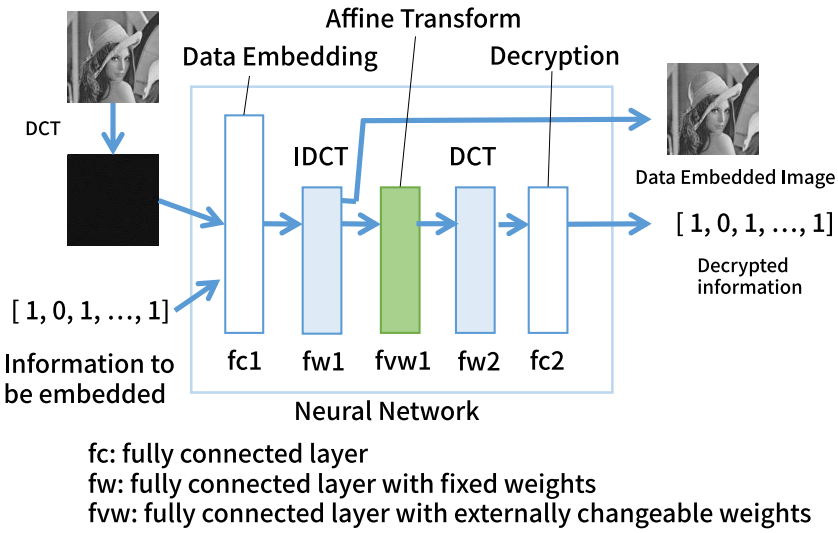


FIGURE 2. Architecture of a multilayer neural network with partially fixed weights

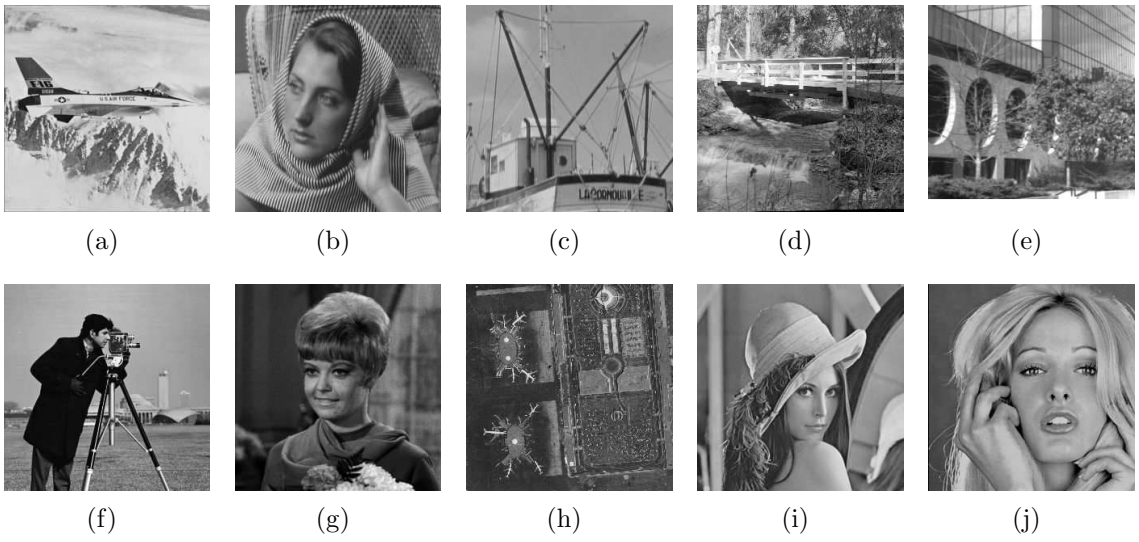
(a) (b)

FIGURE 3. Data embedding and decryption using a part of the trained neural network: (a) Data embedding, and (b) data decryption

3. Experimental Results and Discussions.

3.1. **Experimental settings.** In the experiments, 100,000 samples of image patches were prepared for the learning of the neural network in the proposed method, and the learning of the neural network was performed using it. The bit strings for embedding were generated randomly. Figure 4 shows the images used in the experiments. The samples for training were cropped to the desired size from these images randomly. In this study, the patch size for input to the neural network is decided as 32×32, and the image size of the embedding target is 512×512. Therefore, (code length $\times$ (512/32) $\times$ (512/32)) bits can be embedded in the target image in a lattice order.

The "fvw1" layer is the functional block used to simulate image affine transforms. In this experiment, we designed a Gaussian blur filtering layer and an image shifting layer, and connected those layers as "fvw1". The standard deviation of the Gaussian filter was randomly set to a value from 0.5 to 1.5. In the image shifting layer, the input images are shifted vertically and horizontally according to a shift parameter determined in advance. When the shift parameter is set to "2", the image is vertically and horizontally shifted in the range -2 to 2. Thus, it is possible to simulate the resolution conversion and positional shift that occur when printing and capturing images.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

FIGURE 4. Images used in the experiments: (a) Airplane, (b) Barbara, (c) Boat, (d) Bridge, (e) Building, (f) Cameraman, (g) Girl, (h) Lax, (i) Lenna, and (j) Woman

In the proposed method, the designs of the "fc1" block and "fc2" block allow for great flexibility. We used two layers of a fully connected layer at those blocks in this experiment. The ReLU function was used as the activation function at each layer, and the sigmoid function was used as the activation function in the output layer of "fc2".

In the experiment, the extraction accuracy rate of an embedded bit string and the peak signal to noise ratio (PSNR) with the original image are used as the index of performance evaluation. The defining formulas are shown as follows:

$$\text{Extraction accuracy rate [\%]} = \frac{\text{Number of bits correctly decrypted}}{\text{Length of embedded bit string}} \times 100 \tag{6}$$

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \{I(i,j) - K(i,j)\}^2 \tag{7}$$

$$\text{PSNR [dB]} = 10 \cdot \log_{10} \frac{MAX_I^2}{MSE} \tag{8}$$

Here, $I(i,j)$ and $K(i,j)$ are pixel values in images. m and n are the width and height of an image. MAX_I is the maximum allowable value for images.

3.2. **Performance evaluation of the proposed method.** In this section, we evaluate the performance of the proposed method, and verify the relationship between the performance and the parameters of the method. We selected six images – (a), (b), (c), (f), (i), and (j) – from the experimental images of Figure 4 that have textures that easily show design damage, and used them as target images for data embedding. In the experiment, a bit string was embedded in an image according to the setting described in the previous section. The embedded image was printed out, and then imported into the computer by using a scanner. The image size of a printed image was approximately 8 cm square, and the scanner capture resolution was 150 dpi. The resolution of a scanned image was 454×454. Therefore, we resized a scanned image to 512×512, and decrypted with the proposed method.

Table 1 shows the extraction accuracy rate [%] of decrypted bit string for various shift parameters in the proposed method. The embedded length of the bit string is fixed to "8" in this experiment. In the table, the higher the value, the better the result. In Table

TABLE 1. Evaluation of performance of data embedding (average bit extraction accuracy rate [%] with respect to shift parameter)

Image	Shift parameter [pix]			
	0	1	2	4
Airplane	65.04	90.04	83.94	56.88
Barbara	51.22	87.06	68.51	37.94
Boat	54.49	93.26	81.69	52.98
Cameraman	51.66	83.06	81.74	45.12
Lenna	50.93	74.61	64.89	46.73
Woman	53.22	67.09	81.01	50.10

TABLE 2. Evaluation of quality of data-embedded image (PSNR [dB] between the original image and the data-embedded image) with respect to shift parameter

Image	Shift parameter [pix]			
	0	1	2	4
Airplane	38.26	32.07	32.97	30.50
Barbara	36.64	31.32	32.74	30.36
Boat	40.67	33.67	34.49	31.78
Cameraman	37.20	31.22	32.55	29.92
Lenna	40.07	33.16	34.28	31.40
Woman	40.19	33.24	34.30	31.45

(a) (b) (c) (d)

FIGURE 5. Data-embedded images for various shift parameters: (a) 0 pix, (b) 1 pix, (c) 2 pix, and (d) 4 pix

1, good performance is shown when the shift parameter is "1" or "2", but not otherwise. Because image shifting did not occur during the learning of the network when the shift parameter was "0", it appears that the method did not work effectively for the image that was actually printed and scanned. On the other hand, when the shift parameter was "4", the learning of the network did not work well because the amount of image shifting was too large for the image patch size (32×32). Table 2 shows the PSNR [dB] between an original image and a data-embedded image for various shift parameters in the proposed method, and Figure 5 shows the data-embedded images in this experiment. As can be observed from the table and the figures, the data embedding is functioned effectively while suppressing deterioration in the appearance of the images.

Tables 3 and 4 show the evaluation with respect to the change in the length of the bit string in the proposed method, and Figure 6 shows the data-embedded image in this experiment. The shift parameter is fixed to "2" in this experiment. As the embedded bit

TABLE 3. Evaluation of performance of data embedding (average bit extraction accuracy rate [%] with respect to embedding bit length)

Image	Embedding bit length [bit/patch]			
	8	16	32	64
Airplane	83.94	77.69	71.61	80.71
Barbara	68.51	59.30	71.31	85.67
Boat	81.69	85.45	72.47	78.25
Cameraman	81.74	71.58	69.96	84.37
Lenna	64.89	84.62	72.88	86.59
Woman	81.01	85.65	74.81	85.78

TABLE 4. Evaluation of quality of data-embedded image (PSNR [dB] between the original image and the embedded image) with respect to embedding bit length

Image	Embedding bit length [bit/patch]			
	8	16	32	64
Airplane	32.97	31.52	29.33	26.26
Barbara	32.74	31.49	29.06	24.33
Boat	34.49	33.16	31.53	28.48
Cameraman	32.55	31.38	29.21	25.91
Lenna	34.28	32.65	30.37	27.27
Woman	34.30	32.66	30.46	27.44

(a) (b) (c) (d)

FIGURE 6. Data-embedded images for various embedding code lengths: (a) 8 bits, (b) 16 bits, (c) 32 bits, and (d) 64 bits

length increases, the deterioration in image quality increases, but data embedding works well.

3.3. **Performance comparison with conventional method.** In this section, we compare the performance of data embedding using the proposed method with that of the conventional method [3]. In this experiment, information was embedded in 512×512 images. The shift parameter of the proposed method was set to "2", and the length of the bit string in the proposed method was set to "8". Therefore, $(8 \times (512/32) \times (512/32) = 2048)$ bits can be embedded in the target image in a lattice order. The conventional method also employs some parameters for data embedding as mentioned in Section 2.1. In this experiment, we used $a = 6$, $b = 6$, the embedding start point $P = (128, 128)$, and $V_1 = 200$. The length of the embedded bit string was set to 24 in the conventional method. Therefore, bit information was embedded in the pixels from pixel $(128, 128)$ to pixel $(415, 415)$ in a

DCT image diagonally. In this experiment, we embedded information in a single plane of grayscale brightness, not CMYK color planes.

Table 5 shows the evaluation results of the proposed method and the conventional method. Each evaluation value is the average value of the results of the images (Airplane, Barbara, Boat, Cameraman, Lenna, Woman). It can be observed from the table that the proposed method is slightly superior to the conventional method. Figure 7 shows the data-embedded image in this experiment. Although the embedding patterns are different, embedding of information suppressing apparent degradation is achieved in both of them.

Table 5. Comparison of performance

	Conventional [3]	Proposed
Accuracy rate [%]	69.44	76.96
PSNR [dB]	29.77	33.56

(a) (b) (c)

Figure 7. Comparison of data-embedded images: (a) Original image, (b) data-embedded image with the conventional method [3], and (c) data-embedded image with the proposed method

4. **Conclusions.** We proposed a novel data embedding method for printed images using neural networks. In the proposed method, the process of embedding and restoring data is represented by a neural network architecture, and the process of embedding data in DCT coefficients is optimized by the learning of the neural network. In the experiment, we applied the proposed method to images having various textures. The results showed the effectiveness of the proposed method. For future studies, we will further examine the network structure of neural networks.

Acknowledgment. This work was supported in part by funding from Fukuoka University (Grant Nos. 185009 and 215010).

REFERENCES

[1] H.-Y. Fan, Z.-M. Lu and Y. Liu, The digital image watermarking scheme using low frequency construction and histogram, *International Journal of Innovative Computing, Information and Control*, vol.16, no.1, pp.367-384, 2020.

[2] M. Muneyasu, Watermarking techniques for information retrieving –Information embedding into printing images–, *IEICE Fundamentals Review*, vol.2, no.2, pp.53-62, 2008.

[3] T. Mizumoto and K. Matsui, Robustness investigation of DCT digital watermark for printing scanning, *IEICE Information and Communication Engineers*, vol.85, no.4, pp.451-459, 2002.

[4] A. Okou and S. Miyaoka, Automatic determination method of the threshold for print-type watermark detection, *IEICE Technical Report*, vol.111, no.457, pp.11-16, 2012.

[5] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, pp.2278-2324, 1998.
[6] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, vol.25, pp.1097-1105, 2012.
[7] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Comput.*, vol.9, no.8, pp.1735-1780, 1997.
[8] I. Sutskever, O. Vinyals and Q. V. Le, Sequence to sequence learning with neural network, *Proc. of the 27th International Conference on Neural Information Processing Systems (NIPS'14)*, pp.3104-3112, 2014.
[9] H. Orii, H. Kawano, N. Suetake and H. Maeda, Color conversion for color blindness employing multilayer neural network with perceptual model, in *Image and Video Technology, PSIVT 2015, LNCS 9431*, T. Bräunl, B. McCane, M. Rivera and X. Yu (eds.), Cham, Springer, 2016.