Data Embedding Method for Printed Images
Using Deep Neural Networks

Hideaki Oriria,*, Takaharu Kouda, Hideaki Kawano

aDepartment of Electrical Engineering, Faculty of Engineering, Fukuoka University
8-19-1 Nanakuma, Jonan-ku, Fukuoka 814-0180, JAPAN
bDepartment of Electrical Engineering and Electronics, Faculty of Engineering, Kyushu Institute of Technology
1-1 Sensui-cho, Tobata-ku, Kitakyushu, Fukuoka, 804-8550, JAPAN

*Corresponding Author: oriih @ fukuoka-u.ac.jp

Abstract

Data embedding techniques for images, also known as watermarking and steganography, are used in various applications such as copyright protection and security. In recent years, data embedding methods for images are applied not only to electronic media but also to printed textures and other patterns for information presentation. In some of them, a digital watermarking method using DCT (Discrete Cosine Transform) has been used. In this paper, we propose a novel data embedding method for printed images using deep neural networks. In the proposed method, the process of embedding and restoring data is represented by one deep neural network, and the embedding process into DCT coefficients is optimized by learning of the deep neural network. In the experiment, we apply the proposed method to the printed images have various textures. The results showed the effectiveness of proposed method.

Keywords: data embedding, deep neural networks

1. Introduction

Techniques for data embedding in images, also known as watermarking and steganography, are used in various applications such as copyright protection and security. Recently, watermarking is applied not only to electronic media but also to printed textures and other patterns for information presentation systems(1-3). In the information presentation system using the digital watermarking technique, information is embedded in advance in an image having a design such as a poster, the printed pattern is read by a camera or a scanner, and the information is decrypted from the read image. There is a method of using a two-dimensional barcode as a means of embedding information in the paper media, but this method may impair the design of the printed matter in some cases. When digital watermarking technique is used, there is an advantage that there is no such concern because information is embedded by preserving the original pattern possessed by the target media.

A data embedding method for information presentation needs to be resistant to image deterioration due to printing and capturing. Specifically, it needs to be robust against changes in the brightness of the entire image due to printing and geometric changes in photographing. In to make them robust, a digital watermarking method using DCT (Discrete Cosine Transform) has been used(1).

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 to embed the information, but an optimum determination method is insufficiently studied.

On the other hand, neural networks show remarkable performances in fields such as image recognition and speech recognition in recent years(4-7). We also proposed image conversion method using the framework of neural networks, and achieved good results(8). This is realized by increasing the expressiveness of input/output characteristics of the network by deepening the layer of the neural network, and the relationship between input and output of the training data with its powerful expressive ability.
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 one neural network, and the embedding process into DCT coefficients is optimized by learning of the neural network.

2. Proposed Method

2.1 Overview

In the conventionally proposed information embedding method using DCT (Discrete Cosine Transform), embedding is performed by changing the DCT coefficient of the original image in correspondence with the information expressed in binary. At 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. Also, in order not to interfere 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 to the DCT coefficient is optimized in the learning process of the neural network. Specifically, we describe a sequence of DCT transformation, embedding processing, inverse DCT transformation, geometric transformation, DCT transformation and decoding process of images as weights of neural networks. We optimize embedding process parameters based on the mechanism of back propagation algorithm in neural networks.

The overview of conventional data embedding with DCT is described in Sec. 2.2, and the overview of neural networks and its learning process are described in Sec. 2.3. The detail of the optimization method for data embedding in an image using the multi-layer neural network with partially fixed weights and the entire algorithm of the proposed method are described in Sec. 2.4.

2.2 Data embedding with DCT

T. Mizumoto et. al. proposed a watermarking method to the DCT area that has robustness for printing and capturing process\(^2\). 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 by the following procedure.

Let \( N \) be the length of the embedded bit sequence and \( x[n] \) be the bit sequence. As a frequency band to be used for embedding, a straight line extended from the DC component in the direction of the angle \( \theta \) with respect to the \( u \) axis is set as an embedding reference line, and an embedding candidate area is defined around the embedded reference line as shown in Fig.1. Thereby, there is a possibility that influence of image rotation can be avoided by embedding with this line as the center. High-frequency component area is avoided so as not to be affected by scaling and sharing. The candidate region for embedding is also divided into \( a \times a \). The area of \( b \times b \) present between the areas is provided 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, embed it as follows.

\begin{enumerate}
  \item \( E_p = 0 \)
  \item For DCT coefficients in the embedding area, if \( x[E_p \mod N] == 0 \) then \( V_0 \) is embedded, else if \( x[E_p \mod N] == 1 \) then \( V_1 \) is multiplied by the sign of the original coefficient and embedded.
  \item \( E_p = E_p + 1 \), and the point \( P \) is moved by \( a + b \) along the embedded reference line.
\end{enumerate}

In the detection, the DCT area in which data is embedded is examined in the same order as when it is embedded, the absolute values of 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.
2.3 Neural Networks

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

$$y_j = f\left(\sum_i (x_i \cdot w_{ij} + b_j)\right). \quad (1)$$

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

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

$$= f(\mathbf{W} \mathbf{x} + \mathbf{b}) \quad (2)$$

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

2.4 Optimization Method for Data Embedding in Image Using Deep Neural Network

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

![Architecture of deep neural network of proposed method.](image-url)
DCT as described in Sec.2.2 is defined by the following equation.

\[ X(i,j) = \alpha_i \alpha_j \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (I(m,n) \cos \frac{\pi (2m+1)i}{2M}) \cos \frac{\pi (2n+1)j}{2N} \]  

(4)

Here, \( I(m,n) \) is an image, \( X(i,j) \) is DCT image, \( M, N \) are a height and a width of the image, and \( \alpha_i \) and \( \alpha_j \) are expressed as follows:

\[
\alpha_i = \begin{cases} 
\frac{1}{\sqrt{M}} & i = 0 \\
\frac{2}{M} & i = 1, 2, \ldots, M - 1
\end{cases}
\]

(5)

\[
\alpha_j = \begin{cases} 
\frac{1}{\sqrt{N}} & j = 0 \\
\frac{2}{N} & j = 1, 2, \ldots, N - 1
\end{cases}
\]

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

Fig. 2 shows the architecture of the deep neural network in the proposed method. “fc1”, “fw1”, “fwv1”, “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. The prefix “fc-” is mean fully connected, and all neurons of each layer in the block are fully connected. The prefix “fw-” is mean fully connected with fixed weights, and all neurons of each layer in the block are fully connected with fixed weights. The prefix “fwv-” is mean fully connected with externally changeable weights, and the weights of the neurons in each layer in the block can be changed to arbitrary weights during learning. When DCT image data is 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 transformation and the DCT transformation, the parameters in fc1 and fc2 are trained assuming for DCT processing. Fwv1 is a layer that simulates the affine transformation of images, and its parameter is randomly set such as parallel translation, rotation, and image blurring in learning iteration. This makes it possible to perform optimization of data embedding processing which is robust against affine transformation to the embedded image.

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

3. Experimental Results and Discussions

3.1 Experimental Settings

In the experiments, 100,000 samples of image patches were prepared for learning of the neural network, and learning of NN was performed using it. The bit strings for embedding were generated randomly. Fig. 4 shows the images for the experiment. The samples for training were
cropped with the desired size from these images randomly. In this paper, the patch size for input to the neural network is decided as $32 \times 32$, and the image size of embedding target is $512 \times 512$. Therefore, $(\text{code length} \times (512/32) \times (512/32))$ bits can be embedded in the target image in a lattice order.

The “fvw1” layer is the functional block to simulate image affine transforms. In this experiment, we designed gaussian blur filtering layer and 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 of -2 to 2. Thereby, it is possible to simulate a resolution conversion and a positional shift in printing and capturing images.

In the proposed method, the designs of “fc1” block and “fc2” block have a lot of flexibility. We used two layers of 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, an extraction accuracy rate of an embedded bit string and PSNR (Peak Signal to Noise Ratio) with the original image are used as the index of performance evaluation. The definition formulas are shown follows.

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

\[
\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I(i,j) - K(i,j))^2
\]

\[
\text{PSNR [dB]} = 10 \cdot \log_{10} \frac{\text{MAX}_1^2}{\text{MSE}}
\]

Here, $I(i,j)$ and $K(i,j)$ are pixel values in images. $m$ and $n$ are a width and a height of a image. $\text{MAX}_1$ is the maximum allowable value for images.

### 3.2 Performance Evaluation of Proposed Method

In this section, we evaluate the performance of the proposed method, and verify the relation between the performance and the parameters of the proposed method. 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 by the printer, and then

<table>
<thead>
<tr>
<th>Image</th>
<th>Shift parameter [pix]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Airplane</td>
<td>65.04</td>
</tr>
<tr>
<td>Barbara</td>
<td>51.22</td>
</tr>
<tr>
<td>Boat</td>
<td>54.49</td>
</tr>
<tr>
<td>Cameraman</td>
<td>51.66</td>
</tr>
<tr>
<td>Lenna</td>
<td>50.93</td>
</tr>
<tr>
<td>Woman</td>
<td>53.22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image</th>
<th>Shift parameter [pix]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Airplane</td>
<td>38.26</td>
</tr>
<tr>
<td>Barbara</td>
<td>36.64</td>
</tr>
<tr>
<td>Boat</td>
<td>40.67</td>
</tr>
<tr>
<td>Cameraman</td>
<td>37.20</td>
</tr>
<tr>
<td>Lenna</td>
<td>40.07</td>
</tr>
<tr>
<td>Woman</td>
<td>40.19</td>
</tr>
</tbody>
</table>
taken into the computer by using the scanner. The image size of a printed image is about 8 cm square, and the scanner capture resolution is 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 with respect to the changes of the shift parameter in the proposed method. The embedded length of the bit string is fixed to “8” in this experiment. In the table, if the value is the higher, it is the better. In the table 1, the good performances are shown when the shift parameter is “1” or “2”, but it is not good otherwise. Since image shifting did not occur at the learning of the network when the shift parameter is “0”, it seems that it did not work effectively for the image that actually printed and scanned image. On the other hand, when the shift parameter is “4”, the reason is that

Table 3. Evaluation of performance of data embedding (average bit extraction accuracy rate [%] about embedding bit length).

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

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.

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

Table 5. Comparison of performance.

<table>
<thead>
<tr>
<th></th>
<th>Conventional (2)</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy rate [%]</td>
<td>69.44</td>
<td>76.96</td>
</tr>
<tr>
<td>PSNR [dB]</td>
<td>29.77</td>
<td>33.56</td>
</tr>
</tbody>
</table>

Fig. 5. Data embedding images with respect to changes of shift parameter. (a) 0 pix, (b) 1 pix, (c) 2 pix, (d) 4 pix.

Fig. 6. Data embedding images with respect to changes of embedding code length. (a) 8 bit, (b) 16 bit, (c) 32 bit, (d) 64 bit.
the learning of network could not be work well because the amount of image shifting was too large for the image patch size \(32 \times 32\). Table 2 shows PSNR [dB] between an original image and a data embedding image with respect to the shift parameter in the proposed method, and Fig. 5 shows the data embedding images in this experiment. As can be seen from the table and the figures, the data embedding is functioning effectively while suppressing deterioration of the appearance of the images.

Table 3 and Table 4 show the evaluation with respect to the change of the length of the bit string in the proposed method, and Fig. 6 shows the data embedding images in this experiment. The shift parameter is fixed to “2” in this experiment. As the embedded bit length increases, the deterioration of image quality increases, but data embedding works well.

### 3.3 Comparison of Performance with Conventional Method

In this section, we compare the performance of data embedding of the proposed method with the conventional method\(^2\). In this experiment, information was embedded in \(512 \times 512\) 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 has some parameter for data embedding as mentioned in Sec. 2.1. In this experiment, we used that \(a = 6, b = 6\), the embedding stat point \(P = (128, 128), V_1 = 200\). The length of embedded bit string was set to 24 in the conventional method. Therefore, bit information was embedded to the pixels on from \((128, 128)\) to the pixel \((415, 415)\) in a DCT image diagonally. In this experiment, we embedded into a single plane of grayscale brightness, not CMYK color planes.

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

### 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 one neural network architecture, and the embedding process into DCT coefficients is optimized by learning of the neural network. In the experiment, we apply the proposed method to the images have various textures. The results showed the effectiveness of proposed method. For future works, we will further examine the network structure of neural networks.

### Acknowledgment

This work was supported in part by funds (No. 185009) and (No. 155006) from the Central Research Institute of Fukuoka University.

### References


