Damageless Information Hiding using Neural Network on YCbCr Domain

Kensuke Naoe[†], *Yoshiyasu Takefuji*[††]

[†]*Graduate School of Media and Governance, Keio University, 5322 Endo, Fujisawa, Kanagawa, 252-8520, Japan*
[††]*Faculty of Environment and Information Studies, Keio University, 5322 Endo, Fujisawa, Kanagawa, 252-8520, Japan*

Summary
In this paper, we propose a new information hiding technique without embedding any information into the target content by using neural network trained on frequency domain especially on YCbCr domain. Proposed method can detect a hidden bit codes from the content by processing the selected feature subblocks into the trained neural network. Hidden codes are retrieved from the neural network only with the proper extraction key provided. The extraction keys, in proposed method, are the coordinates of the selected feature subblocks and the network weights generated by supervised learning of neural network. The supervised learning uses the coefficients of the selected feature subblocks as set of input values and the hidden bit patterns are used as teacher signal values of neural network. With our proposed method, we are able to introduce a information hiding scheme with no damage to the target content. There are many digital watermarking and steganographic algorithms been proposed, but there are difficulties to use one algorithm together with another because each other obstruct the embedded information and causing one to destroy another. Because our proposed method does not damage the target content, it has an ability to collaborate with another algorithm to strengthen the security of information hiding method.

Key words:
Digital Watermarking, Steganography, Information Hiding, Digital Rights Management, Neural Network

1. Introduction

The emergence of the Internet with rapid progress of information technologies, digital contents are commonly seen in our daily life distributed through the network. Because digital contents are easy to make an exact copy, illegal distribution and copying of digital contents has become main concerns for authors, publishers and legitimate owners of the contents [1].

Information hiding provides a reliable communication by embedding secret code into content for the purposes of intellectual property protection, content authentication, fingerprinting, covert communications, and etc. The researches in information hiding has a history [2] and namely the researches in digital watermarking and steganography have been active [3]. Both are very similar but their applications are different [4].

Digital watermarking became a key technology for protecting copyrights [5]. Digital watermarking protects unauthorized change of the contents and assures legal user for its copyright. Meanwhile, steganography conceals a hidden messages to a content but the existence of a message is kept secret [6]. In another word, steganography is a technique to conceal information into digital content files for purposes such as secret communication and covert channel. There are several ways to protect digital content. One can protect the content by encrypting it, but this avoids free distribution and circulation of the content through the network, which most of the time not desirable to the author of the content. Therefore, for the purpose of digital watermarking, content should not be encrypted or scrambled. It also needs to embed data to the content imperceptibly in order not to destruct the content from the original expression. Perceptible watermark are sometimes used, but it limits the use of the images. Therefore, main concern in this research area has focused on imperceptible watermark.

There are many digital watermarking and steganographic algorithms been proposed, but there are difficulties to use one algorithm together with another because each other obstruct the embedded information and causing one to destroy another. Because our proposed method does not damage the target content, it has an ability to collaborate with another algorithm to strengthen the security of information hiding method. This characteristic is useful where one already manage digital rights using one watermarking algorithm or controls the file integrity using hash functions. If one wishes to strengthen the robustness of watermark using another algorithm, one must examine and assure that applying the algorithm will not affect embedded watermarking signals in advance. Furthermore, applying another watermarking algorithm will alter the fingerprint of the content managed by hash functions and forces administrator to recalculate a new hash values after applying new watermarking algorithm, which most of the

Manuscript received September 5, 2008.
Manuscript revised September 20, 2008.

time, result in higher calculation cost and time. Because proposed method does not affect the target content at all, one can apply new watermark seamlessly without altering the fingerprint using proposed method.

2. Proposed Method

2.1 Overview of proposed method

In this section, we explain how proposed method hide information into target content and retrieve information from the target content without damaging it. With our method, the use of neural network is the key technique. The embedder adjusts a neural network weights with desired hidden bit code to target content by supervised learning of the neural network. This conditioned neural network works as a classifier to recognize a hidden bit pattern from the content which embedder associated to the target content. Therefore, extractor uses this neural network weights for extracting the hidden bit codes. For the applications for digital watermarking and steganography, this extraction keys must be shared among embedder and extractor in order to extract a proper hidden bit codes from the target content. Considering the difficulties for secret key transportation, this method should be applied in situations where the embedder and the extractor are same person or use certification authorities to assure the integrity of the key. Brief introduction of neural network and detailed explanation of proposed method discussing the procedures for embedding and extraction are demonstrated in the following subsections.

2.2 Neural network model

Proposed method uses a multi-layered perceptron model for neural network model. Multi-layered perceptron basically has a synaptic link structure in neurons between the layers but no synaptic link between the neurons within the layer itself [7]. Signals given to the input layer will propagate forwardly according to synaptic weight of neurons through the layers and finally reaches to the output layer as shown in Figure 1.

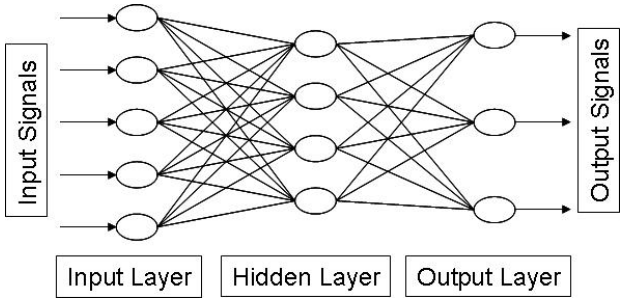

Figure 1 Multi-layered perceptron model

Signal that is being put to the neuron is converted to a certain value using a function and outputs this value as output signal. Normally sigmoid function is used for this model and is expressed as follows:

$$f(x) = \frac{1}{1+e^{-x}}. \tag{1}$$

Each synaptic link has a network weight. The network weight from unit *i* to unit *j* is expressed as w_{ij} and the output value for unit *i* is expressed as O_i. The output value for the unit is determined by the network weight and the input signal. Consequently, to change the output value to a desired value, adjustment of these network weights are needed. In proposed method, we use back propagation learning as learning method.

Back propagation learning is a supervised learning [8]. This method tries to lower the difference between the teacher signal and the output signal by changing the network weight. Changes of the network weight according to the difference in the upper layer propagate backward to the lower layer. This difference between the teacher signal values are called as error and often expressed as δ .

When teacher signal t_k is given to the unit *k* of output layer, the error δ_k will be calculated by following function:

$$\delta_k = (t_k - O_k) \cdot f'(O_k) \tag{2}$$

To calculate the error value δ_j for hidden unit, error value δ_k of the output unit is used. The function to calculate the error value δ_j for hidden unit *j* is as follows:

$$\delta_j = \left(\sum_k \delta_k w_{jk} \right) \cdot f'(O_j) \tag{3}$$

After calculating the error values for all units in all layers, then network can change its network weight. The network weight is changed by using following function:

$$\Delta w_{ij} = \eta \delta_j O_i \tag{4}$$

η in this function is called learning rate. Learning rate is a constant which normally has a value between 0 and 1

and generally represents the speed of learning process.

2.3 Information hiding algorithm

For embedding procedure, frequency transformation of the target content is processed. This frequency transformation is done after converting the target content to YCbCr color domain. Basically, the transformation from RGB color signal to YCbCr signal is used to separate a luma signal Y and two chroma components Cb and Cr and mainly used for JPEG compression and color video signals. In proposed method, instead of using RGB color space directly, YCbCr color space is used to make use of human visual system characteristic. The conversion from RGB to YCbCr is calculated using the following equation:

$$Y = 0.299R + 0.587G + 0.114B \quad (5)$$

$$Cb = -0.169R - 0.322G + 0.500B \quad (6)$$

$$Cr = 0.500R - 0.419G - 0.081B \quad (7)$$

Then, training of neural network is processed. For the training, one must decide the structure of neural network. The amount of units for input layer is decided by the number of pixels selected from target content data. In proposed method, the feature values are diagonal coefficient values from frequency transformed selected feature subblocks. For better approximation, one bias neuron is added for input layer.

The neural network is trained to output a value of 1 or 0 as an output signal. In proposed method, one network represents one binary digit for corresponding secret codes. The adequate amount of neurons in the hidden layer, for backpropagation learning in general, is not known. So the number of neurons in hidden layer will be taken at will. In proposed method, ten hidden units are used. For better approximation, one bias neuron like is introduced for hidden layer as well. Once network weights are converged to certain values, proposed method use these values and the coordinates of selected feature subblocks as extraction keys. This extraction keys must be shared among the embedder and the extractor in order to extract proper hidden signals from the contents.

For extraction process, same neural network structures are constructed. This can be constructed by having the proper network weights. Only with the proper input values of the selected feature subblocks will output the corresponding hidden codes. And proper input values are induced only when you know the proper coordinates of the subblocks for the corresponding hidden signal. These embedding and extraction procedure are shown in diagram in Figure 2 and 3.

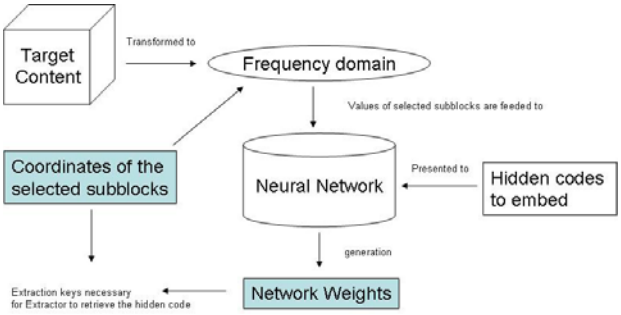

Figure 2 Embedding procedure

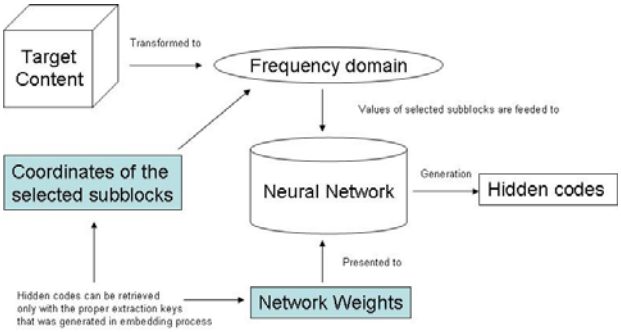

Figure 3 Extracting procedure

2.4 Necessary parameters

For embedding, there are two parameters to decide on. First is the number of classification patterns to embed. More the number of classification patterns, the more data can be embedded, but introducing large number of classification patterns will result in high calculation cost. Second parameter is the coordinate of the subblock which you associate the patterns to. Coordinates will determine the input values for the embedding and extraction process.

For extracting, there are two parameters to be shared between the embedder and extractor. Former is the neural network weights created in the embedding process. Latter is the coordinates of feature subblocks. Only with the presence of the proper network weights and the proper coordinates, are able to output the proper hidden signals as shown in Figure 2 and 3.

2.5 Embedding Procedure

Embedding process consists of following procedures:

1. Frequency transformation of target image
2. Selection of the feature subblocks
3. Backpropagation learning process
4. Save extraction information keys

First, frequency transformation of the image is performed. Same amount of unique subblocks as number of classification patterns must be chosen from the target content. Sufficient number of neural networks must be prepared, which will be the number of binary digits to satisfy the classification patterns. In case for 32 input patterns, five networks are enough to represent 32 different identification values because five binary digits are sufficient to distinguish for 32 patterns. Learning of all network is repeated until the output value satisfies a certain learning threshold value. After all network weights are converged, the coordinates of subblocks and the values of network weights are saved. Extractor will use this information to extract hidden codes in the extraction process.

2.6 Extraction Procedure

Extraction process consists of following procedures:

1. Frequency transformation of target image
2. Obtain necessary information from extraction keys
3. Generate an output signals from neural network

First procedure is an equivalent process as the first procedure in embedding process. Extractor must receive the coordinate from embedded user. Only by knowing the proper coordinates of feature subblocks will lead user to the proper input values. By knowing proper network weights, extractor can induce the structure of the network and only proper network weights are able to output the proper hidden codes. After constructing the neural network, extractor examines the output value from the network with the input values induced from the feature subblocks. This procedure is shown in Figure 4. Each network output either 1 or 0 with the aid of threshold for output unit.

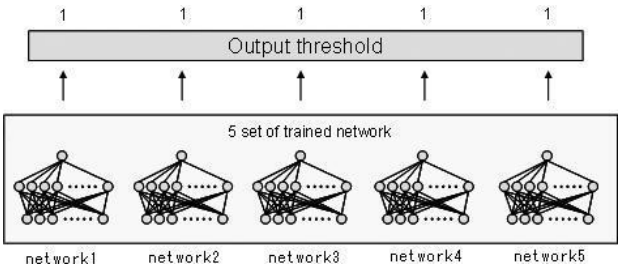

Figure 4 System structure for extraction (pattern 32)

3. Experimental conditions and results

In this experiment, we used TIFF format pepper image, which is 512*512 pixels in size, as target content data. Both original and high pass filtered pepper image is shown in Figure 5. We embedded 32 different identification patterns as hidden signals. That is, hidden signals are [00000] for pattern 1, [00001] for pattern 2, ... [11111] for pattern 32. Five neural networks are used for classification of 32 patterns. Each network output value representing the binary digits of hidden codes. In this experiment, network 1 represents the largest binary bit and network 5 represents the smallest binary bit. The numbers of hidden layer units were set to 11 including one bias unit. Learning process is repeated until the output values converge to a learning threshold of 0.1. Also, the output threshold in this experiment is set to 0.5. This means that if output value is larger than 0.5, output signal is set to 1, and if it is less than 0.5, output signal is set to 0.


Figure 5 Original pepper and high pass filtered pepper

With the threshold value of 0.5, proposed method was able to extract proper signals for proper patterns. For example, output signals for pattern 1 is [00000], output signals for pattern 32 is [11111] and etc. The output signals retrieved for high pass filtered image are shown to be slightly different compared to the output signals for the original image. But with the same output threshold of 0.5, we were able to retrieve same hidden bit codes for all 32 set of input patterns from high pass filtered image as well. This result showed robustness to high pass filtering alteration.

The results of this experiment, output signal values for both original image and high pass filtered image are shown in the following figures. Figure 6, 7, 8, 9 and 10 are the output signal values for each neural network.

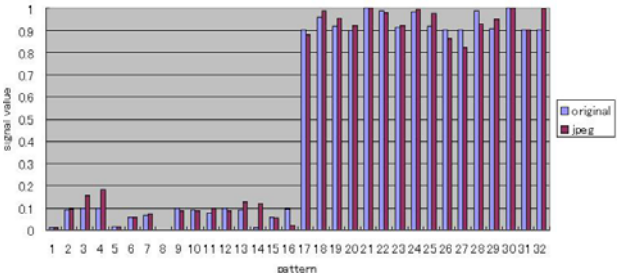

Figure 6 Signal values of network 1

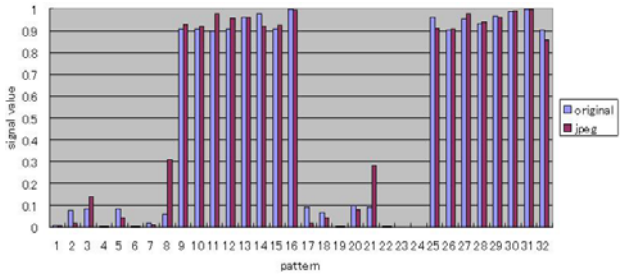

Figure 7 Signal values of network 2

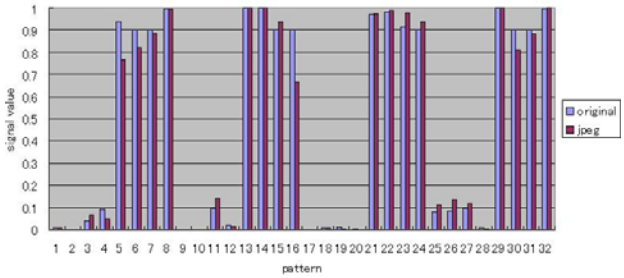

Figure 8 Signal values of network 3

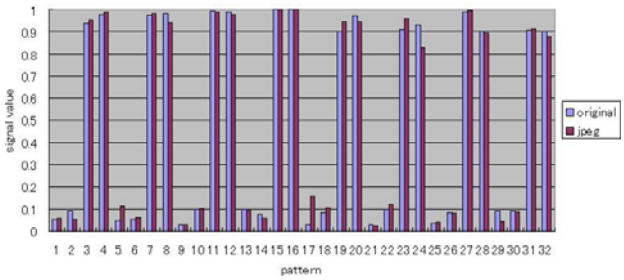

Figure 9 Signal values of network 4

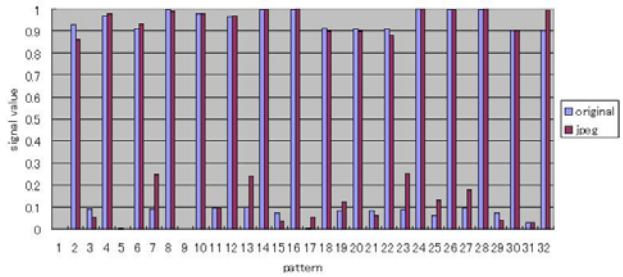

Figure 10 Signal values of network 5

4. Conclusion

In this paper, we proposed an information hiding technique without embedding any data into target content. This characteristic is effective when user must not damage the content but must conceal a secret code into target content.

Proposed method uses multi-layered neural network model for classifying the input patterns to corresponding hidden signals. For input values, we used DCT coefficients on YCbCr domain. Proposed method does not limit to DCT as frequency transformation method only, one can use DFT and DWT otherwise.

In the experiment, proposed method showed robustness to high pass filtering. Proposed method has both robust and fragile watermarking characteristics and can be conditioned with the parameter adjustments. Also, because propose method does not alter the target content, it is applicable to steganography. Meanwhile, because proposed method relies on the position of the feature subblocks, it is weak to geometric attacks like shrinking or rotation of the image. This must take into consideration for future work.

References

[1]H. Sasaki, editor. Intellectual Property Protection for Multimedia Information Technology. IGI Global, 12 2007.

[2] D. Kahn. The history of steganography. In Proceedings of the First International Workshop on Information Hiding, pages 1.5, London, UK, 1996. Springer-Verlag.

[3]S. Katzenbeisser and A. P. Fabien, editors. Information Hiding Techniques for Steganography and Digital Watermarking. Artech House Publishers, 1 2000.

[4] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker. Digital Watermarking and Steganography. Morgan Kaufmann, 2 edition, 11 2007.

[5] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for images, audio and video. Image Processing, 1996. Proceedings., International Conference on, 3, 1996.

[6]F. Rosenblatt. The perceptron a probabilistic model for information storage and organization. Brain Psych. Revue, 62:386.408, 1958.

[7]D. Rumelhart and J. McClelland. Parallel distributed processing: explorations in the microstructure of cognition, vol.1: foundations. MIT Press Cambridge, MA, USA, 1986.

[8] D. Artz. Digital steganography: hiding data within data. IEEE Internet Computing, 5(3):75.80, May/June 2001.

Kensuke Naoe graduated faculty of environment and information studies at Keio University in 2002. He received the Master degree at Graduate School of Media and Governance at Keio University in 2004. His major is artificial neural network and information security. Interested in the area of research in watermark using neural network, network intrusion detection, cryptography and malware detection.

Yoshiyasu Takefuji is a tenured professor on faculty of environmental information at Keio University since April 1992. He was an Editor of the Journal of Neural Network Computing, an associate editor of IEEE Trans. on Neural Networks, Neural / parallel / scientific computations, and Neurocomputing, and a guest editor of Journal Analog Integrated Circuits and Signal Processing in the special issue on analog VLSI neural networks and also guest editor of Neurocomputing in the special issue on neural network optimization.