Information Hiding by Machine Learning:
A Method of Key Generation for Information Extracting Using Neural Network

Kensuke Naoe, Keio University, Japan

Hideyasu Sasaki, Ritsumeikan University, Japan

Yoshiyasu Takefuji, Keio University, Japan

ABSTRACT

In this paper, the authors propose information hiding by machine learning: a method of key generation for information extracting using neural network. The method consists of three layers for information hiding. First, the proposed method prepares feature extraction keys, which are saved by feature extraction attributes like feature coordinates and the region of frequency coefficients. Second, the proposed method prepares hidden patterns in advance to the embedding procedure as a watermark signal of the target contents. Finally, the proposed method generates information extraction keys by using machine learning to output presented hidden patterns. The proper hidden patterns are generated with the proper information extraction key and feature extraction key. In the experiments, the authors show that the proposed method is robust to high pass filtering and JPEG compression. The proposed method contributes to secure visual information hiding without damaging any detailed data of the target content.

Keywords: *Digital Watermarking, Feature Extraction, Image Authentication, Information Hiding, Key Generation, Neural Network*

INTRODUCTION

In this paper, we propose a method of key generation scheme for static visual information hiding by using machine learning technology, neural network as its exemplary approach for machine learning method.

The proposed method is to hide bit patterns as training data set for machine learning and the generated classifier acts as an information extraction key. These hidden patterns are prepared in advance to embedding procedure for the purpose of fingerprint or watermark signal of the target content. Machine learning is processed so that the generated classifier herein is neural network, to output the presented hidden patterns when feature values are presented to the classifier. This feature values are extracted using feature extraction key which is prepared before the embedding process. Feature extraction key

DOI: 10.4018/joci.2011010102

contains information such as feature coordinates and regions of frequency coefficients. Also, the generated classifier in the machine learning process is saved as information extraction key for extraction process. Information extraction procedure is properly processed only by having two proper keys of feature extraction key and information hiding key.

Our method consists of three layers for digital watermarking as an application of our information hiding and information retrieval method: The first layer prepares feature extraction key, which is saved by feature extraction attributes including feature coordinates and regions of frequency coefficient of visual information; The second layer prepares watermark bit patterns to be related with the target content, in advance to the third embedding procedure, as watermark values of the visual information; And, the third layer generates watermark extraction key which is the classifier generated by the machine learning process. The discussed watermark extraction key and feature extraction key identify the related or associated hidden pattern which is the watermark values for proper digital watermarking procedure as shown in Figures 1 and 2.

The proposed method is to contribute to secure visual information hiding without losing any detailed data of visual objects. The proposed method has used neural network for its training approach not limited but open in its applications to other machine learning approaches including fuzzy, Bayesian network and others. In this paper, the target content is a static visual data which are constructed with discrete data set and we have demonstrated the feasibility of solving this problem by using neural network model. We would enhance our method by using those other approaches, such as fuzzy for dynamic visual data like video stream data and Bayesian network for continuous data structures. This paper is different from the previous work by Ando and Takefuji (2003) in terms of embedding size where this paper does not embed any information to the target content and also implies

Figure 1. Key generation scheme in the proposed method

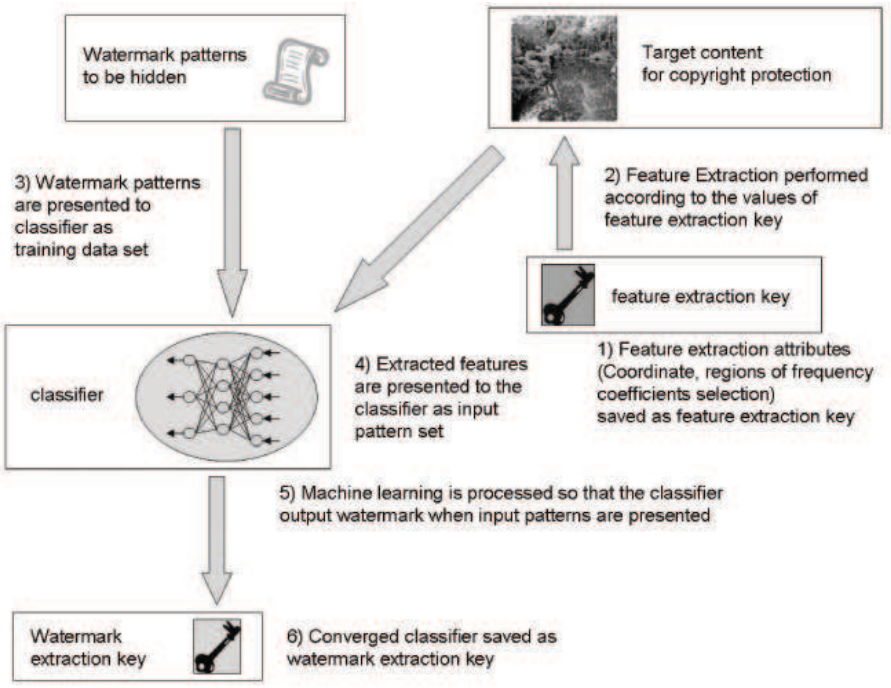
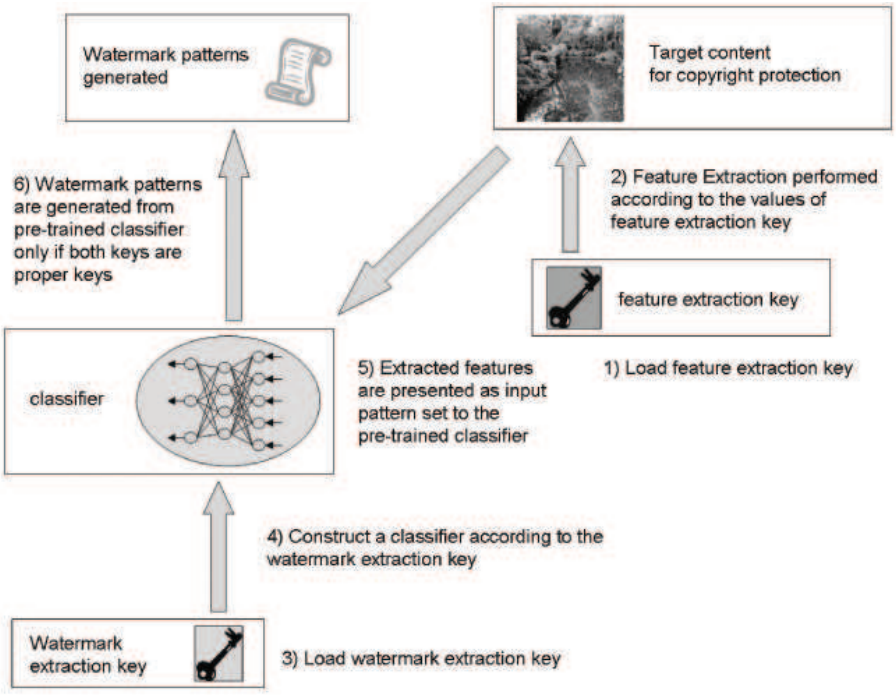


Figure 2. Watermark extraction scheme in the proposed method



that the machine learning algorithm is not limited only to the neural network model as proposed in our previous work (Naoe & Takefuji, 2008).

BACKGROUNDS AND RELATED WORKS

Here, we overview the backgrounds and related works in the research areas of information hiding and machine learning, neural network as an exemplary model.

Information Hiding Scheme

The emergence of the Internet with rapid progress of information technologies, digital contents are commonly seen in our daily life distributed through the network. Due to the characteristics of digital contents are easy to make an exact copy and to alter the content itself, illegal distribution and copying of digital contents has become main concerns for authors, publishers and legitimate owners of the contents (Sasaki, 2007). There are several ways to protect digital content. One can protect the content by cryptographic-based schemes (Rothe, 2002), but this avoids free distribution and circulation of the content through the network because the decryption key must be shared, which most of the time not desirable for authors of the contents.

Information hiding provides a reliable communication by embedding secret code into a content for the purposes of intellectual property protection, content authentication, fingerprinting, covert communications, content tracking, end-to-end privacy, secure distribution and etc (Hung et al., 2007; Wolf et al., 2007; Kwok et al., 2003). The researches in information hiding has a history (Kahn, 1996) and namely the researches in digital watermarking and steganography have been active (Katzenbeisser & Fabien, 2000). Both are very similar but their

applications are different (Reither & Rubin, 1998; Cox et al., 2007).

Digital watermarking became a key technology for protecting copyrights. Digital watermarking protects unauthorized change of the contents and assures legal user for its copyright. There are several ways to protect digital content. One example is to encrypt the content and to share decryption key between the author and the observer. But this method limits the use of the content because the observer must receive the decryption key to observe the content. This feature avoids a free distribution and circulation of the content through the network which most of the time, not desirable to the author of the content. While digital watermarking only embeds watermark information to the content and one can observe the content to anyone whom does not have the watermark extraction key, which do not avoid free distribution of the content.

Digital watermark is often embedded imperceptibly to human receptors to avoid contaminating the content. For imperceptible images, human visual system (HVS) model is often used. There are many still image watermark researches which make use of HVS model (Westen et al., 1996; Swanson et al., 1996; Delaigle et al., 1998; Kim et al., 2002). For imperceptible audio, psychoacoustic model is often used. Basic idea of psychoacoustic model is that human ears are insensitive to the change in phase and amplitude, but very sensitive to the change in the time domain. Also human has limitation to high frequency domain and low frequency domain (Cox et al., 2002). There are many researches for audio watermark and many of them use psychoacoustic model for implementations (Boney et al., 1996; Baumgarte et al., 1995; Wolfe & Godsill 2000; Gruhl et al., 1996).

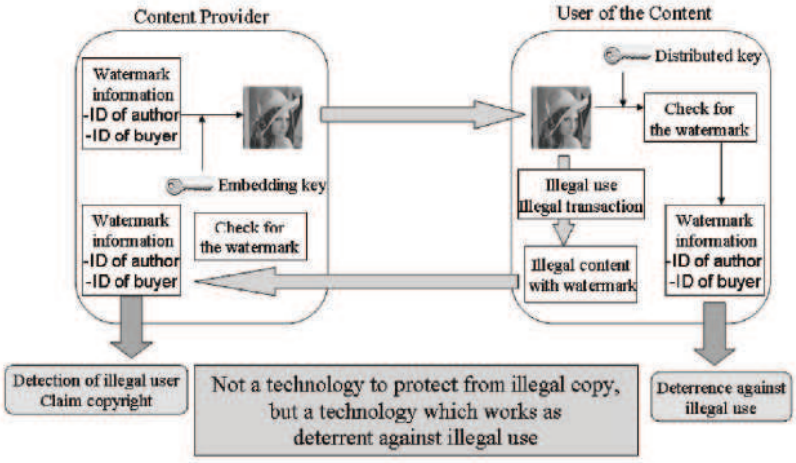
Most of imperceptible still image watermark method uses HVS model in order to embed data into frequency domains after the frequency transformation of multimedia content, such as DFT, DCT and DWT. Perceptible watermark are sometimes used, but it limits the use of the images. Therefore main concern in this research area has focused on imperceptible watermark. In general, digital watermark is a technique to conceal a code imperceptibly to the content to anybody who observes it, and is also difficult to remove the code from the content to protect its intellectual property rights. Figure 3 describes the basic model of digital watermarking.

Standard digital watermarking for copyright protection in still image prepares watermark information to be embedded in to the protecting image. This watermark information can be ID of the author, ID of buyer and copyright information of the image. This watermark information is then embedded into the content by using the watermark embedding key. User of the image will receive the extraction key in a proper way and use it to extract the digital watermark from the target content. The extracted content should contain the same proper watermark information as the embedded watermark information even through removal attacks. In case when the users illegally distribute this content to the network, the provider will examine if the distributed content contains the same watermark information. If they are the same watermark information, provider can detect the illegal user who distributed this content to the network and claim the copyright of the content. Digital watermarking does not protect from illegal copies of content, but it is more of a technology which works as deterrent against illegal use. Meanwhile, steganography conceals a hidden messages to a content but the existence of a message is kept secret (Artz, 2001).

For our proposed method to be used in digital watermarking, in general, the robustness and imperceptibility of digital watermarking take trade-off relationships. Embedding information must be placed in perceptually significant signal for it to be robust against removal attacks, but it is known that modifying these regions will lead to perceptual degradation of signal. Therefore, an ideal digital watermarking algorithm should have minimal amount of embedding information.

The robustness and imperceptibility of information hiding method take trade-off rela-

Figure 3. Basic model of digital watermarking



tionships. The amount of additional information being added to the target content can be calculated by using PSNR value. Obviously, many of the information algorithms damages target content because they must embed information to target content, which causes PSNR values to be declined. Whereas, our proposed method does not alter the value of PSNR since it does not embed any information to the target content. This is the major contribution of our proposed method compared to the conventional information hiding algorithms and yet possesses robustness against minor graphical alterations.

There are many digital information hiding algorithms been proposed in the past, 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 information hiding algorithm or controls the file integrity using hash functions. If one wishes to strengthen the robustness of information hiding algorithm, one must examine and assure that applying the algorithm will not affect the hidden embedded signals in advance. Furthermore, applying another information algorithm will alter the fingerprint of the content managed by hash functions and forces administrator to recalculate a new hash values after applying new information hiding algorithm, which most of the time, result in higher calculation cost and time.

Because our proposed method does not affect the target content at all, one can apply new information hiding scheme seamlessly without altering the fingerprint of the target content using the proposed method.

There is also a difficulty to fulfill the non-repudiation requirement by using a robust watermarking scheme alone. To address this problem, there must be a solid distribution protocol and secure infrastructure framework to put in practice. A contribution from previous work (Cheung et al., 2008) which proposed a distribution protocol and secure system framework addressed this problem and its watermarking algorithm can be replaced with another algorithms as long as the watermark can be inserted in the encrypted domain where digital contents are encrypted by public keys. Our proposed method generates pair of extraction keys where one of them can act as public key and other as private key and its algorithm also does not alter the target content. Therefore, our proposed method

has an ability to collaborate with this previous work. This public key characteristic is stressed because watermarking algorithms based on a secret key present a major drawback: they do not allow a public recovery of the watermark. In order to overcome this limitation, public key watermarking algorithms have been proposed; such systems consist of two keys: a public and a private one. An image can be watermarked using the private key, whereas the public key is used to verify the mark.

Machine Learning Approaches

Machine learning and statistical analysis are very effective approach to approximate unknown class separating functions and to find potentially useful patterns in the data set (Bishop, 2006). Basically, machine learning algorithm uses training set to adjust the parameter of the model adaptively. When target vector or teacher patterns are presented for an training set, training or learning process is performed to condition the model to output a proper patterns which is same or close to as the target vector. Some of the noticeable machine learning models are Bayesian network (Bernardo & Smith, 2001), fuzzy logic (Klir & Yuan, 1995), support vector machine (Cortes & Vapnik, 1995), and neural network (Kohonen, 1995; Bishop, 1995). The proposed method uses machine learning approach to generate information extraction keys and herein uses neural network model in this paper.

The basic principle of neural network is that neuron, or most atomic unit of neural network, only has a simple function of input and output signal but is capable of complex function when these neurons are organically connected forming a network. Mathematical models of these networks are called neural networks, and processing these artificial neural networks on computers is called neural computing. Neural computing is a classic research topic and neural networks are known to be capable of solving various kinds of problems by changing the characteristics of neuron, synaptic linking and structure of the network.

The 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 (Rosenblatt, 1958). Multilayered perceptron with more than three layers are known to have an approximation ability of a nonlinear function if properly trained, but then there was no ideal learning method for this kind of training. This model became popular when a training method called back propagation learning was introduced (Rumelhart & McClelland, 1986). Other popular neural network models are RBF network model (Broomhead & Lowe, 1988), pulsed neural network model (Eckhorn et al., 1990; Johnson, 1994), chaotic neural network model (Aihara, 1990), Elman network model (Elman, 1990) and many others.

METHODOLOGY

Overview of the Proposed Method

In this section, we explain how our proposed method generates extraction keys from the target content and how to retrieve hidden information from the target content using the extraction keys generated in the information hiding procedure. With our method, the use of machine learning approach is the key methodology. Here, adjustment of neural network weights to output desired hidden information pattern by supervised learning of the neural network is performed. This conditioned neural network works as a classifier or information extraction key to recognize a hidden information pattern from the content. Therefore, extractor uses this neural network weights as extraction keys for extracting the hidden information patterns. Extractor must have proper visual features used for generation of extraction key and proper network weights of neural network which is the extraction key in our proposed method.

Here, we explain the procedures for generation of extraction keys and extraction of hidden patterns. Generation of extraction key is included in the embedding process and

extraction of hidden information patterns is included in the extraction process. First, we simply demonstrate the procedures which must be taken for embedding process and extraction process.

Embedding process consists of following procedures:

1. Frequency transformation of target image
2. Selection of the feature values according to the feature extraction key
3. Prepare bit patterns to be hidden
4. Generation of information extraction key to output hidden patterns by back propagation learning of neural network using feature values as input patterns
5. Save generated classifier as information extraction key

First, frequency transformation of the image is performed. There are several methods to transform an image to frequency domain, such as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). In our method, DCT is used to be robust against some image compression. Compression using DCT is known to be very effective, and is employed in MPEG format and JPEG format. DCT divides an image into N*N pixel small blocks, here we call sub-blocks. General DCT generates 8*8 pixel size sub-blocks. We must select certain amount of sub-blocks from frequency domain of target image and DCT coefficients are chosen diagonally from those sub-blocks. The same amount of unique sub-blocks must be chosen from the target image as number of classification patterns, which in this method is the hidden information pattern. Sufficient number of neural networks must be prepared, which will be the number of binary digits to satisfy the hidden information patterns. In case for choosing 32 hidden information patterns, five networks are enough to represent 32 different classification values because five binary digits are sufficient to distinguish for 32 patterns. Learning of all networks is repeated until the output value of neural network satisfies a certain learning threshold value. After all network weights are converged, the coordinates of sub-blocks and the values of network weights are saved. Extractor will use this information to extract hidden codes in the extraction process.

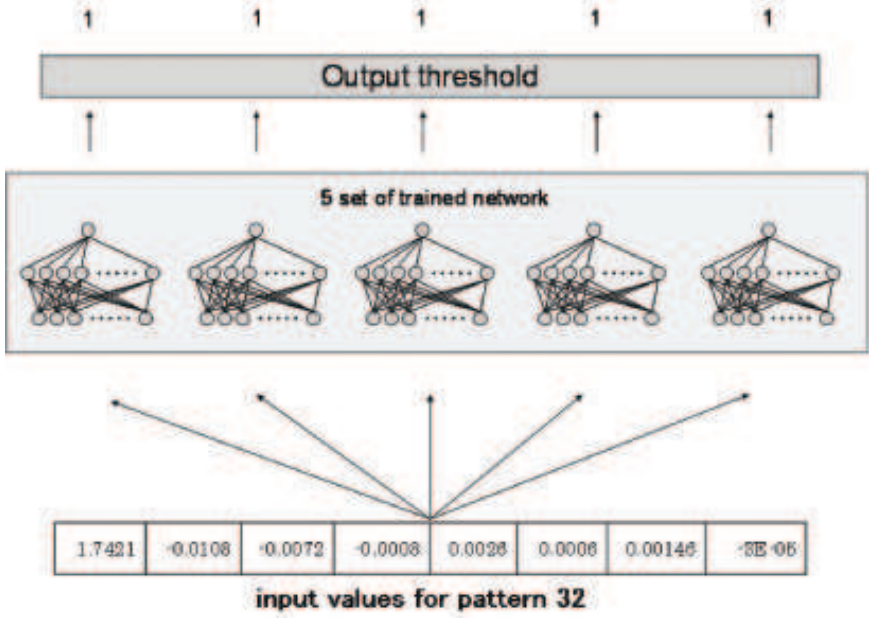
Extraction process consists of following procedures:

1. Obtain feature extraction key and information extraction key
2. Feature extraction of target image using the feature extraction key
3. Construct a classifier using information extraction key
4. Observe the output patterns from classifier

Extractor must receive both proper feature extraction key and information extraction key to obtain proper hidden pattern. Only by having the proper feature extraction key will lead the user to the proper input patterns to the neural network. By knowing proper information extraction key, extractor can induce the structure of the neural network and only proper neural network is able to output the proper hidden information patterns. After constructing the neural network, extractor examines the output value from the network with the input values induced from the feature extraction key. This procedure is shown in Figure 4. Each network output either 1 or 0 with the aid of threshold for output unit.

Furthermore, here we explain each procedure more concretely. For key generation procedure, frequency transformation of the target content is processed for visual feature extraction of target content. Frequency transformation can be anything like DCT, DFT or DWT, here we use DCT because of its simplicity and its structure of feature vectors of a transformed image in frequency domain. 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 our proposed method, instead of using RGB color space di-

Figure 4. System structure for watermark extraction



rectly, 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:

$$Y = 0.299R + 0.587G + 0.114B$$
$$Cb = -0.169R - 0.322G + 0.500B$$
$$Cr = 0.500 - 0.419G - 0.081B$$

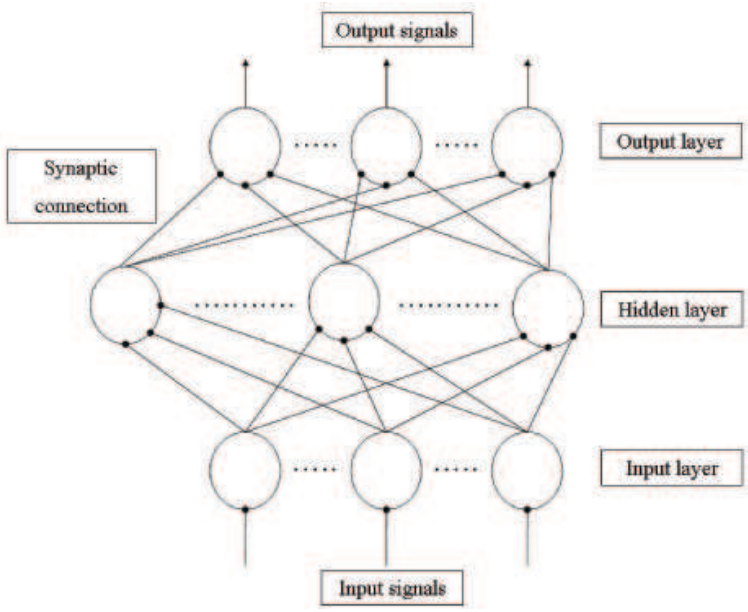
Then, training of neural network is processed. For the training, one must decide the structure of neural network. The amount of units for input layer is decided by the number of pixels selected from target content data. In our proposed method, the feature values are diagonal coefficient values from frequency transformed selected feature sub-blocks. 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 our proposed method, one network represents one binary digit for corresponding hidden information patterns. The adequate amount of neurons in the hidden layer, for back propagation learning in general, is not known. So the number of neurons in hidden layer will be taken at will. In our proposed method, ten hidden units are used. For better approximation, one bias neuron is introduced for hidden layer as well. Once network weights are converged to certain values, the proposed method use these values and the coordinates of selected feature sub-blocks as feature extraction key and information extraction key. These keys must be shared among the hider and the extractor in order to extract proper hidden information patterns from the contents.

Now, we demonstrate an overview of multilayered perceptron model. In multilayered perceptron model, 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 5.

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 as shown in Figure 6 is used for this model and this function is expressed as follows:

Figure 5. Multi-layered perceptron model



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

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 values for the unit are determined by the network weight and the input signal from the former layer. Consequently, to change the output value to a desired value for a certain input value patterns, the adjustment of these network weights must be conducted and this process is called learning. In our proposed method, we use back propagation learning as the learning method.

Back propagation learning is the process of adjusting the network weights to output a value close to the values of the teacher signal values which are presented to the neural network. Back propagation learning is a supervised learning. This method tries to lower the difference between the presented teacher signal and the output signal dispatched for certain input value patterns by adjusting the network weight. The difference between the teacher signal values and the actual output signals are called as error and often expressed as δ. The error will propagate backward to the lower layer and network weights are adjusted using these values. When teacher signal t_k is given to the unit k of output layer, the error δ_k will be calculated by following function:

$$\delta_k = (t_k - O_k) \cdot f'(O_k)$$

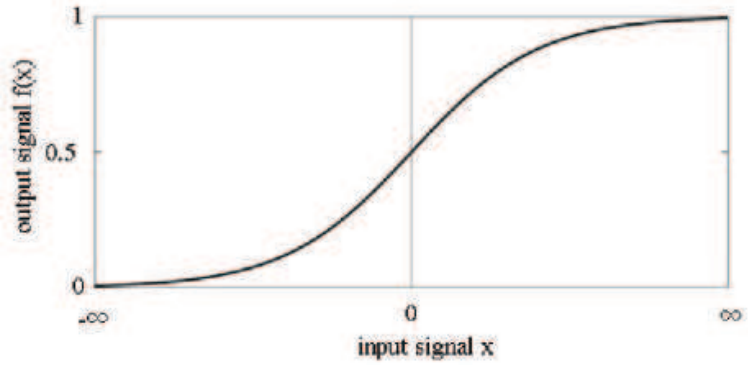
To calculate the error value δ_j for hidden unit, error value δ_k of the output unit is used. The function to calculate the error value δ_j for hidden unit j is as follows:

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

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:

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

Figure 6. Graph of sigmoid function



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. The lower the learning rate is the more gradual the learning process will be, and the bigger the learning rate is the more acute the learning process will be. Sometimes, this parameter must be tuned for stable learning.

For extraction process, same neural network structures are constructed using the converged information extraction key. Only with the proper feature extraction key and information extraction key will be able to output the corresponding information patterns. These embedding and extraction procedure are shown in diagram in Figures 7 and 8.

For embedding, there are two parameters to decide on. First is the number of information patterns to embed. More the number of information patterns to be hidden, the more data can be embedded, but introducing large number of information patterns will result in high calculation cost. Second parameter is the coordinate of the feature values, herein is the coordinate of the sub-block. Coordinates will determine the input patterns to the neural network for the embedding and extraction process.

For extracting, there are two keys to be shared, that is the information extraction key and feature extraction key, between the embedding and extracting users. Former is the neural network weights created in the embedding process. Latter is the coordinates of feature sub-blocks. Only with the presence of the proper keys are able to generate proper information patterns as shown in Figures 7 and 8.

EXPERIMENTS

Experiment 1

In this experiments, we examine the feasibility of our proposed method for information hiding in the two cases of images, from original images and altered images by high pass filtering and JPEG compression. Also, we have used two different types of images, one with asymmetrical and the other with symmetrical characteristic in vertical structure. In this experiment, we used TIFF format Lenna and Baboon images, which are 512*512 pixels in size, as target content data. Original and high pass filtered Lenna and Baboon images are shown in Figure 9 and Figure 10.

We have associated 32 different patterns as hidden signals with respect to the feature input values from the original image. That is, hidden signals are [00000] for pattern 1, [00001] for pattern 2, ... [11111] for pattern 32 as shown in Figure 11. Five neural networks are used for classification of 32 patterns. Each network output value is representing the binary digits of hidden patterns. In this experiment, network 1 represents the largest binary bit and network

Figure 7. Embedding procedure

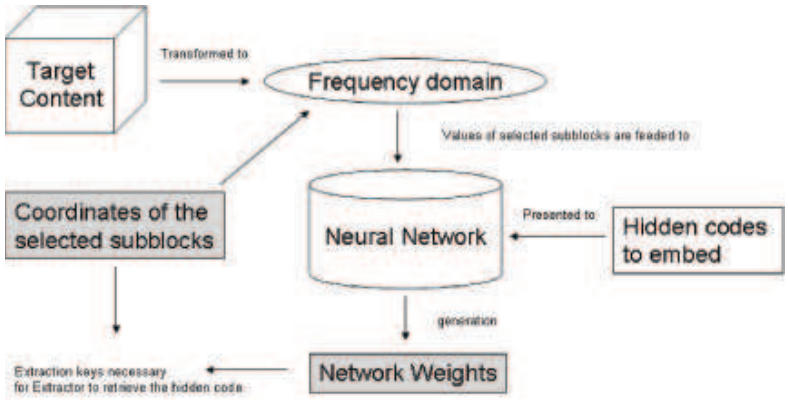
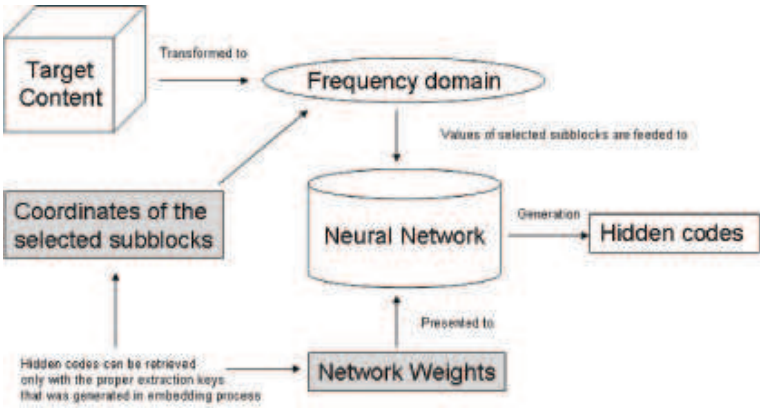


Figure 8. Extracting procedure



5 represents the smallest binary bit. The number of hidden layer units was set to 11 including one bias unit. Learning process is repeated until the output values converge to a learning threshold of 0.1.

The criteria for evaluating our experimental results is to see if the observed signal with threshold set to 0.5 is generating proper discrete signal of 1 or 0. This means that if output value is larger than 0.5, the output signal is set to 1, and if it is less than 0.5, output signal is set to 0. For example, output signals in binary bit pattern for pattern 1 is [00000], output signals for pattern 11 is [01010] and etc. This means that, with the threshold of 0.5, the expected discrete output values should be same as Figure 11, if the proposed method is robust against the alteration.

The results of this experiment, output signals for original image, high pass filtered image and JPEG compressed image of Lena are shown in Figure 15, 16, 17, 18 and 19 for each neural network (Figures 12-14). For output signals for original image, high pass filtered image and for JPEG compressed image of Baboon are shown in Figure 23, 24, 25, 26 and 27 for each neural network (Figures 20-22).

The output signals retrieved from high pass filtered image are shown to be slightly different compared to the output signals for the original

Figure 9. Original image and filtered image [Lenna]

image and also the output signals retrieved from JPEG compressed image is damaged more than of filtered image, but with the same output threshold of 0.5, we were able to retrieve same hidden information patterns for all 32 set from high pass filtered image and JPEG compressed image. These results showed the robustness to high pass filtering and JPEG compression alteration for both two different static visual images.

Experiment 2

In this experiment, we have tested to see the output results from those neural networks which were generated by using the Lenna image as shown in previous experiment and to apply them to various images which are perceptually distinguishable that those images are not variants of Lenna image. For this experiment, we have used the famous test images from the USC-SIPI Image Database like Tiffany (Figure 28), Airplane (Figure 29), House (Figure 30) and Splash (Figure 31). Same conditions which were used in previous experiment such as the coordinates of the selected feature sub blocks and parameters for learning process of neural networks are used for this experiment as well. Hypothetically, we should observe very different output values from other pictures compared to the output values generated by original Lenna image.

Experimental result for the four images mentioned above is shown in Figure 32, 33, 34, 35 and 36. As you can see the results for all the images tend to output a random noisy output signals which do not resemble any of the proper watermark patterns from the original image. Therefore, we can say that the proper output signal comes only from the original images or slightly modified images.

Experiment 3

In our proposed method, it will re-allocate the coordinates of the feature sub-blocks if the learning procedure fails to output the proper output pattern. However, this rate will be altered by the characteristic of the target image and the randomly chosen coordinates. Hence, we have tested to see the quantitative performance of our proposed method by examine the successful rate of our learning procedure in this experiment.

We have used Lenna, Baboon, Tiffany, Airplane, House and Splash images which were used in the previous experiments. We

Figure 10. Original image and filtered image [Baboon]

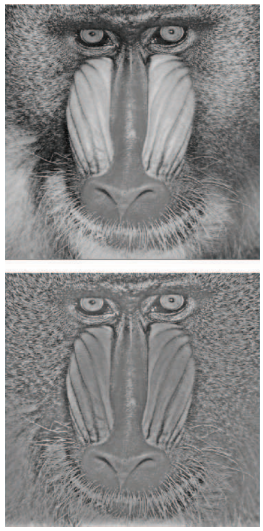


Figure 11. Teacher signal for 32 patterns in 5 networks

		pattern															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
network	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	4	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

		pattern															
		17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
network	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	4	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

have tried 20 different set of randomly chosen feature sub-blocks and examined the successful rate of the learning procedure for each images. The learning procedure will be repeated up to 500,000 times and stops before this count if it succeeded to output the proper pattern. We have used the same conditions like the network model and number of patterns as previous experiments. Therefore, there are 32 patterns in each set which means we have conducted experiment to see 640 different patterns in 20 set. The experimental results are shown in Figure 37.

In this experiment, there are 32 patterns in a set and total time for the procedure to be completed is a total time to complete learning procedure in all 20 set. From Figure 37, we can see that the most of the patterns are successfully completed its learning procedure. However, the proposed method considers that the learning process has failed to converge if even one pattern is failed in a set. It is obvious that the image with higher successful rate in a set has a faster computational time to complete its learning procedure. This difference is observed

Figure 12. Result for original lenna

	network1	network2	network3	network4	network5	result
pattern1	0.0521	0.0009	0.0937	0.0012	0.0004	00000
pattern2	0.0027	0.0405	0.0216	0.0760	0.9096	00001
pattern3	0.0923	0.0054	0.0861	0.9962	0.0876	00010
pattern4	0.0946	0.0896	0.0000	0.9746	0.9480	00011
pattern5	0.0206	0.0870	0.9962	0.0874	0.0827	00100
pattern6	0.0982	0.0521	0.9518	0.0653	0.9783	00101
pattern7	0.0026	0.0009	0.9230	0.9738	0.0010	00110
pattern8	0.0016	0.0006	0.9883	0.9935	0.9085	00111
pattern9	0.0169	0.9942	0.0005	0.0140	0.0528	01000
pattern10	0.0059	0.9156	0.0843	0.0019	0.9194	01001
pattern11	0.0828	1.0000	0.0011	0.9713	0.0604	01010
pattern12	0.0113	0.9084	0.0710	0.9788	0.9078	01011
pattern13	0.0544	0.9941	0.9116	0.0035	0.0900	01100
pattern14	0.0010	1.0000	0.9042	0.0937	0.9103	01101
pattern15	0.0883	0.9574	0.9835	0.9860	0.0987	01110
pattern16	0.0844	0.9965	0.9097	1.0000	0.9099	01111
pattern17	0.9965	0.0361	0.0451	0.0817	0.0852	10000
pattern18	0.9990	0.0875	0.0000	0.0814	0.9127	10001
pattern19	0.9903	0.0221	0.0940	0.9564	0.0858	10010
pattern20	0.9020	0.0966	0.0001	0.9982	0.9721	10011
pattern21	0.9998	0.0000	0.9999	0.0001	0.0878	10100
pattern22	0.9430	0.0907	0.9043	0.0986	0.9550	10101
pattern23	0.9742	0.0263	0.9209	0.9210	0.0035	10110
pattern24	0.9022	0.0001	0.9399	0.9067	0.9863	10111
pattern25	0.9243	1.0000	0.0000	0.0167	0.0303	11000
pattern26	0.9101	0.9999	0.0890	0.0807	0.9334	11001
pattern27	0.9044	0.9652	0.0013	0.9961	0.0899	11010
pattern28	0.9016	0.9022	0.0980	0.9090	0.9031	11011
pattern29	0.9962	0.9472	0.9759	0.0750	0.0127	11100
pattern30	0.9986	0.9221	0.9400	0.0793	0.9459	11101
pattern31	0.9292	0.9998	0.9096	0.9902	0.0388	11110
pattern32	0.9016	0.9174	0.9212	0.9918	0.9964	11111

because the failed patterns tries the learning process up to 500,000 times and then give up its procedure as failure. If there is a way to determine that the given coordinates of the feature sub-blocks are inappropriate for the learning procedure, we can reduce the computational time for the failed set and retry the learning procedure for the pattern with different coordinate being allocated.

The proposed method will re-allocate the coordinate and repeats the same learning procedure, when the failed set is found. Therefore, the failure rate per set is important. From this experiment, we expect an error rate of about 11.67 percent or 14 failed set out of 120 set. But one can also only consider the trained patterns to be the hidden signal patterns to be associated if the user wishes to because one can choose to use only several patterns from all the trained patterns to be the signal patterns. In this case, the error rate is about 0.49 percent or 19 failed patterns out of 3,840 patterns.

Experiment 4

Firstly, we define our scheme to calculate the difference of original image I_O and the compared image I_C by using the output signal patterns. We will compare the output patterns of each sub-block S_i for I_O and I_C in all of the networks N_j. We calculate the summation of these differences in each network for all patterns and divide by total number of patterns, which is the number of trained patterns P for each network multiplied by total number of the networks N used in the

Figure 13. Result for jpeg compressed lenna

	network1	network2	network3	network4	network5	result
pattern1	0.0094	0.0061	0.0054	0.0575	0.0002	00000
pattern2	0.0911	0.0154	0.0008	0.0520	0.8642	00001
pattern3	0.1538	0.1400	0.0659	0.9528	0.0517	00010
pattern4	0.1811	0.0041	0.0452	0.9913	0.9782	00011
pattern5	0.0115	0.0422	0.7684	0.1114	0.0014	00100
pattern6	0.0600	0.0029	0.8248	0.0605	0.9326	00101
pattern7	0.0723	0.0089	0.8877	0.9842	0.2483	00110
pattern8	0.0014	0.3073	0.9978	0.9403	0.9917	00111
pattern9	0.0872	0.9283	0.0000	0.0283	0.0000	01000
pattern10	0.0872	0.9185	0.0001	0.1014	0.9775	01001
pattern11	0.0984	0.9764	0.1430	0.9919	0.0933	01010
pattern12	0.0876	0.9571	0.0124	0.9787	0.9687	01011
pattern13	0.1250	0.9635	0.9993	0.0914	0.2402	01100
pattern14	0.1203	0.9210	0.9999	0.0590	0.9972	01101
pattern15	0.0512	0.9243	0.9409	1.0000	0.0345	01110
pattern16	0.0216	0.9924	0.6700	0.9999	0.9987	01111
pattern17	0.8828	0.0164	0.0000	0.1571	0.0525	10000
pattern18	0.9871	0.0451	0.0076	0.1061	0.8978	10001
pattern19	0.9529	0.0029	0.0034	0.9468	0.1244	10010
pattern20	0.9188	0.0825	0.0015	0.9459	0.8989	10011
pattern21	0.9999	0.2822	0.9766	0.0252	0.0626	10100
pattern22	0.9804	0.0032	0.9906	0.1221	0.8801	10101
pattern23	0.9193	0.0013	0.9783	0.9625	0.2535	10110
pattern24	0.9940	0.0004	0.9381	0.8292	0.9996	10111
pattern25	0.9743	0.9110	0.1124	0.0412	0.1298	11000
pattern26	0.8602	0.9068	0.1350	0.0802	0.9969	11001
pattern27	0.8254	0.9740	0.1148	0.9970	0.1781	11010
pattern28	0.9281	0.9422	0.0023	0.8962	0.9994	11011
pattern29	0.9510	0.9585	1.0000	0.0440	0.0398	11100
pattern30	0.9997	0.9904	0.8110	0.0861	0.9021	11101
pattern31	0.9002	0.9978	0.8865	0.9128	0.0272	11110
pattern32	0.9982	0.8591	0.9994	0.8809	0.9933	11111

learning procedure. This calculation can be done by the following equation:

$$c = \frac{\sum_{i=1}^{P}\sum_{j=1}^{N} abs(I_{O_{S_i N_j}} - I_{C_{S_i N_j}})}{P \times N}$$

Since our proposed method uses random positions for the selected sub-blocks, we should repeat this calculation for certain trial and calculate the average, so that we can have the statistical average of the difference between I_O and I_C in our proposed method. This calculation can be done by the following equation:

$$Average(C) = \frac{\sum_{i=1}^{T} C_i}{T}$$

In this experiment, we will examine the feasibility of our scheme to calculate the difference and similarity between the original image and the compared image which we mentioned above. Scheme 1 is our proposed scheme for calculating the average difference of the output values for all patterns in all networks by functions described above. Scheme 2 is a simple matrix norm of 32 by 5 dimensional output signal pattern matrix. In this experiment, we have used Stirmark (Petitcolas et al., 1998) which is well-known benchmark software for testing the robustness of digital watermarking that generates various modified images (Figure 38).

When we look at the results of this experiment more in detail, the results with scheme 1 can be fairly said to satisfy our requirement.

Figure 14. Result for high pass filtered lenna

	network1	network2	network3	network4	network5	result
pattern1	0.0897	0.0009	0.0934	0.0014	0.0003	00000
pattern2	0.0027	0.0412	0.0226	0.0737	0.9061	00001
pattern3	0.0918	0.0048	0.0876	0.9963	0.0878	00010
pattern4	0.1001	0.0931	0.0000	0.9614	0.9496	00011
pattern5	0.0410	0.0562	0.9987	0.1514	0.1211	00100
pattern6	0.0254	0.0920	0.8345	0.0777	0.9765	00101
pattern7	0.0026	0.0009	0.9231	0.9738	0.0010	00110
pattern8	0.0015	0.0006	0.9897	0.9867	0.8813	00111
pattern9	0.0177	0.9922	0.0006	0.0154	0.0483	01000
pattern10	0.0059	0.9198	0.1334	0.0021	0.9165	01001
pattern11	0.0629	1.0000	0.0008	0.9952	0.0610	01010
pattern12	0.0113	0.9089	0.0678	0.9768	0.8978	01011
pattern13	0.0430	0.9965	0.7230	0.0033	0.0937	01100
pattern14	0.0009	1.0000	0.9035	0.0936	0.9105	01101
pattern15	0.0626	0.9657	0.9883	0.9760	0.0514	01110
pattern16	0.0977	0.9965	0.8929	1.0000	0.9095	01111
pattern17	0.9974	0.0763	0.0302	0.1740	0.1607	10000
pattern18	0.9985	0.3235	0.0000	0.1197	0.9114	10001
pattern19	0.9983	0.0044	0.0795	0.9237	0.0708	10010
pattern20	0.9114	0.1042	0.0004	0.9963	0.9712	10011
pattern21	0.9998	0.0000	0.9999	0.0001	0.1437	10100
pattern22	0.9377	0.1115	0.8899	0.1339	0.9561	10101
pattern23	0.9692	0.0320	0.9024	0.9205	0.0040	10110
pattern24	0.8248	0.0002	0.9568	0.9600	0.9858	10111
pattern25	0.9292	1.0000	0.0000	0.0340	0.0401	11000
pattern26	0.9101	0.9999	0.0891	0.0807	0.9334	11001
pattern27	0.9052	0.9651	0.0013	0.9961	0.0904	11010
pattern28	0.9036	0.8765	0.1275	0.8928	0.9059	11011
pattern29	0.9960	0.9458	0.9773	0.0643	0.0128	11100
pattern30	0.9986	0.9225	0.9405	0.0794	0.9462	11101
pattern31	0.9200	0.9999	0.7143	0.9935	0.0233	11110
pattern32	0.9382	0.7912	0.8857	0.9850	0.9969	11111

Figure 15. Observed signal values for network 1 [Lenna]

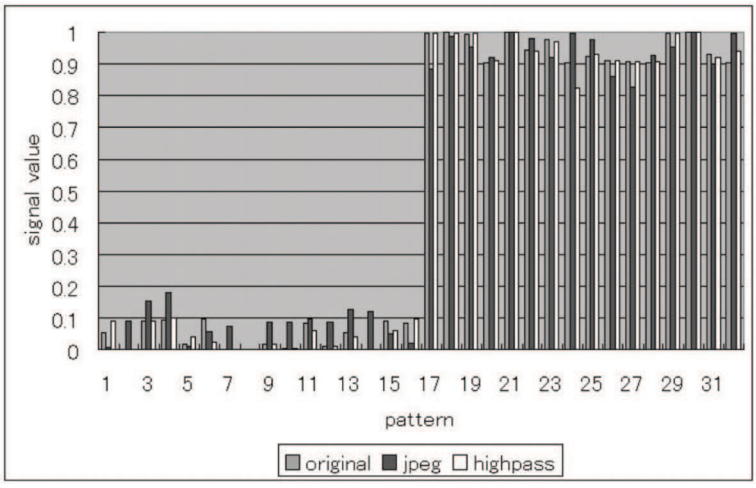


Figure 16. Observed signal values for network 2 [Lenna]

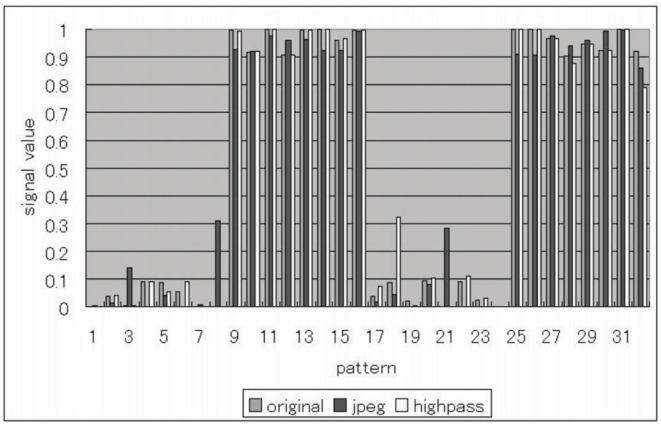


Figure 17. Observed signal values for network 3 [Lenna]

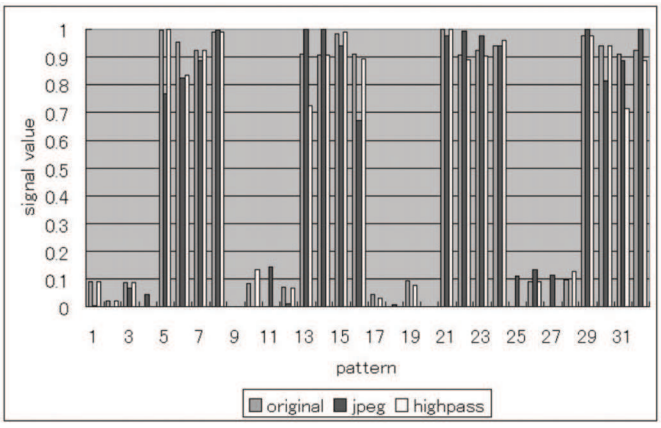


Figure 18. Observed signal values for network 4 [Lenna]

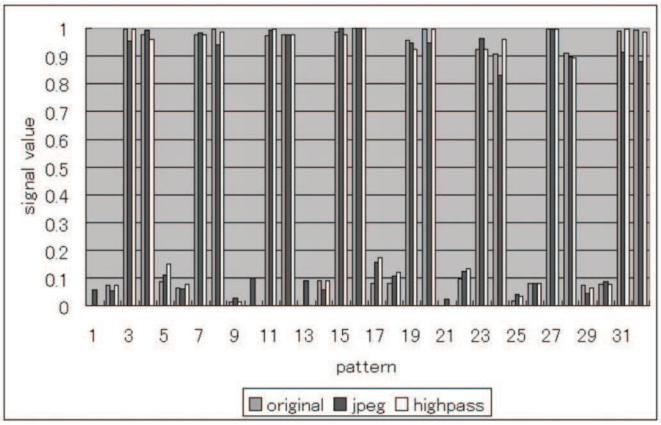


Figure 19. Observed signal values for network 5 [Lenna]

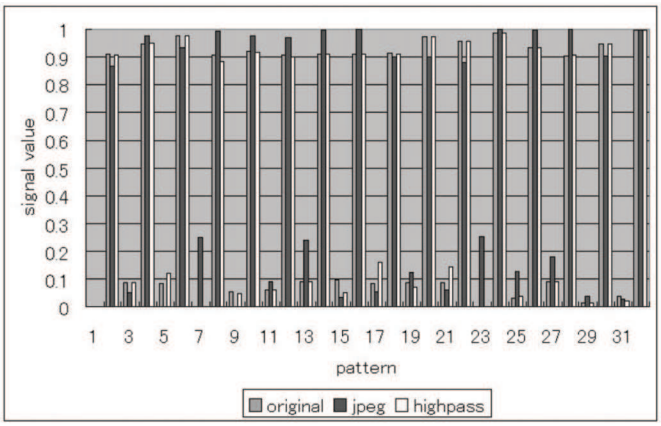


Figure 20. Result for original baboon

	network1	network2	network3	network4	network5	result
pattern1	0.0137	0.0990	0.0984	0.0958	0.0953	00000
pattern2	0.0036	0.0612	0.0003	0.0943	0.9112	00001
pattern3	0.0962	0.0002	0.0960	0.9086	0.0955	00010
pattern4	0.0068	0.0942	0.0551	0.9026	0.9002	00011
pattern5	0.0923	0.0676	0.9152	0.0708	0.0014	00100
pattern6	0.0494	0.0993	0.9632	0.0944	0.9103	00101
pattern7	0.0113	0.0003	0.9954	0.9195	0.0169	00110
pattern8	0.0872	0.0986	0.9094	0.9038	0.9018	00111
pattern9	0.0023	0.9393	0.0984	0.0889	0.0006	01000
pattern10	0.0963	0.9866	0.0150	0.0991	0.9881	01001
pattern11	0.0000	0.9981	0.0922	0.9752	0.0015	01010
pattern12	0.0140	0.9803	0.0094	0.9030	0.9940	01011
pattern13	0.0070	0.9282	0.9077	0.0999	0.0955	01100
pattern14	0.0131	0.9184	0.9307	0.0014	0.9999	01101
pattern15	0.0019	0.9036	0.9756	0.9653	0.0000	01110
pattern16	0.0954	0.9895	0.9827	0.9277	0.9110	01111
pattern17	0.9988	0.0941	0.0105	0.0964	0.0038	10000
pattern18	0.9976	0.0080	0.0861	0.0159	0.9094	10001
pattern19	0.9900	0.0396	0.0258	0.9122	0.0974	10010
pattern20	0.9576	0.0990	0.0008	0.9987	0.9460	10011
pattern21	0.9011	0.0828	0.9104	0.0951	0.0905	10100
pattern22	0.9697	0.0821	0.9998	0.0865	0.9084	10101
pattern23	0.9371	0.0944	0.9824	0.9241	0.0557	10110
pattern24	0.9030	0.0502	0.9103	0.9689	0.9085	10111
pattern25	0.9078	0.9522	0.0312	0.0575	0.0177	11000
pattern26	0.9734	0.9919	0.0973	0.0332	0.9031	11001
pattern27	0.9603	0.9551	0.0930	0.9646	0.0434	11010
pattern28	0.9088	0.9724	0.0246	0.9077	0.9047	11011
pattern29	0.9706	0.9232	0.9850	0.0700	0.0871	11100
pattern30	0.9094	0.9025	0.9350	0.0251	0.9835	11101
pattern31	0.9842	0.9409	0.9025	0.9994	0.0933	11110
pattern32	0.9179	0.9116	0.9038	0.9665	0.9976	11111

Figure 21. Result for jpeg compressed baboon

	network1	network2	network3	network4	network5	result
pattern1	0.0133	0.0984	0.1020	0.0970	0.0945	00000
pattern2	0.0036	0.0622	0.0003	0.0910	0.9147	00001
pattern3	0.0962	0.0002	0.0960	0.9086	0.0955	00010
pattern4	0.0077	0.1105	0.0559	0.9005	0.9166	00011
pattern5	0.0928	0.0672	0.9159	0.0695	0.0014	00100
pattern6	0.0497	0.0997	0.9632	0.0943	0.9105	00101
pattern7	0.0113	0.0003	0.9954	0.9200	0.0167	00110
pattern8	0.0663	0.0960	0.9219	0.9020	0.8855	00111
pattern9	0.0019	0.9531	0.1005	0.1016	0.0005	01000
pattern10	0.1022	0.9867	0.0152	0.0958	0.9882	01001
pattern11	0.0000	0.9981	0.0932	0.9746	0.0015	01010
pattern12	0.0152	0.9787	0.0041	0.8916	0.9952	01011
pattern13	0.0072	0.9540	0.9186	0.0701	0.0661	01100
pattern14	0.0157	0.8954	0.9532	0.0014	0.9999	01101
pattern15	0.0022	0.8298	0.9815	0.9553	0.0001	01110
pattern16	0.1233	0.9897	0.9817	0.9336	0.9162	01111
pattern17	0.9988	0.0944	0.0106	0.0893	0.0038	10000
pattern18	0.9969	0.0096	0.0931	0.0157	0.9204	10001
pattern19	0.9884	0.0356	0.0416	0.9169	0.1066	10010
pattern20	0.9575	0.0989	0.0008	0.9987	0.9465	10011
pattern21	0.9012	0.0828	0.9103	0.0952	0.0905	10100
pattern22	0.9701	0.0835	0.9998	0.0952	0.9085	10101
pattern23	0.9367	0.0934	0.9831	0.9241	0.0562	10110
pattern24	0.9642	0.0690	0.9609	0.9701	0.8959	10111
pattern25	0.8112	0.7360	0.0350	0.0370	0.1245	11000
pattern26	0.9735	0.9920	0.0966	0.0333	0.9033	11001
pattern27	0.9640	0.9527	0.0924	0.9664	0.0316	11010
pattern28	0.9088	0.9724	0.0246	0.9077	0.9047	11011
pattern29	0.9705	0.9293	0.9835	0.0731	0.0805	11100
pattern30	0.8659	0.8466	0.9474	0.0289	0.9794	11101
pattern31	0.9856	0.9508	0.8938	0.9995	0.0826	11110
pattern32	0.9155	0.9111	0.9040	0.9665	0.9976	11111

The results in scheme 1 and scheme 2 correlate to the human sense in a certain extent, but also have the ability to distinguish the differences between the similar images to human perception. The PSNR values indicate that the amount of added noise does not necessary be conscious of our sense. It can be said that our proposed scheme 1 has an ability to calculate the similarity in terms of the features in frequency domain. Note that our proposed method does not embed any data to target content which means that we cannot calculate the PSNR vale of our method. Hence, it is incomparable with any conventional information hiding method like digital watermarking or steganography in terms of imperceptibility. Our method is completely imperceptible because there are no data being embedded to target content and still can extract meaningful data which can be used for digital watermarking or similarity comparison as shown in this experiment.

DISCUSSION

In this paper, we have shown the basic idea of the damageless information hiding and retrieval scheme which can be used for digital watermarking. This framework uses original image to generate information extraction key which could be called differently according to its application, herein called as watermark extraction key. Also, this framework must prepare feature extraction key which holds the information about the coordinates of selected feature subblocks and the regions of the DCT frequency domain used in the machine learning process.

Based on the previous experimental results, we have shown the possibility for a robust infor-

Figure 22. Result for high pass filtered baboon

	network1	network2	network3	network4	network5	result
pattern1	0.0328	0.1141	0.0541	0.0662	0.1404	00000
pattern2	0.0037	0.0576	0.0003	0.1175	0.8898	00001
pattern3	0.0962	0.0002	0.0960	0.9086	0.0955	00010
pattern4	0.0073	0.0911	0.0580	0.8995	0.8994	00011
pattern5	0.0881	0.0763	0.9017	0.0856	0.0014	00100
pattern6	0.0494	0.0994	0.9632	0.0944	0.9103	00101
pattern7	0.0114	0.0003	0.9953	0.9187	0.0171	00110
pattern8	0.4198	0.1348	0.7606	0.9079	0.9735	00111
pattern9	0.0078	0.8528	0.0754	0.0369	0.0012	01000
pattern10	0.0742	0.9861	0.0148	0.1126	0.9892	01001
pattern11	0.0000	0.9980	0.0843	0.9794	0.0017	01010
pattern12	0.0117	0.9627	0.0061	0.8804	0.9926	01011
pattern13	0.0071	0.9490	0.9166	0.0759	0.0724	01100
pattern14	0.0150	0.9393	0.8830	0.0013	0.9999	01101
pattern15	0.0031	0.7162	0.9892	0.9553	0.0001	01110
pattern16	0.0378	0.9888	0.9872	0.8938	0.8867	01111
pattern17	0.9986	0.0929	0.0100	0.1524	0.0041	10000
pattern18	0.9964	0.0092	0.1346	0.0158	0.9260	10001
pattern19	0.9924	0.0504	0.0590	0.9270	0.0793	10010
pattern20	0.9363	0.0960	0.0008	0.9985	0.9441	10011
pattern21	0.9010	0.0827	0.9104	0.0950	0.0904	10100
pattern22	0.9551	0.0747	0.9997	0.0165	0.8758	10101
pattern23	0.9204	0.0891	0.9809	0.9227	0.0638	10110
pattern24	0.9831	0.0306	0.9739	0.9707	0.7104	10111
pattern25	0.9147	0.9609	0.0323	0.0619	0.0198	11000
pattern26	0.9693	0.9902	0.1290	0.0291	0.8939	11001
pattern27	0.9852	0.9547	0.2777	0.9640	0.0678	11010
pattern28	0.9088	0.9724	0.0246	0.9077	0.9047	11011
pattern29	0.9705	0.9304	0.9832	0.0736	0.0794	11100
pattern30	0.9634	0.9692	0.7569	0.0177	0.9909	11101
pattern31	0.9842	0.9404	0.9023	0.9994	0.0932	11110
pattern32	0.8753	0.9042	0.9063	0.9672	0.9979	11111

Figure 23. Observed signal values for network 1 [Baboon]

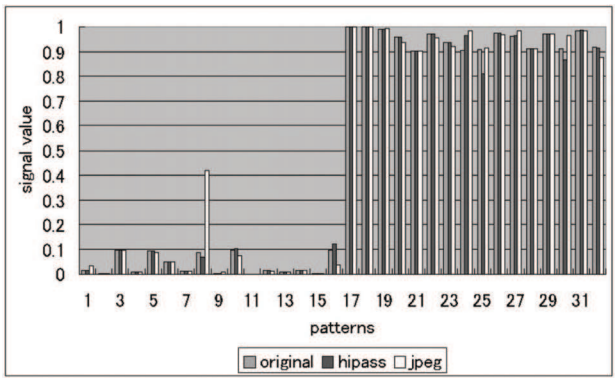


Figure 24. Observed signal values for network 2 [Baboon]

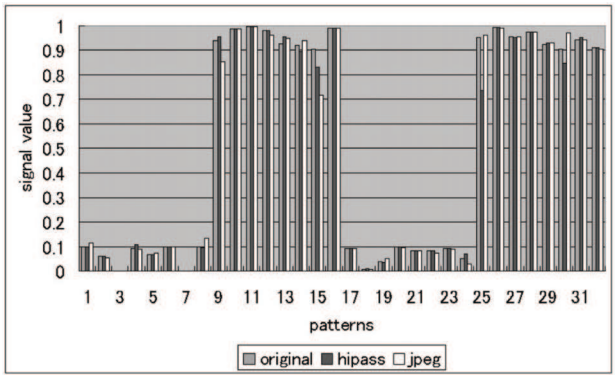


Figure 25. Observed signal values for network 3 [Baboon]

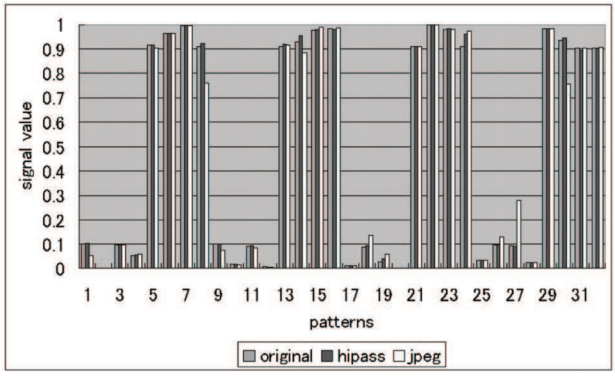


Figure 26. Observed signal values for network 4 [Baboon]

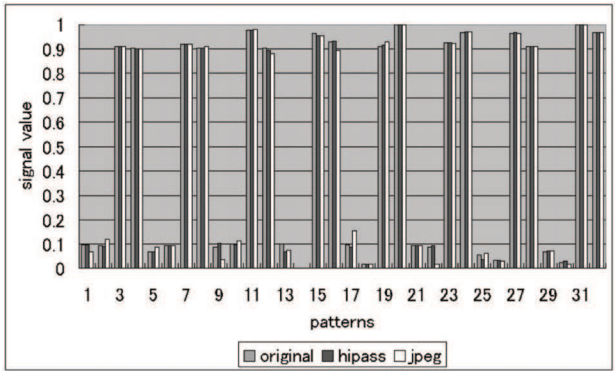


Figure 27. Observed signal values for network 5 [Baboon]

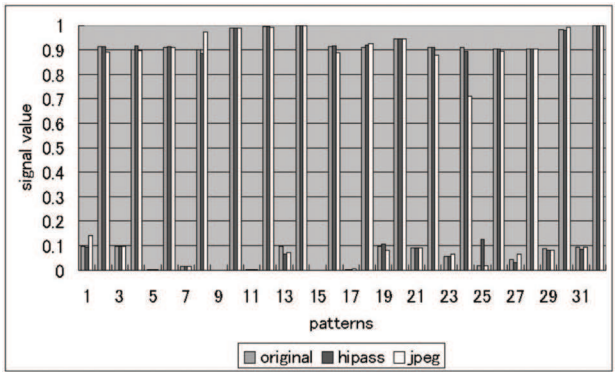


Figure 28. Image of Tiffany

Figure 29. Image of Airplane

mation hiding scheme without embedding any data into the target content. The contribution of our proposed method is that it has ability for damageless information hiding.

Some of the possible application of our proposed method is digital watermarking, steganography and fingerprinting in information security. For our proposed method to be used as digital watermarking, we must consider the generation time of feature extraction key and information extraction key. For feature extraction process, the processing time was less than a second when processed in ordinary computers. However, because the generation

Figure 30. Image of House

Figure 31. Image of Splash

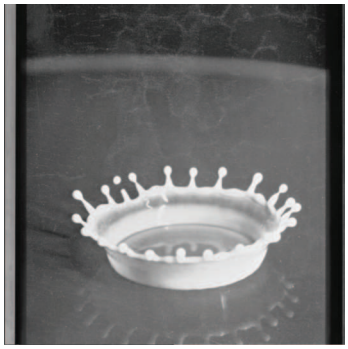


Figure 32. Observed signal values for network 1 [Various Images]

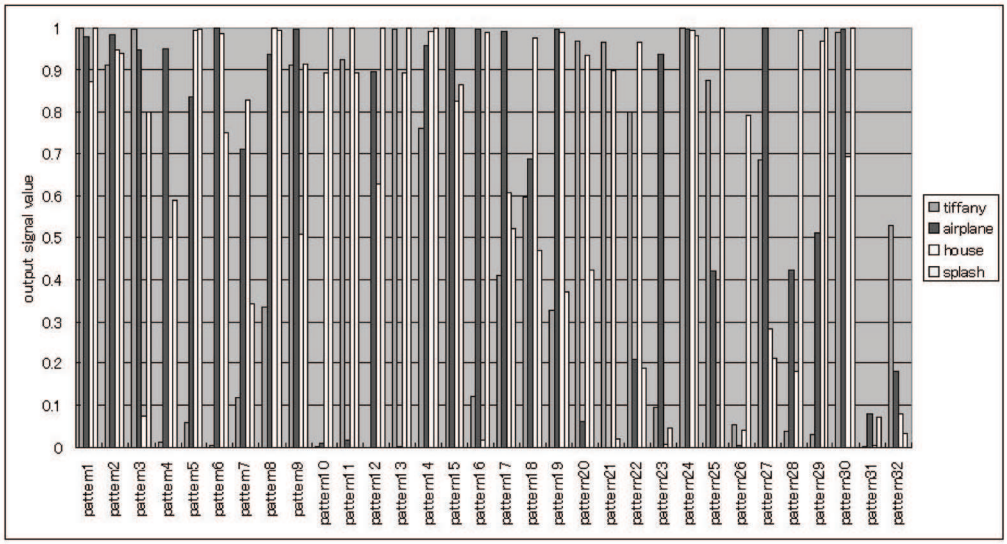


Figure 33. Observed signal values for network 2 [Various Images]

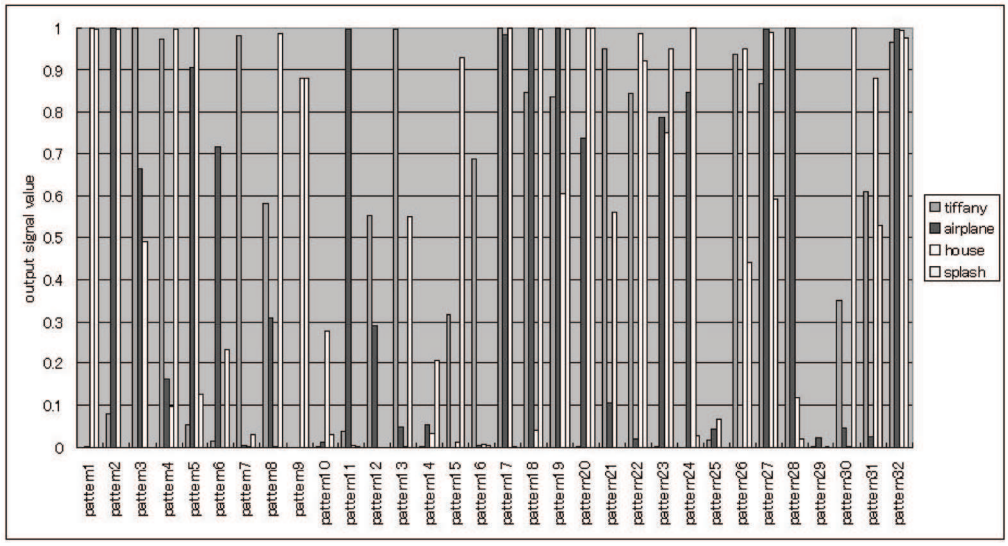
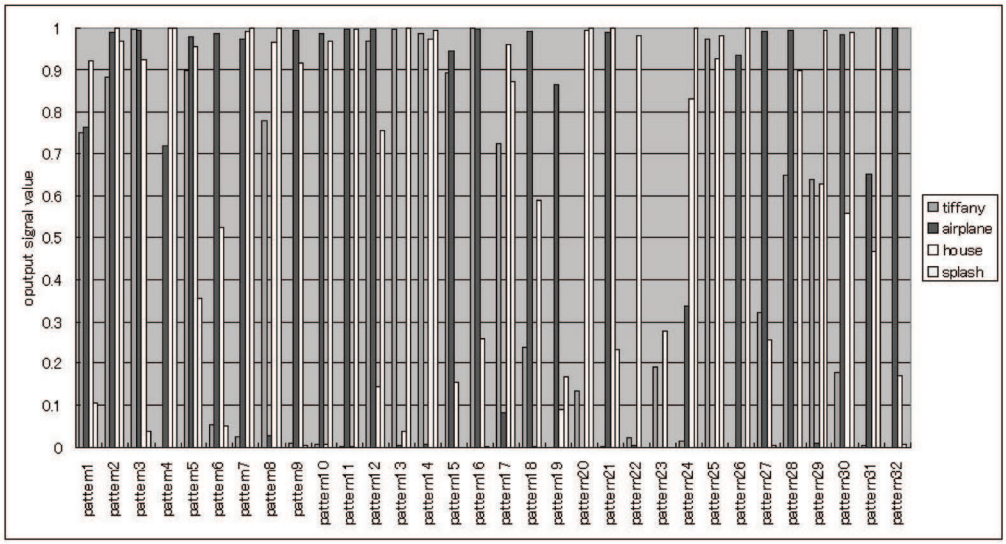


Figure 34. Observed signal values for network 3 [Various Images]



of information extraction key is a learning process of neural network which is equivalent to a non-linear approximation of patterns, the process measured about few minutes, although the generation of information patterns using information extraction key took less than a second. This is not considered as a problem, because the generation speed of key is not influential and only the speed for extraction of information pattern is important in information extraction process.

For extracted visual feature values, we used DCT coefficients on YCbCr domain. The proposed method does not limit to DCT as fre-

Figure 35. Observed signal values for network 4 [Various Images]

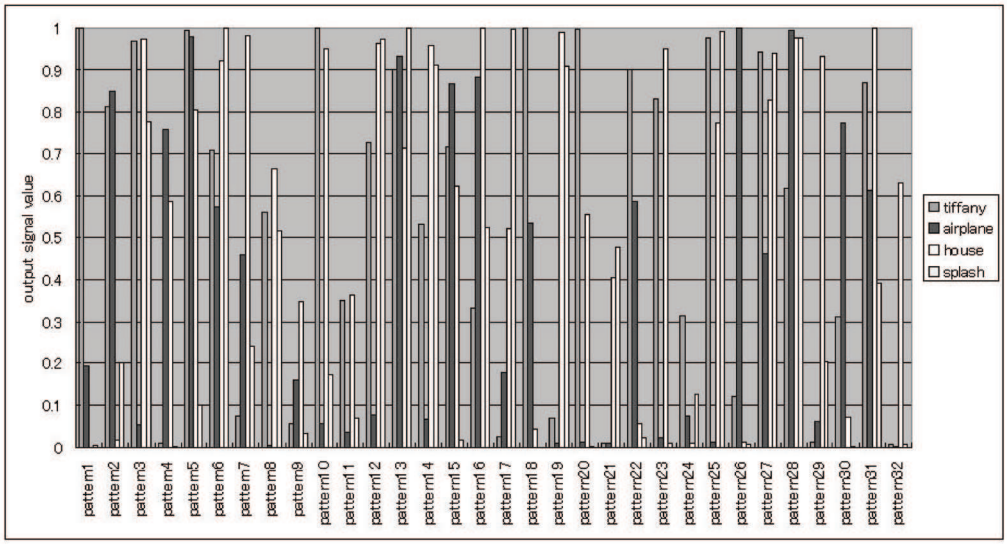
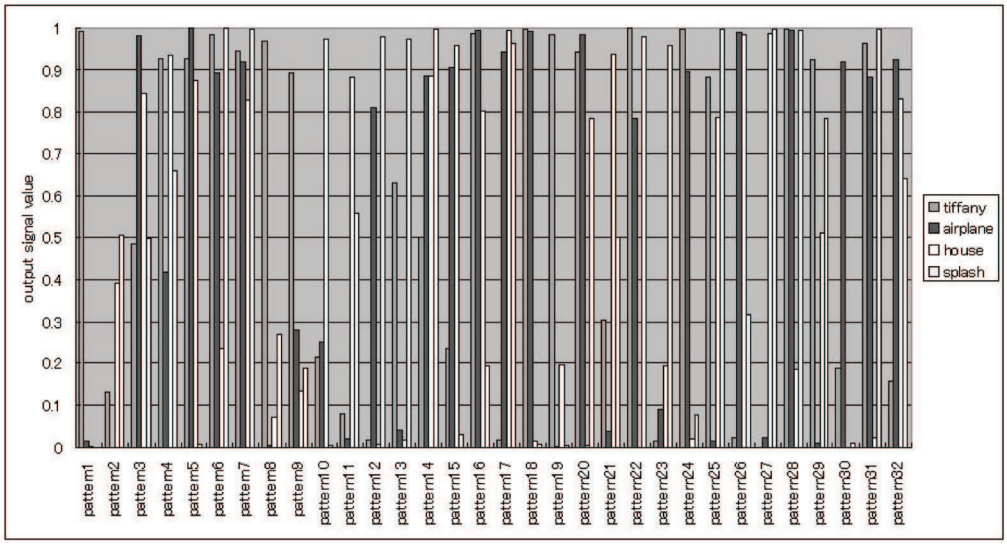


Figure 36. Observed signal values for network 5 [Various Images]



quency transformation method only, one can use DFT and DWT otherwise. Also, machine learning approach is not limited to neural networks only. Other machine learning approaches like Bayesian networks and fuzzy can be used as its learning method. Meanwhile, the limitation of the proposed method is that our proposed information extraction method relies on the position of the feature sub-blocks; it is weak to geometric attacks like shrinking, expanding, and rotation of the image. This problem must be considered as future works.

Figure 37. Experimental results for experiment 3

	In total patterns	In total set	Total time for the procedure to be completed	Average computational time for succeeded set	Average computational time for failed set
Lenna	639/640	19/20	440.427046s	11.700158s	173.858654s
Baboon	634/640	15/20	1668.796110s		
Tiffany	639/640	19/20	489.858861s		
Airplane	638/640	18/20	638.853265s		
House	632/640	16/20	1028.013773s		
Splash	639/640	19/20	556.783893s		

Figure 38. Experimental results for experiment 4

	Scheme 1	Scheme 2	PSNR Red	PSNR Green	PSNR Blue
Original Image	0	6.8277	inf	inf	inf
Convolution Filter [1]	0.3648	6.5152	1.10E+01	9.29E+00	8.41E+00
Convolution Filter [2]	0.1703	6.7676	3.81E+00	8.29E+00	8.41E+00
Median Filter [3]	0.3751	6.4491	2.45E+01	2.39E+01	2.39E+01
Median Filter [5]	0.4118	6.4443	2.46E+01	2.40E+01	2.41E+01
Median Filter [7]	0.4299	6.3832	2.43E+01	2.37E+01	2.40E+01
Median Filter [9]	0.4372	6.3926	2.39E+01	2.32E+01	2.38E+01
Noise Test [0]	0.0031	6.8197	2.59E+01	2.59E+01	2.59E+01
Noise Test [20]	0.4648	7.0637	6.59E+00	9.53E+00	9.40E+00
Noise Test [40]	0.4757	7.0629	4.88E+00	7.78E+00	7.77E+00
Noise Test [60]	0.4818	7.1527	4.43E+00	7.11E+00	7.24E+00
Noise Test [80]	0.4796	6.9672	4.21E+00	6.79E+00	7.00E+00
Noise Test [100]	0.4781	7.088	4.09E+00	6.65E+00	6.86E+00
Spread Spectrum Test [1]	0.281	6.6193	2.55E+01	2.59E+01	2.56E+01
Spread Spectrum Test [2]	0.0729	6.7933	2.59E+01	2.59E+01	2.49E+01
Spread Spectrum Test [3]	0.413	6.5428	2.53E+01	2.54E+01	2.49E+01
PSNR Test [20]	0.0015	6.8282	3.42E+01	3.42E+01	3.42E+01
PSNR Test [40]	0.0035	6.8218	2.82E+01	2.81E+01	2.81E+01
PSNR Test [60]	0.0059	6.8021	2.47E+01	2.46E+01	2.46E+01
PSNR Test [80]	0.0122	6.7933	2.23E+01	2.21E+01	2.21E+01
PSNR Test [100]	0.0132	6.8079	2.04E+01	2.02E+01	2.02E+01

CONCLUSION AND FUTURE WORK

In this paper, we have proposed a key generation method for damageless information hiding by machine learning, here, neural network as an exemplary technique. It is effective as its users do not damage any contents by concealing secret codes in target contents. The proposed method uses multi-layered neural network model for classifying the input visual feature patterns to corresponding hidden watermark patterns. For input visual feature values, we used DCT coefficients on YCbCr domain. The proposed method does not limit to DCT as frequency transformation method only, one can use DFT and DWT otherwise. Also, machine learning method can be replaced with others such as Bayesian network and fuzzy.

In the experiments, we have shown that our method is robust to high pass filtering and JPEG compression. The fact that our proposed method has robustness against graphical alteration implies that it is applicable in practical use, where noiseless environment will not be considered. Also, because our method does not alter any target contents in its structure, it is applicable to robust digital watermarking and

steganography too. Meanwhile, because the proposed method relies on the position of the feature values, it is weak to geometric attacks. In our future work, we will expand our method to geometrically altered images without damaging the target contents.

REFERENCES

Aihara, K. (1990). Chaotic neural networks . In Kawakami, H. (Ed.), *Bifurcation Phenomena in Nonlinear System and Theory of Dynamical Systems* (*Vol. 8*, pp. 143–161). Advanced Series on Dynamical Systems.

Ando, R., & Takefuji, Y. (2003). Location-driven watermark extraction using supervised learning on frequency domain. *WSEAS Transactions on Computers*, *1*(2), 163–169.

Artz, D. (2001). Digital steganography. *Internet Computing*, *5*(3), 75–80. doi:10.1109/4236.935180

Baumgarte, F., Ferekidis, C., & Fuchs, H. (1995). *A nonlinear psychoacoustic model applied to the ISO MPEG layer 3 coder.* Paper presented at the 99th AES Convention, New York, NY.

Bernardo, J., & Smith, A. (2001). Bayesian theory. *Measurement Science & Technology*, *12*, 221–222.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. New York, NY: Oxford University Press.

Bishop, C. M. (2006). *Pattern recognition and machine learning (information science and statistics)*. New York, NY: Springer-Verlag.

Boney, L., Tewfik, A. H., & Hamdy, K. N. (1996). Digital watermarks for audio signals. In *Proceedings of the International Conference on Multimedia Computing and Systems* (pp. 473-480).

Broomhead, D., & Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, *2*, 321–355.

Cheung, S. C., Chiu, D. K., & Ho, C. (2008). The use of digital watermarking for intelligence multimedia document distribution. *Journal of Theoretical and Applied Electronic Commerce Research*, *3*(3), 103–118. doi:10.4067/S0718-18762008000200008

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. doi:10.1007/BF00994018

Cox, I. J., Miller, M. L., Bloom, J., Fridrich, J., & Kalker, T. (2007). *Digital watermarking and steganography* (2nd ed.). San Francisco, CA: Morgan Kaufmann Publishers.

Cox, I. J., Miller, M. L., & Muttoo, S. K. (2002). *Digital watermarking*. San Francisco, CA: Morgan Kaufmann Publishers.

Delaigle, J. F., De Vleeschouwer, C., & Macq, B. (1998). Watermarking algorithm based on a human visual model. *Signal Processing*, *66*, 319–335. doi:10.1016/S0165-1684(98)00013-9

Eckhorn, R., Reitboeck, H. J., Arndt, M., & Dicke, P. (1990). Feature linking via synchronization among distributed assemblies: Simulations of results from Cat Visual Cortex. *Neural Computation*, *2*, 293–307. doi:10.1162/neco.1990.2.3.293

Elman, J. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211. doi:10.1207/s15516709cog1402_1

Gruhl, D., Lu, A., & Bender, W. (1996). Echo hiding. In *Proceedings of the First International Information Hiding Workshop* (LNCS 1174, pp.295-316).

Hung, P. C., Chiu, D. K., Fung, W. W., Cheung, W. K., Wong, R., & Choi, S. P. (2007). End-to-end privacy control in service outsourcing of human intensive processes: A multi-layered web service integration approach. *Information Systems Frontiers*, *9*(1), 85–101. doi:10.1007/s10796-006-9019-y

Kahn, D. (1996). The history of steganography. In *Information Hiding* (LNCS 1174, pp. 1-5).

Katzenbeisser, S., & Petitcolas, F. A. P. (2000). *Information Hiding Techniques for Steganography and Digital Watermarking*. London, UK: Artech House.

Kim, Y. C., Byeong, C., & Choi, C. (2002). Two-step detection algorithm in a HVS-based blind watermarking on still images. In *Digital Watermarking: Proceedings of the First International Workshop* (LNCS 2613, pp. 235-248).

Klir, G., & Yuan, B. (1995). Fuzzy sets and fuzzy logic: Theory and applications . In Kohonen, T. (Ed.), *Self-Organizing Maps*. New York, NY: Prentice Hall.

Kwok, S., Cheung, S., Wong, K., Tsang, K., Lui, S., & Tam, K. (2003). Integration of digital rights management into the Internet Open Trading Protocol. *Decision Support Systems*, *34*(4), 413–425. doi:10.1016/S0167-9236(02)00067-2

Naoe, K., & Takefuji, Y. (2008). Damageless information hiding using neural network on YCbCr domain. *International Journal of Computer Sciences and Network Security*, *8*(9), 81–86.

Petitcolas, F., Anderson, R., & Kuhn, M. (1998). Attacks on copyright marking systems. In *Proceedings of the Second International Workshop on Information Hiding* (pp. 218-238).

Reither, M. K., & Rubin, A. D. (1998). Crowds anonymity for web transactions. *Transaction on Information and System Security*, *1*(1), 66–92. doi:10.1145/290163.290168

Rosenblatt, F. (1958). The perceptron: Probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386–408. doi:10.1037/h0042519

Rothe, J. (2002). Some facets of complexity theory and cryptography: A five-lecture tutorial. *ACM Computing Surveys*, *34*(4), 504–549. doi:10.1145/592642.592646

Rummelhart, D. E., McClelland, J. L., & PDP Research Group. (1986). *Parallel distributed processing* (Vol.1). Cambridge, MA: The MIT Press.

Sasaki, H. (2007). *Intellectual property protection for multimedia information technology*. Hershey, PA: Idea Group Publishing.

Swanson, M. D., Zhu, B., & Tewfil, A. H. (1996). Transparent robust image watermarking. In . *Proceedings of the International Conference on Image Processing*, *3*, 211–214.

Westen, S. J. P., Lagendijk, R. L., & Biemond, J. (1996). Optimization of JPEG color image coding using a human visual system model. In *Proceedings of the Conference on Human Vision and Electronic Imaging* (LNCS 2657, pp. 370-381).

Wolf, P., Steinbach, M., & Diener, K. (2007). Complementing DRM with digital watermarking: mark, search, retrieve. *Online Information Review*, *31*(1), 10–21. doi:10.1108/14684520710731001

Wolfe, P. J., & Godsill, S. J. (2000). Towards a perceptually optimal spectral amplitude estimator for audio signal enhancement. In . *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, *2*, 821–824.