EURASIP Journal on Applied Signal Processing 2003:3, 252–263

Audio Watermarking Based on HAS and Neural Networks in DCT Domain

Hung-Hsu Tsai

Department of Information Management, National Huwei Institute of Technology, Yunlin, Taiwan 632, Taiwan
Email: thh@sunws.nhit.edu.tw

Ji-Shiung Cheng

No. 5-1 Innovation Road 1, Science-Based Industrial Park, Hsin-Chu 300, Taiwan
Email: FrankCheng@aiptek.com.tw

Pao-Ta Yu

Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan 62107, Taiwan
Email: csipty@ccunix.ccu.edu.tw

Received 8 August 2001 and in revised form 13 August 2002

We propose a new intelligent audio watermarking method based on the characteristics of the HAS and the techniques of neural networks in the DCT domain. The method makes the watermark imperceptible by using the audio masking characteristics of the HAS. Moreover, the method exploits a neural network for memorizing the relationships between the original audio signals and the watermarked audio signals. Therefore, the method is capable of extracting watermarks without original audio signals. Finally, the experimental results are also included to illustrate that the method significantly possesses robustness to be immune against common attacks for the copyright protection of digital audio.

Keywords and phrases: audio watermarking, data hiding, copyright protection, neural networks, human auditory system.

1. INTRODUCTION

The maturity of networking and data-compression techniques promotes an efficient distribution for digital products. However, illegal reproduction and distribution of digital audio products become much easier by using the digital technology with lossless data duplication. Hence, the illegal reproduction and distribution of music become a very serious problem in protecting the copyright of music [1]. Recently, the approach of digital watermarking has been effectively employed to protect intellectual property of digital products including audio, image, and video products [2, 3, 4, 5, 6, 7, 8].

The techniques of conventional cryptography protect the content from anyone without private decrypted keys. They are actually useful in protecting an audio from being intercepted during data transmission [1]. However, the encryption data (cipher-text) must be decrypted for the access to the original audio data (plain-text). In contrast to the conventional cryptography, the watermarking straightforwardly accesses encryption data (watermarked data) as original data. Moreover, a watermark is designed for residing permanently in the original audio data after repeated reproduction and redistribution. Furthermore, the watermark cannot be removed from the audio data by the intended counterfeiters. Consequently, the watermark technique could be applied to establish the ownership of digital audio for copyright protection and authentication. An audio watermarking method has been proposed in [4] to effectively protect the copyright of audio. However, Swanson's method requires the original audio for the watermark extraction. This kind of watermarking methods fails to identify the owner copyright of audio due to the ambiguity of ownerships. More specifically, a pirate inserts his (or her) counterfeit watermark into the watermarked data, and then extract the counterfeit watermark from contested data. This problem is also referred to as the deadlock problem in [4]. Therefore, on the basis of the characteristics of the human auditory system (HAS) and the techniques of neural networks, this paper presents a new audio watermarking method without the original audio for the watermark extraction.

In order to achieve the copyright protection, the proposed method needs to meet the following requirements [5]:

(i) the watermark should be inaudible to human ears;

(ii) watermark detection should be done without referencing the original audio signals;

(iii) the watermark should be undetectable without prior knowledge of the embedded watermark sequence;

(iv) the watermark is directly embedded in the audio signals, not in a header of the audio;

(v) the watermark is robust to resist common signal-processing manipulations such as filtering, compression, filtering with compression, and so on.

Section 2 introduces basic concepts for the frequency-masking used in the MPEG-I Psychoacoustic model 1. Section 3 states the watermark-embedding algorithm on the discrete cosine transformation (DCT) domain. Section 4 describes the watermark-extraction algorithm on the DCT domain. Section 5 exhibits the experimental results illustrating that the proposed method is capable of protecting the ownership of audio from attacks. A brief conclusion is available in Section 6.

2. FREQUENCY-MASKING

Frequency-masking refers to masking between frequency audio components [4]. If two signals, which occur simultaneously, are close together in frequency, the lower-power (fainter) frequency components may be inaudible in the presence of the higher-power (louder) frequency components. The masking threshold of a mask is determined by the frequency, sound pressure level (SPL), and tonal-like or noise-like characteristics of both the mask and the masked signal [9]. When the SPL of the broadband noise is larger than the SPL of the tonal, the broadband noise can easily mask the tonal. Moreover, higher-power frequency signals are masked more easily. Note that the frequency-masking model defined in ISO-MPEG I Audio Psychoacoustic model 1 for layer I is exploited in the proposed method to obtain the spectral characteristics of a watermark based on the inaudible information of the HAS [10, 11, 12].

An algorithm for the calculation of the frequency-masking in the MPEG-I Psychoacoustic model 1 is described in Algorithm 1. For convenience, the algorithm is named determining-frequency-masking-threshold (DFMT) algorithm. More details on the DFMT algorithm can be obtained from [4].

As a result, Figure 1 shows a portion of an audio with 44.1 kHz sampling rate, which is expressed by the power spectrum. Frequency samples and masking values are represented by the solid line and dash line, respectively. The dash line, the frequency-masking threshold, is denoted by LTg in this paper.

```
Step 1: Calculation of the power spectrum
Step 2: Determination of the threshold in quiet (absolute
        threshold)
Step 3: Finding the tonal and nontonal components of the
audio
Step 4: Decimation of tonal and nontonal masking components
Step 5: Calculation of the individual masking thresholds
Step 6: Determination of the global masking threshold
```

Algorithm 1: Algorithm of the frequency-masking.

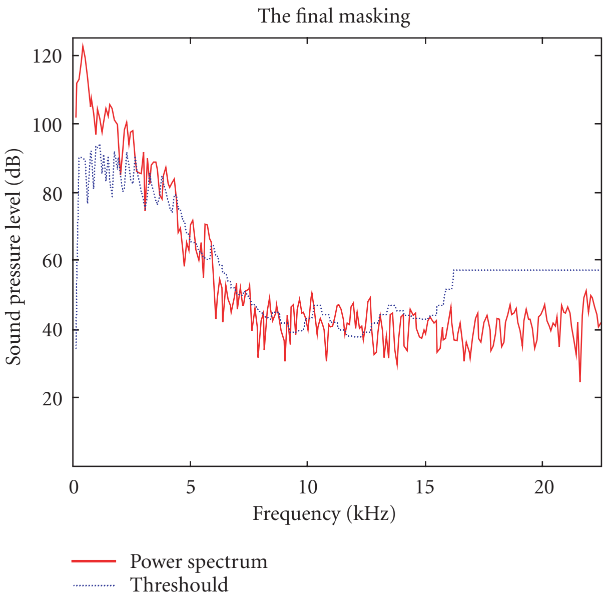


Figure 1: Original spectrum and frequency-masking threshold LTg.

3. WATERMARK EMBEDDING

Let an audio $X = (x_1, \ldots, x_N)$ with N PCM (pulse-code modulation) samples be segmented into $\phi = \lfloor N/256 \rfloor$ blocks. Each block includes 256 samples. Accordingly, a set of blocks Ψ can be defined by

$$\Psi = \{s_1, \ldots, s_i, \ldots, s_\phi\}, \tag{1}$$

where $s_i = (s_i(0), \ldots, s_i(k), \ldots, s_i(255))$ and $s_i(k)$ denotes the kth sample of the ith block. In order to secure information related to the watermark against attacks, we use a pseudo-random number generator (PRNG) to determine a set of target blocks φ selected from Ψ [13]. This φ can be represented by

$$\varphi = \{s_{\rho_j} \mid j = 1, \ldots, p \times q \text{ and } \rho_j \in \{0, \ldots, \phi - 1\}\} \tag{2}$$

when $p \times q$ blocks are selected. Note that p and q will be further defined in the following subsection. A scheme for the PRNG is expressed by

$$r = \text{PRNG}(z), \tag{3}$$

where r is a random number and z denotes a seed of the PRNG. This ρ_j can be calculated by

$$\rho_j = r \bmod \phi. \tag{4}$$

In this paper, a binary stamp image with size $p \times q$ is taken as a watermark. The stamp image can be represented

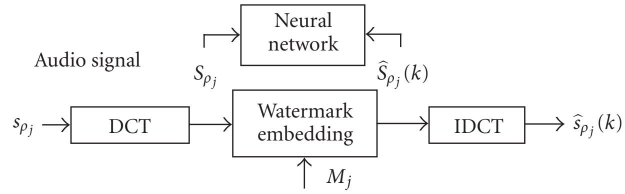


Figure 2: The structure of watermark embedding used in the proposed method.

by a sequence in a row-major fashion and expressed by

$$\begin{aligned} H_{p,q} &= (\sigma_{11},\dots,\sigma_{1q},\sigma_{21},\dots,\sigma_{2q},\dots,\sigma_{ik},\dots,\sigma_{p1},\dots,\sigma_{pq}) \\ &= (w_1,\dots,w_j,\dots,w_{pq}), \end{aligned} \tag{5}$$

where $H_{p,q}$ is a $(p \times q)$-bits binary sequence, $\sigma_{ik} \in \{0,1\}$, $1 \le i \le p$, and $1 \le k \le q$. Moreover, σ_{ik} stands for a pixel at position (i,k) in the binary image. For convenience, $H_{p,q}$ can be denoted by $\mathbf{w} = (w_1, w_2, \dots, w_{pq})$ as a vector with $p \times q$ components where $w_j = 2\sigma_{ik} - 1$, $j = (i-1) \times q + k$, and $1 \le j \le p \times q$. Consequently, we have $w_j \in \{-1, 1\}$ for each j. More specifically, w_j is -1 if a pixel of the binary stamp image is black ($\sigma_{ik} = 0$) and w_j is 1 if a pixel of the binary stamp image is white ($\sigma_{ik} = 1$).

The structure of the watermark embedding is depicted in Figure 2, which consists of four components: DCT, watermark embedding, inverse DCT (IDCT), and neural network (NN). This s_{ρ_j} can be DCT transformed to be the DCT transformed block S_{ρ_j} via using

$$S_{\rho_j}(l) = \sum_{n=1}^{256} \varpi(n) s_{\rho_j}(n) \cos \frac{\pi(2n-1)(l-1)}{512}, \tag{6}$$

where $1 \le l \le 256$, $s_{\rho_j}(n)$ denotes the nth PCM sample in the block s_{ρ_j} on the time domain, $S_{\rho_j}(l)$ is the lth DCT coefficient (frequency value) in S_{ρ_j}, and

$$\varpi(n) = \begin{cases} \dfrac{1}{256}, & \text{if } n = 1, \\ \sqrt{\dfrac{2}{256}}, & \text{if } 2 \le n \le 256. \end{cases} \tag{7}$$

Using (6) and (7), a set of the DCT transformed blocks Φ, associated with φ can be obtained and represented by

$$\Phi = \{S_{\rho_j} \mid j = 1,\dots,p \times q \text{ and } \rho_j \in \{0,\dots,\phi - 1\}\}. \tag{8}$$

During the watermark-embedding process, a watermark $\mathbf{w}$ is embedded into Φ by hiding w_j into $S_{\rho_j}(j_0)$ for each j where j_0 is a fixed index of each DCT transformed block and $j_0 \in \{100,\dots,200\}$. This fixed index, j_0, is determined by an algorithm as described in Algorithm 2. Note that the middle band in one block contains DCT coefficients with indices from 100 to 200.

```
Step 1: For each $s_i \in \Psi$, using the DFMT algorithm to obtain $S_i$
        and the global masking threshold $\mathrm{LTg}_i$ where
        $i = 1, 2, \dots, \phi$
Step 2: Set each $\mathrm{acc}(j)$ to 0 for $j = 100, \dots, 200$
Step 3: For each $S_i(j)$, $\mathrm{acc}(j) = \mathrm{acc}(j) + 1$ if
$[\mathrm{LTg}_i(j) - S_i(j) - \alpha] > 0$,        $\alpha$ is a constant
Step 4: $j_0 = \arg\max_{100 \le j \le 200}\{\mathrm{acc}(j)\}$
Step 5: Output $j_0$
```

Algorithm 2: The algorithm of determining j_0.

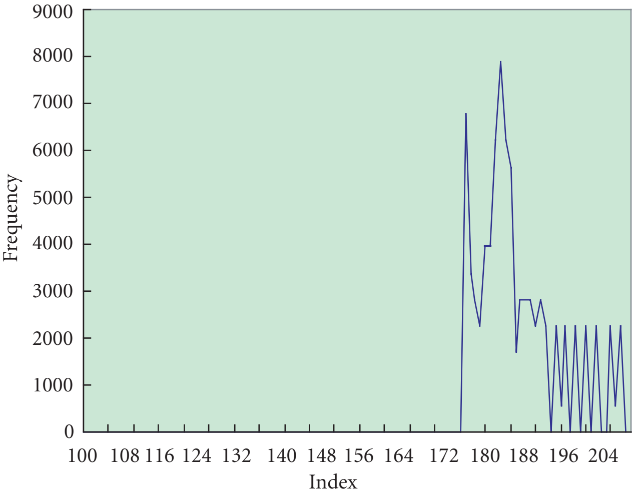


Figure 3: The frequency of each positive difference ($\mathrm{LTg}_i(j) - S_i(j) - \alpha > 0$) as a function of indices j where $100 \le j \le 200$.

The main purpose of the algorithm is to select an index j_0 such that the differences $\mathrm{LTg}_i(j_0) - S_i(j_0)$ of most blocks at index j_0 are greater than 0. Different j_0 may be chosen for distinct audio signals. An example of a test audio signal, a curve shown in Figure 3 plots the frequency of each positive difference (only considering $\mathrm{LTg}_i(j) - S_i(j) - \alpha > 0$) as a function of indices j where $100 \le j \le 200$. From Figure 3, the highest frequency occurs at index 183, thus we choose $j_0 = 183$.

After j_0 is determined for an audio signal, each w_j is embedded into $S_{\rho_j}(j_0)$ via the modification to $S_{\rho_j}(j_0)$ during the watermark-embedding process. The formula of the modification to $S_{\rho_j}(j_0)$ can be defined by

$$\hat{S}_{\rho_j}(j_0) = S_{\rho_j}(j_0) + M_j, \tag{9}$$

where $w_j \in \{-1, 1\}$, $M_j = w_j \times \alpha$, and $\alpha = 200$. Appropriate values for α can balance imperceptible (inaudible) and robust capabilities of our watermarking method. Lower α makes watermarks imperceptible. However, it reduces the robustness of the watermarks on resisting attacks or signal manipulations. In contrast, higher α makes the watermarks robust. However, it leads the watermarks to be

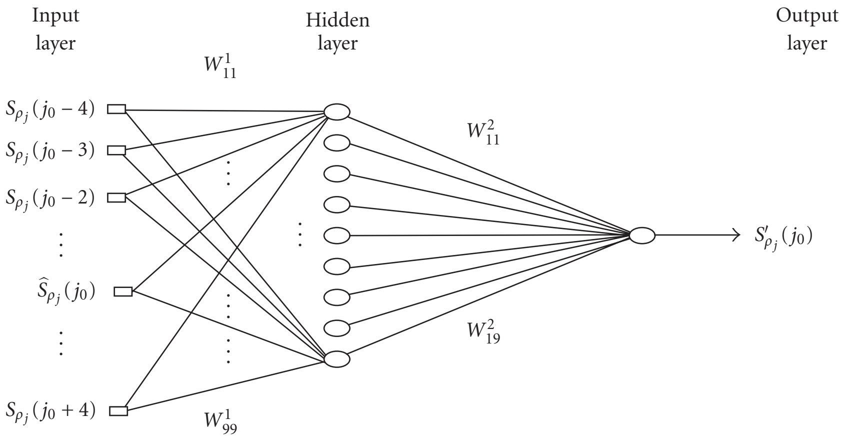


FIGURE 4: The architecture of a neural network used in the process of watermark embedding.

perceptible. Here, $\hat{S}_{\rho_j}$ indicates a watermarked-and-DCT-transformed audio block. For each j, a set of watermarked-and-DCT-transformed audio blocks $\hat{\Phi}$ can be calculated by (9) and denoted by

$$\hat{\Phi} = \{\hat{S}_{\rho_j} \mid j = 1,\dots,p\times q \text{ and } \rho_j \in \{0,\dots,\phi-1\}\}. \quad (10)$$

Each $\hat{S}_{\rho_j}$ can be transformed by IDCT to obtain $\hat{s}_{\rho_j}$, called a watermarked audio block. Then, a set of watermarked audio blocks $\hat{\varphi}$ can be obtained, and $\hat{\varphi}$ is denoted by

$$\hat{\varphi} = \{\hat{s}_{\rho_j} \mid j = 1,\dots,p\times q \text{ and } \rho_j \in \{0,\dots,\phi-1\}\}. \quad (11)$$

Consequently, the watermarked audio can be obtained and represented by

$$\hat{\Psi} = \{\hat{s}_1,\dots,\hat{s}_i,\dots,\hat{s}_\phi\} \quad (12)$$

or

$$\hat{X} = (\hat{x}_1,\dots,\hat{x}_k,\dots,\hat{x}_N), \quad (13)$$

where each $\hat{s}_i$ and each $\hat{x}_k$ may be altered.

Figure 4 shows the architecture of NN, called a 9-9-1 multilayer perceptron. Namely, the NN comprises an input layer with 9 nodes, a hidden layer with 9 nodes, and an output layer with a single node [14]. In addition, the backpropagation algorithm is adopted for training the NN over a set of training patterns Γ that is specified by

$$\Gamma = \{(A_j, B_j) \mid j = 1,2,\dots,p\times q\}, \quad (14)$$

where $|\Gamma|$ is $p\times q$. Moreover, an input vector A_j for the NN can be represented by

$$A_j = (S_{\rho_j}(j_0-4),\dots,S_{\rho_j}(j_0-1),\hat{S}_{\rho_j}(j_0), S_{\rho_j}(j_0+1),\dots,S_{\rho_j}(j_0+4)), \quad (15)$$

and the desired output B_j corresponding to the input vector A_j is $S_{\rho_j}(j_0)$. The dependence of the performance of the NN on the number of hidden nodes can be found in [14]. In this case, the performance of using more than 9 nodes in the hidden layer of the NN is not improved significantly. As the training process for the NN is completed, a set of synaptic weights W, characterizing the behavior of the trained neural network (TNN), can be obtained and represented by

$$W = \{W^1_{uv} \mid u = 1,2,\dots,9,\ v = 1,2,\dots,9\} \cup \{W^2_{uv} \mid u = 1,\ v = 1,2,\dots,9\}. \quad (16)$$

Accordingly, the TNN performs a mapping from the space in which A_j is defined to the space in which B_j is defined. In other words, the TNN can memorize the relationship (mapping) between the watermarked audio and the original audio.

4. WATERMARK EXTRACTION

One of the merits of the proposed watermarking method is to extract the watermark without the original audio. The TNN, obtained from the watermark embedding, can memorize the relationships between an original audio and the corresponding watermarked audio. Listed below are the parameters which are required in the watermark extraction and which have to be secured by the owner of the watermark or the original audio.

(i) All synaptic weights of the TNN, W.
(ii) The seed z for the PRNG.
(iii) The embedding index j_0 for each block.
(iv) The number of the bits $p\times q$ of the watermark $\mathbf{w}$.

Figure 5 shows the structure of watermark extraction in the method, which is composed of two components: DCT and TNN. First, the watermarked blocks in $\hat{\Psi}$ are selected by using (3) and (4) to construct $\hat{\varphi}$. Each watermarked audio block $\hat{s}_{\rho_j}$ in $\hat{\varphi}$ can be transformed by (17), and then, we have

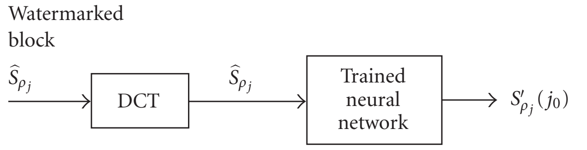


FIGURE 5: The structure of watermark extraction for the use of the TNN.

the watermarked-and-DCT-transformed audio block $\hat{S}_{\rho_j}$,

$$\hat{S}_{\rho_j}(l) = \sum_{n=1}^{256} \varpi(n)\hat{s}_{\rho_j}(n)\cos\frac{\pi(2n-1)(l-1)}{512}, \tag{17}$$

where $\hat{s}_{\rho_j}(n)$ denotes the nth PCM sample in the watermarked audio block $\hat{s}_{\rho_j}$, and $1 \le l \le 256$. Accordingly, a set of watermarked-and-DCT-transformed audio blocks $\hat{\Phi}$ can be obtained before the procedure of estimating the original audio.

During the watermark-extraction process, the TNN is employed to estimate the original audio. Let an input vector for the TNN be expressed by

$$\begin{aligned}&(\hat{S}_{\rho_j}(j_0-4),\ldots,\hat{S}_{\rho_j}(j_0-1),\hat{S}_{\rho_j}(j_0),\\&\hat{S}_{\rho_j}(j_0+1),\ldots,\hat{S}_{\rho_j}(j_0+4)),\end{aligned} \tag{18}$$

which is selected from $\hat{S}_{\rho_j}$ in $\hat{\Phi}$ that may be further distorted by attacks or manipulations of signal processing. In addition, $S'_{\rho_j}(j_0)$ denotes the physical output for the TNN when (18) is fed into the TNN. Figure 6 shows the input pattern and the corresponding physical output for the TNN. An extracted watermark can be represented by

$$\mathbf{w}' = (w'_1,\ldots,w'_j,\ldots,w'_{pq}). \tag{19}$$

Using (9), simple algebraic operations, the watermarked sample $\hat{S}_{\rho_j}(j_0)$, and the corresponding physical output (estimated sample) $S'_{\rho_j}(j_0)$ for the TNN, the jth bit of the extracted watermark w'_j can be estimated by

$$w'_j = \begin{cases} 1, & \text{if } [\hat{S}_{\rho_j}(j_0) - S'_{\rho_j}(j_0)] > 0,\\ -1, & \text{else.}\end{cases} \tag{20}$$

Note that the estimated sample $S'_{\rho_j}(j_0)$ will be equal to the original sample $S_{\rho_j}(j_0)$ if no estimated errors occur for the TNN. In fact, it is impossible for the TNN to perform the exact mapping in many applications [14]. The extracted watermark can be reconstructed into a binary stamp image according to (20). The corresponding pixel of the binary stamp image (watermark) is black if $w'_j = -1$. Otherwise, the pixel of the binary image is white if $w'_j = 1$.

5. EXPERIMENTAL RESULTS

In this experiment, two binary stamp images with size 64×64 (i.e., $p = q = 64$), displayed in Figure 7, are taken as the proof (original) watermark $\mathbf{w} = (w_1, w_2, \ldots, w_{4096})$. Three tested audio (excerpts) with 44.1 kHz sampling rate, as depicted in Figures 8a, 8c, and 8e, are used for examining the performance of our watermarking method. During the watermark-embedding process, $\mathbf{w}$ is embedded into an audio X (Ψ) to obtain the watermarked audio X' (Ψ'). In the case under consideration, Figure 7a is embedded into the first and the second original audio separately. Their watermarked versions are depicted in Figures 8b and 8d, respectively. Figure 7b is embedded into the third audio, and its watermarked audio is depicted in Figure 8f. To observe Figure 8, these three watermarked audio are almost similar to their original versions. Therefore, the proposed method remarkably possesses imperceptible capability for making watermarks inaudible. More specifically, imperceptible capability of the method is granted by frequency-masking and the algorithm, as described in Table 2, of selecting an index j_0.

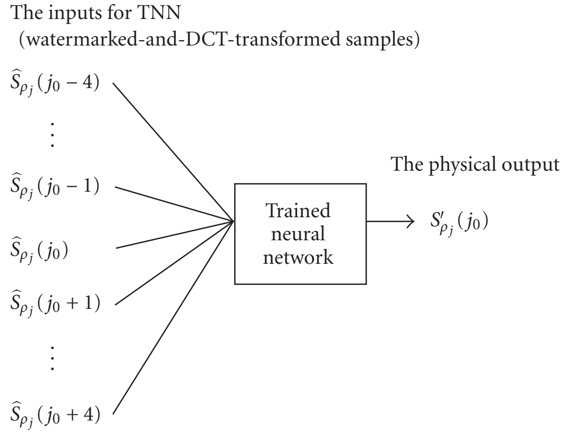
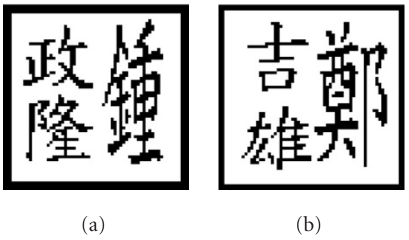

The inputs for TNN
(watermarked-and-DCT-transformed samples)
$\hat{S}_{\rho_j}(j_0-4)$
$\hat{S}_{\rho_j}(j_0-1)$
$\hat{S}_{\rho_j}(j_0)$
$\hat{S}_{\rho_j}(j_0+1)$
$\hat{S}_{\rho_j}(j_0+4)$
Trained neural network
The physical output
$S'_{\rho_j}(j_0)$

FIGURE 6: The inputs and output for the TNN when a watermark is extracted.



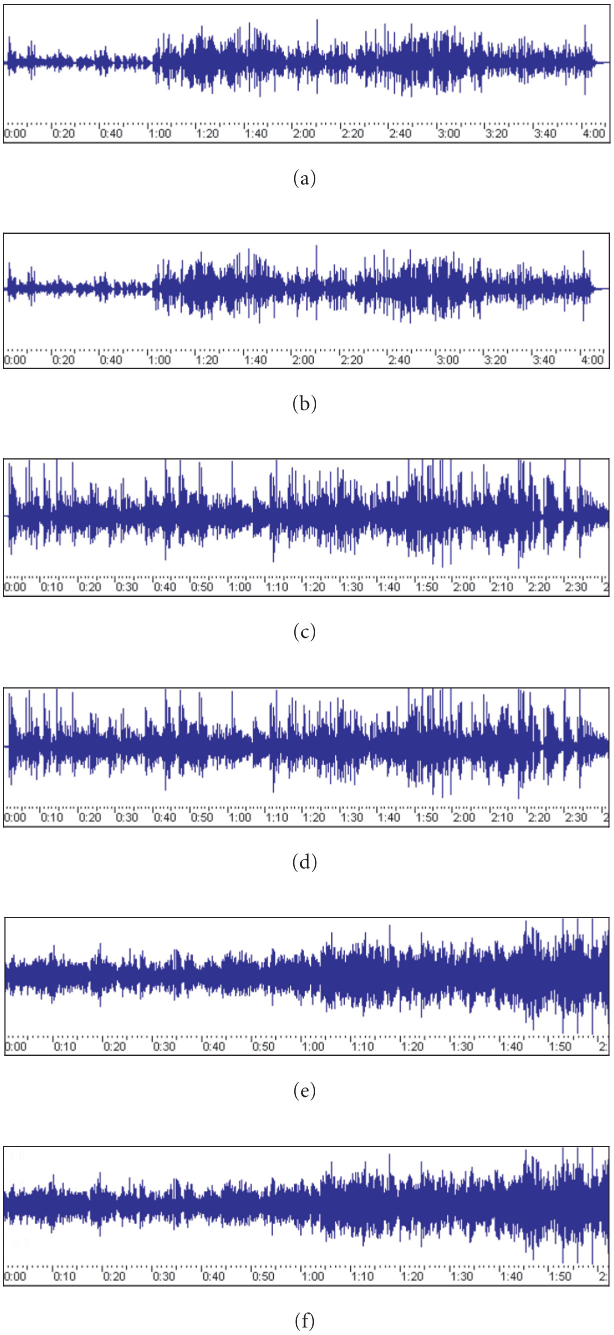
(a) (b)

FIGURE 7: Two proof (original) watermarks with size 64 × 64.

In order to evaluate the performance of watermarking methods, one quantitative index, that is employed to measure the quality of an extracted watermark, is defined by

$$DR(\mathbf{w},\mathbf{w}') = \frac{\mathbf{w}'\mathbf{w}^T}{p \times q}, \tag{21}$$

where $\mathbf{w}$ is a vector that denotes an original watermark (a binary stamp image) and $\mathbf{w}'$ is a vector that stands for an



(a)

(b)

(c)

(d)

(e)

(f)

Figure 8: (a), (c), and (e) show the first, the second, and the third original audio (X), respectively. (b), (d), and (f) show their corresponding watermarked audio ($\hat{X}$) with $\alpha = 200$ and $j_0 = 183$, respectively.

extracted watermark. Note that DR indicates the similarity between $\mathbf{w}$ and $\mathbf{w}'$. The vector $\mathbf{w}'$ is more similar to $\mathbf{w}$ if DR is closer to 1.

In this experiment, the method is investigated for the memorized, adaptive (generalized), and robust capabilities. The memorized capability of the method is evaluated by

Table 1: The DR values and the number of correct pixels in $\mathbf{w}'_{\text{filter},m}$ for $m = 16, 18, 20$, and 22 when these three audio are examined.

	The first audio is examined	
m	DR	# of correct pixels in $\mathbf{w}'_{\text{filter},m}$
16	0.248535	2557
18	0.929199	3951
20	0.961426	4017
22	0.963379	4021
	The second audio is examined	
m	DR	# of correct pixels in $\mathbf{w}'_{\text{filter},m}$
16	0.641602	3362
18	0.995117	4086
20	0.998535	4093
22	0.998535	4093
	The third audio is examined	
m	DR	# of correct pixels in $\mathbf{w}'_{\text{filter},m}$
16	−0.025391	1996
18	0.934082	3961
20	0.962891	4020
22	0.965820	4026

Table 2: The DR values and the number of correct pixels in $\mathbf{w}'_{\text{MF},l}$ for $l = 5, 7, 9$, and 11 when these three audio are examined.

	The first audio is examined	
l	DR	# of correct pixels in $\mathbf{w}'_{\text{MF},l}$
5	0.813477	3714
7	0.817383	3722
9	0.817383	3722
11	0.770996	3627
	The second audio is examined	
l	DR	# of correct pixels in $\mathbf{w}'_{\text{MF},l}$
5	0.744141	3572
7	0.771484	3628
9	0.732422	3548
11	0.679688	3440
	The third audio is examined	
l	DR	# of correct pixels in $\mathbf{w}'_{\text{MF},l}$
5	0.836426	3761
7	0.847168	3783
9	0.830078	3748
11	0.817383	3722

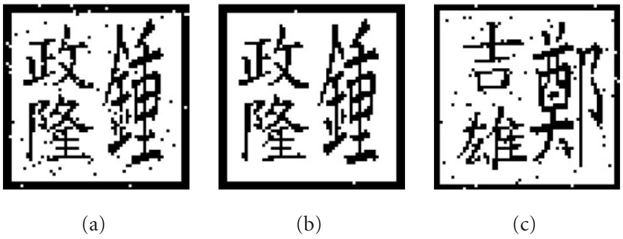


Figure 9: (a), (b), and (c) are estimated watermarks that are extracted from Figures 8b, 8d, and 8f, respectively, in the case of attack free.

taking the training audio as the testing audio. On the other hand, the adaptive and robust capabilities of the method can be simultaneously assessed by taking the distorted-and-watermarked audio as the testing audio. A watermarked audio is called the distorted-and-watermarked audio if the watermarked audio is further degraded by signal-processing manipulations such as filtering, MP3 compression/decompression (ISO/MPEG-I audio layer III), and multiple manipulations (filtering and MP3 compression/decompression).

5.1. *Attack free*

Let Γ denote a set of training patterns constructed by using a pair of the original audio X and watermarked audio $\hat{X}$ ($\hat{\Psi}$) that is not distorted by signal-processing manipulations. After the watermark-embedding process of the method is completed, a set of synaptic weights W can be identified to characterize the TNN. We collect the input vectors in Γ to form a set of the testing patterns $\Upsilon = \{A_j \mid j = 1, 2, \ldots, p \times q\}$. That is, the set of test patterns is the same as the set of the input vectors in the training patterns. Hence, only memorized capability of the method is examined in this case. During the watermark-extraction process, the set of the testing patterns is fed into the TNN to estimate the original samples. Then, $\mathbf{w}'$ can be extracted. Note that $\mathbf{w}'$ stands for $(w_1', w_2', \ldots, w_{4096}')$, and the length of $\hat{X}$ is the same as that of X. Three estimated watermarks ($\mathbf{w}'$) for these three audio are shown in Figure 9. Their *DR* values of the extracted watermarks are 0.963, 0.999, and 0.966, respectively. These three *DR* values are very close to 1. Besides the measure of using quantitative index *DR*, Figure 9 is further compared with Figure 7 via the measure of using visual perception. Here, Figure 9 is very similar to Figure 7. More specifically, in Figure 9, these three Chinese words can be recognized clearly. Manifestly, the method possesses a well-memorized capability so as to extract watermarks without the information of the original audio. In addition to the assessment of the memorized capability of the method, Sections 5.2, 5.3, and 5.4, we further exhibit the adaptive and robust capabilities of the method against five common audio manipulations.

5.2. *Robustness to filtering*

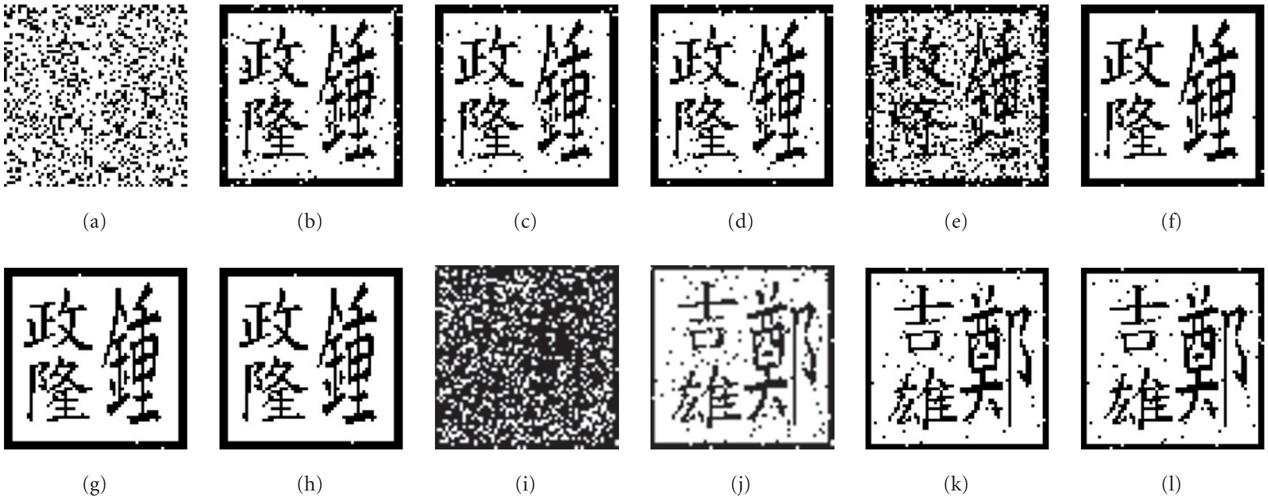
Let $\hat{X}_{\text{filter},m}$ ($\hat{\Psi}_{\text{filter},m}$) be represented as a filtered-and-watermarked audio. Namely, a watermarked audio $\hat{X}$ is further filtered by a filter with the cutting-off frequency in m kHz. Note that the behavior of the filter is to pass the frequency below m kHz. In this test, there are four different filtered-and-watermarked audio $\hat{X}_{\text{filter},m}$ for $m = 16, 18, 20$, and 22. The adaptive and robust capabilities of the method under the case of filtering attack are examined by extracting the watermark from the filtered-and-watermarked audio $\hat{X}_{\text{filter},m}$. First, the watermarked blocks in $\hat{\Psi}_{\text{filter},m}$ are selected by using (3) and (4) to construct $\hat{\varphi}_{\text{filter},m}$. Let $\Upsilon_{\text{filter},m}$ stand for a set of testing patterns obtained from the watermarked audio $\hat{\varphi}_{\text{filter},m}$. Then, $\Upsilon_{\text{filter},m}$ is fed into the TNN, and the estimated watermark $\mathbf{w}_{\text{filter},m}'$ is obtained by using (20). Table 1 shows the results of evaluating the robust performance of the method for assisting the filtering attacks. Using the measure of the visual perception, the similarity between $\mathbf{w}$ and $\mathbf{w}_{\text{filter},m}'$ is exhibited in Figure 10 for each m. However, the method breaks down in two cases of examining the first and the third audio when m is less than or equal to 16.

A class of nonlinear filters is called median filters (MFs) that have been employed to efficiently restore the signals (audio and images) corrupted by impulse or salt-peppers noises [15, 16]. We denote $\hat{X}_{\text{MF},l}$ ($\hat{\Psi}_{\text{MF},l}$) as an MF-and-watermarked audio if a watermarked audio $\hat{X}$ is further filtered by an MF with window length l. Four distinct cases, for $l = 5, 7, 9$, and 11, are examined in this experiment. By the similar procedure used in the case of filtering, the estimated watermark $\mathbf{w}_{\text{MF},l}'$ can be obtained by using (20) for each l. Table 2 exhibits the results of assessing the robust performance of the method for assisting the MF attacks. In addition, Figure 11 displays the similarity between $\mathbf{w}$ and $\mathbf{w}_{\text{MF},l}'$ for each l.

Observing Figures 10 and 11, these three Chinese words can be specifically identified in most cases under consideration. Consequently, the proposed method manifestly possesses the adaptive and robust capabilities against two kinds of filtering attacks above.

5.3. *Robustness to MP3 compression/decompression*

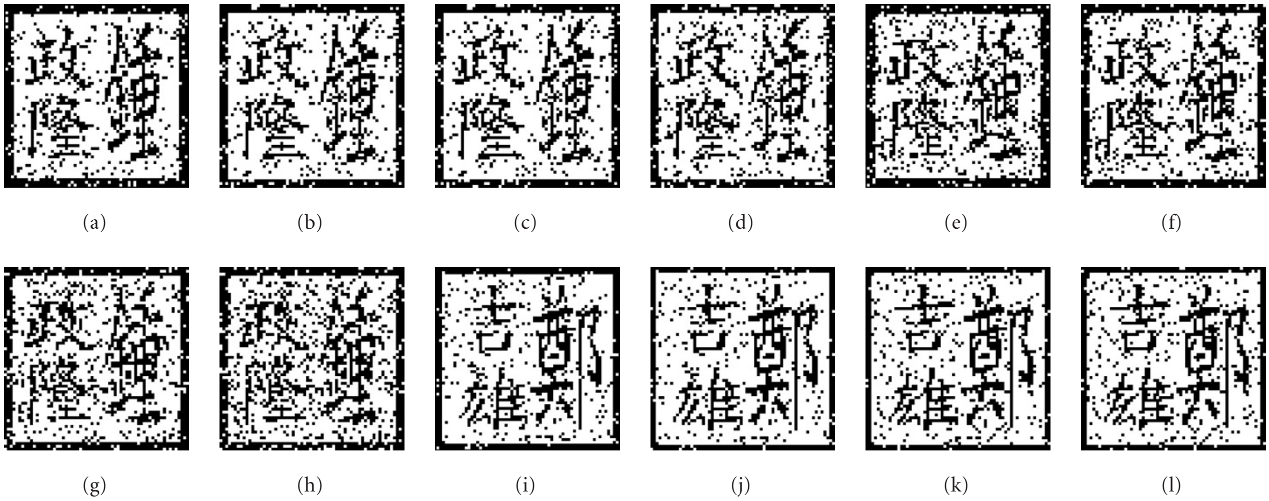
The adaptive and robust capabilities against the compression/decompression attack are tested by using MP3 compression/decompression. Let $\hat{X}_{\text{MP3},m}$ ($\hat{\Psi}_{\text{MP3},m}$) represent an MP3-and-watermarked audio. That is, a watermarked audio $\hat{X}$ is further manipulated by MP3 compression/decompression



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

(g) (h) (i) (j) (k) (l)

FIGURE 10: (a), (b), (c), and (d) show four estimated watermarks $\mathbf{w}'_{\mathrm{filter},m}$, extracted from four filtered-and-watermarked audio $\widehat{X}_{\mathrm{filter},m}$, for $m = 16, 18, 20$, and 22, respectively, in the case of testing the first audio. (e), (f), (g), and (h) show four estimated watermarks in the case of testing the second audio. (i), (j), (k), and (l) exhibit four estimated watermarks in the case of testing the third audio.



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

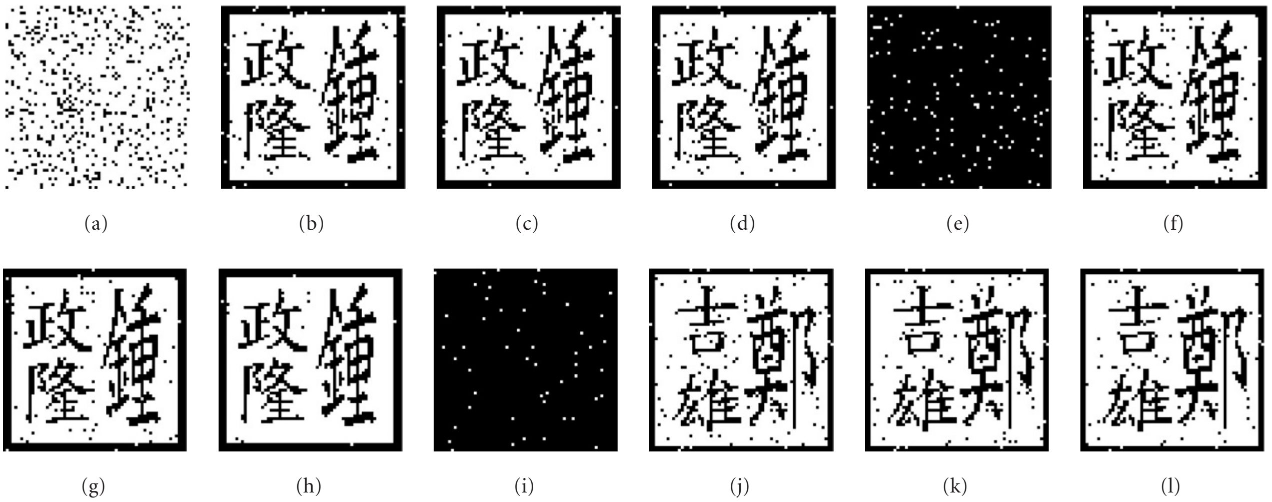
(g) (h) (i) (j) (k) (l)

FIGURE 11: (a), (b), (c), and (d) show four estimated watermarks $\mathbf{w}'_{\mathrm{MF},l}$, extracted from four MF-and-watermarked audio $\widehat{X}_{\mathrm{MF},l}$ for $l = 5, 7, 9$, and 11, respectively, in the case of testing the first audio. (e), (f), (g), and (h) show four estimated watermarks in the case of testing the second audio. (i), (j), (k), and (l) exhibit four estimated watermarks in the case of testing the third audio.

with a compression rate of m kbps. Four cases, for $m = 64, 96, 128$, and 160, are investigated in this experiment. Using the similar way stated in Section 5.2, a set of testing patterns, denoted by $\Upsilon_{\mathrm{MP3},m}$, is obtained from the watermarked audio $\widehat{\varphi}_{\mathrm{MP3},m}$. Then, $\Upsilon_{\mathrm{MP3},m}$ is fed into the TNN, and the estimated watermark $\mathbf{w}'_{\mathrm{MP3},m}$ is obtained by using (20). Table 3 shows the results of investigating the robust performance of the method for assisting the MP3 attacks. To assess the similarity between $\mathbf{w}$ and $\mathbf{w}'_{\mathrm{MP3},m}$ from Figure 12, these three Chinese words can be patently recognized. However, the method breaks down in the case of examining the third audio when m is less than or equal to 64.

5.4. Robustness to multiple attacks

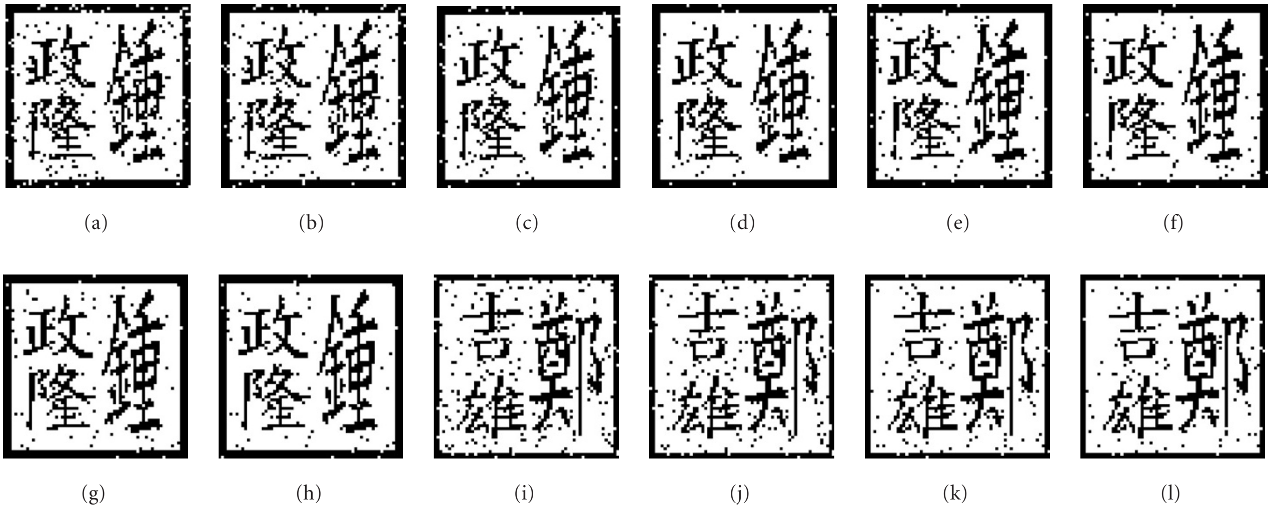
First, a watermarked audio is filtered by a filter, and then, the filtered-and-watermarked audio is further manipulated by the MP3 compression/decompression. Let $\widehat{X}^{\mathrm{Filter},m_1}_{\mathrm{MP3},m_2}$ $(\widehat{\Psi}^{\mathrm{Filter},m_1}_{\mathrm{MP3},m_2})$ be referred to as a watermarked audio $\widehat{X}$ that is further manipulated by a filter with cutting-off frequency in m_1 kHz and MP3 compression/decompression with



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

(g) (h) (i) (j) (k) (l)

FIGURE 12: (a), (b), (c), and (d) show four estimated watermarks $\mathbf{w}'_{\mathrm{MP3},m}$, extracted from four MP3-and-watermarked audio $\hat{X}_{\mathrm{MP3},m}$ for $m = 64, 96, 128$, and 160, respectively, in the case of testing the first audio. (e), (f), (g), and (h) show four estimated watermarks in the case of testing the second audio. (i), (j), (k), and (l) exhibit four estimated watermarks in the case of testing the third audio.



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

(g) (h) (i) (j) (k) (l)

FIGURE 13: (a), (b), (c), and (d) show four estimated watermarks $\mathbf{w}'_{m_1,m_2}$, extracted from $\hat{X}^{\mathrm{Filter},m_1}_{\mathrm{MP3},m_2}$ for $(m_1, m_2) = (18, 96), (18, 128), (20, 96)$, and $(20, 128)$, respectively, in the case of testing the first audio. (e), (f), (g), and (h) show four estimated watermarks in the case of testing the second audio. (i), (j), (k), and (l) exhibit four estimated watermarks in the case of testing the third audio.

a compression rate of m_2 kbps. Four different cases, for $(m_1, m_2) = (18, 96), (18, 128), (20, 96)$, and $(20, 128)$, are examined in this experiment. Using a similar way as stated in Section 5.2, a set of testing patterns, denoted by $\hat{\Upsilon}^{\mathrm{Filter},m_1}_{\mathrm{MP3},m_2}$, can be obtained from the watermarked audio $\hat{\varphi}^{\mathrm{Filter},m_1}_{\mathrm{MP3},m_2}$. Then, $\hat{\Upsilon}^{\mathrm{Filter},m_1}_{\mathrm{MP3},m_2}$ is fed into the TNN and the estimated watermark $\mathbf{w}'_{m_1,m_2}$ is obtained by using (20). Table 4 shows the results of assessing the robust performance of the method for assisting the filtering-and-MP3 attacks. The similarity between $\mathbf{w}$ and $\mathbf{w}'_{m_1,m_2}$ is exhibited in Figure 13 for the assessment of using the visual perception.

Another kind of multiple attacks is referred to as an MF-and-MP3 attack if the filter, used in the case of the filtering-and-MP3 attack, is replaced by an MF. Let $\hat{X}^{\mathrm{MF},l}_{\mathrm{MP3},m}$ $(\hat{\Psi}^{\mathrm{MF},l}_{\mathrm{MP3},m})$ stand for a watermarked audio $\hat{X}$ that is further manipulated by an MF with window length l and then by MP3 compression/decompression with a compression rate of m kbps. Four cases, for $(l, m) = (7, 96), (7, 128), (9, 96)$, and $(9, 128)$, are investigated in this experiment. Table 5 shows the results of assessing the robust performance of the method for assisting the filtering-and-MP3 attacks. Figure 14 displays the similarity between $\mathbf{w}$ and $\mathbf{w}'_{l,m}$. In these two multiple-attacks cases,

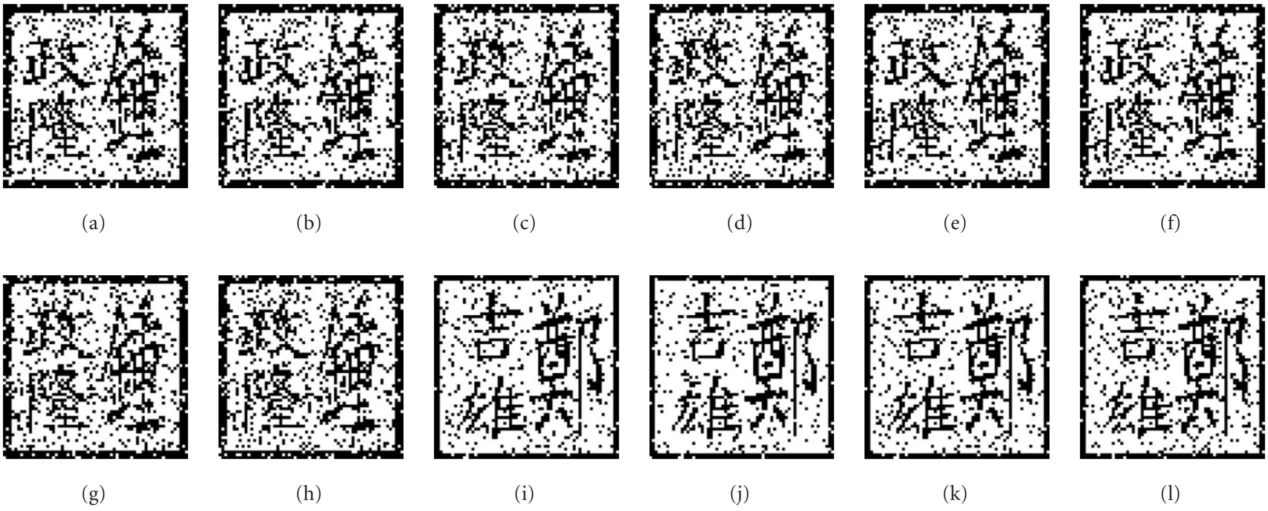


FIGURE 14: (a), (b), (c), and (d) show four estimated watermarks $\mathbf{w}'_{l,m}$, extracted from $\hat{X}^{\text{MF},l}_{\text{MP3},m}$, respectively, for $(l, m) = (7, 96), (7, 128), (9, 96)$, and $(9, 128)$ in the case of testing the first audio. (e), (f), (g), and (h) show four estimated watermarks in the case of testing the second audio. (i), (j), (k), and (l) exhibit four estimated watermarks in the case of testing the third audio.

TABLE 3: The *DR* values and the number of correct pixels in $\mathbf{w}'_{\text{MP3},m}$ for $m = 64, 96, 128$, and 160 when these three audio are examined.

The first audio is examined

m	*DR*	# of correct pixels in $\mathbf{w}'_{\text{MP3},m}$
64	0.242676	2545
96	0.958008	4010
128	0.964844	4024
160	0.964355	4023

The second audio is examined

m	*DR*	# of correct pixels in $\mathbf{w}'_{\text{MP3},m}$
64	−0.297363	1439
96	0.952637	3999
128	0.968262	4031
160	0.993164	4082

The third audio is examined

m	*DR*	# of correct pixels in $\mathbf{w}'_{\text{MP3},m}$
64	−0.434570	1158
96	0.939941	3973
128	0.949707	3993
160	0.959473	4013

TABLE 4: The *DR* values and the number of correct pixels in $\mathbf{w}'_{m_1,m_2}$ for $(m_1, m_2) = (18, 96), (18, 128), (20, 96)$, and $(20, 128)$ when these three audio are examined.

The first audio is examined

(m_1, m_2)	*DR*	# of correct pixels in $\mathbf{w}'_{m_1,m_2}$
(18, 96)	0.890625	3872
(18, 128)	0.910156	3912
(20, 96)	0.938477	3970
(20, 128)	0.956543	4007

The second audio is examined

(m_1, m_2)	*DR*	# of correct pixels in $\mathbf{w}'_{m_1,m_2}$
(18, 96)	0.945801	3985
(18, 128)	0.955566	4005
(20, 96)	0.954590	4003
(20, 128)	0.969238	4033

The third audio is examined

(m_1, m_2)	*DR*	# of correct pixels in $\mathbf{w}'_{m_1,m_2}$
(18, 96)	0.887207	3865
(18, 128)	0.902344	3896
(20, 96)	0.930176	3953
(20, 128)	0.943359	3980

these three Chinese words can be discerned clearly in Figures 13 and 14.

The results above illustrate that the proposed method significantly possesses the adaptive and robust capabilities to effectively resist these five common attacks for protecting the copyright of digital audio.

Table 5: The *DR* values and the number of correct pixels in $\mathbf{w}'_{l,m}$ for $(l,m) = (7,96)$, $(7,128)$, $(9,96)$, and $(9,128)$ when these three audio are examined.

The first audio is examined		
(l,m)	*DR*	# of correct pixels in $\mathbf{w}'_{l,m}$
(7, 96)	0.800293	3687
(7, 128)	0.799316	3685
(9, 96)	0.800293	3687
(9, 128)	0.799316	3685
The second audio is examined		
(l,m)	*DR*	# of correct pixels in $\mathbf{w}'_{l,m}$
(7, 96)	0.744629	3573
(7, 128)	0.747559	3579
(9, 96)	0.713867	3510
(9, 128)	0.707520	3497
The third audio is examined		
(l,m)	*DR*	# of correct pixels in $\mathbf{w}'_{l,m}$
(7, 96)	0.822266	3732
(7, 128)	0.841797	3772
(9, 96)	0.822266	3732
(9, 128)	0.797363	3681

6. CONCLUSIONS

In this paper, the techniques of neural networks have successfully been incorporated into audio watermarking to develop a novel watermarking for digital audio. The proposed method has effectively employed an NN for memorizing the relationships between the original audio and the watermarked audio. Because the NN possesses the memorized and the adaptive (generalization) capabilities, the method can extract watermarks without original audio in contrast to the other proposed methods, such as a scheme proposed in [4], requiring the original audio for the watermark extraction. Moreover, the method makes the watermark imperceptible via exploiting the audio-masking characteristics of the HAS. Finally, the experimental results are exhibited to illustrate that the method significantly possesses robustness to be immune against common attacks for the copyright protection of digital audio.

ACKNOWLEDGMENTS

Tsai and Yu, wish to acknowledge their gratitude to the National Science Council (NSC), Taiwan, for its partial financial support, with Grant number NSC 89-2218-E-150-010 and NSC 89-2218-E-194-010, respectively. Gratitude is extended to the anonymous reviewers for their valuable comments and professional contributions to the improvement of this work.

REFERENCES

[1] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal*, vol. 35, no. 3-4, pp. 313–336, 1996.

[2] X.-G. Xia, C. G. Boncelet, and G. R. Arce, "A multiresolution watermark for digital images," in *Proc. IEEE International Conference on Image Processing*, vol. 1, pp. 548–551, Santa Barbara, Calif, USA, July 1997.

[3] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1079–1107, 1999.

[4] M. D. Swanson, B. Zhu, A. Tewfik, and L. Boney, "Robust audio watermarking using perceptual masking," *Signal Processing*, vol. 66, no. 3, pp. 337–355, 1998.

[5] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proceedings of the IEEE*, vol. 86, no. 6, pp. 1064–1087, 1998.

[6] W. Zeng and B. Liu, "On resolving rightful ownerships of digital images by invisible watermarks," in *Proc. IEEE International Conference on Image Processing*, vol. 1, pp. 552–555, Santa Barbara, Calif, USA, July 1997.

[7] P.-T. Yu, H.-H. Tsai, and J.-S. Lin, "Digital watermarking based on neural networks for color images," *Signal Processing*, vol. 81, no. 3, pp. 663–671, 2001.

[8] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol. 6, no. 12, pp. 1673–1687, 1997.

[9] P. Noll, "Wideband speech and audio coding," *IEEE Communication Magazine*, vol. 26, no. 11, pp. 34–44, 1993.

[10] ISO/IEC IS 11172 (MPEG), "Information technology—coding of moving pictures and associated audio for digital storage up to about 1.5 Mbits/s," 1993.

[11] P. Noll, "MPEG digital audio coding," *IEEE Signal Processing Magazine*, vol. 145, pp. 59–81, November 1997.

[12] D. Pan, "A tutorial on mpeg audio compression," *IEEE Multimedia Journal*, vol. 2, no. 2, pp. 60–74, 1995.

[13] A. Shamir, "On the generation of cryptographically strong pseudo-random sequences," in *8th International Colloquium on Automata, Languages, and Programming*, vol. 62 of *Lecture Notes in Computer Science*, Spring-Verlag, Berlin, 1981.

[14] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Company, New York, NY, USA, 1995.

[15] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters—Principles and Applications*, Kluwer Academic, Boston, Mass, USA, 1990.

[16] P.-T. Yu and R.-C. Chen, "Fuzzy stack filters—their definitions, fundamental properties, and application in image processing," *IEEE Trans. Image Processing*, vol. 5, no. 6, pp. 838–854, 1996.

Hung-Hsu Tasi received the B.S. and M.S. degrees in applied mathematics from the National Chung Hsing University, Taichung, Taiwan, in 1986 and 1988, respectively, and the Ph.D. degree in computer science and information engineering from National Chung Cheng University, Chiayi, Taiwan, in 1999. He has been with the Department of Information Management at National Huwei Institute of Technology, Yunlin, Taiwan, where he is currently an Associate Professor. His research interests include soft computing, digital watermarking, intelligent filter design, data mining, and web programming.

Ji-Shiung Cheng received the B.S. degree in computer science and engineering from Tatung University, Taipei, Taiwan, in 1998, and the M.S. degree in computer science and information engineering from National Chung Cheng University, Chiayi, Taiwan, in 2000. He currently serves in the AIPTEK International, Inc. His research interests include neural networks, fuzzy systems, and digital watermarking.

Pao-Ta Yu received the B.S. degree in mathematics from the National Taiwan Normal University, Taipei, Taiwan, in 1979, the M.S. degree in computer science from the National Taiwan University, Taipei, Taiwan, in 1985, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, Indiana, in 1989. Since 1990, he has been with the Department of Computer Science and Information Engineering at the National Chung Cheng University, Chiayi, Taiwan, where he is currently a Professor. His research interests include neural networks and fuzzy systems, nonlinear filter design, intelligent networks, XML technology, and e-learning.