Audio Classification and Categorization Based on Wavelets and Support Vector Machine

Chien-Chang Lin, Shi-Huang Chen, *Member, IEEE*, Trieu-Kien Truong, *Fellow, IEEE*, and Yukon Chang, *Member, IEEE*

Abstract—**In this paper, an improved audio classification and categorization technique is presented. This technique makes use of wavelets and support vector machines (SVMs) to accurately classify and categorize audio data. When a query audio is given, wavelets are first applied to extract acoustical features such as subband power and pitch information. Then, the proposed method uses a bottom-up SVM over these acoustical features and additional parameters, such as frequency cepstral coefficients, to accomplish audio classification and categorization. A public audio database (Muscle Fish), which consists of 410 sounds in 16 classes, is used to evaluate the performances of the proposed method against other similar schemes. Experimental results show that the classification errors are reduced from 16 (8.1%) to six (3.0%), and the categorization accuracy of a given audio sound can achieve 100% in the Top 2 matches.**

Index Terms—**Audio categorization, audio classification, support vector machine (SVM), wavelets.**

I. Introduction

In the age of digital information, audio data has become an important part in many modern computer applications. A typical multimedia database often contains millions of audio clips, including environmental sounds, machine noise, music, animal sounds, speech sounds, and other nonspeech utterances. The need to automatically recognize to which class an audio sound belongs makes audio classification and categorization an emerging and important research area. In general, audio classification and categorization can be performed in two steps. In the first step, an audio sound is reduced to a small set of parameters using various feature extraction techniques, and in the second step, classification or categorization algorithms ranging from simple Euclidean distance methods to sophisticated statistical techniques are carried out over these parameters. The efficacy of an audio classification or categorization system depends on the ability to capture proper audio features and to accurately classify each feature set corresponding to its own class.

Many researchers [1]–[4] have studied or proposed methods capable of extracting audio features from a sound. In one of the most notable previous works, Wold *et al.* [1] presented a system called "Muscle Fish," in which statistical values including means, variances, and autocorrelations of several time- and frequency-domain measurements are used to represent various perceptual features such as loudness, brightness, bandwidth, pitch, and timbre. Recently, Li [2] presented a method for content-based audio classification and retrieval. The features selected in [2] are combinations of mel-frequency cepstral features (MFCCs) and other perceptual features including brightness, bandwidth, subband energies, and the shape of the frequency spectrum features. In addition, he presented a new pattern classification method called the nearest feature line (NFL), which contrasts with the nearest neighbor (NN) classification. The rationale of the NFL is based on the following considerations: A sound corresponds to a feature set in the feature space. When one sound changes continuously to another in some way, it draws a trajectory linking the corresponding feature sets in the feature space. The trajectories due to changes between prototype sounds of the same class constitute a subspace representing that class. An audio sound belonging to this class should be close to the subspace but may not be so at the original prototypes. His experiments show that the NFL method performs better than the nearest neighbor (NN), nearest center (NC), and the 5-NN classifiers, resulting in 40 classification errors when classifying 198 sounds selected from the Muscle Fish database [21].

Several statistical techniques, such as neural networks, hidden Markov models (HMMs), or support vector machines (SVMs), can also be applied to these parameters for audio classification and categorization. Among these techniques, SVMs, which were proposed by Vapnik [8], have been regarded as a new learning algorithm for various applications, such as audio classification [5]–[7] and pattern recognition [11]–[15]. By using SVMs instead of the NFL method, Guo and Li [5] managed to significantly improve the previous work [2] on classification performance. They achieved as few as 16 classification errors in classifying 198 sounds into 16 classes, or an equivalent error rate of 8.1%.

This paper adopts Guo and Li's method [5] and creates an improved audio classification and categorization system by incorporating additional wavelet functions and a bottom-up SVM. The improvement can be attributed to better feature extraction and SVM construction. Wavelets [16], [17] are a widely used technique that have also been applied to speech and audio feature extraction. In contrast with conventional methods using the Fourier transform, the pitch information [18] and subband power [19] extracted from the wavelet domain can improve performance, as shown in various experimental results. To increase

Manuscript received July 2, 2004; revised January 18, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Roger K. Moore.

C.-C. Lin, T.-K. Truong, and Y. Chang are with the Department of Information Engineering, I-Shou University, Kaohsiung 840, Taiwan, R.O.C. (e-mail: 9003003d@stmail.isu.edu.tw; truong@isu.edu.tw; yukon@isu.edu.tw).

S.-H. Chen is with the Department of Computer Science and Information Engineering, Shu-Te University, Kaohsiung 824, Taiwan, R.O.C. (e-mail: shchen@mail.stu.edu.tw).

Digital Object Identifier 10.1109/TSA.2005.851880

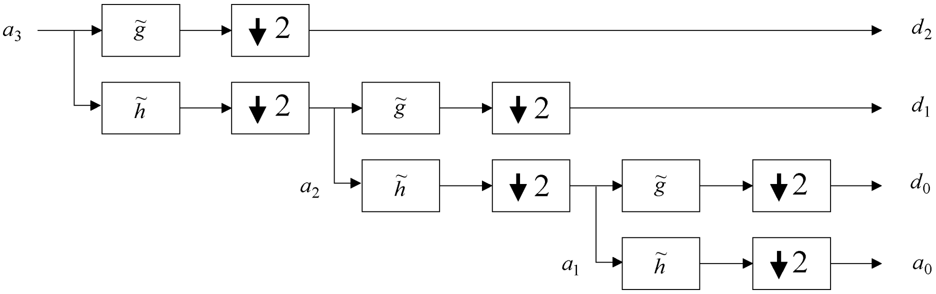

Fig. 1. Three-level wavelet transform.

the discriminability of sounds, the overlap size between windowed frames of the proposed feature extraction algorithm is redesigned, and a normalization process is carried out after feature extraction. It is worth noting that wavelets also help the proposed method in reducing the size of the feature vector from $(18+2L)$ [5] to $(14+2L)$. In addition, this paper modifies two SVM training parameters–the upper bound C and the variance σ^2 of the exponential radial basis function (ERBF)–to improve the accuracy of classification. Experimental results show that our method reduces the number of classification errors in [5] from 16 (an error rate of 8.1%) to 6 (3.0%).

Finally, this paper proposes a bottom-up SVM categorization strategy that uses an iterative procedure to match a given audio to progressively larger subsets (or categories) of classes. It is shown in experimental results that the categorization accuracy of a given audio sound can achieve 97.0% and 100% in the Top 1 and Top 2 matches, respectively.

The rest of the paper is organized as follows. In Section II, detailed descriptions of feature extraction and wavelets are given. Then, the SVM for the two-class classification problem is described in Section III. Section IV illustrates various experimental results and compares them with existing methods. Finally, conclusions are given in Section V.

II. Feature Extraction

As mentioned in Section I, several features are extracted from each input sound to facilitate further classification or categorization. In this paper, a $(14 + 2L)$-dimensional feature vector is proposed for audio classification and categorization. The $(14 + 2L)$-dimensional feature vector is constructed from perceptual features and frequency cepstral coefficients. Detailed preprocessing and feature extraction processes are described as follows.

A. Preprocessing

The original audio sounds in Muscle Fish [21] were sampled at 8000 Hz with 16-bit resolution. Each sound is divided into frames. The frame length is 256 samples (32 ms) with a 192-sample (75%) overlap between adjacent frames. Due to radiation effects of the sound from lips, high-frequency components have relatively low amplitude, which will influence the capture of the features at the high end of the spectrum. One simple solution [20] is to augment the energy of the high-frequency spectrum. This procedure can be implemented via a pre-emphasizing filter that is defined as

$$s'_n = s_n - 0.96 \times s_{n-1}, \quad \text{for} \quad n = 1, \cdots, 255 \qquad (1)$$

where s_n is the nth sample of the frame s and $s'_0 = s_0$. Then, the pre-emphasized frame is Hamming-windowed by

$$s^h_i = s'_i * h_i, \quad \text{for} \quad i = 0, \cdots, 255 \qquad (2)$$

with $h_i = 0.54 - 0.46 \times \cos(2\pi i/255)$. The pre-processed frame will be detected as a nonsilent frame for feature extraction if the total power is large, i.e.,

$$\sum_{i=0}^{255} \left(s^h_i\right)^2 > 400^2 \qquad (3)$$

where 400 is an experience value [2], [5].

B. Feature Extraction From Nonsilent Frames

The Fourier transform is the most popular method that maps audio signals from the time domain to the frequency domain. The wavelet transform [16], [17] is another choice in many previous works. It follows from [18] and [19] that a three-level wavelet transform, as shown in Fig. 1, gives a better performance for an audio signal with a sampling rate of 8000 Hz. Hence, this paper applies both Fourier and wavelet transforms to increase the ability to capture proper audio features.

1) Brief Introduction to the Wavelet Transform: The wavelet transform discussed here is implemented via a filterbank structure. A fast discrete algorithm proposed by Mallat [16] is shown in Fig. 1, where $\tilde{h}(n)$ and $\tilde{g}(n)$ are the analysis lowpass and highpass filters. In addition, the symbol $\downarrow 2$ denotes the downsampling by 2. Let $\{a_3(n)\}_{n\in Z}$ be the input to the analysis filterbank. Then, the outputs of the analysis filterbank are given by

$$a_i(k) = \sum_n \tilde{h}(n-2k)a_{i+1}(n) \qquad (4)$$

and

$$d_i(k) = \sum_n \tilde{g}(n-2k)a_{i+1}(n) \qquad (5)$$

TABLE I
LIST OF EXTRACTED FEATURES

Feature		Type of transforms	Number of features
Perceptual feature	Subband power P_j	Wavelet	3
	Pitch frequency f_p	Wavelet	1
	Brightness ω_c	Fourier	1
	Bandwidth B	Fourier	1
Frequency cepstral coefficient (FCC) c_n		Fourier	L

where $a_i(k)$ and $d_i(k)$ are called the approximation and detail coefficients of the wavelet decomposition of $a_{i+1}(n)$, respectively. In this paper, the calculation of the wavelet transform is implemented using the length-8 orthogonal wavelet introduced by Daubechies [16], [17].

2) Perceptual and FCC Feature Extractions: There are in total $6+L$ features, shown in Table I, derived from wavelet coefficients and fast Fourier transform (FFT) coefficients $F(u)$ of each nonsilent frame s_i^h. The $6+L$ features contain perceptual features and an L-order frequency cepstral coefficient (FCC). The detailed extraction process of each feature is given in the following.

1) Subband Power P_j: Three sections of subband power calculated in the wavelet domain are used in this paper. Let ω be the half sampling frequency. Then, the subband intervals are $[0, \omega/8]$, $[\omega/8, \omega/4]$, and $[\omega/4, \omega/2]$, corresponding to the approximation and detail coefficients $a_0(k)$, $d_0(k)$, and $d_1(k)$ of a given audio sound $a_3(k)$, respectively. The subband power is calculated by $P_j = \sum_k z_j^2(k)$, where $z_j(k)$ is the corresponding approximation or detail coefficients of subband j.
2) Pitch Frequency f_p: A noise-robust wavelet-based pitch detection method is used to extract the pitch frequency [18]. The first stage of the pitch-detection method is to apply the wavelet transform with aliasing compensation to decompose the input sound into three subbands, as shown in Fig. 1. Then, this method makes use of a modified spatial correlation function, which was determined from the approximation signals obtained in the previous stage to extract the pitch frequency. The experiments in [18] show that this algorithm is capable of outperforming other time-, frequency-, and wavelet-domain pitch detection algorithms.
3) Brightness ω_c: The brightness is the frequency centroid of the Fourier transform and is computed as $\omega_c = \int_0^\omega u|F(u)|^2 du / \int_0^\omega |F(u)|^2 du$.
4) Bandwidth B: It is the square root of the power-weighted average of the squared difference between the spectral components and the frequency centroid, i.e., $B = \sqrt{\int_0^\omega (u-\omega_c)^2 |F(u)|^2 du / \int_0^\omega |F(u)|^2 du}$.
5) Frequency Cepstral Coefficient (FCC): The L-order coefficients are calculated as $c_n = \sqrt{2/256}\sum_{u=0}^{255} (\log_{10} F(u))\cos(n(u-0.5)\pi/256)$, where $n = 1, 2, \cdots, L$.

C. Feature Vector Formation

The mean and the standard deviation of each of the $6+L$ features are computed to result in a $(12+2L)$-dimensional feature vector. Furthermore, the pitch ratio (number of pitched frames/total number of frames) and the silence ratio (number of silent frames/total number of frames) are added to the above feature vector to form a $(14+2L)$-dimensional feature vector.

D. Normalization for Training and Testing

An experimental audio feature set is partitioned into a training set T and a testing set E. The detailed partition process is given in Section IV. The training set T can be defined as an $n_T \times (14+2L)$ array of elements $T(i,j)$, where n_T is the number of training vectors, and the subscripts i and j denote the row and column positions, respectively.

First, each of the preceding 14 columns is shifted by $T'(i,j) = (T(i,j) - \mu_j)/\sigma_j$, where $\mu_j = \sum_i T(i,j)/n_T$ and $\sigma_j = \sum_i (T(i,j) - \mu_j)^2/n_T$ are the mean and standard deviation of column j. Second, the normalization is done for the other $2L$ columns by computing $T^N(i,j) = T'(i,j)/m_j$, where m_j is the maximum of the absolute values in column j. Thus, each feature will have similar weightings after the normalization process. Finally, the values μ_j, σ_i, and m_i computed from the training set are also used to prepare the testing set, i.e., $E'(i,j) = (E(i,j) - \mu_j)/\sigma_j$ and $E^N(i,j) = E'(i,j)/m_j$. These normalized perceptual cepstral coefficients $T^N(i,j)$ and $E^N(i,j)$, which are abbreviated NPC-L, are used for training and testing in SVM, respectively.

III. SVMs

SVMs [5]–[15] have recently been proposed as popular tools for learning from experimental data. The reason is that SVMs are much more effective than other conventional nonparametric classifiers (e.g., the RBF neural networks, nearest neighbor (NN), nearest center (NC), and the k-NN classifier [12]) in terms of classification accuracy, computational time, and stability to parameter setting. They also prove to be more effective than the traditional pattern recognition approaches based on the combination of a feature selection procedure and a conventional classifier [2]–[4]. SVMs use a known kernel function to define a hyperplane in order to separate given points into two predefined classes. An improved SVM called the soft-margin SVM can tolerate minor misclassifications [7]–[9]. It is considered to be more suitable for classification and, therefore, is used in this paper.

A. Introduction to SVMs

Let $x_i \in X \subseteq R^n$ and $y_i \in Y = \{1, -1\}$ be the input vector and the target variable, respectively, where R^n denotes the n-dimensional real space. Suppose a training set $S = \{(x_1, y_1), \cdots, (x_l, y_l)\}_{i=1}^{l} \subseteq (X \times Y)^l$ and a kernel function $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ on $X \times X$ is given, where $\langle \cdot, \cdot \rangle$ denotes the inner product, and ϕ maps the input space X to another high-dimensional feature space F. With suitably chosen ϕ, the given nonlinearly separable samples S may be linearly separated in F, as shown in Fig. 2.

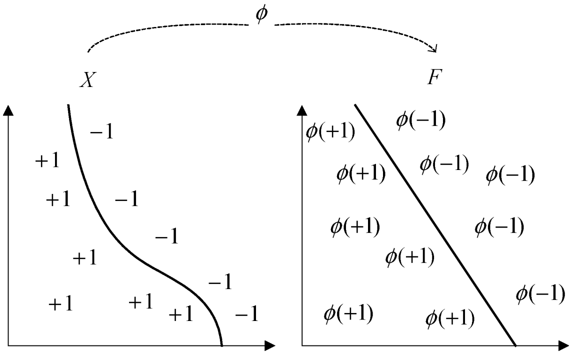

Fig. 2. Feature map can simplify the classification task.

Many hyperplanes can achieve the above separation purpose, but our goal is to find the one that maximizes the margin (the minimal distance from the hyperplane to each points). A hyperplane, which is denoted by $(w, b) \in R^n \times R$, consists of all x satisfying $\langle w, x\rangle + b = 0$. The problem thus can be formed as

$$\begin{cases} \text{minimize} & \frac{1}{2}\|w\|^2 \\ \text{subject to} & y_i\left(\langle w, x_i\rangle + b\right) \geq 1. \end{cases} \tag{6}$$

The solution to the optimization problem of SVMs is given by the saddle point of the Lagrange function. Let C be the upper bound of the Lagrange multipliers α_i, and then, (6) can be formulated as

$$L(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle x_i, x_j\rangle \tag{7}$$

with constraints $\sum_{i=1}^{l} \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$.

Suppose that α_i^* maximizes (7); then, the solution is given by $w = \sum_{i=1}^{l} \alpha_i^* y_i x_i$ and $b = y_k - \sum_{i=1}^{l} \alpha_i^* y_i \langle x_i, x_k\rangle$ for any k such that $0 < \alpha_k^* < C$. The optimal discriminant function can be derived as $H(x) = \langle w, x\rangle + b = \sum_{i=1}^{l} \alpha_i^* y_i \langle x_i, x\rangle + b$. Accordingly the decision function $f(x) = sign(H(x))$ is the classifier that is termed the optimal separating hyperplane.

The soft-margin SVM [7]–[10], which includes slack variables $\xi_i \geq 0$, is proposed to solve nonseparable problems. The slack variables $\xi_i = \max(0, \gamma - y_i(\langle w, x_i\rangle + b))$, which are shown in Fig. 3, measure the amount by which the training set fails to have margin γ and take into account any misclassification of the training data. Consequently, the training process tolerates some misclassified points and is suitable in most classification cases.

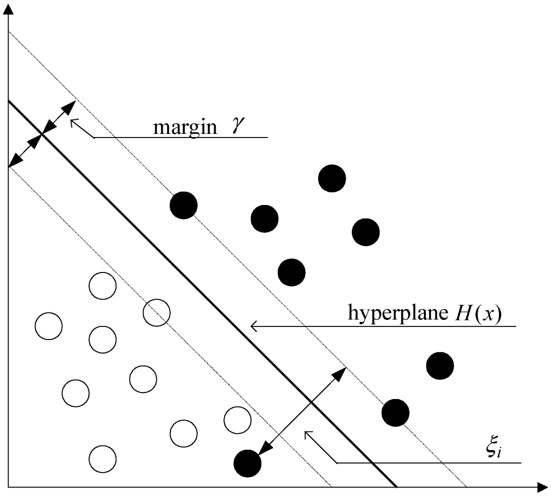

Fig. 3. Margin and the slack variable for a classification problem.

TABLE II
COMPARISON AMONG FOUR SCHEMES

Process Scheme	Total Training Complexity	Total Testing Times	Result Combination
one-against-one	$\frac{c(c-1)}{2} T(n,n)$	$\frac{c(c-1)}{2}$	Vote / Statistic
one-against-all	$c\,T(n,(c-1)n)$	$c-1$	
top-down binary tree	$\sum_{i=1}^{\log_2 c} 2^{i-1} T(\frac{cn}{2^i}, \frac{cn}{2^i})$	$\log_2 c$	No requirement
bottom-up binary tree	$\frac{c(c-1)}{2} T(n,n)$	$c-1$	

* Each of the c classes contains n members, where c is a number of power of 2.

*$T(U,V)$: The training complexity with U vectors in the plus-class and V vectors in the minus-class.

There are three common kernel functions for the nonlinear feature mapping, which are shown in Fig. 2: 1) ERBF $K(x, \overline{x}) = \exp(-|x - \overline{x}|/2\sigma^2)$, 2) Gaussian function $K(x, \overline{x}) = \exp(-|x - \overline{x}|^2/2\sigma^2)$, where parameter σ^2 is the variance of the Gaussian function, and 3) polynomial function $K(x, \overline{x}) = (\langle x, \overline{x}\rangle + 1)^d$, where parameter d is the degree of the polynomial. Many classification problems are always separable in feature space and are able to obtain better accuracy by using the Gaussian kernel function than the linear and polynomial kernel functions [6], [7], [11], [12]. Recently, Guo and Li [5] proposed the ERBF function as the kernel function and produced vastly improved results. Therefore, the Gaussian and ERBF kernel functions are compared using the same feature sets in this paper.

B. Multicase Classification in SVMs

A typical SVM is a two-class classifier that organizes all training sets into two classes, namely, plus-class $(+1)$ and minus-class (-1). It has to be augmented with other strategies to achieve multicase classification. Four commonly used schemes are given below, and their training and testing complexities are shown in Table II.

1) One-against-one: Classify between each pair of classes.
2) One-against-all: Classify between each class and all other remaining classes.
3) Top-down binary tree: An initial group contains all classes. A recursive process is done to separate and reduce a larger candidate group of classes into a smaller one until the test pattern is assigned to a final class.
4) Bottom-up binary tree: A recursive comparison process is performed between pairs of classes. The class with a shorter distance from the test pattern is retained for further comparison until the test pattern is assigned to a final class.

TABLE III
EXPERIMENTAL RESULTS OF (a) ERBF AND (b) GAUSSIAN KERNEL FUNCTIONS FOR THE PRESELECTED VALUES C AND σ^2, WHERE E_m IS COMPUTED AS THE LEAST VALUE OF ERRORS, AND L_m INDICATES WHICH FCC LEVEL L THE FIRST E_m HAPPENS

(a) ERBF kernel

E_m/L_m		C											
		1	5	10	20	30	40	50	60	70	80	90	100
σ^2	1	43/ 3	38/ 2	38/ 2	38/ 2	38/ 2	38/ 2	38/ 2	38/ 2	38/ 2	38/ 2	38/ 2	38/ 2
	5	40/39	18/11	18/11	18/11	18/11	18/11	18/11	18/11	18/11	18/11	18/11	18/11
	10	41/54	12/51	12/51	12/51	12/51	12/51	12/51	12/51	12/51	12/51	12/51	12/51
	20	60/89	7/54	7/54	7/54	7/54	7/54	7/54	7/54	7/54	7/54	7/54	7/54
	30	80/99	9/84	7/54	7/54	7/54	7/54	7/54	7/54	7/54	7/54	7/54	7/54
	40	91/95	12/86	7/54	7/54	7/54	7/54	7/54	7/54	7/54	7/54	7/54	7/54
	50	97/95	14/85	7/54	7/54	7/54	7/54	7/54	7/54	7/54	7/54	7/54	7/54
	60	102/95	16/82	7/66	6/80	6/80	6/80	6/80	6/80	6/80	6/80	6/80	6/80
	70	110/82	17/98	8/83	6/80	6/80	6/80	6/80	6/80	6/80	6/80	6/80	6/80
	80	114/96	19/84	11/81	6/80	6/80	6/80	6/80	6/80	6/80	6/80	6/80	6/80
	90	123/87	19/88	12/87	6/80	6/80	6/80	6/80	6/80	6/80	6/80	6/80	6/80
	100	127/98	22/90	13/99	6/80	6/80	6/80	6/80	6/80	6/80	6/80	6/80	6/80

(b) Gaussian kernel

E_m/L_m		C											
		1	5	10	20	30	40	50	60	70	80	90	100
σ^2	1	35/ 9	20/11	20/11	19/11	20/ 8	19/11	20/11	20/11	19/11	20/11	20/11	20/11
	5	51/99	19/53	14/53	13/55	13/55	13/55	13/55	13/55	13/55	13/55	13/55	13/55
	10	73/89	20/96	15/96	14/96	14/96	14/96	14/96	14/96	14/96	14/96	14/96	14/96
	20	93/93	30/96	18/96	17/87	17/91	17/58	17/58	17/58	17/58	17/58	17/58	17/58
	30	101/77	35/95	22/96	16/95	16/95	17/91	16/95	16/95	16/95	16/95	16/95	16/95
	40	110/54	48/97	26/96	18/95	16/95	17/91	17/95	16/95	16/95	16/95	16/95	16/95
	50	120/81	58/98	31/96	19/97	18/95	16/95	17/91	17/95	16/95	16/95	16/95	16/95
	60	131/99	64/95	35/95	22/97	18/95	16/95	16/95	17/95	17/95	16/95	16/95	16/95
	70	135/95	72/99	39/96	23/96	19/95	18/95	16/95	17/95	17/95	17/95	16/95	16/95
	80	137/97	78/99	45/98	25/99	21/95	18/95	16/95	17/95	17/95	17/95	17/95	17/95
	90	142/99	86/99	52/98	28/96	22/97	19/96	17/96	16/96	17/96	18/89	18/89	18/96
	100	147/97	89/99	57/99	30/97	24/95	19/96	18/96	18/96	17/95	18/89	18/89	18/89

Even though these schemes have similar classification capability, many previous works [6], [12], [13] show that the bottom-up binary tree scheme has the best performance in terms of classification accuracy and time complexity. The reasons are summarized as the other schemes 1) require additional decision rules, 2) have more training complexities, as listed in Table II, and 3) use mixed features from many classes in the training strategy that will cause disharmony in the training data and, thus, reduce the classification accuracy. Therefore, the bottom-up binary tree scheme is taken to be our multicase classifier.

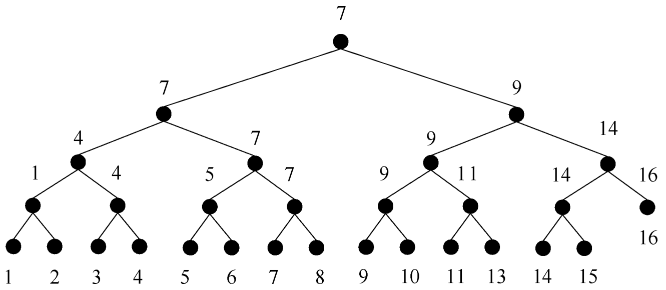

Fig. 4. Reconstructed bottom-up binary tree scheme after node 12 was removed.

C. Audio Categorization of an SVM

A typical SVM with a bottom-up binary tree scheme only achieves multicase classification, in which a single class is selected as the class to which a given audio data belongs. Depending on applications, it is sometimes more suitable to select more than one class as the possible candidates. A method based on this tree scheme can be carried out to rank classes with respect to a given audio by proceeding iteratively. In each round, it removes the winning class number from the root of the tree and then reconstructs a new tree structure. The class removed first is the class to which the given audio is most similar. Conversely, the class number removed last belongs to the class to which the given audio is least similar. Fig. 4 illustrates the reconstructed bottom-up binary tree scheme after node 12 was first removed. The collected numbers determine an ordering of similarities between classes and the tested audio data. The first N classes in the ordering as a group are referred to as the Top N group. For a fixed N, a method by which an audio data can be more correctly classified in the Top N would be considered to be more effective.

IV. EXPERIMENTAL RESULTS

In the following experiments, a public audio database named Muscle Fish [21] is utilized to evaluate both classification and categorization performances. The Muscle Fish database consists of 410 sounds classified into 16 classes. These audio classes include alto-trombone (13), animals (9), bells (7), cello-bowed (47), crowds (4), female (35), laughter (7), machines (11), male

TABLE IV
ERROR RATES (NUMBER OF ERRORS) COMPARISON AMONG PROPOSED AND EXISTING METHODS

Method	Proposed		Guo and Li's [5]	Li's [2]			
Feature Set	NPC-L		PercCeps$L$	PercCeps$L$			
Classifier and Kernel	SVM ERBF $C=30, \sigma^2=60$	SVM Gaussian $C=100, \sigma^2=5$	SVM ERBF $C=200, \sigma^2=6$	NFL	NN	5-NN	NC
$L=5$	11.6 % (23)	12.6 % (25)	12.6 % (25)	12.1 % (24)	17.7 % (35)	21.2 % (42)	43.4 % (86)
$L=8$	9.5 % (19)	10.6 % (21)	8.1 % (16)	9.6 % (19)	13.1 % (26)	22.2 % (44)	38.9 % (77)
$L=60$	3.5 % (7)	7.5 % (15)	10.6 % (21)	13.1 % (26)	16.7 % (32)	21.7 % (43)	32.3 % (64)
$L=80$	3.0 % (6)	9.5 % (19)	10.1 % (20)	12.1 % (24)	15.7 % (31)	20.7 % (41)	32.8 % (65)

* NPC-L = number of errors/199 × 100% * PercCeps$L$ = number of errors/198 × 100%

(17), oboe (32), percussion (99), telephone (17), tubular-bells (20), violin-bowed (45), violin-pizz (40), and water (7), where the numbers in parentheses indicate the number of sound files in the corresponding class. A training set of 211 audio sounds is selected from the database using the following partition procedure: Sort the sounds in alphabetical order of the filenames in each class, and then assign odd-numbered sound files to the training set. The remaining 199 sounds belong to the testing set.

Every training or testing vector consists of a $(14+2L)$-dimensional feature vector that is extracted from each sound and then used in soft-margin SVMs with a bottom-up tree scheme to complete evaluations. Two kernel functions (ERBF and Gaussian kernels) are compared over a range of preselected values of upper bound C and variance σ^2, whereas FCC level L varies from 1 to 99. This creates a total of 144 pairs of C and σ^2 for each of the two kernel functions. For each combination of C and σ^2, E_m is computed as the least value of errors, and L_m indicates at which FCC level L the first E_m happens. As shown in Table III, the accuracy of ERBF kernel is clearly better than that of Gaussian kernel in most settings. Specifically, the best classification result with an ERBF kernel is six misclassifications, which is significantly better that the 13 misclassifications the Gaussian kernel produces. Another interesting observation is that the ERBF kernel function is more stable when C is greater than 20, suggesting that a larger value of C will have no additional benefit. Table IV shows the error rates of the proposed methods along with methods proposed in [2] and [5]. Note that [2] and [5] apply the same feature set to achieve audio classification, but method [5] outperforms methods [2] by using SVM instead of NFL, NN, 5-NN, and NC. It reconfirms the common belief that given the same feature set, the choice of the classifier and the kernel function is important.

As shown in Fig. 5, the proposed method using the ERBF kernel function performs best with $50 \leq L \leq 90$ for the preselected values $C = 30$ and $\sigma^2 = 60$. The best result is only six classification errors when L is between 80 and 82, attaining a high accuracy of 97.0%. In contrast with [5], Table IV shows that the proposed method reduces the number of errors from 16 (8.1%) to 6 (3.0%) by applying feature selection and the normalization process that are improvements from [5]. It follows from the results that the efficacy of an audio classification system depends on the ability to capture proper audio features to help classify each feature set to its own class accurately. Thus, it proves that properly selected features and accurately designed classifiers are equally instrumental to the success of a classification method.

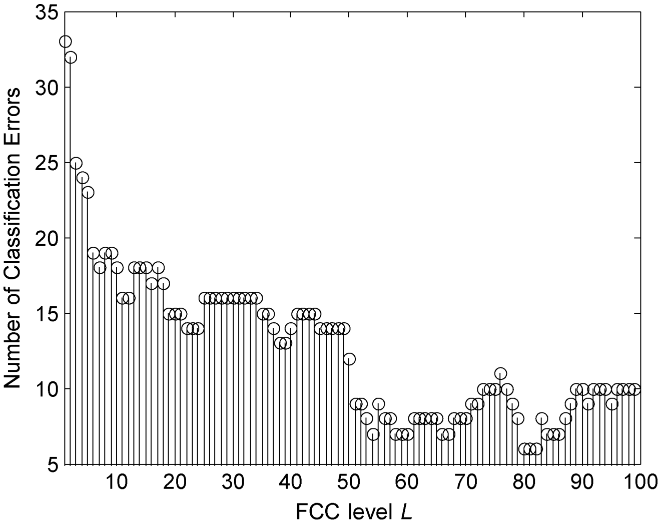

Fig. 5. Classification errors in various FCC levels L using the ERBF kernel function with $C = 30$ and $\sigma^2 = 60$.

For the misclassified audio, the audio categorization method based on the bottom-up binary tree scheme is used for further comparison. Table V shows that the proposed method can achieve 100% accuracy in the Top 2 for most settings of C and σ^2. Furthermore, this high accuracy is achievable by many reasonably sized L. Fig. 6 plots the accuracy rates in Top 1 through Top 3 at $C = 30$ and $\sigma^2 = 60$ as functions of L. It shows that 100% accuracy in the Top 2 and Top 3 is achieved for all $L \geq 29$. Each of the six misclassified testing sounds, which are listed in Table VI, is misplaced because it sounds similar to the training sounds in another class, even to the human ear. The probability of it being misclassified a second time for the same reason is quite small, and the proposed method was able to categorize it correctly in the Top 2 at most FCC level L, upper bound C, and variance σ^2 settings.

V. CONCLUSIONS

Starting with Guo and Li's method, our proposed method reduces the size of the feature set using the wavelet transform instead of the Fourier transform. Even though the conventional Fourier transform is suitable for audio samples with concentrated energy, the wavelet transform is more natural and effec-

TABLE V
CATEGORIZATION ERRORS IN THE TOP 2 USING THE ERBF KERNEL FUNCTION FOR THE PRESELECTED VALUES C AND σ^2, WHERE E_m IS COMPUTED AS THE LEAST VALUE OF ERRORS, AND L_m INDICATES WHICH FCC LEVEL L THE FIRST E_m HAPPENS

E_m/L_m		C											
		1	5	10	20	30	40	50	60	70	80	90	100
σ^2	1	23/ 1	19/ 4	19/ 4	19/ 4	19/ 4	19/ 4	19/ 4	19/ 4	19/ 4	19/ 4	19/ 4	19/ 4
	5	11/29	3/29	3/29	3/29	3/29	3/29	3/29	3/29	3/29	3/29	3/29	3/29
	10	17/37	0/46	0/46	0/46	0/46	0/46	0/46	0/46	0/46	0/46	0/46	0/46
	20	22/98	0/38	0/37	0/37	0/37	0/37	0/37	0/37	0/37	0/37	0/37	0/37
	30	28/96	0/49	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29
	40	37/96	1/77	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29
	50	49/99	3/81	0/40	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29
	60	66/91	3/97	0/46	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29
	70	76/99	4/99	0/50	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29
	80	79/85	5/76	0/68	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29
	90	84/93	5/95	1/80	0/33	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29
	100	90/76	8/90	2/87	0/39	0/29	0/29	0/29	0/29	0/29	0/29	0/29	0/29

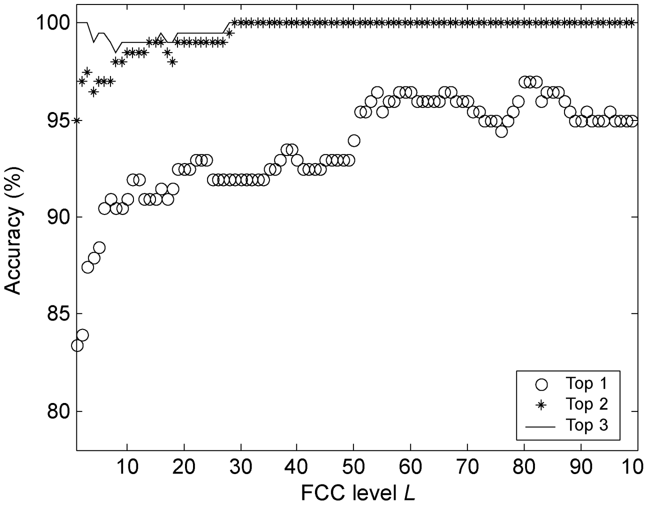

Fig. 6. Categorization accuracy for L from 1 to 99 with $C = 30$ and $\sigma^2 = 60$.

TABLE VI
LIST OF MISCLASSIFIED SOUNDS

Class name	Filename of misclassification	Which class misclassified to
alto-trombone	mcgill_altotrombone_altotrb1	oboe
oboe	mcgill_oboe_oboe28	violin-bowed
female	speech_female_female1_issued	male
water	water_rainAndThunder water_rainInWoods water_runningWater	machines

tive for describing audio characteristics. A great improvement in classification accuracy is achieved from 91.9% to 97.0% in the Top 1. Furthermore, each misclassified sound can be classified 100% correctly into its own class in the Top 2 at most FCC level L, upper bound C, and variance σ^2 settings. Although the Gaussian kernel is widely used to carry out classification, the ERBF kernel turns out to be more suitable for the Muscle Fish audio database in our experiments.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the editors, who gave us many valuable comments and much advice. They also wish to thank the Muscle Fish group for providing the audio database used in this study.

REFERENCES

[1] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search and retrieval of audio," *IEEE Multimedia*, vol. 3, no. 3, pp. 27–36, Jul. 1996.

[2] S. Z. Li, "Content-based audio classification and retrieval using the nearest feature line method," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 5, pp. 619–625, Sep. 2000.

[3] T. Zhang and C.-C. J. Kuo, "Audio content analysis for online audiovisual data segmentation and classification," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 4, pp. 441–457, May 2001.

[4] L. Lu, H.-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 7, pp. 504–516, Oct. 2002.

[5] G. Guo and S. Z. Li, "Content-based audio classification and retrieval by support vector machines," *IEEE Trans. Neural Networks*, vol. 14, no. 1, pp. 209–215, Jan. 2003.

[6] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[7] P. Clarkson and P. J. Moreno, "On the use of support vector machines for phonetic classification," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, vol. 2, Mar. 1999, pp. 585–588.

[8] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[9] V. Kecman, *Learning and Soft Computing*. Cambridge, MA: MIT Press, 2001.

[10] P. Ding, Z. Chen, Y. Liu, and B. Xu, "Asymmetrical support vector machines and applications in speech processing," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, vol. 1, May 2002, pp. I-73–I-76.

[11] A. Ganapathiraju, J. E. Hamaker, and J. Picone, "Applications of support vector machines to speech recognition," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2348–2355, Aug. 2004.

[12] F. Schwenker, "Hierarchical support vector machines for multi-class pattern recognition," in *Proc. IEEE Fourth Int. Conf. Knowledge-Based Intelligent Eng. Syst. Allied Technologies*, vol. 2, Sep. 2000, pp. 561–565.

[13] T. Luo, K. Kramer, D. B. Goldgof, L. O. Hall, S. Samson, A. Remsen, and T. Hopkins, "Recognizing plankton images from the shadow image particle profiling evaluation recorder," *IEEE Trans. Syst., Man Cybern.—B: Cybern.*, vol. 34, no. 4, pp. 1753–1762, Aug. 2004.

[14] H. Fenglei and W. Bingxi, "Text-independent speaker recognition using support vector machine," in *Proc. Int. Conf. Info-Tech Info-Net*, vol. 3, Nov. 2001, pp. 402–407.

[15] M. Pontil and A. Verri, "Support vector machines for 3-D object recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 6, pp. 637–646, Jun. 1998.

[16] S. Mallat, *A Wavelet Tour of Signal Processing*. New York: Academic, 1998.

[17] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms*. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[18] S.-H. Chen and J.-F. Wang, "Noise-robust pitch detection method using wavelet transform with aliasing compensation," *Proc. Inst. Elect. Eng. Vision, Image Signal Process.*, vol. 149, no. 6, pp. 327–334, Dec. 2002.

[19] C.-T. Hsieh, E. Lai, and Y.-C. Wang, "Robust speech features based on wavelet transform with application to speaker identification," *Proc. Inst. Elect. Eng. Vision, Image Signal Process.*, vol. 149, no. 2, pp. 108–114, Apr. 2002.

[20] W. C. Chu, *Speech Coding Algorithms*. New York: Wiley, 2003.

[21] *http://www.musclefish.com/cbrdemo.html* [Online]

Chien-Chang Lin was born in Kaohsiung, Taiwan, R.O.C., in 1976. He received the B.E. degree in electronic engineering from National Kaohsiung University of Applied Sciences in 1999 and the M.E. degree in information engineering from I-Shou University, Kaohsiung, in 2001. He is now working toward the Ph.D. degree in Department of Information Engineering, I-Shou University.

His research interests include audio and video processing.

Shi-Huang Chen (M'05) was born in Tainan, Taiwan, R.O.C., in 1972. He received the B.E. and M.E. degrees from the Kaohsiung Polytechnic Institute, Kaohsiung, Taiwan, in 1994 and 1996, respectively, and the Ph.D. degree from the National Cheng Kung University, Tainan, in 2002, all in electrical engineering.

Between 2001 and 2003, he was a research engineer at the AVXing Inc., Kaohsiung. He also worked as an adjunct lecturer with the Department of Information Engineering, I-Shou University, Kaohsiung, from February 1998 to June 2002. In February 2003, he joined the Shu-Te University, Kaohsiung, where he is now an Assistant Professor with the Department of Computer Science and Information Engineering. His research interests include wavelet transforms, speech/audio processing, video coding, and multimedia communication standards.

Trieu-Kien Truong (M'82–SM'83–F'99) was born in Vietnam on December 4, 1944. He received the B.S. degree from National Cheng Kung University, Tainan, Taiwan, R.O.C., in 1967, the M.S. degree from Washington University, St. Louis, MO, in 1971, and the Ph.D. degree from the University of Southern California, Los Angeles, in 1976, all in electrical engineering.

From 1975 to 1992, he was a Senior Member of Technical Staff (E6) with the Communication System Research Section, Jet Propulsion Laboratory, Pasadena, CA. Currently, he is a Chair Professor and the Dean of Collage of Electrical and Information Engineering, I-Shou University, Kaohsiung, Taiwan. His research interests include error-correcting codes, VLSI architecture design, communication systems, signal processing, and image compression.

Dr. Troung served as an Editor in the Asia area for the *Journal of Visual Communication and Image Representation* and as an Editor in the area of Coding Theory and Techniques for the IEEE TRANSACTIONS ON COMMUNICATIONS.

Yukon Chang (M'82) received the Ph.D. degree in computer science from Pennsylvania State University, University Park, in 1986.

He was an assistant professor of computer science with the State University of New York at Albany from 1986 to 1992. In 1992, he joined the Department of Information Engineering, I-Shou University, Kaohsiung, Taiwan, R.O.C., where he is now an associate professor and the Chair of the Department. He also served as Director of Computer Center from 1993 to 1996 and Director of the Library at I-Shou University from 1998 to 2001. His primary research interest is in multimedia networks.