Applied Acoustics 72 (2011) 841–848

Contents lists available at ScienceDirect

Applied Acoustics

journal homepage: www.elsevier.com/locate/apacoust

Environmental noise classifier using a new set of feature parameters based on pitch range

Buket D. Barkana *, Burak Uzkent

Department of Electrical Engineering, University of Bridgeport, 221 University Ave., Bridgeport, CT, USA

ARTICLE INFO

Article history:
Received 17 February 2010
Received in revised form 11 March 2011
Accepted 12 May 2011
Available online 15 June 2011

Keywords:
Environmental noise classification
Automatic Noise Recognition (ANR)
Feature extraction
SVM
k-Means clustering

ABSTRACT

Automatic Noise Recognition was performed in two stages: (1) feature extraction based on the pitch range, found by analyzing the autocorrelation function and (2) classification using a classifier trained on the extracted features. Since most environmental noise types change their acoustical characteristics over time, we focused on the "pitch range" of the sounds in order to extract features. Two different classifiers, Support Vector Machines (SVM) and *k*-means clustering, were performed and compared using the proposed features. The SVM and *k*-means clustering classifiers achieve recognition rates up to 95.4% and 92.8%, respectively. Although both classifiers provided high accuracy, the SVM-based classifier outperformed the *k*-means clustering classifier by approximately 7.4%.

1. Introduction

Environmental noise recognition is increasingly important in such diverse fields as speech recognition, environmental sound recognition, echo cancelation, active noise control, human–machine interface, and in many other areas. Researchers are interested in environmental noise recognition for several reasons: (1) signal communications often takes place in environments with large background noise (such as engine noise, street noise, and rain, for example); it would be beneficial to know the type of background noise in order to implement better noise reduction algorithms; signal processing technology presently and in the future focuses on two things: process a communications signal and classify and process background noise (the latter most often handled by a separate module or modules included with the communications electronics); and (2) context-aware systems and audio surveillance systems use information of environmental sounds for mobile computers, for portable tele-assistive devices, for security purposes and for other things.

A noise event is defined as an unexpected increase of the acoustic level. Automatic Noise Recognition (ANR) will identify the source of the noise event. For this purpose, different classifiers based on Artificial Neural Networks (ANN), Hidden Markov Models (HMM), Fuzzy Logic systems, Support Vector Machines (SVM), and *k*-means clustering were developed using both traditional and novell feature extractions [1–3].

Feature selection is one of the most important tasks in a classification algorithm. It allows for a low computational load without increasing the misclassification error. The aim is to obtain an efficient and small vector of acoustic features which represent the input pattern for the classification algorithms being trained. Mel Frequency Cepstral Coefficients (MFCCs) are the most common feature representation for non-speech audio recognition [2]. A combination of statistical features and labels describing the energy envelope, harmonicity, pitch contour for each sample [4], zero-crossing rates [5], autocorrelation based features [6], and LPC-cepstral features [7] are other feature representations of noise recognition systems.

In recognition systems, the most popular approach is based on HMM's with Gaussian mixture observation densities. It is shown that HMM based sound recognition systems perform very well on closed-loop tests but their performance degrades significantly on open-loop tests. These systems typically use a representational model based on maximum likelihood decoding and expectation maximization-based training. The SVM have exhibited important advantages over more traditional techniques. For the SVM, few parameters need to be tuned and the optimization problem does

Abbreviations: SVM, Support Vector Machines; ANR, Automatic Noise Recognition; ANN, Artificial Neural Networks; HMM, Hidden Markov Models; MFCCs, Mel Frequency Cepstral Coefficients; LPC, Linear Prediction Coefficients; GMM, Gaussian Mixture Models; ACF, Autocorrelation Function; ASR, Automatic Speech Recognition; RBF, Radial Basis Function; LVQ, Learning Vector Quantization.

* Corresponding author.
E-mail addresses: bbarkana@bridgeport.edu (B.D. Barkana), buzkent@bridgeport.edu (B. Uzkent).

0003-682X/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.apacoust.2011.05.008

not have numerical difficulties. Furthermore, the ability of the SVM to be applied in a generalized fashion can be controlled easily through the parameter v, which admits a simple interpretation in terms of the number of outliers [8].

We briefly reviewed the previous work in environmental noise recognition systems in the next section.

2. Related works

ANR has become a very active subject of research during the last decades, since it can be implemented into a very wide area of topics from speech recognition to context aware applications. Researchers designed and developed classifiers using different feature parameters in both time-domain and frequency-domain. ANN, HMM, Fuzzy Logic systems, SVMs, and k-means clustering based classifiers are used widely in recognition applications. Although we have used the SVM and k-means methods in this study, we also briefly review ANN, HMM, and fuzzy systems.

ANN has been used successfully in the noise classification area. ANN can be applied to multisource data sets. Couvreura and Laniray [1] used ANN to classify feature vectors with respect to the nature of urban noises: scooter and horn. The feature vector was extracted by a time–frequency analysis. The time evolution of the power spectral envelope was represented in a compact form. The system performed well with accuracy rates between 85% and 95%.

Shao and Bouchard [9] explored the classification of active and inactive noisy speech using some well-known classifiers: linear classifiers, Nearest Neighbor classifiers, Quadratic Gaussian classifiers, and ANN. They reported that ANN produced the best and most robust performance for the classification of active and inactive noisy speech.

Hidden Markov Modeling is another powerful method in classification applications. Ma et al. [2] proposed a good application of HMM for acoustic environment classification. In their study, classification was based on combining digital signal processing technology with pattern recognition methods. HMM was applied to a range of different acoustic environments; lecture, bus, car, rail station, beach, bar, launderette, soccer match, and city center, and street noise. Classification accuracy is 75%–100%. The HMM classifier was the poorest at identifying street noise because of the wide variety of sounds in the street sample. Gaunard et al. [7] reported a HMM classifier which recognizes five noise events (car, truck, moped, aircraft, and train) with better recognition performance than human listeners.

Fuzzy logic has also been a popular method for classification. Beritelli et al. [10] proposed a new, simple, and robust background noise classifier. Seven categories of noise with similar characteristics were used. These were Bus, Car, Train, Dump, Babble, Street, Factory, and Construction noises. First, the noise database was classified as stationary or non-stationary. Then, the fuzzy logic algorithm classified the background noises. A total of 15 parameters were extracted. The proposed classifier used a subset of these parameters: a maximum of four inputs for each system. Feature extraction included differential power, diff. variance, diff. left variance, diff. short term prediction gain, norm of cepstrum, norm of Log Area Ratio coefficients, norm of LPC coefficient, norm of 10 cepstral coefficients, first normalized autocorrelation coefficient, first cepstral coefficient, first LPC coefficient, regularity, right variance, and normalized energy in the band 0–900 Hz. The accuracy rate was 58–99%.

During the last decade, the Support Vector Machines were found to be able to cope with hard classification problems in text categorization, hand-written character recognition, image classification, biosequence analysis, noise reduction etc. SVM is a non-parametric classifier. Since SVM's maximize the margin, they are also known as maximum margin classifiers. One of the key advantages of the SVM is that they are able to cope with learning in high dimensional spaces with very limited training examples. SVM seek to maximize the distance between critical data points so that the test data is easily separable [11].

In the literature, one can find many SVM applications in a wide variety of areas such as speech and speaker recognition, image recognition, and biomedical pattern recognition. However, there are very few environmental noise recognition studies based on the SVM. Environmental noise classification/recognition is very important because most signals today are digital. Context aware applications, audio surveillance systems and multimedia digital libraries use environmental noise to provide discriminatory information. Bressan and Tan designed, implemented, and evaluated an experimental multimedia library system for video clips and sound recordings in which scenes were indexed, classified and retrieved according to their environmental noise using the SVM. Their system distinguished between such scenes as traffic scenes, canteen scenes, and gunfight scenes with an accuracy of up to 90% [12]. Gerosa et al. [13] have tested a system using both Gaussian Mixture Models (GMM) and SVM as classifiers to detect events such as gunshots and screams in noisy environments. The system achieved an accuracy of 90%. They reported that the GMM provided higher precision than the SVM.

The classifier developed in this paper is proposed for the recognition of a limited number of classes of environmental sounds (engine, restaurant, and rain) and is motivated by the context aware and audio surveillance applications. Such applications, which are able to classify a number of different environmental sounds related to daily life events, can advance the quality of people's life. We propose a new set of feature parameters based on pitch, and this is different from traditional feature parameters in previous studies. The performance of the proposed new set of feature parameters is evaluated by the SVM and the k-means classifiers.

3. Proposed feature extraction method

Pitch analysis of audio signals is useful for many purposes. Applications include automatic music transcription, speech separation, structured audio coding, and music information retrieval [14]. Pitch is a perceptual feature of sounds. Its perception plays an important part in human hearing and understanding of different sounds. In an acoustic environment, human listeners are able to recognize the pitch of several simultaneous sounds and make efficient use of the pitch to acoustically separate a sound in a mixture [15]. However, background noise such as street noise, restaurant chatter, rain, the sound of a fan, unwanted music, or engine noise do not have a constant pitch value but a range of values.

Most of the noise types have different characteristics, and they can be classified according to how rapidly they change over time as stationary, quasi-stationary, and non-stationary. Stationary noise signals do not contain large or rapid changes in its spectrum over time. Quasi-stationary noise signals have a mainly constant spectrum over time and non-stationary noise signals contain large or rapid changes in spectrum over time. Techniques such as the Fast Fourier Transform and spectral subtraction can recognize and remove stationary noise using conventional digital signal processing. However, it is difficult to recognize and remove quasi-and non-stationary noise because of their changing characteristics.

We calculated feature parameters which characterize different noise types efficiently for classification purposes. Quasi-stationary noise types, rain, restaurant, and engine were used. Using a time-domain technique (the Autocorrelation Function (ACF)), pitch can be estimated. The ACF is written as (1), where $x(n)$ is the

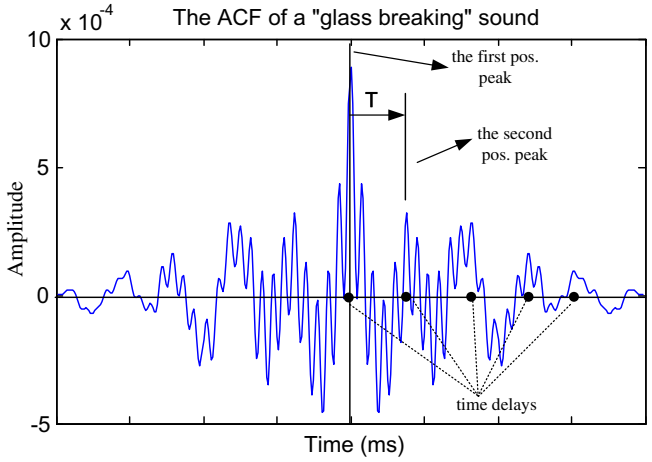

Fig. 1. The autocorrelation function (ACF) and the calculation of time delay between the first positive peak and the second positive peak (for "*Glass breaking*" sound).

time-dependent signal and N is the window size. The ACF measures the extent to which a signal correlates with a time offset (τ) version of itself. Because a periodic signal will correlate strongly with itself when offset by the fundamental period, we can expect to find a peak in the ACF at the value corresponding to a period.

$$\phi(\tau) = \sum_{n=o}^{N-1} x(n)x(n+\tau) \tag{1}$$

Pitch tracking in real-time situations usually involves additional steps beyond frame-by-frame pitch detection to enhance the quality of the measured pitch [16]. The ACF technique generates the instantaneous pitch for the input signal which will invariably contain some tracking errors. If the input signal changes pitch during an analysis frame, the resulting pitch measurement may be misleading. Since quasi- and non-stationary noise types may change their acoustical characteristics in time, we focus on the "pitch range" of the noise signal instead of the pitch itself. Pitch values are calculated by using the short-time ACF method. The following Fig. 1 illustrates this operation.

The time delay, T, between the first and the second positive peak values of the ACF for each window is calculated as shown in Fig. 1. Pitch, P, is defined as the reciprocal of the time delay, T, as (3), where M is the total number of the windows for any sound event.

$$T(i), \quad 1 < i < M. \tag{2}$$

$$P(i) = \frac{1}{T(i)}, \quad 1 < i < M. \tag{3}$$

Two features are calculated using the pitch range and its mean value for every noise instance:

(1) Feature 1 is chosen as the ratio of the maximum and minimum of the mean value (blue-solid line in graphics).
(2) Feature 2 is chosen as the ratio of the standard deviation of the pitch range and mean value.

$\overline{P}$ is the mean and *std* is the standard deviation of $P(i)$.

$$\overline{P} = \frac{1}{N}\sum_{i=1}^{M} P(i). \tag{4}$$

$$std\{P(i)\} = \left(\frac{1}{N-1}\sum_{i=1}^{M}(P(i)-\overline{P})^2\right)^{1/2}. \tag{5}$$

The pitch range of rain, restaurant, and engine noise signals is given in Fig. 2a–c. Calculated pitch values are shown in red in Fig. 2a–c. Solid blue lines in these figures show the mean value of every 40 frames without any overlap.

The three groups, Rain, Restaurant, and Engine, are represented using two feature parameters in Fig. 2d. These feature parameters form the feature vector. The proposed feature extraction method is able to separate the noise types but with some overlap.

4. Classification algorithms

Two classification algorithms, Support Vector Machines and k-means Clustering, used in our study are briefly described in this section.

The application of kernel-based machines to recognition applications is troublesome for two reasons: (1) the SVM were formulated as binary classifiers while the ASR problem is multi-class; and (2) training algorithms for SVM generally are not able to deal with the massive database used in ASR. For the first problem, we follow the "one versus all" method. In this method, one variable chosen to be recognized is taken as positive (+1) while the remaining variables combine into one cluster which is called as negative (−1). This step is followed for every variable in the database. If we have n variables, then this step is followed n times.

4.1. Support Vector Machines (SVMs) classifier

The SVM developed by Vapnik [17] has become a popular classifier algorithm recently because of its promising performance on different type of studies. The SVM is based on structural risk minimization where the aim is to find a classifier that minimizes the boundary of the expected error [9]. In other words, it seeks a maximum margin separating the hyperplane and the closest point of the training set between two classes of data. Fig. 2 shows a classification of a series of points into two different classes of data, Class I and Class II.

The SVM attempts to adjust the boundary where the distance between the boundary and the nearest data point in each class is maximal. Then, the boundary is placed in the middle of this margin. The margins are defined by the nearest data points and represented by dotted lines in Fig. 3. Nearest data points are also called support vectors. Support vectors provide all necessary information to define the classifier [18].

To find the boundaries that maximize the margins for the SVM classifier involves a classic optimization process. When the data are linearly seperable, the goal is to find the hyperlane that maximizes the margin, M. In SVM analyses, the two different classes are assumed to be identified as ±1. The decision boundary can be defined at $y = 0$ as:

$$y = \sum_{i=1}^{N} w_i x_i + b = x_i w + b = 0, \quad w \in R^N, \quad b \in R. \tag{6}$$

where x_i are the inputs, w is the boundary or weight vector, and b is the offset or bias. In each example $x_i \in R^N$ belongs to a class determined by the parameter $y = \pm 1$. The value of y must be ±1 at the margins, where the SV's are located:

$$y = \sum_{i=1}^{N} x_i w + b = \pm 1. \tag{7}$$

The decision function can be written to classify any data as belonging to either Class I or II. This equation states that w and b should be such that the two classes fall on the appropriate side of the SV lines:

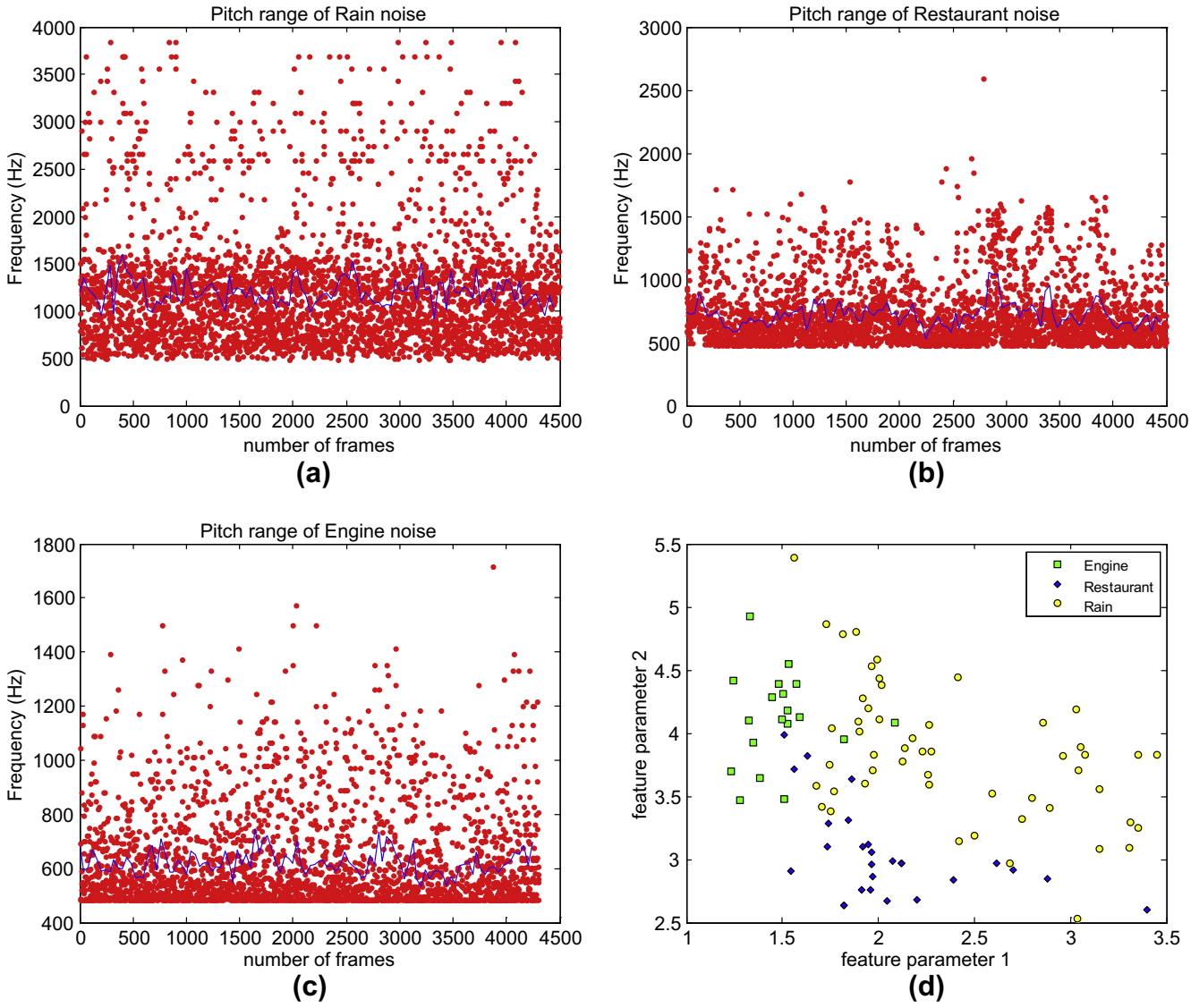

Fig. 2. Feature parameters of environmental noise signals: Pitch range of Rain, Restaurant, and Engine noise signals are given in (a)–(c) respectively (in red). Blue line in the graphics shows the mean value of 40 frames without overlapping. (d) Representation of the three groups using two feature parameters. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

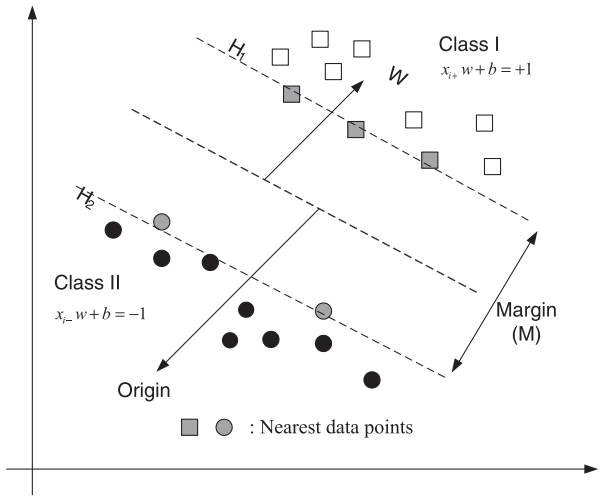

Fig. 3. An example of classification of two classes by SVM. The hyperplane separates the training data.

$$f(x) = sign\left(\sum_{i=1}^{N} x_i w + b\right). \tag{8}$$

The margin, M, can be obtained by minimizing $\|w\|$ as

$$M = \frac{2}{\|w\|}. \tag{9}$$

If the only possible way to access the feature space is via dot products computed by the kernel, then, (9) cannot be solved directly since w lies in that feature space. We can avoid the explicit use of w by forming the dual optimization problem of Lagrange. Detailed information can be easily found in [18]. Solving the dual optimization problem, α_i coefficients will be obtained in order to define the w in (9). In this case, Eq. (8) can be written as

$$f(x) = sign\left(\sum_{i=1}^{N} \alpha_i y_i (x_i \cdot x) + b\right) \tag{10}$$

Linearly separable data is not very common in real life since real life applications include complex systems. If the data is not linearly separable in the input space, some non-linear transformation

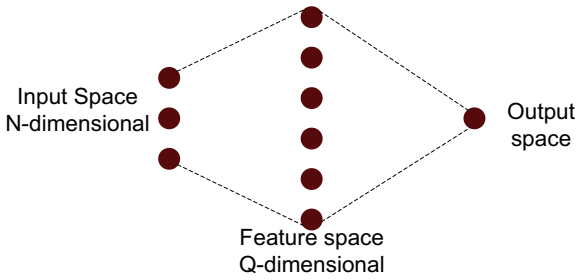

Fig. 4. Mapping the input space into high dimensional feature space.

process is followed as shown in Fig. 4. This transformation process maps the data points $x \in R^N$ into the higher dimension which is called *feature space*. The SVM achieves this through a transformation $\theta(x)$ that converts the data from an N-dimensional input space to Q-dimensional feature space. Then, this mapped data in the feature space is classified by the optimized hyperplanes.

Training data are separated into two clusters as shown in Fig. 3. The squares belong to the Class I while the circles belong to the Class II. Hyperplanes are determined by small subsets of the training set. SVM choose one of these hyperplanes, which separates the data in an optimum way, as a classifier. To summarize, SVM algorithms separate the training data in feature space by a hyperplane defined by the type of kernel function used.

Kernel functions have a central role on feature space transformation for high accuracy classification. Generally, they are used to avoid the problem of mapping data with a large number of features. They define an inner product in Q as:

$$K(x,t) = \theta(x).\theta(t), \quad x \in R^N \quad \text{and} \quad t \in R^Q \tag{11}$$

The decision function will have the form:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b, \tag{12}$$

where N is the number of data points and $y_i \in \{-1, 1\}$ is the class label of the training point x_i.

The meaning of the support vectors can be understood from the parameter α_i. It can be said that support vectors are the nearest points to the separation boundary and are the only ones for which α_i in Eq. (12) can be non-zero [19].

There are several popular kernel functions such as Polynomial, Gaussian Radial Basis Function, Exponential Basis Function and Multilayer Perceptron etc. Selection of kernel functions is the key to high performance since it is the feature space in which the training set will be classified by the Kernel function. In this study, polynomial and Radial Basis Functions (RBF) are evaluated and explained below.

4.1.1. Polynomial

A polynomial mapping has been a very popular method for the modeling of non-linearly separable data. There are two kernel functions for polynomial mapping. See (13) and (12). d is defined as the degree of polynomial mapping. The kernel in (14) is used in this study.

$$K(x,x') = (x,x')^d. \tag{13}$$

$$K(x,x') = ((x,x') + 1)^d. \tag{14}$$

4.1.2. Gaussian Radial Basis Function

Radial Basis Functions (RBF) have received significant attention due to their high accuracy in complicated studies. Commonly, it is defined by a Gaussian of the form:

$$K(x,x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right), \tag{15}$$

where σ is the variance of the Gaussian. Typically, a method of clustering is employed to select a subset of centers. This selection is implicit with each support vector centered at that data point contributing one local function.

Naturally, most of the classification problems contain more than two classes. There are some ways to reduce multiclass classification problems to multiple binary classification problems. Generally, a one-against-all classifier is built for each class. Also, one-against-one classifier is constructed for each pair of class. This study uses the one-against-all method for the classification of data.

4.2. The k-means clustering classifier

The k-means clustering classifier is a simple classification method. It represents the training data with a number of data centers known as prototypes. k is the number of prototype centers or centroids. The test data are assigned to the class of the closest prototype. The position of the prototypes determines the boundary, and the number of prototypes chosen to represent each class determines the complexity of the boundary. The main idea behind this classifier is it tries to minimize the sum of the squares of the distances between data and the matching cluster centroid. To get a more accurate output, the centroids should be placed as far from each other as possible. This process is repeated iteratively until the placement of all centroids begins to repeat previous pattern of placement. The squared error is minimized by:

$$E = \sum_{j=1}^{m} \sum_{k=1}^{n} \left\| X_k^{(j)} - c_j \right\|^2. \tag{16}$$

$\left\| X_k^{(j)} - c_j \right\|$ is the Euclidean distance measure between a data point, x_k^j, and the cluster center, c_j. It is an indicator of the distance of the n data points from their respective cluster centers.

There are several different methods for training the prototypes to find the best location to represent the data. The method used in this study is known as the Learning Vector Quantization (LVQ) method and is straighforward and fast. In the LVQ method, initial prototypes are placed randomly within each class. During training, these prototypes are moved to more ideal locations. A random training data point is selected and the closest prototype is found. If that prototype is of the same class as the training point, the prototype is moved toward the training point. If not, the prototype is moved away from the training point. The k-means clustering algorithm is composed of the following steps as given in Fig. 5.

5. Experimental results and evaluation

5.1. Database

Noise samples are taken from the Freesound Project database. The Freesound Project is a collaborative database of Creative Commons licensed sounds [20]. Nineteen Engine, 52 Rain, and 28 Restaurant noise instances were analyzed at a 96 kHz sampling frequency. All signals in the database have a 16 bits resolution. Engine noise events include the sound of cars and trucks; rain noise events include the sound of a heavy rain and light rain fall; and restaurant noise events include the sound of dining utensils and light conversation. To calculate pitch, at least 4.69 s of each instance was processed using 2.1 ms-rectangular windows with 50% overlap. The number of test and training data and the profile of the features are given in Tables 1 and 2, respectively.

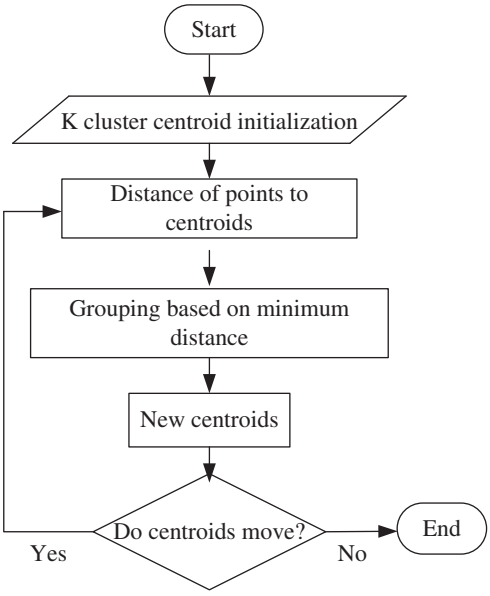

Fig. 5. The *k*-means clustering algorithm.

Table 1
The numbers of test and training data.

Noise types	Total data	Number of training data	Number of test data
Rain	52	5	47
Restaurant	28	4	24
Engine	19	5	14
Total	99	14	85

Table 2
The profile of the feature vector.

Noise types	Feature 1			Feature 2		
	Average	Standard deviation	Min–Max	Average	Standard deviation	Min–Max
Rain	2.38	0.56	1.56–3.45	3.834	0.537	2.53–5.394
Restaurant	2.04	0.429	1.512–3.39	3.04	0.371	2.60–3.988
Engine	1.486	0.203	1.235–2.08	4.111	0.368	3.47–4.922

5.2. The experimental results of the SVMs

Polynomial and RBF kernel functions were used with three different degrees: $d = 1, 2, 3$. Figs. 6–8 show the SVM classifier applied to a rain, restaurant, and engine test set of non-linearly separable data using the mapping degree of 2.

To evaluate the performance of the SVM classifier, the confusion matrix and measurements of sensitivity and specificity were calculated. The confusion matrix is a table of correct and incorrect classifications given in percent. The recognition rate is computed as the ratio of the correctly recognized sounds and the total number of the sounds to be recognized. It is given in Tables 3 and 4 for Polynomial and RBF functions, respectively. Classification accuracies are given in bold type in Table 3, 4, and 6. The recognition rate of the first degree polynomial function ranges between 65.2 and 93.3%, the second degree ranges between 91 and 93.475%, and the third degree ranges between 86.5 and 93.47%.

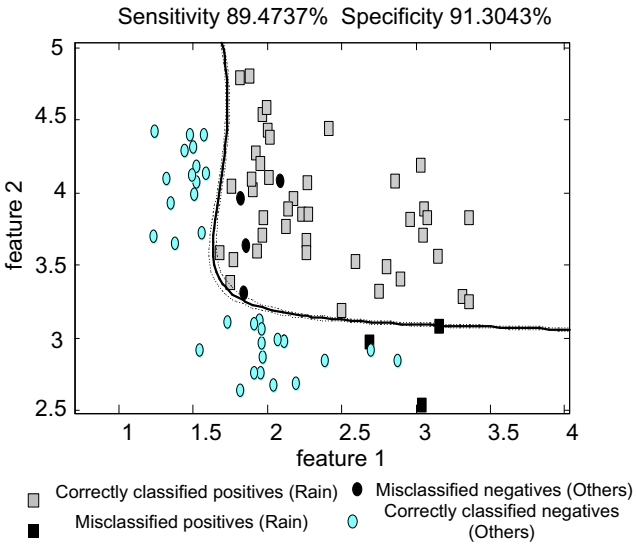

Fig. 6. The classification of rain noise type with RBF.

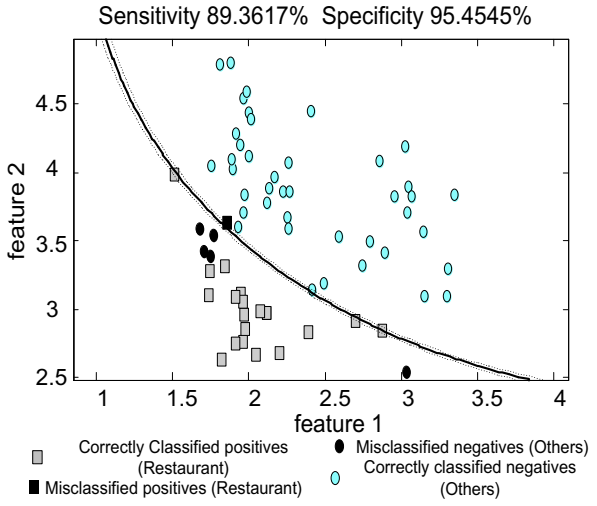

Fig. 7. The classification of restaurant noise type with RBF.

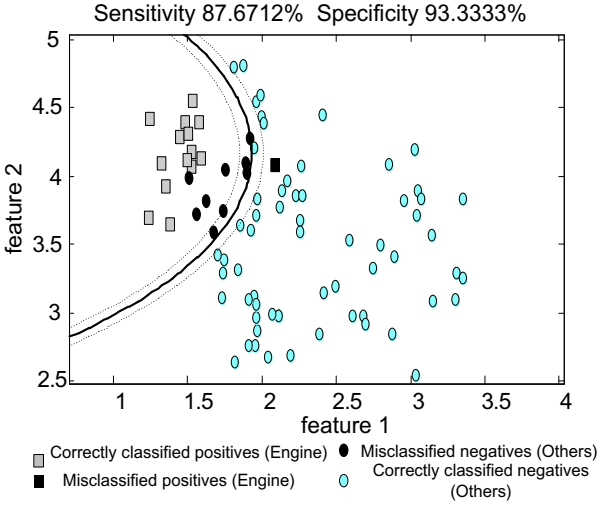

Fig. 8. The classification of engine noise type with RBF.

Table 3
Confusion matrix for SVMs with polynomial function.

True class	Polynomial Predicted class (%)								
	$d = 1$			$d = 2$			$d = 3$		
	Eng.	Rain	Rest.	Eng.	Rain	Rest.	Eng.	Rain	Rest.
Engine	**93.3**	6.67	0	**93.3**	6.67	0	**93.3**	6.67	0
Rain	8.69	**65.2**	26.08	0	**93.47**	6.53	0	**93.47**	6.53
Restaurant	9.1	0	**90.9**	0	9	**91**	4.5	9	**86.5**

Table 4
Confusion matrix for SVMs with RBF function.

True class	Gaussian Radial Basis Function (RBF) Predicted class								
	$d = 1$			$d = 2$			$d = 3$		
	Eng.	Rain	Rest.	Eng.	Rain	Rest.	Eng.	Rain	Rest.
Engine	**93.3**	6.67	0	**93.3**	6.67	0	**93.3**	6.67	0
Rain	2	**93.5**	4.5	0	**91.3**	8.7	0	**93.47**	6.53
Restaurant	9.1	0	**90.9**	0	4.6	**95.4**	0	9	**91**

Table 5
Sensitivity and specificity of the SVM classifier.

$d = 2$	Engine		Rain		Restaurant	
	Poly	RBF	Poly	RBF	Poly	RBF
Sensitivity	87.67	87.67	86.84	89.47	89.36	89.36
Specificity	93.3	93.3	93.47	91.3	91	95.4

For both kernel functions, the RBF function provided higher accuracy rates than the polynomial function. The RBF function achieved minimum 90.9% and maximum 95.4% accuracy rates. The second degree kernel outperformed the degrees 1 and 3 in both functions.

In addition to classification accuracy, the test performance of the classifiers can be determined by the computation of specificity and sensitivity. The specificity, sensitivity and total classification accuracy are defined as:

5.2.1. Specificity

Number of true negative decisions/number of actual negative cases.

5.2.2. Sensitivity

Number of true positive decisions/number of actual positive cases.

The sensitivity (a.k.a. true positive) and specificity (a.k.a. false positive) measures are given in Table 5. Our classifier performed well, with a sensitivity of at least 87.6% and a specificity of at least 91.3%.

As the degree is increased from 1 to 2, both polynomial and RBF mapping showed much better performance. Nevertheless, increasing the degree from 2 to 3 decreases the recognition rate for Restaurant data in both RBF and polynomial mapping while it increases the accuracy for Rain data in RBF. To sum up, the degree 2 demonstrates the best performance for both RBF and polynomial mapping.

5.3. The experimental results of k-means clustering

We used the same database in the SVM classifier to test the performance of the *k*-means clustering based classifier. Classification

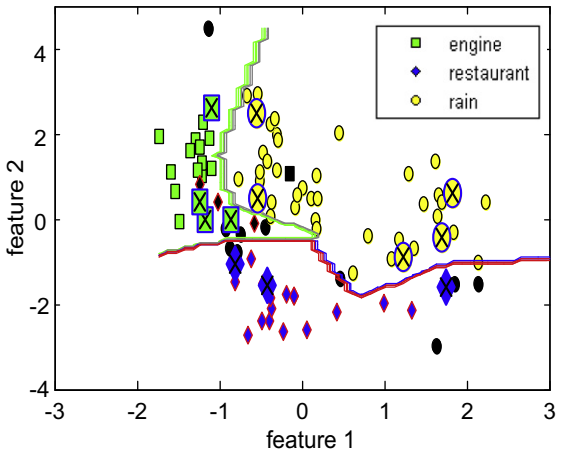

Fig. 9. The classification of data (Rain + Engine + Restaurant) with *k*-means.trun 0

Table 6
Confusion matrix for *k*-means clustering.

True class	Predicted class (%)		
	Engine	Rain	Restaurant
Engine	**92.8**	7.2	0
Rain	9	**81**	10
Restaurant	16	0	**84**

of Rain, Restaurant and Engine data is depicted in Fig. 9. The confusion matrix is given in Table 6. The recognition rates of this classifier range between 81% and 92.8%. While the highest accuracy rate is found for Engine data, the least accurate rate is calculated for Rain data.

The SVM-based classifier outperformed the *k*-means clustering classifier. Indeed, the SVM-based classifier yielded more than 91.3% of correct classifications and even more than 95% for restaurant type of noise samples. It appears that the newly proposed feature extraction based on the pitch of the sound performs at least as well as the best of the previously proposed systems. To compare the performance of our system with other noise recognition systems is difficult because of the followings: The number and type of noise sources are different; the recording conditions are different, the feature extraction is different, etc. However, recognition rates that are lower than ours were reported in previous studies using more traditional feature extraction techniques.

The performance of our system can be improved by using noise samples which are recorded by the same equipment. Our database includes environmental noise samples recorded by different persons and different type of equipment. Our results are also limited by the limited size of our database. Because of the limited size of the database, some variations of a noise event may end up being present in the test set but not in the training set. Obviously, this will be detrimental to the classifier's performance. With this consideration, we can expect higher recognition rates for an expanded database.

6. Conclusions

Automatic environmental noise recognition has been studied with SVM and *k*-means clustering algorithms using newly proposed feature extraction. The present study showed how the classifiers provide a reasonable recognition rates in classification of the engine, restaurant and rain noise samples. The extracted feature vectors were based on the pitch (or fundamental frequency)

calculations which were calculated using autocorrelation function analysis. The SVMs classifier performed well since it maps the features to a higher-dimensional space. Its recognition rates ranges from 91.3% and 95.4%. Beside this, the *k*-means clustering classifier provided recognition rates between 81% and 92.8%. Overall, these classifiers are found to improve the performance of the ANR systems.

Future work

More research is needed on understanding how the proposed new feature extraction can improve the performance of the classifiers which use traditional feature extraction methods such as MFCCs. In addition, the database needs to be expanded for human–machine and audio surveillance applications.

Acknowledgement

The authors would like to thank Dr. Lawrence V. Hmurcik at the University of Bridgeport for his advice and input during the preparation of this paper.

References

[1] Couvreura L, Laniray M. Automatic noise recognition in urban environments based on artificial neural networks and hidden markov models. In: The 33rd international congress and exposition on noise control engineering, Internoise; 2004.
[2] Ma L, Milner B, Smith D. Acoustic environment classification. ACM Trans Speech Lang Process 2006;3(2):1–22.
[3] Uzkent B, Barkana BD, Yang J. Automatic environmental noise source classification model using fuzzy logic. Expert Syst Appl 2011. doi:10.1016/j.eswa.2011.01.084.
[4] Cai R, Lu L, Zhang HJ, Cai LH. Using structure patterns of temporal and spectral feature in audio similarity measure. In: Proceedings of the ACM multimedia conference. Berkeley, CA; 2003. p. 219–22.
[5] Srinivasen S, Petkovic D, Poncelon DB. Towards robust features for classifying audio in the CueVideo system. In: Proceedings of the ACM multimedia conference; 1999. p. 340–93.
[6] Yang J, Barkana BD. The acoustic properties of different noise sources. In: Proceedings of 6th international conference on information technology – new generations, ITNG; 2009.
[7] Gaunard P, Mubikangiey CG, Couvreur C, Fontaine V. Automatic classification of environmental noise events by hidden Markov models. Appl Acoust 1998;54(3):187–206.
[8] Schölkopf B, Smola A. Learning with kernels. Cambridge, USA: MIT Press; 2002.
[9] Shao C, Bouchard M. Efficient classification of noisy speech using neural networks. In: Proceedings of ISSPA; 2003. p. 357–60.
[10] Beritelli F, Casale S, Ruggeri G. New results in fuzzy pattern classification of background noise. In: Proceedings of ICSP, vol. 1; 2000. p. 11–20.
[11] Semmlow JL. Biosignal and medical image processing. 2nd ed. CRC Press; 2009.
[12] Bressan S, Tan BT. Environmental noise classification for multimedia libraries, DEXA, LNCS 3588; 2005. p. 230–9.
[13] Gerosa L, Valenzise G, Tagliasacchi M, Antonacci F, Sarti A. Scream and gunshot detection in noisy environments. In: EURASIP european signal processing conference. Poznan, Poland; 2007.
[14] Klapuri A. Multi-pitch analysis of polyphonic music and speech signals using an auditory model. IEEE Trans Audio, Speech, Lang Process 2008;16(2).
[15] Bregman S. Auditory scene analysis: the perceptual organization of sound. Cambridge, MA: MIT Press; 1990.
[16] Cuadra P, Master A, Sapp C. Efficient pitch detection techniques for interactive music. In: Proceedings of the international computer music conference (ICMC). Havana, Cuba; 2001. p. 403–6.
[17] Vapnik, Vladimir N. The nature of statistical learning theory. Springer-Verlag Inc. New York; 1995.
[18] Yang B, Hwang W, Ko MH, Lee SJ. Cavitation detection of butterfly valve using support vector machines. J Sound Vib 2005;287:25–43.
[19] Yao Y, Frasconi P, Pontil M. Fingerprint classification with combination of support vector machines. In: international conference on audio- and video-based biometric person authentication; 2001. p. 253–8.
[20] http://www.freesound.org; 2009.