TURKISH NAVAL ACADEMY
NAVAL SCIENCE AND ENGINEERING INSTITUTE
DEPARTMENT OF COMPUTER ENGINEERING

WARSHIP SOUND SIGNATURE RECOGNITION USING MEL FREQUENCY CEPSTRAL COEFFICIENTS

MASTER THESIS

MAHMUT ALKAN

Advisor: Assist. Prof. M. Elif KARSLIGİL

İstanbul, 2005

WARSHIP SOUND SIGNATURE RECOGNITION USING MEL FREQUENCY CEPSTRAL COEFFICIENTS

Submitted in partial fulfillment of the requirements for degree of

MASTER OF SCIENCE IN COMPUTER ENGINEERING

From the

TURKISH NAVAL ACADEMY

Author:

Mahmut Alkan

Defense Date: 21/07/2005
Approved by:

Assist. Prof. M.Elif Karslıgil (Advisor)

Prof. Fikret Gürgen (Defense Comittee Member)

Assist. Prof. A. Tevfik İNAN (Defense Comittee Member)

ABSTRACT (TURKISH)

SAVAŞ GEMİSİ SES İMZALARINI MEL-FREKANS KEPSTRUM KATSAYILARINI KULLANARAK TANIMA

Anahtar Kelimeler: Ses tanıma, Destek Vektör Makineleri, Kepstral katsayılar, Savaş gemileri akustiği, MFCC.

Ses kaynaklarını tanıma konusunda müzik enstrümanlarını, insanları ve kara taşıtlarını tanıma üzerine birçok çalışma yapılmış olup deniz taşıtlarını tanıma hususunda literatürde yer alan bir çalışma bulunmamaktadır. Bu tezde seyir halindeki bir geminin ve üzerinde çalışan tüm makinaların (motor, pervane, jeneratör vb.) ortama yaydıkları sesleri değerlendirerek gemi modelini tanıyan bir sistem geliştirilmiştir.

Gemilerden alınan seslere önce alçak geçirgen filtre uygulanarak parazitler temizlenmiştir. Daha sonra Hamming pencereleme işlemi ile parçalara ayrılan seslere özellik çıkarımı için Mel-Frekans Kepstrum Katsayıları (MFCC), Delta Mel-Frekans Kepstrum Katsayıları (DeltaMFCC) ve Perde Tespit (Pitch Detection) yöntemleri uygulanmıştır. Her çerçeve için Mel-Frekans Kepstrum Katsayıları ve Delta Mel-Frekans Kepstrum Katsayıları yöntemleri ile 13'er ve 9'ar katsayı elde edilmiş, bunlardan toplam enerji yoğunluğunu temsil eden ve diğerlerinden çok farklı olan birinci özellik hariç kalan (8 ve 12) katsayı ile Perde Tespit Yöntemi ile elde edilen perde değerlerinin minimum, maksimum, ortalama, medyan, standart sapma ve genel dağılım değerlerinden oluşan katsayılar kümesi özellik kümesi olarak kullanılmıştır. 10 sn'lik ses örneği için çerçeve sayısı 992-1009 arası olduğundan vektör kuantalama yöntemi ile tanıma için kullanılacak özellik sayısı 64'e azaltılmıştır. 12 farklı gemiye ait farklı süratlerdeki 90 farklı ses eğitim seti için kullanılarak k En Yakın Komşu ve Destek Vektör Makinaları yöntemleri kullanılarak tanıma işlemi gerçekleştirilmiştir. Gemi tanıma çok sınıflı

bir işlem olduğundan Destek Vektör Makinelerinin one-against-one (bire-karşı-bir) yöntemi kullanılmıştır.

Eğitim setinde kullanılan 12 gemiden alınan 110 farklı test örneği kullanılarak yapılan 11 farklı test işlemi sonunda en yüksek sistem başarısı; 12 MFCC ve Destek Vektör Makinaları yöntemi kullanılarak yapılan test sonucunda % 82 olarak elde edilmiştir.

ABSTRACT (ENGLISH)

WARSHIP SOUND SIGNATURE RECOGNITION USING MEL FREQUENCY CEPSTRAL COEFFICIENTS

Keywords: Sound source recognition, signal processing, support vector machine, cepstral coefficients, warship acoustic, feature extraction, MFCC.

Literature survey about sound source recognition/classification systems shows that there are studies about music, musical instruments, human and vehicle sounds but none about ship sounds. In this thesis, a system which recognizes the type of a ship by evaluating the sound of working machinery on it (engine, generator, propeller etc.) was developed.

First the interferences are cleaned by Low Pass Filter (LPF). Then the sounds are divided into frames using Hamming window. For each frame 13 and 9 coefficients are extracted by Mel Frequency Cepstral Coefficients (MFCC) and Delta Mel Frequency Cepstral Coefficients (DeltaMFCC); 6 coefficients (minimum, maximum, mean, median, standart deviation and range values of pitchs) are extracted by Pitch Detection. Except the first coefficient which represents the total energy density and because of this it is so far from the others, remaining (12/8) coefficients of MFCC/Delta MFCC are used. For 10 second ship sounds, 992-1009 frames are formed and these coefficients are reduced to [12*64] matrix for recognizing by vector quantization method. The system is trained using k Nearest Neighbour (k-NN) and Support Vector Machine (SVM) methods with the training set which consist of 90 different sounds recorded at different speeds of 12 different ships. Because of the ship recognition process is a multi-class problem; SVM's one-against-one (pairwise) approach is used for recognition.

The system was tested with 110 different ship sounds and the true recognition rate was 82% with MFCC (12 coefficients) and SVM.

DISCLAIMER STATEMENT

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Turkish Navy, Naval Academy and Naval Science and Engineering Institute.

DEDICATION

To my my father Emin Alkan, my mother Sevim Alkan and my little niece Simge Özçelenk.

ACKNOWLEDGEMENT

I would like to thank my advisor Assist. Prof. Elif Karslıgil for her helpful discussions, comments, and Captain Ömer Livvarçin for obtaining the required data and support.

Finally, I would like to express my deepest gratitude to my father Emin Alkan, my mother Sevim Alkan and my little niece Simge Özçelenk for their love, patience and everlasting support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS, ACRONYMS, AND SYMBOLS

WSSRS	Warship Sound Signature Recognition System
TNRCCARG	Turkish Navy Research Center Command Acoustic Research Group
LPF	Low Pass Filter
MFC	Mel Frequency Cepstrum
MFCC	Mel Frequency Cepstral Coefficients
SVM	Support Vector Machine
SSRS	Sound Source Recognition System
VQ	Vector Quantization
HMM	Hidden Markov Model
GMM	Gaussian Mixture Models
NN	Neural Networks
LPC	Linear Prediction Algorithm
TDNN	Time Delay Neural Networks
k-NN	k Nearest Neighbour
FFT	Fast Fourier Transform
DFT	Discrete Fourier Transform
VC	Vapnik- Chervenenkis dimensionality
DCT	Discrete Cosine Transform

I. INTRODUCTION

A. MOTIVATION

The primary function of a Sound Source Recognition System (SSRS) is recognizing objects in the environment from the sounds they produce. We can recognize our friends's voice from a single word "Hi". While a beautiful song is playing on my radio, I can tell that it is Sertap Erener singing and, a piano and a violin plays. The telephone rings, and I answer it. While I am walking down the street, I can hear cars, trucks and buses driving by. A human operator in a warship listen the sound of another ship and can tell the class of it with little difficulty. Human SSRS which consists of ears, nerves and the brain, can discriminate the sounds, is shown in Figure 1.

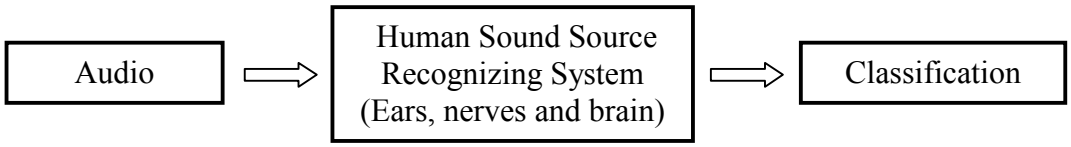

Figure 1.Human Sound Source Recoginition System

There exists some work for recognizing musical instrument sounds [1, 2, 3, 4, 5], vehicle sounds [6, 7, 8] and human speech [9, 10, 11, 12, 13, 14] as sound source recognition applications.

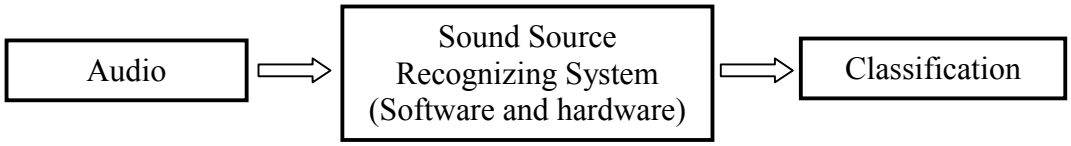

Figure 2.Artificial Sound SourceRecoginition System.

SSRS which is shown in Figure 2, analyzes the input audio signal and creates a label that describes the signal at the output.

Although the sound of a cruising ship and the machinery on it (engine, generator, propeller etc.) provides an important clue to recognize its type, in literature review no research on this subject was found. This thesis introduces a novel approach to warship recognition by sound characteristics.

B. PROBLEM DEFINITION AND PROPOSED SCHEME

One of the application areas that well fit the characteristics of SSRS is ship/warship SSRS. Depending on the environmental conditions, its quiet high costs and military confidentiality, recording warship sounds is a difficult task. Our warship sound database was constructed from Turkish Navy Research Center Command Acoustic Research Group (TNRCCARG)'s ship sound records. The database consists of 200 samples of 12 different kind ships.

In the preprocessing stage, first the noise in audio signals is removed using the Low Pass Filter (LPF) [15]. After filtering we broke the sound into windows (frames) using Hamming Window [16] and then extracted the features by Pitch Detection, Mel-Frequency Cepstral Coefficients (MFCC) [17, 18, 19] and Delta MFCC. Since the extracted features for 10 second are about 1000, they are reduced to 64 by Vector Quantization (VQ) [20] method. Finally unknown ship is recognized using k Nearest Neighbour (k-NN) and Support Vector Machine (SVM) [21, 22]. The warship sound source recognition system block diagram is shown in Figure 3.

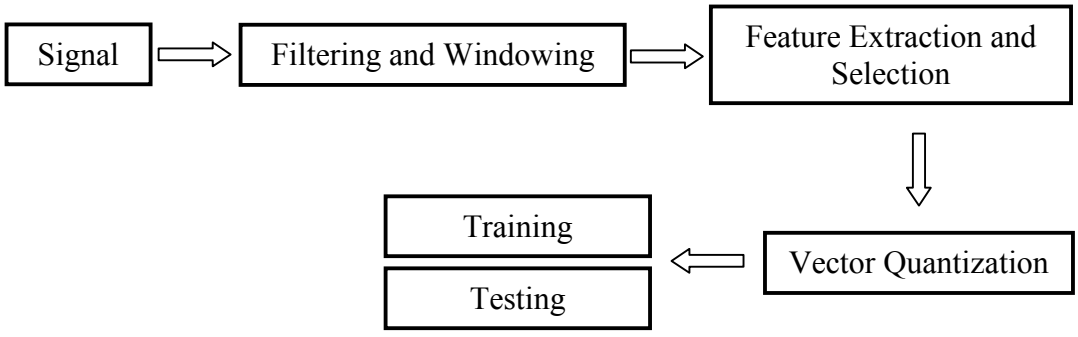

Figure 3.Block-diagram of Warship Sound Signature Recognition System.

The warship sound source recognition system's major task is to recognize ships' class and ships' itself which is detected by submarines, sensor nodes or surface ships. This task is executed by human operators at present. The scope of our study is to develop an efficient recognition and classification scheme for ships, by their audio signals. Our scheme first gets an unknown ship sound which is detected or recorded, then extracts, selects and quantizes the features. And scheme recognizes the target comparing it with training sounds in database.

C. ORGANISATION OF THIS THESIS

In Chapter II, background information about other SSRSs is given. Descriptions of the preprocessing, feature extraction and selection, vector quantization, classifiers (statistical models) and recognition of targets by using k-Nearest Neighbour and Support Vector Machine are given in Chapter III. The performance of the scheme is evaluated in Chapter IV, and the thesis is concluded in Chapter V.

II. BACKGROUND INFORMATION AND RELATED WORK

Recognition is a process of gathering information about objects in the environment and then to estimete their behavior. Many systems have been built to recognize sounds in different domains. To name a few, systems have been constructed to recognize musical instruments in a recording, to identify talkers on a telephone, and to classify vehicle sounds.

In this section, sound-source recognition systems from several domains will be considered and evaluated generally.

A. HUMAN VOICE RECOGNITION

The well known studied sound source recognition problem is *speaker recognition and verification*. Speaker recognition, which involves two applications: speaker identification and speaker verification, is the process of automatically recognizing who is speaking on the basis of individual information included in speech waves. Speaker verification is the process of determining whether the speaker identity is who the person claims to be. Speaker identification is the process of finding the identity of an unknown speaker by comparing his/her voice with voices of registered speakers in the database. Three major approaches are;

Long-term averages of acoustic features, which computes phonetic variations of features and average out these variations [23].

*Model the speaker-dependent features within phonetic sounds***,** which compares within similar phonetic sounds in the train and test utterance [23].

Discriminative neural networks (NN), which are trained to model the decision function which best discriminates speakers within a known set [23].

The problems of speaker recognition and verification systems are; Acoustic conditions of test utterances are very important. If they vary from those

used in training, these systems do not work well. Also the performance of these systems decreases as population size grows.

Case Reynolds's system [13], uses MFCC as input features. These coefficients, based on 20 ms windows of the acoustic signal, are thought to represent human vocal-tract resonances. A recorded utterance is given to the system which forms it with Gaussian distributions. During training, these models are stored in memory. To recognize a novel sound, the system finds the model from the memory that is most likely match the novel one.

The performance of the system depends on the noise characteristics of the signal, and the population size. With utterances recorded in good acoustic conditions, performance is nearly perfect. Under varying acoustic conditions, performance decreases.

System evaluates the acoustic properties that the human listeners use for talker identification but does not use frequency and rhythm of the speaker's voice, which are known to be an important clue for human listeners.

B. VEHICLE SOUND RECOGNITION

Several systems have been constructed to recognize vehicle sounds. A typical example of such a system is one constructed to recognize different kinds of motor vehicles from the engine and road noise they produce [6]. First, a human-selected segment of sound waveform is coded by a linear prediction algorithm (LPC). Then, the LPC coefficients are presented to a time delay neural network (TDNN) that classifies the source of the sound waveform as belonging to one of four categories (trucks, sedans, motorcycles, and vans).

Nooralahiyan and his coleagues performed two studies: one with sounds recorded carefully in isolated conditions, to evaluate the quality of the feature set;

and one with sounds recorded on a city street, to evaluate the system in more realistic conditions. In both cases, supervised learning was used. For the city street case, the system was trained with 450 sounds and tested with 150 independent sounds. The system's performance was, with correct classification of 96% of the training samples and 84% of the test samples.

Mel Siegel, Huadong Wu and Pradeep Khosla constructud a Vehicle Sound Signature Recognition System by Frequency Vector Principal Component Analysis [7]. They recorded all training and test sounds under stable recording conditions, when the microphone fixed in the same place to record all samples. The performance evaluation of their system is shown in Figure 4.

Case Gaunard's system [24] uses MFCC as input features. 12 cepstral coefficients, in this case based on 50-100 ms frames of car, truck, moped, aircraft and train sounds are given the system. And the system forms these utterances with GMM. The recognition performance of the system is 90-95% with cepstral coefficients as features.

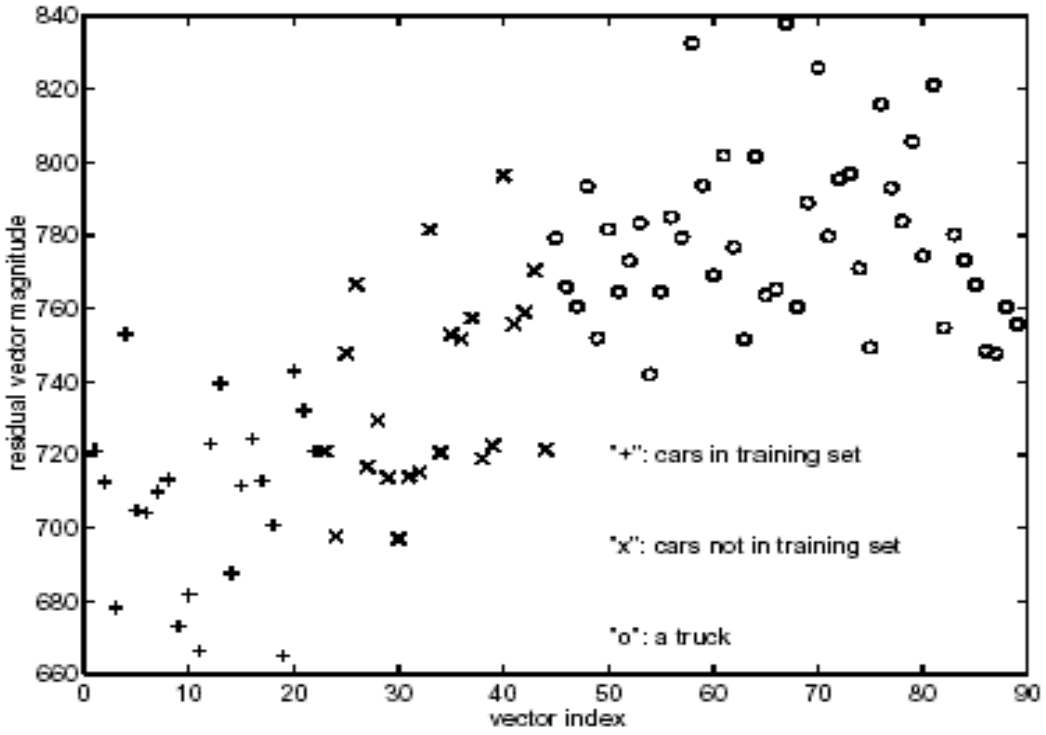

Figure 4(a) Classification of a heavy truck from sedan car class [7]

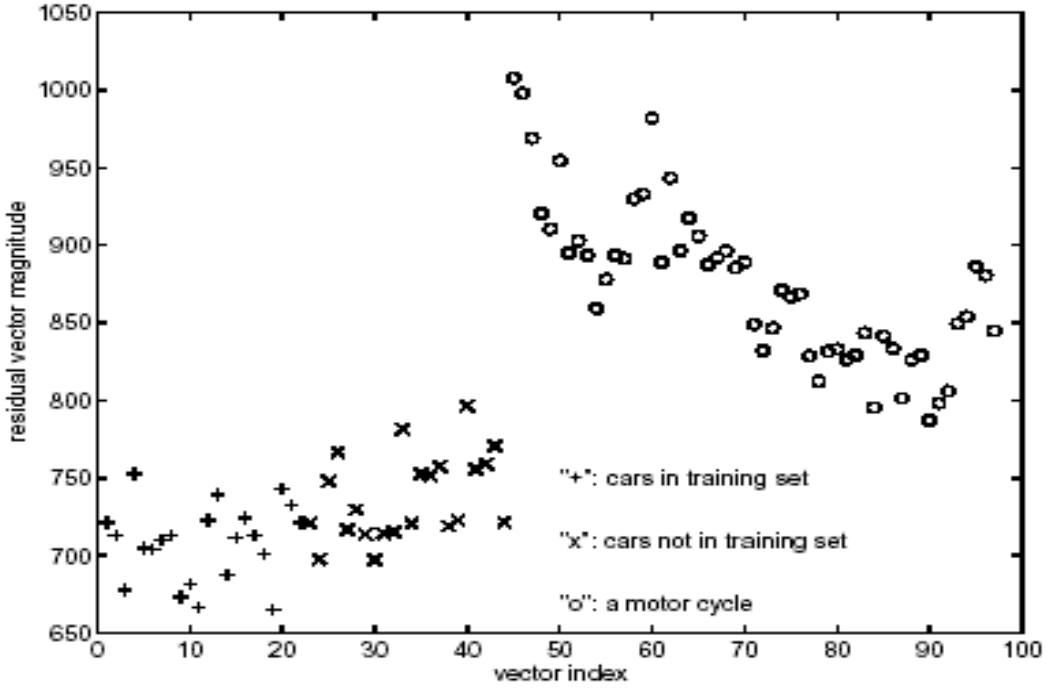

Figure 4(b) Classification of a motor cycle from sedan car class [7]

C. MUSICAL INSTRUMENTS RECOGNITION

Several musical instrument recognition systems have been constructed with varying approaches and levels of performance. These systems have operated on recordings of either single, isolated tones or synthesized and natural tones.

Kaminskyj and Materka used features which are derived from a root-mean-square (RMS) energy envelope [5]. They used a neural network and a k-nearest neighbor (k-NN) classifier to classify guitar, piano, marimba and accordion tones over a one-octave band. Both classifiers achieved approximately 98% performance.

III. WARSHIP SOUND SIGNATURE RECOGNITION USING MEL FREQUENCY CEPSTRAL COEFFICIENTS

Figure 5 presents the block diagram of the main stages of our Warship Sound Signature Recognition System.

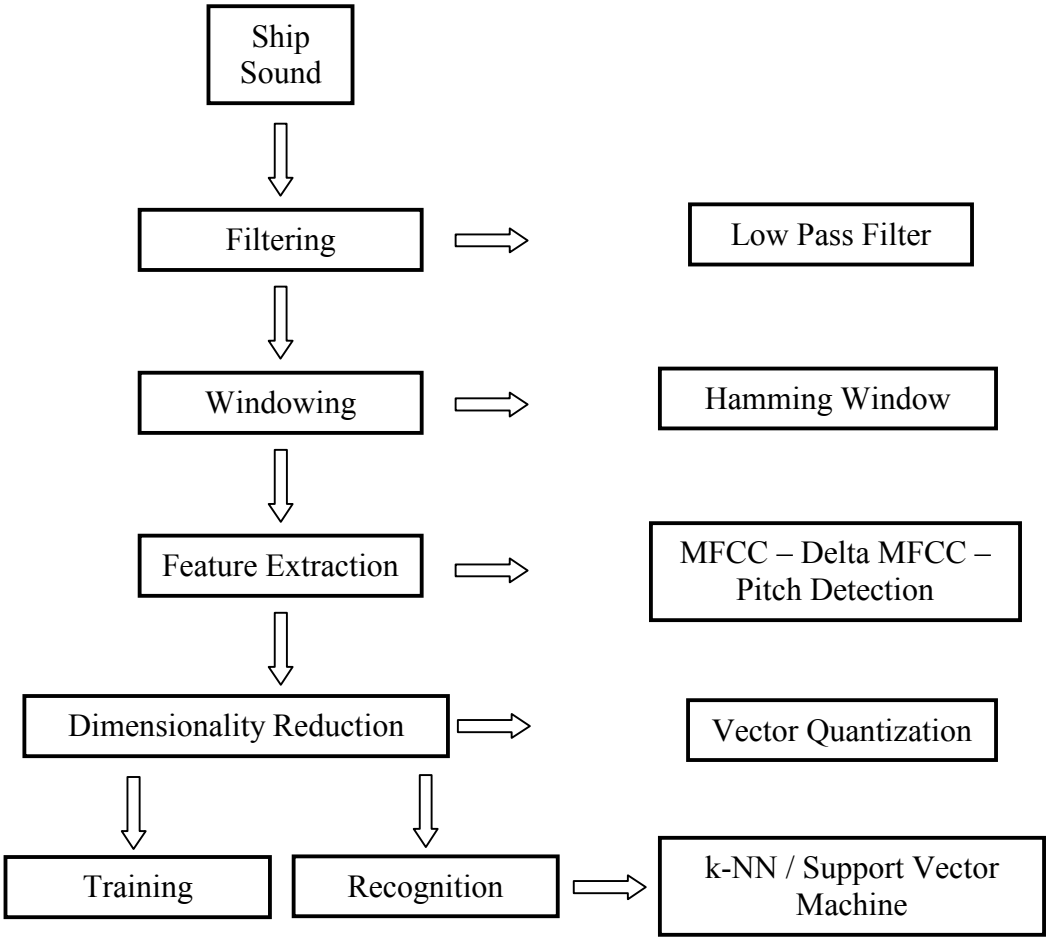

Figure5. Block Diagram of Warship Sound Signature Recognition System.

A. SHIP SOUND SIGNAL ACQUISITION

The warship sounds have been recorded using hydrophones at different dates by Turkish Navy Research Center Command Acoustic Research Group (TNRCCARG). They are saved in wav format. Each of 200 sound samples of 12 different ships takes 10 second long.

Hence the environmental conditions (sea, sea temperature, air temperature etc.) and the recording location were different for the warships. The recording condition is shown in Figure 6.

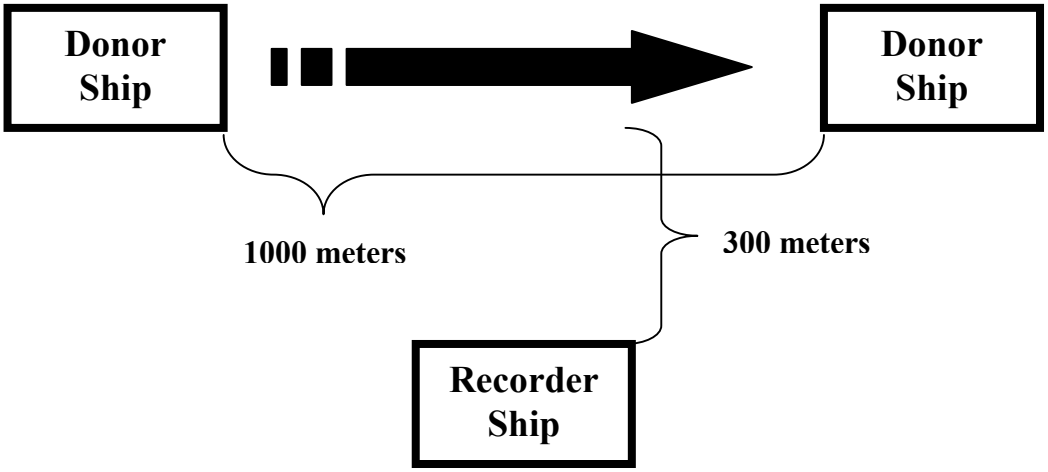

Figure 6.Recording Position of WSSRS Dataset

The Warship sounds are recorded while they are cruising along 1000 meters line which is 300 meters far away from the recorder ship.

B. FILTERING

The noises in the audio signals are eliminated by simple noise reduction filter [25] which is shown below,

$$Y_n = a * X_n + (1-a) * X_{(n+1)}$$

X_n: input signal Y_n: *output signal* *a: cut off rate and* $0 < a < 1$

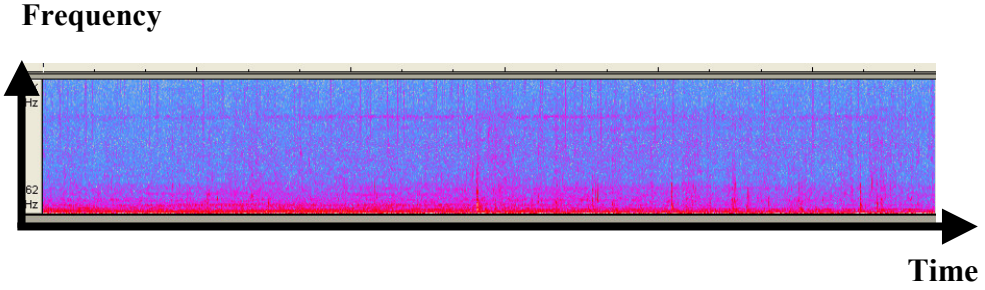

Figure 7.Spectrogram of warship sound.

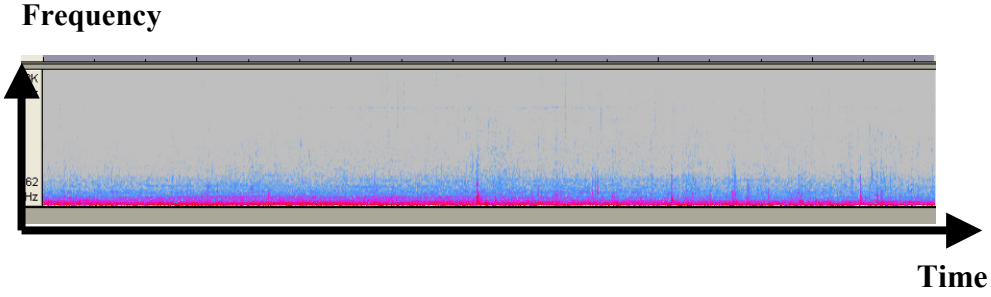

Figure 8.Spectrogram of filtered warship sound for a=0,4.

B. WINDOWING

In this section we will examine the windowing methods. Signals are changing with time, where we usually want to analyse a snapshot of the signal, short enough in which we could consider the generating system to be stationary. To do this we must take a sample or a *window* of the signal over some defined interval. If we simply cut out a part of the signal, we can not be sure that part is the best one which contains whole information about entire signal. While forming out windows it

is very important to overlap them to examine whole signal. Another inconvenience of simply cut out a part of a signal is, not to be able to overlap. We used Hamming Window in our WSSRS. Definitions of some windowing functions are;

Rectangular Window

A Rectangular Window is the simplest windowing method which approaches zero sharply. This is equal to multiplying it with a rectangular function which is 1 inside and 0 outside. Its general form and function are shown below.

$$\mathrm{Re}\,ct\,(\chi,\tau) = \begin{Bmatrix} 1\,|\chi| < \tau \\ 0\;else \end{Bmatrix} \qquad [25]$$

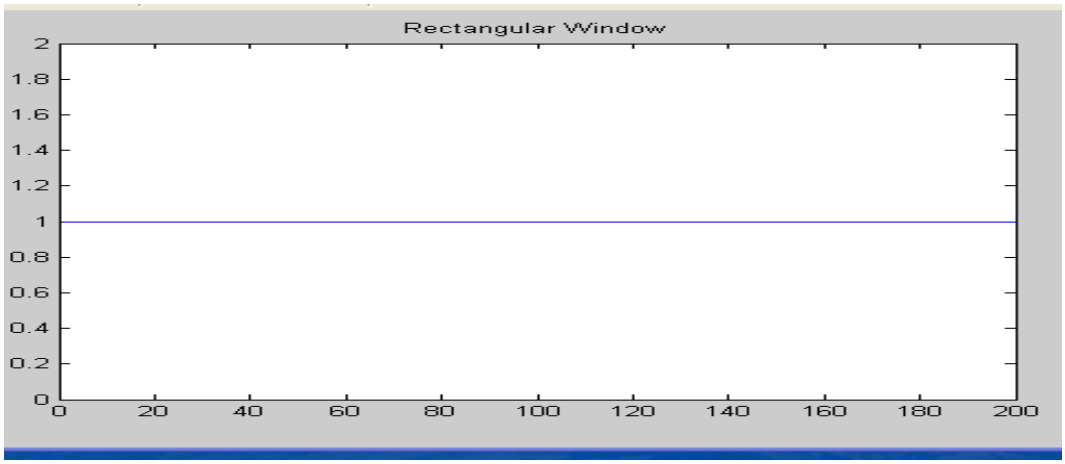

Figure 9.Rectangular window function.

Bartlett Window

A Bartlett Window is a rather simple windowing function which approaches zero more smoothly. Its general form and function are shown below.

$$Bartlett \quad (\chi, \tau) = \begin{Bmatrix} 1 - \frac{|\chi|}{\tau} |\chi| \\ 0 \quad else \end{Bmatrix} \qquad [25]$$

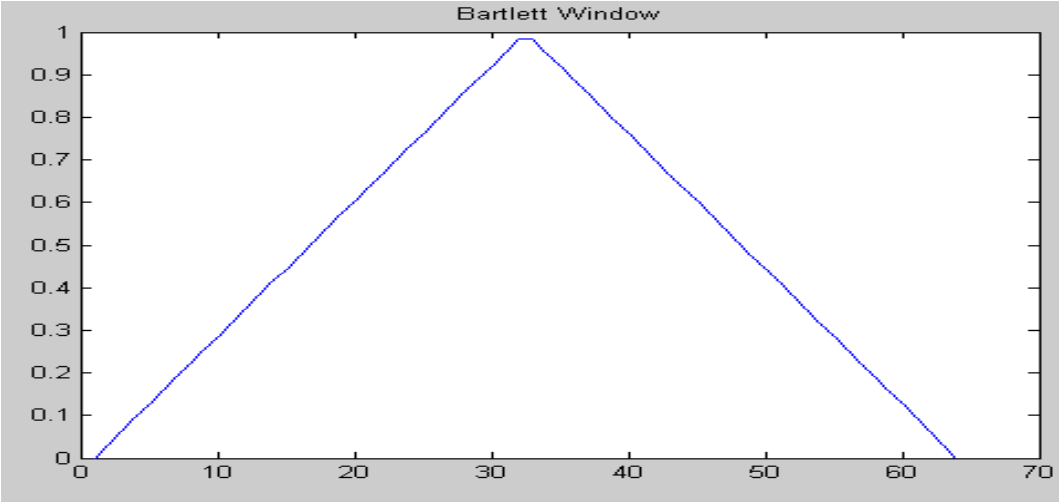

Figure 10.Bartlett window function.

Blackman Window

The Blackman window approaches zero more smoothly than Bartlett Window. It's quite similar to Hann and Hamming windows except one more cosine function. Its general form and function are shown below.

$$Blackman(\chi, \tau) = \begin{Bmatrix} 0.42 + \frac{1}{2}\cos(\pi \frac{\chi}{\tau}) + 0.08\cos(2\pi \frac{\chi}{\tau}) & |\chi| < \tau \\ 0 & else \end{Bmatrix} \qquad [25]$$

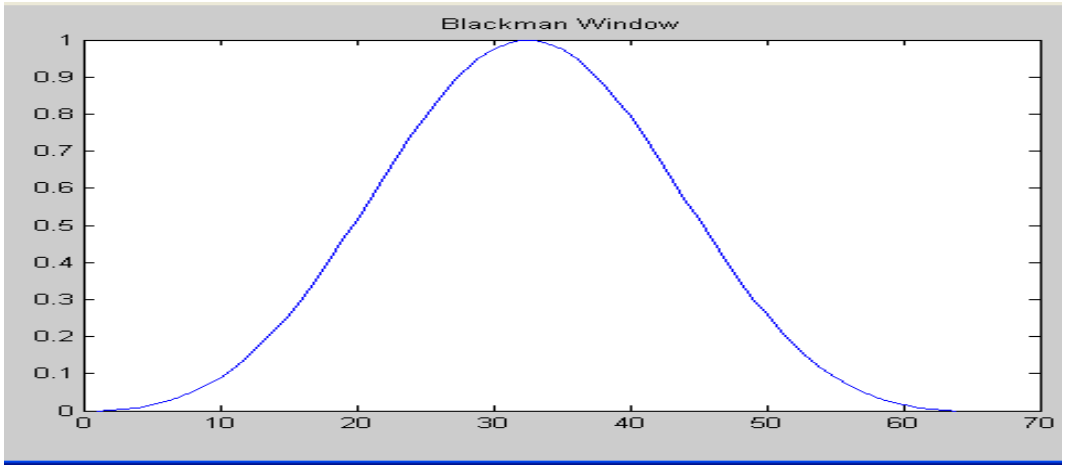

Figure 11.Blackman window function.

Hann and Hamming Window

The Hann and Hamming window are quite similar; they only differ in the parameter α with $\alpha = \frac{1}{2}$ being the Hann window and $\alpha = 0.54$ the Hamming Window.

$$H(\chi,\tau,\alpha) = \begin{Bmatrix} \alpha + (1-\alpha)\cos(\pi\frac{\chi}{\tau} & |\chi| < \tau \\ 0 & else \end{Bmatrix} \quad [25]$$

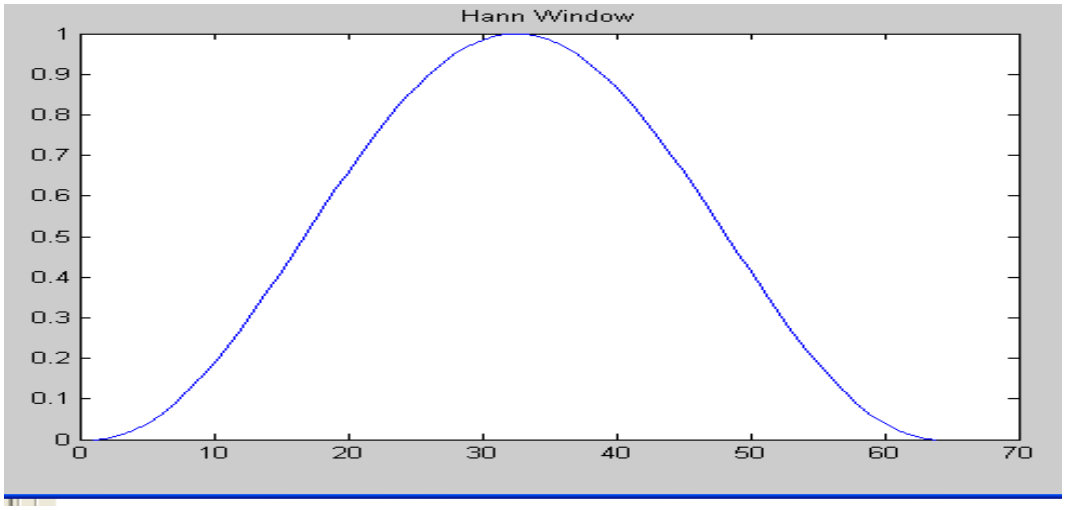

Figure 12.Hann window function.

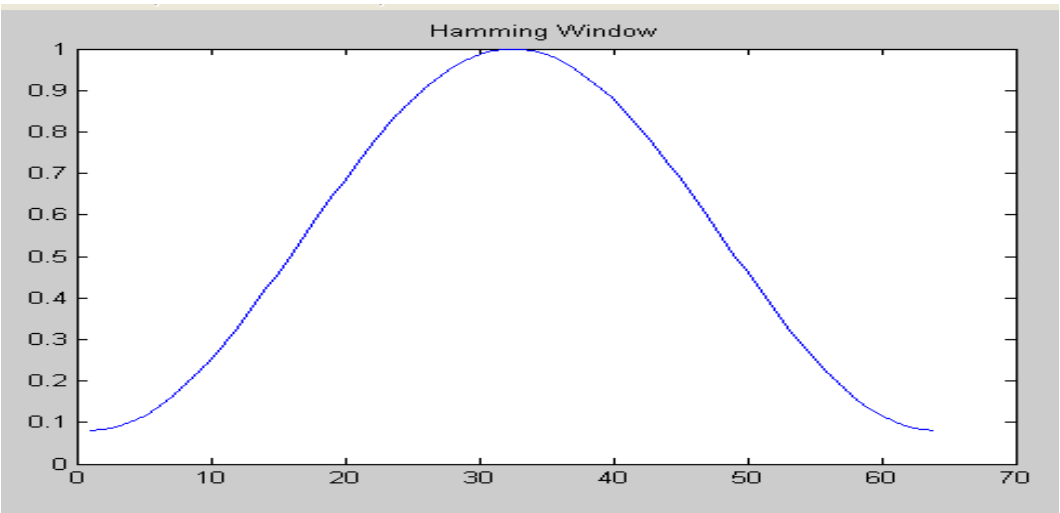

Figure 13.Hamming window function.

The Hamming window weights are computed as a function of a cosine

$$w_i = (0.54 - 0.64\cos[2\pi i/(N-1)])/\sum_{i=0}^{N-1} w_i \; 0 \le i \le (N-1) \quad [25]$$

where N is the length of the window.

Warship SSRS Hamming Window function is shown in Figure 14, where **window length = 256 (16 msec)**

sampling rate = 16000 Hz. (16 kHz)

window step = 160 (10 msec)

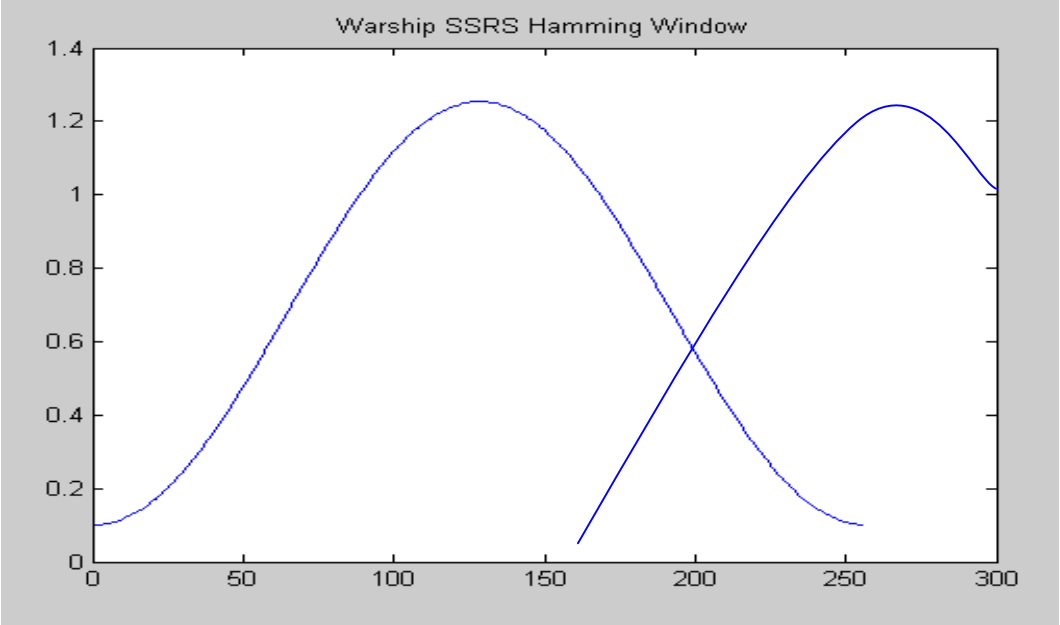

Figure 14.Warship SSRS Hamming window function.

C. FEATURE EXTRACTION

Feature extraction involves the analysis of the input of the audio signal. Audio signal features are mostly extracted by breaking the input signal into a succession of analysis windows or frames, each of around 10-40-ms length, and computing one feature value for each of the windows, explained in previous section. There are two approaches to use values of windows.

One approach is to take the values of all features for a window to form the feature vector which is used for the classification decision. This approach is realizing a real-time classifier.

Another approach is to take the values of all windows for a feature to form the long-term characteristics of the signal. This approach provides a better description of the signal than the window itself. A feature based approach is a long-term segment in the range of seconds containing a number of analysis windows. Therefore real-time classification is not possible, since at least one whole texture window has to be processed to obtain a class decision.

Since the analyzed audio files are supposed to contain only one type of audio, a single class decision is made for each type of audio, which can be derived following one of two possible approaches.

The second approach is the texture window mode, which consists of defining shorter texture windows and making several class decisions along each file, one for each texture window. At the end of the file the decisions are averaged to obtain a final class decision. This average computation is weighted by the certainty of each class decision.

As discussed previously feature extraction plays an important role in classification of an audio signal. Hence it becomes all the more important to select those features that help the classification process more efficient.

The whole feature vector is more important than a window's feature vector for our system. Hence we preferred the *feature based approach* in our WSSRS.

As the second step, we used *Single Vector Mode* for recognition to shorten processes and the processing time. Because a valuation of each texture window's classification weight process and an averaging process would be added on our system.

There are different types of features that have been used for classification of musical, human or other application areas. We explained well known and mostly used features below.

1. FEATURES FOR SOUND RECOGNITION

a. Pitch:

A sound consists of pulses which are a combination of a fundamental frequency and its harmonics. The fundamental frequency of a pulse is known as the *pitch*[18, 19, 26, 27, 34]. In music, the fundamental or first harmonic of any tone is perceived as its pitch.

After the voiced parts of the sound are selected the pitch has to be determined. There are several algorithms currently in use for determining the pitch. These could be categorized into Time-domain and Frequency-domain analysis. In time domain analysis the pitch could be estimated by using the peaks, but due to the harmonics this method could give a wrong estimation. So the formant frequencies are filtered out using a low pass filter or any other suitable method is used to determine the pitch. The speech signal is also passed through a low pass filter in the frequency domain analysis and then the pitch is determined by analyzing the spectrum.

b. Timbral features:

The ear can recognize the sounds that have same pitch and loudness. Timbre or quality is a general term of recognizable characteristics of a tone. Timbre is mainly measured from harmonic content of a sound. In music timbre is the quality of a musical note that distinguishes different types of musical instrument [18, 19, 26, 27, 34]. The following are some of the timbral features,

(1) Zero crossings

The zero crossings feature counts the number of times that the sign of the signal amplitude changes in the time domain in one frame.

(2) Centroid

The spectral centroid which models the sound sharpness, is defined as the center of gravity of the spectrum.

(3) Rolloff

The rolloff is defined as the frequency below which 85% of the magnitude distibution of the spectrum is concentrated.

(4) Flux

The spectral flux is defined as the squared difference between the normalized magnitudes of successive spectral distributions that correspond to successive signal frames.

(5) Mel frequency cepstral coefficients (MFCC)

MFCCs are a compact representation of the spectrum of an audio signal taking into account the nonlinear human perception of pitch. MFCCs are computed by grouping the Short Time Fourier Transform (STFT) coefficients of each frame into a set of 13-40 coefficients, using a set of 40 weighting filters that simulate the frequency perception of the human hearing system. Then the logarithm of the

coefficients is taken, and a discrete cosine transform (DCT) is applied to decorrelate them.

c. Rhythm features:

When there are individually perceivable events in the sound that repeat in a predictable manner, it is considered rhythmic. To extract rhythm features from a piece of sound, repetitive events in energy level, pitch or spectrum distribution are examined. Rhythm features are rhythmic regularity [18, 19, 26, 27, 34] and beat strength [18, 19, 26, 27, 34], that define the characteristic of an audio signal because they follow a particular pattern.

d. MPEG-7 features:

Moving Pictures Experts Group (MPEG) has defined international standards of techniques for analyzing and describing raw data in terms of certain features that have been discussed so far. It is an attempt to standardize the features that are used in audio signal classification. It deals the content-based description so that data can be described in terms of features. [17] The features are,

(1) Audio spectrum centroid (ASC)

An adapted definition of the centroid, which introduces a logarithmic frequency scaling centered at 1 kHz.

(2) Audio spectrum spread (ASS)

It describes concentration of the spectrum around the centroid

(3) Audio spectrum flatness (ASF)

It can be defined as the deviation of the spectral form from that of a flat spectrum.

(4) Harmonic ratio (HR)

A measure of the proportion of harmonic components within the spectrum, defined as the maximum value of the autocorrelation of each frame.

We used pitch features, MFCCs and Delta MFCCs within four different ways shown in Table 1.

Table 1.WSSRS Coefficient Types and Usage

Type 1	12 MFCC coefficients	Totaly 12 coefficients
Type 2	8 MFCC coefficients	Totaly 8 coefficients
Type 3	12 MFCC coefficients 12 Delta-MFCC coefficients 6 Pitch features	Totaly 30 features
Type 4	8 MFCC coefficients 8 Delta-MFCC coefficients 6 Pitch features	Totaly 22 features

First we trained and tested our WSSRS using 12 MFCC coefficients. Then we decreased the number of coefficients to decrease the processing time and complexity. Then we appended Delta-MFCC and 6 pitch coefficients (minimum, maximum, mean, median, standart deviation and range values of pitchs) to examine the recognition rate of the system.

2. FEATURE EXTRACTION PHASE

Before any audio signal can be classified under a given class, the features in that audio signal are to be extracted. These features will determine the class of the signal. Feature extraction involves the analysis of the input of the audio signal. Hence it becomes the more important to select those features that help the classification process more efficient. Therefore the goal of feature extraction is to

transform the high-dimensional audio signal space to a low dimensional features subspace. Some of the well known feature extraction techniques are,

a. Power Spectral Analysis (FFT)

One of the well known techniques of examining a audio signal is Power Spectral Analysis which describes the frequency components of a time domain signal.

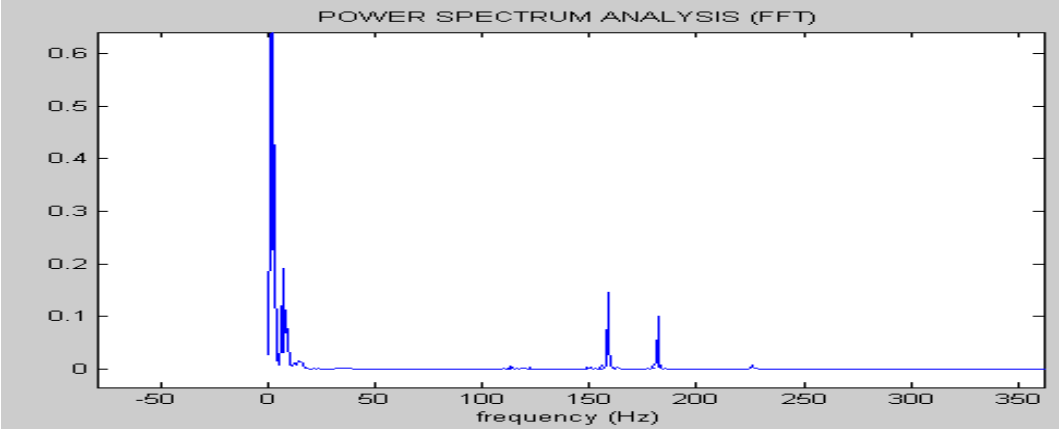

Figure 15.Power Spectrum Analysis (FFT).

The power spectrum of a signal is derived from a Discrete Fourier Transform (DFT) which computes the frequency information of the time domain signal and a Fast Fourier Transform (FFT) which increases efficiency [25, 28]. The output contains both the magnitude and phase information of the original time domain signal.

b. Linear Predictive Analysis (LPC)

Linear Predictive Analysis is a speech processing technique which estimates the values of a signal, using a linear combination of past samples. Through minimizing the sum of squared differences between the previous speech

samples and linear predicted values, predictor coefficients are determined. These coefficients form the basis for linear predictive analysis of speech.

The representation of LPC is;

$$\chi(n) = -\sum_{i=1}^{p} a_i \chi(n-i) \quad [28]$$

where ***x(n)*** = predicted signal value

***x* (*n* - *i*)** = the previous observed values

$\mathbf{a_i}$ = the predictor coefficients

Predictor coefficients are not used in recognition directly, since they show high variance. The predictor coefficients are transformed to cepstral coefficients for recognizing.

c. Mel Scale Cepstral Analysis (MEL)

Mel Scale Cepstral Analysis [28] is one of the most used techniques in speech recognition and musical signals classification.

The Cepstrum was defined in 1963 [35] is the result of taking the Fourier transform of the spectrum. The Cepstral Analysis detects bearing tones by looking at the spectrum of a spectrum. There exists a common frequency spacing separating the peaks of signature groups. Cepstral Analysis converts a spectrum back into a time domain signature, which has peaks corresponding to the period of the frequency spacings common in the spectrum. These peaks can be used to find the bearing wear peaks in the original spectra.

Cepstrum of signal = FT(log(FT(the signal)))

The mel scale, proposed by Stevens, Volkman and Newman in 1937 is a perceptual scale of pitches judged by listeners to be equal in distance one from another. The reference point between this scale and normal frequency measurement is defined by equating a 1000 Hz (1kHz) tone with a pitch of 100 mels. The normal frequency *f* hertz can be converted to the mel range by the following equation, [19]

mel = 1127.01048 log (1 + *f* / 700)

Mel Frequency Cepstral Coefficients (MFCC) [17, 18, 19] that are formed using Mel Scale Cepstral Analysis, are computed by grouping the Discrete Fourier Transform (DFT) coefficients of each frame into a set of several coefficients that simulate the frequency perception of the human hearing system. We used *Mel Scale Cepstral Analysis* to extract MFCC features that are used for warship sound signature recognizing. The detailed explication of MFCC is below.

Pseudo Code of MFCC

1. Break into windows in overlapping steps *(Hamming window in Warship SSRS)*
2. Compute frequency response using DFT for each window
(FFT size = 512 in Warship SSRS)
3. Groupe magnitude of frequency response into 25-40 channels using triangular weighting functions which are shown in Figure 16. *(13 linear + 27 log filters = totally 40 filters in Warship SSRS)*
4. Compute log of weighted magnitudes for each channel *(Mel Scaling)*
5. Take inverse DFT of weighted magnitudes for each channel, producing 13-30 cepstral coefficients for each *frame (13 cepstral coefficients for Warship SSRS)*

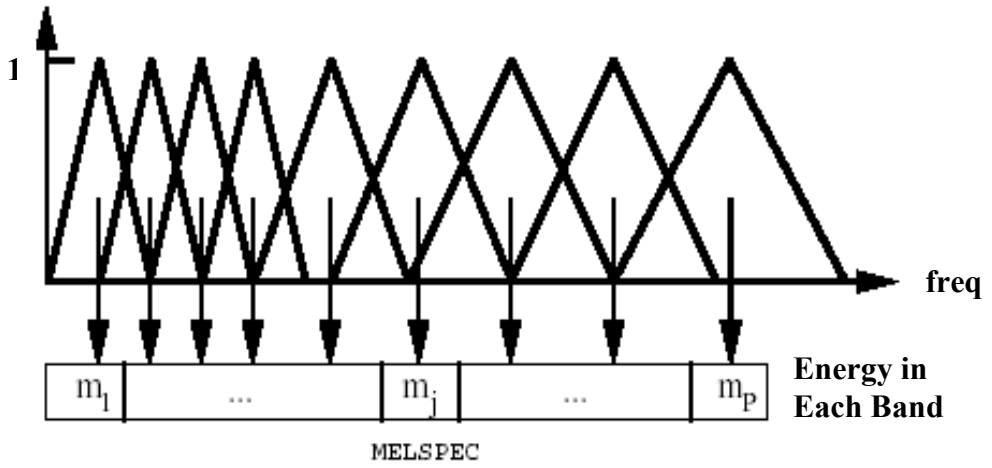

Figure 16.Mel-scale filter bank [19].

.

E. DIMENSIONALITY REDUCTION

After the feature extraction process we trained and tested our system using the four different types of our feature set. Because of the greatness of our feature set, both the system performance and the recognizing rate were so low. Hence we decided to apply dimensionality reduction to increase them.

Vector quantization is used in many applications such as image and voice compression and voice recognition for dimensionality reduction. A vector quantizer protects the important information in the feature set and forms codewords using the Euclidean distance from the input vectors is shown in Figure 16[20]. The set of all the codewords is called a *codebook*. The pseudo code of VQ is below,

Pseudo Code of Vector Quantization

1. Define the number of codewords, *N* (We tested our system with 8, 32, 64, 128 and got the best results with 64. Hence we used 64 in Warship SSRS)

2. Select *N* codewords at random, and let that be the initial codebook. The initial codewords can be randomly chosen from the set of input vectors.

3. Clusterize the vectors around each codeword using the Euclidean distance measure.
4. Compute the new set of codewords.

5. Repeat steps 2 and 3 until the either the codewords don't change or the change in the codewords is small.

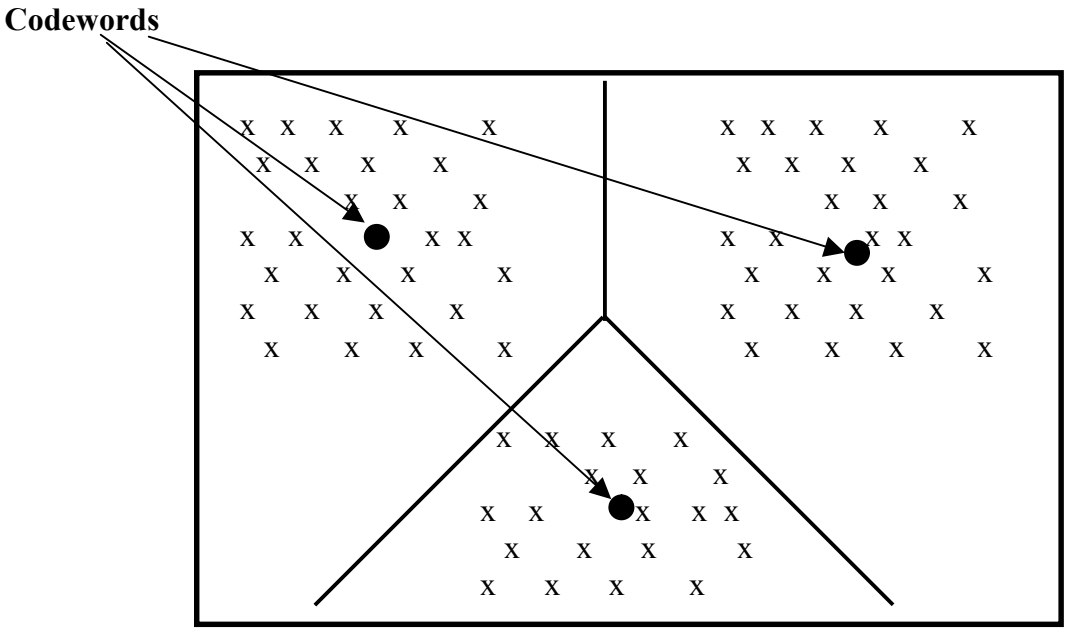

Figure 17.Codewords in 2-dimensional space. Input vectors are marked with an x, codewords are marked with a black circle, and the Voronoi regions are separated with boundary lines.

F. RECOGNITION PHASE

After the feature extraction process it is important to classify the signal. Classification is the process by which a particular label which defines the signal

and its origin is assigned to a particular audio format. Classifiers are categorized by their real time capabilities, on the basis of the approach and their character.

On the basis of their real time capabilities, there are real time and non-real time classifiers. Real time classifiers can update classification results in time intervals of milliseconds. Hence their application comes of importance in the areas where the input signal consists of a sequence of different types of audio and it is absolutely necessary to keep updating, for class detection. In case of the non-real time classifiers, they analyze a longer fragment of the signal before they provide a classification result. Accuracy in this case is more than real time classifiers because they analyze a longer fragment of the incoming signal, which plays a prominent role to describe the signal.

On the basis of approach, there are direct and hierarchical approach classifiers. The direct approach classifiers decide the class of an audio directly by using all the features in one single step. The hierarchical approach classifiers suggest a hierarchical scheme so that, at each step only the features that are most appropriate to distinguish between the sub-features are used. The errors are more acceptable in this case than the direct classification techniques. The addition of a new class in a direct classifier means that the feature selection algorithm would have to be run with all the training samples. But in the hierarchical classifier to add a new class effects only a branch and the rest of the model remains unchanged. The hierarchical classifier is more complicated and its implementation is computationally expensive because more classification decisions have to be made and more features have to be computed.

In case of all the different types of classifiers it is important use an efficient algorithm that would classify the different audio inputs with less of computational complexities. At the same time the accuracy must be preserved. The two most commonly used methods in Sound Source Recognizing Systems are k-nearest neighbour (k-NN), the Gaussian mixture model (GMM) [26, 28] and the

Support Vector Machine (SVM) [21, 22, 31] classifiers. We used k-NN and SVM classifiers in our Warship SSRS and then compared the results. These three classifiers are explained in the following sections briefly.

1. k-nearest neighbour (k-NN) classifier

The k-nearest neighbour method signs the unknown feature vector with the label of the training vector that is nearest to it in the feature space [33]. In k-NN, a training set T is used to determine the class of an unknown sample X. First, we determine the mean and maximum values in T, and similarly, for the unseen sample X. Then a suitable distance measure in the feature space is used to determine k elements in T closest to X. If most of this k nearest neighbours contains similar values, then X gets classified accordingly. This classification scheme clearly defines nonlinear decision boundaries and thus improves the performance.

There are two main problems in this classification method. The first is to find a suitable distance measure that which sample is closer to the unknown sample [33]. The mostly used distance metrics are,

Euclidian distance:

Euclidian distance between the samples (x, y) is defined as

$$|x - y| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad [33]$$

Where *n* is the number of features describing *x* and *y*.

City-block distance:

The city-block distance can be defined as

$$D_{city-block}(x, y) = \sum_{i=1}^{n} |x_i - y_i| \qquad [33]$$

Where *n* is the number of features describing *x* and *y*.

The second problem is the choice of k, choosing k large generally results in a linear classifier whereas small k results in nonlinear ones[18]. This influences the generalization capability of the k-NN classifier. The optimal k can be found by using for instance the leave out one method on the training set. A disadvantage of this method is its large computing power requirement, since for classifying an object its distance to all the objects in the learning set has to be calculated.

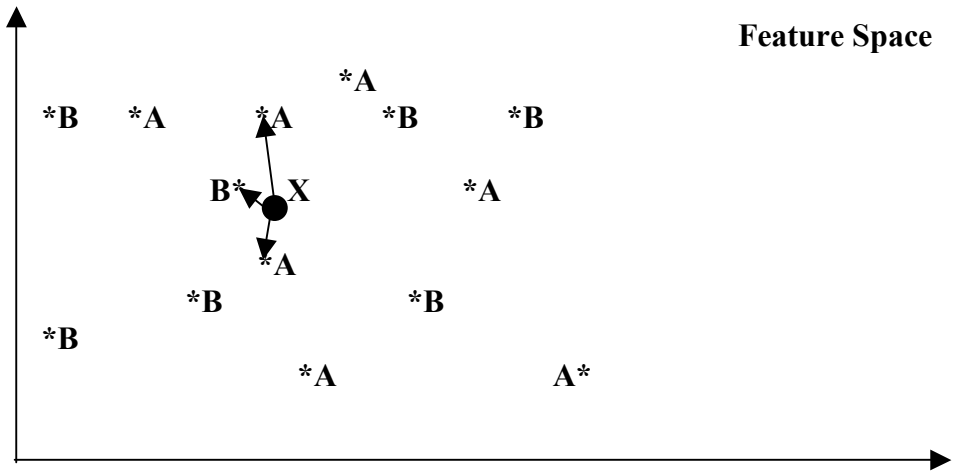

Figure 18.Example of k nearest neighbour rule (for k=3)

In Figure 18, for k=3, although B is the nearest class to unknown sample X, there are two samples of A and one sample of B. Hence the unknown sample X is belonged to A.

2. Gaussian Mixture Model (GMM) classifier

Gaussian mixture model is a weighted sum of Gaussian probability density functions (Gaussian components)[13]. The Gaussian probability density function in one dimension is defined by two parameters, mean and variance. The

Gaussian distribution is usually quite good approximation for a class model shape in a suitably selected feature space. In the Gaussian distribution lies an assumption that the class model is truly a model of one basic class. If the actual model, the actual probability density function, is multimodal, it fails. Gaussian mixture model (GMM) is a mixture of several Gaussian distributions and can therefore represent different subclasses inside one class. The GMM classifier models each class as a linear combination of Gaussian or normal densities that is; each class k is represented by the multidimensional conditional density.

$$p(x\left|\omega_k\right.) = \sum_{m-1}^{M} w_{km}\, p_{km}(x) \quad [13]$$

Where ω is the event that belongs to class *k*, $\boldsymbol{x}$ denotes feature vector, $\boldsymbol{w_{km}}$ are the weights of the mixture and M is the total number of densities (components) in the mixture and $\boldsymbol{p_{km}}$ is normal density. [13]

In a particular case when M=1, each class gets modelled by normal distribution and the classifier simplifies to a simple Gaussian classifier. Estimation of the Gaussian mixture parameters for one class can be considered as unsupervised learning of the case where samples are generated by individual components of the mixture distribution and without the knowledge of which sample was generated by which component. Clustering usually tries to identify the exact components, but Gaussian mixtures can also be used as an approximation of an arbitrary distribution.

The expectation maximization (EM) algorithm is an iterative method used to handle cases where an analytical approach for maximum likelihood estimation is infeasible, such as Gaussian mixtures with unknown and unrestricted covariance matrices and means. The values of $\boldsymbol{w_{km}}$, the mean vectors and the covariance matrices for each component in a particular class, which are the parameters for that class, are estimated using expectation maximization (EM) algorithm only. When an input vector has to be classified its conditional density in each of the classes is computed using the estimated parameters. The class for which the density value is

highest becomes the class that is chosen for that vector. This decision rule is called as the maximum likelihood condition. This rule can be applied if the different classes are equally probable.

The GMM classifier has to only store the set of estimated parameters for each class while a k-NN classifier needs to store all the training vectors in order to compute the distances to the input feature vector. Also the number of features that are required to attain the same level of accuracy is more in the case of k-NN classifier as compared to GMM classifier. Therefore these features make the GMM more computationally optimal but the k-NN classifier is still an efficient classifier that is very simple in methodology.

3. Support Vector Machine (SVM)

Support Vector Machines are learning machines. Its fundamental idea is very simple; locating the boundary that is most distant from the vectors nearest to the boundary in both sets.

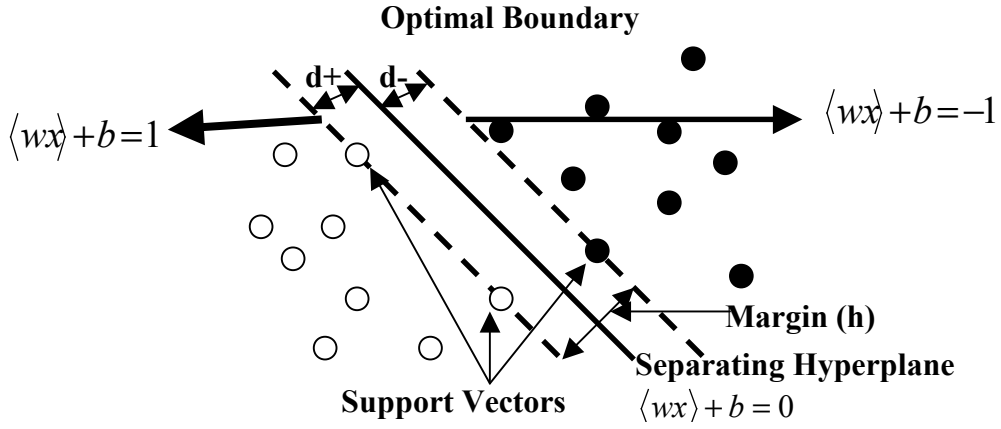

Figure 19.Optimal boundary by SVM

The "optimal" boundary is defined as the most distant hyperplane from both sets which is shown in Figure 19. In other words, this boundary passes the "midpoint" between the two sets. Although the distribution of each set is unknown,

this boundary is expected to be the optimal classification of the sets. The training vectors nearest to the boundary are called *support vectors*.

Let $\boldsymbol{x}$ be a vector in a vector space. A boundary hyperplane is expressed as one of the hyperplanes [21]:

$$\langle wx \rangle + b = 0$$

where $\boldsymbol{w}$ is a weight coefficient vector and b is a bias term. The distance between a training vector x_i and the boundary, called *margin,* is expressed as follows [21] :

$$\frac{|\langle wx \rangle + b|}{|w|}$$

If the classes are not linearly separable, a hyperplane exactly classifying the sets does not exist, as shown in Figure 20.

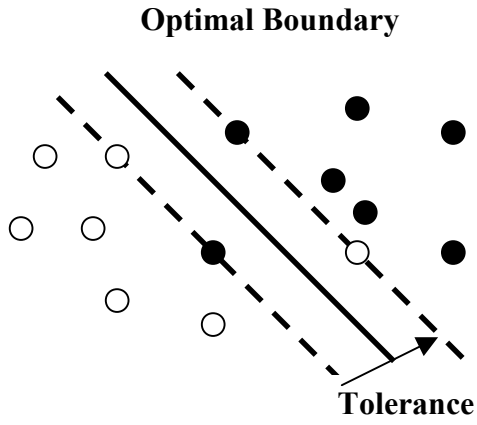

Figure 20.Linearly nonseparable case

The method called *soft margin* is a solution to such case [21]. The *slack variables* are positive variables that indicate tolerances of misclassification [21]. This replacement indicates that a training vector is allowed to exist in the limited area in the erroneous side along the boundary, as shown in Figure 20. The soft margin method is an extension of support vector machine within the linear framework.

The *kernel method* is a method of finding truly nonlinear boundaries. The fundamental concept of the kernel method is deformation of the vector space itself to a higher dimensional space. See a linearly nonseparable example presented in the previous session, shown in Figure 21(a) and Figure 22(a). The two-dimensional space is transformed to the threedimensional one as shown in Figure 21(b) and Figure 22(b), "black" vectors and "white" vectors are linearly separable.

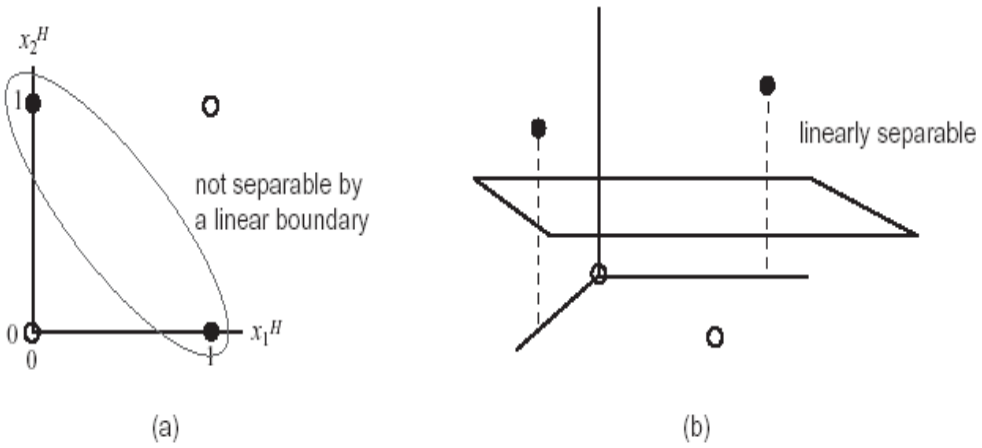

Figure 21.Transformation to higher dimensional space. [21]

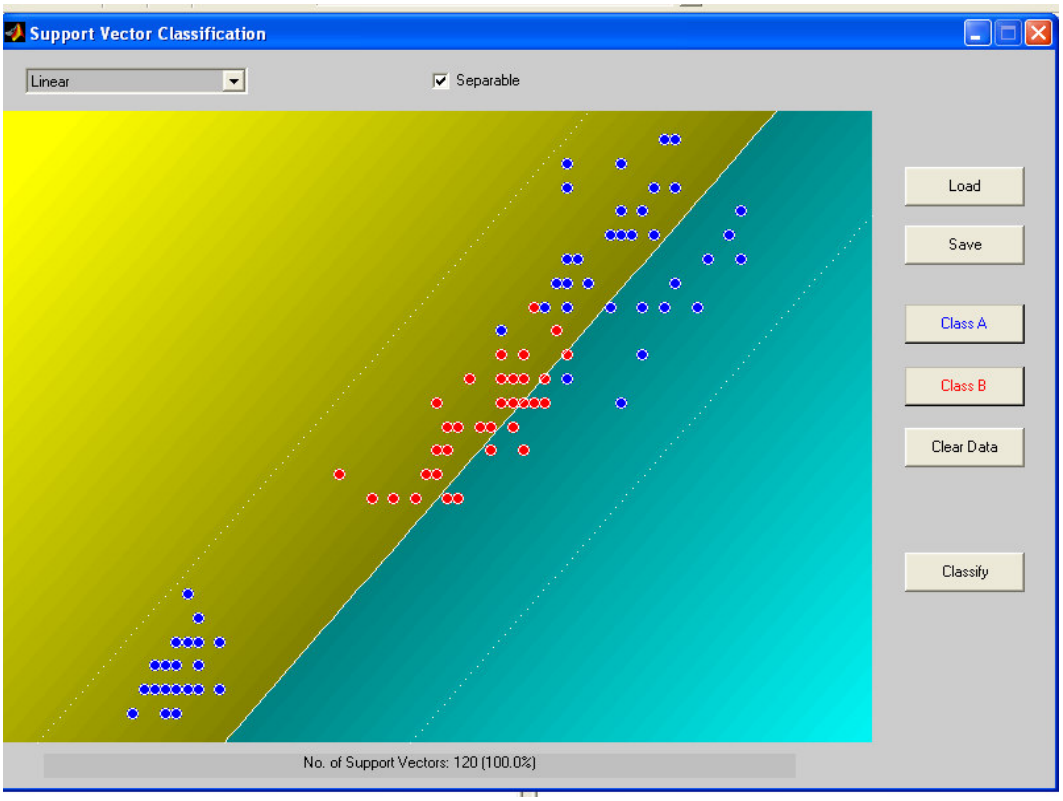

*Figure 22(a).*Linearly nonseparable example.

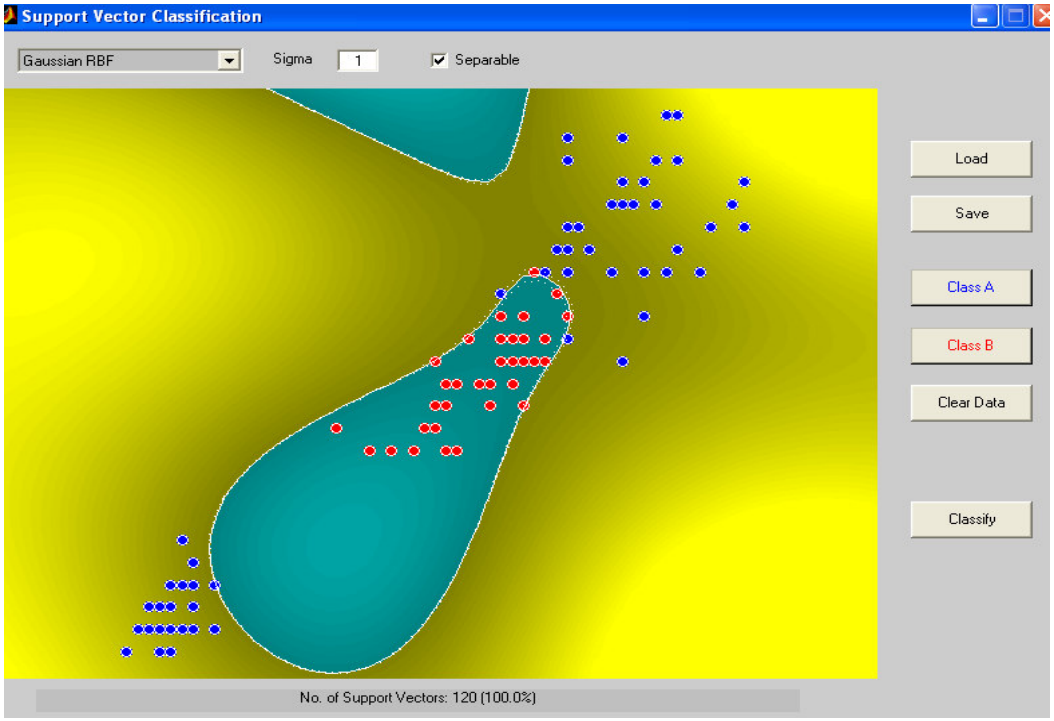

Figure 22(b).Transformation to higher dimensional space using Gaussian RBF.

The transformed space should satisfy that the distance is defined in the transformed space and the distance has a relationship to the distance in the original space.

The term *empirical risk* means the misclassification rate for *known* training vectors. It is not what we want to minimize; Our objective is minimizing the misclassification rate for *all* vectors in each set, including *unknown* vectors. This misclassification rate is called *expected risk*. In case of linearly separable problems, there exists a boundary plane that makes the empirical risk zero. The concept of support vector machine to find the boundary with the largest margin is equivalent to selecting a hyperplane minimizing the expected risk, from the set of hyperplanes that makes the empirical risk zero. This is formally explained in the framework of *structural risk minimization* with the concept of *Vapnik-Chervenenkis (VC) dimensionality*. [21]

They perform the structural risk minimisation principle. SV machines create a classifier with minimised VC dimension. If the VC dimension is low, the expected probability of error is low as well, which means good generalisation.

Support Vector Machines non-linearly map their n-dimensional input space into a high dimensional feature space. In this high dimesional feature space a linear classifier is constructed. Two results make this approach successful:

The generalization ability of this learning machine depends on the VC dimension of the set of functions that the machine implements rather than on the dimensionality of the space. A function that describes the data well and belongs to a set with low VC dimension will generalize well regardless of the dimensionality of the space.

Construction of the classifier only needs to evaluate an inner product between two vectors of the training data. An explicit mapping into the high dimensional feature space is not necessary. In Hilbert space inner products have simple kernel representations and therefore can be easily evaluated.

So far the discussion has been restricted to the case where the training data is linearly separable. However, in general this will not be the case. There are two approaches to generalising the problem, which are dependent upon prior knowledge of the problem and an estimate of the noise on the data. In the case where it is expected (or possibly even known) that a hyperplane can correctly separate the data, a method of introducing an additional cost function associated with misclassification is appropriate.

Support Vector Machine parameters are completed with combined coefficients. In this section we will examine how the SVM tool recognizes the test sounds. SVM needs training set and training targets for creating support vectors. Training targets consist of (1) and (-1). The first training data is marked with (1) and the second one with (-1). The test data is compared with two support vectors, which belong to first and second training data and then SVM reports the output as (1) or (-1). We can determine only for our first test data by this process. While working with multiple data there are two methods for recognizing. These methods are:

a. One-Against-Rest Method:

In this method the test data is compared with two classes which consist of one of the train data and the rest of the train data which is shown in Figure 24. The first class which has only one train data is labeled with (+1) and the second which consists of other seven train data is labeled with (-1). The test data belongs to the class which labels it (+1). For the number of (n) classes, true classification might be done with minimum 1, maximum (n) comparisons. This method increases the processing time. Hence it can be used with small amount of train data sets. Because that there are big amount of warships in the Navy, this method is not adequate for the Warship Sound Signature Recognition system.

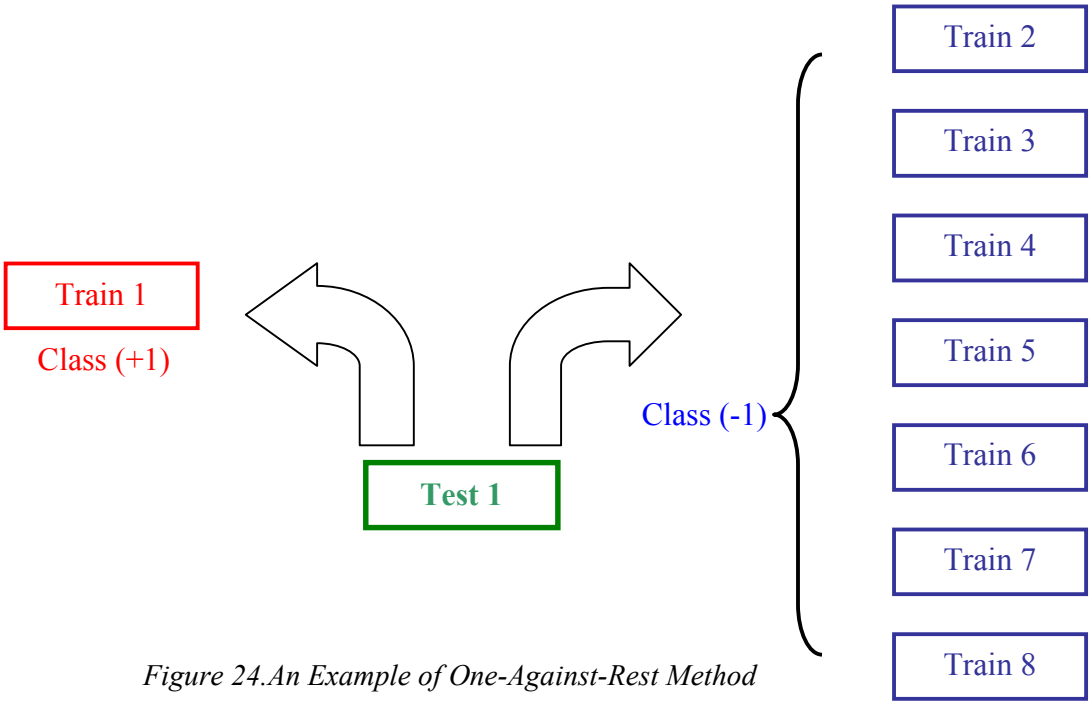

Figure 24.An Example of One-Against-Rest Method

b. Pairwise Method

In this method the test data is compared with pairs of two train data which is shown in Figure 25. Train data which is not in the same class with test data is discarted. Process is repeated with half of the training set and at the end, class of the test data arises.

For the number of (n) classes, true classification is done with i comparisons where $(n) \leq 2^i$.

This method prevents repeats of same training data usage and decreases the processing time. Hence we used Pairwise method in Warship SSRS.

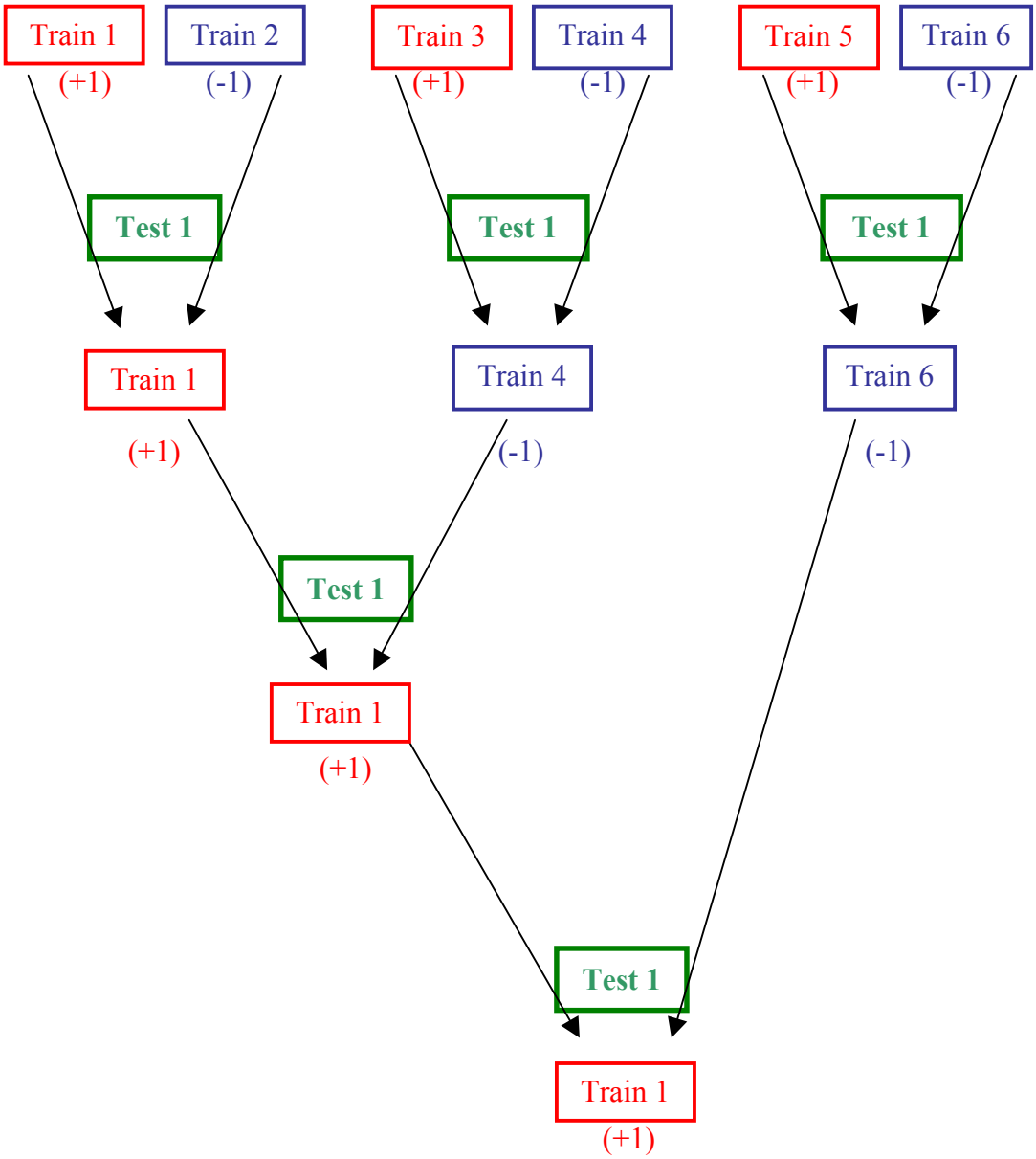

Figure 25.An Example of Pairwise Method

IV. PERFORMANCE EVALUATION

A. CONSTRUCTING WARSHIP SOUND DATABASE

Our warship sound database was constructed from Turkish Navy Research Center Command Acoustic Research Group (TNRCCARG)'s ship sound records.The raw data which we constructed from TNRCCARG were 90-170 seconds long. We divided them into 10 second long 9-17 frames and obtained 200 sound samples of 12 different kind ships. The 90 sounds of these samples are selected by listening for training. The remaining 110 samples are used for testing that is shown in Table 2. After forming the train and test data set we filtered and windowed all data using Hamming Window. Approximately 1000 frame arised after windowing.

Table 2.Training and Test data set.

NO	SHIP CLASS	SPEED	Train/Test Data
1	A-1	7 KNOT	A-1-1/…/A-1-9
	A-2	14 KNOT	A-2-1/…/A-2-13
2	B-1	7 KNOT	B-1-1/…/B-1-13
	B-2	14 KNOT	B-2-1/…/B-2-12
3	C-1	7 KNOT	C-1-1/…/C-1-13
	C-2	14 KNOT	C-2-1/…/C-2-11
4	D-1	7 KNOT	D-1-1/…/D-1-12
	D-2	14 KNOT	D-2-1/…/D-2-17
5	E-1	7 KNOT	E-1-1/…/E-1-9
	E-2	14 KNOT	E-2-1/…/E-2-9
6	F-1	7 KNOT	F-1-1/…/F-1-9
7	G-1	12 KNOT	G-1-1/…/G-1-9
8	H-1	12 KNOT	H-1-1/…/H-1-9
9	I-1	10 KNOT	I-1-1/…/I-1-15
	I-2	20 KNOT	I-2-1/…/I-2-9
10	J-1	20 KNOT	J-1-1/…/J-1-9
11	K-1	1 KNOT	K-1-1/…/K-1-11
12	L-1	9 KNOT	L-1-1/…/L-1-9

B. FEATURE EXTRACTION

After constructing the database we used our MFCC, Delta MFCC and Pitch Detection for feature extraction. We derived 6 pitch features which are minimum, maximum, mean, median, standart deviation and range, 13-9 Delta Mel frequency cepstral coefficients and 13-9 Mel frequency cepstral coefficients for approximately 1000 frames from these sounds. An example of Mel Frequency Cepstral Coefficients of Ship Sounds is shown in Table 3. We used totally 40 filters that consist of 13 linear and 27 log filters to derive MFCC and Delta-MFCCs. FFT size is 512 in our system. The first coefficient which is so far from the others is discarded and not used in our recognizing system.

Table 3.An Example of Mel Frequency Cepstral Coefficients of Ship Sounds

	FRAME 981	FRAME 982	FRAME 983	FRAME 984	FRAME 985	FRAME 986	FRAME 987
MFCC (1)	-28.4411	-28.0989	-28.2988	-28.1981	-28.4815	-28.4116	-28.0461
MFCC (2)	1.8948	1.6840	1.0732	0.7244	0.4230	0.3493	1.0038
MFCC (3)	0.3257	0.3785	0.6085	1.0656	1.3821	1.5115	1.7103
MFCC (4)	-1.8113	-2.2666	-2.3777	-2.3735	-1.9281	-1.9823	-1.6427
MFCC (5)	0.1433	0.1567	0.0692	0.1732	0.2620	0.0304	0.1888
MFCC (6)	0.2245	0.2247	0.1930	0.0496	0.0606	0.0608	0.4477
MFCC (7)	0.3682	0.1489	0.0628	0.0834	0.0782	0.2342	0.1816
MFCC (8)	1.4558	1.7594	1.6377	1.6472	1.6466	1.5567	1.2380
MFCC (9)	0.6294	0.2068	0.5963	0.2552	0.3750	0.4788	0.1453
MFCC (10)	0.4890	0.3893	0.2730	0.0067	0.0488	0.3335	0.1324
MFCC (11)	0.7115	0.5707	0.4608	0.8073	1.0068	0.6002	0.7179
MFCC (12)	0.4792	0.5622	0.4512	0.1951	0.1341	0.3996	0.1794
MFCC (13)	0.2825	0.0400	0.1367	0.0127	0.0865	0.1087	0.1400

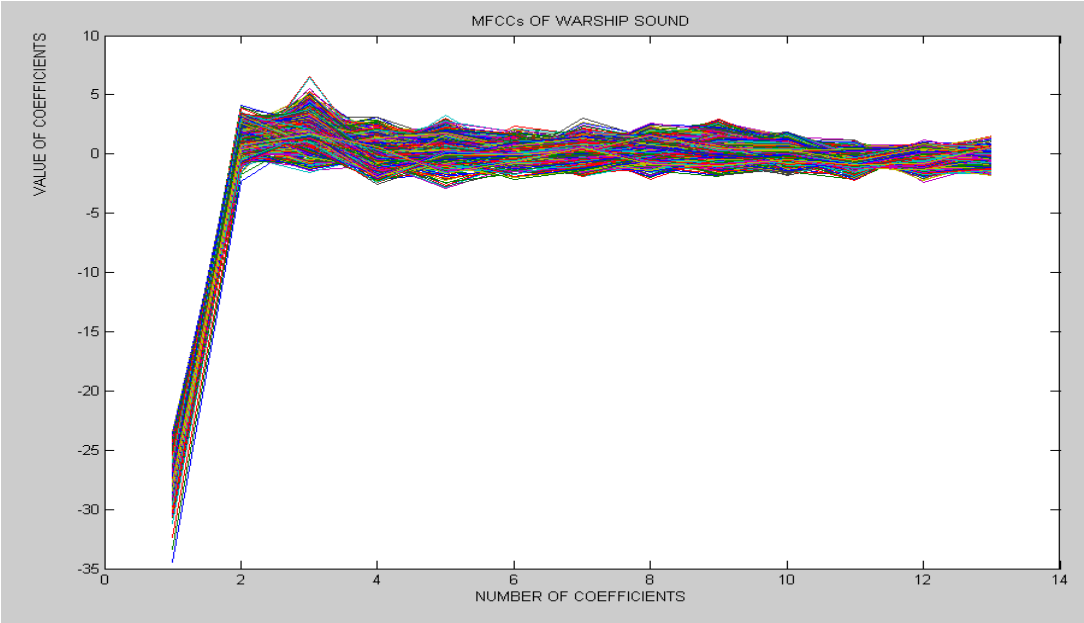

Figure 26.MFCC of Warship Sound

After extracting [13*1000] and [9*1000] MFCC and Delta-MFCC matrixes every matrix is quantized to increase within class similarity and between class distances by Vector Quantization method. By Vector Quantization execution we derived [13*64] coefficient matrixes. An Example of Vector Quantization is shown in Table 4.

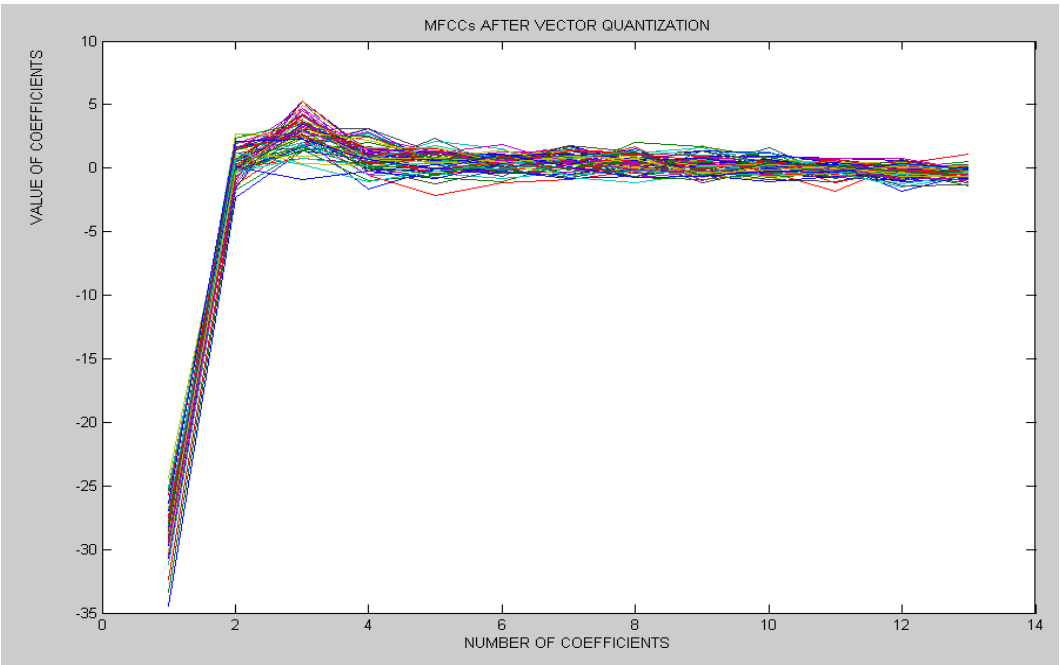

Figure 27.MFCC after VECTOR QUANTIZATION

Table 4.An Example of Vector Quantization

	FRAME 57	FRAME 58	FRAME 59	FRAME 60	FRAME 61	FRAME 62	FRAME 63
MFCC (1)	-26.9160	-28.0649	-27.4529	-26.8005	-29.3988	-28.7122	-28.5707
MFCC (2)	0.0750	0.0797	1.5242	-0.3105	1.9156	-0.2969	1.5947
MFCC (3)	-0.8573	2.5285	2.5710	0.8043	2.8487	3.7669	3.0074
MFCC (4)	-0.2670	2.5461	0.4059	0.1526	1.5658	1.6236	-0.2090
MFCC (5)	-0.8317	0.7176	-0.0396	0.3490	1.3680	1.6037	1.2612
MFCC (6)	0.3847	0.8431	0.4110	-0.1093	0.0783	0.4591	0.0919
MFCC (7)	1.3749	0.8562	1.3529	0.8105	0.6355	0.9803	0.5353
MFCC (8)	0.6302	1.1600	1.0753	1.3830	-0.1059	0.7155	-0.7665
MFCC (9)	1.3418	-0.5643	0.6128	1.0370	-0.2421	-0.3590	1.3365
MFCC (10)	1.2241	0.3711	-0.2556	1.2481	-0.4377	-0.1791	0.4748
MFCC (11)	0.3523	0.1545	0.7020	0.2899	0.1288	0.1084	0.1435
MFCC (12)	-1.8340	-0.2671	-0.1535	-1.4546	-0.9332	-0.6561	-1.4236
MFCC (13)	-0.4809	-0.5692	-0.5710	-0.5541	-0.7574	-0.6280	-1.3514

For each sound, quantized coefficients, except the first one, are combined to be able to represent whole sound which is shown in Table 5.

Table 5.Combining Coefficients

1*64	1*64	1*64	1*64	1*64	1*64	1*64	1*64	1*64	1*64	1*64	1*64
Coef 2	Coef 3	Coef 4	Coef 5	Coef 6	Coef 7	Coef 8	Coef 9	Coef 10	Coef 11	Coef 12	Coef 13

13 MFCC and Delta-MFCC Coefficients [1*(12*2^n)] = [1*768]

1*64	1*64	1*64	1*64	1*64	1*64	1*64	1*64
Coef 2	Coef 3	Coef 4	Coef 5	Coef 6	Coef 7	Coef 8	Coef 9

9 MFCC and Delta-MFCC Coefficients [1*(12*2^n)] = [1*512]

1*1	1*1	1*1	1*1	1*1	1*1
Coef 1	Coef 2	Coef 3	Coef 4	Coef 5	Coef 6

6 Pitch Coefficients [1*6]

C. RECOGNITION USING k-NEAREST NEIGHBOURHOOD AND SUPPORT VECTOR MACHINE

In this section we evaluate the results of our recognition tests for various test data. We realized 11 different tests that are shown in Table 6.

Table 6.WSSRS RECOGNITION METHODS

	RECOGNITION METHOD	FEATURES
Test 1	k-NN / Euclidean Distance	MFCC (8 Coefficients)
Test 2	k-NN / City-Block	MFCC (8 Coefficients)
Test 3	k-NN / Euclidean Distance	MFCC (12 Coefficients)
Test 4	k-NN / City-Block	MFCC (12 Coefficients)
Test 5	k-NN / Euclidean Distance	MFCC and Delta-MFCC (16(8+8) Coefficients) / Pitch (6 Coefficients)
Test 6	k-NN / City-Block	MFCC and Delta-MFCC (16(8+8) Coefficients) / Pitch (6 Coefficients)
Test 7	k-NN / Euclidean Distance	MFCC and Delta-MFCC (24(12+12) Coefficients) / Pitch (6 Coefficients)
Test 8	k-NN / City-Block	MFCC and Delta-MFCC (24(12+12) Coefficients) / Pitch (6 Coefficients)
Test 9	SVM / (Gaussian RBF)	MFCC (8 Coefficients)
Test 10	SVM / (Gaussian RBF)	MFCC (12 Coefficients)
Test 11	SVM / (Gaussian RBF)	MFCC and Delta-MFCC (24(12+12) Coefficients) / Pitch (6 Coefficients)

Experimental Setup

We have *18 ship classes* and *have* 5 different training data for each class. For k-NN classification; we used both Euclidean Distance and City-Block Methods as distance metrics and we choosed k=3 for 5 training data. For classification by SVM, our system classifies the test data in 5 steps. After extraction of MFCCs we formed the pairs for training. We have formed *153 pairs*

for training. We trained our system with these 153 pairs and obtained *1530 Support Vectors*. After training we have tested 110 test sounds. SVM makes 17 comparisons between pairs and test data at each test. Totally 1887 comparisons (17*110) have been made.

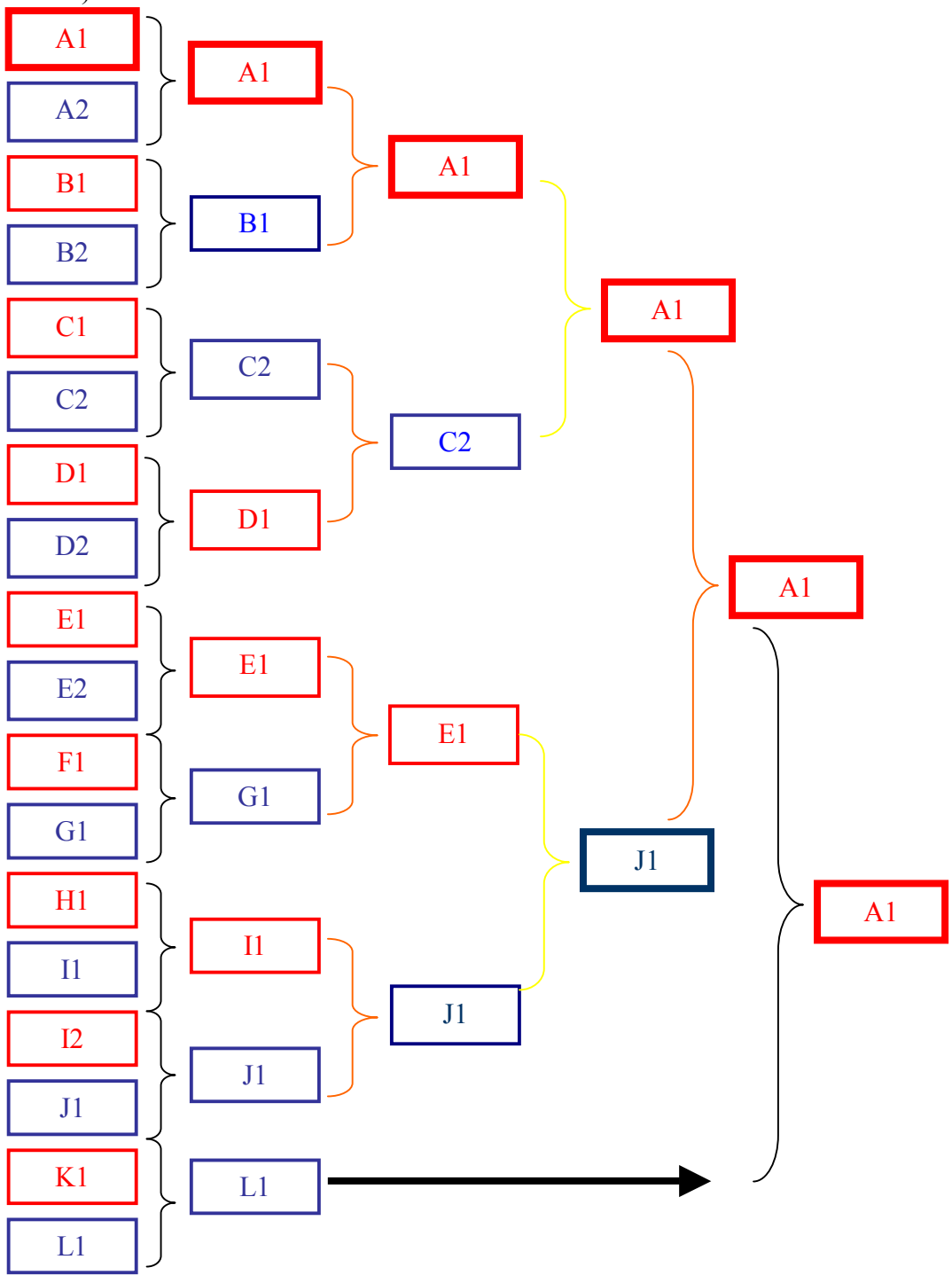

Figure 28.An Example of Recognizing a Test Data using SVM

An example of recognizing test data using SVM is shown in Figure 28. Red colored training sounds represent positive classes and blue colored training sounds represent negative classes.

Evaluation of Our WSSRS

In this section the recognition success of warships that belong to 18 classes is examined. The results of 11 different test scenarios using different features are shown in Tables 7-12 one bye one.

Table 7.k-NN/Euclidean and City-Block MFCC (8) Test Results

kNN / Euclidean MFCC (8)	SPEED DEPENDENT	SPEED INDEPENDENT	kNN/CityBlock MFCC (8)	SPEED DEPENDENT	SPEED INDEPENDENT
Class	NUMBER OF CORRECT RECOGNITION	NUMBER OF CORRECT RECOGNITION	Class	NUMBER OF CORRECT RECOGNITION	NUMBER OF CORRECT RECOGNITION
A1	3/4	3/4	**A1**	4/4	4/4
A2	4/8	4/8	**A2**	5/8	5/8
B1	4/8	4/8	**B1**	5/8	5/8
B2	3/7	6/7	**B2**	4/7	6/7
C1	8/8	8/8	**C1**	8/8	8/8
C2	6/6	6/6	**C2**	5/6	5/6
D1	7/7	7/7	**D1**	7/7	7/7
D2	8/12	10/12	**D2**	8/12	11/12
E1	3/4	3/4	**E1**	3/4	3/4
E2	0/4	0/4	**E2**	1/4	1/4
F1	3/4	3/4	**F1**	3/4	3/4
G1	4/5	4/5	**G1**	4/5	4/5
H1	2/4	2/4	**H1**	2/4	2/4
I1	10/10	10/10	**I1**	10/10	10/10
I2	2/5	4/5	**I2**	3/5	5/5
J1	1/4	1/4	**J1**	1/4	1/4
K1	6/6	6/6	**K1**	6/6	6/6
L1	0/4	0/4	**L1**	0/4	0/4

The best recognition rates are obtained from the classes C1, C2, D1, I1 and K1. The worst recognition rates are obtained from the classes J1 and L1.

Table 8.k-NN/Euclidean and City-Block MFCC (12) Test Results

kNN / Euclidean MFCC (12)	SPEED DEPENDENT	SPEED INDEPENDENT	kNN/CityBlock MFCC (12)	SPEED DEPENDENT	SPEED INDEPENDENT
Class	NUMBER OF CORRECT RECOGNITION	NUMBER OF CORRECT RECOGNITION	Class	NUMBER OF CORRECT RECOGNITION	NUMBER OF CORRECT RECOGNITION
A1	4/4	4/4	**A1**	3/4	3/4
A2	6/8	6/8	**A2**	6/8	6/8
B1	6/8	6/8	**B1**	8/8	8/8
B2	3/7	4/7	**B2**	4/7	5/7
C1	8/8	8/8	**C1**	8/8	8/8
C2	6/6	6/6	**C2**	5/6	5/6
D1	6/7	6/7	**D1**	7/7	7/7
D2	9/12	11/12	**D2**	9/12	11/12
E1	3/4	3/4	**E1**	3/4	3/4
E2	2/4	2/4	**E2**	1/4	1/4
F1	4/4	4/4	**F1**	4/4	4/4
G1	3/5	3/5	**G1**	3/5	3/5
H1	3/4	3/4	**H1**	4/4	4/4
I1	10/10	10/10	**I1**	10/10	10/10
I2	4/5	5/5	**I2**	2/5	5/5
J1	1/4	1/4	**J1**	1/4	1/4
K1	6/6	6/6	**K1**	6/6	6/6
L1	0/4	0/4	**L1**	0/4	0/4

Table 9.k-NN/Euclidean and City-Block MFCC (8), Delta MFCC (8) and Pitch Test Results

kNN / Euclidean DeltaMFCC(8)-MFCC(8)-Pitch	SPEED DEPENDENT	SPEED INDEPENDENT	kNN/CityBlock DeltaMFCC(8)-MFCC(8)-Pitch	SPEED DEPENDENT	SPEED INDEPENDENT
Class	NUMBER OF CORRECT RECOGNITION	NUMBER OF CORRECT RECOGNITION	Class	NUMBER OF CORRECT RECOGNITION	NUMBER OF CORRECT RECOGNITION
A1	3/4	3/4	**A1**	3/4	3/4
A2	5/8	7/8	**A2**	8/8	8/8
B1	1/8	1/8	**B1**	4/8	4/8
B2	3/7	3/7	**B2**	4/7	5/7
C1	7/8	7/8	**C1**	8/8	8/8
C2	2/6	2/6	**C2**	2/6	2/6
D1	3/7	4/7	**D1**	7/7	7/7
D2	3/12	4/12	**D2**	3/12	8/12
E1	2/4	4/4	**E1**	4/4	4/4
E2	1/4	2/4	**E2**	3/4	3/4
F1	1/4	1/4	**F1**	3/4	3/4
G1	3/5	3/5	**G1**	3/5	3/5
H1	2/4	2/4	**H1**	4/4	4/4
I1	0/10	0/10	**I1**	0/10	0/10
I2	1/5	1/5	**I2**	0/5	0/5
J1	0/4	0/4	**J1**	0/4	0/4
K1	0/6	0/6	**K1**	0/6	0/6
L1	0/4	0/4	**L1**	0/4	0/4

Table 10.k-NN/Euclidean and City-Block MFCC (12), Delta MFCC (12) and Pitch Test Results

kNN / Euclidean DeltaMFCC(12)-MFCC(12)-Pitch	SPEED DEPENDENT	SPEED INDEPENDENT	kNN/CityBlock DeltaMFCC(12)-MFCC(12)-Pitch	SPEED DEPENDENT	SPEED INDEPENDENT
Class	NUMBER OF CORRECT RECOGNITION	NUMBER OF CORRECT RECOGNITION	Class	NUMBER OF CORRECT RECOGNITION	NUMBER OF CORRECT RECOGNITION
A1	3/4	3/4	**A1**	3/4	3/4
A2	5/8	5/8	**A2**	8/8	8/8
B1	2/8	2/8	**B1**	4/8	4/8
B2	3/7	3/7	**B2**	3/7	4/7
C1	6/8	6/8	**C1**	8/8	8/8
C2	1/6	2/6	**C2**	2/6	2/6
D1	2/7	3/7	**D1**	7/7	7/7
D2	2/12	4/12	**D2**	4/12	9/12
E1	2/4	3/4	**E1**	4/4	4/4
E2	1/4	3/4	**E2**	2/4	2/4
F1	0/4	0/4	**F1**	3/4	3/4
G1	2/5	2/5	**G1**	3/5	3/5
H1	2/4	2/4	**H1**	4/4	4/4
I1	1/10	1/10	**I1**	0/10	0/10
I2	0/5	0/5	**I2**	0/5	0/5
J1	0/4	0/4	**J1**	0/4	0/4
K1	0/6	0/6	**K1**	0/6	0/6
L1	0/4	0/4	**L1**	0/4	0/4

Table 11.SVM MFCC (8) and SVM MFCC (12) Test Results

SVM MFCC (8)	SPEED DEPENDENT	SPEED INDEPENDENT	SVM MFCC (12)	SPEED DEPENDENT	SPEED INDEPENDENT
Class	NUMBER OF CORRECT RECOGNITION	NUMBER OF CORRECT RECOGNITION	Class	NUMBER OF CORRECT RECOGNITION	NUMBER OF CORRECT RECOGNITION
A1	3/4	3/4	**A1**	4/4	4/4
A2	5/8	5/8	**A2**	5/8	5/8
B1	4/8	4/8	**B1**	7/8	7/8
B2	3/7	6/7	**B2**	5/7	5/7
C1	8/8	8/8	**C1**	8/8	8/8
C2	6/6	6/6	**C2**	6/6	6/6
D1	5/7	5/7	**D1**	6/7	6/7
D2	8/12	10/12	**D2**	9/12	11/12
E1	3/4	3/4	**E1**	3/4	3/4
E2	0/4	0/4	**E2**	1/4	1/4
F1	3/4	3/4	**F1**	3/4	3/4
G1	3/5	3/5	**G1**	4/5	4/5
H1	3/4	3/4	**H1**	4/4	4/4
I1	10/10	10/10	**I1**	10/10	10/10
I2	4/5	5/5	**I2**	4/5	5/5
J1	0/4	0/4	**J1**	1/4	1/4
K1	6/6	6/6	**K1**	6/6	6/6
L1	1/4	1/4	**L1**	0/4	0/4

Table 12.SVM MFCC (12), Delta MFCC (12) and Pitch Test Results

SVM DeltaMFCC(12) MFCC(12) Pitch	SPEED DEPENDENT	SPEED INDEPENDENT
Class	NUMBER OF CORRECT RECOGNITION	NUMBER OF CORRECT RECOGNITION
A1	2/4	2/4
A2	6/8	6/8
B1	3/8	3/8
B2	7/7	7/7
C1	7/8	7/8
C2	6/6	6/6
D1	5/7	6/7
D2	5/12	9/12
E1	4/4	4/4
E2	2/4	2/4
F1	0/4	0/4
G1	3/5	3/5
H1	3/4	3/4
I1	9/10	9/10
I2	3/5	4/5
J1	1/4	1/4
K1	4/6	4/6
L1	2/4	2/4

In Table 13, the classification accuracy of different features and different recognition methods is presented.

Table 13.WSSRS TEST results

	SPEED DEPENDENT		SPEED INDEPENDENT	
TEST	NUMBER OF CORRECT RECOGNITION	RATIO OF CORRECT RECONITION	NUMBER OF CORRECT RECOGNITION	RATIO OF CORRECT RECONITION
k-NN / Euclidean MFCC (8)	74/110	67%	81/110	73%
k-NN / CityBlock MFCC (8)	79/11	71%	86/110	77%
k-NN / Euclidean MFCC (12)	84/110	76%	88/110	79%
k-NN / City-Block MFCC (12)	88/110	79%	89/110	80%
k-NN / Euclidean MFCC/DMFCC (8) and PITCH	36/110	32%	41/110	37%
k-NN / City-Block MFCC/DMFCC (8) and PITCH	56/110	50%	62/110	56%
k-NN / Euclidean MFCC/DMFCC (12) and PITCH	32/110	29%	39/110	35%
k-NN / City-Block MFCC/DMFCC (12) and PITCH	52/110	47%	61/110	55%
SVM MFCC (8)	74/110	64%	80/110	72%
SVM MFCC (12)	***88/110***	***79%***	***91/110***	***82%***
SVM MFCC/DMFCC (12) and PITCH	77/110	69%	66/110	59%

CONCLUSION

The Sea warfare requires correct target detection and classification. The eyes of submarines and surface ships are currently sonars and human sonar operators. But these eyes can only estimate the class of a target according to their experiences and this type of recognition can not be acceptable for warfare.

Our Ship Sound Source Recognition System solves this problem and creates a new approach in sound recognition systems that the system we present successfully recognizes the sound source. We consider that this new approach will also mark an epoch in tactical underwater and surface warfare.

In conclusion, the performance evaluation of SVM and k-NN methods with MFCC, Delta MFCC and Pitch features is stated below;

- The best result (82%) is obtained from SVM (Gaussian Kernel) with MFCC 12 coefficients.
- The worst result (29%) is obtained from k-NN/Euclidean with MFCC/DMFCC (12) and Pitch feature set.
- SVM (Gaussian Kernel) gives more successful results than k-NN method.
- Speed Independent results are more successful than Speed Dependent results.
- MFCC (12 coefficients) gives the best results (82% with SVM and 80% with k-NN).
- In k-NN recognition, the City Block distance metric is more successful than the Euclidean.
- MFCC obtains optimum success with 12 coefficients.
- Pitch features decrease the recognition rate between 10%-47%.

Because of the weakness of our training and test datasets, number of tests we have done is not enough to make generalization. To increase the success ratio in real time applications our system should be trained and tested with quite more sound data that are recorded in different conditions.

SSRS can be adapted to the surface ships, submarines and field defence systems as a target recognizing system for tactical underwater and surface warfare. Warship SSRS can be adapted to Fire Support Systems as a target detection and identification system. Also our SSRS can be adapted to the underwater sensor networks.

LIST OF REFERENCES

1. Antti Eronen and Anssi Klapuri, *Musical Instrument Recognition Using Cepstral Coefficients and Temporal Features,* (2001)

2. Tero Tolonen, *Object-Based Sound Source Modeling for Musical Signals*, (2001)

3. Ian Antti Eronen, *Automatic Musical Instrument Recognition,* (2001)

4. Keith D. Martin, *Toward Automatic Sound Source Recognition: Identifying Musical Instruments*, (1998)

5. Kaminskyj and Materka, *Automatic Source Identification of Monophonic Musical Instrument Sounds,* (1995)

6. Nooralahiyan, A. Y., Kirby, H. R. & McKeown, *Vehicle classification by acoustic signature,* (1998)

7. Huadong Wu, Mel Siegel and Pradeep Khosla, *Vehicle Sound Signature Recognition by Frequency Vector Principal Component Analysis,* (1998)

8. Naoaki Suetsugi, Hideki Nakamichi and Masahiro Toya, *A New Traffic Incident Detector By Processing The Traffic Sounds,*(1997)

9. Aadra Adrid, Jean Paul Barjaktarevic, Ozgu Ozun, Michael Smith and Philipp Steurer, *Automatic Speech Recognition for Isolated Words,*

10. Ling Feng, *Speaker Recognition,* (2004)

11. Douglas A. Reynolds, Larry P. Heck, *Automatic Speaker Recognition,* (2000)

12. James G. Droppo, *Time-Frequency Features For Speech Recognition*, (2000)

13. Case Reynolds, *Speaker identification and verification using Gaussian mixture speaker models,* (1995)

14. Ben J. Shannon and Kuldip K. Paliwal, *A Comparative Study of Filter Bank Spacing for Speech Recognition*, (2003)

15. Digital Filter Design, http://www.phon.ucl.ac.uk/courses/spsci/dsp/filter.html (2005)

16. Windowing, (2005) http://www.cg.tuwien.ac.at/studentwork/CESCG/CESCG99/TTheussl/index.html

17. Ziyou Xiong, Regunathan Radhakrishnan, Ajay Divakaran and Thomas S. Huang, *Comparing MFCC and MPEG-7 Audio Features for Feature Extraction, Maximum Likelihood HMM and Entropic Prior HMM for Sports Audio Classification*, (2003)

18. Keith D. Martin, *Sound-Source Recognition: A Theory and Computational Model*, (1999)

19. Philip Jackson, *Feature extraction1*, http://www.ee.surrey.ac.uk/Teaching/Courses/eem.ssr/ (2005)

20. M. Qasem. Vector Quantization. Introductory article. http://www.geocities.com/mohamedqasem/vectorquantization/vq.html (2005)

21. Steve R. Gunn, *Support Vector Machines for Classification and Regression,* (1998)

22. Chih-Wei and Chih-Jen Lin, *A Comparison of Methods for Multi-Class Support Vector Machinas*, (2003)

23. Antti Eronen, *Sound Source Recognition and Modeling,* (2000)

24. Case Gaunard, *Automatic Noise Recognition,* (1998)

25. The MathWorks (1999)

26. ISMIR Graduate School, Audio signal classification, (2004)

27. A. Eronen, *Comparison Of Features For Musical Instrument Recognition*, (2001)

28. Digital Signal Processing, http://www.phon.ucl.ac.uk/courses/spsci/dsp/

29. Swati Rastogi and David Mayor, *an Automatic Speaker Recognition System*, (2000)

30. Mark D. Skowronski and John G. Haris, *Increased MFCC Filter Bandwidth for Noise-Robust Phoneme Recognition,*

31. K.K. Chin, *Support Vector Machines applied to Speech Pattern Classification*

32. J. J. Burred and A. Lerch, *Hierarchical Automatic Audio Signal Classification*, (2004)

33. Asano, *Pattern information processing*, (2002)

34. David Gerhard, *Audio Signal Classification: History and Current Techniques, (2003)*