Universidade Nova de Lisboa

Faculdade de Ciências e Tecnologia

Departamento de Informática

Analysis and recognition of similar environmental sounds

José Rodeia, nº 26657

Dissertação apresentada na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa para obtenção do grau de Mestre em Engenharia Informática

Orientador

Prof. Doutora Sofia Cavaco

Lisboa

28 de Julho de 2009

Nº do aluno: 26657

Nome: José Pedro dos Santos Rodeia

Título da dissertação:

Analysis and recognition of similar environmental sounds

Key Words:

- Sound Recognition
- Sound Classification
- Sound Feature Analysis
- Independent Component Analysis (PCA)
- Principal Component Analysis (ICA)

Resumo

Os seres humanos conseguem identificar uma fonte sonora baseando-se apenas no som que produz. O mesmo problema pode ser adaptado a computadores. Vários reconhecedores de som foram desenvolvidos durante a última década. A sua eficácia reside nas propriedades dos sons que são extraídas e no método de classificação implementado. Existem várias abordagens a estes dois tópicos mas a maioria destina-se a reconhecer sons com características muito diferentes.

Esta dissertação apresenta um reconhecedor de sons semelhantes. Como os sons têm propriedades muito parecidas o processo de reconhecimento torna-se mais difícil. Assim, o reconhecedor usa tanto as propriedades temporais como as espectrais dos sons. Para extrair estas propriedades usa o método Intrinsic Structures Analysis (ISA), que usa, por sua vez, Independent Component Analysis e Principal Component Analysis. O método de classificação implementado usa o algoritmo k-Nearest Neighbor.

Os testes desenvolvidos permitem-nos concluir que estas propriedades são bastante eficazes em reconhecimento de som. Testámos o nosso reconhecedor com vários conjuntos de propriedades extraídas pelo método ISA obtendo óptimos resultados. De forma a comparar a capacidade humana com a do nosso reconhecedor fizemos um user study concluindo que os sons são de facto muito semelhantes e muito mais difíceis de identificar para um ser humano do que para o nosso reconhecedor.

Abstract

Humans have the ability to identify sound sources just by hearing a sound. Adapting the same problem to computers is called (automatic) sound recognition. Several sound recognizers have been developed throughout the years. The accuracy provided by these recognizers is influenced by the features they use and the classification method implemented. While there are many approaches in sound feature extraction and in sound classification, most have been used to classify sounds with very different characteristics.

Here, we implemented a similar sound recognizer. This recognizer uses sounds with very similar properties making the recognition process harder. Therefore, we will use both temporal and spectral properties of the sound. These properties will be extracted using the Intrinsic Structures Analysis (ISA) method, which uses Independent Component Analysis and Principal Component Analysis. We will implement the classification method based on k-Nearest Neighbor algorithm.

Here we prove that the features extracted in this way are powerful in sound recognition. We tested our recognizer with several sets of features the ISA method retrieves, and achieved great results. We, finally, did a user study to compare human performance distinguishing similar sounds against our recognizer. The study allowed us to conclude the sounds are in fact really similar and difficult to distinguish and that our recognizer has much more ability than humans to identify them.

List of abbreviations

Intrinsic Structures Analysis	ISA
Independent Component Analysis	ICA
Principal Component Analysis	PCA
k-Nearest Neighbor	k-NN
Gaussian Mixture Models	GMM
Hidden Markov Models	HMM
Discrete Fourier Transform	DFT
Discrete Wavelet Transform	DWT
Discrete Cosine Transform	DCT
Short-Time Fourier Transformation	STFT
Zero-Crossing Rate	ZCR
Mel-frequency cepstral coefficients	MFCCs
Wavelet Transform	WT
Fast Fourier Transform	FFT
Recurrent Neural Network	RNN
Multi-Layer Perceptron Network	MPN
Nearest Feature Line	NFL
Support Vectors Machine	SVM

Index

1. Introduction………………………………………………………… 12

2. Audio Processing Theory review…………………...………………… 16

3. State of Art…………………………………………………………….. 22

3.1 Sound Feature Extraction………………………………………………. 22

3.2 Sound Classification……………………………………………………. 32

4. Proposed Solution……………………………………………………… 42

5. Implementation………………………………………………………… 50

6. Result Analysis…………………………………………………………. 60

7. User Study……………………………………………………………… 68

8. Conclusions…………………………………………………………….. 77

9. Bibliography…………………………………………………………… 80

List of Figures

Figure 2.1 A sound wave example 16

Figure 2.2 Spectrogram from a violin playing two notes 19

Figure 2.3 Three-level wavelet decomposition tree 20

Figure 3.1 The three different type features of the same sound 24

Figure 3.2 Accuracy results using MFCC’s 27

Figure 3.3 Accuracy results using Auditory Filterbanks 27

Figure 3.4 Accuracy results using wavelet features 28

Figure 3.5 Sound classes division for multi-class classification using SVM 35

Figure 3.6 Features’ recognition rate with two different classifiers 37

Figure 3.7 Recognition rates of 10 contexts using HMM 39

Figure 3.8 Comparison between computer and human recognizing contexts 40

Figure 4.1 The three steps of an environmental sound recognizer 42

Figure 4.2 The three steps of the proposed sound recognizer 43

Figure 4.3 Temporal analysis with ICA from a wood sound 46

Figure 4.4 Spectral analysis with ICA from a wood sound 47

Figure 4.5 Classification scheme of our recognizer 49

Figure 5.1 The spectrogram of one of the sounds made by a steel bar 51

Figure 5.2 Example of a matrix of spectrograms for temporal analysis 52

Figure 5.3 Example of a matrix of spectrograms for spectral analysis 52

Figure 5.4 ICA feature obtained with temporal analysis 54

Figure 5.5 ICA feature obtained with spectral analysis 54

Figure 5.6 PCA feature obtained with temporal analysis 55

Figure 5.7 PCA feature obtained with spectral analysis 55

Figure 5.8 Features extracted with PCA for spectral analysis 56

Figure 5.9 Temporal feature extracted with ICA of two different sounds 58

Figure 5.10 Spectral feature extracted with ICA of two different sounds 59

Figure 6.1 Temporal analysis with ICA results 61

Figure 6.2 Spectral analysis with ICA results 61

Figure 6.3 Temporal analysis with PCA results 63

Figure 6.4 Spectral analysis with PCA results 63

Figure 6.5 Results using MFCCs 65

Figure 6.6 Overall recognition rates provided by all the analyses made 66

Figure 7.1 Dialog box for the first listening samples 69

Figure 7.2 Dialog box to listen to the train and test samples 70

Figure 7.3 Dialog box to identify the material that caused the heard sound 70

Figure 7.4 Dialog box with the right answer during the training samples 71

Figure 7.5 Comparison of the recognition rates obtained by two user studies 76

List of Tables

Table 3.1 The error rate of some short-time features 25

Table 3.2 Accuracy obtained with MFCC’s and with MPEG-7 descriptors 30

Table 3.3 Results with and without ICA transformation 30

Table 3.4 Error rates of feature extraction with ICA against MFCCs 31

Table 3.5 Accuracy rates of the recognizer using FilterBank and k-NN 33

Table 3.6 Error Rates with MLP network and with Bayesian Classifier 34

Table 3.7 Error rates retrieved by four classification methods 35

Table 3.8 Accuracy of multi-class recognizer using SVM 36

Table 7.1 The table of answers of our first user study 72

Table 7.2 The table of answers of our second user study 73

1. Introduction

After hearing a sound, normally, humans are able to distinguish what caused it. For instance, we can recognize someone only by hearing his voice on the phone and we are able to distinguish our cell phone’s ring from the others. Over the years the possibility of computers doing the same has been studied. This is called (automatic) sound recognition.

The potentialities of a sound recognizer are vast. Systems that actuate according to the recognized sound can improve our quality of life as they can help doing and even do activities for us (like turning lights on when someone claps, calling 112 when someone cries for help, etc.). There are two main types of sound recognizers: those that recognize the words in voice messages and those that recognize the sound sources.

The most common sound recognition systems studied and developed are *speech recognizer*s, such as those in cellular phones, cars, etc. The goal of a speech recognizing system is to recuperate the messages contained in the sound wave. This is done matching the tested samples with letters, combinations of letters and word samples.

From another point of view, there are sound recognizers that focus on finding the source that produced the sound, namely *environmental sound recognizers.* Most of these recognizers distinguish different sources of sounds like door bells, keyboards or whistles. Often, such recognizers can rely in the temporal signatures of the sounds because these sounds have very different temporal characteristics as they are produced not only by different objects but also by different events: there are differences in the properties of the objects (shape, size, material), the nature of the event (impact, ring, speech, etc.) and its characteristics (force, location of impact, etc.). On the other hand, the sounds used in this dissertation will be very similar to each other, which results in a more difficult problem and relying in their temporal signatures may not be enough.

There are some examples of environmental sound recognizers that show how recognizing a sound source can be useful in mobile computing devices. In [1], sound classification methods to distinguish moving ground vehicle sounds (from different cars, trucks, SUVs and mini vans) are tested to use in a wireless sensor network. In [2], the main goal is to show that using audio, motion and light sensors can improve context awareness computing for mobile devices and artifacts. [3] shows how distinguishing sounds is also essential to multimedia information retrieval systems.

Another good example of these potentialities is the importance of sound recognition in humanoid robots. A humanoid robot actuates according to the data sensed and could do a lot of everyday activities helping to improve our quality of life [4, 5]. It can obey to someone's instructions or react according to the sensed sound (like a beep or an alarm).

While most environmental sound recognizers use sounds with different temporal signatures, here the sounds are produced by the same event, and consequently they will have very similar temporal signatures. We focus on the recognition of impact sounds caused by objects with the same size and shape which only differ on the materials. Since our sounds are so similar, we name the proposed recognizer a *similar sound recognizer*.

We test our recognizer with both temporal and spectral features of the sounds. There are many proposals of how to extract these features and what temporal and spectral features to use. We use the Intrinsic Structures Analysis method [6], which in turn uses Independent Component Analysis (ICA) and Principal Component Analysis (PCA), to do that. Further, we compare the results retrieved using temporal and spectral signatures.

Using the features extracted we define the classes of sounds, one for each of the object we use. Then we can implement the classification system: an application that matches a test sample with one of those classes of sounds. Many classification methods have been studied in order to find the one which can ensure better recognition results. Until today, the most useful methods use a k-Nearest Neighbor algorithm or implement Gaussian Mixture Models (GMM) or Hidden Markov Models (HMM) to model the existing classes of sounds. We use the 1-NN algorithm in our classification system.

Section 2 reviews the basic theory of audio digitization and processing, while section 3 describes previous work done in environmental sound recognition. We propose our solution in section 4 and discuss its implementation in section 5. Then we analyze the results in section 6. We did a user study to compare the abilities of our recognizer with humans' ability classifying the sounds used in this dissertation. We discuss this user study in section 7. We take conclusions in section 8.

2. Audio Processing Review

In this section we review the theory of audio processing by computers: how we convert the audio signal from analog to digital and how we can process the digital representation of a sound. These topics can be extended in [7], [8], [9] and [10].

A sound is generated when something causes a disturbance in the density of gas, liquid or solid. These disturbances can be represented graphically as illustrated in figure 2.1. When those disturbances reach humans' ears, the brain converts them into electric signals that travel along the brain and allow us to recognize sounds (as well as localize sounds sources, etc.).

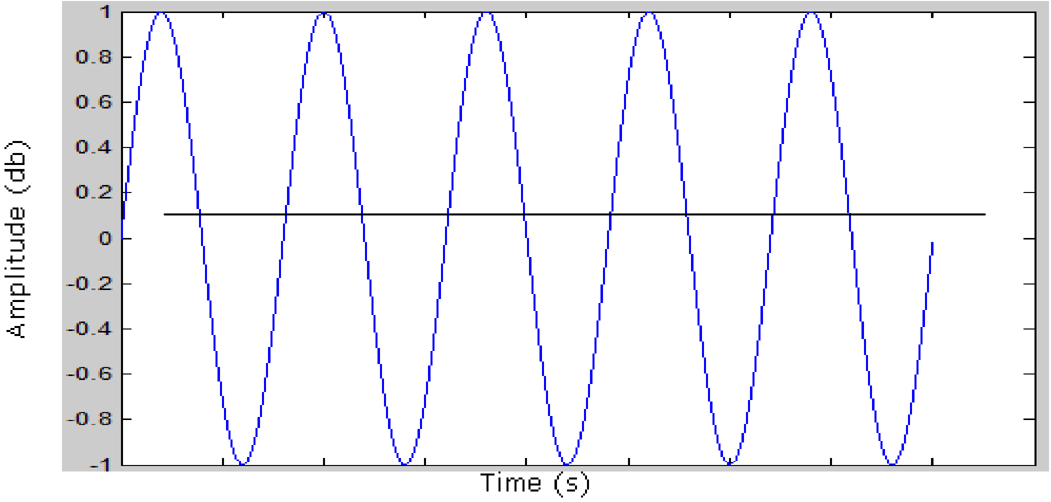


Figure 2.1: A sound wave example. The signal can be represented as a waveform, where each instant of time t has an amplitude value $x(t)$.

Our goal is to process these audio signals in computers as we want to build a sound recognizer. Therefore, we have to see how we create the digital audio signal. Converting the audio signal from analog to digital, involves two processes: sampling and quantization. Sampling consists in storing the values of the wave at certain time instants. Each value stored is a sample point. After sampling is done, it returns a finite set of values. These values are rounded and converted to a k bit number. This is called quantization. Using a low number of bits can produce higher quantization errors as the true amplitude values are further from their digital representation. By combining these processes it is possible to transform audio into a binary format that can be used in the computer.

After the sound wave is digitized it can be processed in order to obtain more information about the sound. This section discusses some sound processing techniques: Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT).

Fourier Transform converts a one variable function in another variable function. It is used to transform the time-varying waveform obtained after digitizing the sound into a frequency-varying spectrum more convenient to study the sound properties.

If the input function is discrete and its non-zero values have a limited duration we can call this operation Discrete Fourier Transform (DFT). DFT transforms the time domain amplitudes sequence $x_0, x_1, \dots, x_{N-1}$ into the frequency domain amplitudes sequence $X_0, X_1, \dots, X_{N-1}$ as we can see in equation 2.1 where N is the number of sampled points.

$$X_k = \sum_{k=0}^{N-1} x_k * e^{-\frac{2\pi i}{N}kn} \quad ,k = 0, \dots, N-1 \qquad (eq\ 2.1)$$

This shows that sounds can be represented as a sum of N sinusoidal components. These components describe the spectral characteristics of the sound and they can be used in order to recognize sound sources.

However, we are interested in studying the sound in detail and Fourier transform may not be enough to detect highly localized information. Instead of applying the Fourier transform to the whole signal, we will cut the signal $x(t)$ into different sections, or *windows*, with a time based function and apply the DFT to each section. This process is called Short-Time Fourier Transform (STFT) and can be defined in equation 2.2 with N being the number of sampled points and where $w(t)$ is the time based function (*window function*) and m and ω are the instants of time and frequencies respectively.

$$STFT\{x(t)\} = X(m, \omega) = \sum_{k=0}^{N-1} x(k) * w(k-m) * e^{-\omega k i} \qquad (eq\ 2.2)$$

The result of STFT can be represented as a matrix S where a_{ft} is the amplitude at instant $t \in \{0, \dots, T\}$ and frequency $f \in \{0, \dots, F\}$ (equation 2.3).

$$S = \begin{bmatrix} a_{00} & \cdots & a_{0T} \\ \vdots & & \vdots \\ a_{F0} & \cdots & a_{FT} \end{bmatrix} \qquad (eq\ 2.3)$$

A graphical display of the magnitude of this matrix is called a spectrogram. In this graphic, the horizontal axis represents time and the vertical axis represents frequency. We can see the temporal evolution of each frequency bin in the horizontal axis. The amplitude for each instant of time and frequency bin is given by the intensity or the color of each point in the image. For instance, figure 2.2 shows a spectrogram of a sound sampled from a staccato violin sound. The rows in the image are the frequency bins (higher frequencies have shorter duration). The brightness of the image points indicates the amplitude values (higher amplitude if brighter).

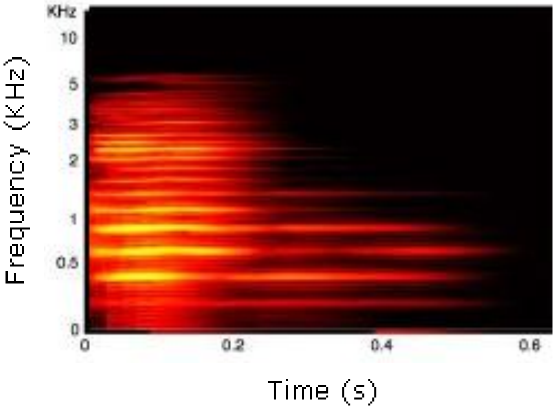


Figure 2.2: Extracted from [11], spectrogram from a staccato violin sound.

There is a tradeoff between the temporal and spectral representation given by the STFT related to the window function. A long window length provides a more detailed frequency representation. On the other hand, a short window provides a more detailed temporal representation and is, therefore, more appropriate for a time-analysis of the sound.

One alternative to STFT is the Discrete Wavelet Transform (DWT). Its purpose is to decompose the sound wave function recurrently in more scaled functions. To obtain the DWT of a signal $x(t)$, the signal has to be passed through a series of filters with different cutoff frequencies at different scales. The choice of the filters has to guarantee a perfect reconstruction of the sound wave from the coefficients extracted.

A one level transform of a signal $x(t)$ would be given by equations 2.4 and 2.5 where $h(t)$ and $g(t)$ are the high and low filter respectively.

$$y_{detail}\,(t) = (x * h)(t) = \sum_{k=-\infty}^{\infty} x(k) * h(2n - k) \qquad (eq2.4)$$

$$y_{approximate}\ \ (t) = (x * g)(t) = \textstyle\sum_{k=-\infty}^{\infty} x(k) * g(2n - k) \quad (eq\ 2.5)$$

For instance, we can see a three-level wavelet decomposition tree in figure 2.3. $X[n]$ is the sound signal, H_0 the high filter and $G_0$the low filter. $\downarrow 2$ is the operation that retrieves the detail coefficients $y_{detail}\,(t)$ $(d_1[n], d_2[n], d_3[n])$ and the approximate coefficients $y_{app\,roxmate}\ \ (t)$ $(a_3[n])$.

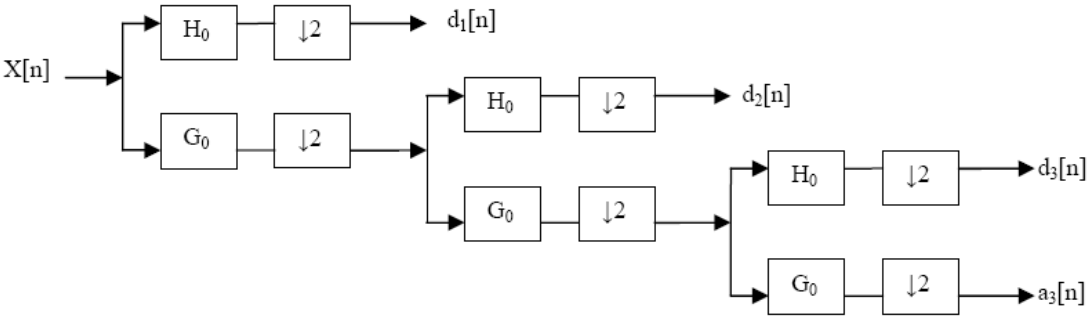


Figure 2.3: Three-level wavelet decomposition tree.

Other sound transformation functions exist. An also popular one is the Discrete Cosine Transform (DCT) which defines the sound wave as a sum of sinusoids with different amplitudes and frequencies. The difference to the DFT is that it only uses cosine functions.

The points $x_0, x_1, \dots, x_{N-1}$ of the signal are transformed into $X_0, X_1, \dots, X_{N-1}$ as shown by equation 2.6 where N is the number of sampled points.

$$X_k = \sum_{k=0}^{N-1} x_n * \cos\left[\frac{\pi}{N}\,(n + \frac{1}{2})(k + \frac{1}{2})\right], k = 0, \dots, N-1 \qquad (eq\ 2.6)$$

3. State of Art

There are two main factors that change the accuracy rate in sound recognition: the sound features used and the classification algorithm implemented. This section is divided in two parts: one that describes the progress in sound features extraction and another that describes the classification methods.

Sound Features Extraction

A sound can be described by temporal and spectral properties. These properties are the features sound recognizers use to classify the sounds. It is possible to extract several features from the sounds. We can divide these features in three categories: features derived from volume contour, features derived from pitch contour and frequency domain features [12].

Features derived from volume contour describe the temporal variation of the sound's magnitude. The volume value depends on the recording process. However, the temporal variation of its value reflects properties of the sound useful for sound recognition. Features derived from pitch contour describe the fundamental period of the sound. As an instance, these features can be used to distinguish voice samples from music samples which, usually, have a longer period than speech samples. Finally, there are frequency domain features which describe the frequency variation of the sound. These features can be extracted with Fourier Transform.

We can see these three different feature types in figure 3.1. This figure shows the variation of the volume, the pitch and the frequency centroid of a sound sample recorded from a news television show. As mentioned above, these features describe properties of the sound. For instance, the volume variation shows the speaker talks always at approximately the same volume and allows identifying silence intervals when volume decreases to zero. The pitch variation shows when the speaker is talking slower or faster if the values are lower or higher respectively. We can also detect the silence intervals using the frequency centroid variation. Other sources can also be identified by inspecting this feature, for instance we can identify a source different from the speaker by noting a peak in this feature.

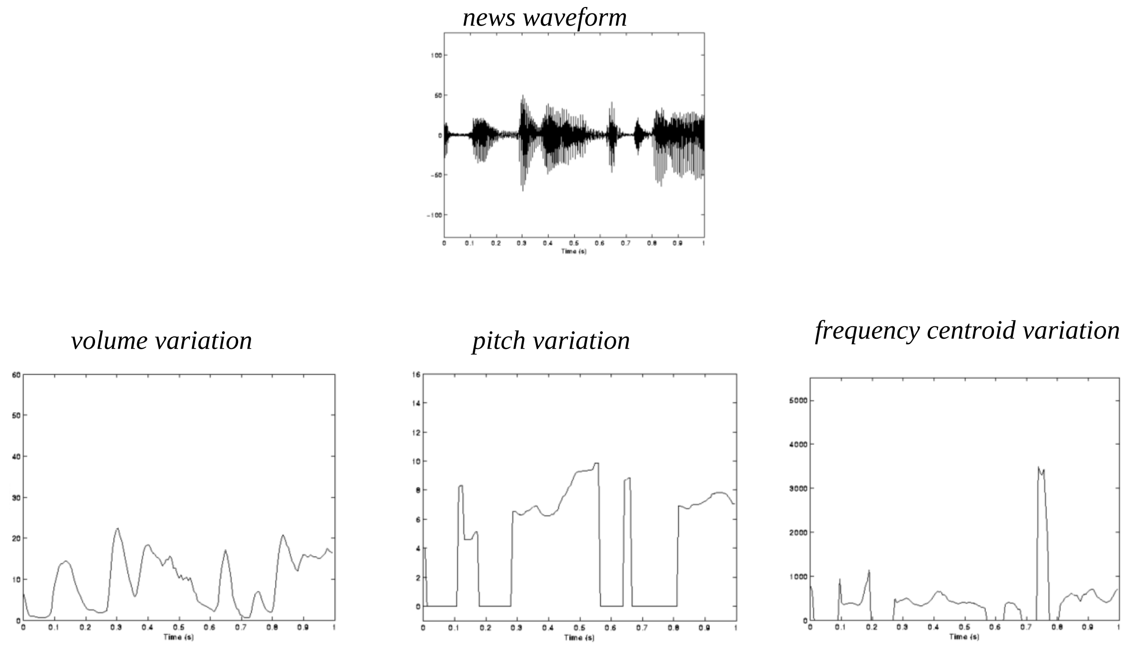


Figure 3.1: Extracted from [12], three different type features of the same sound. On top the waveform of the sound. On bottom from left to right: the volume variation, the pitch variation and the frequency centroid variation of the sound.

There are several algorithms to extract these features [13]. Even though, there are several approaches in audio feature extraction, most of them compute spectrograms to extract the features. Consequently, these are short-time features. However, selecting what features to use in sound recognition is not a closed topic.

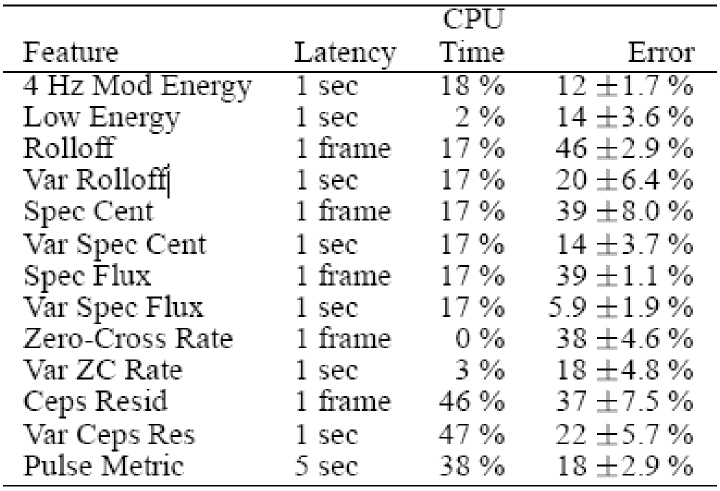
There are several proposals of what features to use. Scheirer *et al.* developed a sound recognizer [14] which used a set of 8 features:

- 4 Hz modulation energy
- Percentage of “Low-Energy” Frames
- Spectral Rolloff Point

- Spectral Centroid
- Spectral “Flux” (Delta Spectrum Magnitude)
- Zero-Crossing Rate (ZCR)
- Cepstrum Resynthesis Residual Magnitude
- Pulse metric

Analyzing some of these features may make other good features. The variances of the derivative of Spectral Rolloff Point, of Spectral Centroid, of Spectral “Flux”, of Zero-Crossing Rate and of Cepstrum Resynthesis Residual Magnitude can be used as features, too.

As described in Table 3.1, each feature can be tested to find its usefulness. Although some features alone show a big error rate (for instance, note that Spectral Rolloff Point has an error rate of almost 50%), using the features all together provided great results (around 90% accuracy).



Feature	Latency	CPU Time	Error
4 Hz Mod Energy	1 sec	18 %	12 ±1.7 %
Low Energy	1 sec	2 %	14 ±3.6 %
Rolloff	1 frame	17 %	46 ±2.9 %
Var Rolloff	1 sec	17 %	20 ±6.4 %
Spec Cent	1 frame	17 %	39 ±8.0 %
Var Spec Cent	1 sec	17 %	14 ±3.7 %
Spec Flux	1 frame	17 %	39 ±1.1 %
Var Spec Flux	1 sec	17 %	5.9 ±1.9 %
Zero-Cross Rate	1 frame	0 %	38 ±4.6 %
Var ZC Rate	1 sec	3 %	18 ±4.8 %
Ceps Resid	1 frame	46 %	37 ±7.5 %
Var Ceps Res	1 sec	47 %	22 ±5.7 %
Pulse Metric	5 sec	38 %	18 ±2.9 %

Table 3.1: Extracted from [14], the error rate of the selected short-time features.

Other recognizer using short-time features was proposed by Eronen *et al* [15]. This recognizer used and tests separately a set of 10 features:

- Zero-crossing rate
- Short-time average energy
- Band-energy ratio
- Spectral Centroid
- Bandwidth
- Spectral roll-off point
- spectral flux
- Linear prediction coefficients,
- Cepstral Coefficients
- Mel-frequency cepstral coefficients (MFCC)

When the features were tested, MFCCs provided the best recognition rate (above 60%). This is not the only example where MFCCs out-perform other features. In fact, MFCCs are very popular in sound recognition. They consist of a set of coefficients that make a short-term power spectrum of a sound, just like what is done by the Fourier Transform. However, while the Fourier Transform uses the Hertz scale, MFCCs use the Mel scale, which has frequency bands that are not equally spaced (in similarity to what happens in the human ear).

One alternative to the MFCCs consists of extracting a set of features called Auditory Filterbank temporal envelopes. These features are representations of the sound processed as it is processed in the human auditory system. In order to extract these features the sound has to pass through filters designed to imitate the frequency resolution of human hearing.

These features were tested in a recognizer [16] which has the purpose of distinguishing scenes (popular music, classic music, speech, noise and crowd). The results showed they could out-perform MFCCs: Figures 3.2 and 3.3 show the classification result for each scene using MFCCs and the classification results using Filterbank envelopes respectively. We can see they provide better results in classes more related to speech sounds (noise, crowd). In classical music, MFCCs still provide much better results than these features.

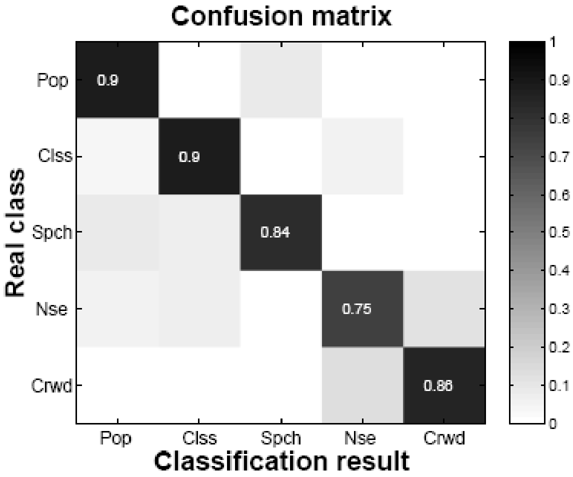


Figure 3.2: Extracted from [16], accuracy results provided by MFCCs.

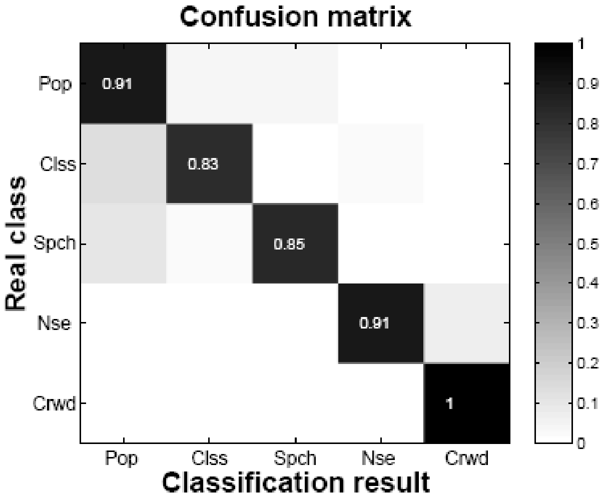


Figure 3.3: Extracted from [16], accuracy results provided by Auditory Filterbanks.

Other works process audio using Wavelet Transform (WT) instead of Fourier Transform. This results in different features as STFT transforms the sound in a spectrogram and WT decomposes the sound in various functions, as mentioned before in section 2. These features are extracted from these functions and were tested against MFCCs and short-time features. Figure 3.4 shows how MFCCs are more useful in sound recognition. While short-time features are comparable to the wavelet features, MFCCs retrieves higher recognition rates in every class tested.

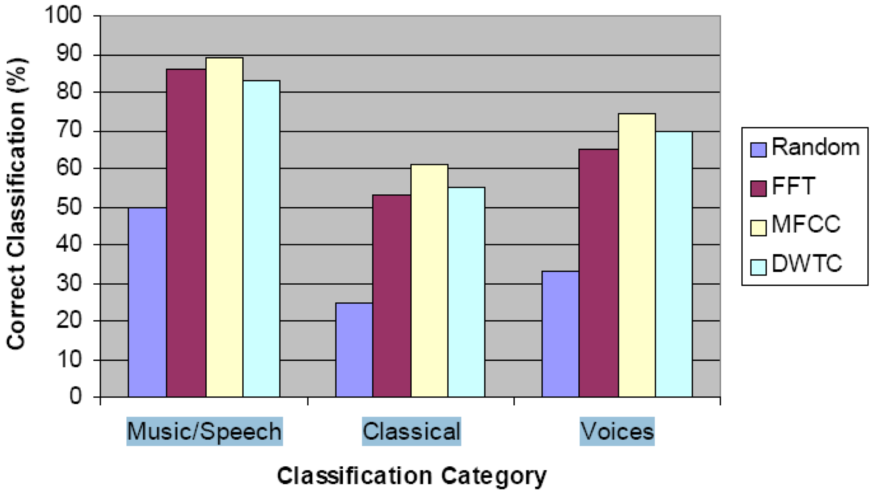
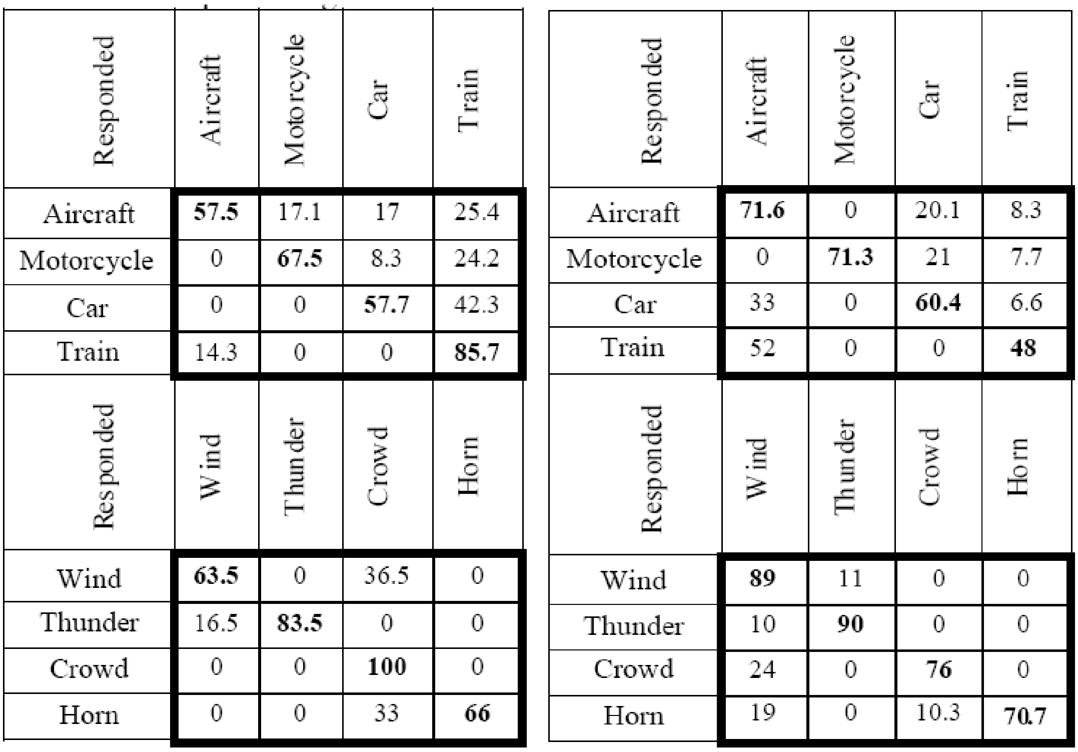


Figure 3.4: Extracted from [17], the accuracy results from wavelet features (DWTC) against MFCC and short-time features (FFT).

One alternative to the features described above is using the MPEG-7 descriptors [16]. MPEG-7 is a standard that describes the multimedia content retrieved. It has a set of features which can be used instead of the MFCCs. In 2008, it was developed an environmental sound recognizer which used some Low Level Descriptors (LLDs) from MPEG-7:

- Audio Waveform – description of the shape of an audio signal
- Audio Power – temporal descriptor of the evolution of the sampled data
- Audio Spectrum Envelope – series of features that describe the basic spectra
- Audio Spectrum Centroid – center of the log-frequency spectrum's gravity
- Audio Spectrum Spread – measure of signal's spectral shape
- Audio Spectrum Flatness – measure of how flat a particular portion of the signal is
- Harmonic Ratio – proportion of harmonic components in the power spectrum
- Upper Limit of Harmonicity – measure of the frequency value beyond which the spectrum no longer has any harmonic structure
- Audio Fundamental Frequency – estimation of the fundamental frequency

Table 3.2 allows us to compare the results from MFCCs with the results from MPEG-7 descriptors. We can see that using the MPEG-7 allowed higher recognition rates but only for some classes. The rates for crowd sounds and train sounds are higher using MFCCs. However, overall MPEG-7 descriptors retrieve great results and are powerful for environmental sound recognition consequently.



Responded	Aircraft	Motorcycle	Car	Train
Aircraft	**57.5**	17.1	17	25.4
Motorcycle	0	**67.5**	8.3	24.2
Car	0	0	**57.7**	42.3
Train	14.3	0	0	**85.7**

Responded	Wind	Thunder	Crowd	Horn
Wind	**63.5**	0	36.5	0
Thunder	16.5	**83.5**	0	0
Crowd	0	0	**100**	0
Horn	0	0	33	**66**

Responded	Aircraft	Motorcycle	Car	Train
Aircraft	**71.6**	0	20.1	8.3
Motorcycle	0	**71.3**	21	7.7
Car	33	0	**60.4**	6.6
Train	52	0	0	**48**

Responded	Wind	Thunder	Crowd	Horn
Wind	**89**	11	0	0
Thunder	10	**90**	0	0
Crowd	24	0	**76**	0
Horn	19	0	10.3	**70.7**

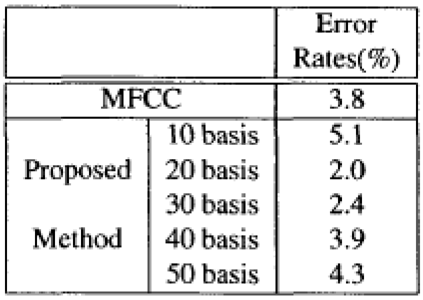
Table 3.2: Extracted from [18], the accuracy obtained with MFCCs (on the left) and with MPEG-7 descriptors (on the right).

To provide better recognition rates some improvements over MFCCs [19] have been done, too. ICA can be applied over these features. The result features showed ICA transformation is useful in sound recognition [20]. One good example is a sound recognizer based on kitchen events (water boiling, vegetable cutting, microwave beep, etc.) [21]. ICA was tested and also improved the accuracy results (Table 3.3) from 80.6% to 85% as we can see in table 3.6.

System	Error	Precision
BASE	12.4%	80.6%
ICA	9.2%	85.0%

Table 3.3: Extracted from [21], the results with and without ICA transformation over the set of MFCCs.

Both ICA and PCA are not fully explored in environmental sound recognition. Instead of being applied only to the set of MFCCs, these techniques could be applied to the whole spectrogram retrieving new features that can be used. In speech recognition there are examples of how ICA and PCA can provide better results. In [22], the sound features were extracted applying ICA to spectrograms which were computed to represent the speech samples. The number of features extracted influences the results. However, we can see in Table 3.4 how features extracted with ICA could out-perform MFCCs: using only 20 or 30 basis functions extracted with ICA as features provides a lower error rate than using MFCCs.



		Error Rates(%)
MFCC		3.8
Proposed Method	10 basis	5.1
	20 basis	2.0
	30 basis	2.4
	40 basis	3.9
	50 basis	4.3

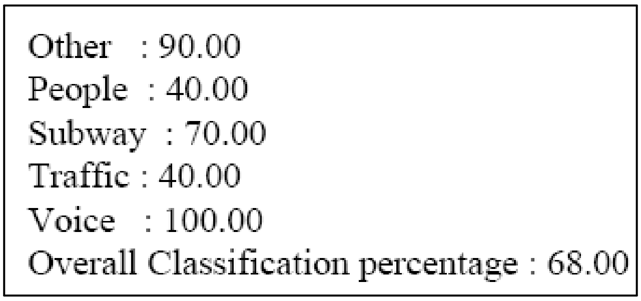
Table 3.4: Extracted from [22], the error rates of feature extraction with ICA against MFCCs.

PCA was tested in distorted speech recognition [23]. PCA is applied to the spectrograms retrieving the features used in the classification method. It improved the recognition rates in more than 10% showing how these techniques can be useful in sound recognition.

Sound Recognition and Classification

Above, we have seen several approaches in sound feature extraction. After extracting these features, it is necessary to implement a classification algorithm that uses the features to match the tested sounds to the right category. This section discusses some sound recognizers focusing on the techniques they use for classification.

A technique that is used very commonly in sound classification and that gives very good results is the k-Nearest Neighbor (k-NN). As an illustration of a recognizer that uses k-NN, here we discuss Nitin Sawhney's environmental sound recognizer, which distinguishes pre-defined classes such as people, voices, subway and traffic [24]. To implement the k-NN, each tested class is represented by a vector in multi-dimensional feature space. Then each tested sample is added to this space and it is assigned the most represented class in his *k* nearest neighbors. Sawhney concluded that the k-Nearest Neighbor classification technique, combined with Auditory FilterBank envelopes, gives good results on environmental sound recognition. Table 3.5 shows the results this recognizer obtained showing the recognition rate of each of the tested classes. While some classes show great recognition rate (for instance the recognition rate for voice is 100%), some other classes (namely people and traffic) retrieved poor recognition results: 40%. The overall accuracy is 68%.



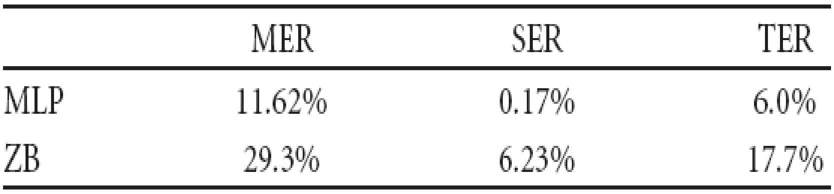
Other	: 90.00
People	: 40.00
Subway	: 70.00
Traffic	: 40.00
Voice	: 100.00
Overall Classification percentage	: 68.00

Table 3.5: Table extracted from [24]. The accuracy rates of the recognizer using FilterBank and k-NN.

A recurrent neural network (RNN) was also used to classify these sounds but with very poor results, which shows that the k-NN is more appropriate for these classes. The implemented RNN uses RASTA (Relative Spectral Transform) coefficients which are useful in speech recognition and are short-time features. The maximum recognition rate returned was 73.5% using the data used to train the RNN. Using other data that would fit in the classes used, the recognition rate was 24% which is a very poor result.

Neural networks can also provide good results in sound classification. An example of a neural network that can be used for sounds recognition is a Multi-Layer Perceptron network (MPN). Bugatti *et al.* used a MPN to classify sounds from five different contexts: instrumental music without voice, melodic songs, rhythmic songs, pure speech and speech superimposed on music [25]. The results from this classifier were compared against those of a (Naïve) Bayesian Classifier which uses only the ZCR feature.

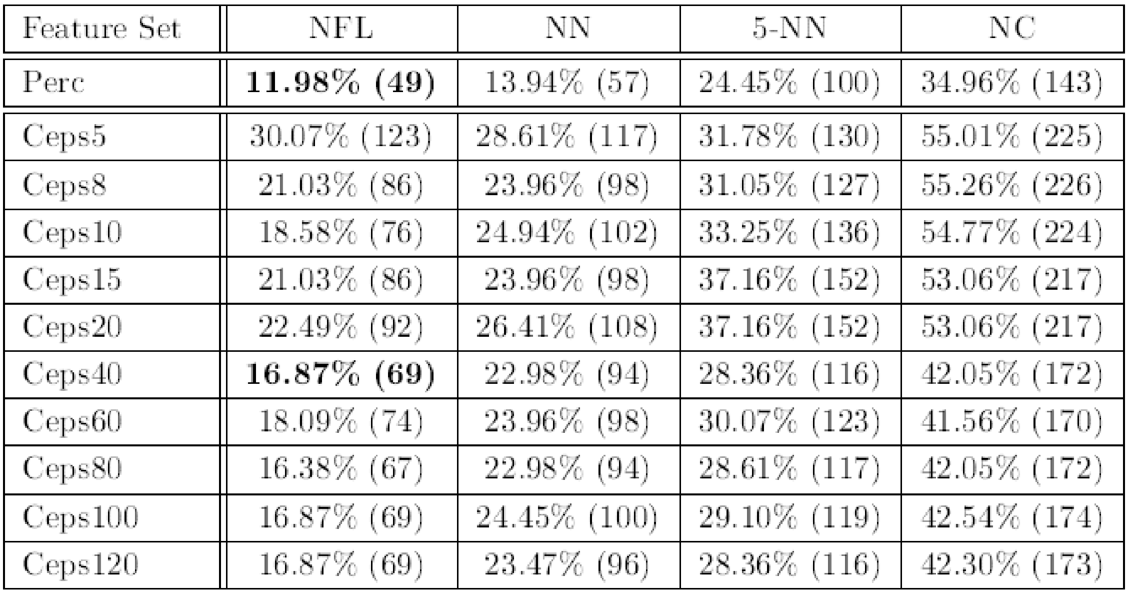
Table 3.6 shows the music error rate, the speech error rate and the total error rate of both methods. We can see MPN provides much lower error rates. Although the Bayesian Classifier is a simple algorithm, the difference in the music error rate is considerable, which shows that the MPN is more appropriate to classify these sounds.



	MER	SER	TER
MLP	11.62%	0.17%	6.0%
ZB	29.3%	6.23%	17.7%

Table 3.6: Extracted from [25], Error Rates returned by the MPN network (first row) and by ZCR with the Bayesian Classifier (second row).

Neural Networks are not the only alternative to k-NN. Other alternative proposed is the Nearest Feature Line (NFL) and the Nearest Center methods. These methods were compared to 1-NN and 5-NN [26]. To do this test, 16 different classes of sounds were used. Table 3.7 shows NFL usually provides higher recognition rates than k-NN. Nearest Center algorithm returns the worse results so we can assume it is not useful for sound recognition.



Feature Set	NFL	NN	5-NN	NC
Perc	**11.98% (49)**	13.94% (57)	24.45% (100)	34.96% (143)
Ceps5	30.07% (123)	28.61% (117)	31.78% (130)	55.01% (225)
Ceps8	21.03% (86)	23.96% (98)	31.05% (127)	55.26% (226)
Ceps10	18.58% (76)	24.94% (102)	33.25% (136)	54.77% (224)
Ceps15	21.03% (86)	23.96% (98)	37.16% (152)	53.06% (217)
Ceps20	22.49% (92)	26.41% (108)	37.16% (152)	53.06% (217)
Ceps40	**16.87% (69)**	22.98% (94)	28.36% (116)	42.05% (172)
Ceps60	18.09% (74)	23.96% (98)	30.07% (123)	41.56% (170)
Ceps80	16.38% (67)	22.98% (94)	28.61% (117)	42.05% (172)
Ceps100	16.87% (69)	24.45% (100)	29.10% (119)	42.54% (174)
Ceps120	16.87% (69)	23.47% (96)	28.36% (116)	42.30% (173)

Table 3.7: Extracted from [26]: the error rates retrieved by the each classification method using different sets of features: a set of short-time features (Perc) and a set of *k* MFCC's (Ceps*K*).

Support Vectors Machine (SVM) is another alternative to the k-NN algorithm. This was tested using five classes of sounds (silence, music, background sound, pure speech, non-pure speech) [27]. However, we can only apply SVM if we use only two classes each time. In order to do this, the silence class was separated from the others 4 classes (non-silence). Figure 3.5 shows how these other four classes were divided.

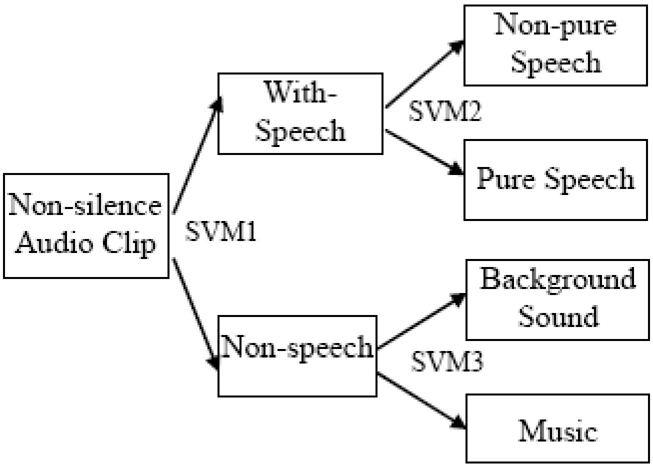
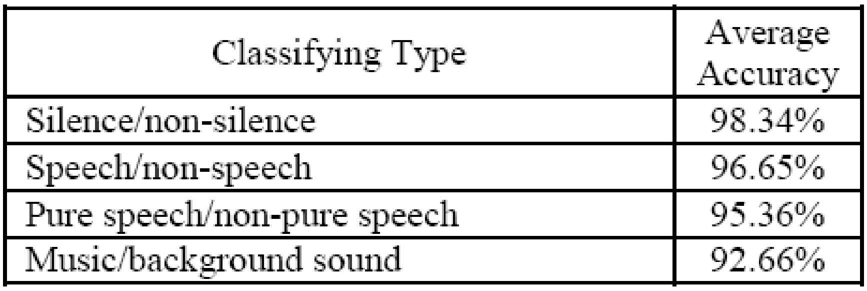


Figure 3.5: Extracted From [27], sound classes division for multi-class classification using SVM.

Therefore, SVM is not a very practical method for environmental sound recognition, especially if using several sound classes. However, it can provide impressive recognition rates as shows Table 3.8: all the accuracy rates are higher than 90%.



Classifying Type	Average Accuracy
Silence/non-silence	98.34%
Speech/non-speech	96.65%
Pure speech/non-pure speech	95.36%
Music/background sound	92.66%

Table 3.8: Extracted from [27], sound classes division for multi-class classification using SVM.

Other alternative to these techniques consists of using Gaussian Mixture Models (GMM). GMM consists in modeling a new cluster for each of the studied classes using its information. Assuming a cluster is denoted by y_k, each one of them will have a related function $p(x|y_k)$. Assuming N is the number of clusters generated, the classification algorithm assigns the cluster y_k to the tested object x if

$$y_k = \arg\max_{1 \leq i \leq N} p(x|\, y_i)$$

GMM was tested in a recognizer proposed by Eronen *et al* [15]. To study the results GMM was tested with MFCCs and with a set of short-time features. The same test was done using 1-NN instead of the GMM.

Comparing the different features used and the classification methods, GMM classifier provides better recognition accuracy (63.4%) when using MFCCs. Considering MFCCs were proven to be very useful in sound recognition, GMM consists in a good technique for sound classification (figure 3.6).

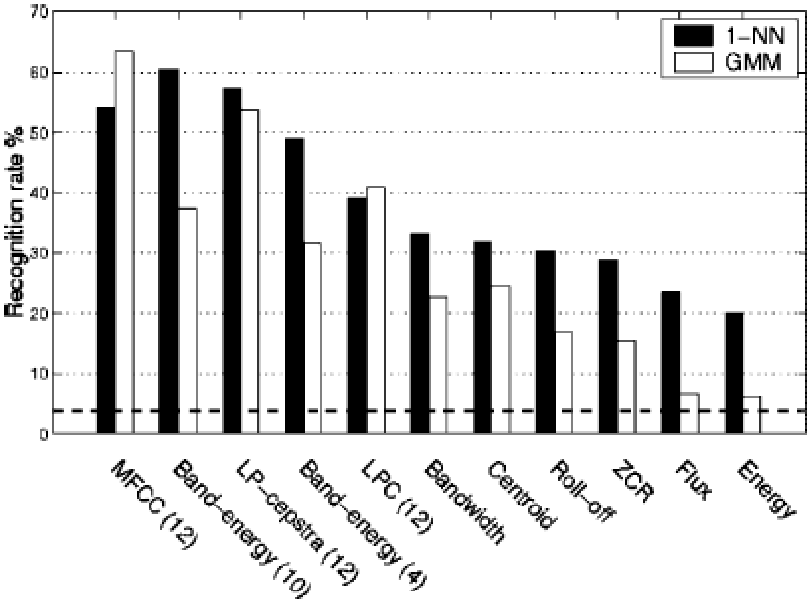


Figure 3.6: Extracted from [15], features' recognition rate with two different classifiers GMM and 1-NN.

Figure 3.6 shows that the 1-NN classifier was less impressive using MFCCs. However, using the Band-energy ratio features also grants good results (61.5%). This last example was once again tested for more general classes returning 68.4% accuracy. Using more general classes should improve the accuracy of the recognizer, though, because the temporal properties of the sound are more different making the classification process easier.

Instead of a k-NN or a GMM, a hidden Markov Model (HMM) can also be used as the classification framework. An HMM consists of a set of states and the probabilities of changing from each state to each of the other states. The probability of observing a sequence of states

$$Y = S_a, S_b, S_c, S_d$$

is given by

$$P(Y) = \sum_{i=0}^{N} P(Y|S_i)P(S_i)$$

where N is the number of states.

Now it is necessary to adapt the HMM to the sound classification problem. Each class v is modeled into a HMM λ^v using the features extracted. For matching the tested sample with one of the classes, the recognizer identifies the sequence of states

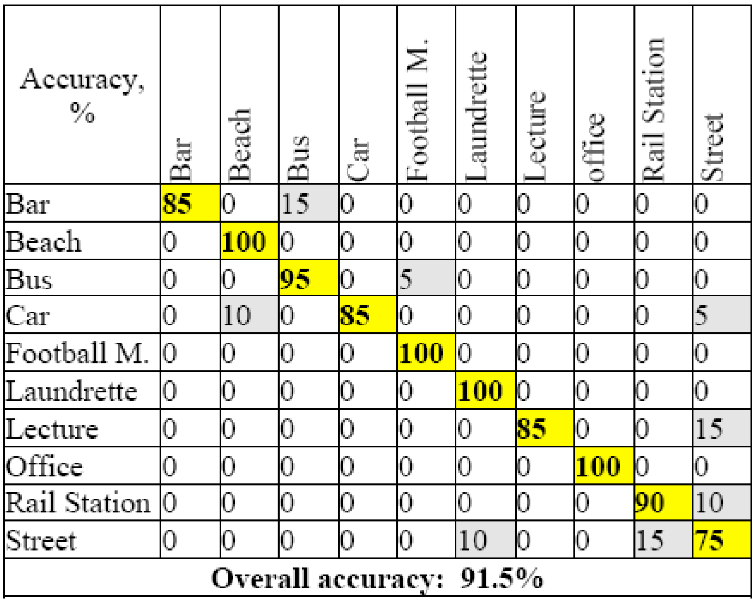
$$O = S_x \dots S_y$$

using its sound features and assigns it to the class v if

$$v = \arg \max_{1 \leq v \leq V} p(O|\,\lambda^v)$$

One sound recognizer which uses HMMs was proposed by L. Ma *et al.* [28]. This recognizer uses MFCCs as features for the classification because they were demonstrated to achieve better performances with GMM.

The overall accuracy obtained with ten different environments was 91.5%. The same test was done with persons resulting in a 35% accuracy which can be justified by the short samples used in the test. However, the number is much smaller than the obtained by the HMM model. As we can see, combining MFCCs and HMM can provide great results (figure 3.7): in four different contexts the recognition rate was 100%. However, note that street context has a lower recognition rate of 75%. This happens because these sounds are similar to the Rail Station, the lecture and the car context sounds and similar sounds make the recognizer task harder.



Accuracy, %	Bar	Beach	Bus	Car	Football M.	Laundrette	Lecture	office	Rail Station	Street
Bar	85	0	15	0	0	0	0	0	0	0
Beach	0	100	0	0	0	0	0	0	0	0
Bus	0	0	95	0	5	0	0	0	0	0
Car	0	10	0	85	0	0	0	0	0	5
Football M.	0	0	0	0	100	0	0	0	0	0
Laundrette	0	0	0	0	0	100	0	0	0	0
Lecture	0	0	0	0	0	0	85	0	0	15
Office	0	0	0	0	0	0	0	100	0	0
Rail Station	0	0	0	0	0	0	0	0	90	10
Street	0	0	0	0	0	10	0	0	15	75
Overall accuracy: 91.5%										

Figure 3.7: Extracted from [28], recognition rates of 10 contexts using HMM.

HMM was, again, compared directly to GMM using MFCCs [29]. The test showed that HMM guaranteed 10% higher accuracy than GMM. To compare these results with the human ability, the same tests were done with persons. The average accuracy obtained is slightly higher than HMM which showed there is still progress to be done (figure 3.8).

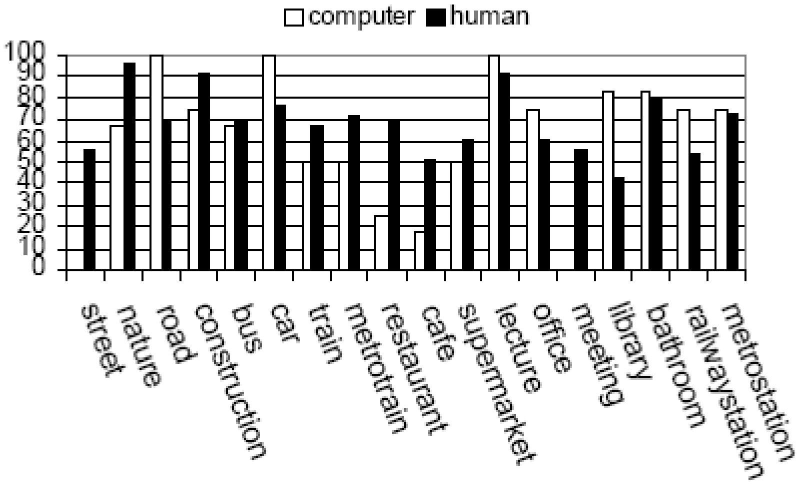


Figure 3.8: Extracted from [29], the comparison of recognition rates for 18 contexts.

In this same work, discriminative training was tested. Discriminative training adjusts the parameters of the HMMs assigning rules to reduce the error rate in recognition. However, it did not improve the accuracy obtained. On the other hand, discriminative training for the classification process was also tested by Eronen in 2003 [18] showing minimal improvements.

We can conclude the classification framework used will always depend on the features used and their behavior. In this dissertation we will focus more on the feature extraction process than in the classification method. As we use signals (see next section) that can be seen as short-time features we will use the k-NN algorithm which was proven to retrieve good results (figure 3.6).

We can also conclude MFCCs are the features most popular in sound recognition since they provide high recognition rates. We will test the results of our recognizer with the features extracted by the ISA method against its results with MFCCs. We will show our features work better with such similar sounds as the ones we use. In the next section, we describe our solution extensively.

4. Proposed Solution

Sound recognizers have to perform three steps to achieve their goal (figure 4.1). First, the sound has to be digitized and processed; it is transformed into a representation that is suitable for the extraction of features from its new digital representation. The next step consists of extracting the chosen audio features. Finally, these features will be used in the last step where the sound is classified (assigned to a class).

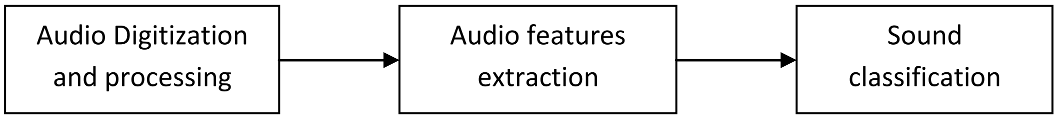


Figure 4.1: The three steps of an environmental sound recognizer.

The data used to test the proposed similar sound recognizer is a set of impacts on rods [6], which includes samples from four rods with the same length and diameter but different materials (wood, aluminum, steel and zinc plated steel). The sounds were all produced by impacts on the same region of the rod (close to the edge). Even though the sounds differ from each other due to variations on the impact force and on the location of impact, they are quite similar as they are produced by the same type of event and by objects with the same shape and size which only differ in material. The first step of our classifier consists of digitizing the sounds, which was already accomplished in [6] with a sampling frequency of 44100 Hz, and representing them with spectrograms so that they are ready to be used by the next step of the recognizer.

The next step (second stage of figure 4.1) consists of extracting the sound features. Instead of extracting pre-defined features of the sounds such as MFCCs or short-time features, our recognizer learns them from the data. In order to learn the features, the recognizer uses the Intrinsic Structures Analysis (ISA) method, which in turn uses Independent Component Analysis (ICA) and Principal Component Analysis (PCA) of the spectrogram of the sounds [6]. Here we prove that the features learned in this manner are powerful enough for sound classification of very similar sounds. These techniques allow us to extract time and frequency-varying functions that we use as features for classification in the third stage of the classifier (the third box in figure 4.1).

The final step is to classify the sound. We use the features extracted with the ISA method in a 1-Nearest Neighbor Algorithm as we mentioned in the end of section 3 and explain it with detail below. We can see these three steps in figure 4.2.

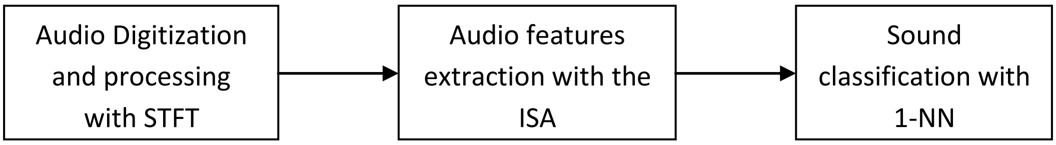

Audio Digitization and processing with STFT
Audio features extraction with the ISA
Sound classification with 1-NN

Figure 4.2: The three steps of the proposed sound recognizer.

The remaining of this section is divided in two parts. First, we describe both ICA and PCA techniques and the features we will extract after representing the sounds with spectrograms. Then, we describe the classification method.

1. Features extraction with the ISA method

The ISA method uses spectrograms as the initial representation of the sounds and extracts features from the spectrograms. It proposes two different ways of using the spectrogram (either the original spectrogram or the transpose) along with two distinct techniques to analyze them (ICA and PCA). We will use all these combinations separately and then compare the results obtained by each technique.

ICA [30, 31] is a technique to separate mixed source signals from signal mixtures. Given a matrix S containing the signal mixtures, ICA's goal is to find the matrix *W* so that the rows of *X*, the source signals, are the most independent possible:

$$X = WS \Leftrightarrow$$

$$S = W^{-1}X = AX$$

As we have seen, the STFT of a sound *S* is a matrix of amplitudes a_{ft}. By assuming that *S* consists of a set of signal mixtures, ICA is able to separate the source signals by learning a matrix *A* that verifies

$$S = AX \Leftrightarrow$$

$$\begin{bmatrix} a_{00} & \cdots & a_{0T} \\ \vdots & & \vdots \\ a_{F0} & \cdots & a_{FT} \end{bmatrix} = \begin{bmatrix} \alpha_0 & \cdots & \gamma_0 \\ \vdots & & \vdots \\ \alpha_F & \cdots & \gamma_F \end{bmatrix} \begin{bmatrix} x_{00} & \cdots & x_{0T} \\ \vdots & & \vdots \\ x_{N0} & \cdots & x_{NT} \end{bmatrix}$$

where $[a_{i0} \ldots a_{iT}]$ is a signal mixture, $[x_{i0} \ldots x_{iT}]$ is a source signal and $N \leq F$.

This separation is done by representing each column of A as an orientation vector (i.e. basis function) of the correspondent line of X so that

$$[a_{i0} \ldots a_{iT}] = \mathrm{A_i}^{\mathrm{T}} X$$ [*]

$$[a_{i0} \ldots a_{iT}] = [\alpha_i \ldots \gamma_i] \begin{bmatrix} x_{00} & \cdots & x_{0T} \\ \vdots & & \vdots \\ x_{N0} & \cdots & x_{NT} \end{bmatrix} \Leftrightarrow$$

$$[a_{i0} \ldots a_{iT}] = \alpha_i [x_{00} \ldots x_{0T}] + \ldots + \gamma_i [x_{N0} \ldots x_{NT}]$$

where a_{it} is the inner product of the vector $\mathrm{A_i}^{\mathrm{T}} = [\alpha_i \ldots \gamma_i]$ and the i^{th} column in X: $[\mathrm{x}_{0\mathrm{i}} \ldots \mathrm{x}_{\mathrm{Ni}}]^{\mathrm{T}}$.

Just like ICA, PCA [30, 32] also decomposes S into matrixes A and X. The main difference is that PCA's goal is not to find a matrix W such that the rows of X are independent but uncorrelated.

The rows of X, the source signals, are the values of the features we will use for classification. They can be seen as time-varying functions (figure 4.3, left column) each one with an associated orientation vector. This vector can be seen as a frequency-varying function (figure 4.3, right column).

[*] $\mathrm{A_i}^{\mathrm{T}}$ is the i^{th} line of matrix A

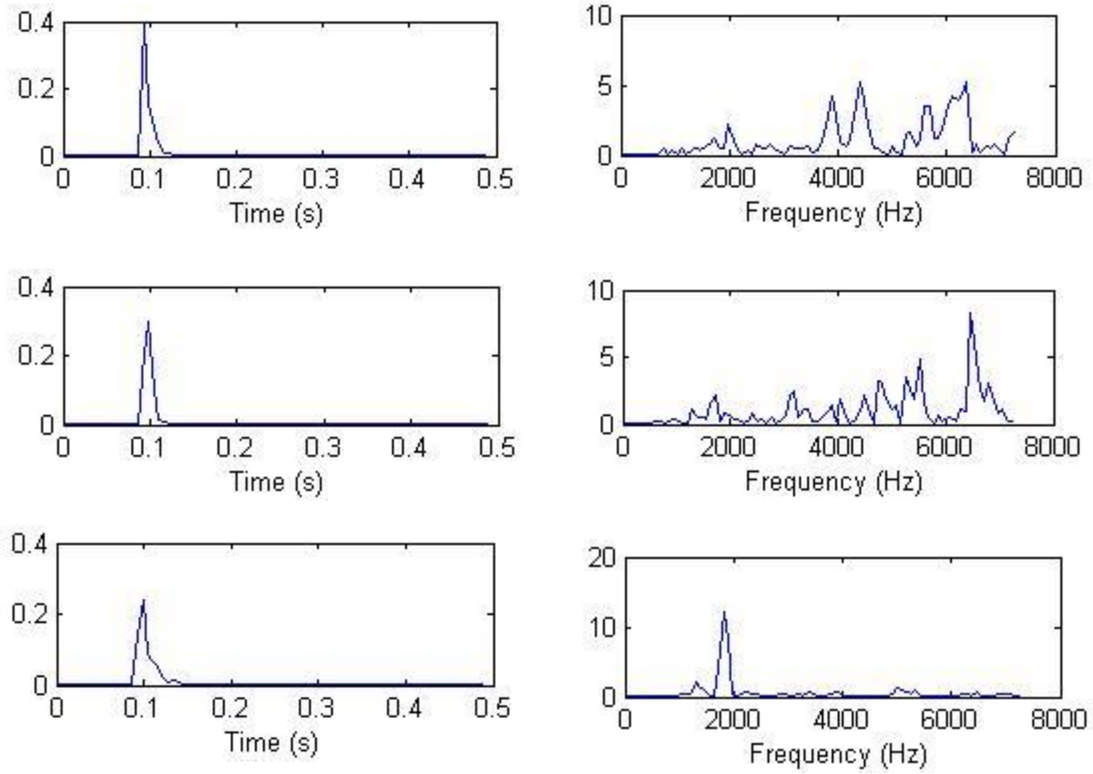


Figure 4.3: Temporal features extracted with ICA from a wood sound (left column: source signals [x_{i0} ... x_{iT}]; right column: the associated orientation vector [δ_0 ... δ_F]^T).

As ICA and PCA are matrix operations we can perform them over the transpose of the sound's spectrogram

$$S^T = AX \Leftrightarrow$$

$$\begin{bmatrix} a_{00} & \dots & a_{0F} \\ \vdots & & \vdots \\ a_{T0} & \cdots & a_{TF} \end{bmatrix} = \begin{bmatrix} \alpha_0 & \dots & \gamma_0 \\ \vdots & & \vdots \\ \alpha_T & \cdots & \gamma_T \end{bmatrix} \begin{bmatrix} x_{00} & \dots & x_{0F} \\ \vdots & & \vdots \\ x_{N0} & \cdots & x_{NF} \end{bmatrix}$$

where [a_{i0} ... a_{iF}] is a signal mixture and [x_{i0} ... x_{iF}] is a source signal and $N \leq T$. The rows of X are now frequency-varying functions (figure 4.4, left column) with an orientation vector associated that can be seen as a time-varying function (figure 4.4, right column).

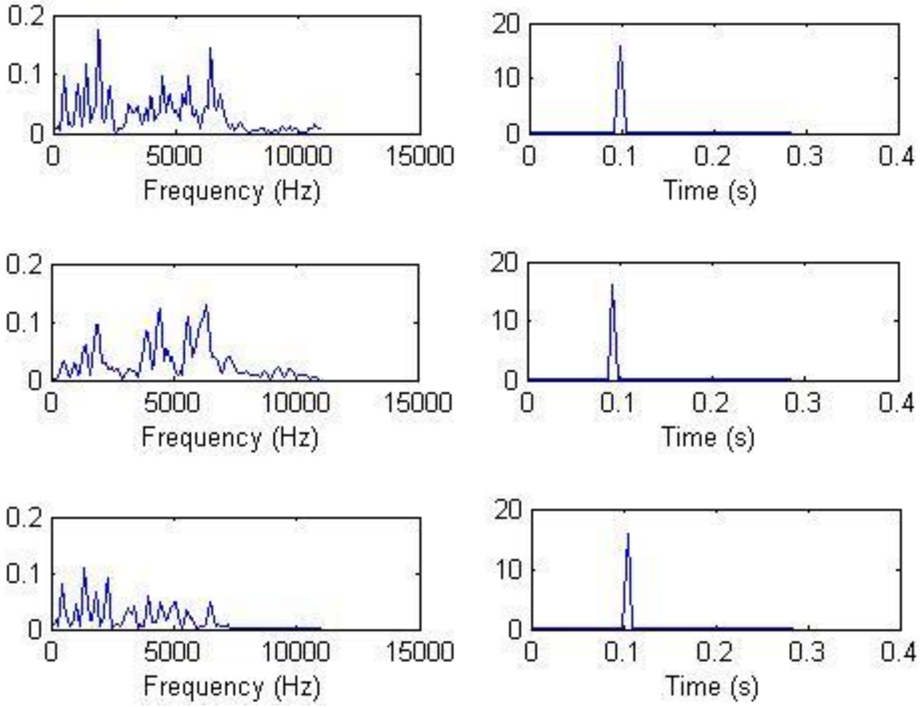


Figure 4.4: Spectral features extracted with ICA from a wood sound (left column: source signals $[x_{i0} \ldots x_F]$; right column: associated orientation vector $[\delta_0 \ldots \delta_T]^T$).

These are the two different types of analysis we perform. When we use the rows of X as time-varying functions for features in our recognizer, we call this *temporal analysis*. When we use the rows of X as frequency-varying functions, we call this *spectral analysis*.

1. Sound classification with 1-NN

The third stage of our classifier (figure 4.2) uses a 1-Nearest Neighbor algorithm. In order to implement this algorithm, we need to build a training data set. Once that is done, the algorithm is ready to classify test samples: it matches the test sample to one record of the training data set based on the Euclidean distance. It just sees which training data sample is the closest to the tested sample. The tested sample is assigned the class that occurs more in the k neighbors (1 in our case).

The classification method of our recognizer is described by figure 4.5. Our training data is composed of the coefficients associated to M orientation vectors of N sounds ($N/4$ sounds for each of the four classes we have). Each record in the training data set consists of those M coefficients from one sound (figure 4.5). The tested sample is composed by the M coefficients associated to the same orientation vectors.

Then, using 1-NN, we compare the value of the M coefficients in one instant of time or frequency (depending on the analysis) of the training sample

$$(C_1^{Training\ \ Sample}, C_2^{Training\ \ Sample}, \dots, C_{M-1}^{Training\ \ Sample}, C_M^{Training\ \ Sample})$$

with the M coefficient values of the test sample

$$(C_1^{Test}, C_2^{Test}, \dots, C_{M-1}^{Test}, C_M^{Test})$$

The 1-NN algorithm computes the Euclidean distances between the test sample and all training samples and returns the index of the training sample which is closest to the test sample:

$$i = \arg\min_{1 < i < N} \sqrt{\sum_{x=1}^{X} (C_x^{Test} - C_x^{Training\ \ Sample\ \ i})^2}$$

with N being the number of sounds in the training data and M the number of used features. We repeat this process for various instants of time or frequency (depending on the analysis). The final class assigned to the tested sound is the one which occurs more in the instants selected.

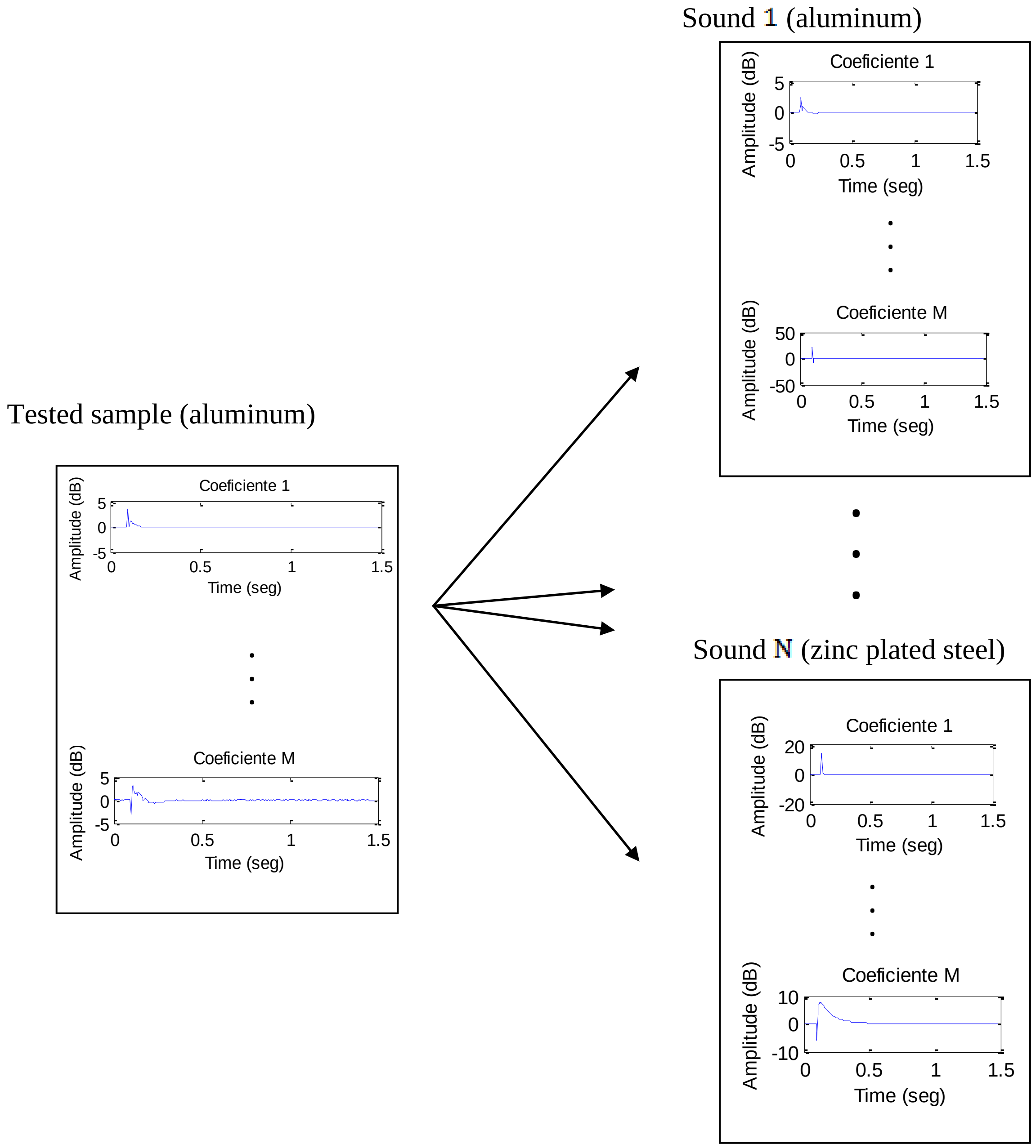


Figure 4.5: Classification scheme of our recognizer.

5. Implementation

As discussed before, the implemented recognizer is divided in three phases. The first phase consists of transforming the data from a time representation (that is, from waveforms) into a spectrotemporal representation (which is suitable to be used in the next phase of the recognizer). This consists of taking the spectrograms of the sounds and concatenating them into a unique matrix. Then, the second phase consists of learning features that will later be used in a recognition algorithm. The features are learned by either ICA or PCA of the concatenated spectrograms. Finally, we use the features extracted by ICA or PCA for classification with the 1-NN algorithm.

In order to train our recognizer we have to focus in two tasks. First, we have to build the training data. Then, we have to define how the classifier receives a sound and assigns it one of the classes of the training data.

- **Building the training data**

As mentioned above, the sounds are initially represented by spectrograms, which are obtained by the STFT function with a 512-Hanning window, 512-point FFT and 256 overlap. Figure 5.1 shows an example of a spectrogram from one of our sounds. This way, each sound is described by a matrix of frequency by time, i.e. an array of vectors $f_1, f_2, \dots, f_t$, with t being the number of spectrogram frames (that can be thought of as time instants) and f_x the vector with the frequency values in instant x. The spectrograms are then concatenated into a bigger matrix. The way this concatenation is made depends on whether we use temporal or spectral analysis (see section 4).

Let us first look into temporal analysis. In this case, the spectrograms, which have size $(F \times T)$, are concatenated horizontally to obtain a bigger matrix of size $(F \times NT)$ where N is the number of sounds, just as if the sounds were reproduced in sequence, one after the other. This bigger matrix is then used as input to ICA or PCA and we can see an example in figure 5.2.

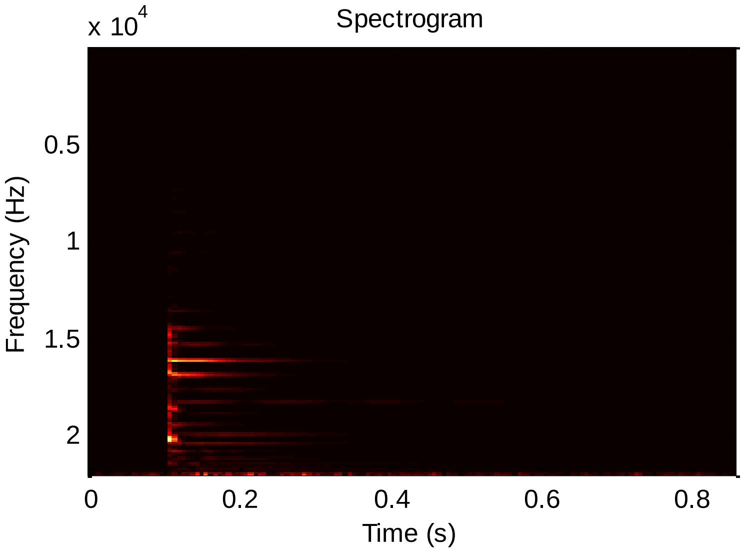


Figure 5.1: The spectrogram of one of the sounds made by a steel rod.

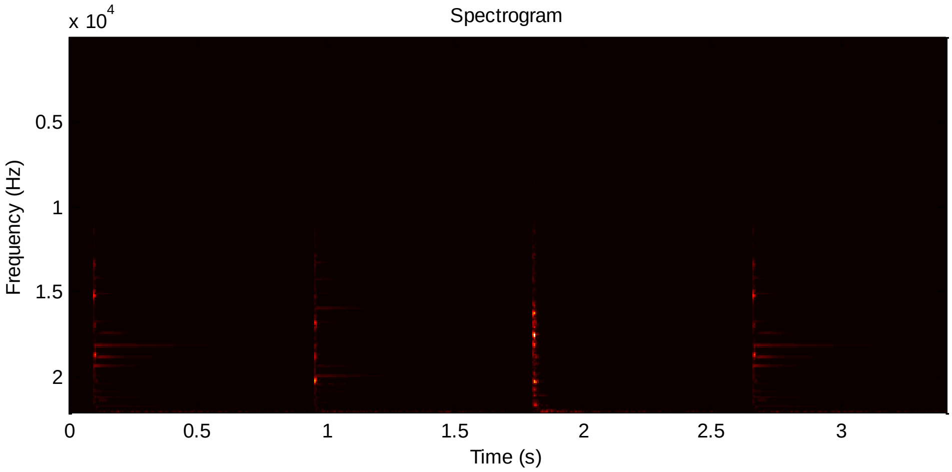


Figure 5.2: Example of a matrix of spectrograms for temporal analysis: (S_1, S_2, S_3, S_4) *. This matrix contains four sounds, each one made by a different rod.

Now, let us look into spectral analysis. The spectrograms of size $(F \times T)$ are now concatenated vertically in order to compose a bigger matrix of size $(NF \times T)$. The transpose of the obtained spectrogram, with size $(T \times NF)$, is the matrix we use as input for ICA or PCA (figure 5.3).

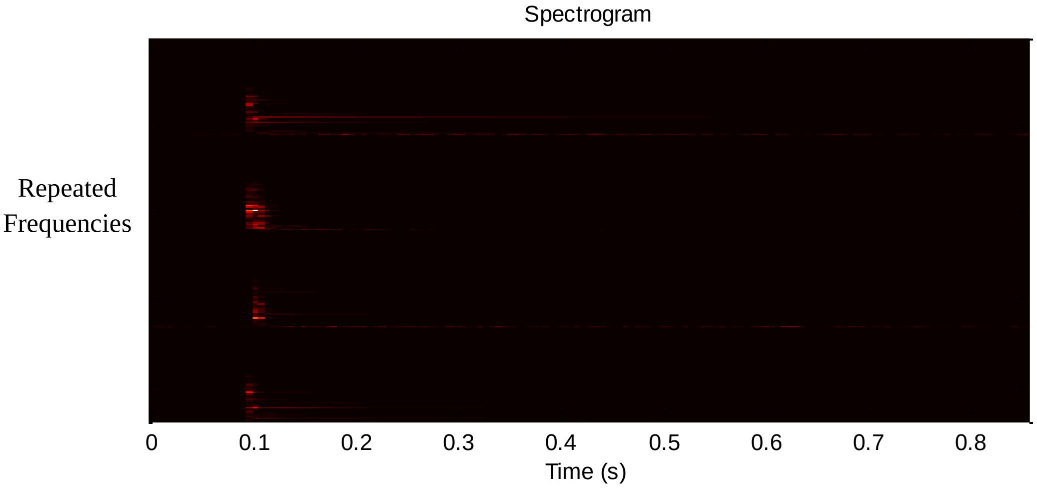


Fig 5.3: Example of a transposed matrix of transposed spectrograms for spectral analysis: $({S_1}^T, {S_2}^T, {S_3}^T, {S_4}^T)$. This matrix contains four sounds, each one made by a different rod.

* (A,B) is the horizontal concatenation of matrixes A and B.

The next step consists of applying ICA or PCA over the concatenated spectrograms described above. As explained above, ICA and PCA transform a matrix of mixed sources S in a matrix of (estimated) source signals X finding a transformation matrix A where each column is the orientation vector of each row of the matrix X:

$$S = AX$$

In our implementation the matrix S is the matrix of concatenated spectrograms. ICA learns matrix A and along with it, it returns matrix X. In particular, we use the fastICA algorithm, which is the Matlab ICA implementation by Hyvärinen *et al.* [33]. Each row of the matrix X is the value of one of the extracted features, strictly speaking it has the concatenation of N values of one of the extracted features, where N is the number of sounds.

Figure 5.4 shows the values of one of the features obtained by temporal analysis with ICA. These values are arrays of amplitude varying over time until time instant NT. Figure 5.5 shows the values of one of the features retrieved with spectral analysis with ICA. These values can be seen as arrays of frequency-varying amplitude.

In both figures, the behavior of the selected feature for five aluminum sounds is represented in red, in blue for five zinc plated steel sounds, in green for five wood sounds and in black for five steel sounds. By looking into the values of the features in each figure, we can detect differences in the sounds according to their classes. In figure 5.4, the steel sounds behave notoriously different from the other sounds: the amplitude on the beginning of first two sounds is much higher than for any other. On the other hand, in figure 5.5, it is the aluminum sounds that behave differently: we can see the aluminum sounds with much more amplitude than the others.

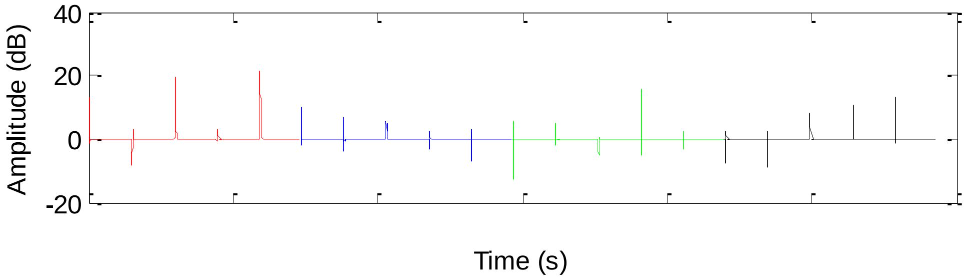


Fig 5.4: Values of an ICA feature obtained with temporal analysis over a spectrogram with 20 sounds (five from each class: aluminum in red, zinc plated steel in blue, wood in green and steel in black).

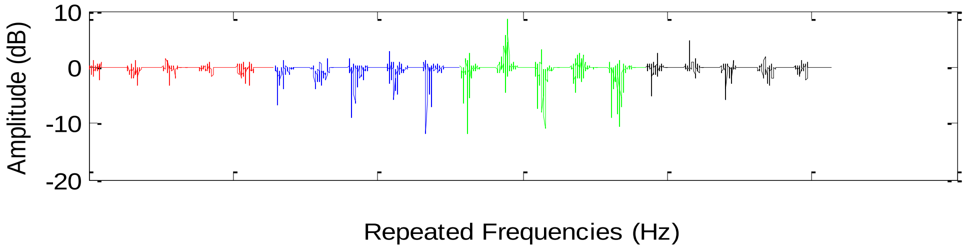


Fig 5.5: Values of an ICA feature obtained with spectral analysis over a spectrogram with 20 sounds (five from each class: aluminum in red, zinc plated steel in blue, wood in green and steel in black).

We use PCA like we use ICA because PCA also learns matrix A to return matrix X. To apply PCA we use *princomp* Matlab function. Like in ICA, each row of matrix X contains N values of one feature being N the number of sounds. One of these rows extracted with PCA for temporal analysis can be seen in figure 5.6 and one extracted with PCA for spectral analysis can be seen in figure 5.7.

Like with the ICA features, PCA features also allow us to distinguish some classes of the sounds immediately. In both figure 5.6 and 5.7, the aluminum sounds and the steel sounds have low amplitude values compared to the zinc platted steel sounds and the wood sounds.

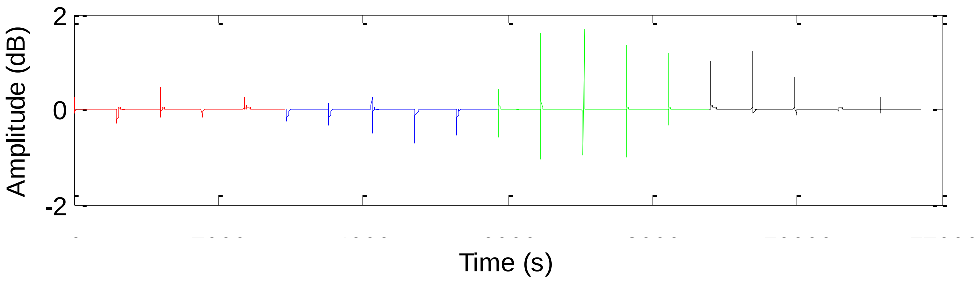


Fig 5.6: Values of a PCA feature obtained with temporal analysis over a spectrogram with 20 sounds (five from each class: aluminum in red, zinc plated steel in blue, wood in green and steel in black).

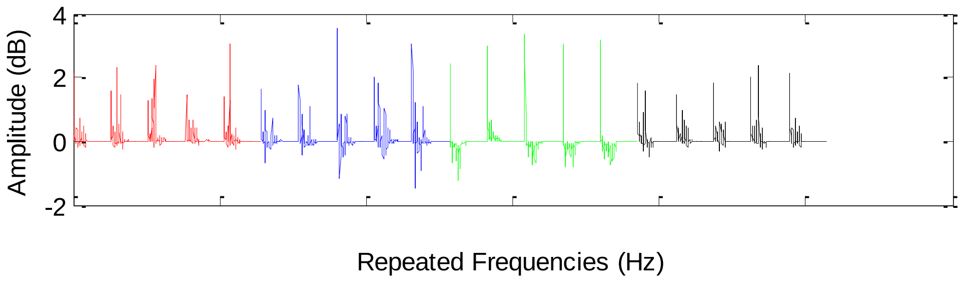


Fig 5.7: Values of a PCA feature obtained with spectral analysis over a spectrogram with 20 sounds (five from each class: aluminum in red, zinc plated steel in blue, wood in green and steel in black).

Then we have to build our training data using the matrix X which is returned by PCA or ICA. As we have seen, each record of our training data set consists of the selected coefficients of one of the N sounds. To select the coefficients we sort the orientation vectors according to the percentage of variance they account for. We select some of the most dominant orientation vectors and use the corresponding coefficients to build the training data set. Sorting the orientation vectors is an easy task for PCA but hard for ICA as the orientation vectors are perpendicular with PCA but not with ICA (that is, with PCA the inner product is zero while it may be different from zero with ICA), which means that it is hard to know how much of the variance of the data one orientation vector from ICA accounts for.

To build the training data set we will group the values for the M features of each sound as is illustrated in figure 5.8: we will build N matrixes with size $M \times T$ for temporal analysis or N matrixes with size $M \times F$ for spectral analysis. Each matrix built is one record of the training data set.

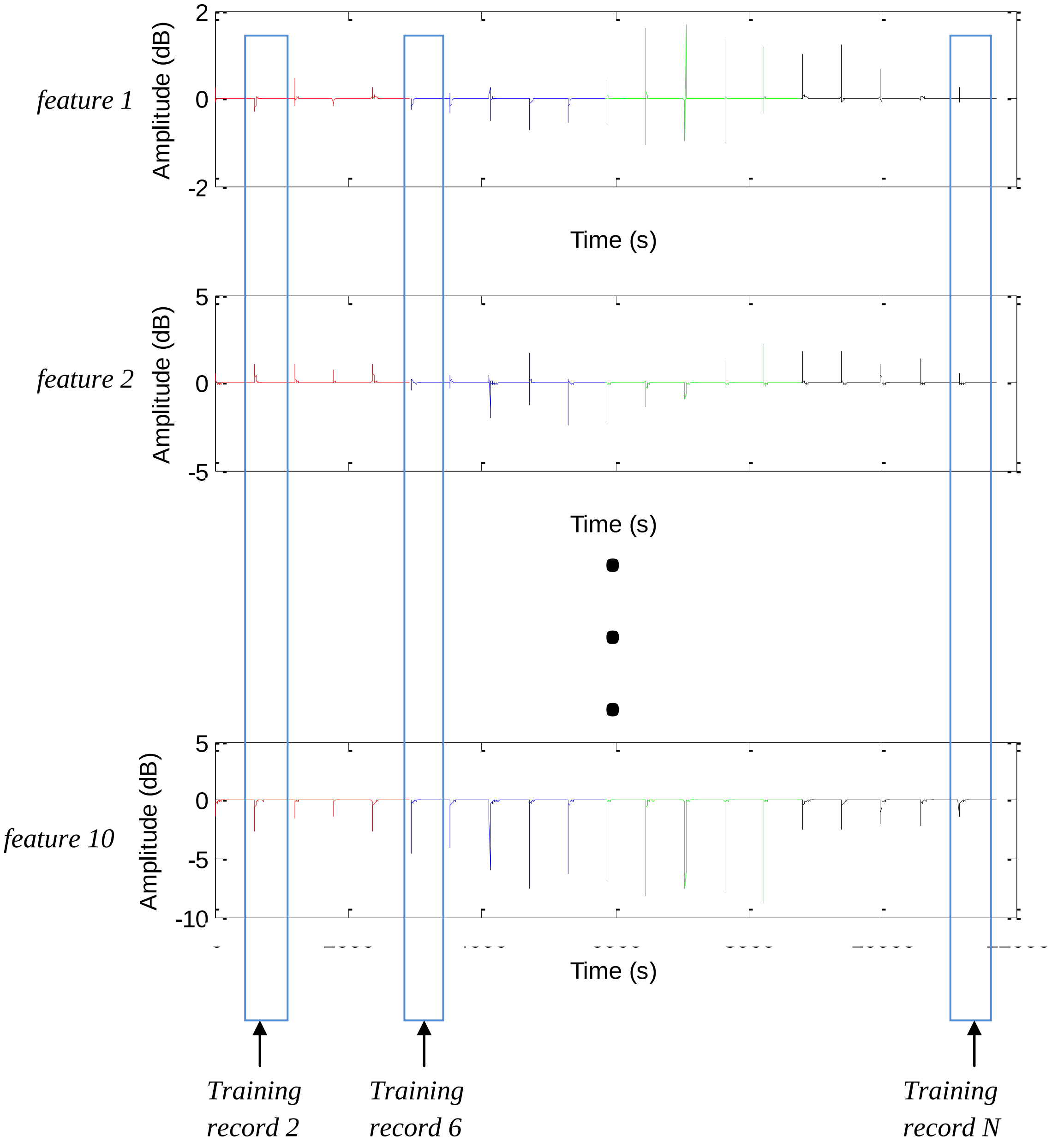


Figure 5.8: A set of 10 features extracted with PCA for spectral analysis and how we use it to build the training data set.

- **Classification of the tested data**

To classify a sound from the test data set, we do not need to perform ICA or PCA over its spectrogram and compare the obtained features with the ones from our training data. Instead, we use the orientation vector matrix that was previously learned by ICA or PCA (that is, matrix A).

As we have seen after performing ICA and PCA we obtain two matrixes A and X and we use sub matrixes A' and X' to build the training data set: A' of size $(F \times M)$, and X' of size $(M \times NT)$ with M being the number of used features.

$$\underset{F \times NT}{S} = \underset{F \times M}{A} * \underset{M \times NT}{X}$$

Now, let's assume $S1$ is our test sound spectrogram instead of the spectrogram of all the training sounds

$$\underset{F \times T}{S1} = \underset{F \times M}{A} * \underset{M \times T}{X1} \Leftrightarrow$$

$$A^{-1}S1 = A^{-1} * A * X1 \Leftrightarrow$$

$$X1 = A^{-1}S1$$

$X1$ is the matrix with the features represented according to the orientation vectors stored in matrix A (that was learned by ICA or PCA of the training data). The obtained matrix $X1$ is the one we use for classification.

As we have seen, each record of our training data is a matrix $(M \times F)$ or $(M \times T)$ just like the matrix we obtained above. Each column of this matrix represents the value of M features in one instant of time or for one frequency bin depending on the analysis (spectral or temporal) we are using.

We do not need to use all of the columns of these matrixes in the classification method due to the features usual behavior. Since this dissertation is based on impact sounds with very short duration, using few time instants is enough to describe the sound and distinguish its class. This is shown by figure 5.9 where we compare the values of the first feature of one aluminum sound and one wood sound extracted with temporal analysis respectively. It turned out that when values of the features are spectra (extracted by spectral analysis) it is also not necessary to use the whole spectrum to distinguish sounds from different classes. We obtained equally good results using the whole spectrum and only a sub-spectrum (figure 5.10).

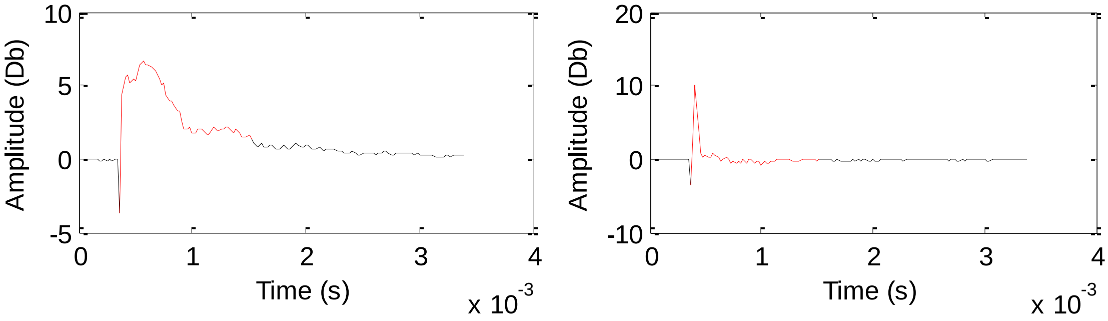


Fig 5.9: Temporal feature extracted with ICA of an aluminum sound (left) and one wood sound (right). In red, the instants we use in the classification.

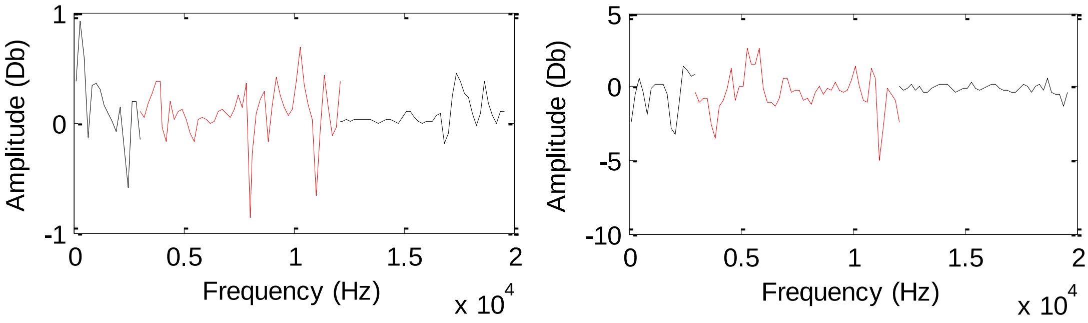


Fig 5.10: Spectral feature extracted with ICA of an aluminum sound (left) and one wood sound (right). In red, the frequency bins we use in the classification.

The 1-NN algorithm compares each (tested) column of the matrix $X1$ obtained for the test sample with the respective columns of the N records of the training data set. For each tested column, the algorithm returns the index of the training record whose column is nearer the one obtained for the tested sound. Each training record has a class assigned. The class that occurs more often in the amount of columns we use is the one assigned to the tested sound.

6. Result Analysis

We have conducted several experiments to test our recognizer. In this section we analyze the results obtained after doing both spectral and temporal analysis with both ICA and PCA. In the end, we validate our results: we test our recognizer with MFCCs and compare the results with the ones we obtained with our features.

Our data consists of 18 sounds of aluminum, 15 of zinc plated steel, 16 of wood and 15 of steel. We build two training sets for all analysis we perform: one with 20 sounds (five of each class) and one with 40 sounds (ten of each class). The remaining sounds are the sounds we use to test our recognizer.

- **ICA – Temporal Analysis**

We can see the results of temporal analysis with ICA in figure 6.1. When using the smaller training set, we obtained recognition rates of 92% for aluminum, 90% for steel with zinc, 100% for wood and 80% for steel. When using the bigger training set, the rates were 80% for zinc plated steel and 100% for all other sounds.

- **ICA – Spectral Analysis**

The results of spectral analysis with ICA can be seen in figure 6.2. We obtained rates of 100% for aluminum and zinc plated steel sounds, 73% for wood sounds and 60% for steel sounds with the smaller training set. On the other hand, we obtained rates of 83% for wood sounds and 100% for the rest of them when using the bigger training set.

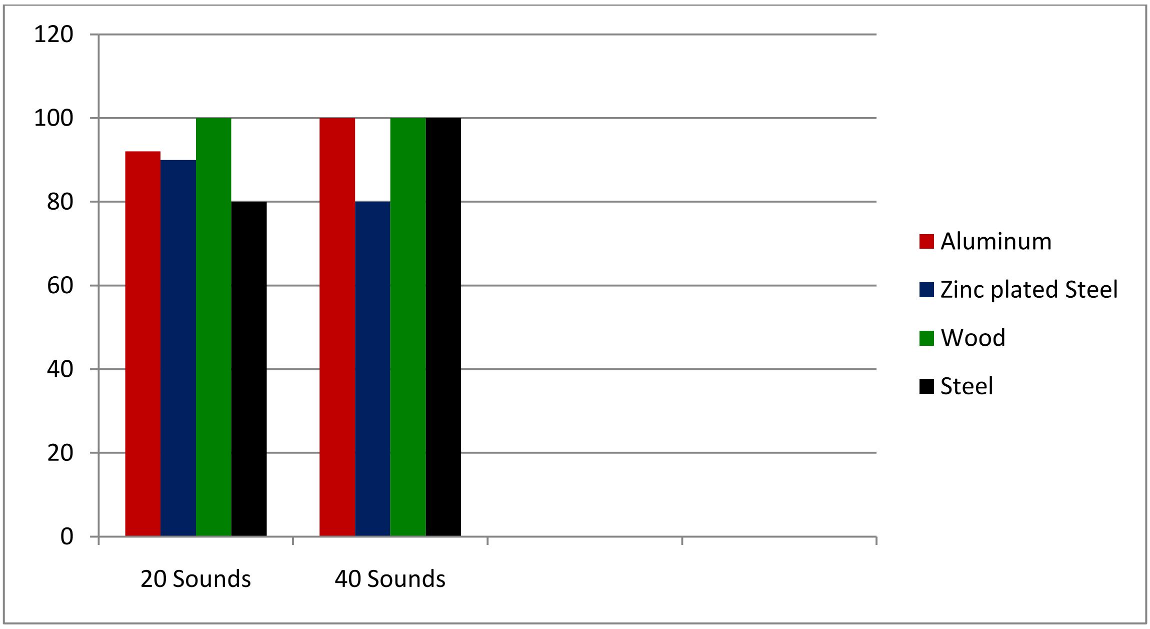


Fig 6.1: Temporal analysis with ICA results.

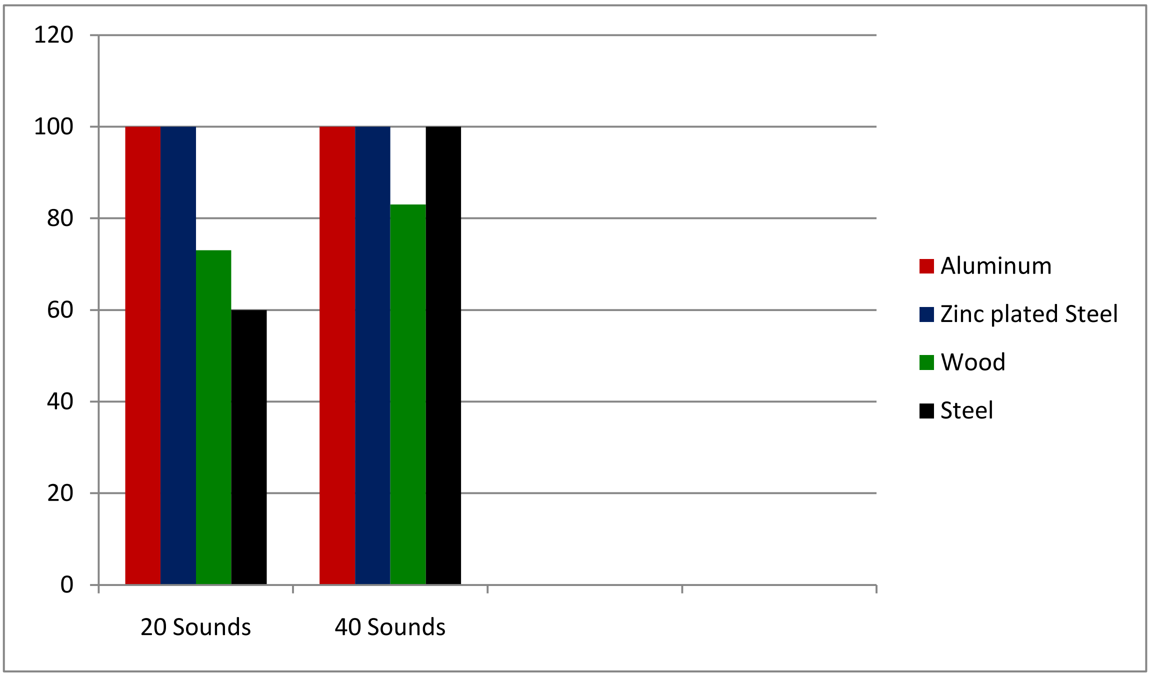


Fig 6.2: Spectral analysis with ICA results.

If we compare the spectral analysis results with the temporal analysis results we see the difference is minimal. When using the bigger training set the results are overall the same. The difference is that, in temporal analysis, one sound of zinc plated steel is not well recognized (which results in a recognition rate of 80% for this rod) and in spectral analysis this happens with a wood sound (which results in recognition rate of 83% for this rod). When using the smaller training set, we only get 100% recognition with wood sounds when doing temporal analysis. We suspect that the temporal functions from the other rods have more similarities and are harder to distinguish. If we look at the spectral analysis results, we get a recognition rate of 100% in both aluminum and zinc plated steel sounds. However, the rates for wood and steel sounds are smaller. We only got 60% for the steel sounds as the recognizer classified 40% of the steel sounds as aluminum instead. Nonetheless, the overall results are excellent and show how powerful the features extracted with ICA can be for similar sound recognition.

- **PCA – Temporal Analysis**

The results of temporal analysis with PCA are shown in figure 6.3. When using the smaller training set, we obtained recognition rates of 100% for all sounds except for steel sounds where we get 70% rate. When using the bigger training set, the rates are always 100%.

- **PCA – Spectral Analysis**

The results of spectral analysis with PCA are shown in figure 6.4. When using the smaller training set, we obtained recognition rates of 100% for aluminum sounds, 90% for zinc plated steel sounds, 91% for wood sounds and 90% for steel sounds. When using the bigger training set, the rates are 100% for all materials except for wood sounds where we obtain a recognition rate of 83%.

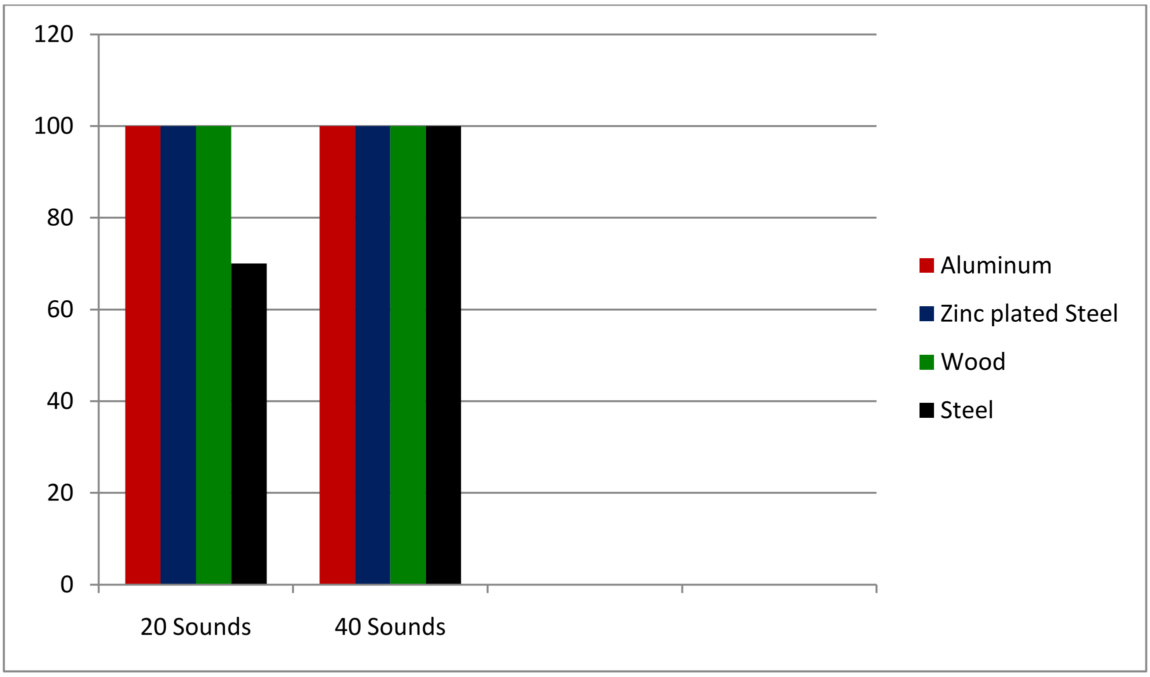


Fig 6.3: Temporal analysis with PCA results.

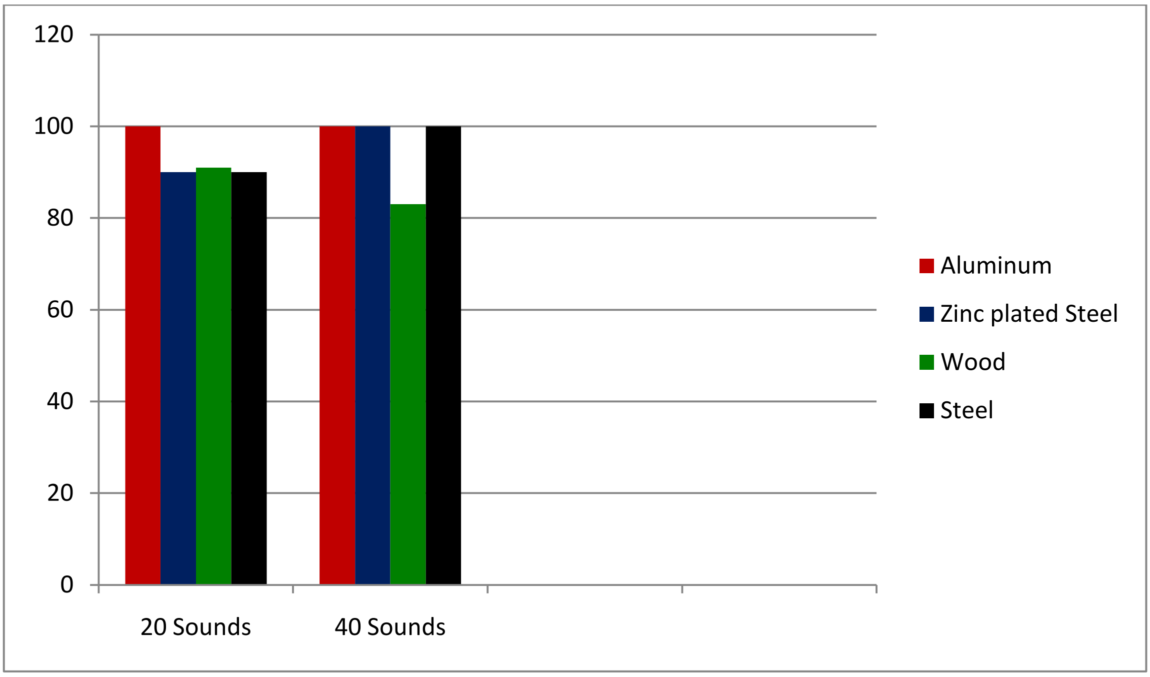


Fig 6.4: Spectral analysis with PCA results.

The results from both spectral and temporal analysis with PCA are very similar. When using the bigger training set, we get an overall recognition rate of 100% with temporal analysis and only one wood sound was not well recognized with spectral analysis (retrieving a recognition rate of 83% for this rod).

When using the smaller training set, we registered three steel sounds badly recognized with temporal analysis. On the other hand, when we performed spectral analysis our recognizer failed to classify three sounds as well: one of zinc plated steel, one of wood and one of steel.

Comparing these results to the ICA results, the difference is minimal. The overall results show PCA provided slightly better rates. We suppose this happens because when selecting the ICA set of features we use, we sort it by variance and use the first ones. Instead, we could find the features that describe the difference between the materials and we suppose we could retrieve better results even using fewer features than the ones we use. However, the main conclusion these results allow us to make is that both ICA and PCA are really powerful techniques to extract sound features.

We can also compare the spectral analysis with the temporal analysis in general. Although temporal analysis also retrieves slightly better results, it is hard to conclude if it is really better than spectral analysis, too. Initially we thought spectral features would retrieve better results since the sounds are so similar. However, these techniques are powerful enough to extract the temporal features which distinguish the sounds.

- **MFCCs**

As previously mentioned, MFCCs are very popular in sound recognition. MFCCs consist in a set of coefficients that describe the power spectrum of a sound. Since MFCCS are so popular and we want to see which features perform better, we used MFCCs with our data to compare their results with those obtained with our features.

The results of replacing our features by MFCCs in our recognizer are shown in figure 6.5. We use the *kannumfcc* Matlab function [34] and extract 11 coefficients which we use in our recognizer. When using the smaller training set, we obtained recognition rates of 83% for aluminum sounds, 70% for zinc plated steel sounds, 64% for wood sounds and 50% for steel sounds. When using the bigger training set, the rates are 100% for all aluminum and zinc plated steel sounds while we get a rate of 67% for wood sounds and 60% for steel sounds.

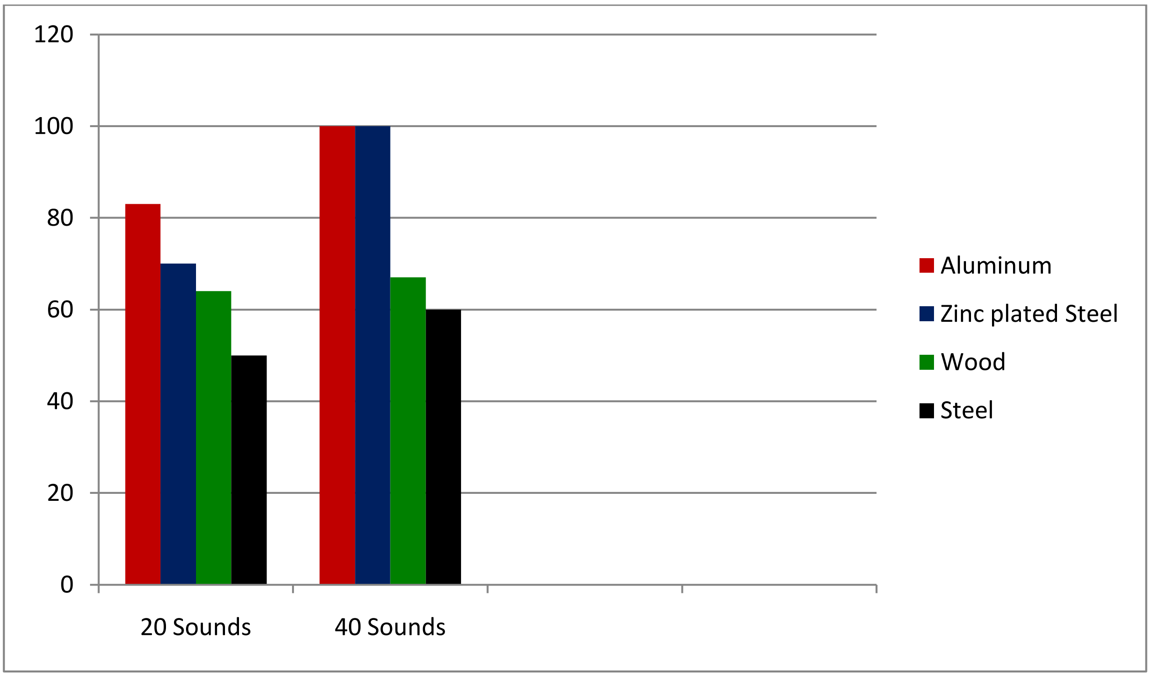


Fig 6.5: Results using MFCCs.

MFCCs with the lower training set provide a low overall recognition rate of 66.75%. We can see clearly that increasing the training set provides much better results. However, even with the bigger training set the results obtained with MFCC are lower than any other analysis we previously performed.

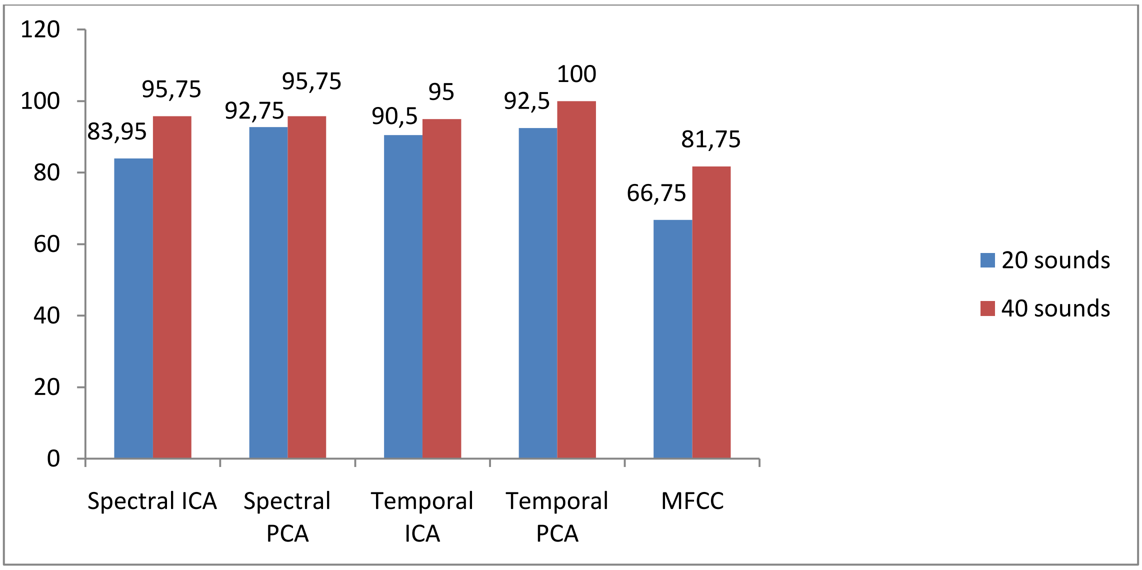


Fig 6.6: Overall recognition rates provided by all the analyses made.

Figure 6.6 shows the comparison of all the analysis made. The temporal analysis with PCA retrieves the best results achieving 100% recognition in all four classes of sounds we have. The other three analyses we performed only were mistaken in one sample when using a training set of 40 sounds. Even with a smaller training set of 20 sounds, both temporal and spectral analyses retrieved an overall recognition above 83% when using ICA and above 90% when using PCA.

Compared to the results we obtained using MFCCs, any of our features set is much better. MFCC with the bigger training set only returns an overall recognition rate of 81,75% which is smaller than any recognition rate any analyses (even with the smaller training set) we made before provided.

It is important to remember the sounds are very similar. This makes the recognizer task more difficult. Therefore, the MFCCs results can be seen as good. However, this just shows the features learned by ICA and PCA are much stronger in our recognizer and consequently in environmental sound recognition.

7. User Study

As seen in the previous section, the proposed recognizer gives excellent recognition rates, which are even higher than those obtained when we substitute the features learned by the ISA method by MFCCs, which, as mentioned above, are very commonly used in tasks of this nature. A natural question that follows is if the recognizer can surpass human ability to classify similar sounds and if the sounds are actually hard to distinguish. To answer these questions, we conducted two user studies.

In these studies, the subjects heard sounds from impacts on four rods and were asked to try to identify the class of the sound (that is, if the sound is from an aluminum rod, a wooden rod, etc.). The sounds used in the studies were the same as those used to train and test our recognizer. Below, we give more details about the protocols and the results. Lastly, we analyze the results by material and compare them with the results our recognizer obtained.

- **Protocol 1 – user study without feedback**

Before the actual test starts, the users hear two sounds of each of the four classes (aluminum, steel, zinc plated steel and wood). A dialog box (figure 7.1) is presented that indicates the type of sound that is going to be played and that guarantees that the users only hear the sound when they are ready. In order to have no presentation order effects, the order of presentation of the sounds varies so that different subjects can hear the sounds in different orders but the sounds from the same class are always presented consequently. For example, the user may first listen to two aluminum sounds, then two wood sounds, etc. or he/she may first hear two steel sounds, then two aluminum sounds, etc. The same sounds were used for all subjects.

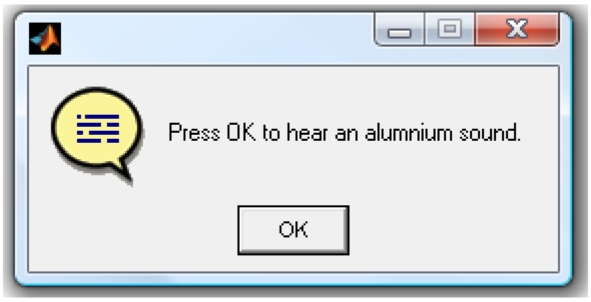

Press OK to hear an alumnium sound.

Figure 7.1: Dialog box for the first listening samples

Afterwards, subjects were explained that they would hear sounds from the same four classes that they heard in the first part of the test and that they had to identify the class of the sound. In order to familiarize the subjects with the process there were eight training trials (with no feedback): using the dialog boxes presented in figure 7.2 and figure 7.3 the user hears eight sounds presented randomly (two from each class) and tries to identify the material of the rod that produced the sound. The order of the buttons varies from user to user but the sounds used are always the same.

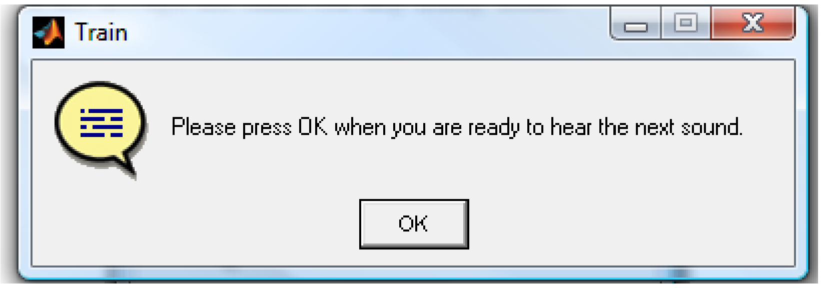


Figure 7.2: Dialog box to listen to the train and test samples

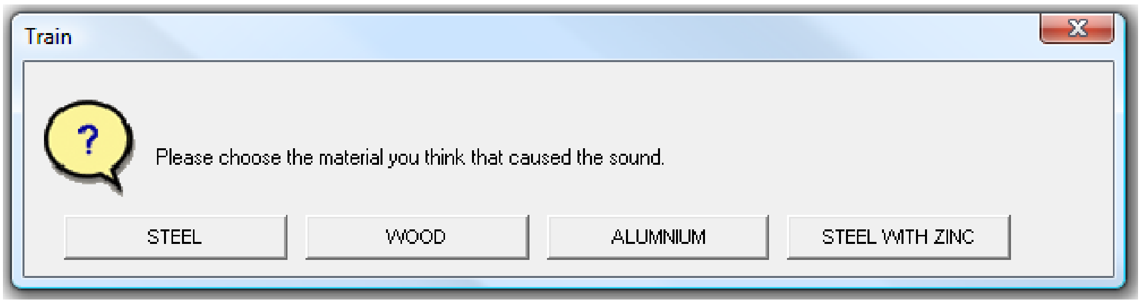


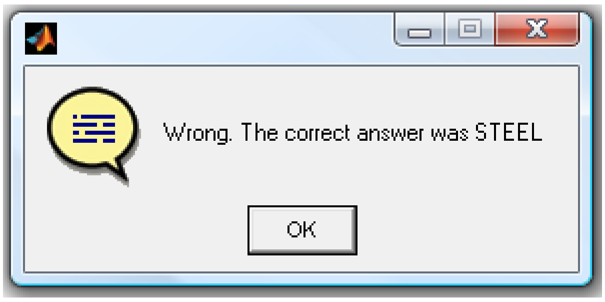
Figure 7.3: Dialog box to identify the material that caused the heard sound

Then the actual test begins with the same dialog boxes (figures 7.2, 7.3). The subjects heard 41 sounds present in random order and with no repetitions: 7 of aluminum, 11 of zinc plated steel, 12 of wood and 11 of steel. The sounds presented are the same for all tests done, only the order of presentation differs. The study is performed in a laptop using headphones.

- **Protocol 2 – user study with feedback**

While doing the user study described above, we concluded it was very difficult for the users to distinguish the sounds. Therefore, we conducted a new user study very similar to the previous one but in which we provided feedback in the training phase.

Like before with protocol 1, first the users heard two sounds from each class. Also like before, they were explained that in the next stage they would hear sounds from the same four classes that they heard in the first part of the test and that they had to identify the class of the sound. They are presented 12 training trials but here they receive feedback on whether their answers were correct or wrong: the subjects heard 12 sounds (three from each class) and tried to identify the class of the rod which caused the sound (figures 7.2 and 7.3). After the user identifies the class, a dialog box (figure 7.4) pops up with the answer: “correct” if the user identifies the class with success or “wrong” with the right answer otherwise.



Wrong. The correct answer was STEEL

Figure 7.4: Dialog box with the right answer during the training samples.

Lastly, like in protocol 1, the actual test begins. The user, using the dialog boxes of figures 7.2 and 7.3, tries to identify the classes of 32 sounds: 7 of aluminum, 10 of zinc plated steel, 5 of wood and 10 of steel. The sounds used are always the same, only the order changes. The study is performed in a laptop using headphones.

- **Results**

The first user study we conducted had 12 participants with ages between 23 and 55. None had hearing problems and two had acoustics knowledge. The results can be seen in table 7.1.

	Aluminum	**Zinc plated steel**	**Wood**	**Steel**
Aluminum	**22.619%**	53.571%	0%	23.81%
Zinc plated steel	43.939%	**24.242%**	0%	31.818%
Wood	1.3889%	0%	**97.917%**	0.69444%
Steel	28.03%	25.758%	0%	**46.212%**

Table 7.1: The table of answers of our first user study. Each row states the answers obtained for each material.

For aluminum sounds, there were only 22.619% right answers. When mistaken, 53.571% of the answers were zinc plated steel and 23.81% were steel. No user thought that an aluminum sound came from a wooden rod. For zinc plated steel sounds, we registered 24.242% right answers. Like with aluminum sounds no user mistaken zinc plated steel with wood. However, 43.939% of the answers were aluminum and 31.818% were steel. The set of answers for the wood sounds had much better results: 97.917% of right answers. On the other hand, we had three wrong answers: two answers with aluminum (1.3889%) and one with steel (0.6944%). Finally, we had 46.212% right answers for the steel sounds. For the wrong answers, 28.03% were aluminum and

25.758% were zinc plated steel. Like with aluminum and zinc plated steel sounds, none of the wrong answers of the steel sounds were wood.

The second user study we conducted provided better results as the users had more “training” than in the first user study. We had 11 participants with ages between 23 and 50. None had hearing problems and one had a high level of acoustics knowledge. The results are shown in table 7.2.

	Aluminum	**Zinc plated steel**	**Wood**	**Steel**
Aluminum	**53.247%**	23.377%	0%	23.377%
Zinc plated steel	26.364%	**40.909%**	0%	32.727%
Wood	0%	0%	**100%**	0%
Steel	27.273%	28.182%	0%	**44.545%**

Table 7.2: The table of answers of our second user study. Each row states the answers obtained for each material.

For the aluminum sounds, we registered 53.247% of right answers. 23.377% of the answers were wrong stating zinc plated steel and 23.377% stating steel. Like in the first user study nobody has mistaken wood for aluminum. For the zinc plated steel sounds, we saw 40.909% right answers. 26.364% of the answers were mistaken for aluminum and 32.727% for steel. The wood sounds we got 100% of right answers. Finally, we registered 44.545% of right answers for the steel sounds. 27.273% of the answers were mistaken for aluminum and the rest (28.182%) for zinc plated steel.

- **Result Analysis**

The results show that the sounds used in this dissertation are very hard for humans to distinguish. The low percentage of right results for aluminum and zinc plated steel sounds registered in the first study made us do a new user study with more training for the users. The results improved for these materials in the second study. The percentages of right answers are inferior to 55%, though. This shows how hard it was for the users to distinguish the metal sounds. The random order of the sounds during test also influences the results because people tend to forget the characteristics of the sounds they memorized before.

The mistaken answers also reveal that the sounds are so similar. For instance, in the second study, when missing the aluminum answer, we detect the same percentage of wrong answers for steel and zinc plated steel. The users assume it can be any of the three metals when confused.

On the other hand, the wood sounds are easy to recognize for the users. The sound from the wood rods is really different from the sounds of the metal rods. This is proved by the 100% right answers we registered in the second user study we performed.

The percentage of steel sounds recognition was much higher than the percentage of right answers for aluminum or zinc plated steel in the first user study. However, it did not improve in the second user study where users got more training.

We did register in the second user study, one test where the user had a high knowledge of acoustics. He only missed six answers out of thirty-two: three zinc plated steel sounds and three steel sounds. This shows the sounds have in fact different characteristics and can be distinguished from each other. It is just hard for the untrained human ear to do so.

We can see in figure 7.5 the comparison of the results obtained in the two user studies with the ones obtained from our worst case scenario recognizer (spectral ICA) and the ones obtained using MFCCs. Our recognizer has much better recognition rates than those of humans. Even with MFCCs, which retrieve much worse results than our recognizer, the system gives better recognition rates than those of humans. The only case where humans register more correct answers is for the wood sounds. However, our recognizer also registers 100% rate for wood sounds with temporal ICA and temporal PCA.

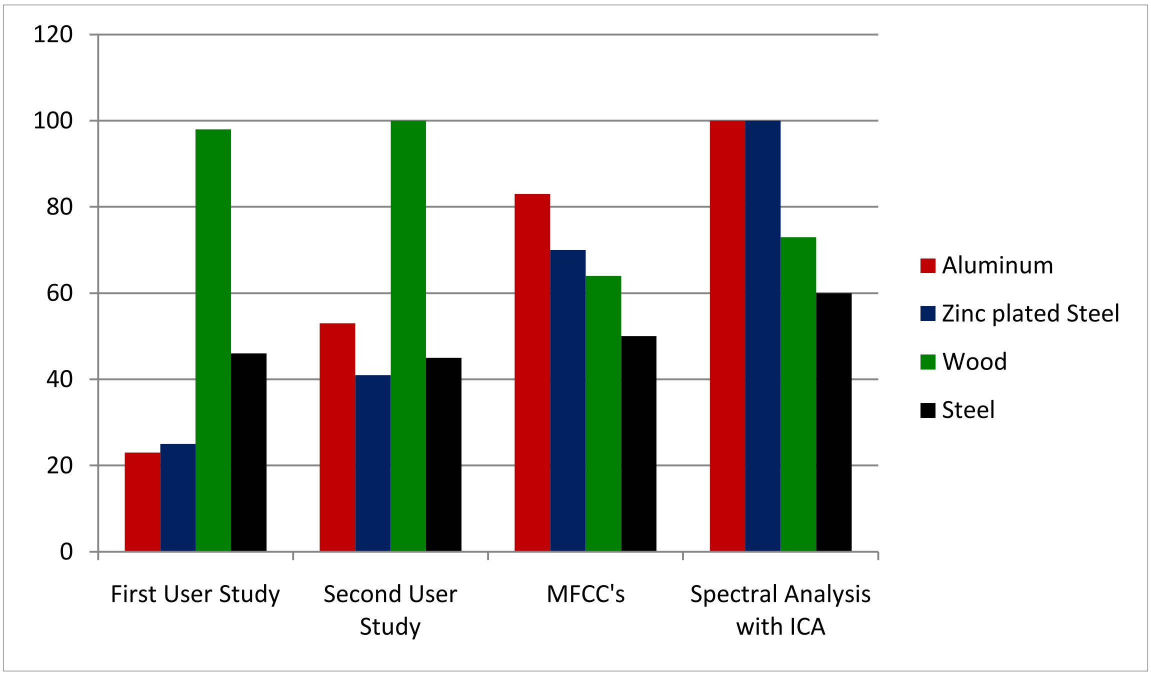


Figure 7.5: Comparison of the recognition rates obtained by our user studies with the ones obtained with MFCCs and with Spectral Analysis with ICA (both with training set of 20 sounds).

8. Conclusions

A system for recognizing very similar sounds has been described. This environmental sound recognizer uses the features retrieved by the ISA method which in turn uses ICA and PCA to learn them. Afterwards, the recognizer uses these features to train a 1-NN algorithm that can then be used to classify new sounds.

The results obtained give very good recognition rates. We performed several tests with different sets of features and in our worst case scenario it retrieved an overall recognition rate of 83.5%. However, increasing the number of training samples of our recognizer improved these results for an overall recognition rate of 95.75%. In our best case scenario, the recognizer retrieved an overall recognition rate of 100%.

We used four different set of features: we made two different analyses with both ICA and PCA. First, we did temporal analysis where we extracted the temporal properties from the sound spectrogram. Then, we made spectral analysis where we extracted the spectral properties of the sound.

It is difficult to say which analysis is better because the results are very similar. The higher rate we obtained with the smaller set was with spectral analysis with PCA (92.75%). On the other hand, the higher rate we obtained with the bigger training set was with temporal analysis with PCA (100%). Therefore, both retrieve really good results and it is hard to say one is better.

Comparing ICA to PCA, PCA returns slightly better results. However, the difference is not significant since it is not over 5%. Both techniques are really powerful to perform both spectral and temporal analysis.

In order to validate our recognizer, we, then, tested it with MFCC features instead of our features since MFCCs are really popular in sound recognition and, usually, return good results. We obtained an overall recognition of 66.75% with the smaller training set and 81.75% with the bigger set. Considering the smaller recognition rate we obtained with our features with the bigger training set was 95% we can conclude our features are much more powerful for sound recognition than MFCCs at least with the 1-NN algorithm in the classification method.

Finally, to compare human ability to distinguish our sounds from our recognizer's ability, we performed two user studies where users had to hear a sound and then identify what material caused it. The sounds used are the same as those used to train and test our recognizer: impact sounds made by rods which only differ in their material. We concluded that the metallic rods (composed by aluminum, zinc plated steel or steel) were really hard to distinguish. Only nearly half of the metallic sounds were well identified in the second user study. On the other hand, the wood sounds were very easy to identify for the users since in our second user study all of them were well identified. However, the overall recognition rate is much smaller than our recognizer's. We concluded the sounds are, in fact, very similar and hard to distinguish which makes the recognizer's task harder. Nonetheless, the recognition rates obtained are very good even with few training samples proving ICA and PCA extract powerful features for sound recognition.

- **Future Work**

For future work, it is possible that other classification method retrieves even better results. We saw in section 3 that MFCCs combined with HMMs retrieved great results. However, as our features act as short-time features we implemented the classification method based on a 1-Nearest Neighbor algorithm since it always retrieved good results in sound recognition with this type of features. It does not necessarily mean 1-NN algorithm is really the best for these features and other methods should be tried such as k-NN with *k* higher than 1, GMM, neural networks, SVM, or *k*-means.

As seen above, the ISA method is able to learn both temporal and spectral features. We only used these sets of features separately. However, together, they could retrieve even better results. It would be interesting to see the type of improvement on the recognition rate that would result by combining both types of features.

9. References

[1] Baljeet Malhotra, Ioanis Nikolaidism and Janelle Harms. ***"**Distributed classification of acoustic targets in wireless audio-sensor networks**"***, in: Computer Networks: The International Journal of Computer and Telecommunications Networking, volume 52, issue 13, pages 2582-2593. Year of Publication: 2008.

[2] Hans-W. Gellersen, Albrecht Schmidt and Michael Beigl. *"Multi-Sensor Context-Awareness in Mobile Devices and Smart Artifacts"*, in: Mobile Network Applications, volume 7, issue 5, pages 341-351. Year of Publication: 2002.

[3] Joan C. Nordbotten, *"Multimedia Information Retrieval Systems"* web-book in *http://nordbotten.com/ADM/ADM_book/* . Year of Publication: 2008.

[4] MIT Artificial Intelligence Laboratory, webpage:

http://www.ai.mit.edu/projects/humanoid-robotics-group/

[5] European Robotics research Network, webpage: *http://www.euron.org/*

[6] Sofia Cavaco and Michael S. Lewicki, *"Statistical modeling of intrinsic structures in impact sounds"*, in: Journal of the Acoustical Society of America, vol. 121, n. 6, pages 3558-3568. Year of Publication: 2007.

[7] Ken C. Pohlmann, *"Principles of Digital Audio"* (edition 5). Year of Publication: 2000. Publisher: McGraw-Hill/TAB Electronics.

[8] Karlhein Gröchenig, *"Foundations of Time-Frequency Analysis"*. Year of Publication: 2000. Publisher: Birkhäuser Boston.

[9] Stéphane G. Mallat, *"A Wavelet Tour of Signal Processing"*. Year of Publication: 1999. Publisher: Academic Press.

[10] K. R. Rao, P. Yip, *"Discrete Cosine Transform: Algorithms, Advantages, Applications"*. Year of Publication: 1990. Publisher: Academic Press.

[11] Tristan Jehan. *"Creating Music by Listening"*. Massachusetts Institute of Technology, dissertation submitted September 2005.

[12] Zhu Liu, Yao Wang and Tsuhan Chen. "*Audio Feature Extraction and Analysis for Scene Segmentation and Classification*", in: The Journal of VLSI Signal Processing, vol. 20, numbers 1-2, pages 61-79. Year of Publication: 1998.

[13] Silvia Pfeiffer, Sptephan Fischer and Wolfgang Effelsberg. "*Automatic audio content analysis*", in: Proceedings of the fourth ACM international conference on Multimedia, pages 21-30. Year of Publication: 1997.

[14] E. Scheirer and M. Slaney. "*Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator*", in: Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97), vol. 2, page 1331. Year of Publication: 1997.

[15] A.J. Eronen, V.T. Peltonen, J.T. Tuomi, A.P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, J. Huopaniemi. "*Audio-based Context Recognition*", in: Audio, Speech, and Language Processing, vol. 14, issue 1, pages 321-329. Year of Publication: 2006.

[16] J. Breebaart and M. McKinney. "*Features for audio classification*", in Proc. SOIA2002, Philips Symposium on Intelligent Algorithms, Eindhoven. Year of Publication: 2002.

[17] George Tzanetakis, Georg Essl and Perry Cook. *“Audio Analysis using the Discrete Wavelet Transform”*, in: Proc. Conf. in Acoustics and Music Theory Applications. Year of Publication: 2001.

[18] Stavros Ntalampiras, Ilyas Potamitis and Nikos Fakotakis. *“Automatic Recognition of Urban Environmental Sounds Events”*, in: New Directions in Intelligent Interactive Multimedia , pages 147-153. Year of Publication: 2008.

[19] Selina Chu, Narayanan, S. and Jay Kuo, C.-C. . *“Environmental sound recognition using MP-based features”*, in: Acoustics, Speech and Signal Processing, pages 1-4. Year of Publication: 2008.

[20] A. Eronen. *“Musical instrument recognition using ICA-based transform of features and discriminatively trained HMMs”*, in: Signal Processing and Its Applications, vol. 2, pages 133-136. Year of Publication: 2003.

[21] Florian Kraft, Thomas Schaaf, Alex Waibel and Rob Malkin. *“Temporal ICA for Classification of Acoustic Events in a Kitchen Environment*”, in ICSA International Conference on Speech and Language Processing / Interspeech. Year of Publication: 2005.

[22] Jong-Hwan Lee, Ho-Young Jung, Te-Won Lee and Soo-Young Lee . *”Speech feature extraction using independent component analysis”*, in: Acoustics, Speech, and Signal Processing, vol. 3, pages 1631-1634. Year of Publication: 2000.

[23] Tetsuya Takiguchi and Yasuo Ariki. *“PCA-Based Speech Enhancement for Distorted Speech Recognition*”, in: Journal of Multimedia, vol. 2, no. 5, pages 13–18. Year of Publication: 2007.

[24] Nitin Sawhney and Pattie Maes. *“Situational Awareness from Environmental Sounds*”. MIT Media Lab. Final Report for Modeling Adaptive Behavior (MAS 738), 1997.

[25] Alessandro Bugatti, Alessandra Flammini, and Pierangelo Migliorati. "*Audio Classification in Speech and Music: A Comparison between a Statistical and a Neural Approach*", in: Applied Signal Processing, vol. 2002, issue 1, pages 372-378. Year of Publication: 2002.

[26] Stan Z. Li. *"Content-based Classification and Retrieval of Audio Using the Nearest Feature Line Method"*, in: Speech and Audio Processing, vol. 8, issue 5, pages 619-625. Year of Publication: 2000

[27] Lie Lu, Stan Z. Li and Hong-Jiang Zhang. *"Content-based audio segmentation using support vector machines"*, in: Multimedia Systems, vol. 8, nr. 6, pages 749-752. Year of Publication: 2001

[28] L. Ma, D.J. Smith and B.P. Milner. *"Context Awareness using Environmental Noise Classification*", in: Proceedings of Eurospeech, vol. 3. Year of Publication: 2003.

[29] Antti Eronen, Juha Tuomi, Anssi Klapuri, Seppo Fagerlund, Timo Sorsa, Gaëtan Lorho and Jyri Huopaniemi. "*Audio-Based Context Awareness Acoustic Modeling and Perceptual Evaluation*", in: Proc. IEEE International Conference on Audio, Speech and Signal Processing (ICASSP). Year of Publication: 2003.

[30] James V. Stone. *"Independent Component Analysis, A Tutorial Introduction"*. Year of Publication: 2004. Publisher: The MIT Press.

[31] Aapo Hyvärinen, Juha Karhunen and Erkki Oja. "*Independent Component Analysis*". Year of Publication: 2001. Publisher: Wiley-Interscience.

[32] I.T. Jolliffe. "*Principal Component Analysis*" (edition 2). Year of Publication: 2002. Publisher: Springer

[33] H. Gävert, J. Hurri, J. Särelä, A. Hyvärinen, *FastICA for Matlab 5.x,* version 2.1, January 15, 2001.

[34] O.Omogbenigun, *kannuMFCC for Matlab 5.x,* September 11, 2007.