Telecommunication Engineering Master

UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

DENOISING APPLIED TO SOUND EVENT DETECTION USING WEIGHTING NON-NEGATIVE MATRIX FACTORIZATION

Student: Andrés Romero de los Ríos

Supervisors: Dr. Fco Jesús Cañadas Quesada
Dr. Damián Martínez Muñoz

Department: Telecommunication Engineering Department

June, 2020

CONTENTS

1. ABSTRACT 10

1.1 RESUMEN 11

2. INTRODUCTION 12

2.1 Context 12

2.2. Sound characterization 13

2.2.1 Fundamental frequency 13

2.2.2 Sound intensity 13

2.2.3 Timbre 13

2.2.4 Attack, Decay, Transient, Onset, Offset 14

2.2.5 Duration 15

2.2.6 Single channel and stereo sounds 15

2.2.7 Harmonic and inharmonic sounds 16

2.2.8 Sounds of babies crying, broken glass and gunshots 17

2.2.8.1 Baby cry sounds 18

2.2.8.2 Gunshot sounds 20

2.2.8.3 Glass break sounds 23

3. OBJECTIVES 26

4. STATE OF THE ART 27

4.1 Event detection 27

4.1.2 CONVOLUTIONAL MIXTURES 29

4.1.3 SOURCE SEPARATION BASED ON ICA IN FREQUENCY DOMAIN 30

4.1.4 SOURCE SEPARATION BASED ON NMF 32

4.1.4.1 Basic NMF (unsupervised NMF) 36

4.1.4.2 Generalized NMF 37

4.1.4.3 Constrained NMF 40

4.1.4.4 Structured NMF 43

4.1.5 Convolutional Neural Networks (CNNs) 45

5. DEVELOPMENT AND IMPLEMENTATION 49

5.1 Pre-processing 50
5.2 Training 50
5.3 Noise dictionary learning by robust NMF 57
5.4 Source separation by supervised and weighted NMF 61
5.4.1 Frequency weighting 63
5.4.2 Temporal weighting based on event probability 68
5.4.3 Combined time-frequency weighting 72
5.5 Event detection 73
5.6 Post-processing 78
6. EXAMPLES 84
7. EVALUATION 93
7.1. Databases 93
7.2. Metrics 93
7.3. Optimization 95
7.4 Setup 100
7.5 Results 101
8. CONCLUSIONS 104
9. FUTURE LINES 105
10. ANNEXES 106
10.1 Graphical user interface 106
10.2 Minimization of Kullback-Leibler divergence in NMF 115
11. COMPUTATIONAL COST 117
12. REFERENCES 118

INDEX OF FIGURES

Figure 1. Example of a situation in a park where an event detection system would be useful to detect a bird, proposed in [2]..... 12
Figure 2. Different musical instruments playing a musical note with the same period T amplitude and duration time, proposed in [3]. 14
Figure 3.Onset, attack time, transient, decay and offset in an event, from [4]. 14
Figure 4. Multichannel system 5.1, proposed in [5]. 16
Figure 5. Spectrogram of a musical note Si on a piano, whose fundamental frequency Fo is 430 Hz. 17
Figure 6. Curve energy from a baby cry event. 18
Figure 7. Baby cry spectrogram, in which the different harmonics can be seen. .. 19
Figure 8. Fundamental frequency Fo , 450 Hz, from baby cry spectrogram 19
Figure 9. Normalized energy distribution between 0 and 1, for a baby cry event. This distribution shows that the most amoung of energy is concentrated in low and medium frequency. 20
Figure 10. Curve energy from a gunshot event. 21
Figure 11. Shock wave and Muzzle Blast from a gunshot sound, recorded with a microphone at a distance of 60 meters, proposed in [6]. 22
Figure 12. Gunshot spectrogram, in which is easy to see that the gunshot events are inharmonic sounds. 22
Figure 13. Normalized energy distribution between 0 and 1, for a gunshot event. This distribution shows that the most amoung of energy is concentrated in low frequency..... 23
Figure 14. Curve energy from a glass break event. 24
Figure 15. Glass break spectrogram, in which is easy to see that the glass break events are inharmonic sounds. 24
Figure 16. Normalized energy distribution between 0 and 1, for a glass break event. This distribution shows that the most amoung of energy is concentrated in medium frequency. 25
Figure 17. Basic source separation model, proposed in [9]. 29
Figure 18. Permutations problem in ICA frequency domain, from [12]. 31
Figure 19. Permutation solver is added, in order to solve the permutations problem, from [12]. 31

Figure 20. Example of a decomposition of an input spectrogram V into two matrix, spectral base dictionary W, and its temporal activations H, from [16]..... 33
Figure 21. MNMF scheme, proposed in [19]. 35
Figure 22. Different NMF models, proposed in [21]. 36
Figure 23. Example Convolutional Neural Network, proposed in [37]. 46
Figure 24. Example convolution operation from a convolutional layer, proposed in [35]. 47
Figure 25. Two fully connected layers using three channels, proposed in [35]..... 48
Figure 26. Max pooling with filter size (2,2), proposed in [38]..... 48
Figure 27. System implemented based on [39]. 49
Figure 28. STFT with overlap 51
Figure 29. Example hamming window with N=21..... 52
Figure 30. $VTrain$ for baby cry class. 53
Figure 31. $VTrain$ for baby cry class using zoom. 54
Figure 32. $VTrain$ for gunshot class. 54
Figure 33. $VTrain$ for gunshot class using zoom. 55
Figure 34. $VTrain$ for glass break class..... 55
Figure 35. $VTrain$ for glass break class using zoom. 56
Figure 36. Convergence NMF in the training phase 57
Figure 37. Convergence in noise dictionary learning..... 59
Figure 38. Input noisy spectrogram V, which contains a baby cry event around the second 5..... 59
Figure 39. Estimated noise spectrogram Ln from the input spectrogram V of the figure 43. 60
Figure 40. Sparse matrix S for λ=0. 60
Figure 41. Sparse matrix S for λ=0.9. 61
Figure 42. Convergence NMF in the source separation phase 63
Figure 43. Energy curve from input spectrogram V of the figure 44. 65
Figure 44. Average spectral template es from the training matrix $VTrain$. 65
Figure 45. Energy template corresponding to es, that represents how much normalized energy there is in all frequencies. 66
Figure 46. Smoothed noise spectrogram en , from Ln, figure 44..... 66

Figure 47. Energy template corresponding to en ,that represents how much normalized energy there is in all frequencies. 67
Figure 48. Frequency weights matrix $Gfreq$ for the input spectrogram V of the figure 43, which enhances the subbands between 8 KHz and 10 KHz. So this subbands have more contributions in constructing the event. 67
Figure 49.Filtered spectrogram $Vfiltered$, for the input spectrogram V of the figure 43 using its frequency weight matrix of the figure 53. 69
Figure 50. Energy curve from filtered spectrogram $Vfiltered$, figure 54..... 70
Figure 51. Temporal weights with the limit values rmin and rmax, for the input spectrogram V of the figure 44. 71
Figure 52. Temporal weights matrix $Gtemp$ for the input spectrogram V of the figure 43. In this matrix the frames near to the second five have the probaibilities higher associated to an event. 71
Figure 53. Combined weights matrix $Gfreq + temp$ for the input spectrogram V of the figure 43, whose enhanced area is in the frames around the second 5 and the subband near to 9 KHz. 72
Figure 54. Energy curve from smoothed noisy spectrogram VS composed by a glass break event, area inside the red rectangle, and noise. The dashed red line indicates the threshold value TH. 74
Figure 55. Signal after applying the threshold method. 74
Figure 56. Example of Energy curve from smoothed noisy spectrogram VS composed by a baby crying event with breaths, area inside the red rectangle, and noise. The dashed red line indicates the threshold value TH. 75
Figure 57. Signal after applying the threshold method, for the curve energy of the figure 61. 76
Figure 58. Energy curve from output spectrogram VS for the input spectrogram V of the figure 43. 76
Figure 59. Signal after applying the threshold method $OutputTH$, for the energy curve of the figure 63. 77
Figure 60. Output spectrogram $Voutput$, for the input noisy spectrogram of the figure 43. 78

Figure 61. Energy curve, composed of a baby cry event, area inside the red rectangle, and noise, from the output spectrogram VS , before applying the median filter..... 79
Figure 62. Energy curve, composed of a baby cry event, area inside the red rectangle, and noise, from the output spectrogram VS , after applying the median filter..... 80
Figure 63. Energy curve, composed of a gunshot event, area inside the red rectangle, and noise, from the output spectrogram VS , before applying the median filter..... 81
Figure 64. Energy curve, composed of a gunshot event, area inside the red rectangle, and noise, from the output spectrogram VS , after applying the median filter..... 81
Figure 65. Input noisy spectrogram V, which contains 2 baby cry events around the second 5 and second 19. 84
Figure 66. Frequency weights matrix for the input spectrogram V of the figure 70, which enhances the subbands between 6 KHz and 8 KHz. So this subbands have more contributions in constructing the events. 85
Figure 67. Temporal weight matrix $Gtemp$ for the input spectrogram V of the figure 70. In this matrix the frames near to the second 5 and the second 19 have the probaibilities higher associated to an event..... 86
Figure 68. Combined weights matrix $Gfreq + temp$ for the input spectrogram V of the figure 70, whose enhanced area is in the frames around second 5 and 19 and the subbands near between 6 KHz and 8 KHz..... 86
Figure 69. Clean spectrogram for glass break example, figure 70. 87
Figure 70. Input noisy spectrogram V, which contains a baby cry event around the second 25..... 87
Figure 71. Frequency weights matrix $Gfreq$ for the input spectrogram V of the figure 75, which enhances the subbands between 11 KHz and 16 KHz. So this subbands have more contributions in constructing the event. 88
Figure 72. Temporal weights $Gtemp$ matrix for the input spectrogram V of the figure 75. In this matrix the frames near to the second 24 have the probaibilities higher associated to an event..... 88

Figure 73. Combined weight matrix $Gfreq + temp$ for the input spectrogram V of the figure 75, whose enhanced area is in the frames around second 24 and the subbands between to 11 KHz and 16 KHz 89
Figure 74. Output spectrogram $Voutput$, for the input spectrogram V of the figure 75. In which the glass break event from the input spectrogram V is recovered eliminating the noise..... 89
Figure 75. Input noisy spectrogram V, which contains a gunshot event around the second 19..... 90
Figure 76. Frequency weights matrix $Gfreq$ for the input spectrogram V of the figure 80, which enhances the subbands between 2 KHz and 4 KHz and other subbands between 5 KHz and 8 KHz. So this subbands have more contributions in constructing the event..... 90
Figure 77. Temporal weight matrix $Gtemp$ for the input spectrogram V of the figure 80. In this matrix the frames near to the second 19 have the probaibilities higher associated to an event..... 91
Figure 78. Combined weights matrix $Gfreq + temp$ for the input spectrogram V of the figure 80, whose enhanced area is in the frames around second 19 and the subbands 2 KHz and 4 KHz. 91
Figure 79. Output spectrogram $Voutput$ for the input spectrogram V of the figure 80. In which the gunshot event from the input spectrogram V is recovered eliminating the noise..... 92
Figure 80. Comparative F-score (%) for different values of K, $\boldsymbol{\lambda}$ for baby cry..... 96
Figure 81. Comparative F-score (%) for different values of K, $\boldsymbol{\lambda}$ for gunshot..... 97
Figure 82. Comparative F-score (%) for different values of k, $\boldsymbol{\lambda}$ for glass break... 98
Figure 83. Comparative F-score (%) between different models 102
Figure 84. Comparative F-score (%) between events 103
Figure 85. Interface home screen..... 107
Figure 86. Interface file selector. 107
Figure 87. Warning from the interface, because input audio is not selected. 108
Figure 88. Interface screen when playing an audio. 109
Figure 89. Parameter list from the interface, to select the parameter that the user want to change of value..... 110

Figure 90. Parameters from interface with a new value, lambda=0.8 and threshold value = 3000. 110
Figure 91. Interface screen when finish the processing of the input audio. 111
Figure 92. Interface screen when is selected the weight matrix graph from the graphs menu. 112
Figure 93. Interface screen when is selected the weight matrix graph from the graphs menu. 112
Figure 94. Warning from interface because the graph rmin and rmax is not available because was chosen no temporal weighting previously. 113
Figure 95. Warning from interface because the graph weight matrix is not available because was chosen no weighting previously. 114
Figure 96. Interface screen when is pressed the button SAVE. 114

INDEX OF TABLES

Table 1. F-score (%) using different values of k, $\boldsymbol{\lambda}$ for baby cry 95
Table 2. F-score (%) using different values of k, $\boldsymbol{\lambda}$ for gunshot 97
Table 3. F-score (%) using different values of k, $\boldsymbol{\lambda}$ for glass break 98
Table 4.Error ratio and F-score (%) for different models and events 101
Table 5. Computational cost 117

1. ABSTRACT

Some sound events (screams, gunshots ...) are often associated with situations of danger and violence. Nevertheless, the noise interference that appears with them decreases the detection performance to a great extent. Most existing methods that employ a trained noise classifier have the problem that they cannot provide an optimal result when such noise does not resemble the trained noise or is highly variable in time. Therefore, the task of acoustic noise reduction in this acoustic scenario remains a great challenge for sound event detection. For this Master Thesis, we propose the design and development of a system implemented in MATLAB capable of reducing noise for the detection of sound events using a Weighted Non-Negative Matrix factorization (WNMF) approach. While the different time-frequency inputs within the spectrogram receive the same importance using a standard NMF approach, Weighted NMF (WNMF) can vary the importance of different components by introducing a weighting matrix into the standard NMF model, weighting each frequency band or time region as a compensation between its contributions to the construction of the target event class and the noise. In other words, WNMF forces the standard NMF decomposition to emphasize more the dominant frequencies or frames of the target event class. Next, we will create an audio database composed of mixtures with sound events and noises suitable for the correct evaluation of the implemented system, using widely used metrics in the detection of sound events, comparing with some standard state-of-the-art methods. Finally, an easy-to-use interface will be created.

1.1 RESUMEN

Algunos eventos sonoros (gritos, disparos…) están asociados a situaciones de peligro y violencia. Sin embargo, las interferencias ruidosas que aparecen junto con estos eventos sonoros, hacen que la detección de estos eventos sea más dificultosa. La mayoría de los métodos existentes que emplean un clasificador de ruido entrenado tienen el problema de no proveer un buen resultado, cuando el ruido no se parece al ruido entrenado, o este es variable con el tiempo. Por lo tanto, la tarea de reducir el ruido en estos escenarios acústicos continúa siendo un gran reto para la detección de eventos sonoros. En este trabajo fin de master, proponemos el diseño de un sistema implementado en MALTLAB, que sea capaz de reducir el ruido para la detección de eventos sonoros usando la factorización no negativa de matrices pesada (WNMF). Mientras que las diferentes entradas frecuencia-tiempo dentro del espectrograma reciben la misma importancia usando un enfoque básico de factorización no negativa de matrices (NMF), usando WNMF, puede variar la importancia de los diferentes componentes introduciendo una matriz de pesos al enfoque básico de NMF, asignando pesos para cada banda de frecuencia o intervalo de tiempo como una compensación entre sus contribuciones en la construcción del evento de clase objetivo y del ruido. En otras palabras, WNMF fuerza al estándar básico de NMF a enfatizar las bandas dominantes de frecuencia o los frames temporales de la clase de evento a detectar. Después, se creará una base de datos de audios, compuesta de grabaciones con eventos sonoros y ruido para la correcta evaluación del sistema implementado, empleando diferentes métricas en la detección de eventos sonoros, comparando después con algún método del estado del arte. Finalmente, será creada una interfaz de fácil uso para el usuario.

2. INTRODUCTION

2.1 Context

In recent years, separation of sound sources is being a highly research field within the area of signal processing. There are many systems that use this type of algorithms for different purposes, for example some systems are used to facilitate learning for musicians with a monitoring of the musical partiture, like Beatik app, with which the musicians can follow the score automatically on a tablet without having to manually switch sheets, others systems use this technology for speech recognition [1], but this project is focused on sound event detection. Sound events are defined like, individual sounds that transmit some type of information about what is happening in a situation, for example fire alarm, or glass breaking, etc. However, detecting these events is not an easy task, because the events signals can appear with a very low power level, in addition these events may be mixed with other undesirable signals, like people talking, children playing or noise from a car. The biggest problem of these undesirable signals, that are considered noise for us, is that are unpredictable and non-stationary. So, how to reduce the noise and how to generalize the varying noise is a difficult challenge for this algorithms.

In the following image 1, the problem explained above is shown, in this example the event detection system has to detect birds singing from a park, however the microphone located in the park capture an input signal with more undesirable sounds like speech, cars.

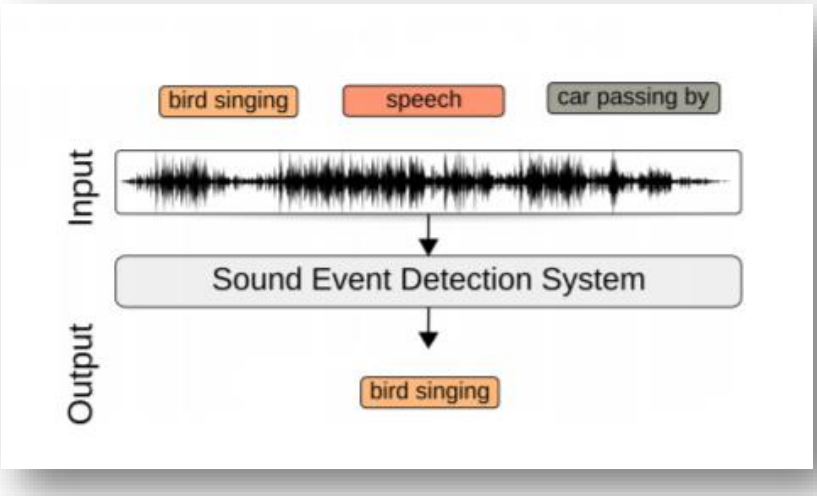

Figure 1. Example of a situation in a park where an event detection system would be useful to detect a bird, proposed in [2]

2.2. Sound characterization

2.2.1 Fundamental frequency

Considering a periodic signal $x(t)$, the fundamental frequency $F_o\ (Hz)$ is denominated like the inverse of the fundamental period $T_0\ (s)$, equation (1), being the fundamental period the minimum time interval that satisfies the equation (2).

$$F_o\ (Hz) = \frac{1}{T_0} \tag{1}$$

$$x(t) = x\ (t + T_0)\ \ \forall\ t \tag{2}$$

2.2.2 Sound intensity

Is the amount of power $P(W)$ transferred by a wave per unit area $A\ (m^2)$, perpendicular to the direction of the propagation. Power can be expressed as energy $E\ (W*s)$ divided by time t(s).

$$I\ (W/m^2) = \frac{P}{A} = \frac{E}{A*t} \tag{3}$$

2.2.3 Timbre

Is the property of sound that allows us to distinguish between two sounds with the same frequency and amplitude and duration, depending on the intensity and frequency of the harmonics that accompany the fundamental sound. The image below, figure 2, shows different musical instruments playing a musical note with the same period T amplitude and duration time.

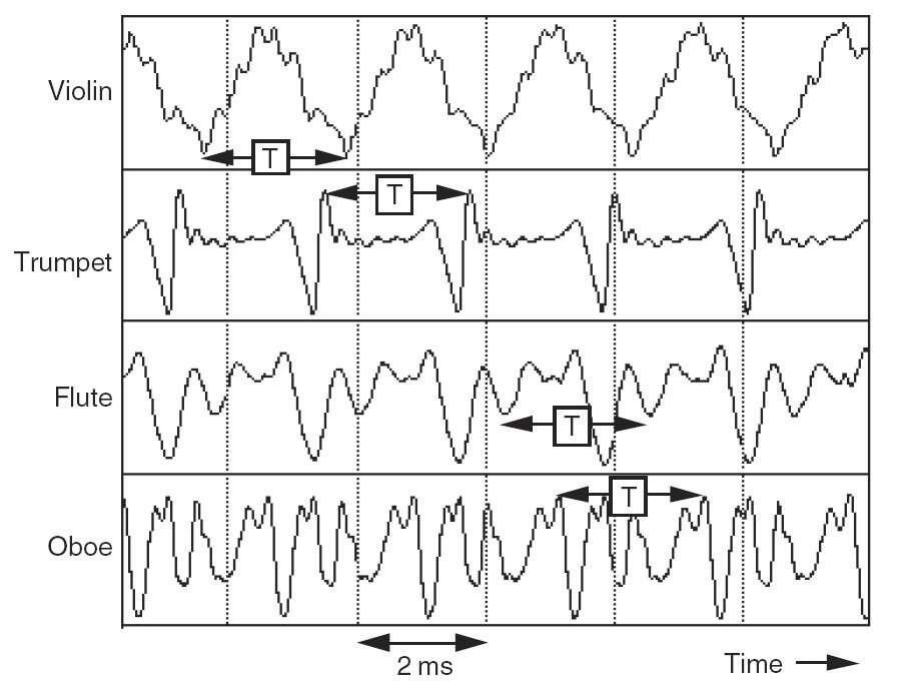

Figure 2. Different musical instruments playing a musical note with the same period T amplitude and duration time, proposed in [3].

2.2.4 Attack, Decay, Transient, Onset, Offset

Considering a sound event different zones can be distinguished: onset, Attack time, transient, decay and offset.

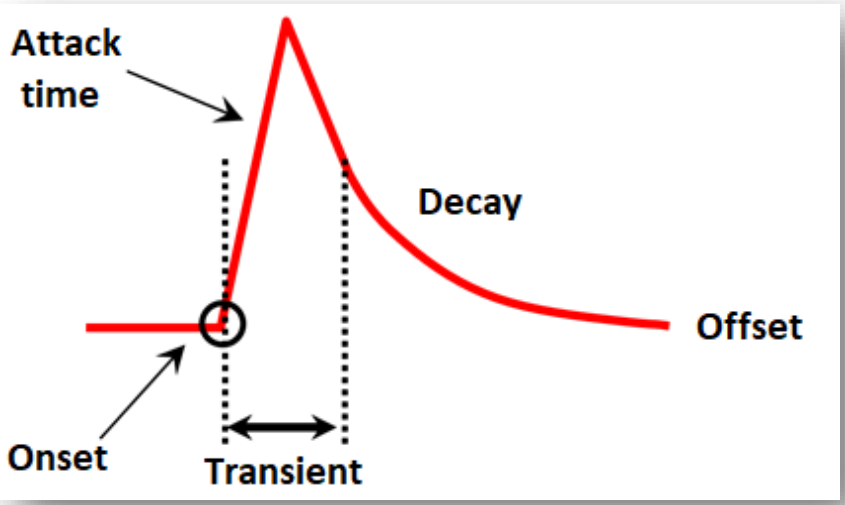

Figure 3.Onset, attack time, transient, decay and offset in an event, from [4].

- **Onset:** It is the starting point of attack time interval.
- **Attack time:** It is the time interval, in which the amplitude envelope increases.
- **Transient:** It is the time interval in which the sound is predominant.
- **Decay:** It is the time interval in which the amplitude envelope decreases.
- **Offset:** It is the endpoint of the decaying time interval.

2.2.5 Duration

It is the time in which the vibrations produced by a sound are maintained, can be expressed as:

$$duration(s) = onset(s) - offset(s) \quad (4)$$

Taking into account that onset is referred to the beginning of a sound event and offset to the end.

2.2.6 Single channel and stereo sounds

Depending of the number of channels can be distinguished:

- **Single channel:** This type of sounds are recorded by only one microphone, all sound sources are added, and therefore in this case, there isn't information about the location of the sound sources.

- **Stereo sounds:** It is a sound recorded by two microphones and played on two channels. These type of sounds give information about the location of the sound sources.

- **Multichannel audio:** For these sounds are used more than two microphones, m microphones, obtaining sounds with a number m of channels, providing spatial information.

A common example of a multichannel system can be 5.1 surround sound, which is used to play music, movies, creating audio environments that allow to the user to achieve an immersion for example, in the movies environment, or places like a cathedral, simulating its acoustic effects. This system is composed by 5 speakers (2 front stereo speakers, 2 rear stereo speakers and a front center speaker) and one subwoofer.

The image below, figure 4, shows a multichannel system 5.1.

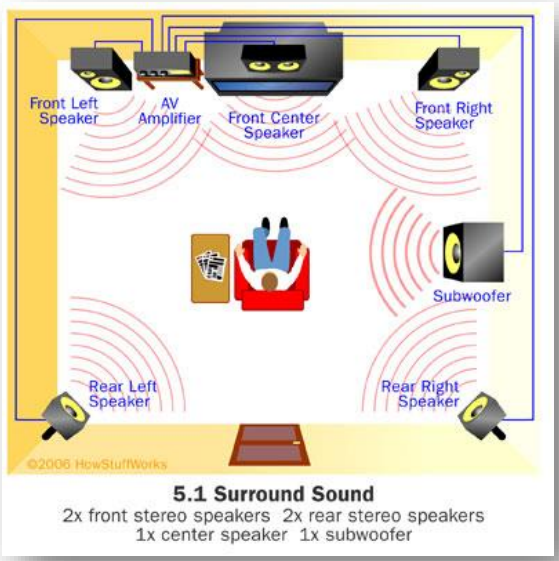

Figure 4. Multichannel system 5.1, proposed in [5].

2.2.7 Harmonic and inharmonic sounds

Taking into account the periodicity of the frequencies of a signal is possible to distinguish between:

- **Harmonic sounds:** Are those periodic sounds whose frequencies are integer multiples of the fundamental frequency F_o (Hz). Regarding, that the total number of harmonics is N_h, each harmonic, $Harmonic_n$ can be expressed like as:

$$Harmonic_n = n\,F_o \quad take\ into\ account \quad n = 1 \ldots . N_h \tag{5}$$

- **Inharmonic sounds:** Are those non-periodic sounds, whose frequencies are not integer multiples of the fundamental frequency F_o (Hz).

In the following image, figure 5, can be seen the spectrogram corresponding to a piano signal, whose fundamental frequency F_o is 430 Hz and its harmonics in integer multiples until reaching the total number of harmonics N_h.

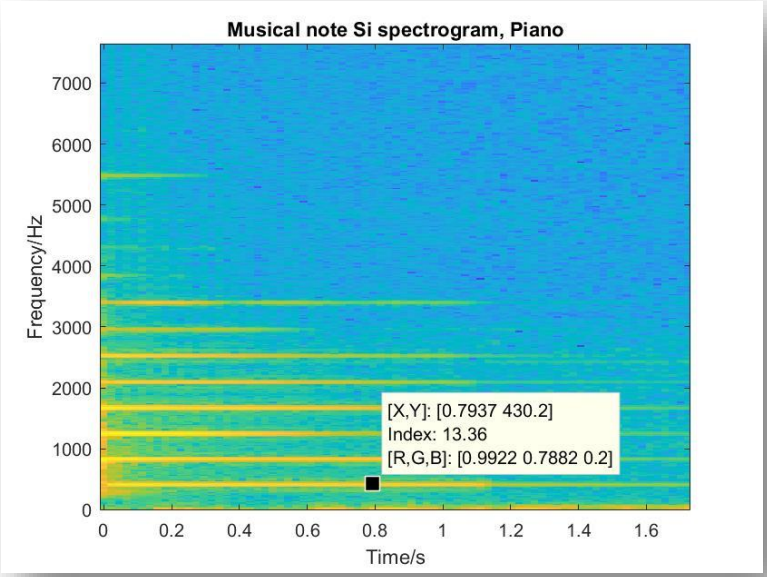

Figure 5. Spectrogram of a musical note Si on a piano, whose fundamental frequency F_o is 430 Hz.

2.2.8 Sounds of babies crying, broken glass and gunshots

In this section, a brief description of the characteristic for each kind of event (baby cry, gunshot and glass break) that will be used in the development of the system.

2.2.8.1 Baby cry sounds

Baby cry event are characterized by having different energy peaks, this is because the baby breathes while he is crying. Each baby cry area is defined by a fast attack, a transient zone with a duration of a few milliseconds and then a fast energy decay. The duration of these events is variable but it is always greater than 0.5 s .

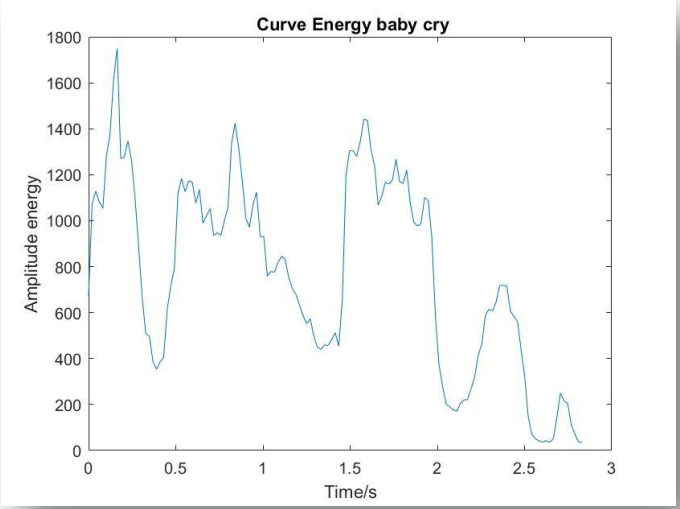

Figure 6. Curve energy from a baby cry event.

Harmonic sound

Another feature of baby cry events is that are harmonic signals, like can be seen in the following spectrogram, figure 7:

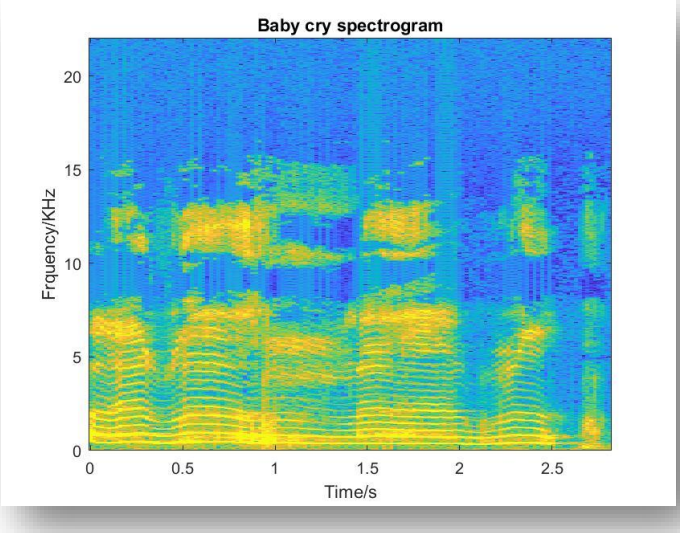

Figure 7. Baby cry spectrogram, in which the different harmonics can be seen.

Whose fundamental frequency F_o is around 450 Hz and its harmonics in integer multiples:

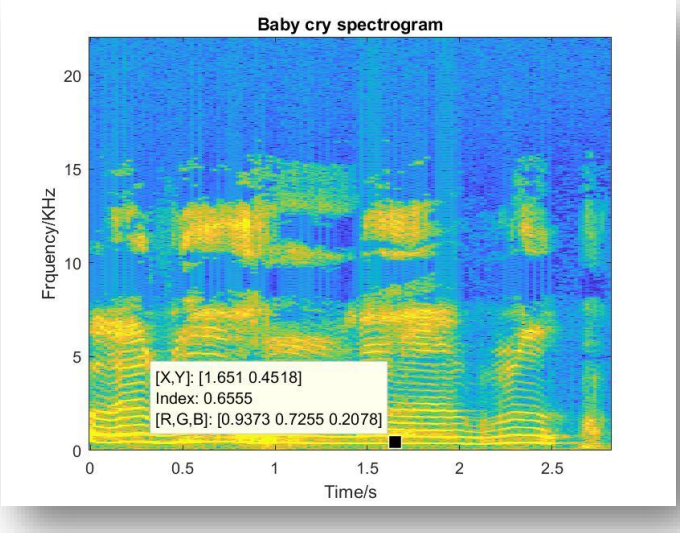

Figure 8. Fundamental frequency F_o, 450 Hz, from baby cry spectrogram

Distribution of energy in the different frequencies

The following graph, figure 9 shows the normalized energy distribution t between 0 and 1 for a baby cry event.

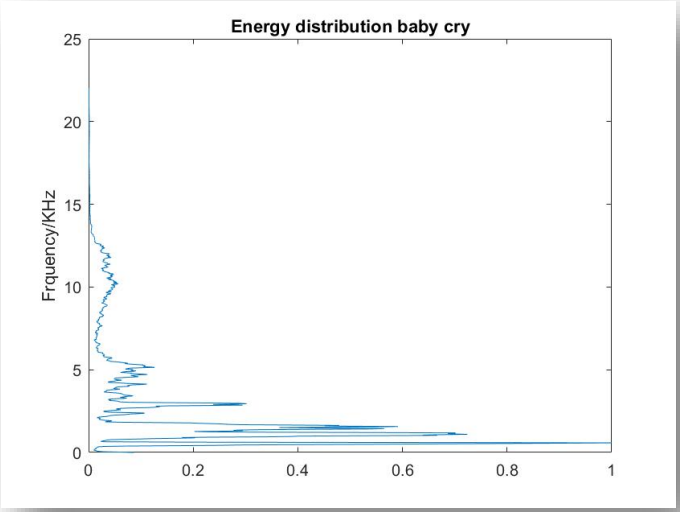

Figure 9. Normalized energy distribution between 0 and 1, for a baby cry event. This distribution shows that the most amoung of energy is concentrated in low and medium frequency.

As can be seen in the figure, in baby cry events, the energy is mainly concentrated in the low and medium frequencies especially between 0 KHz- 12 KHz.

2.2.8.2 Gunshot sounds

Gunshots events are characterized by having a fast attack, practically instantaneosly reaches the envelope peak and a short decay of energy. The duration of these events is short around 0.4 s.

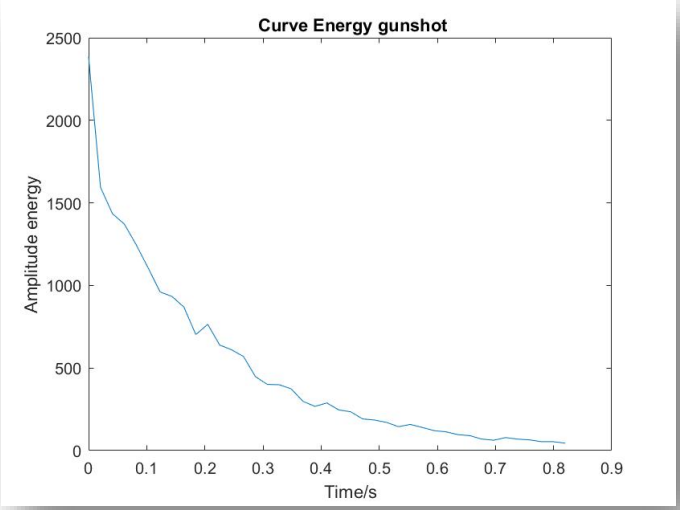

Figure 10. Curve energy from a gunshot event.

A gunshot event is composed of two parts, the explosion that discharges the bullet that is called muzzle blast and a shock wave generated when the bullet goes through the air faster than the speed of sound, this means that the shock wave would arrive to a microphone before that the muzzle blast. However to perceive these two parts the microphone has to be put far.

The following image, figure 11, shows a gunshot recording captured by a microphone at 60 m, in which the two parts of a gunshot event, shock wave and muzzle blast can be identified:

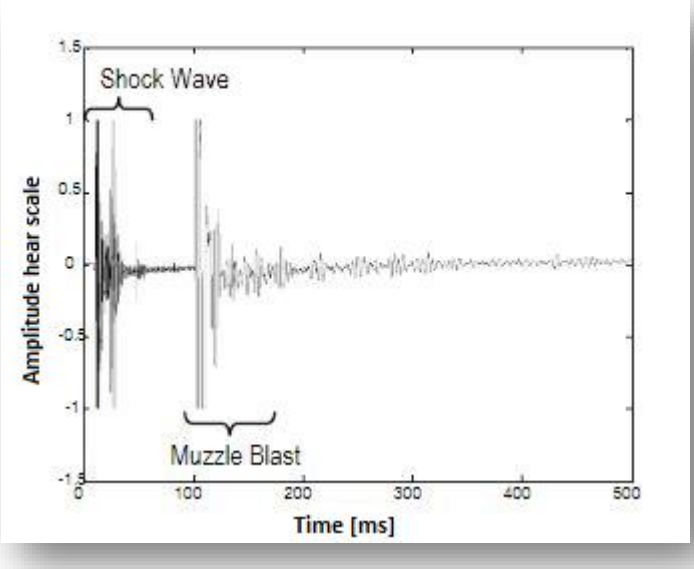

Figure 11. Shock wave and Muzzle Blast from a gunshot sound, recorded with a microphone at a distance of 60 meters, proposed in [6].

Inharmonic sound

As can be seen in the following spectrogram, figure 12, the gunshot events are not harmonic sounds, there is not periodicity between the frequencies.

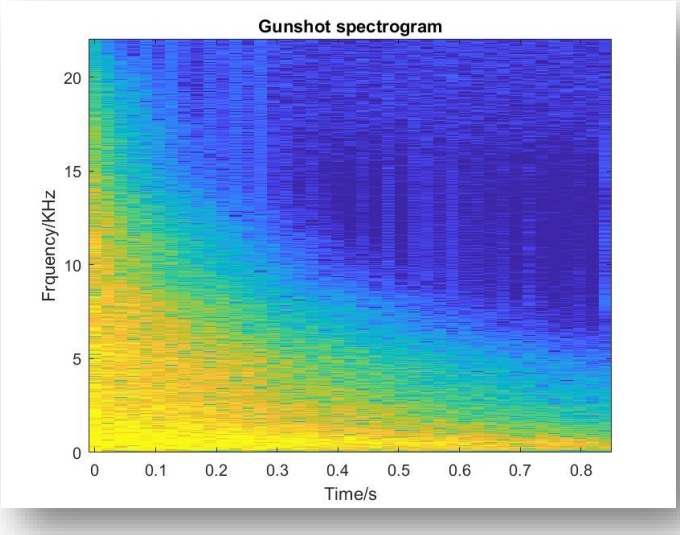

Figure 12. Gunshot spectrogram, in which is easy to see that the gunshot events are inharmonic sounds.

Distribution of energy in the different frequencies

The following graph, figure 13, shows the normalized energy distribution between 0 and 1 for the gunshot event.

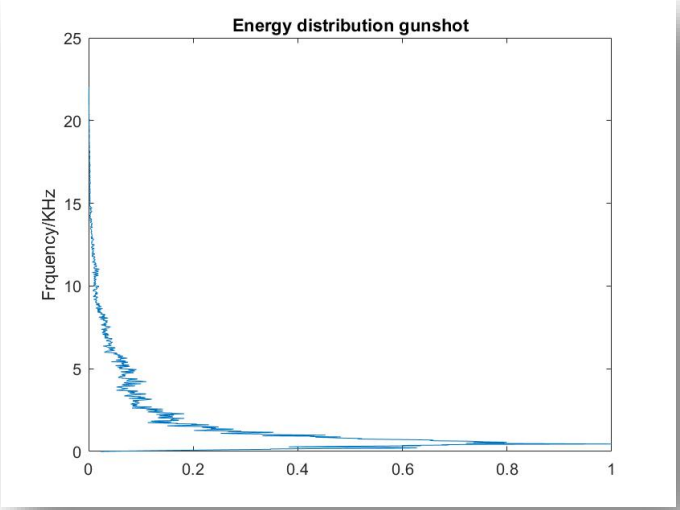

Figure 13. Normalized energy distribution between 0 and 1, for a gunshot event. This distribution shows that the most amoung of energy is concentrated in low frequency.

As can be seen in the figure 13, in a gunshot signal, most of the energy is concentrated in low frequency, which will be a problem, as will be seen later, because the noise also is concentrated in low frequency.

2.2.8.3 Glass break sounds

Glass break events like in the case of gunshots events, are characterized by having a fast attack, and then a quick energy decay. The duration of these events is very short around 0.3 s .

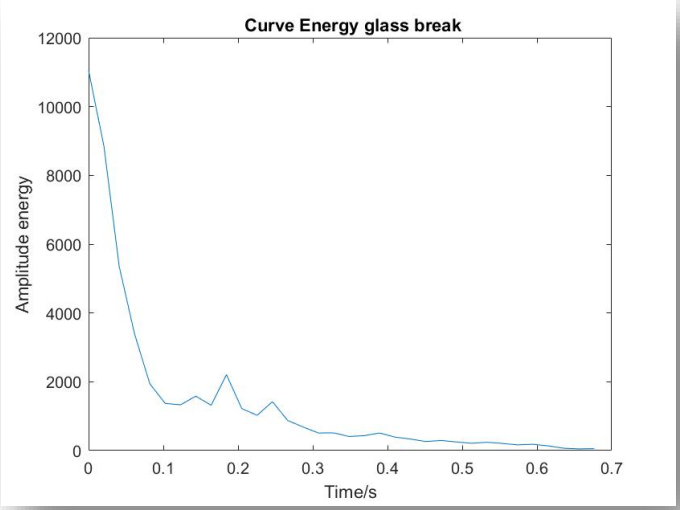

Figure 14. Curve energy from a glass break event.

Inharmonic sound

As shows the following spectrogram, figure 15, like in the case of gunshot events, the glass break events are not harmonic sounds, there is not periodicity between the frequencies.

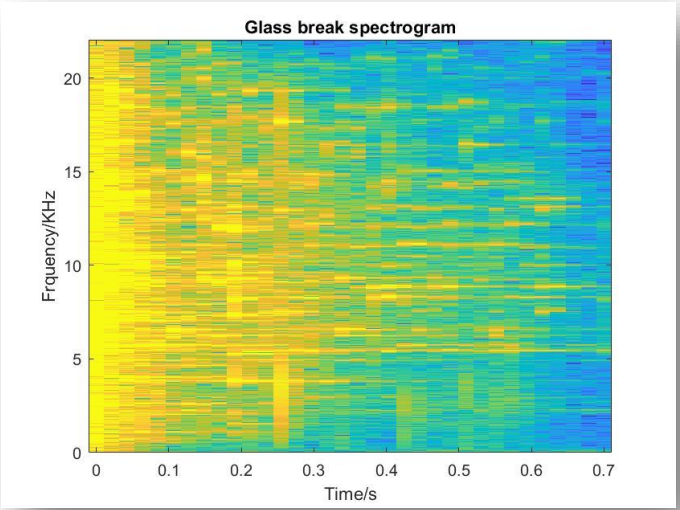

Figure 15. Glass break spectrogram, in which is easy to see that the glass break events are inharmonic sounds.

Distribution of energy in the different frequencies

The following graph, figure 16 shows the normalized energy distribution between 0 and 1 for glass break event.

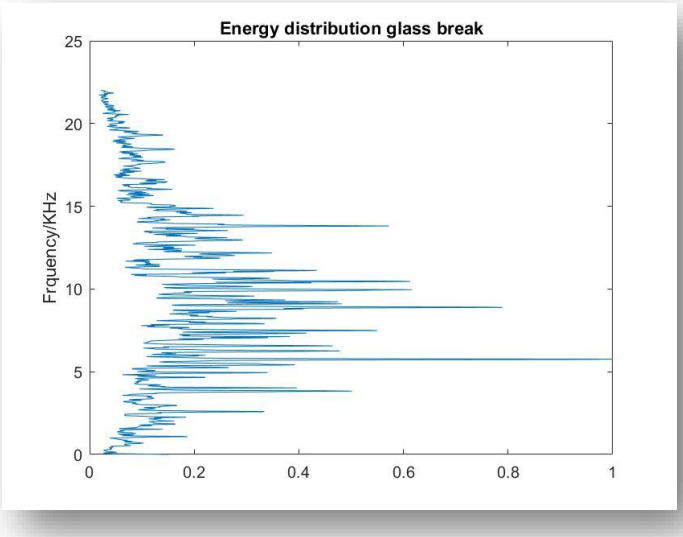

Figure 16. Normalized energy distribution between 0 and 1, for a glass break event. This distribution shows that the most amoung of energy is concentrated in medium frequency.

As shows the figure 16, in glass break events, the energy is concentrated especially in medium frequencies, having a more uniform distribution of energy compared to the others kind of events.

3. OBJECTIVES

We propose the design and development of a system implemented in Matlab capable of reducing noise for the detection of sound events using a weighted Non-Negative Matrix factorization (NMF) approach.

Also a series of secondary objectives are intended to achieve with the development of this project:

- Compilation and study of state-of-the-art references related to the main topic of this Master thesis, audio event detection and weighted-NMF.

- Creation of a database of audio mixtures composed of sound events and noises.

- Evaluation of the implemented system.

- Development of a Matlab user friendly interface for the interaction of the user with the system.

4. STATE OF THE ART

4.1 Event detection

Recent years, sound separation is a field that is being addressed by many researchers in the signal processing area. The first separation method were based on a model called, instant mixes model.

4.1.1 Instant mixes model

This model assumes that the sound sources reach to the reception sensor (microphone) without transmission delays, in other words the transmission distances are considered to be very short or with very high transmission speed. Each source will be affected by an attenuation coefficient a_{ij}, which takes into account the energy losses produce from the jth source to the ith sensor (microphone). Considering that the original sources are $s_{j\,(t)}$ and x_i(t) are the signals received by the sensors (microphones), these can be expressed as:

$$x_i(t) = \sum_{j=1}^{p} a_{ij}\, s_j\,(t) \tag{6}$$

Grouping into vectors, can be expressed:

$$x(t) = a\, s(t) \tag{7}$$

Or isolating the variable $s(t)$:

$$s(t) \approx a^{-1}\, x(t) \tag{8}$$

In order to make a separation is needed to find a separation matrix R, that when multiply by the signals received at the sensors, an estimation from the original sources $s(t)$, can be obtained:

$$s(t) \approx R\, x(t) \tag{9}$$

Matching the equation with equation (8) is obtained that:

$$R \approx a^{-1} \tag{10}$$

In the previous equations, the attenuations a and the original sources $s(t)$ are unknown, so there are more unknown variables than equations. For this reason, some assumptions are necessary to make, for example suppose that the original sources were uncorrelated and therefore, look for a separation matrix R that generates signals that are as uncorrelated as possible. One of the most important properties is the statistical independence of the sources, in order to maximize this independence, a cost function is proposed in order to maximize the independence of the separation matrix R, this statistical model is called ICA (Independent Component Analysis [7]), which has been applied in a different fields, such as image processing, proposed in [8].

The following image, figure 17, shows a basic source separation scheme:

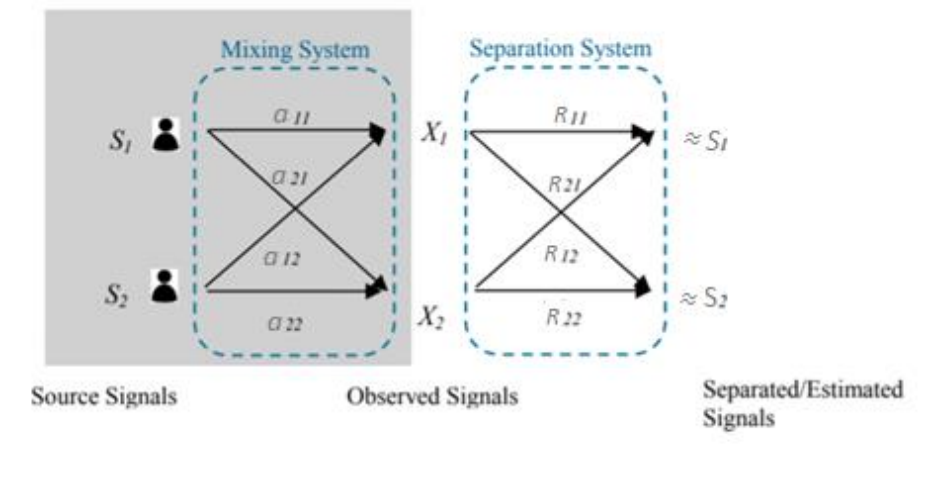

Figure 17. Basic source separation model, proposed in [9].

This method hasn`t good performance for audio, for example in an anechoic environment, should take into account the relative transmissions delays, or in a reverberant environment a convolutional mixing model should be applied, using impulse responses, which consider the behaviour of the sounds in the physical environments .

For this reason, in the 90 s, different approaches appeared generalizing the ICA methods to the case of convolutional mixtures in the temporal domain.

4.1.2 CONVOLUTIONAL MIXTURES

In this model [10], [11], the signals received in the sensors are described by the following equation (11), considering $h(t)$ as a mixing matrix, and the symbol $*$ the convolution operation:

$$x(t) = h(t) * s(t) \tag{11}$$

As in the previous case for the instant mixes model, a separation matrix is searched R, in order to estimate the original sources $s(t)$:

$$s(t) \approx R * x(t) \tag{12}$$

As explained before, R is optimized in order to maximize the independence using a ICA method .However the big problem is the computational cost due to the optimizations involve convolutions.

4.1.3 SOURCE SEPARATION BASED ON ICA IN FREQUENCY DOMAIN

The computational cost of the convolutional model can be reduced working in the frequency domain, because the convolutions become in multiplications. Taking into account that $H(w)$ is the mixing matrix in the frequency domain and $S(\mathrm{w}, \tau)$ the original sources in the frequency domain, the signals received by the sensors $X(w, \tau)$ are defined as:

$$X(w, \tau) = H(w) * S(\mathrm{w}, \tau) \tag{13}$$

Now, it would be applied an ICA algorithm but allowing work with complex signals.
The main problem of this algorithm can be visualized in the following image, figure 18, from [12], in which a scenario with two sources and two microphones is proposed:

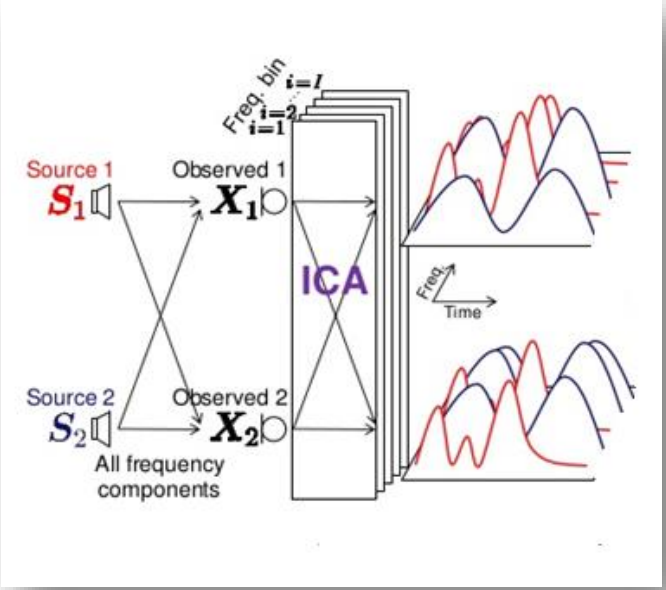

Figure 18. Permutations problem in ICA frequency domain, from [12].

As shown in the figure 18, complex ICA is applied to each frequency i, however are produced permutations, so the signals are mixed. A work developed by Murata and collaborators [13] was able to solve this problem. This work was based on the correlation between adjacent bins, comparing its amplitude correlations, if these amplitudes are similar belong to the same source.

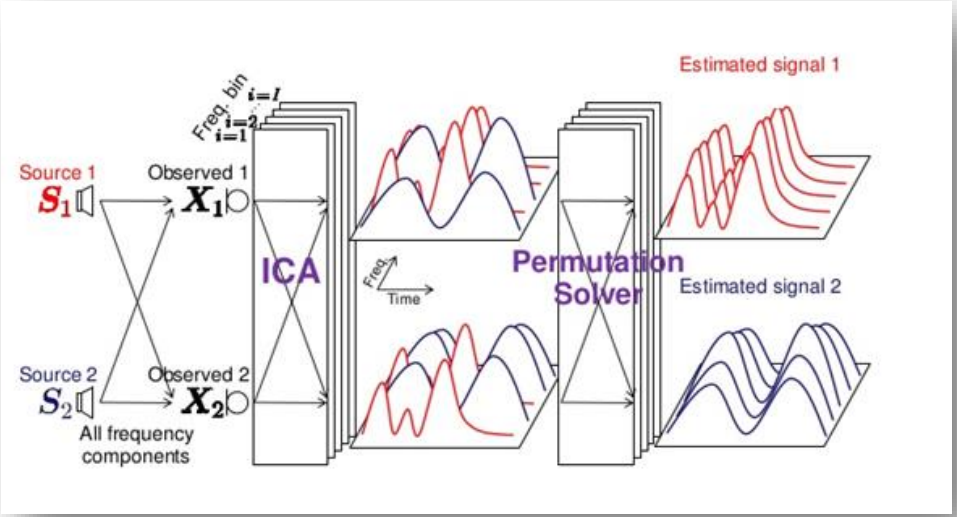

Figure 19. Permutation solver is added, in order to solve the permutations problem, from [12].

However, although these algorithms have a lower computational cost than in the time domain, so the application of these algorithms in real time is complicated. In addition this method had a bad performance when the reverberation time increased due to the reflections of the sound.

In 2009 another model was proposed [14], which simplified the computational cost and took into account the different delays of the sounds due to the reflections with the physical medium. For this was considered that the signals were acquired with very close microphones, of this way the impulse responses from one source to each microphone can be approximated as scaled and delayed versions. This model was called as, pseudo anechoic mixture model and the biggest advantage that this model had, was that it produced that the mixing matrix $H(w)$ for each frequency were related between them, achieving that the separation problem was solved of a global way, applying ICA only one time.
However, the ICA method needs as many microphones as sound sources, for this reason the interest in this type of algorithms has been decreasing.

4.1.4 SOURCE SEPARATION BASED ON NMF

As previously explained, ICA was not a very efficient method because the same number of microphones as sources were necessary. For this reason NMF (Non – Negative Matrix Factorization emerged in the proposed created by Lee Sung [15].

In first place, in order to work with this method the temporal input signal x_{input} composed of different sources is transformed to the frequency domain, using the STFT, and the magnitude spectrogram, V is taken. So taking the input spectrogram V of dimension $F \times T$, NMF factorize this V into two matrix, W of dimension $F \times K$ and H , of dimension $K \times T$, taking into account that F represents frequency bins, T time frames and K, number of spectral bases. The K columns of the matrix W represent a dictionary that contains the spectral bases of V, and the K rows of H represent to the temporal activations associated to these spectral bases.

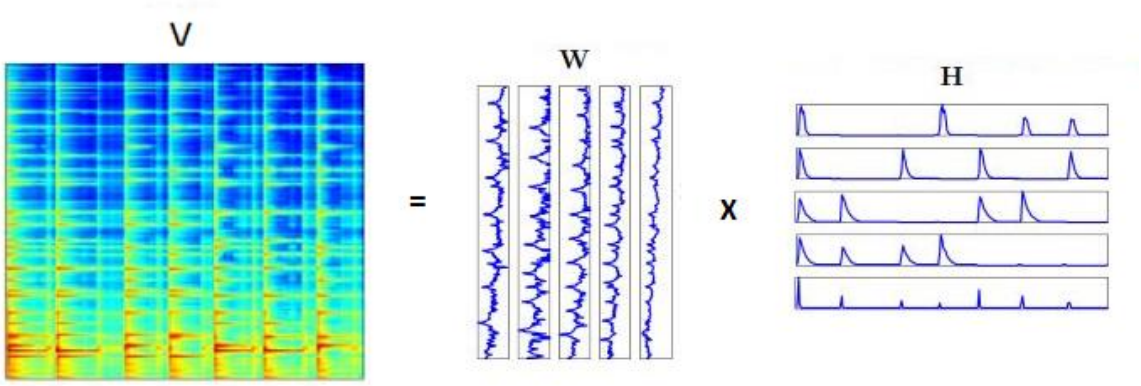

Figure 20. Example of a decomposition of an input spectrogram V into two matrix, spectral base dictionary W, and its temporal activations H, from [16].

In order to estimate the bases dictionary W and its temporal activations H an optimization equation is needed, under a non negative constrain, being the objective to reduce the re-construction error between $W$and H and V, for this is used a distance or cost function, such as Euclidean, Chebyshev, Kullbak Leibler, etc. Below, in equations (14) and (15), two cost functions are shown:

- Euclidean distance:

$$D_{EUC}\,(V|WH) = \|V - WH\|_F^2 \quad subject\ to\ V, W, H \geq 0 \tag{14}$$

Minimizing the Euclidean distance equation, solved in [12], is obtained:

$$W = W \circ \left(\frac{V}{WHH^T}\, H^T\right) \tag{15}$$

$$H = H \circ \left(W^T \frac{V}{w^T WH}\right) \tag{16}$$

In which ○ and / is referred to the element-wise multiplication and division of two matrices and the subscript T indicates the transposition of a matrix.

- Kullback-Leibler divergence:

$$D_{KL}(V|WH) = V \ \log\left(\frac{V}{WH}\right) - V + (WH) \qquad subject\ to \ \ V, W, H \geq 0 \tag{17}$$

Minimizing the Kullback-Leibler divergence equation, solved in annex 10.2, is obtained:

$$W = W \circ \left(\frac{V}{WH} H^T\right) / (1H^T) \tag{18}$$

$$H = H \circ \left(W^T \frac{V}{WH}\right) / (W^T 1); \tag{19}$$

In which ○ and / is referred to the element-wise multiplication and division of two matrices and the subscript T indicates the transposition of a matrix.
These parameters W and H are updated a number of iterations n, until the convergence of their cost functions in order to minimize the reconstruction error.

One of the first source separation works was developed by Smaragdis and Brown [17], who proposed a supervised approach of the algorithm. The main idea of this approach, taking into account that there are two sound sources, for example noise and a source sound of an event, is to train a concatenated array of clean signals for each one of the sources in order to obtain a dictionary of training bases, one for the sound events W_{source} and the other for the noise W_{noise}. Then using these dictionaries W_{source} and W_{noise}, temporary activations H_{source}, that corresponds to the events, are calculated from the input spectrogram V and thus being able to recover the spectrogram from the source multiplying the training bases W_{source} by its temporal activations H_{source}.

In recent years the basic model NMF has been extended for application in multichannel mixes [18], MNMF, this model takes into account acoustic and spatial characteristics of the sound. In first place, the spectrograms λ_{tfl} are calculated by the multiplication of the dictionary of bases that in this case is w_{kf}, the temporal activations, h_{kt} and the mixing parameters u_{kl}, $\lambda_{tfl} = \sum_k^K u_{kl}\, w_{kf}\, h_{kt}$. Then is calculated the spatial correlation matrix G_{fl} and at the end the multichannel complex spectrograms are generated as the sum multichannel complex spectrograms, $\sum_l^L \lambda_{tfl} * G_{fl}$. In order to perform the source separation MNMF estimate the basis w, temporal activations h and spatial correlation G matrices for each individual source.

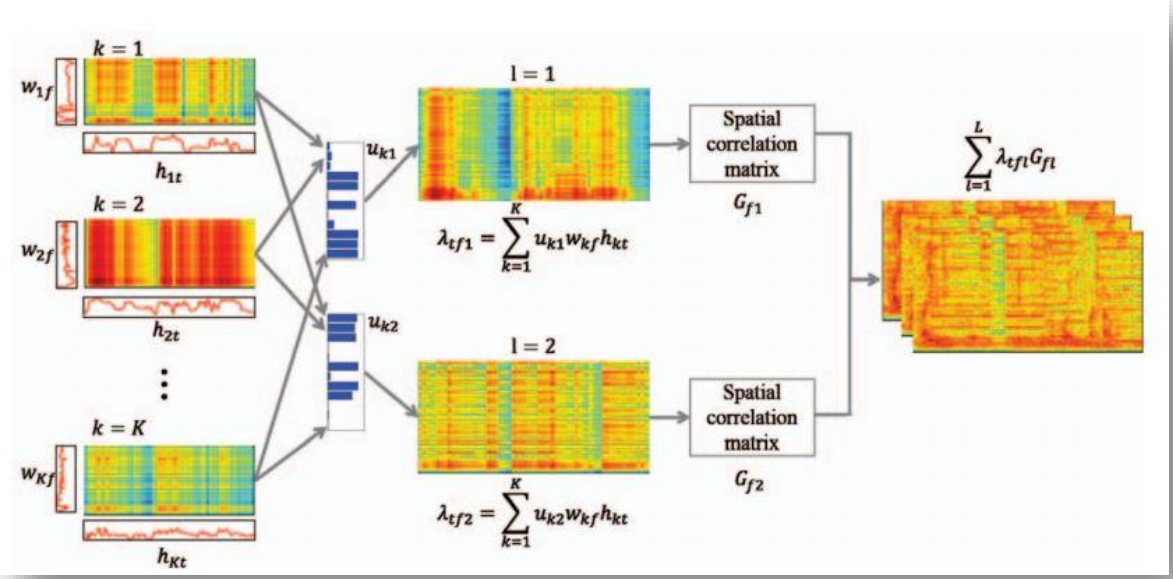

Figure 21. MNMF scheme, proposed in [19].

This line of research is under development and some of the latest algorithms developed for source separation [20] are based on it.

There are a lot of NMF models [21], a general scheme in figure 22 is provided:

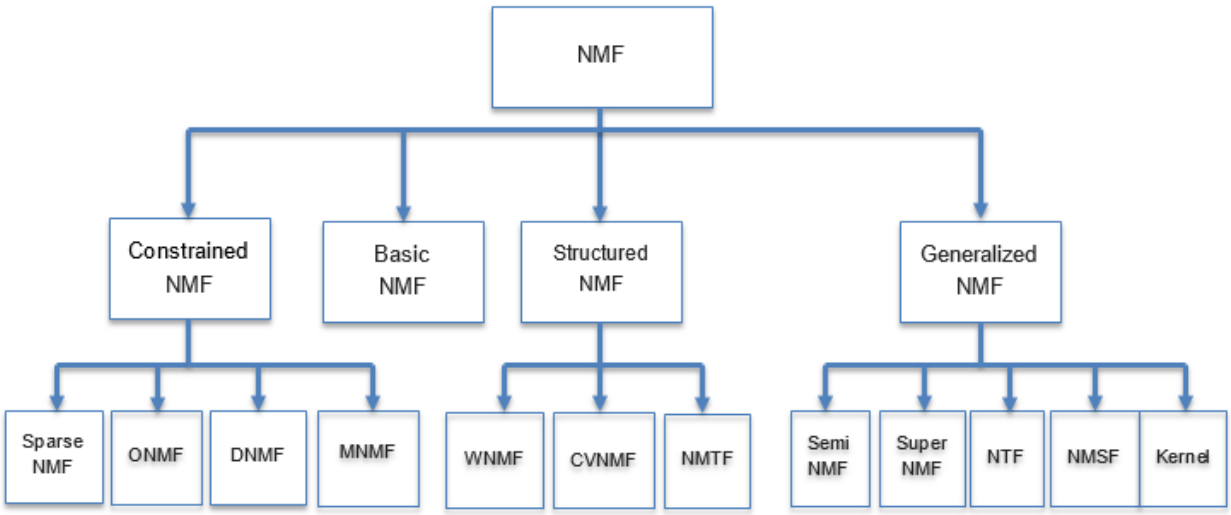

Figure 22. Different NMF models, proposed in [21].

4.1.4.1 Basic NMF (unsupervised NMF)

These systems are those in which the bases dictionary W and the temporal activations H, are unknown, therefore previously calculated information is not used like in the semi-supervised case or supervised, as will be seen later. Unsupervised NMF is normally used in the training stages of the system [22], in order to calculate the bases dictionary training W_{source} from a source. For example considering the input a training spectrogram matrix V_{Train}, is divided into the bases dictionary training W_{source} and its temporal activations H_{source}:

$$V_{Train} = W_{source}\, H_{source} \tag{20}$$

The cost functions for unsupervised NMF are the equations, (14) using the Euclidean distance and (17) using the Kullback-leibler divergence, taking into account that for this case: $V = V_{Train}$, $W = W_{source}$ and $H = H_{source}$.

4.1.4.2 Generalized NMF

Generalized NMF are extensions of NMF like in the case of structured NMF, for example changes the data type or the factorization pattern.

Semi-supervised: Taking into account that the input spectrogram V is composed of a sound source and noisy source, the input spectrogram can be divided into:

$$V \approx \mathrm{WH} = [W_{source}\, W_{noise}] \begin{bmatrix} H_{source} \\ H_{noise} \end{bmatrix} \tag{21}$$

These systems [23] are those where is known just one of the dictionaries that make up the signal, W_{source}, in order to obtain this dictionary is performed a previous training stage just for the sound source.

The cost functions for unsupervised NMF are the equatiions, (14) using the Euclidean distance and (17) using the Kullback-Leibler divergence, but in this case $W = [W_{source}\, W_{noise}]$ and $H = \begin{bmatrix} H_{source} \\ H_{noise} \end{bmatrix}$. This dictionary from the sound source W_{source} will remain constant when the variables obtained minimizing the cost function, are updated:

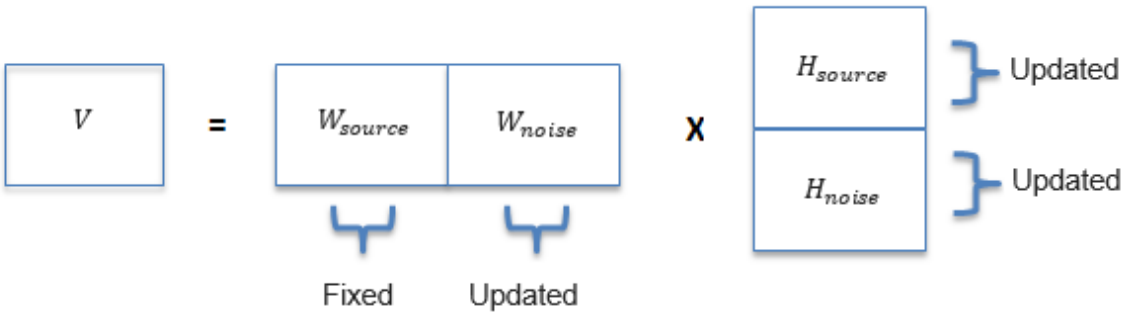

Figure 23. Semi-supervised NMF scheme.

Supervised: unlike the semi-supervised case, considering that the input spectrogram V is composed of a sound source and noisy source, these systems [24] are those where is performed a training stage for the source sound source and another for the noise that make up the signal, obtaining an event dictionary W_{source} and a noise dictionary W_{noise}.
The cost functions for supervised NMF are the equatiions, (14) using the Euclidean distance and (17) using the Kullback-Leibler divergence, but in this case $W = [W_{source}\ W_{noise}]$ and $H = \begin{bmatrix} H_{source} \\ H_{noise} \end{bmatrix}$. This dictionaries from the sound source W_{source} and from the noise source W_{noise} will remain constant when the variables obtained minimizing the cost function, are updated:

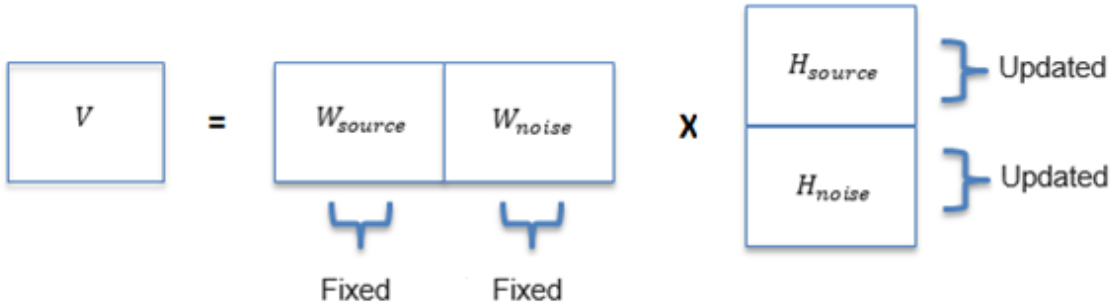

Figure 24.Supervised NMF scheme.

Non-negative Tensor Factorization (NTF): this model [25] generalizes the matrix-form data to higher dimensional data. Is useful, for example in multi-way data analysis, where the data are tensors of second or higher order, being a tensor an algebraic object that generalize scalars, vector and matrices to higher dimensions. For a three dimensional tensor X of size (l_1, l_2, l_3) the factorizations decomposed into three matrix A, B, C and small core tensor D.

The following image, figure 27 shows the decomposition by NTF:

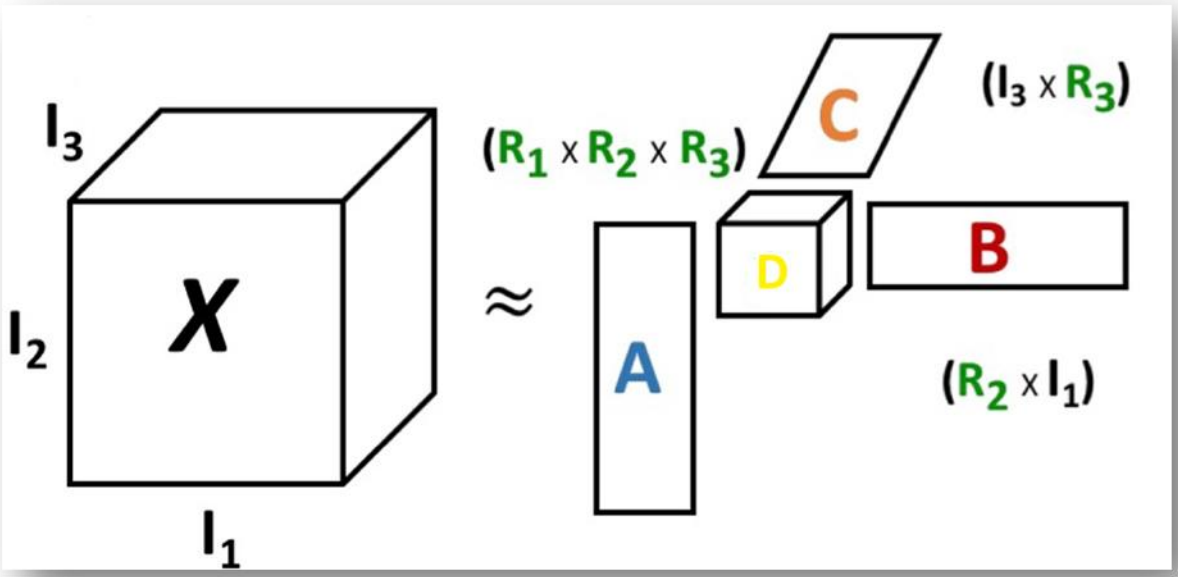

Figure 25. Example descompostion by NTF

The cost function for NTF using the Euclidean distance:

$$D_{NMTF}(X|DABC) = \left\|X_{n,m,l} - \sum_{r_1=1}^{R_1}\sum_{r_2=1}^{R_2}\sum_{r_3}^{R_3} D_{r1,r2,r3}\ A_{r1,n}B_{r_2,m}C_{r3,l}\right\|_F^2 \quad subject\ to\ X, D, A, B, C \geq 0 \tag{22}$$

Non-negative Matrix-set Factorization (NMSF): extends de data sets from matrices to matrix-sets [26]. With this assumption, the learning will be easier, due to the implementation is directly on the matrix set, is not necessary perform a vectorising in the matrix. This model is suitable for images. Considering an input matrix of N observations V^k , NMSF performs a decomposition on, W bases dictionary and a set of temporal activations H^k.

Cost function for NMSF using Euclidean distance:

$$D_{NMSF}(V^k|WH^k) = \sum_{k=1}^{N}\left\|V^k - WH^k\right\|_F^2 \quad \text{subject to } V^kW, H^k \geq 0 \quad \text{k=1,2…N} \tag{23}$$

Kernel NMF (KNMF): apply methods based on kernel by mapping into a feature space through non-linear models [27]. This model can solve some problems that are had with the non-negative constrains in the primitive model of NMF due to in many situations in the real life cannot be satisfied. The objective of KNMF is to represent the input mapped matrix $\phi(V)$ in Hilbert space, using the mapped pre-images $\phi(W)$, and the features matrix H.

The cost function for KNMF using the Euclidean distance:

$$D_{KNMF}(\phi(V)|\phi(W)H) = \|\phi(V) - \phi(W)H\|_F^2 \quad \text{subject to } V, W, H \geq 0 \tag{24}$$

4.1.4.3 Constrained NMF

Sparse NMF: the concept of sparse coding [28], is referred to the representation of the data using only significantly non-zero values being the rest of the values close to zero, achieving a reduction in the dimensions.

There are some types of types of sparse, using different norms but the more common is using the norm one L_1:

In first place is considered a noisy input spectrogram $V_{noisy} = WH + E$, taking into account that WH represent the clean data and E the part to related to the noise, that is sparse, in other words only a small part of entries of E are non-zero, due to norm one L_1 is applied, $\|E\|_1$. For example in a facial recognition system, the glasses of a person can represent this type of noise.

Norm 1 is defined as:

$$\|E\|_1 = max \sum_{i=1}^{n} |E_1| \tag{25}$$

The cost function using the Kullback-Leibler divergence can be formulated as :

$$D_{Sparse}(V|WH+E) = V\ \log\left(\frac{V}{WH}\right) - V + (WH) + \mathrm{E} + \lambda\ \|E\|_1$$
$$subject\ to\ V, W, H, \lambda, E \geq 0 \qquad (26)$$

The cost function using the Euclidean distance can be formulated as:

$$D_{EUC}(V|WH) = \|V - WH - E\|_F^2 + \lambda\ \|E\|_1 \qquad \text{subject to } V, W, H, \lambda, E \geq 0 \qquad (27)$$

Orthogonal NMF: the target of this model [29] is to add orthogonality constraints to W or H, one of them has orthonormal columns. It is useful in order to obtain better results for the clustering problems.

The image below, figure 23, shows a comparative between standard NMF and ONMF in a clustering problem, which has three clusters. ONMF obtain a better solution finding the three clusters.

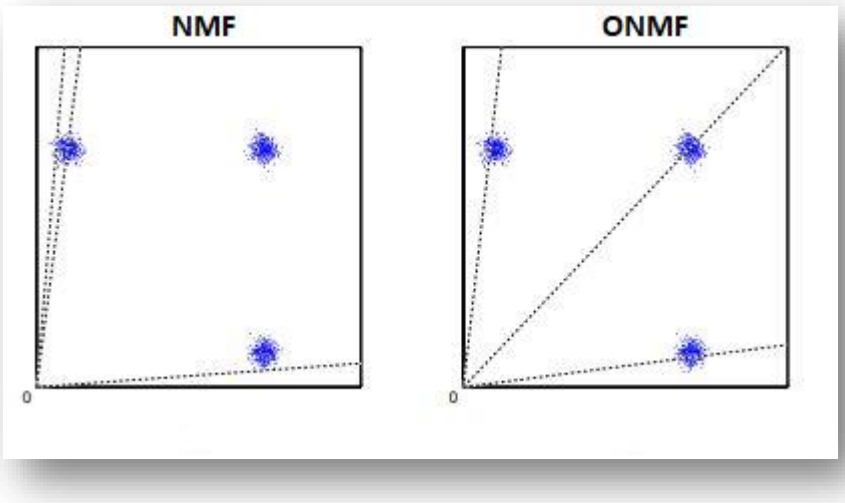

Figure 26. Comparative between NMF and ONMF for a clustering problem, proposed in [23].

Below, equation (28), is shown the Cost function for ONMF, using Euclidean distance, taking into account that Ω is the Lagrangian multiplier matrix, I identity matrix, and Tr the sum of the diagonal elements of the matrix:

$$D_{ONMF}(V|WH) = \|V - WH\|_F^2 + Tr\,[\,\Omega\,(W^T W - I)] \qquad (28)$$
$$\text{subject to } W^T W = I \qquad V, W, H, \Omega \geq 0$$

Discriminant NMF: is the extension of the standard model in order to extract characteristics that enforce the spatial locality, and also the separability between in a discriminant way. This method is very common in genetic , for example [30], considering an input V that is composed by two types of samples *A* and B, each one composed of four replicates. Then is calculated the matrix W and H, the rows of H are the metagenes (up and down) and the column of W represent the weight of each gene in the metagenes. Taking into account the relation between W and H is possible to identify which metagenes are combined with the gene sets.

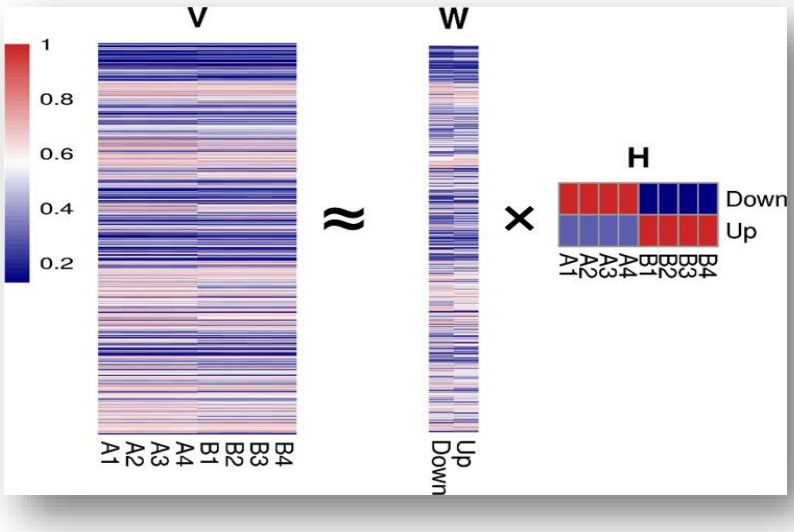

Figure 27. Example of application in a genetic field of Discriminant NMF, proposed in [24].

The cost functions for discriminant NMF are the equatiions, (14) using the Euclidean distance and (17) using the Kullback-Leibler divergence.

NMF on manifold: in some applications like bioinformatic [31], the data are sampled from a nonlinear low-dimensional submanifold inserted in a high dimensional space, for this reason are used different manifold learning algorithms that are able to identify the geometrical structure, combining it with the standard NMF.
One of the most known algorithms is GRNMF [31], which model a geometric structure through the nearest neighbour graph on a scatter of data points.

The optimization problem using the Euclidean distance can be formulated as:

$$D_{GRNMF}(V|WH) = \|V - WH\|_F^2 + \lambda_l (\|W\|^2 + \|H\|^2) + \lambda_m \sum_{i,p=1}^{n} \|w_i - w_p\|^2 S_{i,p}^{m} + \lambda_d \sum_{j,q=1}^{m} \|h_j - h_p\|^2 S_{j,q}^{d} \quad (29)$$
$$subject\ to\ V, W, H, \lambda, S \geq 0$$

Taking into account that λ_l, λ_m, λ_d are the regularization matrix, w_i and h_j ,are the rows of W, H and finally $S_{i,p}^{m}$, $S_{j,q}^{d}$ are sparse weight matrices.

4.1.4.4 Structured NMF

Structured NMF enhances other characteristics in the solution to NMF learning problem. Usually this type of NMF introduce some constraint in order to modify the regular factorization.

Weighted NMF (WNMF): this model uses different weights in order to emphasize the importance of different components, introducing a weight matrix denominated G to the standard NMF [32]. The two most common types of weighting are: frequency weighting, in which the weights are associated with the importance of the different frequency subbands of the

input spectrogram V and temporal weighting, whose weights determine the importance of each of the time frames of the input spectrogram V.

The WNMF model is:

$$G \circ V \approx G \circ (WH) \tag{30}$$

In which $\circ$ is referred to the element-wise multiplication

And the cost function for WNMF using Kullback-Leibler divergence is:

$$D_{weigthed}\,(V \mid WH;\, G) = \sum_{f,t} G\,(f,t)\, d(V(f,t)|[WH]_{f,t}) \quad \text{subject to } V, W, G, H \geq 0 \tag{31}$$

Convolutive NMF (CVNMF): CVNM [33] performs a factorization based on time-frequency. So that, it is needs take into account how the spectrogram varies with the time. Convolutive NMF considers that the input spectrogram V can be decomposed as a sequence of successive of dictionaries of spectral bases W_t and its corresponding temporal activations H.

$$V \approx \sum_{t=0}^{T-1} W_t \overrightarrow{H^t} \tag{32}$$

Cost function for CVNMF using Kullback-Leibler divergence is:

$$D_{KL}(V/WH) = V\ log(\frac{V}{WH}) - V + (WH) + \sum_{t=0}^{T-1} W_t \overrightarrow{H^t}\ \ subject\ to\ V, W, H \geq 0 \tag{33}$$

Taking into account that $\overrightarrow{H^t}$ denotes a column shift operator the moves its argument t places to the right and T length of each spectrum sequence.

Non-negative Matrix Tri-factorization (NMTF): NMTF [34] decomposes the input spectrogram matrix V into three matrix, $W, U, and\ H$, bases dictionary, encode cluster matrix and temporal activations respectively, extending the conventional NMF, which decomposes just in two matrix. So this third factor, provides new features, it is very useful in clustering, for example to co-cluster words and documents.

The NMTF model is:

$$V \approx WUH \tag{34}$$

Cost function for NMTF using Kullback-Leibler divergence is:

$$D_{NMTF}(V|WUH) = \|V - WUH\|_F^{\ 2}$$
$$\text{subject to } V, W, U, H \geq 0,\ HH^T = 1,\ WW^T = 1 \tag{35}$$

4.1.5 Convolutional Neural Networks (CNNs)

In recent year, the detection of sound events using convolutional networks is being researched due to this kind of networks have a high performance a low cost computational, for example in image recognition systems [35].
Convolutional Neural Network [36] receive an input which is processed through a series of hidden layers. This layers are composed by neurons, where each neuron is connected to previous neurons that belong to previous layers and also can shared connection with neurons from posterior layers.

In the following image, figure 28, is shown how the neurons are arranged in layers with three dimensions.

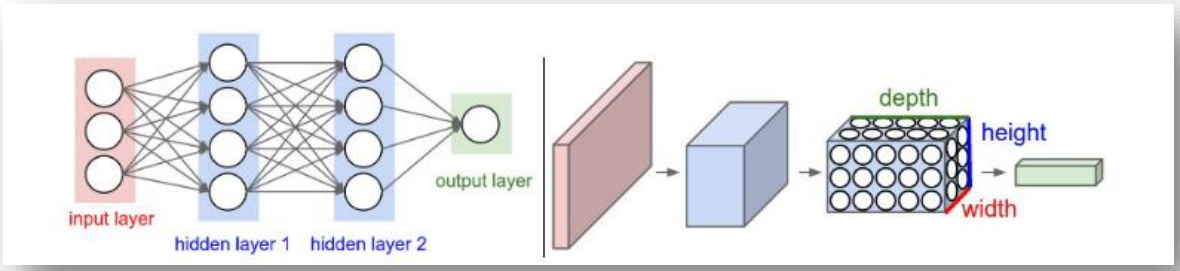

Figure 28. Layers and neurons from a Neural Network, proposed in [36].

The most common layers in a CNN architecture are, convolutional, pooling and fully-connected (FC layers).

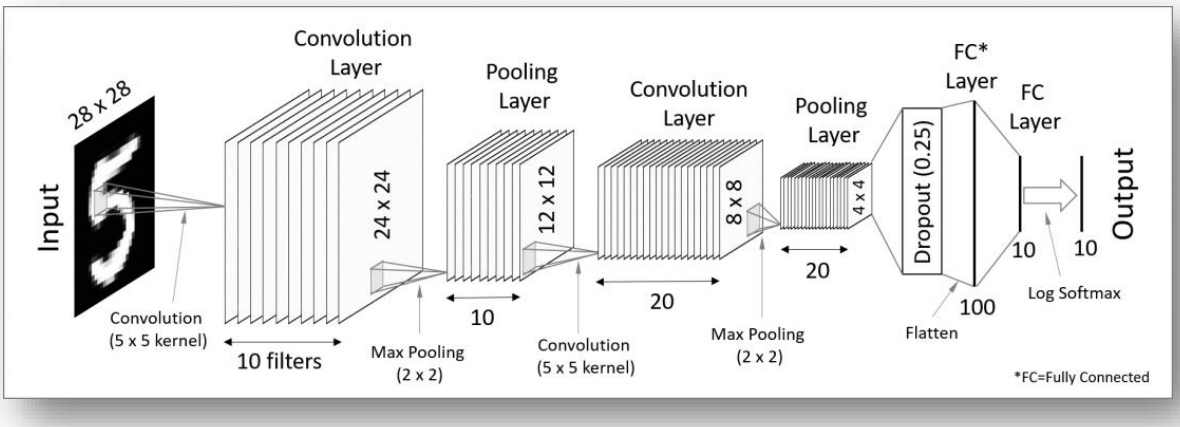

Figure 23. Example Convolutional Neural Network, proposed in [37].

The convolutional layer performs the extracting of the information from the input, and it is composed by a lot of convolution kernels, which are filters, that learn from the input, and generate feature maps, obtaining output neurons, which are calculated, performing the convolution of the input with a learned kernel. The kernels from the lower layers discover low level features, like curves, lines and the upper layers more complex features.

Convolutional layer

The most important advantage of using convolutional layers is reduce the number of the parameters and reduce the cost computational.

Considering a 6x6 matrix with a 3x3 filter is obtained a 4x4 matrix:

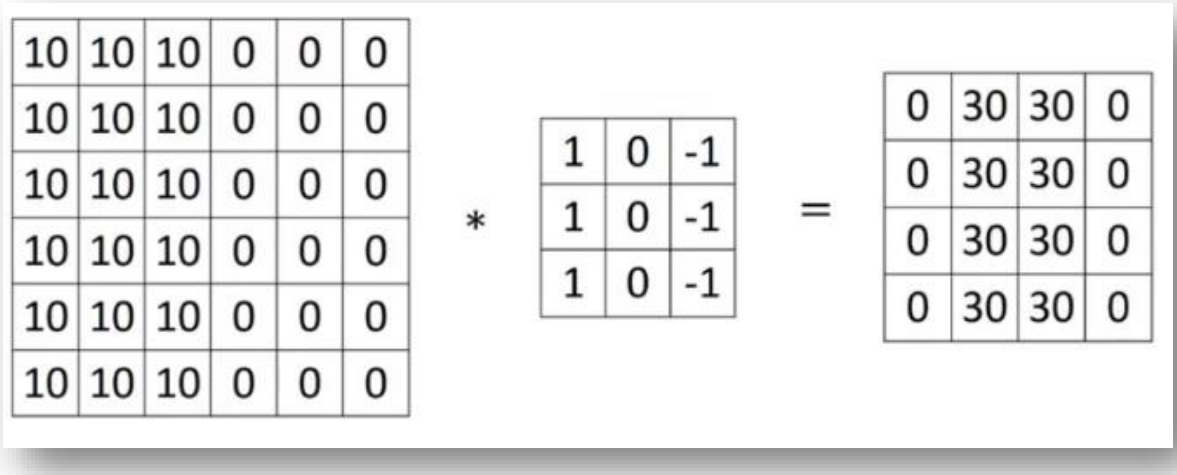

Figure 24. Example convolution operation from a convolutional layer, proposed in [35].

Therefore of general way, the dimensions can be expressed like:

- Input : N x N
- Filter size : F x F
- Output: (N – F +1) x (N – F +1)

For example, considering two fully connected layers, a comparison is made between using and not using a convolutional layer, using three channels:

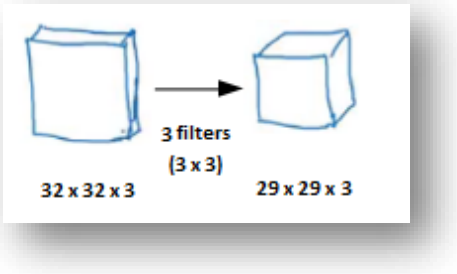

Figure 25. Two fully connected layers using three channels, proposed in [35].

- In the case of not using a convolutional layer the parameters that would be necessary, are: 32*32*3*29*29*3, more or less around 7.7 million parameters.

- In the case of using a convolutional layer the parameter that would be necessary are: (3 x 3)* 3 = 27 parameters.

As can be seen the reduction of parameters is important.

Later the **pooling layer,** which is usually between two convolutional layers, has the function of minimising the feature maps and the speed up computation.

Example using a max pooling with filter size (2, 2):

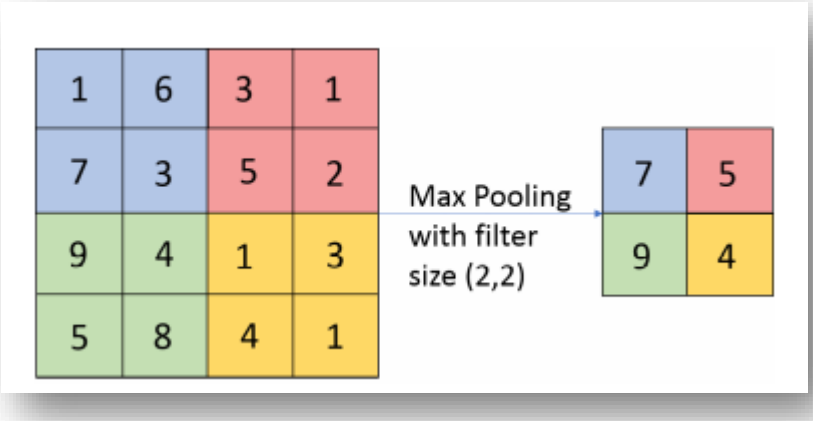

Figure 26. Max pooling with filter size (2,2), proposed in [38].

At the end **fully connected layers** takes the output from previous layer and turns them into a single vector and then will compute the classes score.

5. DEVELOPMENT AND IMPLEMENTATION

The development of this project is based on the following paper [39], in which the following system is proposed:

- In this section in order to abbreviate is considered that: $W_{source} = W_s$, $H_{source} = H_s$, and $W_{noise} = W_n$, $H_{noise} = H_n$.

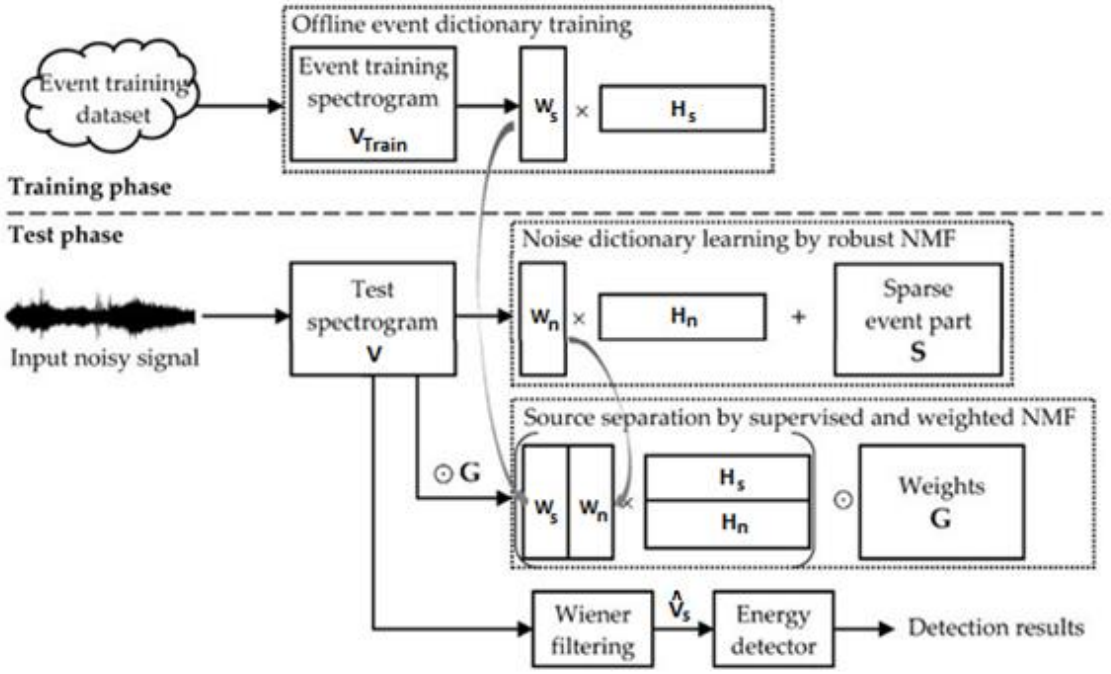

Figure 27. System implemented based on [39].

The main objective of the method is to detect different types of sound events, baby cry, gunshot or glass break events from noisy recordings, to that end, the system will use a source separation by supervised and weighted NMF, so the system needs the spectral bases dictionary from the events W_s, and also an estimated noise dictionary W_n from the input spectrogram V .

The system is divided into two phases: training phase and test phase. In the training phase the objective is to obtain the spectral bases dictionary W_s using events signals without noise concatenated into an array V_{Train}, this dictionary W_s obtained from the events signals, is saved in order to use later in the test phase. In the test phase, in first place is estimated the noise dictionary W_n using a robust NMF from the noisy input V, and then is performed the source separation by supervised, using the spectral bases dictionary from the clean event

signals W_s previously calculated in the training phase and the estimated noise dictionary W_n, in addition will be used a weight matrix G in order to enhance some areas or zones, that could correspond to the sound events, when performing the source separation. At the end is obtained the output spectrogram V_S obtained from the source separation, which is smoothed $\widehat{V_S}$ using a wiener filter, and later is applied an energy detector in order to detect high energies that could correspond with an event.

5.1 Pre-processing

It is important to carry out a pre-processing stage with the aim that all audios used in the system are in the same "conditions" in order to perform at the end the energy detector, which must be universal for each class of events, for this reason is important to scale the amplitude values, also use the same sampling rate F_S, from the inputs etc.

- All signals, are resampled at the same sample rate F_S 44100 Hz, value obtained in [3].
- All stereo audios are converted to single channel.
- All recordings per each kind of event are normalized between 0 and 1 dividing by the maximum of the absolute value of the signal.

5.2 Training

This stage is performed only one time per each one of the event classes that in this system are: baby cry, glass break and gunshot. In this stage are used signals of isolated events without noise for each event class and are concatenated to a data matrix, having three different matrix and then is applied the STFT for each matrix obtaining, three training spectrograms V_{Train}, one for baby cry events other for gunshots and the last for glass break events.. The STFT have N = 2048 points with a frame length of 46 ms and 50% overlap.

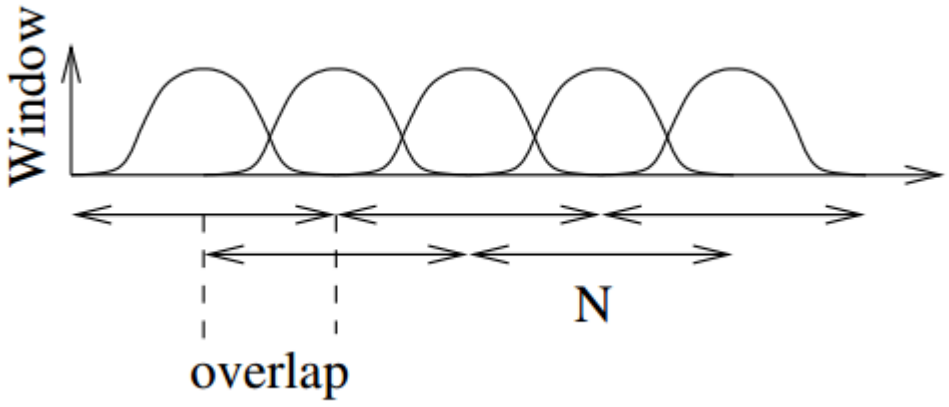

Figure 28. STFT with overlap

FFT considers that the entry data belongs to a finite set of data, assuming that the enter signals are periodic. Therefore FFT considers, that the ends are connected to each other like a circular topology. When the input signal is periodic and contains an integer number of periods, FFT achieves a god result, but in most cases, the input signals are not periodic or not contain an integer number of periods, causing a series of discontinuities, which produce energy leaks at other frequencies, this is called spectral leakage. Applying a window of length N, two main effects are obtained: reduction of spectral resolution and the secondary lobes level, spectral leakage.

The selected window $W[n]$ is the hamming, because it is one of the windows that works best for random signals. It is defined as:

$$W[n] = \begin{cases} 0.54 - 0.46 * cos(\frac{2*\pi*n}{M}) & 0 \leq n \leq M \\ 0 \ \ in\ other\ cases \end{cases} \quad (36)$$

Below in figure [34], is shown an example of a hamming window $W[n]$ composed of N= 21 samples.

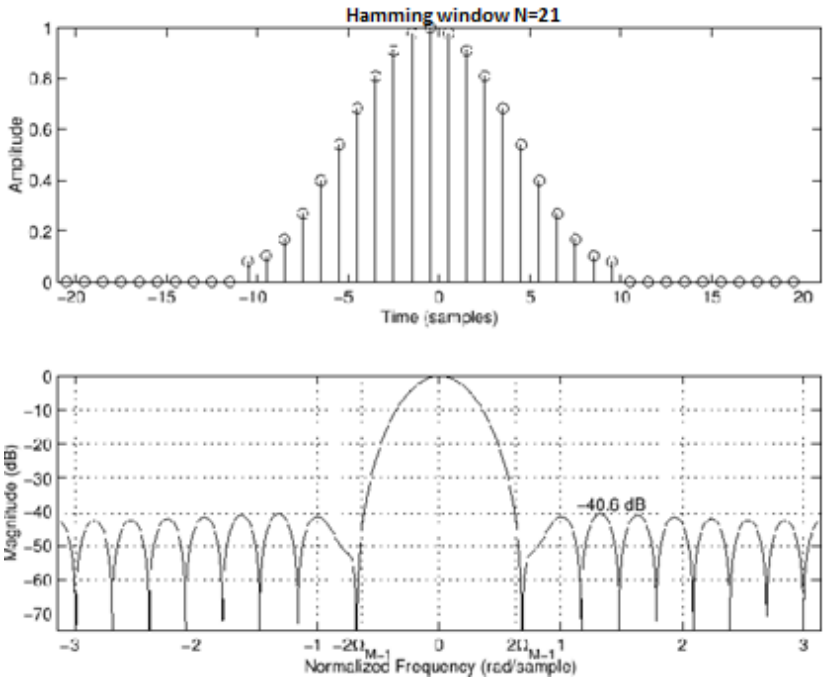

Figure 29. Example hamming window with N=21

The spectral resolution $\Delta f(Hz)$ is influence by the width of the main lobe $W_{LM}\ (Hz)$, which is defined as the bandwidth of the Fourier transform of the window between the two nulls closest to the frequency origin. For the hamming window $W[n]$ the width of the main lobe as $W_{LM}\ (Hz)$ = 4 / N, therefore the spectral resolution is:

$$\Delta f(Hz) = \frac{4}{N} \tag{37}$$

So in this case using N=21 the spectral resolution Δf is 0.19 Hz. In the case of use N=2048 like in this project the spectral resolution Δf is 1,95 mHz .

Regarding secondary lobes level L_{SL}, is defined as the difference between the local maximum of the Fourier transform at the origin and the next largest, equation (38).

$$L_{SL}(dB) = 20 * \log_{10}(A_0 / A_1) \tag{38}$$

So in this case the secondary lobes level L_{SL} is 40.6 dB.

As previously explained, all the signals belonging to the same class event are concatenated in a matrix and the STFT is applied obtaining V_{Train}, so there will be three training matrices, one for baby cry, figures 35 and 36, another for gunshot, figures 37 and 38 and for glass break, figures 39 and 40.

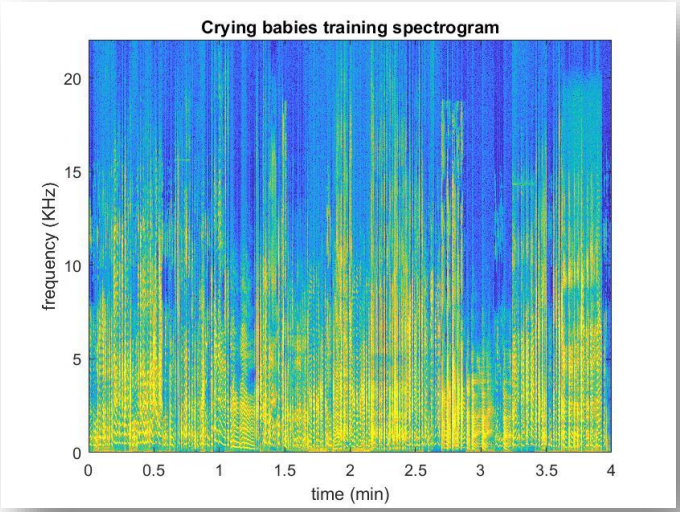

Figure 30. V_{Train} for baby cry class.

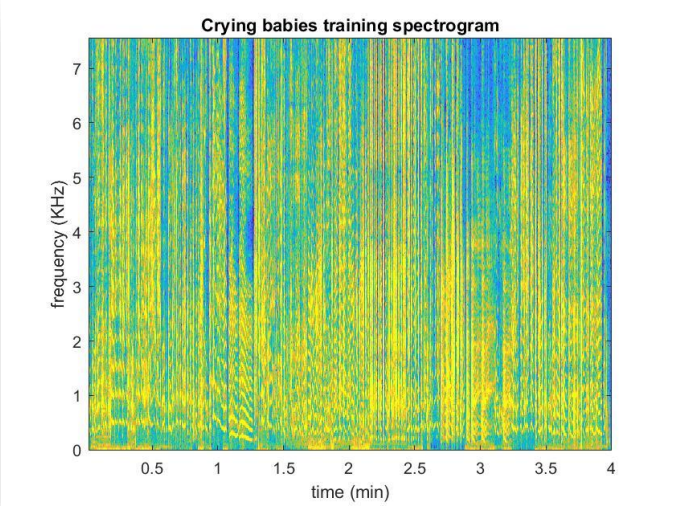

Figure 31. V_{Train} for baby cry class using zoom.

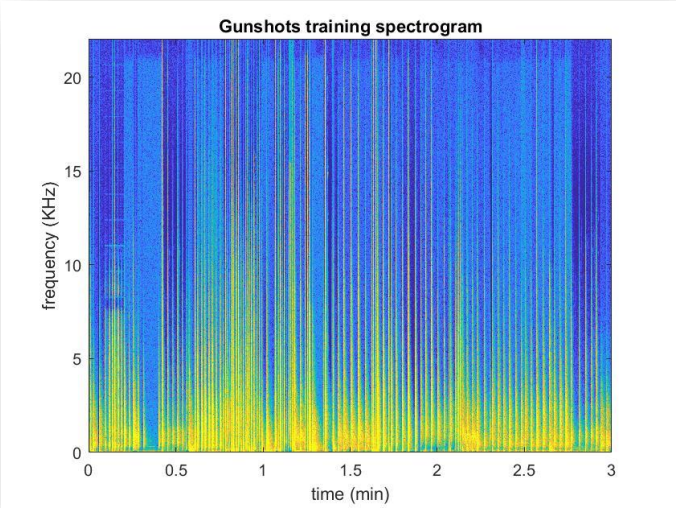

Figure 32. V_{Train} for gunshot class.

Figure 33. V_{Train} for gunshot class using zoom.

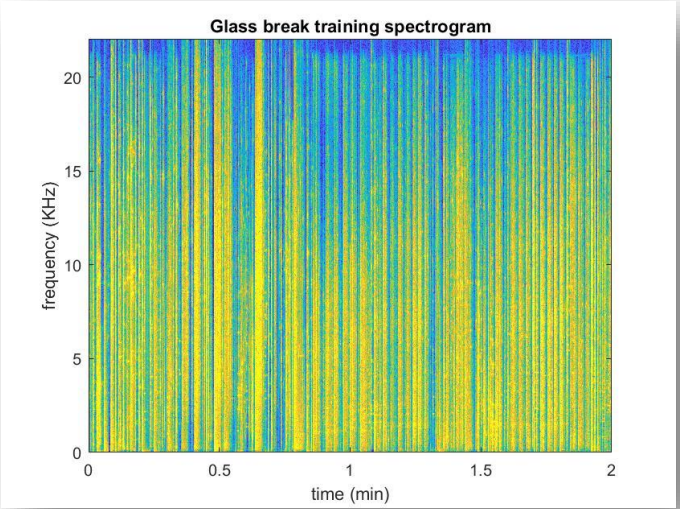

Figure 34. V_{Train} for glass break class.

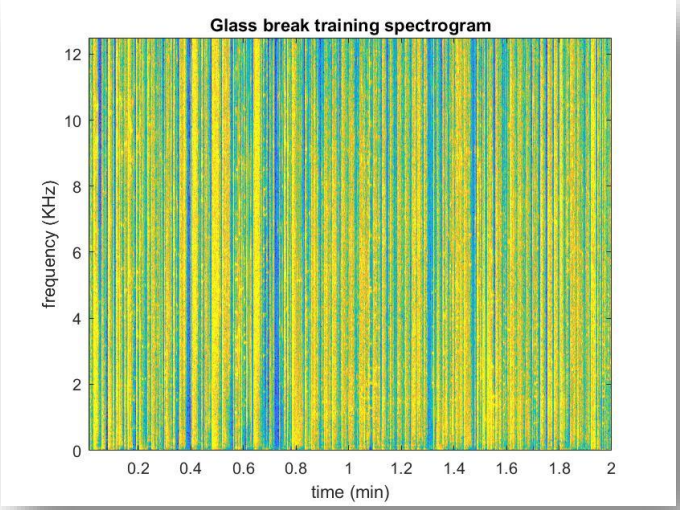

Figure 35. V_{Train} for glass break class using zoom.

When is obtained the spectrogram of the concatenated event signals V_{Train} , is applied the unsupervised model of NMF. The cost function as explained before is the same as the equation (17) using the Kullback-Leibler divergence, taking into account that for this case: $V = V_{Train}$, $W = W_s$ and $H = H_s$, so the equation is:

$$D_{KL}(V_{Train}|W_s\,H_s) = V_{Train}\ \log\left(\frac{V_{Train}}{W_s\,H_s}\right) - V_{Train} + (W_s\,H_s\,) \tag{39}$$

$$subject\ to\ \ V_{Train}, W_s\,, H_s \geq\ 0$$

The following equations are updated n times, until the curve of distance converges, equations (12) and (13):

$$W_s\ =\ W_s\ \circ\ \left(\frac{V_{Train}}{W_s\,H_s}\ {H_s}^T\right)\ /\ \left(1\ {H_s}^T\right) \tag{40}$$

$$H_s = H_s \circ \left(W_s^T \frac{V_{Train}}{W_s H_s} \right) / \left(W_s^T 1 \right) \quad (41)$$

In which ○ and / is referred to the element-wise multiplication and division of two matrices and the subscript T indicates the transposition of a matrix. The dimensions for each matrix are: V_{Train} $(F\ x\ T)$, W_s $(F\ x\ K)$, H_s $(K\ x\ T)$ and 1 is a matrix of dimension $(F\ x\ T)$.

As you can see in the next graph, is enough 200 iterations or updates of the previous parameters so that the cost function of the equation (39) converges:

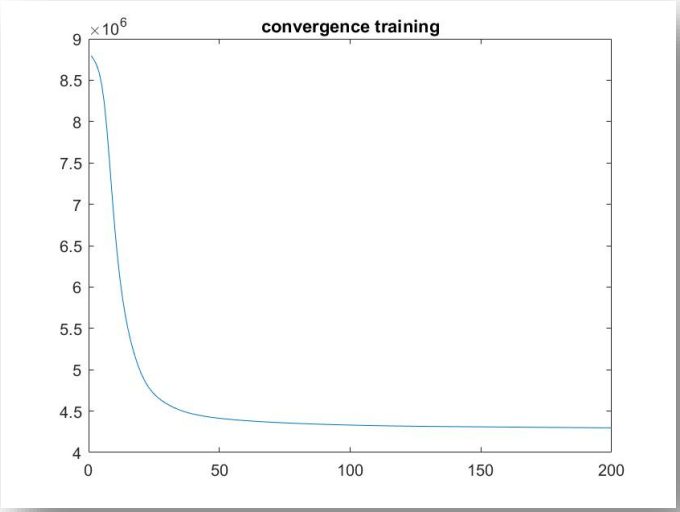

Figure 36. Convergence NMF in the training phase

5.3 Noise dictionary learning by robust NMF

The main objective of applying this algorithm [40] is to estimate the noise contained in the input spectrogram V, considering que is composed of two parts, one part related to the noise and the other part related to the event. The part related to the noise L_n, which is usually dense and stable, can be modelled by multiplying the estimated noise dictionary W_n and its temporal activations H_n, however the part related to the event happen infrequently over the time, so that is expressed by the sparsity parameter S, that only have a few non-zero entries.

$$V \approx L_n + S \approx W_n H_n + S \tag{42}$$

A misconception would be to think, that sparse can be calculated as the subtraction between the input spectrogram V and the part related to the noise L_n, however the subtraction can give negative values into the estimated event spectrogram. So, to solve the problem correctly, we have to solve the following problem in order to minimize the error:

$$\begin{gathered} min_{W_n,H_n,S}\, D(V \mid W_n H_n + S) + \lambda \|S\|_1 \\ s.t.\ W, H, S, \lambda \geq 0 \end{gathered} \tag{43}$$

The parameter λ control the weight of the sparsity parameter S, which is measured by norm one L_1.

Minimizing the cost function solved in [12], is obtained:

$$W_n = W_n \circ \left(\frac{V}{W_n H_n + S} {H_n}^T \right) / \left(1 {H_n}^T\right) \tag{44}$$

$$H_n = H_n \circ \left(\frac{V}{W_n H_n + S} {H_n}^T \right) / \left({H_n}^T 1\right) \tag{45}$$

$$S = S \circ \left(\frac{V}{W_n H_n + S} \right) / (1 + \lambda) \tag{46}$$

In which $\circ$ and / is referred to the element-wise multiplication and division of two matrices and the subscript T indicates the transposition of a matrix. The dimensions for each matrix are: $V\ (F\ x\ T)$, $W_n\ (F\ x\ K)$, $H_n\ (K\ x\ T)$, $S\ (F\ x\ T)$ 1 is a matrix of dimension $(F\ x\ T)$.

For the above cost function, equation (43), the enough number of interactions n or updates of the previous parameters to converge is 200:

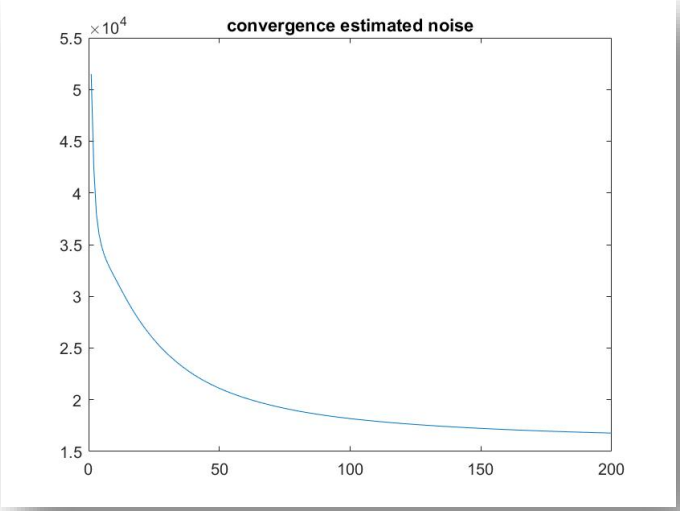

Figure 37. Convergence in noise dictionary learning.

Considering as input V a noisy baby cry spectrogram, in which the baby cry event is located around the second 5.

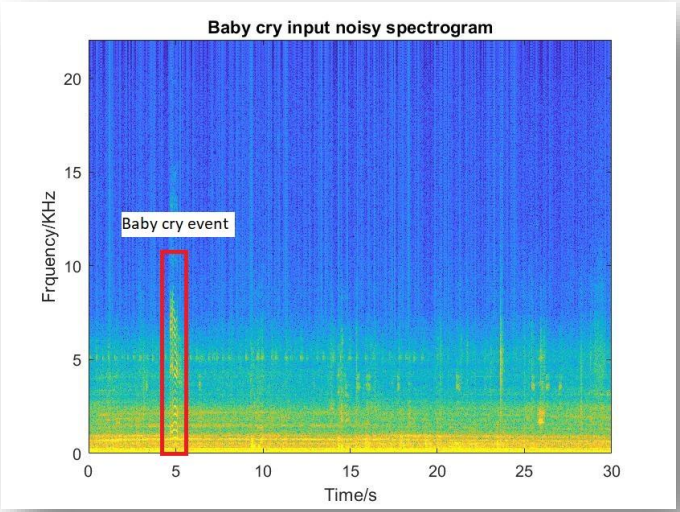

Figure 38. Input noisy spectrogram V, which contains a baby cry event around the second 5.

As previously explained the noise spectrogram L_n can be obtained by multiplying of the noise dictionary W_n and its temporal activations H_n, as the following spectrogram shows, the part related to the event does not appear because it is contained in the sparse parameter S.

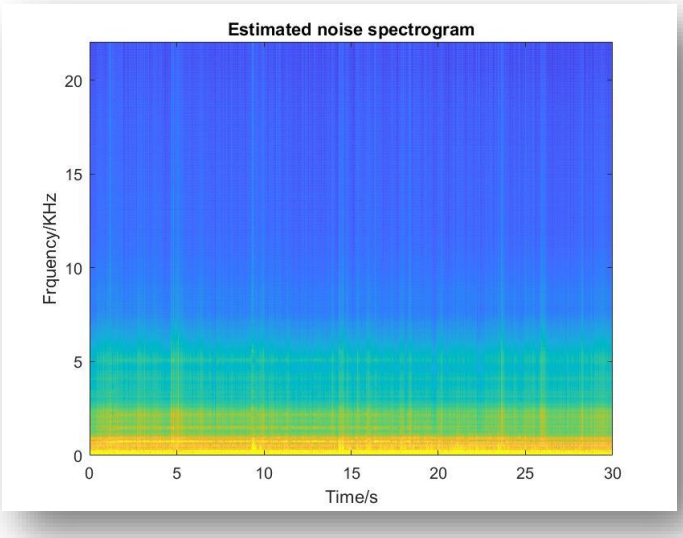

Figure 39. Estimated noise spectrogram L_n from the input spectrogram V of the figure 43.

In the following spectrograms is represent S with different values of λ:

- For $\lambda = 0$ is obtained :

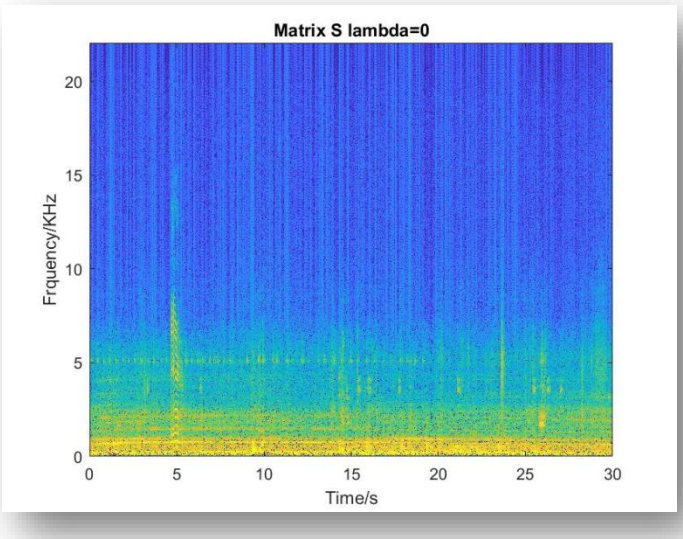

Figure 40. Sparse matrix S for λ=0.

- For $\lambda = 0.9$ is obtained :

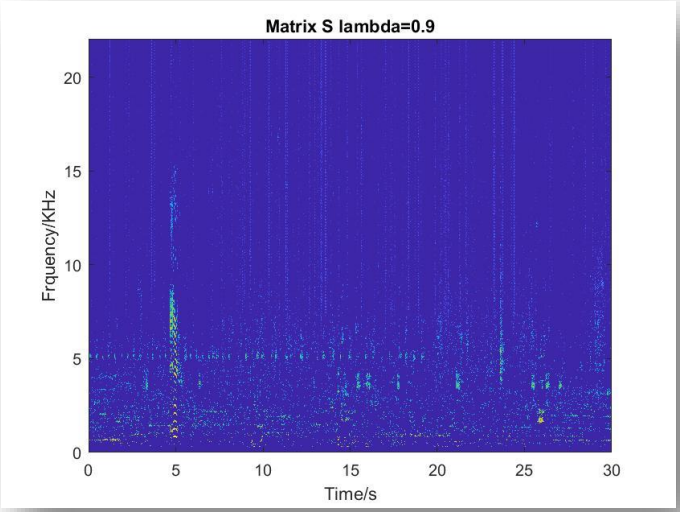

Figure 41. Sparse matrix S for λ=0.9.

As has been shown in figures 45 and 46, a higher value of λ, retain very few bases related to the noise, retaining most of the bases related to the event, when the value of λ, the opposite happens.

5.4 Source separation by supervised and weighted NMF

As previously explained in the state of the art, a source separation by supervised, consists of performing the separation of a sound source and another source that is noise, using the information previously obtained, spectral base dictionaries for the sound source W_s, obtained in the training stage and for the estimated noise W_n, obtained in the before section, noise dictionary learning.

This model [41], modifies the basic model in order to emphasize the importance of the different components of the input spectrogram by introducing the weight matrix G .This weight matrix can be based on two weighting one of them, frequency based and the other temporal based. The first, frequency weighting is based on the importance of the frequency subbands, with the aim of enhancing those frequencies that compose the event target. The

second one, is based on the importance of the different frames, assigning them different probabilities, depending where the event is.

The weighted NMF was previously formulated as, equation (30):

$$G \circ V = G \circ (WH)$$

In the case that G were equal to a matrix composed of 1, would be the basic model.

The cost function for weighted NMF using the Kullback-Leibler divergence is, equation (31):

$$D_{weigthed}\,(V \mid WH;\; G) = \sum_{f,t}\; G\,(f,t)\; d(V(f,t)|[WH]_{f,t})$$

Minimizing the cost function is obtained the following parameters. For the sound source is used the subscript *s* and for the noise source the subscript *n,* remembering that $W = [W_s\; W_n]$ and $H = \begin{bmatrix} H_s \\ H_n \end{bmatrix}$*:*

$$H_C \;=\; H_C \circ \left({W_C}^T \frac{G \circ V}{WH} \right) / \left({W_C}^T G \right) \quad \text{being} \quad c \;=\; s, n; \tag{47}$$

$$W_C \;=\; W_C \circ \left(\frac{G \circ V}{WH}\; {H_C}^T \right) / \left(G\; {H_C}^T \right) \quad \text{being} \quad c \;=\; s, n; \tag{48}$$

In which ○ and / is referred to the element-wise multiplication and division of two matrices and the subscript T indicates the transposition of a matrix. The dimensions for each matrix are: $V\;(F\;x\;T)$, $W\;(F\;x\;K)$, $H\;(K\;x\;T)$ and $G\;(F\;x\;T)$.

For the above cost function, equation (31), the enough number of interactions n or updates of the previous parameters to converge is 200:

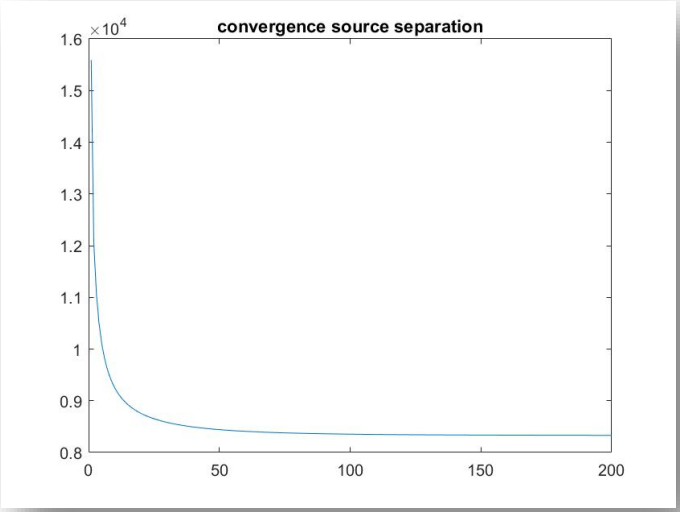

Figure 42. Convergence NMF in the source separation phase

5.4.1 Frequency weighting

As already explained, the frequency weighting [41] quantifies for each frequency band the contribution that it has in the construction of the event, in other words this frequency weighting give high weights to the subbands that are part of an event and small weights to the subbands that are part of noise, obtaining a frequency weight matrix G_{freq}, achieving noise reduction and event amplification. In order to compute the subband weights are used the training events spectrograms that compose the training matrix V_{Train} and also the estimated noise spectrogram L_n.

In first place is computed an average spectral template e_s, using the training matrix formed by the clean event signals V_{Train}.

$$e_s(f) = \frac{1}{T_{train}} \sum_{f} V_{Train}(f,t) \tag{49}$$

In the following step, is computed a smoothed noise spectral template e_n using the estimated noise spectrogram L_n , which is previously computed.This smoothed is done taking into account the neighbouring frames, $\frac{T_0}{2} - 1$ before and $\frac{T_0}{2}$ after each current frame.

$$e_n\,(f,t) = \frac{1}{number\ of\ averaging\ frames} \sum_{j=max\left(1,t-\frac{T_0}{2}\right)}^{min\left(T,t+\frac{T_0}{2}-1\right)} L_n\,(f,j) \tag{50}$$

Finally, in order to compute the frequency weight matrix G_{freq}, is divided the average spectral template e_s by the smoothed noise spectrogram e_n . The average spectral templated is composed by just one column and this column is divide for each frame from the smoothed noise spectrogram, generation a mask for each frame. In summary, this process indicates for each frame the importance of each subband.

$$G_{freq}(f,t) = \frac{e_s\,(f)}{e_n\,(f,t)} \tag{51}$$

However, this G_{freq} is composed of frames with different scales, to solve this problem is performed a normalization, dividing the subband weights of each frame between the maximum value of each one in order to convert the values between 0 and 1.

$$G_{freq}(f,t)/\,max_f\,(G_{freq}\,(f,t)\ ) \tag{52}$$

The weights that have little importance in the construction of the event, which are very small values, are replace by a value very close to 0.

Below is shown an example of frequency weighting for the noisy input spectrogram V of a baby cry event, figure 43, used in the noise dictionary section.

Energy curve, from the input spectrogram V of the figure 43:

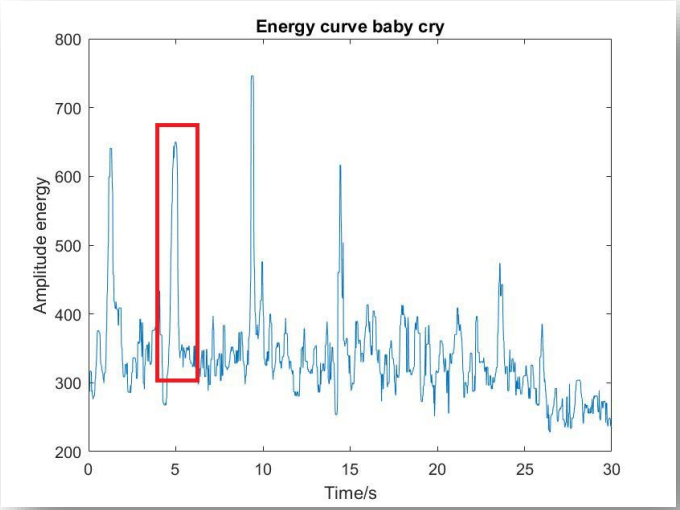

Figure 43. Energy curve from input spectrogram V of the figure 44.

Performing the average is obtained the following spectrogram, figure 49, that corresponds with e_s and also the energy template, figure 50, which represents how much normalized energy there is in all frequencies.

Figure 44. Average spectral template e_s from the training matrix V_{Train}.

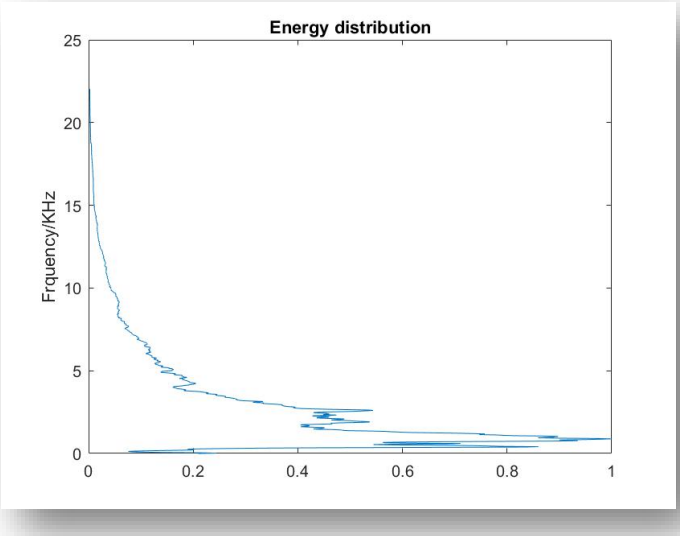

Figure 45. Energy template corresponding to e_s, that represents how much normalized energy there is in all frequencies.

The smoothed noise spectrogram e_n, figure 51 and its energy template, figure 52:

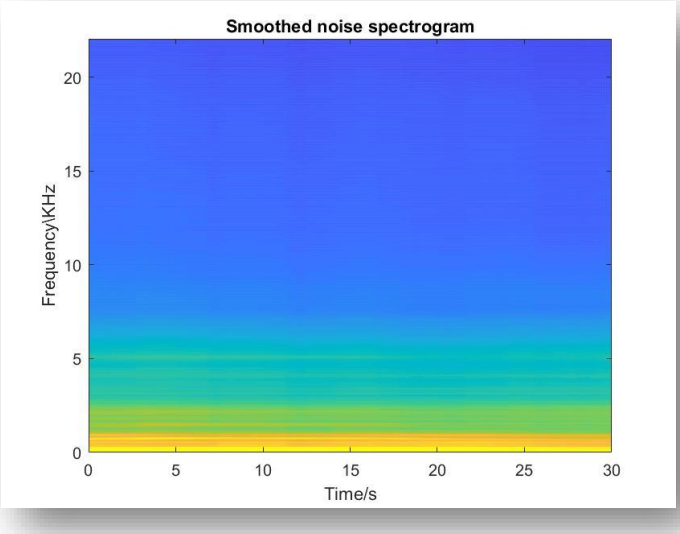

Figure 46. Smoothed noise spectrogram e_n , from L_n, figure 44.

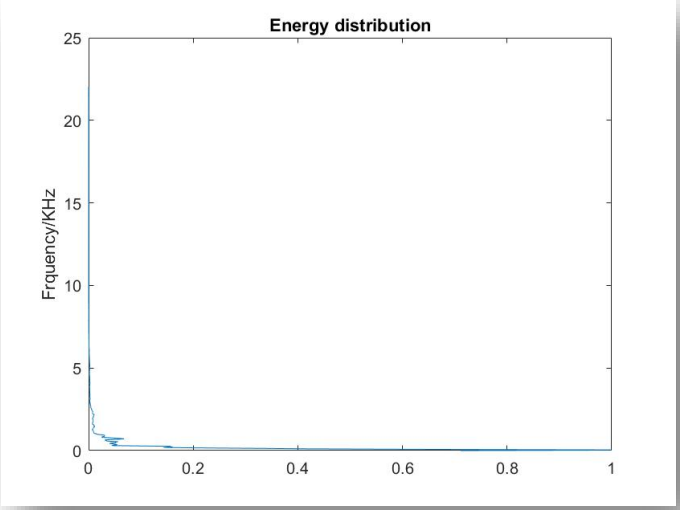

Figure 47. Energy template corresponding to e_n ,that represents how much normalized energy there is in all frequencies.

In this case dividing e_s by e_n, the frequency weight matrix G_{freq} obtained for the input spectrogram V enhances the area between 8 KHz and 10 KHz:

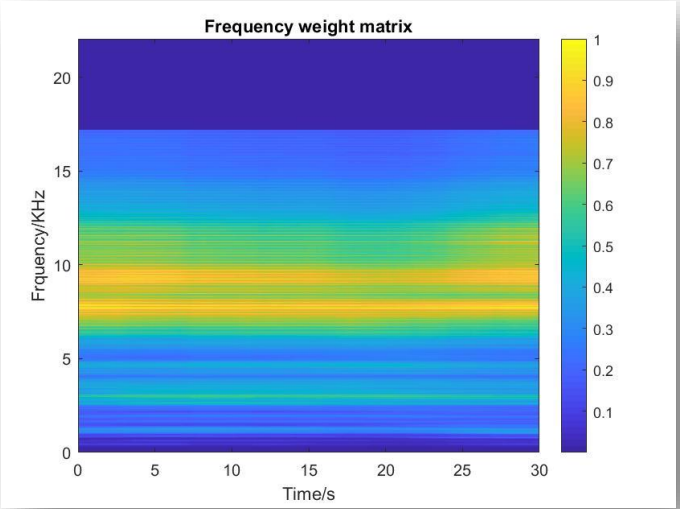

Figure 48. Frequency weights matrix G_{freq} for the input spectrogram V of the figure 43, which enhances the subbands between 8 KHz and 10 KHz. So this subbands have more contributions in constructing the event.

5.4.2 Temporal weighting based on event probability

Temporal weighting [39] is used to assign different probabilities to each frames. High probabilities are assigned to frames where the event is active and low values where the noise is present from the input spectrogram .

In first place, the input spectrogram V, is filtered with the frequency weight matrix G_{freq} calculated in the previous section, obtaining a spectrogram with a lower noise, $V_{filtered}$.

$$V_{filtered}(f,t) = G_{freq}(f,t)\, V(f,t) \tag{53}$$

The filter, that corresponds to the frequency weight matrix G_{freq} , enhances the energy from areas where the event from the input spectrogram V is present and the energy from the noisy areas are reduced. However this effect produced by the filter can be increased making the rest between, the energy for all frequencies of this filtered spectrogram $V_{filtered}$ and the input spectrogram V. It is very important to take into account that the two spectrograms must have the same global energy in order to obtain a correct value. This operation give information about which areas have been increased the energy or attenuated. The areas where the energy has been increased are the areas from the input spectrogram V where the event is active and the areas where the energy has been attenuated correspond to noisy areas.

$$\Delta E(t) = \sum_{f} V_{filtered}(f,t) - \sum_{f} V(f,t) \tag{54}$$

Now, two limits are established, $rmin$ and $rmax$ in order to estimate an event presence probability. When the energy of a frame is higher than $rmax$, this means that there is a high probability that this frame corresponds to an event, for this reason is assigned a very high probability value 0.99.When the energy of a frame is lower than $rmin$, this means the opposite, the probability that this frame corresponds to an event is very low, for this reason is assigned a value of 0.01. If the energy is between these two limits, an intermediate probability value is calculated.

$$G_{temp}(t) = \begin{cases} 0.99 & \Delta E(t) \geq rmax \\ 0.98 * \dfrac{\Delta E(t) - rmin}{rmax - rmin} & rmin < \Delta E(t) < rmax \\ 0.01 & \Delta E(t) \leq rmin \end{cases} \quad (55)$$

Below is shown the temporal weighting using the same input spectrogram V as in the previous section, figure 43:

In first place is filtered the input spectrogram V, using the frequency weight matrix G_{freq}, fugure 53. This filtered spectrogram $V_{filtered}$ is normalized in order to obtain the same overall energy like the input. Below is shown the filtered spectrogram $V_{filtered}$, figure 54 and its energy curve, figure 55.

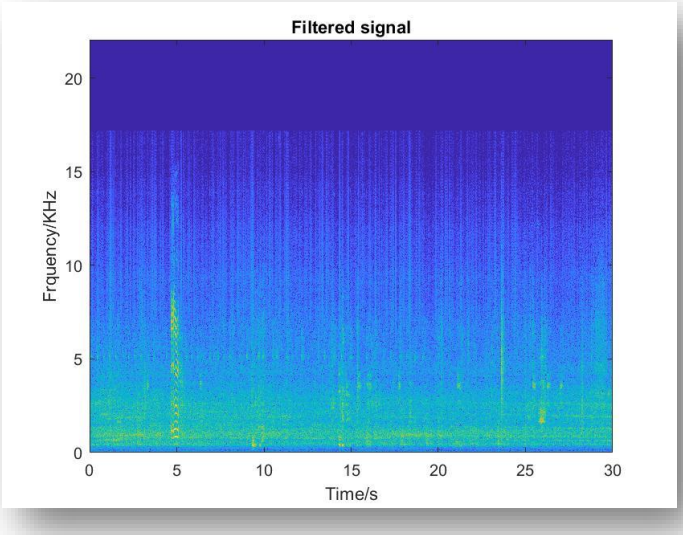

Figure 49.Filtered spectrogram $V_{filtered}$, for the input spectrogram V of the figure 43 using its frequency weight matrix of the figure 53.

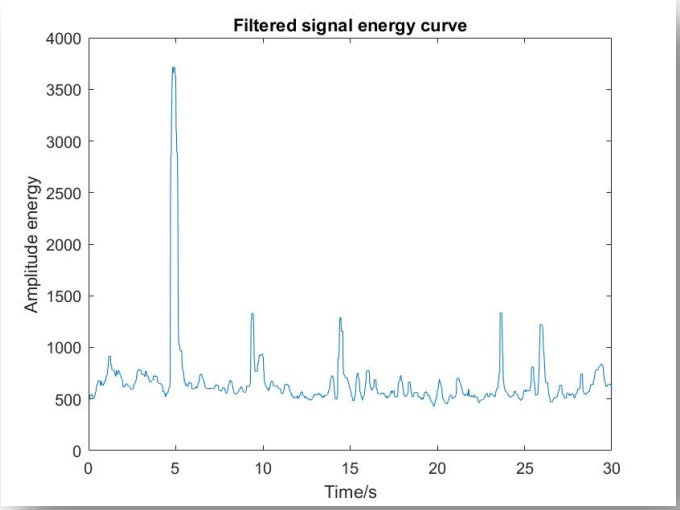

Figure 50. Energy curve from filtered spectrogram $V_{filtered}$, figure 54.

Now is performed the difference between the energy curve from filtered signal $V_{filtered}$ and the energy curve from input signal V. This energy difference is denominated as Ae.

In the figure below 56, are drawn two red lines over the energy difference Ae , the top line is $rmax$ that is set to half the maximum value of this difference of energy and all frames that contain a higher energy value that this line, have a 0.99 of probability. On the contrary the bottom line represent by $rmin$ is set to 0, so all frames that contain a lower energy value that this line, have a probability of 0.01 and the frames with intermediate energy values between these two lines have an intermediate probability value.

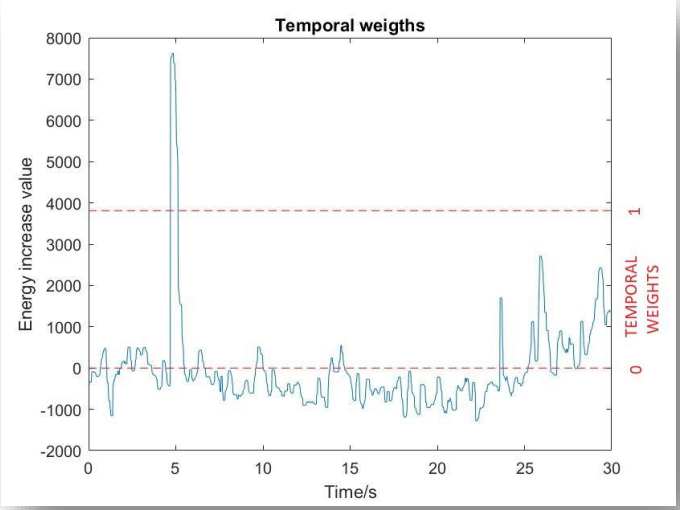

Figure 51. Temporal weights with the limit values rmin and rmax, for the input spectrogram V of the figure 44.

At the end is obtained, the following temporal weights matrix G_{temp}, which indicates that there is an event with much probability near to the second five.

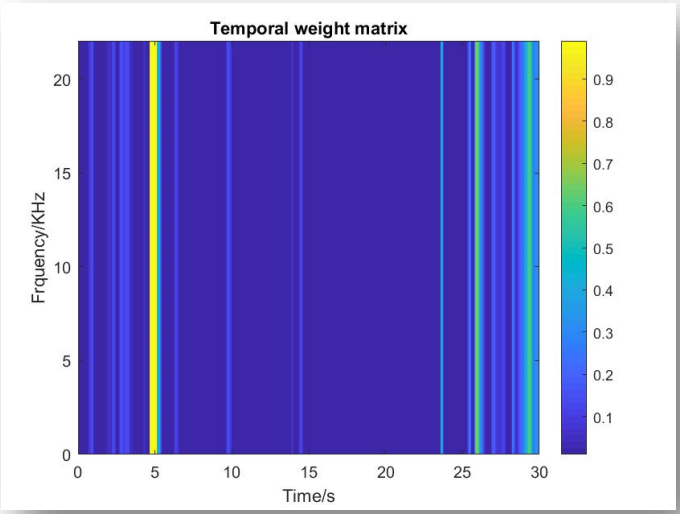

Figure 52. Temporal weights matrix G_{temp} for the input spectrogram V of the figure 43. In this matrix the frames near to the second five have the probaibilities higher associated to an event.

5.4.3 Combined time-frequency weighting

Combining the two types of weighting, the following equation is obtained:

$$G_{freq+temp}(f,t) = G_{freq}(f,t) * G_{temp}(t) \quad (56)$$

Continuing with the previous example for the input spectrogram V, figure 43, multiplying the frecuency weight matrix G_{freq}, figure 53, and the temporal weight matrix G_{temp}, figure 57, the following combined weight matrix $G_{freq+temp}$, figure 58, is obtained:

The enhanced area is in the frames near to the second five and in the subband near to 9 KHz.

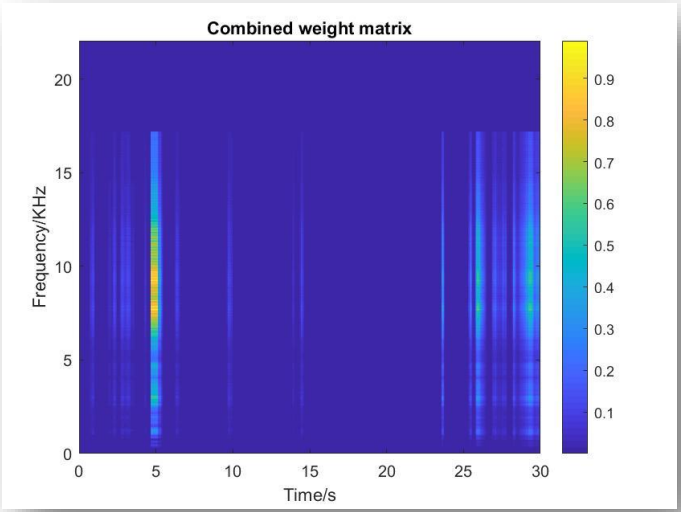

Figure 53. Combined weights matrix $G_{freq+temp}$ for the input spectrogram V of the figure 43, whose enhanced area is in the frames around the second 5 and the subband near to 9 KHz.

5.5 Event detection

When the source separation is performed, the temporal activations matrix are obtained, and therefore the spectrogram associated with the event part can be obtained like, $V_s \approx W_s * H_s$. However, a better result is obtained performing a Wiener-type filtering in order to smooth the output spectrogram, $\widehat{V_S}$.

$$\widehat{V_s} \approx \frac{W_s H_s}{W_n H_n + W_n H_n} \tag{57}$$

Once the output smoothed spectrogram $\widehat{V_S}$ is obtained is applied an energy detector by means of a threshold TH in order to detect frames with high energy, that indicate the presence of an event. In first place the energies per frame e_{frame} of the output spectrogram $\widehat{V_S}$ are obtained, then each energy value per frame e_{frame} is compared to a threshold value TH, if this energy value per frame e_{frame} is greater than the threshold TH in successive frames, this indicates that the frames correspond to an event, therefore all these frames will be equaled to one and the rest of the frames to 0. Each event class has different values of threshold TH and minimum number of successive frames min_{frames}, because each event class has different temporal lengths and the energies are distributed in different ways along the spectrum, as was explained in the sound description section.

Taking into account that the total number of frames from $\widehat{V_S}$ is T_{frames},the equation is:

$$\begin{gathered} if \quad \text{in } min_{frames} \text{ consecutive frames } \quad e(i)_{frame} > TH \quad \rightarrow \quad e(i)_{frame} = 1 \\ \text{In others cases } \quad \rightarrow \quad e(i)_{frame} = 0 \\ for\ i = 0,1 \ldots (T_{frames} - 1) \end{gathered} \tag{58}$$

Below is shown an example of threshold applied to the energy curve from output spectrogram $\widehat{V_S}$, figure 59, composed by a glass break event, area inside the red rectangle, and noise. The dashed red line indicates the threshold value TH.

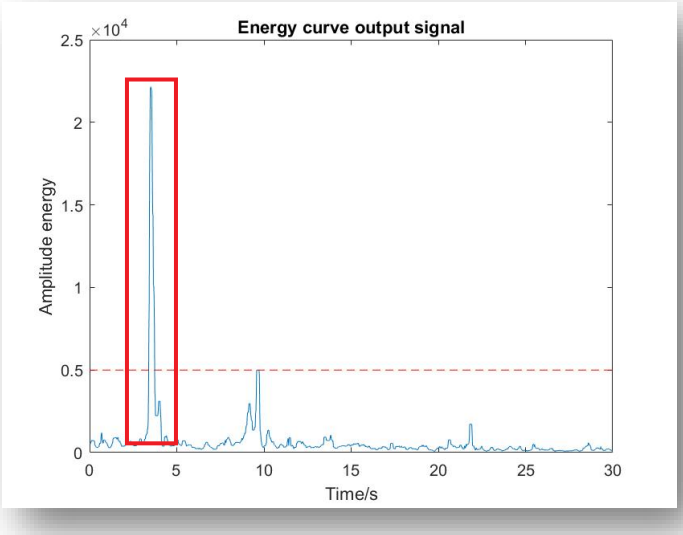

Figure 54. Energy curve from smoothed noisy spectrogram $\widehat{V_S}$ composed by a glass break event, area inside the red rectangle, and noise. The dashed red line indicates the threshold value TH.

As is shown in the figure 59, only a part of the frames have energies per frame e_{frame} above the threshold value TH, and in this case the number of consecutives frames is greater than the minimum number of sucesive frames min_{frames}, therefore this indicates the existence of an event.

In the following image, figure 60, shows the final result after applying the method, $Output_{TH}$:

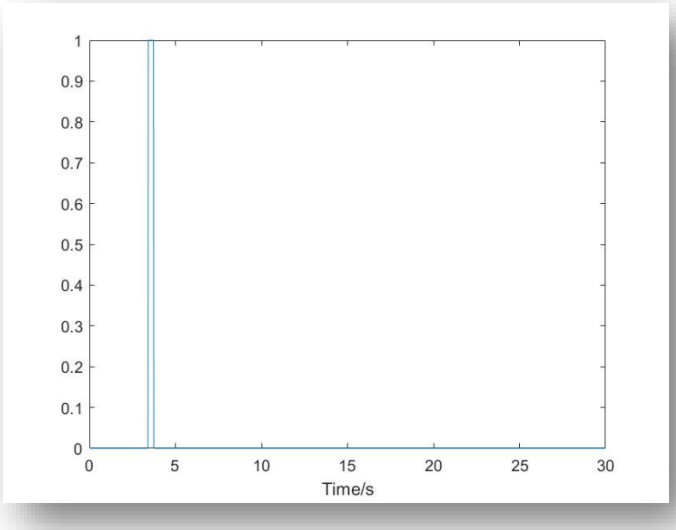

Figure 55. Signal after applying the threshold method.

However, the methodology applied above dos not work well for all kind of events, like in the case of baby cry events, this is because the babies takes a series of breaths between the cry, where the energy per frame e_{frame} is very small, so the event would not be detected because the consecutive frames with high energy would be less than the minimum number of successive frames min_{frames} required to detect the event. For this reason in this case, the main idea is to leave a small margin of consecutive frames with little energy, between the consecutive frames with high energy, considering them as breaths inside the baby cry event.

In the below example, figure 61, is shown an energy curve from output spectrogram $\widehat{V_S}$, which there is an event that corresponds to a baby cry, (21s-23s). As you can see in the photo, there are some zones of breathing zones.

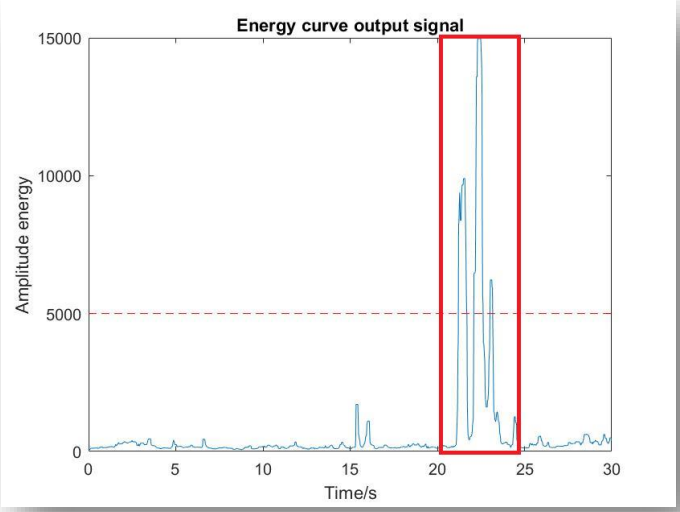

Figure 56. Example of Energy curve from smoothed noisy spectrogram $\widehat{V_S}$ composed by a baby crying event with breaths, area inside the red rectangle, and noise. The dashed red line indicates the threshold value TH.

After applying the previously methodology, the output $Output_{TH}$, is obtained:

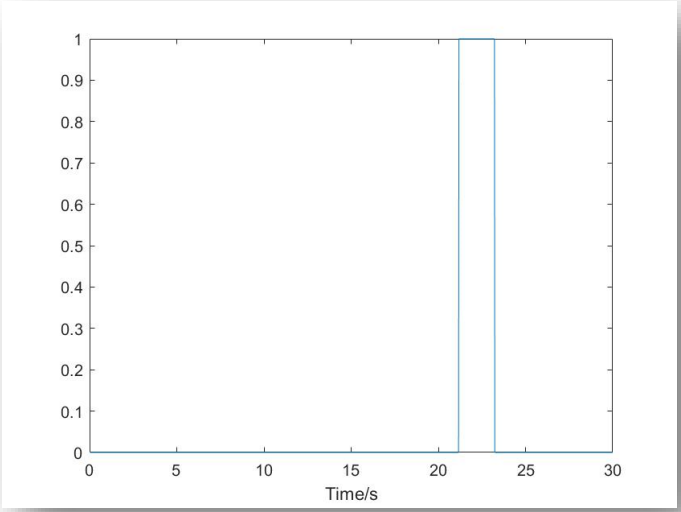

Figure 57. Signal after applying the threshold method, for the curve energy of the figure 61.

In the case of the noisy baby cry spectrogram developed in the previous sections, figure 43, is obtained, the following energy curve from output spectrogram $\widehat{V_S}$:

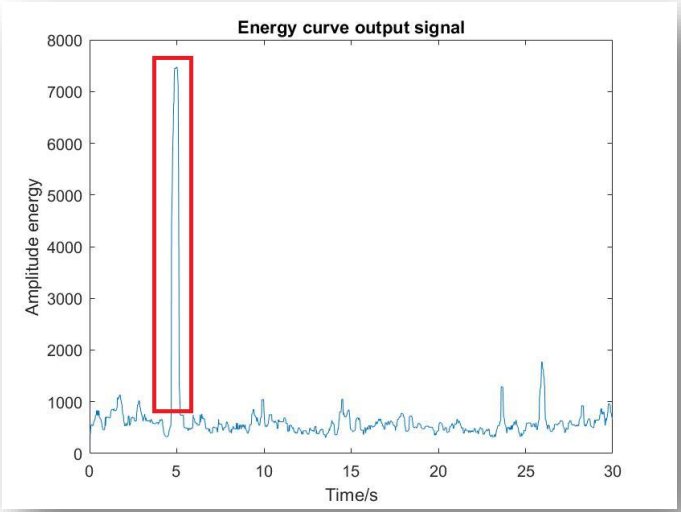

Figure 58. Energy curve from output spectrogram $\widehat{V_S}$ for the input spectrogram V of the figure 43.

And the output obtained $Output_{TH}$ is:

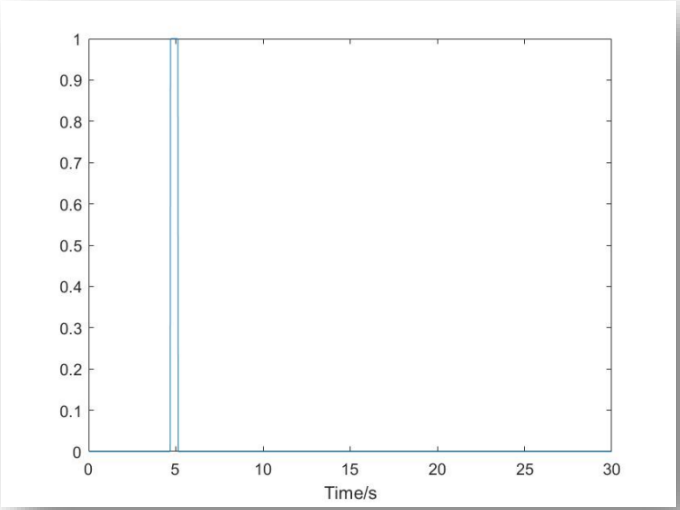

Figure 59. Signal after applying the threshold method $Output_{TH}$, for the energy curve of the figure 63.

,

In order to extract the event signal from the input spectrogram V, the input spectrogram V, is filtered with the output obtained after applying the threshold value TH, $Output_{TH}$, obtaining V_{output} , due to the noise areas will disappear by multiplying by the zeros.

$$V_{output}\,(f,t) = V(f,t) * Output_{TH} \tag{59}$$

To finish the noisy baby cry spectrogram, figure 43, is shown, the output spectrogram V_{output} ,where the event has been recovered, eliminating the noise form the input spectrogram V.

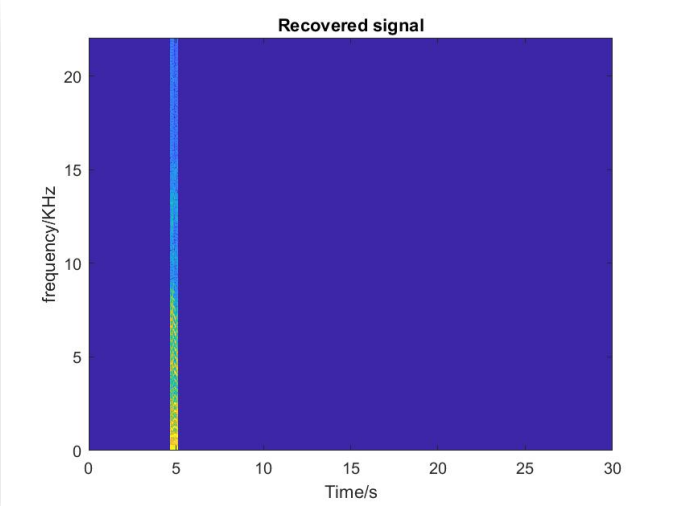

Figure 60. Output spectrogram V_{output} , for the input noisy spectrogram of the figure 43.

5.6 Post-processing

Median filter is a nonlinear digital filtering technique used for signal smoothing. The main idea of this technique is to loop the signal value by value, using a moving window whose length is defined by the user. Each value of the signal is replaced by the median of the window, taking into account that the values of the window have to be sort from least to greatest.

For example the median filter of x:

- Considering, X= [3 4 1 7 5] and taking as filter length n=3, this mean that must be taken a sample before and after, the current one.

- Taking into account that before the first sample there is no other sample before is put 0.

- So, we have 0 3 4 in this case the numbers are already sort and now is selected the number of the intermediate number that is 3.

- Now the window moves to the second position so we have 3 4 1, and the numbers are sort 1 3 4, in this case the intermediate number is 3 to.

- The procedure for the following values is the same, so at the end is obtained that the output is y= [3 3 4 5 5].

Below, figures 66 and 67, is shown a comparative between a signal with an event of a baby crying before and after applying of applying the median filter.

Energy curve, composed of a baby cry event and noise, from the output spectrogram $\widehat{V_S}$, before applying the median filter. The area inside the red section corresponds to the baby cry event:

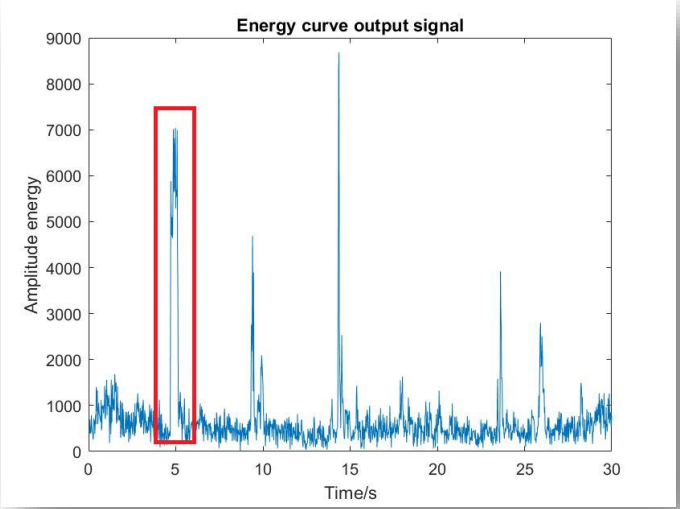

Figure 61. Energy curve, composed of a baby cry event, area inside the red rectangle, and noise, from the output spectrogram $\widehat{V_S}$, before applying the median filter.

And the energy curve after applying the median filter:

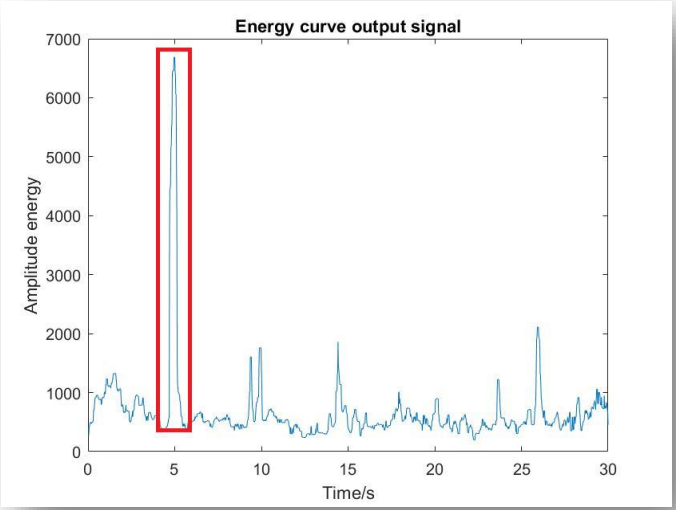

Figure 62. Energy curve, composed of a baby cry event, area inside the red rectangle, and noise, from the output spectrogram $\widehat{V_S}$, after applying the median filter.

As can be seen in the previous figures, 66 and 67, the median filter smooths the signal obtaining a cleaner signal. However, this procedure is not suitable for all events, like in the case of gunshots, which worsens the result

In figure 68 shows an energy curve, composed of a gunshot event and noise, from the output spectrogram $\widehat{V_S}$, before applying the median filter.

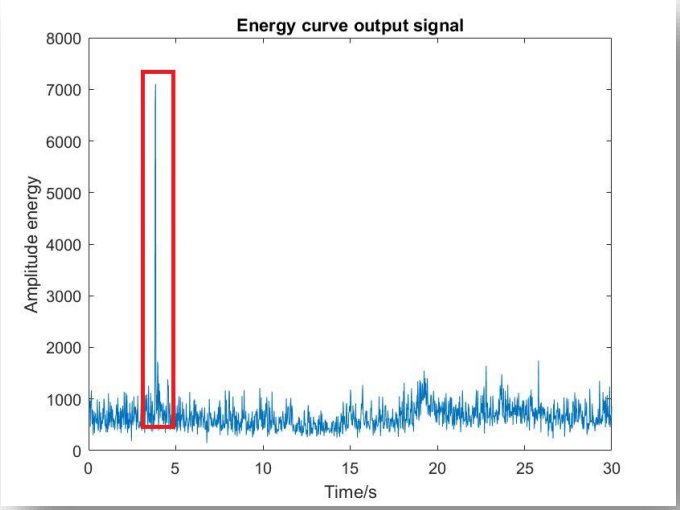

Figure 63. Energy curve, composed of a gunshot event, area inside the red rectangle, and noise, from the output spectrogram $\hat{V}_S$, before applying the median filter.

And the gunshot energy curve after applying the median filter:

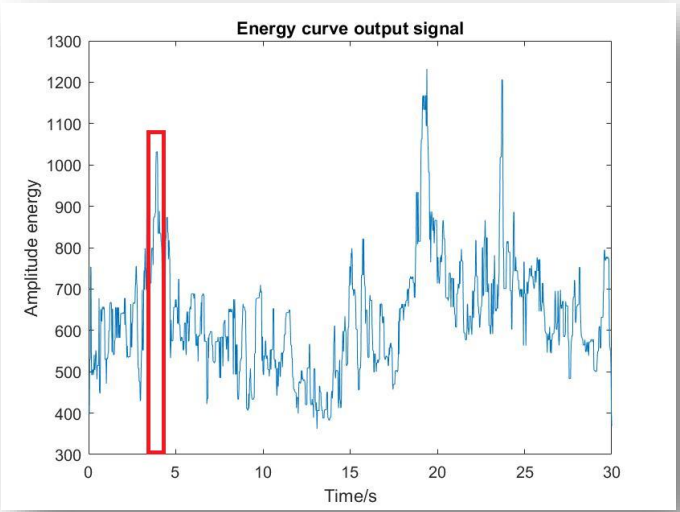

Figure 64. Energy curve, composed of a gunshot event, area inside the red rectangle, and noise, from the output spectrogram $\hat{V}_S$, after applying the median filter.

In this case the median filter doesn't work well because the event that is a gunshot is like an impulse of short duration.

5.2 Scheme algorithm

TRAINING PHASE

Inputs: clean event signals.

Outputs: training spectrogram $\mathbf{V}_{Train}$ and event dictionary $\mathbf{W_S}$.

Initialize W_S, H_S

for i=1: iterations

Update W_S, H_S, *using equations (40) and (41).*

end

TEST PHASE

Noise dictionary

Inputs: input noisy spectrogram **V**, number of basis, sparsity parameter **λ**.

Outputs: estimated noise dictionary $\mathbf{W_n}$ and noise spectrogram $\mathbf{L_N}$.

Initialize: W_n, H_n, S *with non negative values.*

for i=1: iterations

Update W_n, H_n *and* S *using equations (44), (45), (46).*

End

At the end is calculated $L_N = W_n * H_n$

Source separation

Inputs: input noisy spectrogram **V**, training spectrogram $\mathbf{V_{Train}}$, event dictionary $\mathbf{W_s}$, estimated noised dictionary $\mathbf{W_n}$, noise spectrogram $\mathbf{L_N}$,$\mathbf{T_0}$, **rmin**, **rmax**.
Outputs: temporal activations $\mathbf{H_s}$ and $\mathbf{H_n}$.

- **Case frequency weighting**

  Calculate frequency weights using equations (49), (50), (51) and (52).

- **Case temporal weighting**

  Calculate temporal weights using equations (53), (54) and (55).

- **Case combined weighting**

  Calculate temporal weights using equations (56).

  end

  Initialize H_s *and* H_n *with non negative values.*

  for i=1: iterations

  Update H_s, H_n *using equation (47).*

  end

6. EXAMPLES

During the explanation of the implemented method, an example of baby cry event has been developed, in this section others examples will be shown.

2 BABY CRY

The system also is able to detect more than one event, as long as they belong to the same event class.

- Input noisy spectrogram V, which contains 2 baby cry events around the second 5 and second 19.

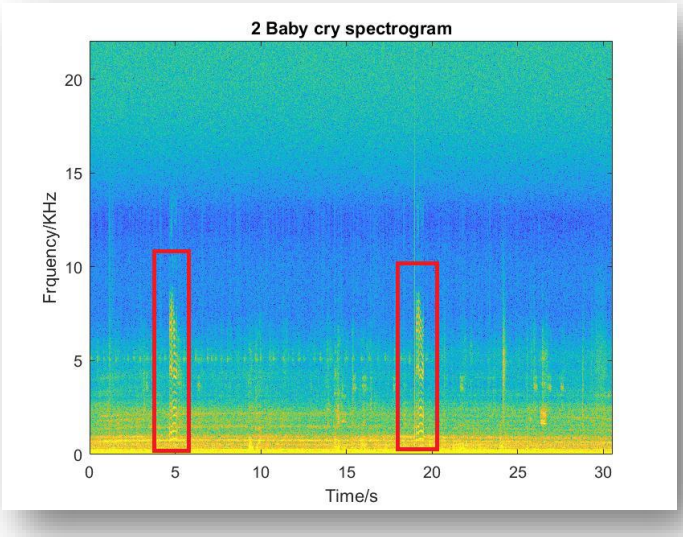

Figure 65. Input noisy spectrogram V, which contains 2 baby cry events around the second 5 and second 19.

- Frequency weight matrix G_{freq}, which enhances the subbands between 6 KHz and 8 KHz.

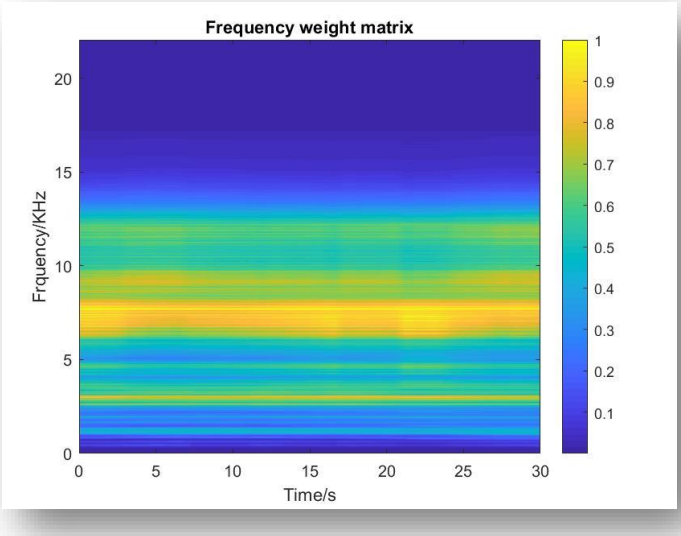

Figure 66. Frequency weights matrix for the input spectrogram V of the figure 70, which enhances the sub-bands between 6 KHz and 8 KHz. So this subbands have more contributions in constructing the events.

- Temporal weight matrix G_{temp}, which assigns high probabilities to the frames near to the second 5 and second 19.

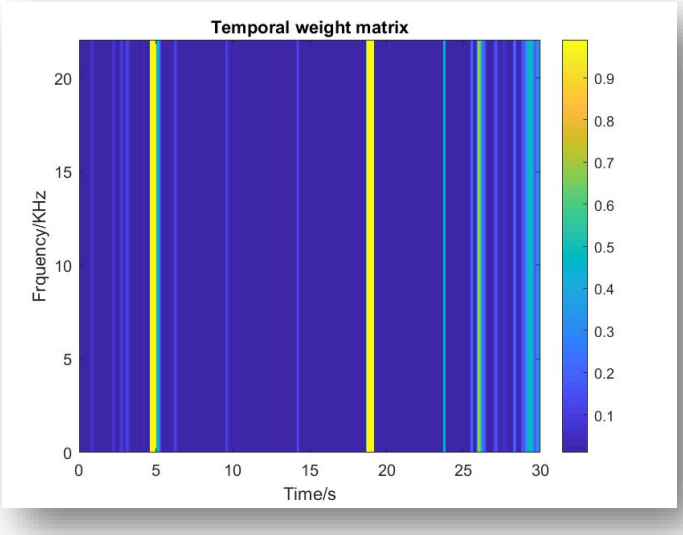

Figure 67. Temporal weight matrix G_{temp} for the input spectrogram V of the figure 70. In this matrix the frames near to the second 5 and the second 19 have the probaibilities higher associated to an event.

- Combined weight matrix $G_{freq+temp}$, which enhances the subbands between 6 KHz and 8 KHz, in the frames near to second 5 and second 19.

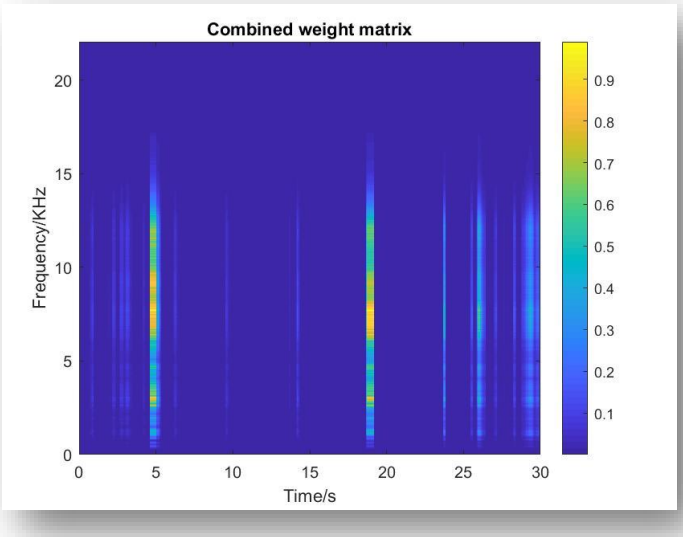

Figure 68. Combined weights matrix $G_{freq+temp}$ for the input spectrogram V of the figure 70, whose enhanced area is in the frames around second 5 and 19 and the subbands near between 6 KHz and 8 KHz.

- Output spectrogram V_{output} , in which the events from the input spectrogram V are recovered eliminating the noise.

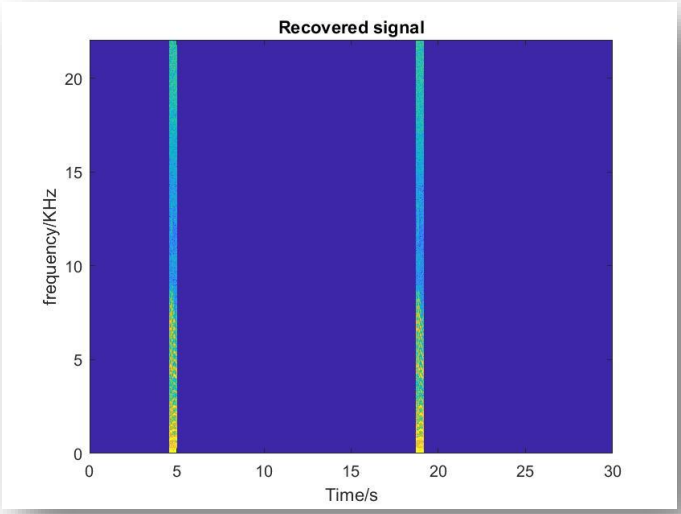

Figure 69. Clean spectrogram for glass break example, figure 70.

Glass break

- Input noisy spectrogram V, which contains a glass break event around the second 25.

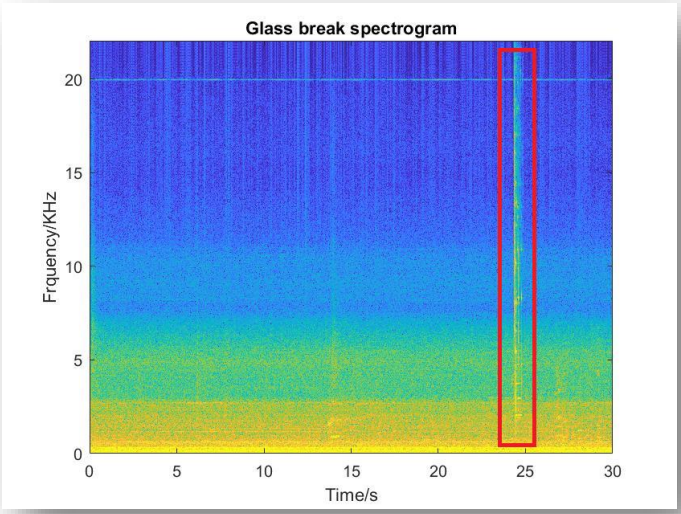

Figure 70. Input noisy spectrogram V, which contains a baby cry event around the second 25.

- Frequency weight matrix G_{freq}, which enhances the subbands between 11 KHz and 16 KHz.

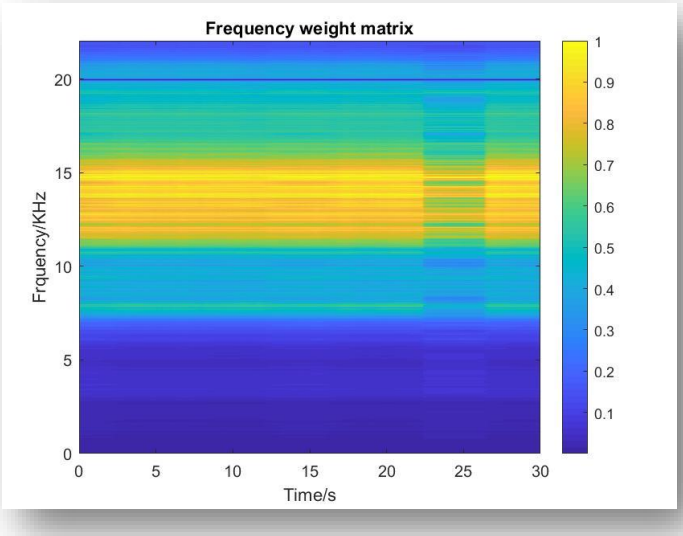

Figure 71. Frequency weights matrix G_{freq} for the input spectrogram V of the figure 75, which enhances the subbands between 11 KHz and 16 KHz. So this subbands have more contributions in constructing the event.

- Temporal weight matrix G_{temp}, which assigns high probabilities to the frames near to the second 24.

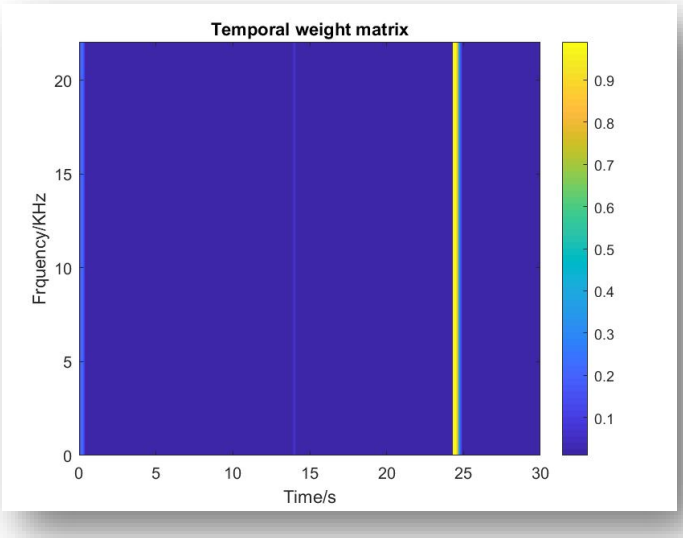

Figure 72. Temporal weights G_{temp} matrix for the input spectrogram V of the figure 75. In this matrix the frames near to the second 24 have the probaibilities higher associated to an event.

- Combined weight matrix$G_{freq+temp}$, which enhances the subbands between 11 KHz and 16 KHz, in the frames near to the second 24.

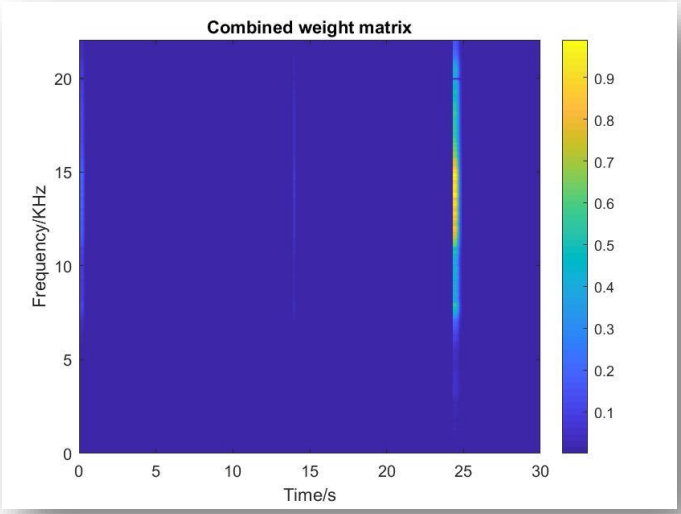

Figure 73. Combined weight matrix $G_{freq+temp}$ for the input spectrogram V of the figure 75, whose enhanced area is in the frames around second 24 and the subbands between to 11 KHz and 16 KHz

.

- Output spectrogram V_{output} , in which the glass break event from the input spectrogram V is recovered eliminating the noise.

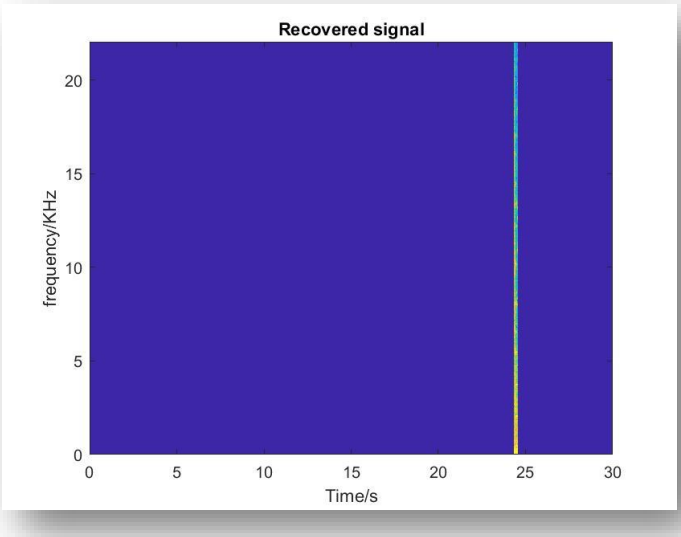

Figure 74. Output spectrogram V_{output} , for the input spectrogram V of the figure 75. In which the glass break event from the input spectrogram V is recovered eliminating the noise.

Gunshot

- Input noisy spectrogram V, which contains a gunshot event around the second 19.

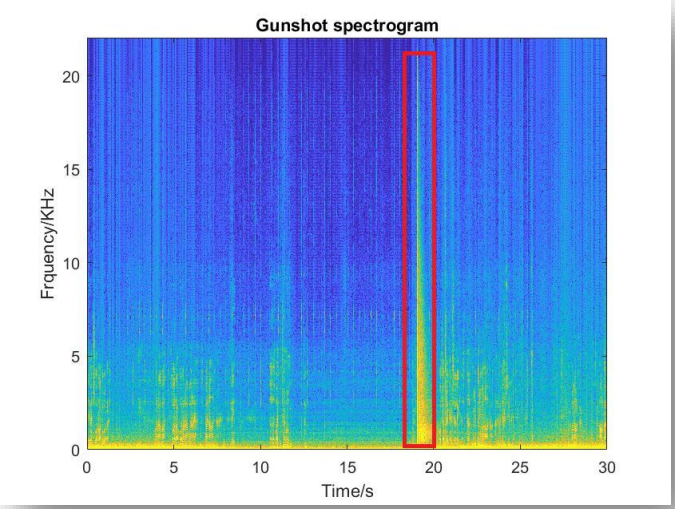

Figure 75. Input noisy spectrogram V, which contains a gunshot event around the second 19.

- Frequency weight matrix G_{freq}, which enhances the subbands between 2 KHz and 4 KHz and other subbands between 5 KHz and 8 KHz.

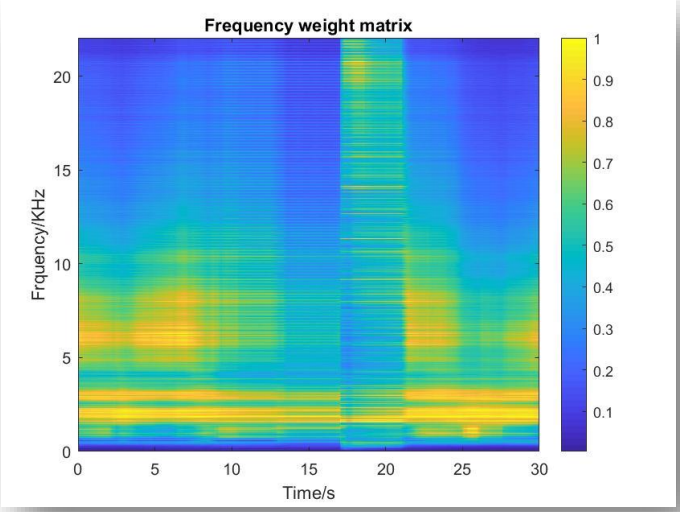

Figure 76. Frequency weights matrix G_{freq} for the input spectrogram V of the figure 80, which enhances the subbands between 2 KHz and 4 KHz and other subbands between 5 KHz and 8 KHz. So this subbands have more contributions in constructing the event

- Temporal weight matrix G_{temp}, which assigns high probabilities to the frames near to the second 19.

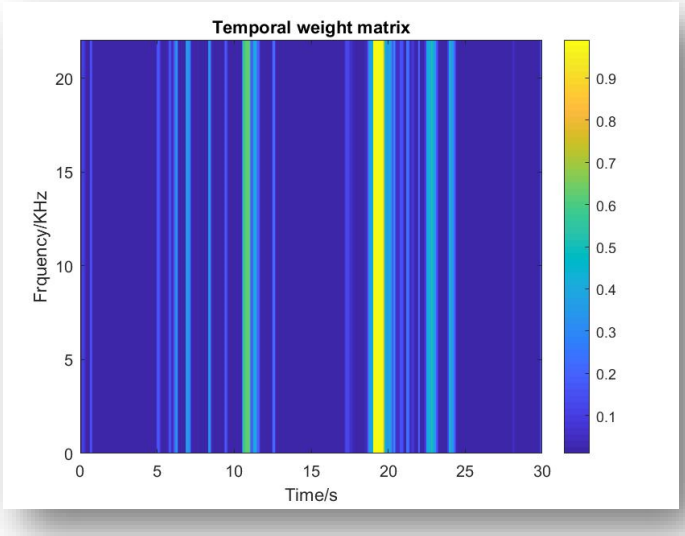

Figure 77. Temporal weight matrix G_{temp} for the input spectrogram V of the figure 80. In this matrix the frames near to the second 19 have the probaibilities higher associated to an event.

- Combined weight matrix$G_{freq+temp}$, which enhances the subbands between 2 KHz and 4 KHz, in the frames near to the second 19.

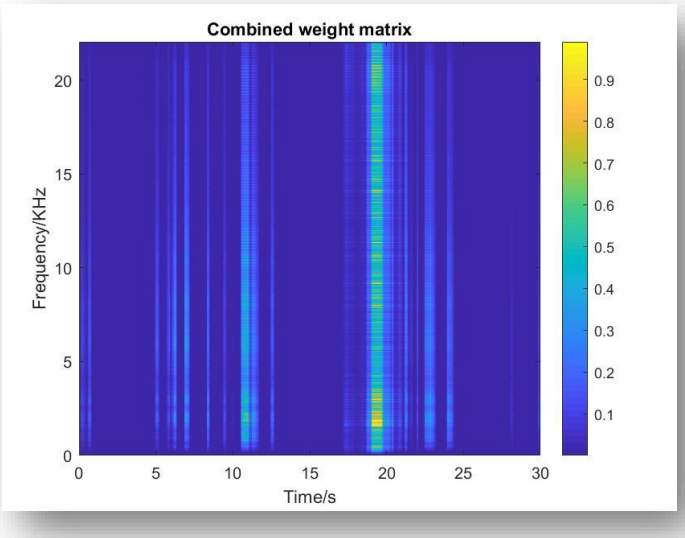

Figure 78. Combined weights matrix $G_{freq+temp}$ for the input spectrogram V of the figure 80, whose enhanced area is in the frames around second 19 and the subbands 2 KHz and 4 KHz.

- Output spectrogram V_{output} , in which the gunshot event from the input spectrogram V is recovered eliminating the noise.

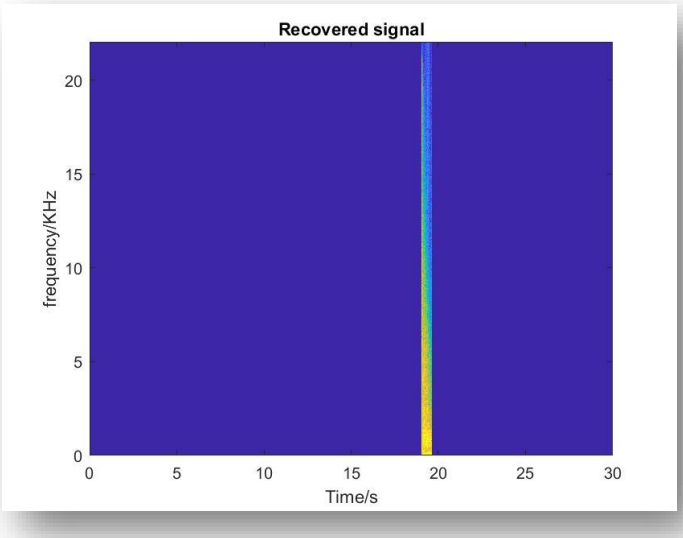

Figure 79. Output spectrogram V_{output} for the input spectrogram V of the figure 80. In which the gunshot event from the input spectrogram V is recovered eliminating the noise.

7. EVALUATION

In this chapter, are described: the database that is used, the metrics used to calculate the percentages of success or errors in the optimization and later in the test phase of the system, the optimization of the system, setup, where are detailed the values for all parameters of the system and finally the results.

7.1. Databases

For the realization of the system and subsequent testing, one database have been used from Task2 of the DCASE 2017 challenge, which contains two datasets, development dataset that is used in the training phase in order to obtain the spectral bases dictionary for the events W_s and also is used to optimize the performance of the system and the other dataset is the evaluation dataset that is used to check the sensitivity in the test phase.

Each dataset contains for each event class (baby cry, gunshot, glass break) a set of 500 recordings with different SNR levels (6dB, 0dB,-6dB) and of duration 30 s each one. These 500 recordings are composed by 250 mixtures of clean event signals with background noise, from different noisy places like parks, people, bus station among others, and the others 250 only composed by background noise. In addition in the development dataset there are three additional subsets, one for each type of event, composed of recordings of isolated clean events, 474 events in total, which are used in the training phase.

7.2. Metrics

In this section are detailed the metrics used, which are necessary to obtain a numerical values that quantifies, how is the sensitivity of the system, in other words, it indicates how good is the system. These metrics are used in the optimization of the system and also in the test phase.

It is important to mention that a reference list of events has been created, in which the start and end times for each event of the different recordings are written, in order to compare with the predictions of the system.

Below are the metrics used in this system:

- **True Positive, TP:** A detected event that corresponds with an event in the reference list, with the condition that the output onset is within a range of 500 ms of the actual onset, as described in the reference [3].

- **False Positive, FP:** A detected event that has no correspondence with any events in the reference list.

- **False Negative, FN:** An event in the reference list that has no correspondence with any event to the system output.

- **N_R:** Number of total recordings.

- **The error ratio ER:** it is a measure that quantifies the proportion of the system's wrong predictions. This measure is calculated like the sum of false negatives FNs plus false positives FPs divided by the number of total recordings N_R.

$$ER = \frac{FNs + FPs}{N_R} \tag{60}$$

- **The precision P**, is the ratio that, quantifies the proportion of the true positives TPs belong to the detected events, that are the sum of true positives TPs and false positives FPs.

$$P = \frac{TPs}{TPs + FPs} \tag{61}$$

- **The recall R**, is the ratio of the true positives TPs to the all recordings with event in the reference list, that are the sum of true positives TPs and false negatives FNs and quantifies the proportion of the events that the system is able to identify.

$$R = \frac{TPs}{TPs + FNs} \tag{62}$$

- ***F-Score***, it is a measure that combines, the precision P and the recall R, so take into account, false positives FPs and false negatives FNs.

$$F - Score = \frac{2PR}{P + R} \tag{63}$$

All the previous measurements are dimensionless, to obtain the result in percentage (%) is necessary to multiply by 100.

7.3. Optimization

In the optimization phase, the values of the system parameters are obtained in order to optimize the operation of the system. To do this, in first place, an evaluation is performed using different threshold values TH for each type of event, the best results have been obtained with the following values threshold TH, 2000 for baby cry and gunshot, and 5000 for glass break. Once these values are obtained, other evaluations are carried out for each type of event, using the previously obtained values using different numbers of bases K and the parameter λ, which controls the weight of sparsity parameter S in the noise dictionary learning.

In the following table are shown the different percentages for F-score for baby cry events:

K \ λ	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 0.9$
$K = 16$	F-score=86 %	F-score=85.9 %	F-score=83 %
$K = 32$	F-score=84.4 %	F-score=83 %	F-score=78.9 %
$K = 64$	F-score=84.5 %	F-score=81.7%	F-score=79.4 %

Table 1. F-score (%) using different values of K, λ for baby cry

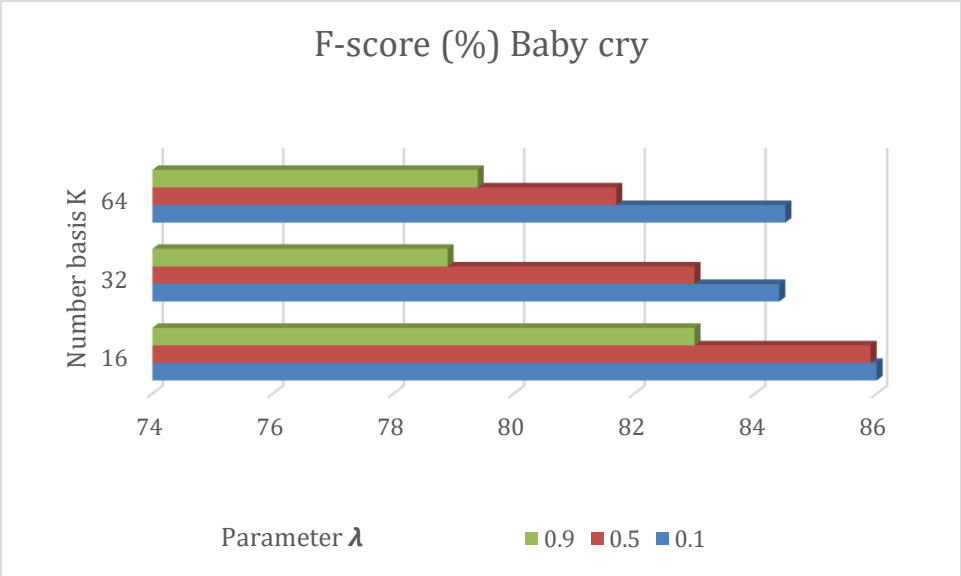

Figure 80. Comparative F-score (%) for different values of K, λ for baby cry

As can be seen in the graph, using a high number of bases the result obtained is worst that using a number of bases K equal to 16, because using a higher number of bases the redundancy of the dictionaries increase and bring problems in the description of the spectrograms, using non-corresponding bases, for example the noise spectrogram L_N described by bases related to the events W_s or vice versa. Regarding the sparse parameter S, in noise dictionary learning section, using a larger value of λ ,the event components retained are bigger and there would be many event components in the noise estimated spectrogram L_N, therefore the performance of the system is better using a small value of λ. In this case using $\lambda = 0.9$ is obtained the worst result, as is expected.

So, as can be seen in the graph, the best result obtained for F-score is 86%, for $K = 16$ bases and a value of $\lambda = 0.1$.

In the following table are shown the different percentages for F-score for gunshots events:

K \ λ	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 0.9$
$K = 16$	F-score=70.5 %	F-score=75.4 %	F-score=77 %
$K = 32$	F-score=71.2 %	F-score=77.5 %	F-score=77 %
$K = 64$	F-score=70.9 %	F-score=74.95 %	F-score=74.4 %

Table 2. F-score (%) using different values of K, λ for gunshot

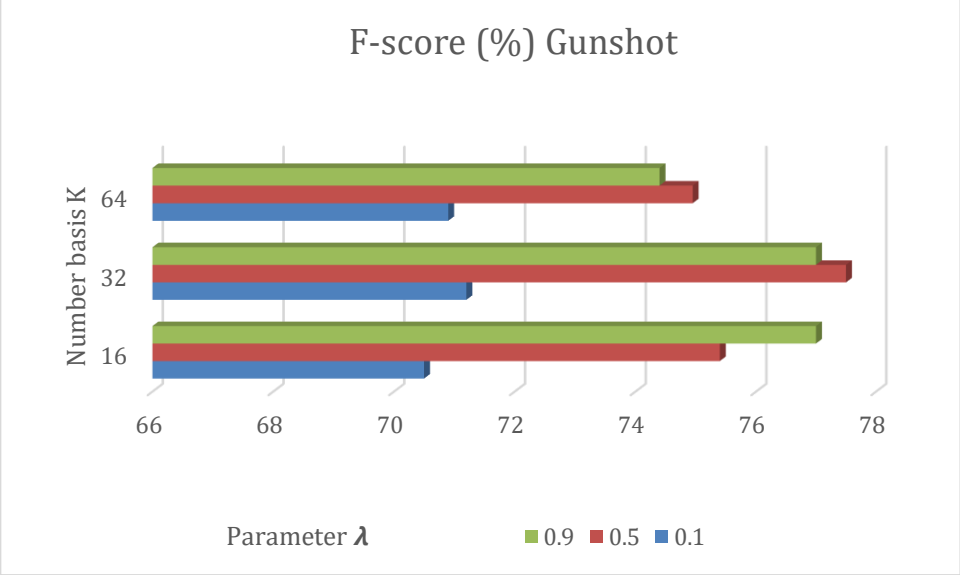

Figure 81. Comparative F-score (%) for different values of K, λ for gunshot

As previously has been explained, a high number of bases K degrades the performance of the system, obtaining for $K = 64$ the worst global result of F-score. However in this case, considering a larger value of λ the system has a better performance, when it should not be, this is because the gunshots events have characteristics similar to the noise, for example both have much energy in low frequency, so the system sometimes considers that the event is noise.

As can be seen in the graph, the best result obtained for F-score is 77.5%, for K= 32 bases and a value of 0.5 for the λ parameter.

In the following table are shown the different percentages for F-score for glass break events:

K \ λ	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 0.9$
$K = 16$	F-score=89.5 %	F-score=90.6 %	F-score=89.3 %
$K = 32$	F-score=89.9%	F-score=90.2 %	F-score=89.2 %
$K = 64$	F-score=89.2 %	F-score=89.9%	F-score=88.6 %

Table 3. F-score (%) using different values of K, λ for glass break

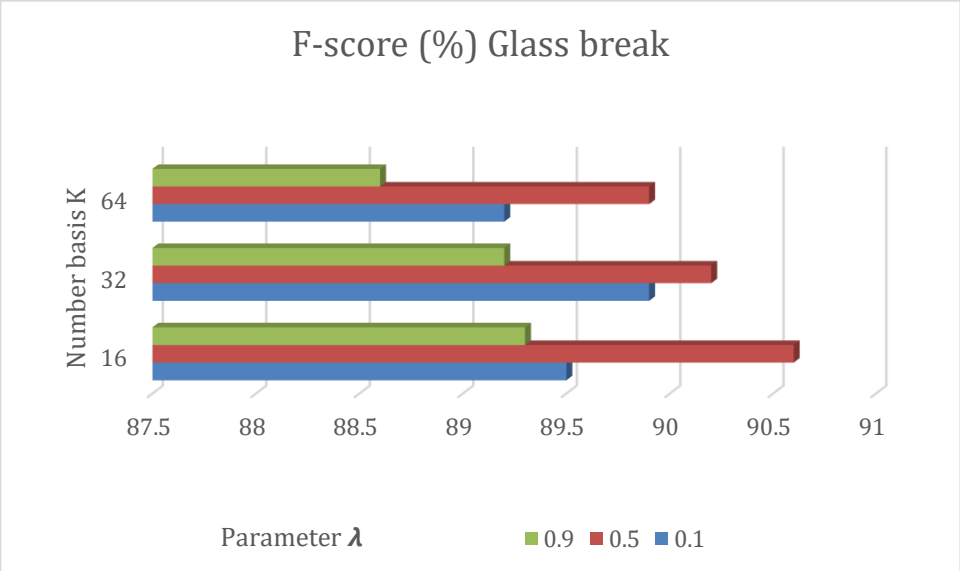

Figure 82. Comparative F-score (%) for different values of k, λ for glass break

As in the two previous cases, the worst result obtained is using a high number of bases $K = 64$ degrading the performance of the system. Regarding the value of λ, the worst result is for a very large value of λ, because it retains many components of the event within the estimated noise.

As can be seen in the graph, the best result obtained for F-score is 90.6%, for $K = 16$ bases and a value of $\lambda = 0.5$.

7.4 Setup

After completing the optimization stage, all values for the implementation of the system are already know, which are:

- Sampling rate: *Fs* = 44100 Hz.
- Number of points STFT: N = 2048.
- Overlap STFT: 50 %, using a hamming window.
- Number of basis K: for baby cry and glass break events $K = 16$ and for gunshot events $K = 32$.
- λ parameter: for baby cry events $\lambda = 0.1$, and for glass break and gunshot events $\lambda = 0.5$.
- Number of iterations NMF: $n = 200$
- Averaging frames for smoothing: $T_0 = 200$, value obtained in [39].
- $rmin = 0$.
- $rmax = 0.5\ x\ maximum(Ae)$.
- Threshold values TH: for baby cry and gunshot events, $TH = 2000$ and for glass break, $TH = 5000$.
- Minimum number of consecutive frames $\mathrm{minframes}$ in threshold stage: for baby cry events $\mathrm{minframes} = 15$, and for gunshot and glass break events $\mathrm{minframes} = 9$.

7.5 Results

Using the parameters K and λ obtained in the optimization stage, a testing stage is carried out in order to check the efficiency of the developed system. For this, is used other database different from the training and optimization database, with 500 recordings for each type of event, 250 with event and others 250 with only noise. Each audio is tested using the model developed with all types of weighting and without it and also using other model like the semi-supervised model, in order to compare the efficiency of the developed model with this more basic model.

In the following table is shown the error ratio and the F-score (%) using the developed system in this project, using different kinds of weighting or without weighting and also this results will be compared with a semi-supervised model.

Method	Baby cry		Gunshot		Glass break		Average	
	ER	F-score (%)	ER	F-score (%)	ER	F-score (%)	ER	F-score (%)
Combined weighting	0.24	86.4	0.37	77.2	0.22	87.6	0.27	83.77
Frequency weighting	0.25	85.9	0.38	76.1	0.22	87.6	0.28	83.2
Temporal weighting	0.26	84.8	0.39	75.8	0.3	82.3	0.32	80.97
No weighting	0.27	84.4	0.39	75.9	0.3	82.3	0.32	80.87
Semi-supervised NMF	0.44	71.63	0.6	57.6	0.47	69.5	0.50	66.24

Table 4.Error ratio and F-score (%) for different models and events

In the following graph is possible to see a comparative between the average percentages of f-score using different models:

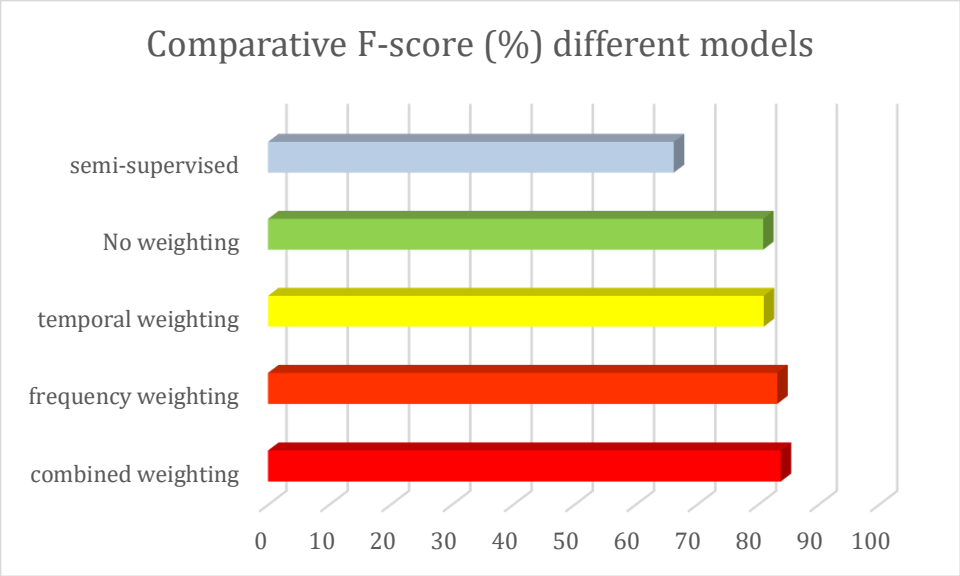

Figure 83. Comparative F-score (%) between different models

As can be seen in the table and in the graph, the model developed in this project, obtains a better score clearly, 14,83% more, with a small error ratio, compared to the semi-supervised model. This difference is mainly due to the fact that the system developed in this project performs a source separation by supervised, using the bases training dictionary W_s and the estimated noise dictionary W_n, in the case of the semi-supervised model, only is used the bases training dictionary W_s and is not used information related to the noise. In addition semi-supervised model don't have any kind of weighting.

Using weighting over the developed system of this project, up to 5% improvement in F-score is obtained for example in the case of glass break events. However if we perform the average improvement in F-score between the three classes, this percentage is lower because for example for baby cry and gunshot the improvement is smaller. Using the combined weighting around 3% of improvement globally is obtained in F-score, using frequency weighting the improvement is around 2,3% of improvement globally, however using only the temporal weighting, the result is practically the same as in the case without weighting. The main reason associated with these small improvements using weighting, is because in the most cases, the false detections, false positive or false negative, are due to the bases

training dictionary W_s does not always work well for all input audios and therefore the source separation is wrong even using weighting.

As previously explained, depending on the type of event used in the developed system, the results obtained are better or worse, this can be seen in the following comparative.

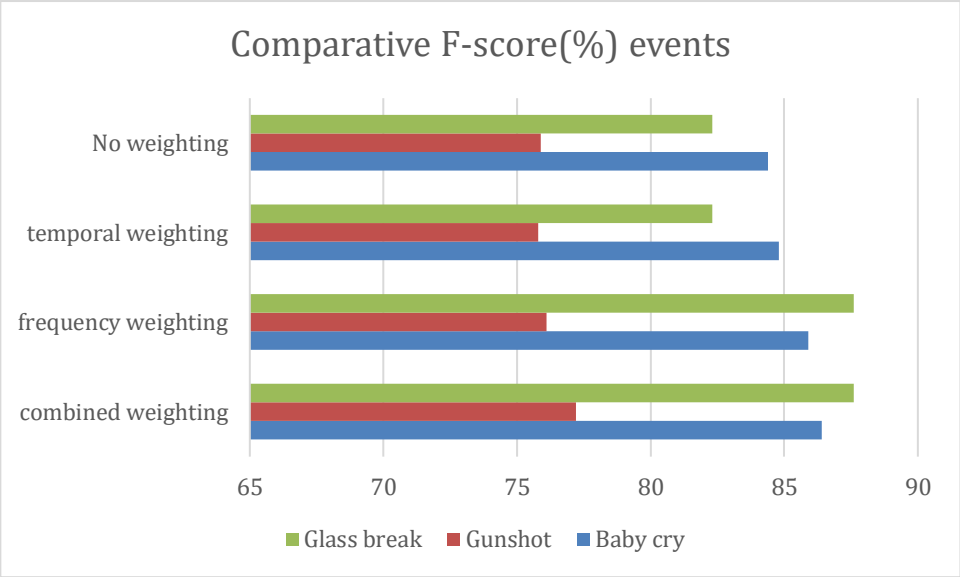

Figure 84. Comparative F-score (%) between events

As is shown in the comparative, the developed system works well for baby cry events and and glass break and the worst operation of the system occurs in the case of gunshot events detection, because they have features very similar to the noise, are impulsive events with the highest amount of energy distributed in the low frequency and thus over-laps with the ambient noise zone, instead baby cry and glass break events have more amount of energy outside this area, at high and medium frequency.

8. CONCLUSIONS

Finally, after developing the proposed system and later checking the performance of the system, the main objective has been achieved, which was, the design and development of a system implemented in Matlab capable of reducing noise for the detection of sound events using a weighted Non-Negative Matrix factorization (NMF) approach. In addition all the secondary objectives that were proposed have been achieved, for example in order to overcome this main objective, an extensive study of the different techniques applied in the detection events was previously carried out, knowing from the first algorithms used and how they have evolved over the years and also the development of a databases was necessary in order to develop the system and later test it. To finish, the testing stage has been analysed obtaining some clear results:

- The developed model has a clearly better score in F-score, comparing it with a semi-supervised model,14,83% more, this is because semi-supervised model does not take into account the noise dictionary W_n and does not use any kind of weighting, like in the case of the develop model in this project.

- The system works better for glass break and baby cry events and worse for glass break, due to the similarity with the noise, where the energy is mainly located in low frequency like in gunshots events.

An user interface has also been developed as initially proposed, being very intuitive and visual for the user, in section 10.1 is explained.

9. FUTURE LINES

In this section, some future lines of researched are listed with the aim of improving the operation of the system and also introducing new functionalities:

- Extension of the events that the system can recognize, expanding the applications to which this type of events detection systems could be applied, for example events of people falling to the ground, in order to detect falls of old people who lives alone.

- Recognition of more than one event of different class in the same audio. It is very useful in order to detect some situations of violence.

- Introduce new methodologies to the system, like the correlation, for better recognition of events such as shots, because they are impulsive sounds and have spectral bases similar to the noise.

- System optimization using a convolutional neural network (CNN) in order to reduce they memory requirement, due to event detection systems need high memory and have a high computational demand.

10. ANNEXES

10.1 Graphical user interface

In order that the user can interact with the system is created a very intuitive graphical interface, which provides a lot of information the user in a visual way .The user can select the input audio, what type of event the system has to detect, baby cry, gunshot or glass break and what model to use for detection, with weighting (frequency, temporal or combined) or without weighting, also the user can modify the default values of some parameters like number of bases K, 16,32 or 64 because the training phase was done with this number of bases, parameter λ, threshold value TH and the limits values $rmin$ and $rmax$. The graphical interface will show multiple graphs, in the first box the spectrogram of the input signal V will be showed, in the second box the user can select with a dropdown the graphs to visualize between several options: weight matrix G, estimated noise spectrogram L_n, in the case of using temporal weighting the limits rmin and rmax, and the convergence functions for the estimated noise dictionary learning and for the source separation. In the third box will be showed the output spectrogram V_{output} , in the case that the system detects an event. Also the interface will show a message when the system finishes of processing the input audio, indicating if the system has detected any event or not, and in the case of detecting event, it indicates the onset and offset times.
In addition the interface has a complete audio player, which the user can select if he want to listen to the input or output audio, also can control the volume or can scroll through the audio, using the audio slider. At the end if the user wishes, can also save the output audio.

When the interface is open the following screen appears:

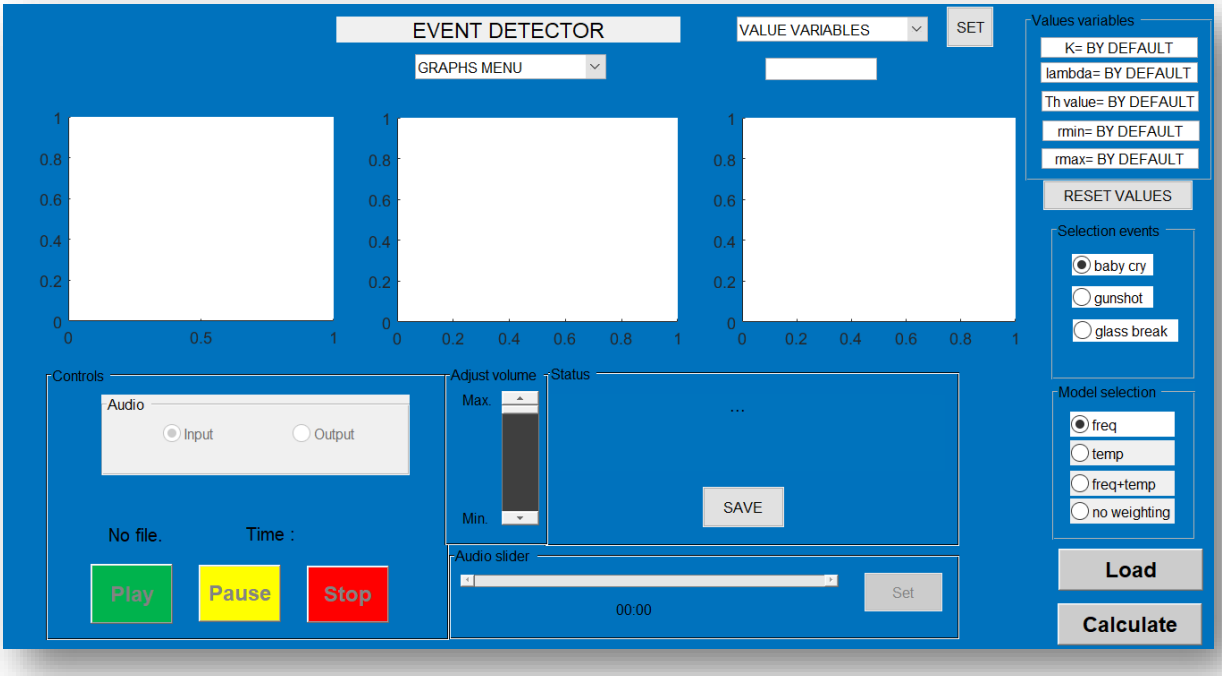

Figure 85. Interface home screen.

Pressing the button “Load” the user can select an input audio from the directory that the user wants:

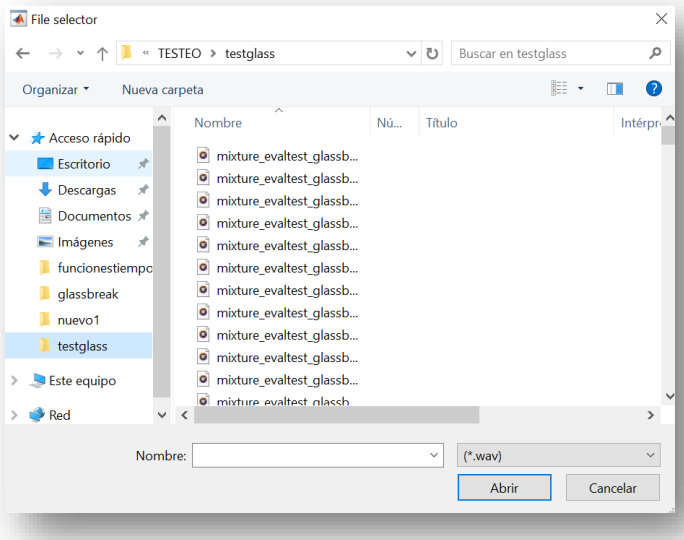

Figure 86. Interface file selector.

If the user does not load any input audio and press the button “Calculate”, a warning appears.

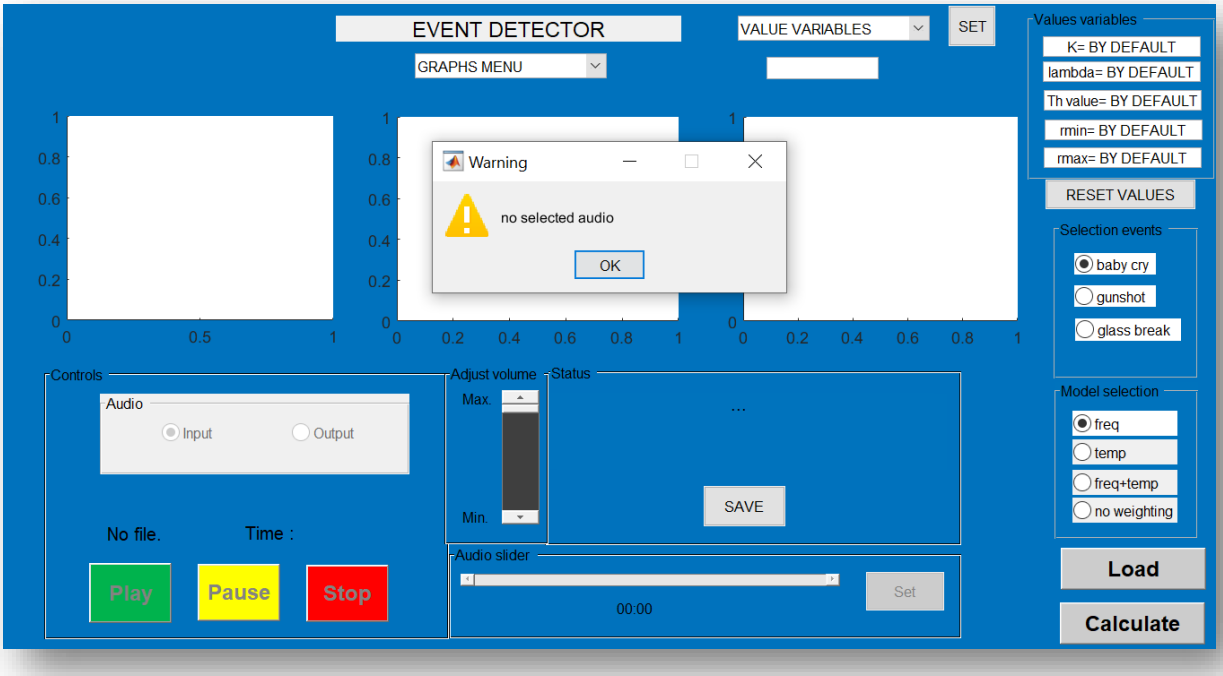

Figure 87. Warning from the interface, because input audio is not selected.

Once the input audio is loaded, the file name appears on the screen and can be heard by the user before processing it. If the user presses the button “Play”, the sound stars to sound and temporary marker will indicate the playing time, figure 93. While the sound is playing the user can adjust the volume and also move forward or backward in audio playback using the audio slider and pressing the button “Set”.

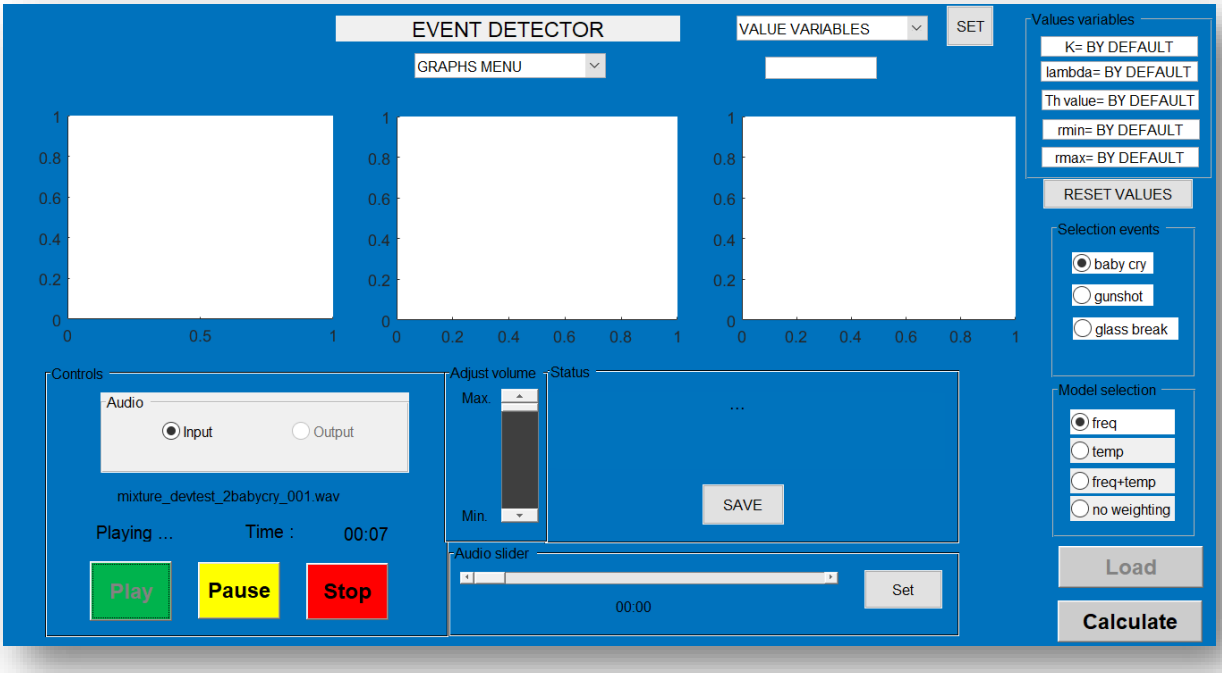

Figure 88. Interface screen when playing an audio.

As previously explained, the user can modify the default value of the paramaters selecting a parameter from the “VALUE VARIABLES” list, figure 92, and then write the new value and press the “SET” button. In the case that the user wants to rest the default values, he would have to press the “RESET VALUES” button. For example, in figure 93, new values for lambda λ and for the threshold TH have been introduced.

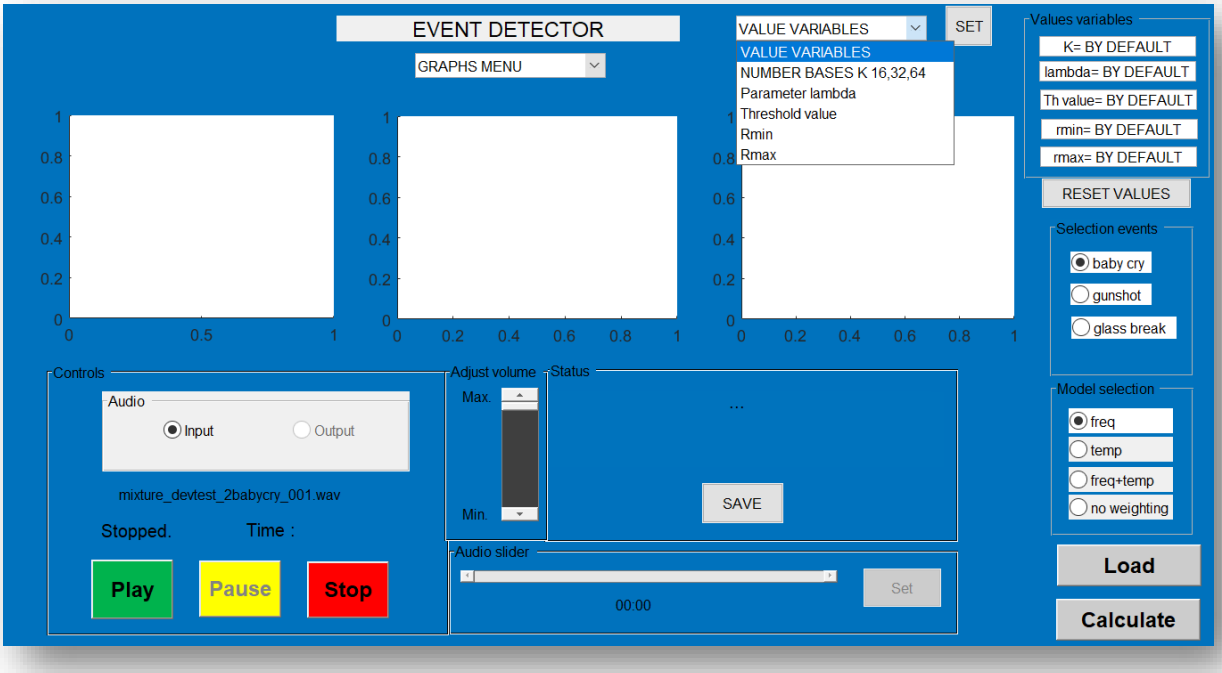

Figure 89. Parameter list from the interface, to select the parameter that the user want to change of value.

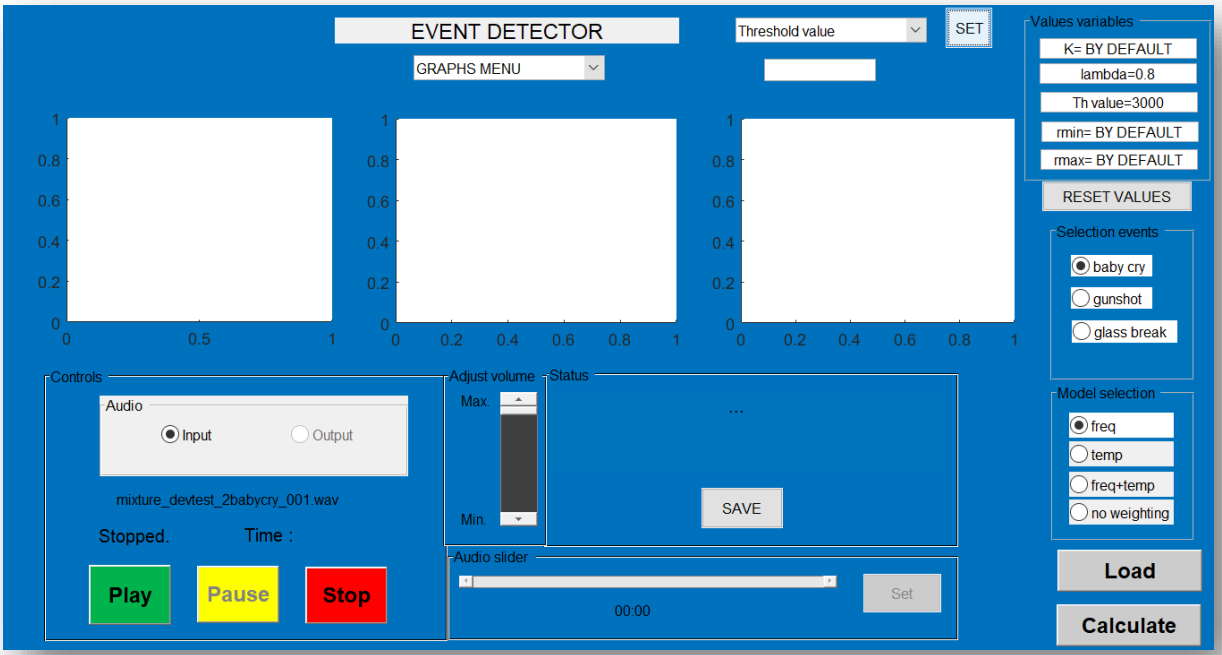

Figure 90. Parameters from interface with a new value, lambda=0.8 and threshold value = 3000.

Then pressing the “Calculate” button the system processes the input audio, when the process ends the input and output spectrograms are shown, in the first box and third box respectively. In the second box the user can choose which graph to show choosing between diferent options: weight matrix, estimated noise spectrogram, rmin and rmax in temporal weighting or the cost functions for estimated noise dictionary or source separation, using the graphs menu. Also the onset and offset times that correspond with the events appears on the screen and the possibility of playing the output audio is activated.

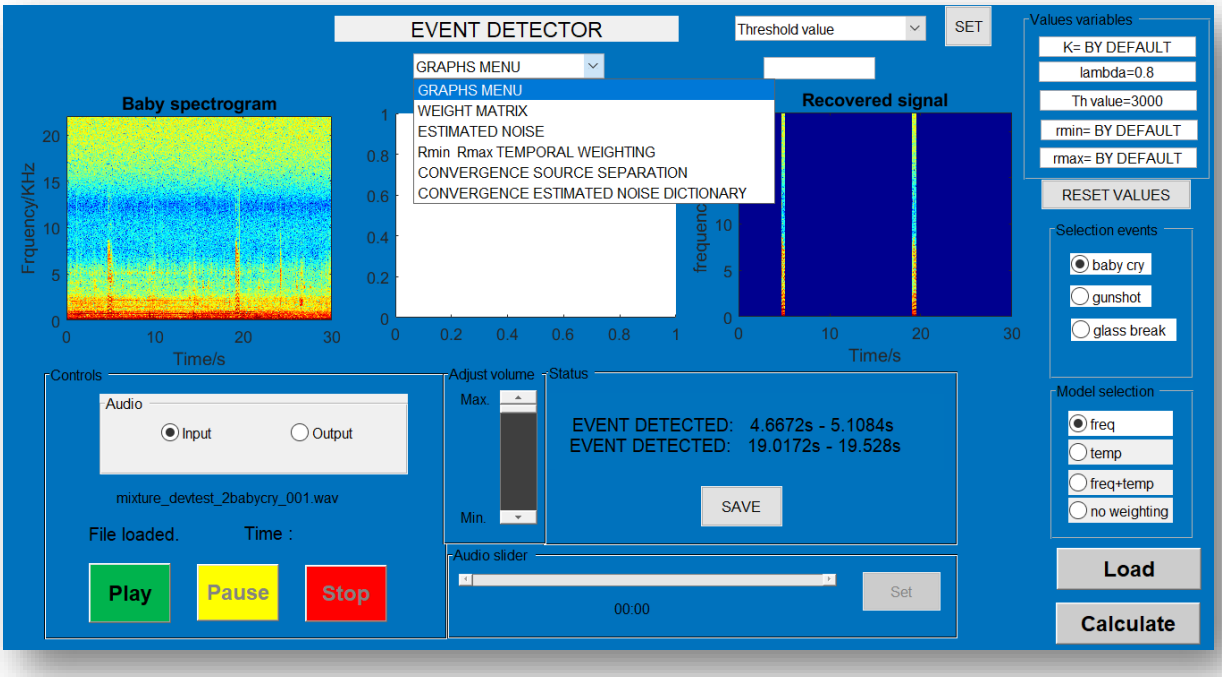

Figure 91. Interface screen when finish the processing of the input audio.

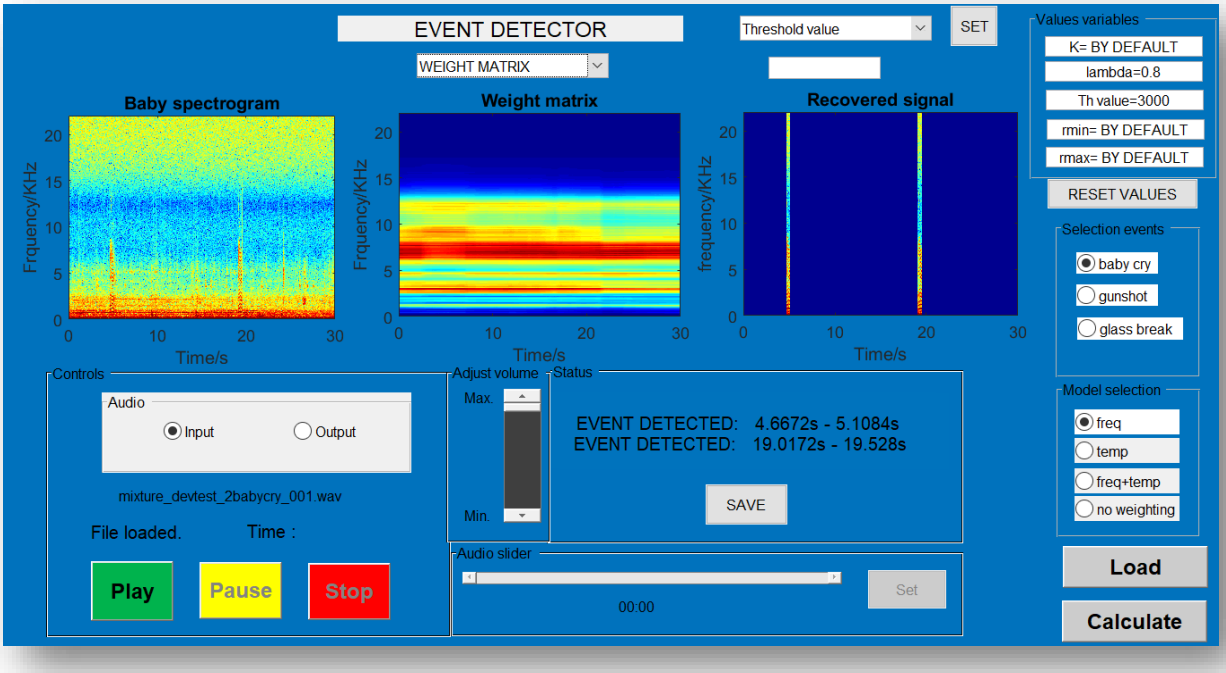

Figure 92. Interface screen when is selected the weight matrix graph from the graphs menu.

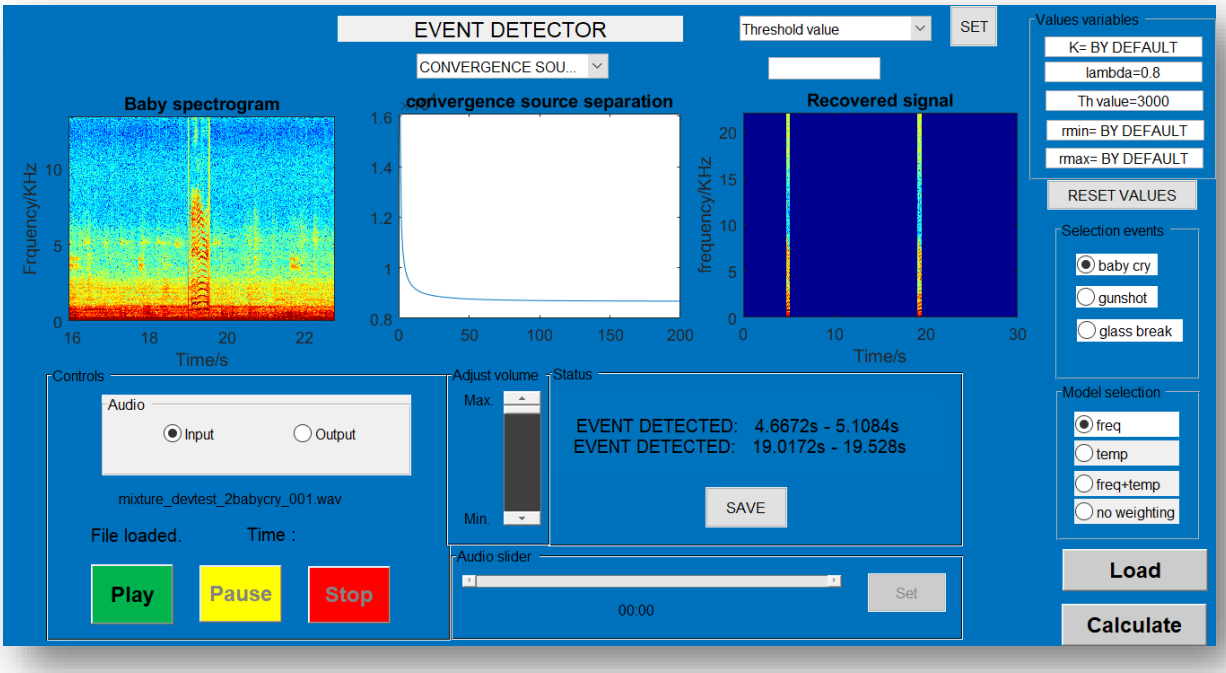

Figure 93. Interface screen when is selected the weight matrix graph from the graphs menu.

In the case that the user chose no weighting, and he select to show the weight matrix G in the graphs menu, the system will show an alert message advising to the user that he has not chosen weighting, figure 99. This will also happen if the user selects to show $rmin$ and $rmax$ and he did not select temporal or combined weighting, figure 100.

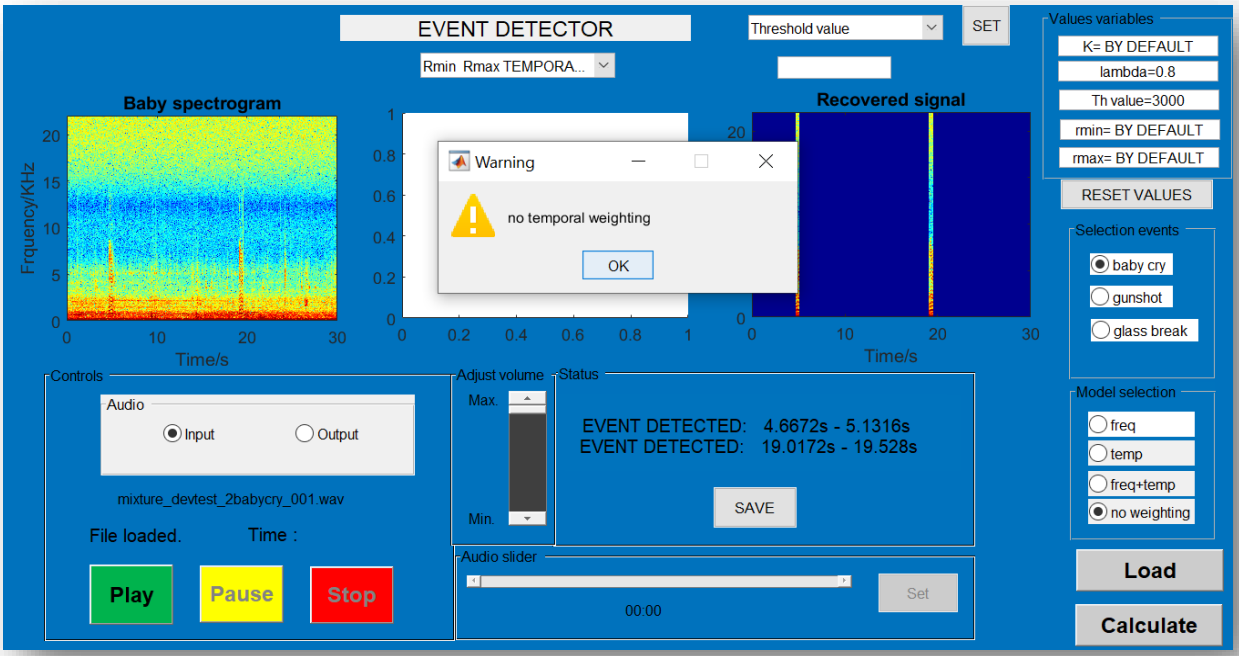

Figure 94. Warning from interface because the graph rmin and rmax is not available because was chosen no temporal weighting previously.

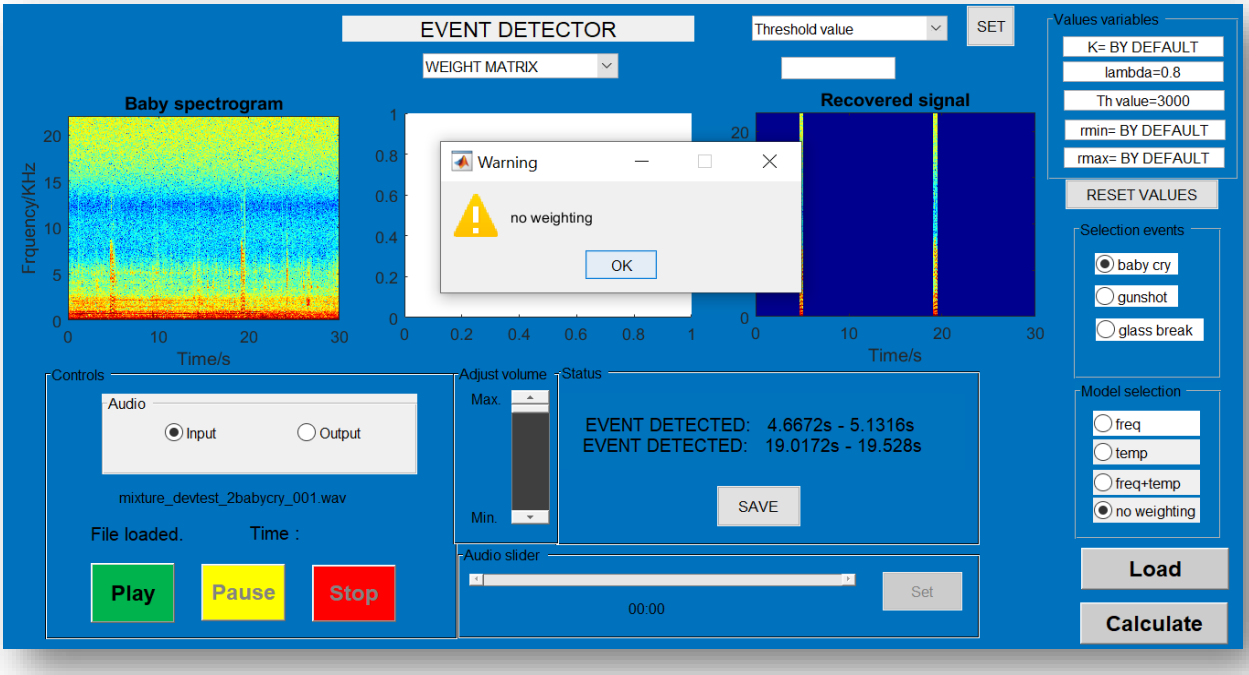

Figure 95. Warning from interface because the graph weight matrix is not available because was chosen no weighting previously.

Finally, if the user wishes, he can save the audio output, pressing the “SAVE” button.

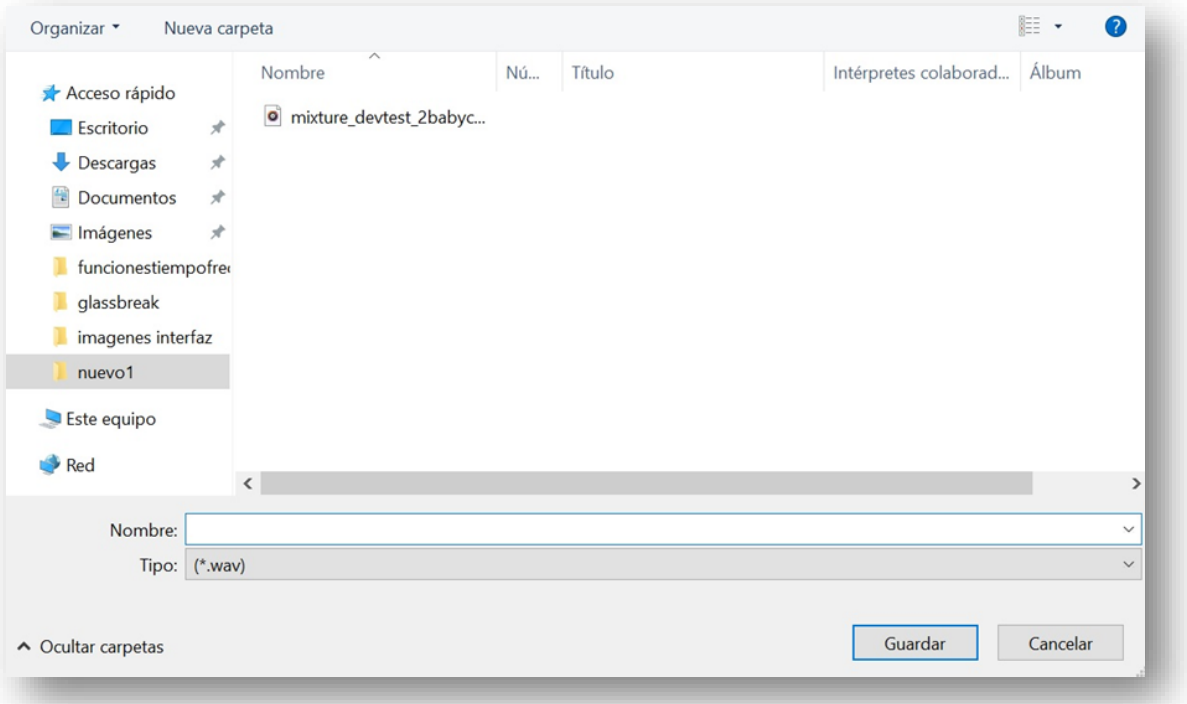

Figure 96. Interface screen when is pressed the button SAVE.

10.2 Minimization of Kullback-Leibler divergence in NMF

Cost function minimization, from [42], for Kullback-Leibler divergence in NMF:

$$\mathrm{DKL}\,(\mathrm{V},\mathrm{WH}) \;=\; \sum_{\mathrm{F}}\sum_{\mathrm{T}}\left(\mathrm{V_{FT}}\,\log\frac{\mathrm{V_{FT}}}{\mathrm{WH}|_{\mathrm{FT}}} - \mathrm{V_{FT}} + \mathrm{WH}|_{\mathrm{FT}}\right)$$

$$\frac{\partial\,\mathrm{DKL}\,(\mathrm{V},\mathrm{WH})}{\partial\,\mathrm{W_{ij}}} = \sum_{\mathrm{F}}\sum_{\mathrm{T}}\frac{\partial}{\partial\,\mathrm{w_{ij}}}\left(\mathrm{V_{FT}}\,\log\frac{\mathrm{V_{FT}}}{\mathrm{WH}|_{\mathrm{FT}}}\right) + \sum_{\mathrm{F}}\sum_{\mathrm{T}}\frac{\partial}{\partial\,\mathrm{W_{ij}}}\mathrm{WH}|_{\mathrm{FT}}$$

- Taking into account that the first term is:

$$\sum_{\mathrm{F}}\sum_{\mathrm{T}}\frac{\partial}{\partial\,\mathrm{W_{ij}}}\left(\mathrm{V_{FT}}\,\log\frac{\mathrm{V_{FT}}}{\mathrm{WH}|_{\mathrm{FT}}}\right) = \sum_{\mathrm{F}}\sum_{\mathrm{T}}\frac{\partial}{\partial\,\mathrm{W_{ij}}}\left(\log\mathrm{V_{FT}} - \log\mathrm{WH}|_{\mathrm{FT}}\right) \;=$$

$$\textstyle\sum_F \sum_T -\mathrm{V_{FT}}\frac{\partial}{\partial\,\mathrm{W_{ij}}}\;\log\mathrm{WH}|_{\mathrm{FT}} =$$

- Property applied: $\frac{\partial}{\partial\,\mathrm{X}}\log\left(\mathrm{f(x)}\right) = \frac{\partial\,\mathrm{f(x)}\,/\partial\,\mathrm{X}}{\mathrm{f(x)}}$

$$= \textstyle\sum_F \sum_T \frac{-\mathrm{V_{FT}}}{\mathrm{WH}|_{\mathrm{FT}}} * \frac{\partial\,\mathrm{WH}|_{\mathrm{FT}}}{\partial\,\mathrm{W_{ij}}} = \sum_F \sum_T \frac{-\mathrm{V_{FT}}}{\mathrm{W_{FK}\,H_{KT}}} * \frac{\partial\,(\mathrm{W_{FK}\,H_{KT}})}{\partial\,\mathrm{W_{ij}}} = \sum_N \frac{-\mathrm{V_{iT}}*\mathrm{H_{jT}}}{\mathrm{WH}|_{\mathrm{iT}}}$$

- And the second term of the sum is:

$$\sum_{\mathrm{F}}\sum_{\mathrm{T}}\frac{\partial}{\partial\,\mathrm{W_{ij}}}\mathrm{WH}|_{\mathrm{FT}} = \sum_{\mathrm{F}}\sum_{\mathrm{T}}\frac{\partial}{\partial\,\mathrm{W_{ij}}}\mathrm{W_{FK}\,H_{KT}} \;= \sum_{\mathrm{T}}\mathrm{H}|_{\mathrm{jT}}$$

- Now are joined the terms again:

$$\nabla_W \, DKL\,(V, WH) \;=\; \sum_T H|_{jT} - \sum_T \frac{V_{iT}\, H_{jT}}{WH|_{iT}}$$

$$\nabla_W \; DKL\,(V, WH) \;=\; 1 * \; H_{ij}^{Tr} \; - \frac{V_{ij}\, H_{ij}^{T}}{WH|_{ij}}$$

General equation form: $\theta \longrightarrow \theta \circ \frac{\nabla_W{}^{-} D(\theta)}{\nabla_W{}^{+} D(\theta)}$

- And is obtained W:

$$W \;=\; W \circ \frac{\frac{VH^{T}}{WH}}{1H^{T}}$$

In order to obtain H the process would be the same but minimizing H:

$$\frac{\partial\, DKL\,(V, WH)}{\partial\, H_{ij}} = \sum_F \sum_T \frac{\partial}{\partial\, H_{ij}} \left(V_{FT} \log \frac{V_{FT}}{WH|_{FT}} \right) + \sum_F \sum_T \frac{\partial}{\partial\, H_{ij}} WH|_{FT}$$

- And would be obtained H:

$$H \;=\; H \circ \frac{W^{T} \frac{V}{WH}}{W^{T} 1}$$

11. COMPUTATIONAL COST

To calculate the computational cost, a laptop with the following characteristics has been used:

- Processor: Intel Core i7
- RAM: 16 GB
- SSD: 512 GB

The times are calculated for one recording of 30 s, taking into account the time that the system takes to process the input spectrogram until the system generates a wav file with the events:

Type of weighting	Execution time (s)
Frequency weighting	35 s
Time weighting	33 s
Combined weighting	37 s
Without weighting	31 s

Table 5. Computational cost

12. REFERENCES

[1] Naoki Hashimot, Kazumasa Yamamoto and Seiichi Nakagawa. Speech recognition for mixed speech and music by NMF using various cost functions and noise adaptive training methods. Deparment of computer Science and Engineering, Toyohashi University of Technology, Japan, 2015.

[2] Anna maria Mesaros, Aleksandr Diment; Benjamin Elizalde, Toni Heittola; Emmanuel Vincent, et al. Sound event detection in the DCASE 2017 Challenge. IEEE/ACM Transactions on Audio, Speech and Language Processing, Institute of Electrical and Electronics Engineers, 2019, 27 (6), 992-1006.

[3] David M. Howard, James Angus. Acoustics and Psychoacoustics. Focal Press; 2 edition 12 Aug. 2000. ISBN: 978-0240516097.

[4] Meinard Müller. Fundamentals of Music Processing, Springer 2015. ISBN: 978-3-319-21944-8.

[5] The music Telegraph - https://m.themusictelegraph.com/553

[6] Robert C. Maher and Steven R. Shaw. Deciphering gunshot recordings. Digital Audio Signal Processing Laboratory, Deparment of Electrical and Computer Engineering, Montana State University, Bozeman, MT USA, 2008.

[7] Hyvärinen, J. Karhunen and E. Oja, Independent Component Analysis, John Wiley and Sons, Inc., New York, 2001.

[8] P. Cavalcanti, J. Scharcanski, L. E. Di Persia & D. H. Milone, Proceedings of the 33rd Annual International IEEE EMBS Conference (EMBC 2011), Boston, 2011.

[9] Norsalina Hassan and Dzati Athiar Ramli. A Comparative Study of Blind Source Separation for Bioacoustics Sounds based on FastICA, PCA and NMF. School of Electrical & Electronic Engineering, University Sains Malaysia, Nibong Tebal, 14300, Penang, Malaysia, 2018.

[10] S. I. Amari, S. C. Douglas, A. Cichocki and H. H. Yang Proceedings of the IEEE work shop on signal processing advances in wireless communications, Paris, 1997.

[11] A. Cichocki and S. I. Amari, Adaptive Blind Signal and Image Processing, John Wiley & Sons, Chichester, 2002.

[12] Daichi Kitamura. Blind source separation based on independent low-rank matrix analysis and its extensions. University of Tokyo, Japan, 2017.

[13] N. Murata, S. Ikeda and A. Ziehe, Neurocomputing 41, 2001.

[14] L. E. Di Persia, D. H. Milone and M. Yanagida, IEEE Transact. Audio Speech Lang. Process. 17, 299 ,2009.

[15] Lee, D. Seung and H. Learning the parts of objects by non-negative matrix factorization. Nature 401 1999, 788–791.

[16] Alexey Ozerov. Contributions in Audio Modeling for Solving Inverse Problems: Source Separation, Compression and Inpainting. Traitement du signal et de image [eess.SP]. Université Rennes 1, 2019.

[17] P. Smaragdis and J. C. Brown, Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, 2003.

[18] A. Ozerov and C. Fevotte, IEEE Transact. Audio Speech Lang. Process. 18, 550, 2013.

[19] Koichi Kitamura, Yoshiaki Bando, Katushoki Itoyama, Kazuyoshi Yoshii. Student's T multichannel nonnegative matrix factorization for blind source separation. Graduate School of Informatics, Kyoto University, Japan, 2007.

[20] Alexey Ozerov, Cédric Févotte, Emmanuel Vincent. An introduction to multichannel NMF for audio source separation. Audio Source Separation, Springer, 2018.

[21] Mehdi Khosrow-Pour. Information science and Technology. Information science Reference, 3 edition 2015. ISBN: 978-1-4666-5888-2.

[22] Sean Wood. Non-negative Matrix Descomposition Approaches to Frequency Domain Analysis of Music Audio Signals. “Département d’informatique et de recherche opérationnelle Université de Montréal”, 2009.

[23] Shoji Makino. Audio Source Separation. Springer 2018. ISBN: 978-3-319-73030-1.

[24] Fabian Theis, Andrzej Cichockki, Arie Yeredor and Michael Zibulevsky. Latent Variable Analysis and Signal Separation. Springer 2012. ISBN: 978-3-642-28550-9.

[25] Boian S. Alexandrov, Valentin G. Stanev, Velimir V. Vesselinov, Kim Rasmussen. Non-negative tensor decomposition with custom clustering for microphase separation of block copolymers. 2019, https://doi.org/10.1002/sam.11407

[26] Le Li and Yu-Jin Zhang. Non-negative matrix-set factorization in ICIG ’07: Proceedings of the Fourth International Conference on Image and Graphics, Washington, DC, USA, 2007, pp. 564–569, IEEE Computer Society.

[27] W. Chen, X. Huang, B. Fan, Q. Wang and B. Wang, "Kernel nonnegative matrix factorization with RBF kernel function for face recognition," *2017 International Conference on Machine Learning and Cybernetics (ICMLC)*, Ningbo, 2017, pp. 285-289.

[28] Patrik O.Hoyer. Non-negative Matrix Factorization with Sparseness Constraints. Journal of Marching Learning Research 5 2004, 1457-1469.

[29] Jiho Yoo and Seugjin Choi. Non-negative Matrix Factorization with Orthogonality Constraints. Department of Computer Science Pohang University of Science and Techonology. Pohang 790-784, Korea.

[30] Jia Z, Zhang X, Guan N, Bo X, Barnes MR, Luo Z . Gene Ranking of RNA-Seq Data via Discriminant Non-Negative Matrix Factorization. PLoS ONE 10(9): e0137782, 2015. https://doi.org/10.1371/journal.pone.0137782.

[31] Qiu Xiao, Jiawei Luo, Cheng Liang, Jie Cai and Pingjian Ding. A graph regularized non-negative matrix factorization method for identifying microRNA-disease associations. *Bioinformatics*, Volume 34, Issue 2, 15 January 2018, Pages 239–248, https://doi.org/10.1093/bioinformatics/btx545

[32] Y. Wang and Y. Zhang, "Nonnegative Matrix Factorization: A Comprehensive Review," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, June 2013, pp. 1336-1353.

.

[33] Paul D. O' Grady and Barak A. Pearlmutter. CONVOLUTIVE NON-NEGATIVE MATRIX FACTORISATION WITH SPARSENESS CONSTRAINT. Hamilton Institute, National University of Ireland, Maynooth.

[34] N. Del Buono and G. Pio. Non-Negative Matrix Tri-Factorization for co-clustering: An analysis of the block matrix. "Dipartimento di Matematica, Universita degli Studi di Bari Aldo Moro, Bari Italy".

[35] Analytics Vidhya-https://www.analyticsvidhya.com/blog/2018/12/guide-convolutional-neural-network-cnn/

[36] CS231n in convolutional Neural Netowrks- https://cs231n.github.io/convolutional-networks/

[37] Code to Light - https://codetolight.wordpress.com/2017/11/29/getting-started-with-pytorch-for-deep-learning-part-3-neural-network-basics/

[38] Iuliana Tabian and Hailing Fu and Zahra Sharif Khodaei. A Convolutional Neural Network for Impact Detection and Characterization of Complex Composite Structures. Basel, Switzerland, 2019.

[39] Qing Zhou, Zuren Feng and Emmanouil Benetos. Adaptive Noise Reduction for Sound Event Detection Using Subband-Weighted NMF. Sensors (Basel, Switzerland) vol. 19, 14 3206, 20 Jul. 2019.

[40] Meng. Sun, Yinan. Li, Jort. F. Gemmeke and Xiongwei. Zhang. Speech Enhancement Under Low SNR Conditions Via Noise Estimation Using Sparse and Low-Rank NMF with Kullback–Leibler Divergence. IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 7, pp. 1233-1242, July 2015.

[41] Zu-Ren Feng, Qing Zhou, Jun Zhang, Ping Jiang and Xue-Wen Yang, X. A target guided subband filter for acoustic event detection in noisy environments using wavelet packets. IEEE/ACM Trans. Audio Speech Lang. Process. 2015, 23, 1230–1241.

[42] Juan José Burred, Detailed derivation of multiplicative update rules for NMF. Paris, France, 2014.