AVERAGED BINARY SPARSOGRAM FOR WILDLIFE INTRUDER DETECTION

Lăcrimioara GRAMA Corneliu RUSU
Bases of Electronics Department, Technical University of Cluj-Napoca, Signal Processing Group
G. Bari□iu 26-28, Cluj-Napoca, {Lacrimioara.Grama, Corneliu.Rusu}@bel.utcluj.ro

Abstract: This paper presents a method for wildlife intruder detection. For the feature extraction step, we have used the averaged binary sparsogram, while for the classification step we have used artificial neural networks. Two frameworks are discussed: in the first framework, four sound classes are considered (birds, chainsaws, tractors, speech), and in the second framework, five sound classes are considered (birds, chainsaws, tractors, speech, gunshots). High overall accuracy classification rates were obtained, for both considered frameworks: 98.8% for the first framework, and 97.4% for the second one. For each framework, two scenarios are tested: first the classification is performed in one step, and second the classification is performed in two steps. A small improvement in the overall accuracy classification rates is observed when the classification is performed in two steps.

Keywords: *intruder detection, short time Fourier transform, sparsogram, artificial neural network, pattern recognition.*

I. INTRODUCTION

Many acoustic and audio signals are locally Fourier sparse [1]. This property is defined by using the Short-Time Fourier Transform (STFT)

$$X(m,\omega) = \sum_{n=-\infty}^{\infty} x(n)w(n-m)e^{-j\omega n} \qquad (1)$$

and it is the Fourier spectrum of the signal x localized around time sample m by the window w.

A signal is locally Fourier sparse if $X(m, \omega) \approx 0$ for most m and ω. We can say that signals are locally Fourier sparse in the sense that at each point in time the signals are well-approximated by a few local sinusoids of constant frequency [1].

In many applications it is easier to use the spectrogram, i.e. the squared magnitude of the STFT $|X(m, \omega)|^2$, to represent the spectrum of a signal. Indeed, the spectrogram provides a visual representation of frequency characteristic of signals. Unfortunately all the frequency components which are present in a given signal will appear on the spectrogram. Thus the spectrogram is sometimes really difficult to understand. The spectrogram as well as the spectrum of a signal, is composed of spectral components corresponding to background noise or of some low-magnitude tones. These are not of major importance for determining the main characteristics of signals. Consequently spectrogram has two main drawbacks: it is noise sensitive and it may provide irrelevant information for a specific application [2].

The recently introduced sparsogram may solve all the issues and limitations of the spectrogram mentioned above. The sparsogram is a time-frequency representation of a signal that displays the dominant frequencies of the signal. Based on the sparsogram of a signal, one can extract the main frequency characteristics that define the signal [1]. Useless information is not anymore present. Consequently the sparsogram may provide the relevant information about a signal. This information can actually help to discriminate between different classes of signals. Such situation may appear in applications like wildlife intruder detection [3].

Previously we have proposed different solutions for the problem of wildlife intruder detection [4]: a solution that uses TESPAR (Time Encoded Signal Processing and Recognition) as a method for sound encoding and classification and two standard sound classification methods, which proved to be more robust. Both of them were using as features the Mel-Frequency Cepstral Coefficients, while for training and classification were used Gaussian Mixture Models and Support Vector Machines. However for wildlife applications, simple and low-power implementations are desired.

Recently, we have used another simpler approach for our wildlife intruder detection system implementation. We extract the main frequency characteristic of the sounds and later on we differentiate into subclasses corresponding to intruders and non-intruders. The solution was to utilize the sparsogram. The results were quite promising, though the sparsogram has been implemented using a rough threshold applied on spectrogram [2]. Besides, the extraction of relevant frequencies has been done manually, and this has to be avoided for a wildlife application.

In this paper the averaged binary sparsogram implementation is proposed and we show that this approach may be suitable for applications such as wildlife intruder detection. The averaged binary sparsogram is computed as follows: first we perform an infinite clipping of the magnitudes of the spectrogram, and then we average the results over certain number of frames. In this way we count for every frequency the number of their relevant appearances in the spectrum. If the signal spectrum contains

Manuscript received July 28, 2015; revised August 31, 2015

a frequency for a long time, then the averaged binary sparsogram will have a large number associated to that frequency. If the signal spectrum seldom contains a frequency, then the averaged binary sparsogram will have a small number associated to that frequency. Consequently the averaged binary sparsogram will emphasize the main components of the signal spectrum. This will help us discriminate between different classes of signals.

The paper is organized as follows: in Section II, the averaged binary sparsogram is introduced and the setting of parameters is discussed. Then, in Section III, the databases used for wildlife intruder detection are presented and the architecture of the proposed system is described in Section IV. Simulation results are presented for two different frameworks, using four scenarios, in Section V. Finally, conclusions are dragged.

II. THE AVERAGED BINARY SPARSOGRAM

A signal is perfectly represented by all its Fourier coefficients. It may happen that we wish to use only few coefficients to represent the signal or its spectrum. Such situations occur when the signal has to be denoised or it has a very narrow spectrum. The practical choice is to select the largest Fourier coefficients, in magnitude.

Many acoustic and audio signals are locally Fourier sparse, in the sense that the signals are well-approximated locally by a few sinusoids of constant frequency. The presence of these few local sinusoids of constant frequency may provide a kind of signature of the local behavior of the signal. To have an overall signature of the signal, we have to count somehow all the presences of the local frequencies.

The proposed averaged binary sparsogram of a signal is computed as follows:

1. the signal is divided in frames;
2. for every frame, the DFT of the signal (included by the frame) is computed;
3. using a threshold, the magnitude of the resulting DFT is infinite clipped;
4. for every index of the DFT, the infinite clipped magnitudes (which have binary values) are collected.

The resulting sum with respect to the index of DFT is the averaged binary sparsogram. Figure 1 illustrates the pseudo code for the above mentioned steps in order to obtain the averaged binary sparsogram.

As one can see, there are two parameters in computing the averaged binary sparsogram: the length of the frame and the threshold for the DFT magnitudes. This means only one extra parameter in comparison to the spectrogram. The use of a window when computing the DFT is optional.

As example, we consider the case of a noisy multitone signal. The signal is composed of three normalized frequencies: f_1 = 1/20, f_2 = 3/20 and f_3 = 5/20, every tone alternates one by one. The magnitude of a single tone is one, and the signal has a noisy component given by a Gaussian noise of mean zero and variance one.

Figure 2 presents the spectrogram and the associated averaged binary sparsogram using 128 frames. The length of every frame was 256 and the window was rectangular for both the spectrogram and the associated averaged binary sparsogram. The threshold for computing the averaged binary sparsogram was set 10% from the maximum of the magnitude in every frame.

As one can see, the spectrogram is noisy and contains a lot of information which is irrelevant for some applications.

`Given:`	N – number of samples;
	$x(n), \quad n = \overline{0, N-1}$ – the signal;
	L – number of frames;
`Compute:`	$M = N / L$ – frame length;
`Set:`	$A(k) = 0, \quad k = \overline{0, M-1}$
`for` $i = \overline{0, L-1}$	$x_i(n) = x(n + iM), \quad n = \overline{0, M-1}$;
	$X_i(k) = DFT\{x_i(n)\}, \quad k = \overline{0, M-1}$;
	$Th = 0.1 \cdot \max \lvert X_i(k) \rvert$ – threshold;
	$A_i(k) = \begin{cases} 0, & \lvert X_i(k) \rvert < Th, \\ 1, & \lvert X_i(k) \rvert \geq Th; \end{cases}$
	$A(k) = A(k) + A_i(k)$;
`end;`	
`Output:`	$A(k)$ – averaged binary sparsogram.

Figure 1. Pseudo code for averaged binary sparsogram.

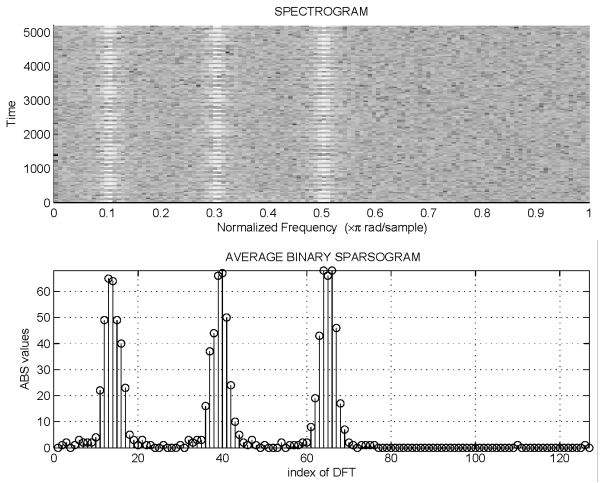

Figure 2. Example of a spectrogram and of associated averaged binary sparsogram.

On the other hand, the sparsogram clear identifies the three frequencies which alternate within the signal and ignore all other frequencies from the spectrum of the signal. This clearly helps us discriminate between different classes of signals. We will further use this property for wildlife intruder detection.

III. DATABASES USED FOR WILDLIFE INTRUDER DETECTION

In any intruder detection system we have to define from the very beginning who are the intruders and the non-intruders. For our research several databases have been used. We have focused in a collection of five types of audio signals. We have performed simulations on signals available on SPG (Signal Processing Group) Sound Database [5], on some signals available online [6-8] and on some speech sounds [9]. The last set of signals belongs to students from the Technical University of Cluj-Napoca. They were asked to

record themselves when uttering different sentences.

- Database 1 (non-intruder) – contains a set of 300 different audio samples originated from 60 different types of Romanian birds [6, 7]; all the recordings are performed in different forests from Romania;
- Database 2 (intruder) – contains a set of 90 different audio samples originated from 10 different types of chainsaws [5];
- Database 3 (intruder) – contains a set of 100 different audio samples originated from 13 different types of tractors [5];
- Database 4 (intruder) – contains a set of 150 speech sounds originated from 50 different people [9]; the recordings are quite short, 2-5 seconds each;
- Database 5 (intruder) – contains a set of 60 different audio samples originated from 25 different types of gunshots [7, 8]; these recordings are quite short, varying from 0.2 to maximum 3.2 seconds.

The signals corresponding to the chainsaws and tractors were recorded using a digital voice recorder. The recordings have been achieved outside, thus they are not studio recordings. This means they are subject of some additive noise from surroundings. Also the signals corresponding to the birds, gunshots and people are not studio recordings.

IV. ARCHITECTURE OF THE PROPOSED INTRUDER DETECTION SYSTEM

The intruder detection system proposed in this work can be divided into four modules: pre-processing, segmentation, feature extraction and pattern recognition, as shown in Figure 3.

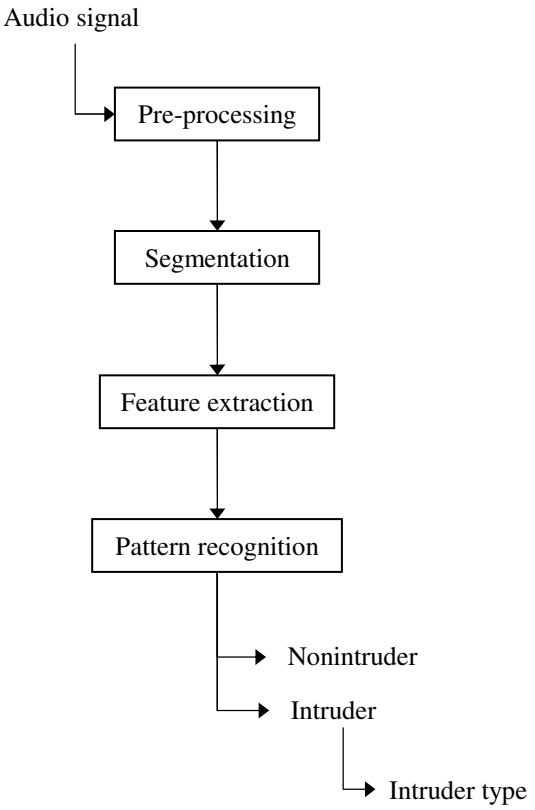

Figure 3. Architecture of the intruder detection system.

Pre-processing: The input acoustic signal is first pre-processed: it is resampled at 16 kHz frequency and saved as 16-bit mono format.

Segmentation: After pre-processing, the audio signal is divided into 256 samples size frames without overlapping.

Feature extraction: Feature extraction is a process where an audio signal is characterized into a compact numerical representation. A DFT algorithm is applied on each frame and the magnitude of the DFT is evaluated, representing the intensity of the sound during that frame at different frequencies. The spectral components that are below a given threshold are considered to be zero and the ones that are above the threshold are considered to be one. The threshold is evaluated as the 10% of the maximum magnitude over the entire signal. Only half of the coefficients from each frame are retained (the first 128). The k^{th} coefficients from each frame are added together and, as the last feature extraction step, the 128^{th} coefficients are normalized by the number of frames of the considered signal. Thus the feature set of the proposed system contains 128 features values.

Pattern recognition: The problem of intruder detection can be seen as a pattern recognition problem. The goal of pattern recognition is to classify objects of interests into a number of classes. The objects of interest are generically called patterns and in this case, they are features sets extracted from an audio signal using the previously described steps. Since the classification procedure is applied on extracted features, it can be also referred to as feature matching [10].

For audio signals classification we used a non-parametric method: Artificial Neural Network (ANN). For all frameworks considered we have constructed feed-forward neural networks using the MATLab Neural Network toolbox, with one hidden layer and one output layer. Hyperbolic tangent sigmoid transfer function (*tansig*) was used as the activation function, for the neurons in both layers. Supervised learning was used with scaled conjugate gradient back-propagation (*trainscg*), for the training algorithm.

In order to improve the neural networks generalization, for each experiment we have trained multiple neural networks and averaged their outputs; their mean squared errors were compared to the means squared error of their average [11]. First, the dataset was loaded and divided into a train and test data set, as shown in Table 1.

Table 1. Sound classes and number of samples in the database used for pattern recognition

Sound class	Train	Test	Total
Bird	210	90	300
Chainsaw	63	27	90
Tractor	70	30	100
Speech	105	45	150
Total framework 1	***448***	***192***	***640***
Gunshot	42	18	60
Total framework 2	***490***	***210***	***700***

Then, 20 neural networks were trained. Next, each network was tested on the second dataset (test set) with both individual performances and the performance for the average output calculated. All neural networks were trained with features corresponding to the training set and were tested with features corresponding to test data set.

V. SIMULATION RESULTS

The implementation of the proposed intruder detection system was carried out in the MATLab integrated development environment, making good use of the built-in functions available in the Signal Processing Toolbox and Neural Networks Toolbox. A series of custom functions and scripts were also necessary, especially for the system development part of the process.

Two frameworks are discussed: in the first framework, four sound classes are considered (birds, chainsaws, tractors, speech), and in the second framework, five sound classes are considered (birds, chainsaws, tractors, speech, gunshots). For each framework, two scenarios are tested: first the classification is performed in one step and, second, the classification is performed in two steps, in order to see if we can increase the overall accuracy classification rates.

Framework 1 *– Scenario 1*

In the first scenario, the following sound types were tested: birds, chainsaws, tractors, and speech. A feed-forward neural network with 22 hidden neurons and 4 outputs (each representing a different sound class) was used for classification.

Table 2 presents the individual performances in the training and test phase, in terms of the mean squared error, for each of the 20 networks.

Table 2. Neural network performance for Scenario 1.

Network number	Train	Test
1	0.0087	0.0272
2	0.0056	0.0212
3	0.0073	0.0303
4	0.0072	0.0246
5	0.0051	0.0243
6	0.0064	0.0291
7	0.0085	0.0221
8	0.0067	0.0207
9	0.0057	0.0210
10	0.0072	0.0227
11	0.0068	0.0249
12	0.0070	0.0231
13	0.0058	0.0206
14	0.0065	0.0254
15	0.0072	0.0260
16	0.0062	0.0275
17	0.0051	0.0241
18	0.0052	0.0196
19	0.0084	0.0272
20	0.0064	0.0248

For the results presented in Table 2, the mean squared error for the average output in the training phase is 0.0055, and in the test phase is 0.0202. The mean squared error for the average output is lower than most of the individual performances. It is likely to generalize better to additional new data.

From Figure 4 it can be noticed that the overall performance is 97.5%, and, most importantly, no bird (non-intruder) is detected to be an intruder or vice versa.

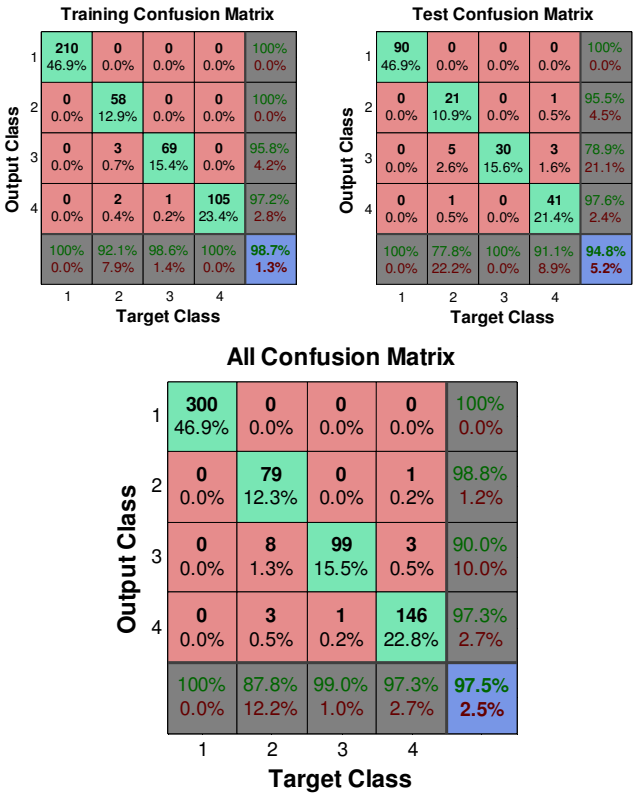

Figure 4. Training, test, and all confusion matrix (1 – bird (non-intruder), 2 – chainsaw (intruder), 3 – tractor (intruder), 4 – speech (intruder)).

Framework 1 *– Scenario 2*

In order to try to improve the performance, another scenario was tested. For the second scenario, the classification was performed in two steps. First, the audio signals were classified as intruders and non-intruders (using a feed-forward neural network with 16 hidden neurons and 2 outputs) and with another neural network (feed-forward with 19 hidden neurons and 3 outputs) the intruders were further classified.

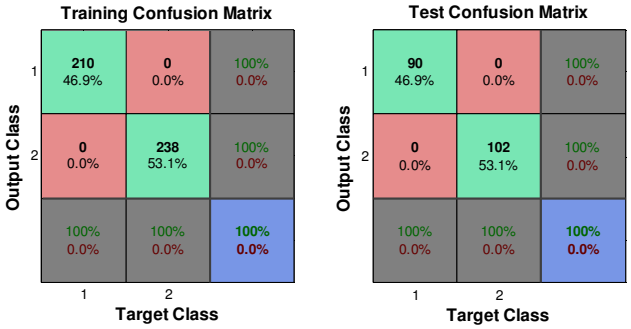

Figure 5. Training and test confusion matrix (1 – non-intruder, 2 – intruder).

For the results illustrated in Figure 5, the mean squared error for the average output in the training phase is 3.0260e-005, and in the test phase is 5.9576e-011.

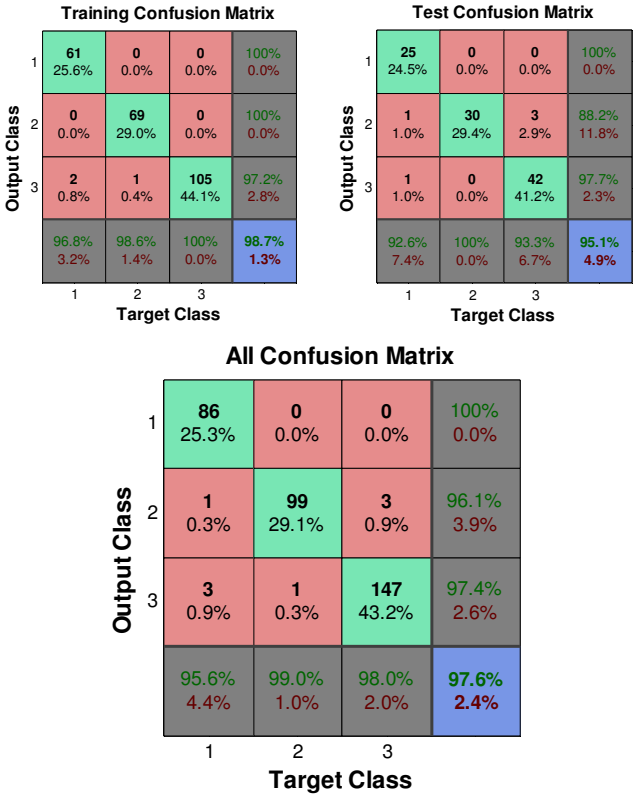

Figure 6. Training, test, and all confusion matrix (intruder classification: 1 – chainsaw, 3 – tractor, 4 – speech).

For the results illustrated in Figure 6, the mean squared error for the average output in the training phase is 0.0064, and in the test phase is 0.0254.

From Figures 5 and 6 it can be noticed that, by performing the classification in two steps, there is an improvement from 98.7% to 99.3% in the training phase, from 94.8% to 97.4% in the test phase, which means an overall improvement from 97.5% to 98.8%.

Framework 2

Scenario 3. In the third scenario the following sound types were tested: birds, chainsaws, tractors, speech, and gunshots. A feed-forward neural network with 22 hidden neurons and 5 outputs (each representing a different sound class) was used for classification.

Table 3 illustrates the individual performances in the training and test phase, in terms of the mean squared error, for each of the 20 networks.

Table 3. Neural network performance for Scenario 3.

Network number	Train	Test
1	0.0101	0.0258
2	0.0144	0.0283
3	0.0087	0.0223
4	0.0137	0.0314
5	0.0096	0.0245
6	0.0161	0.0267
7	0.0392	0.0556
8	0.0119	0.0270
9	0.0095	0.0201
10	0.0098	0.0233
11	0.0095	0.0301
12	0.0125	0.0314
13	0.0102	0.0258
14	0.0111	0.0325
15	0.0094	0.0257
16	0.0112	0.0389
17	0.0078	0.0203
18	0.0137	0.0265
19	0.0082	0.0219
20	0.0131	0.0266

For the results presented in Table 3, the mean squared error for the average output in the training phase is 0.0072, and in the test phase is 0.0172. As in the other cases, the mean squared error for the average output is likely to be lower than most of the individual performances.

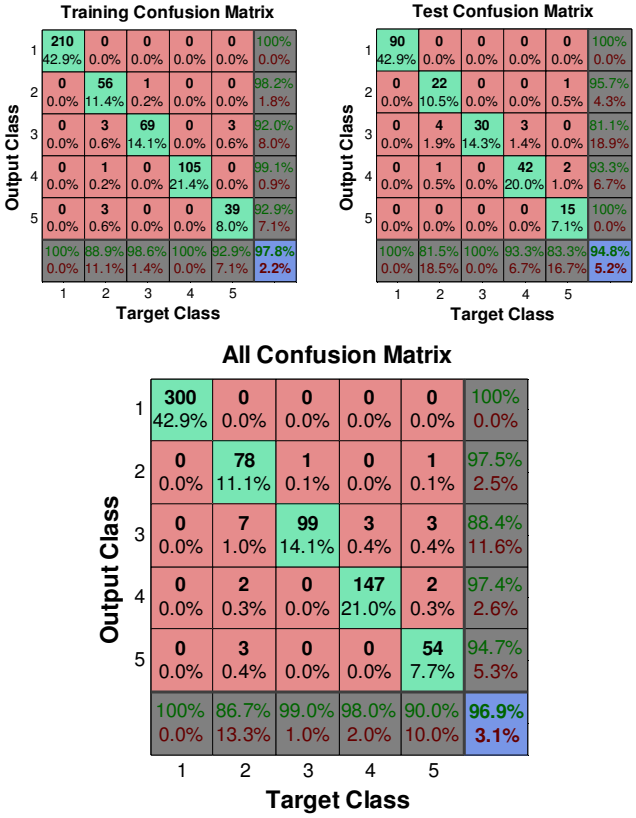

Figure 7. Training, test, and all confusion matrix (1 – bird (non-intruder), 2 – chainsaw (intruder), 3 – tractor (intruder), 4 – speech (intruder), 5 – gunshot (intruder)).

As in the first and second scenario, from Figure 7 it can be noticed that no bird (non-intruder) is detected to be an intruder or vice versa. The overall performance is 96.9%.

Scenario 4. In order to try to improve the performance, another scenario was tested, as in the case of the first framework. For the fourth scenario, the classification was performed in two steps. First, the audio signals were classified as intruder and non-intruder (using a feed-forward

neural network with 16 hidden neurons and 2 outputs) and with another NN (using a feed-forward neural network with 19 hidden neurons and 4 outputs) the intruders were further classified.

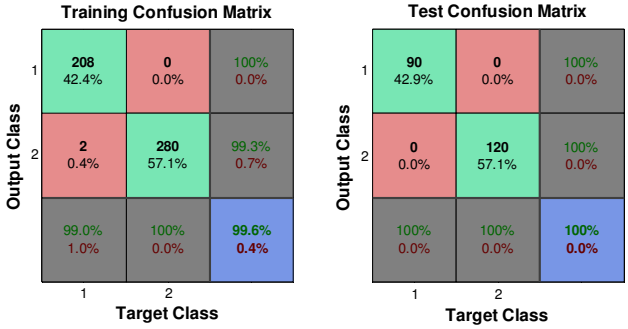

Figure 8. Training and test confusion matrix (1 – non-intruder, 2 – intruder).

For the results illustrated in Figure 8, the mean squared error for the average output in the training phase is 0.0034, and in the test phase is 2.4728e-004.

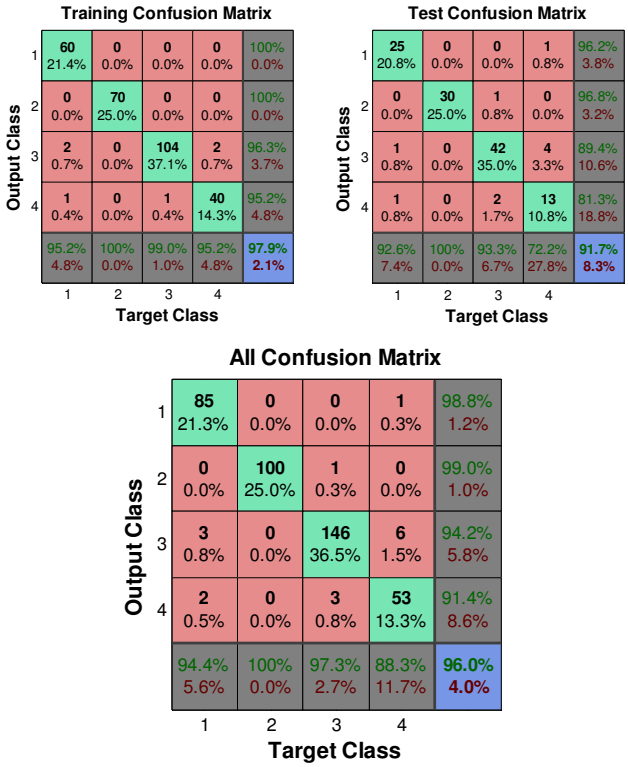

Figure 9. Training, test, and all confusion matrix (intruder classification: 1 – chainsaw, 3 – tractor, 4 – speech, 5 – gunshot).

For the results illustrated in Figure 9, the mean squared error for the average output in the training phase is 0.0098, and in the test phase is 0.0404.

From Figures 8 and 9 it can be noticed that, by performing the classification in two steps, there is an improvement from 97.8% to 98.4% in the training phase (but 2 birds are classified as intruders), from 94.8% to 95.2% in the test phase, which means a small overall improvement from 96.9% to 97.4%.

VI. CONCLUSIONS

In this paper, the averaged binary sparsogram is presented and we show that this approach may be suitable for applications as wildlife intruder detection. In the classification phase, artificial neural networks have been implemented and used for testing.

Two frameworks are considered, and for each framework two scenarios are tested: in the first and third scenario, the classification is performed in one step, while for the second and fourth scenario, the classification is performed in two steps.

In the first framework, four different classes of signals are classified (birds, chainsaws, tractors and speech). For the first scenario, the overall correct classification rate is 97.5%, while for the second scenario, the rate is 98.8%. The most important thing is that in both scenarios no intruder is classified as non-intruder and vice versa.

In the second framework, another sound class is added, the one corresponding to gunshots. For the third scenario, the overall correct classification rate is 96.9%, while for the fourth scenario, it is 97.4%.

The experimental results prove that the above presented method can be used for wildlife intruder detection, with overall high correct classification rates.

ACKNOWLEDGMENT

This work was supported by the Human Resources Development Programme POSDRU/159/1.5/S/137516 - PARTING financed by the European Social Fund and by the Romanian Government.

REFERENCES

[1] A. C. Gilbert, M. J. Strauss, and J. A. Tropp, "A tutorial on fast Fourier sampling," IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 57–66, 2008.
[2] R. G. Rosu and C. Rusu, "A sparsogram implementation for wildlife intruder detection," in *Proceedings ISSCS 2013*, Iasi, Romania, July 2013, pp. 1–4.
[3] C. Rusu, "A sparsogram coding procedure for wildlife intruder detection," in Proceedings ISCCSP 2014, Athens, May 2014, pp. 1–4.
[4] Ghiurcau, M.V., Rusu, C., Bilcu, R.C., Astola, J.: Audio based solutions for detecting intruders in wild areas. Signal Processing 92(3), pp. 829–840 (2012)
[5] L. Todor, V. Zoicas, L. Grama and C. Rusu, "The SPG (Signal Processing Group) sound database," Novice Insights, vol. 15, no. 1, pp. 62–65, 2014.
[6] Xeno-canto, "Bird sounds from Romania". [Online]. Available: http://www.xeno-canto.org/.
[7] FreeSFX, "Bird and gunshot sounds". [Online]. Available: http://www.freesfx.co.uk/sfx/.
[8] SoundBible, "Free sound clips, sound bites, and sound effects". [Online]. Available: http: //soundbible.com/.
[9] E. Lupu, P.G. Pop - An overview of biometrics, Acta Tehnica Napocensis, Electronics and Telecomunications, 2006, Vol.47, pp.42-56, ISSN 1221-6542.
[10] C. Chhabra, A. Kumar, "Intrusion Detection using MFCC, VQA and LBG Algorithm," IJSRD - International Journal for Scientific Research & Development, vol. 2, issue 05, pp. 513-517, 2014.
[11] C. M. Bishop, Pattern Recognition and Machine Learning, 8th ed., ISBN-10: 0-387-31073-8, Springer 2009.