Design of IP Core for Sound Classification in Wireless Sensor Networks

Ilan S. Correa, Lilian C. Freitas, Adalbery Castro, João Crisóstomo Costa and Aldebaro Klautau
LEA/LASSE, Federal University of Pará - UFPA
CP 8619 - Rua Augusto Correa 01 - CEP 66075-110 - Bélem - Pará - Brazil
Web: www.lea.ufpa.br; www.lasse.ufpa.br. E-mails: {ilan, liliancf, adalbery, jwey, aldebaro}@ufpa.br

***Abstract*—This paper shows the current stage of development of an IP Core for sound classification in wireless sensor networks using Mel-frequency cepstrum (MFCC) and support vector machine (SVM) algorithms. This way a sensor node can do the job by itself instead of sensing to another node or a computer, this represents a great issue for power saving. The IP Core implementation was validated in software doing bird identification.**

1. Introduction

The analysis of sounds produced by animals, known as bioacoustics [1], enables identifying and monitoring species, estimates biodiversity of certain place, besides facilitating its study. However the task of capturing the animals sounds must not meddle (whether happens must be minimum) on its habits. An alternative to bypass such limitations is using sensor nodes equipped with a microcontroller and allowing power saving. For instance, such sensor nodes could capture sound of a given animal and assess the region's biodiversity [2], [3].

Howsoever to deal with audio data, the sensor node needs computational power to perform the job at a rate that depends of the application. Application such audio processing requires a lot of CPU time and analog-to-digital converters (A/D) with high resolution, this issue normally goes against common sensor nodes platforms available, such as Mica platform [4].

Therefore this works presents the current stage of development of an IP Core to be used together with a sensor node. This IP Core is currently in development, and is being designed to perform animal sound classification in its habitat in real time.

Artificial intelligence techniques are used to perform the task of sound classification. Basically two techniques form the IP Core kernel the Mel-frequency cepstrum coefficients (MFCC) [5] and Support vector machine (SVM) [6], which are popular pattern recognition techniques. The classification process and the mentioned techniques are explained in Section 2.

This work shows the tests results accomplished in Matlab to build the MFCC and SVM algorithm models that will be implemented in VHDL. Also are shown the preliminary test performed in software and some synthesis results that are already available. The tests presented use a database of bird sounds provided by Montana University [7].

The remainder of this paper is organized as follows: Section 2 describes the process and techniques used for sound classification; Section 3 describes the database used with Matlab to validate the algorithm; Section 4 shows the results and Section 5 shows some conclusions and next tasks.

2. Classification Process

The pattern classification systems are typically divided in two stages: features extraction (front end) and classification (back end), as described in [8] and shows in the Fig. 1. The first block converts the acoustic signal into a set of appropriate parameters for the classifier. The second block makes the identification of the type of sound (class) based on parameters extracted.

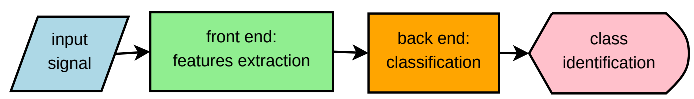

Fig. 1. Characterization of the classification process.

In this work was evaluated the performance of the front end Mel-frequency cepstrum (MFCC) and the back end support vector machine (SVM). These techniques are described in the next subsection.

2.1. Front end: MFCC

The classification process can be divided in two stages, the first one is the extraction of the Mel-frequency cepstrum coefficients (MFCC) [5]. These coefficients are used in the SVM classifier as parameter for bird species classification. A great advantage of using MFCC parameterization is that it was designed to maintain characteristics of human sound perception, this is achieved because frequency bands are equally spaced on mel scale this frequency warping can allow better sound representation; and due to the ability to represent the speech amplitude spectrum in a compact form, for example, in Fig. 2. Another benefit of MFCC is that they can be used with both periodic and non-periodic signal as described in [3].

The procedure by which the mel-frequency cepstral coefficients are obtained consists of several steps, that could be performed in several ways, as evaluated in [5]. A straightforward approach to obtain these coefficients of a given sampled signal is: calculate the discrete fourier transform, take the magnitude of the resulting signal, take the logarithm of this magnitude and finally take the discrete cosine transform. This procedure is illustrated in Fig. 3.

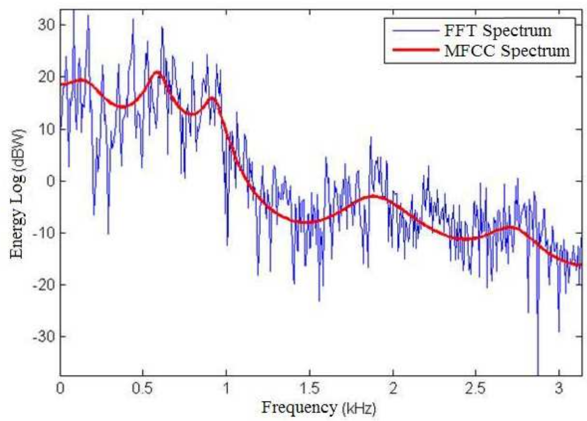

Fig. 2. MFCC compacts FFT informations.

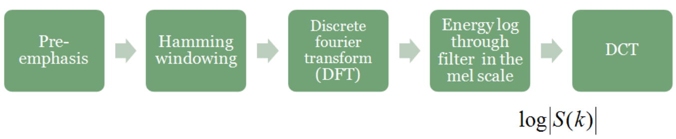

Fig. 3. Algorithm Block diagram to obtain the MFCC.

2.2. Back end: SVM Classifier

Support vector machine is part of a class of learning algorithms based on the statistical learning theory, which implements the principle of the structural risk minimization [6]. The basic idea of SVM is to map the input space into a feature space. This mapping can be done linearly or not, according to the kernel function used for the mapping. In the feature space, the SVM builds optimal hyperplanes to separate classes while minimizing the classification error. The optimal hyperplane can be written as a combination of a few points in the feature space, called the support vectors of the optimal hyperplane.

In this work the linear kernel was adopted, due to easy implementation and good performance, but in the literature, various possibilities for SVM kernels are presented such as the polynomial kernel, the radial basis network and the two-layer perceptron [6].

The SVM (and other kernel methods) can be characterized as an estimation function f that minimizes

$$\frac{1}{N}\sum_{n=1}^{N} L(f(\mathbf{x}_n), y_n) + \lambda ||f||^2_{\mathcal{H}_\mathcal{K}}, \tag{1}$$

where $\mathcal{H}_\mathcal{K}$ is the space generated by the kernel $\mathcal{K}$, $f = h + b$, $h \in \mathcal{H}_\mathcal{K}$, $b \in \mathbb{R}$ and $L(f(\mathbf{x}_n), y_n)$ is the loss function.

The solution for the optimization problem described in (1) and as determined in theorem of *representation* [9], is

$$f(\mathbf{x}) = \sum_{n=1}^{N} \omega_n \mathcal{K}(\mathbf{x}, \mathbf{x}_n) + b. \tag{2}$$

This expression indicates that the SVM classifier and other related classifiers are *example-based* [10], that is, f is determined in terms of the training examples $\mathbf{x}_n$.

The examples effectively used in the final solution are called support vectors. In order to minimize the memory used and the number of calculations, it is convenient to estimate f with few support vectors. In some applications, the number of support vectors can be as high as 90% of the training examples. There are various algorithms for SVM training and the majority of them have a parameter used to influence the number of support vector. In this work, the “complexity” parameter C was adopted [10].

The Weka software [11] was used to construct the SVM model to be synthesized. It is a collection of machine learning algorithms for data mining tasks. Weka is open source software issued under the GNU General Public License. The linear kernel SVM classifier from Weka [12] was chosen as it showed better results for our simulation scenarios.

3. Montana Database

The database used in the tests consists of twelve sounds of bird species in *wav* format. Each bird specie had twenty sample sounds. Therefore the database contains 16 synthesized syllables: 5 single tones (or chirps), 5 harmonic sounds, 5 inharmonic sounds, and 1 two-part syllable [13]. Totaling 560 sound files with sample frequency Fs = 16 kHz and resolution of 16 bits. This database was provided by Montana University [7]. The birds species are listed in Table 3. Thus, the database is composed by 28 different sound types, Item 0 through 11 are natural bird species. Item 12 through 27 are synthetic test signals with deterministic parameters.

Each specie presents different Fig. 4 shows the spectrogram of a sound segment.

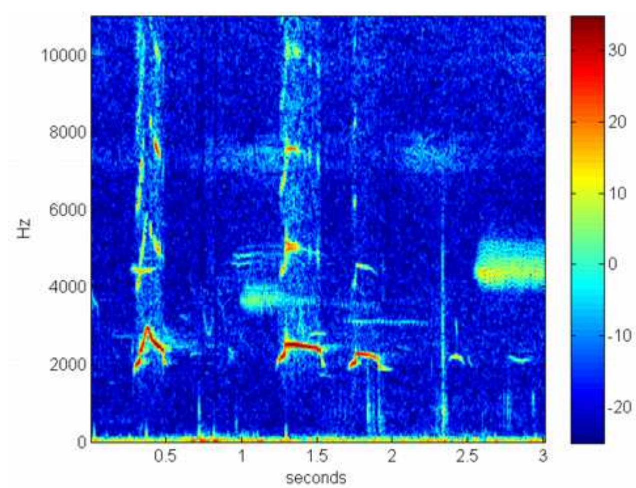

Fig. 4. A vocalization spectrogram of a bird, the color indicates intensity.

4. Results

The tests were performed with 13-dimensional mel-frequency cepstral coefficients features vectors, each sample sound with sample rate of 16 kHz, high pass filtered at 100 Hz and Hamming windowed 16 ms (256 samples frames).

TABLE I
LIST OF SOUND TYPES. ITEM 0 THROUGH 11 ARE NATURAL BIRD SPECIES. ITEM 12 THROUGH 27 ARE SYNTHETIC TEST SIGNALS WITH DETERMINISTIC PARAMETERS.

Item	Description
0	Mallard Anas platyrhynchos
1	American Crow Corvus brachyrhynchos
2	Canada Goose Branta canadensis
3	Baltimore Oriole Icterus galbula
4	Common Nighthawk Chordeiles minor
5	Killdeer Charadrius vociferous
6	Osprey Pandion haliaetus
7	Northern Cardinal Cardinalis cardinalis
8	Blue Jay Cyanocitta cristata
9	Great Horned Owl Bubo virginianus
10	Trumpeter Swan Cygnus buccinator
11	Herring Gull Larus argentatus
12	Single chirp. Frequency linearly increases
13	Single chirp. Frequency linearly decreases
14	Single chirp. Frequency linearly increases and then decreases
15	Single chirp. Frequency linearly decreases and then increases
16	Single tone
17	Harmonic chirp. Frequency linearly increases
18	Harmonic chirp. Frequency linearly decreases
19	Harmonic chirp. Frequency linearly increases and then decreases
20	Harmonic chirp. Frequency linearly decreases and then increases
21	Harmonic tone
22	Inharmonic chirp. Frequency linearly increases
23	Inharmonic chirp. Frequency linearly decreases
24	Inharmonic chirp. Frequency linearly increases and then decreases
25	Inharmonic chirp. Frequency linearly decreases and then increases
26	Inharmonic tone
27	One syllable has two parts: inharmonic tone plus inharmonic chirp

The performance of SVM was compared to other popular classification techniques: k-nearest neighbor algorithm (KNN) [14] and Tree J4.8 [15], both available in Weka software. The KNN is technique that belongs to family IBL (*instance based learning*), which look for the "nearest neighbors" to perform classification of new examples. In this work, the euclidean distance is used as the distance metric. The Tree J4.8 is machine learning model predictive [15] that can be represented as a rules set "if-else". In this model, a importance parameter is the confidence factor. A smaller confidence factor incurs less pruning of the tree. The results showed in this work represents the best results for values of confidence factor varying between [0,1; 0,25 e 0,5]. Concerning SVM classification, the LIBSVM library [16] is used, with linear kernel.

Furthermore, the robustness of the classifiers in noise presence was assessed with noise addition in database. Each file was contaminated with additive white gaussian noise (AWGN), with signal-to-noise ratio (SNR) varying from 30 to 3 dB in steps of 3 or 6 dB. Cross-validation [17] with 10-folds was adopted to test. The original sounds means that the tests were performed without noise.

The Fig. 5 shows the results using the 12 natural bird sounds and the Fig. 6 shows the results using the 28 total sounds.

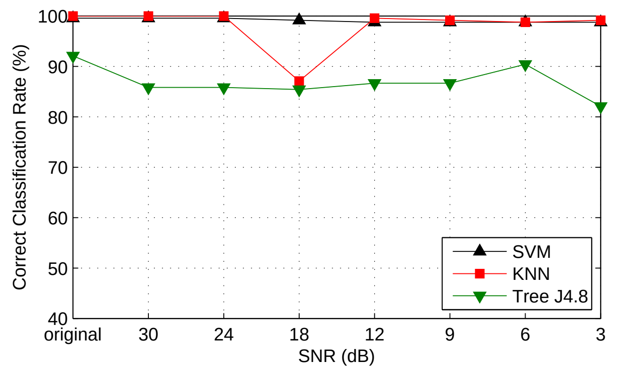

Fig. 5. Classification performance varying SNR. Tests with 12 natural sounds.

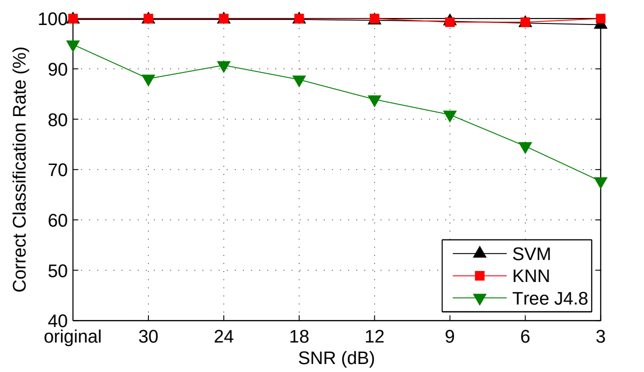

Fig. 6. Classification performance varying SNR. Tests with 28 sounds.

The SVM and KNN classification performance presents better results, moreover a little variation with SNR, unlike of the Tree J4.8.

Although of the KNN classifier presented similar performance than SVM, it has as a disadvantage for real-time applications the fact that even after being trained to classification is still costly because it requires calculating the values of individual closeness between the test and training examples. Hence, was chosen to implement the SVM classifier in VHDL.

4.1. SVM Implementation

This work is under development and its evaluation was performed in software, as described in Section 2. After evaluated was started the hardware implementation of the classifier blocks, the SVM block is under implementation. Its is being described in VHDL and synthesized to the *Altera's* EP2C20F484C7 FPGA.

Currently this block is able to perform classification between four species in Table 4-1, however this is being improved to it can work with all species.

TABLE II
SVM SYNTHESIS RESULTS TO 500 FEATURES.

Resource	Usage	Percentage
Combinational Functions	1770	9%
Logic Registers	644	3%
Pins	49	16%
Memory Bits	4096	2%
Multipliers (9 bits)	48	92%

The current description was synthesized to work with 500 features extracted by MFCC and these results are shown in Table 4-1. Other version of SVM synthesized to allow it to work with 32, 64 and 200 features. The Fig. 7 shows the memory usage to each version, other FPGA features usage remain unchanged.

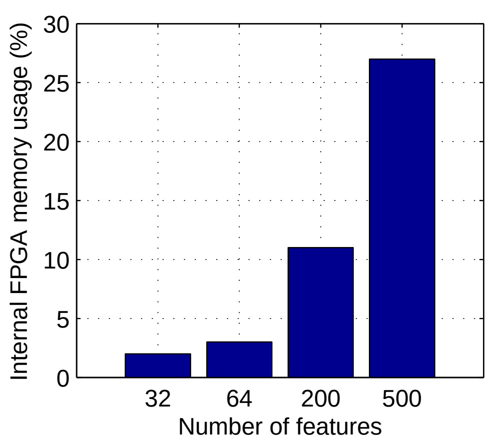

Fig. 7. EP2C20F484C7 FPGA memory usage for 32, 64, 200 and 500 features. Remaining FPGA resources still unchanged.

5. Conclusions

There are several applications to a wireless sensor network that allow the collection and processing of sounds. The IP core in development allows to classify the sensor node itself, using the MFCC and SVM techniques. The results showed the good performance of SVM classifier compared with the KNN and J4.8 classifiers, even with the variation of SNR. Partial results of the synthesis of SVM have shown that even increasing the number of features did not increase the number of logic elements, increasing only the memory usage.

References

[1] Thomas Rossing, *Springer Handbook of Acoustics*, Springer, 1 edition, 2007.

[2] O. Roy and M. Vetterli, "Distributed compression in acoustic sensor networks using oversampled a/d conversion," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006, pp. IV – IV.

[3] Jinhai Cai, D. Ee, Binh Pham, P. Roe, and Jinglan Zhang, "Sensor network for the monitoring of ecosystem: Bird species recognition," in *3rd Inter. Conf. on Intelligent Sensors, Sensor Networks and Information*, 2007, pp. 293 – 298.

[4] "MEMSIC inc.," Accessed in 2011., http://www.memsic.com/.

[5] George Kokkinakis Todor Ganchev, Nikos Fakotakis, "Comparative evaluation of various mfcc implementations on the speaker verification task," *10th International Conference on Speech and Computer (SPECOM 2005)*, vol. 1, pp. 191–194, 2005.

[6] S. Haykin, *Communication Systems*, John Wiley & Sons, 4th edition, 1999.

[7] "Montana state university - electrical and computer engineering," http://www.coe.montana.edu/ee/.

[8] R. Duda, P. Hart, and D. Stork, *Pattern classification*, Wiley, 2001.

[9] G. Kimeldorf and G. Wahba, "Some results on Tchebychean spline functions," *J. Math. Anal. Applic.*, vol. 33, pp. 82–95, 1971.

[10] B. Scholkopf and A. Smola, *Learning with kernels*, MIT Press, 2002.

[11] "http://www.cs.waikato.ac.nz/ml/weka," visited in 2011.

[12] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2006.

[13] Zhixin Chen and Robert C. Maher, "Semi-automatic classification of bird vocalizations using spectral peak tracks," in *Acoustical Society of America*, 2006, pp. 2974–2984.

[14] I. Witten and E. Frank, *Data mining: practical machine learning tools and techniques with java implementations*, Morgan Kaufmann, 2005.

[15] I. Witten and E. Frank, *Data mining: Practical machine learning tools and techniques with java implementations*, Morgan Kaufmann, 1999.

[16] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM - a library for support vector machines," Acessed in 2011.

[17] Simon Haykin, *Redes Neurais: Princípios e Prática. 2. Ed.*, Porto Alegre: Bookman, 2001.