

# Lightweight Acoustic Classification for Cane-Toad Monitoring

Thanh Dang and Nirupama Bulusu  
Department of Computer Science  
Portland State University  
Portland, OR, USA 97201  
Email: dangtx,nbulusu@cs.pdx.edu

Wen Hu  
Autonomous Systems Laboratory  
CSIRO ICT Centre, Brisbane, Australia  
Email: wen.hu@csiro.au

**Abstract**—We propose a light weight algorithm to classify cane-toads, a non-native invasive amphibian species in Australia as well as other native frog species, based on their vocalizations using sharply resource-constrained acoustic sensors. The goal is to enable fast in-network frog classification at the resource-constrained sensors so as to minimize energy consumption of the sensor network by reducing the amount of data transmitted to a central server. Each sensor *randomly* and *independently* samples a signal at a sub-Nyquist rate. The vocalization envelopes are extracted and matched with the original signal envelopes to find the best match. The computational complexity of the algorithm is  $O(n)$ . It also requires less than 2KB of data memory. Our experiments on frog vocalizations show that our approach performs well, providing an accuracy of 90% and a miss rate of less than 5%.

## I. INTRODUCTION

A wireless sensor network consists of small embedded sensing devices with extremely limited energy, processing and memory relative to a personal computer. A typical sensing device is the MicaZ mote from Crossbow, which has an Atmega128 7.4MHz processor, 128KB program memory and 512KB serial flash memory, and run on 2 AA batteries. These sharp resource constraints are necessitated by deployment requirements of low power consumption for unattended, untethered operation and small form factor design for unobtrusive monitoring because sensor networks are often deployed in remote environments to monitor a physical environment, detect, or classify some events.

In this paper, we consider the problem of using resource-constrained embedded acoustic sensors to detect and classify some amphibian species based on their vocalizations. The practical application motivating our work is the detection of cane-toads in northern Australia. The cane-toad, a non-native species was originally introduced in Australia during the 1930s to control sugar-cane pests. In the absence of natural predators, they have colonized most of northeastern Australia and raised concerns about their impact on the environment and other native species [1]. Scientists are interested in detecting the cane-toads from their call patterns as well as studying their impact on native frog species over a period of time.

A major challenge for such outdoor sensor network applications is energy conservation. Sensors are typically battery operated and the energy can be depleted very quickly. Communication consumes a dominant amount of energy, in fact, about 60% of the total energy according to one study[2]. Our goal is to develop a lightweight acoustic classification algorithm for implementation at the resource-constrained motes that minimizes the classification time while preserving the sensor energy to prolong the sensor network lifetime.

The state-of-the-art approach for acoustic classification of cane-toads [1] requires sensors to sample the signal at the Nyquist rate and transfer the samples to higher-capability devices for classification using machine learning algorithms. Unfortunately, this approach is energy and bandwidth intensive because the sensors must sample the signal at a high rate (10 KHz) and expend a lot of energy and bandwidth in transferring the samples. This approach is also not very scalable due to the large amounts of data collected with each sensor sampling at 10 KHz.

Recent results in *compressive sensing* theory [3] show that a small number of random samples can embed the signal structures well, provided that the signal is sparse in some domain. Building on this result, two proposed approaches [4], [5] perform detection and classification on randomly sampled data. Boyle *et al.* [4] show that we can classify signals from the histogram of the signal random samples. Unfortunately, the histograms are difficult to recover under noise. Davenport *et al.* [5] propose a smashed filter, a variant of the matched filter to classify signals using random samples. However, it is computationally infeasible to use the smashed filter in resource-constrained devices for a large number of samples.

To overcome the limitations of previous approaches, we have designed a light-weight classification algorithm that can be executed at the resource-constrained sensors, such as the MicaZ mote class devices. The main intuition for this approach is that animal vocalizations in general, and cane-toad vocalizations in particular, have simple repeated patterns without the variation prevalent in human speech. This implies that simple feature extraction should work for animal classification. Our

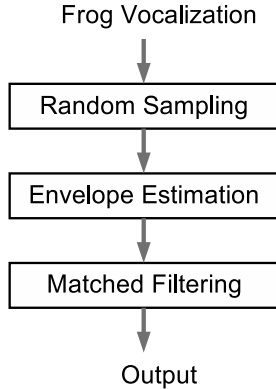


Fig. 1. Lightweight acoustic classification

approach is to estimate the signal envelope from randomly-sampled data at a much lower sampling rate than the signal Nyquist rate and match this envelope with the candidate signal envelope.<sup>1</sup> Our approach differs from using a low pass filter or a uniform sampling rate in that randomly-sampled data can capture more information than uniform sampled data. This approach has the following key advantages.

- It requires a low sampling rate which enables resource-constrained sensors to classify high frequency signals.
- It is computationally feasible at the resource-constrained sensor level.
- It does not require precise timing, which is difficult on resource-constrained sensor devices.
- It requires only a small memory storage.

## II. ALGORITHM

Our proposed classification algorithm has three main steps — random sampling, envelope extraction, and matched filtering (Figure 1).

### A. Random Sampling

Recent results in compressive sensing show that a small number of random samples of a signal can actually embed the signal structure well, provided the signal is sparse in some domain. Random sampling is also suitable for sensor devices because timing control is difficult. The random samples may happen to be close together in time, requiring the sensor to sample at a high frequency for a short period. To overcome this, we generate random indexes in a short time period and scale them to a large time scale. The scale factor is the shortest time period between two samples.

The sampling time is determined by the following equation:

$$\text{sampleTime} = M \times \text{randSample}(n/M, m) + s \times \text{round}(\text{randn}(m, 1)) \quad (1)$$

<sup>1</sup>Note that our proposed approach cannot be implemented if the resource-constrained nodes were sampling at the Nyquist rate of 10 KHz, instead of performing compressed random sampling. As shown in [1], Nyquist rate sampling at 10 KHz requires the node to suspend other tasks.

$n$  is the signal length,  $M$  is the down sampling factor,  $m$  is the total number of measurements, and  $s$  is the scaling factor that controls how large the jitter (discussed shortly) is.  $M$  controls the minimum time interval between each sample. Function  $\text{randSample}(a, b)$  picks  $b$  arbitrary numbers with an independent and identical distribution (i.i.d) from 1 to  $a$ . Function  $\text{randn}(a, b)$  generates an  $a \times b$  matrix with random entries, chosen from a normal distribution with zero mean and variance one.

It is possible that a sensor must collect two random samples that are extremely close in time. In this case, for a short time period, the sensor must be able to sample at Nyquist rate. We overcome this limitation by choosing the sampling time randomly in a short time period  $\tau$  and dilate it to the full time scale  $\Gamma$ . The sampling time sequence is determined by  $M \times \text{randSample}(n/M, m)$ . We select  $m$  sample times randomly from the time scale  $\tau$ ,  $[1, n/M]$  and dilate the sample time to the time scale  $\Gamma$ ,  $[1, n]$  by multiplying the sampling time by  $M$ . The sampling times are now random from 1 to  $n$  and are at least  $M$  time units apart. This ensures that we not only collect a smaller number of samples but also sample at a rate  $M$  times lower than the Nyquist rate. To increase the sampling time randomness, we introduce the *time jitter*, which has a normal distribution at the sampling time. We generate jitter using  $s \times \text{round}(\text{randn}(m, 1))$ .

### B. Envelope Extraction

Roughly speaking, the envelope of a signal  $x(t)$  is the boundary within which the signal is contained. The signal envelope can be estimated by applying a low-pass filter or smoothing the signal. We simplify the computation by just windowing the signal and taking the maximum absolute value of the samples in each non-overlap window.

We subdivide the recorded samples  $x(n)$ ,  $0 \leq n \leq N - 1$  into  $K$  non-overlapping smaller segments as follows:

$$x_i(n) = x(iD + n)w(n) \quad 0 \leq n \leq L - 1, 0 \leq i \leq K - 1 \quad (2)$$

where  $w(n)$  is basically a rectangular window of duration  $L$  and  $D$  is an offset distance.

The envelope is then estimated as:

$$y(i) = \max_{n=0}^{L-1} x_i(n) \quad (3)$$

Sensors can compute the envelope very efficiently because they only need to examine the maximum value of the incoming samples in each window. Therefore, sensors can compute the envelope as they sample the signal. In general, a frog often makes several calls continuously. Hence, we use a threshold  $\epsilon$  to separate each call. Each envelope is normalized to have unit energy. The set of envelopes of frog calls is denoted as  $y^m(i)$  where  $m$  is the index of the call. Figure 2 shows an example of frog calls and the envelopes. Using the same techniques, we

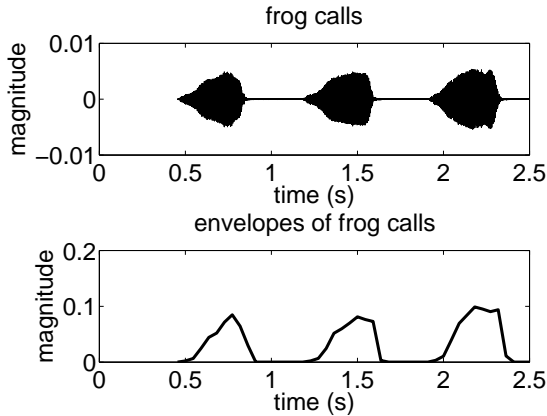


Fig. 2. Example frog call and three extracted envelopes

also extract the benchmark envelopes of all the frog species using the training signals. We denote this set of envelopes as  $Y_k^m(i)$  where  $k$  is the index of the frog.

### C. Matched Filtering

After the set of frog call envelopes  $y^m(i)$  are recorded, we apply the matched filter of  $y^m(i)$  with  $Y_k^m(i)$  to detect the frog species. The matched filter is used extensively in signal processing and communication to detect a signal. It can be performed efficiently by convolution. The classified frog  $k$  is defined as

$$k = \text{arg}_k \max \sum_1^m y^m(i) * Y_k^m(-i) \quad (4)$$

where  $*$  denotes the convolution operator, which is defined as

$$(f * g)(n) = \sum_{m=-\infty}^{\infty} f(m)g(n - m) \quad (5)$$

where  $f$  and  $g$  are two signals.  $k$  can be interpreted as the frog type whose calls have the maximum correlation with the benchmark calls.

## III. EXPERIMENTAL RESULTS

We have conducted several experiments to study the effectiveness of our algorithm. Depending on the frog type, we used between 5 to 20 coefficients to represent a frog call envelope. The total number of benchmark envelopes is less than 2 KBytes. We recorded the different cane toad songs randomly. Then, we perform random sampling, extract the recorded signal envelopes and run the matched filter. For each signal, we iterate 50 times by generating 50 random sample sets. We vary the down sampling factor from 5 to 200 in steps of 5. We also vary the down sampling factor at a larger scale from 100 to 1000 in steps of 100. We observed the following metrics:

- true positive rate, *i.e.*, the percentage of times that the frogs are correctly classified
- false negative rate, the percentage of times that the frogs are present but could not be detected, and

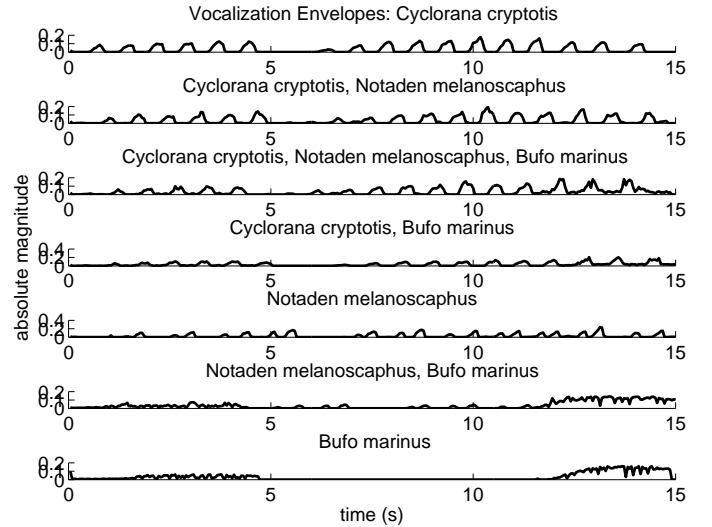


Fig. 3. Envelopes of cane toad calls

- false positive rate, the percentage of times that the frogs are not present but are detected

In general, we wish to have a high true positive rate and a low false negative rate. The low false negative rate is critical because we do not want to miss a frog. Ideally, the false positive rate should also be low. However, it is less critical if this rate is high because sensors can trigger a more capable server to verify the result. It is preferable to detect a frog incorrectly rather than miss a frog.

In the first set of results, we can see that even with a down sampling factor of more than 100, the true positive rates are above 90%. The true positive rates for Bufo Marinus (scientific name of the cane-toad) and Notaden Melanoscapus are always 100% (Figure 4). These rates degrade slowly as the down sampling factor approaches 200. In Figure 5, the false negative rates start at about 5% and increase gradually to 20% as the down sampling factor increases from 5 to 200. Notably, the false positive rates are pretty high, about 50% for Bufo Marinus and Cyclorama Cryptotis (Figure 6). However, as we mentioned before, a high false positive rate is less critical than a high false negative rate as we would not want to miss a frog. However, a false detection of a frog can be verified by triggering a higher computation backend server to check the result.

As we increase the down sampling factor to 1000, the classification degrades very quickly. However, the true positive rate for Notaden Melanoscapus is still very high and its false negative is also low. However, its false positive rate also increases to 80%. As the down sampling factor approaches 1000, the false positive rates for Bufo Marinus and Cyclorama go to 0%. However, the true positive rates and the false negative rates go to 0% and 100% respectively.

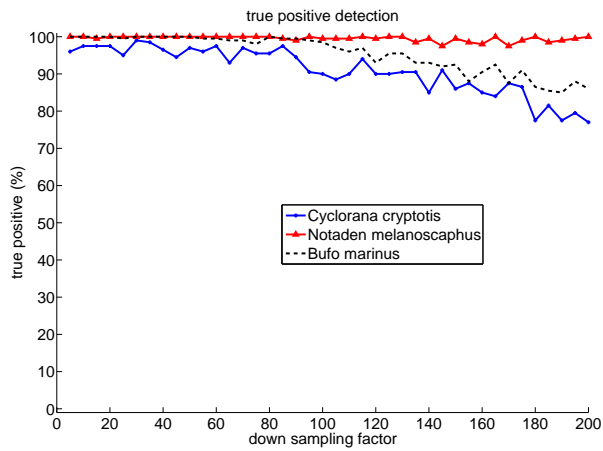


Fig. 4. True positive classification rate when down sampling factor varies from 5 to 200

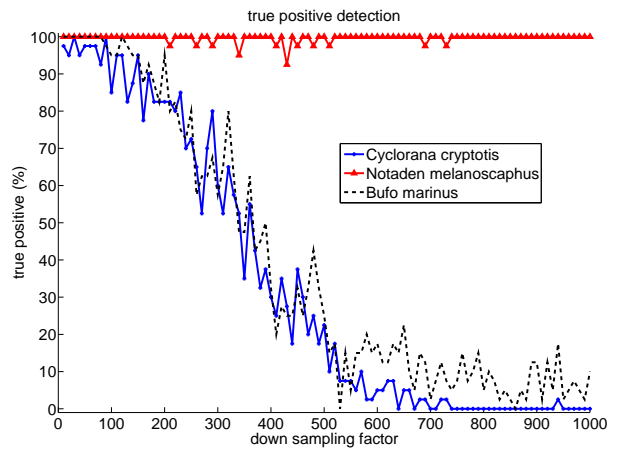


Fig. 7. True positive classification rate when down sampling factor varies from 100 to 1000

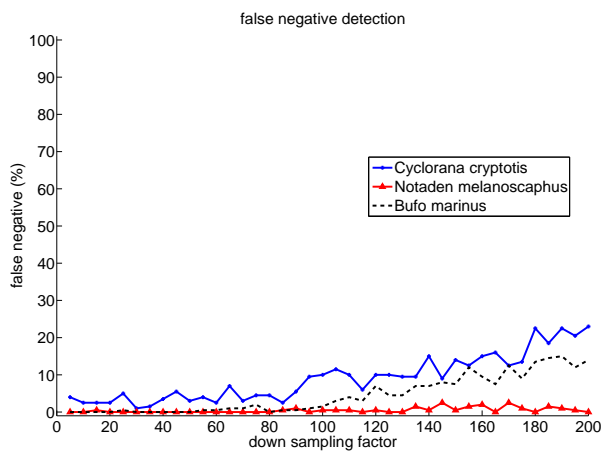


Fig. 5. False negative classification rate when down sampling factor varies from 5 to 200

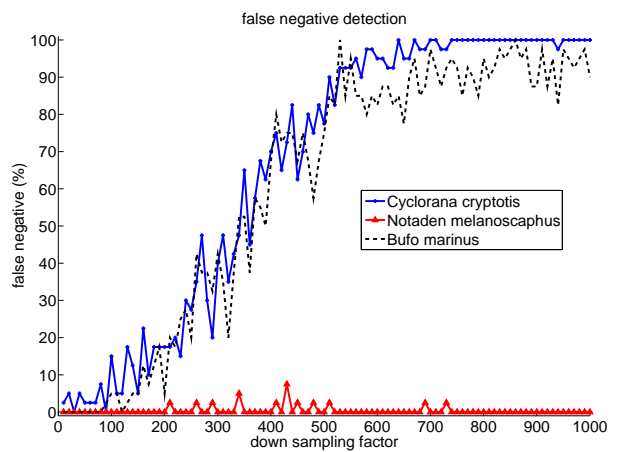


Fig. 8. False negative classification rate when down sampling factor varies from 100 to 1000

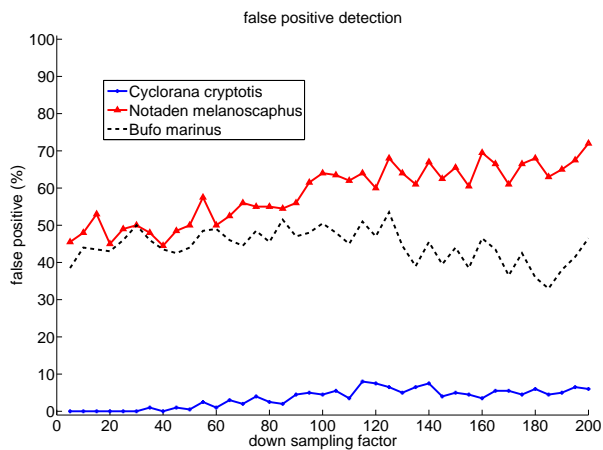


Fig. 6. False positive classification rate when down sampling factor varies from 5 to 200

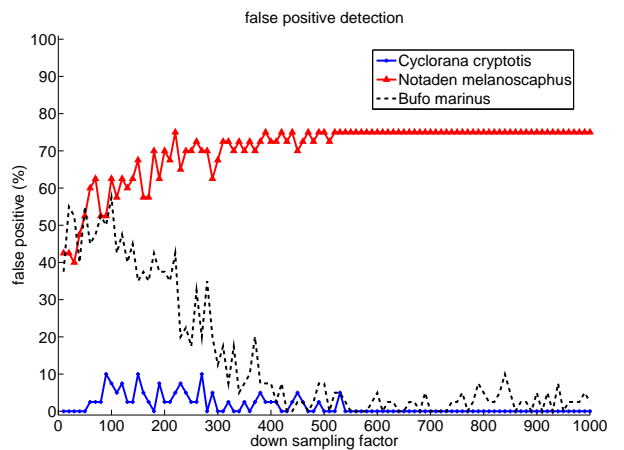


Fig. 9. False positive classification rate when down sampling factor varies from 100 to 1000

#### IV. CONCLUSION

We presented a lightweight algorithm to classify cane toads based on their call patterns. It builds on compressive sensing theory which shows that a small number of random samples can embed the signal structures well. In contrast to prior work, the algorithm can be readily implemented on sharply resource-constrained sensors such as the MicaZ motes, without any specialized hardware (eg. DSP chips). It requires only about 2 KBytes of memory for classification and allows sensors to sample an acoustic signal at a rate 100 times lower, i.e., 1% of the original 20 KHz Nyquist rate of the signal. Our experimental results on real world cane-toad vocalizations show that we can classify cane toads with more than 90% accuracy and less than 5% miss rate. This algorithm clearly is not the optimal algorithm for detection and classification in general. However, this algorithm can significantly extend sensor network lifetime and minimize classification time.

#### ACKNOWLEDGEMENTS

This research was supported in part by the National Science Foundation, through grant 0722063. Although this publication is based upon work supported by NSF, Any opinions, findings, conclusions or recommendations it contains are those of the authors, and do not necessarily reflect the views of the Foundation.

#### REFERENCES

- [1] W. Hu, V. N. Tran, N. Bulusu, C. T. Chou, S. Jha, and A. Taylor, "The design and evaluation of a hybrid sensor network for cane-toad monitoring," in *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*. Piscataway, NJ, USA: IEEE Press, 2005, p. 71.
- [2] C. M. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," in *Proceedings of ACM Conference on Embedded Sensor Systems (Sensys)*, Boulder, Colorado, November 2006, pp. 265–279.
- [3] E. Candes, J. K. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, February 2006.
- [4] F. Boyle, J. Haupt, G. Fudge, and R. Nowak, "Detecting signal structure from randomly-sampled data," in *Proceedings of IEEE Workshop on Statistical Signal Processing (SSP)*, Madison, Wisconsin, August 2007, pp. 326–330.
- [5] M. Davenport, M. Duarte, M. Wakin, J. Laska, D. Takhar, K. Kelly, and R. Baraniuk, "The smashed filter for compressive classification and target recognition," in *Proceedings of Computational Imaging V at SPIE Electronic Imaging*, San Jose, California, January 2007, pp. 326–330.