

# A Hidden Markov Model-Based Acoustic Cicada Detector for Crowdsourced Smartphone Biodiversity Monitoring

Davide Zilli, Oliver Parson, Geoff V Merrett, Alex Rogers

Electronics and Computer Science, University of Southampton, Southampton, UK

{dz2v07, op106, gvm, acr}@ecs.soton.ac.uk

## Abstract

Automated acoustic recognition of species aims to provide a cost-effective method for biodiversity monitoring. This is particularly appealing for detecting endangered animals with a distinctive call, such as the New Forest cicada. To this end, we pursue a crowdsourcing approach, whereby the millions of visitors to the New Forest will help to monitor the presence of this cicada by means of a smartphone app that can detect its mating call. However, current systems for acoustic insect classification are aimed at batch processing and not suited to a real-time approach as required by this system, because they are too computationally expensive and not robust to environmental noise. To address this shortcoming we propose a novel insect detection algorithm based on a hidden Markov model to which we feed as a single feature vector the ratio of two key frequencies extracted through the Goertzel algorithm. Our results show that this novel approach, compared to the state of the art for batch insect classification, is much more robust to noise while also reducing the computational cost.

## 1 Introduction

Biodiversity is a key measure of the health of an ecosystem, and as land-use and climate change impact on the natural environment, many countries are increasingly seeing the need to monitor and protect it. For example, the UK has formalised this within the UK Biodiversity Action Plan (UK BAP) and has established a priority species list to focus work on a small number of critically important species [Joint Nature Conservation Committee, 2010]. One of these, of particular interest in this paper, is the New Forest cicada (*cicadetta montana* s. str., see Figure 1); the only native cicada known to the UK, which was first identified in the New Forest, a national park on the south coast of England, in 1812. Despite being well studied at a number of sites in the 1960s, there has been no confirmed observation of the New Forest cicada in the last 20 years [Pinchen and Ward, 2002]. Understanding whether this is simply due to the migration of the cicada to as yet undiscovered sites, or whether the cicada is now extinct in the UK



Figure 1: *Cicadetta montana*. Photograph by Jaroslav Maly, reproduced with permission.

due to climate change or land-use change, is an important question for UK biodiversity research.

Today, traditional approaches to searching for rare species typically call for trained ecologists to perform detailed manual surveys. However, the obvious costs of such work have led to significant recent research into automated approaches whereby animals and plants can be classified remotely without requiring that trained experts be in the field. In the case of insects, this is most often performed by deploying fixed sensors with sensitive microphones that record the sounds that the insects emit [MacLeod, 2007]. These recordings are then analysed later to automatically identify the insects whose calls were heard. The algorithms to do so typically range from those that operate solely in the time domain, such as time domain signal coding [Chesmore, 2004; Chesmore and Ohya, 2004], to those inspired by the literature of human speech recognition. The latter typically use a hidden Markov model (HMM) for classification [Leqing and Zhen, 2010], and perform a number of pre-processing stages, often taken directly from the human speech recognition literature, to extract features from the raw recording. For example, [Chaves *et al.*, 2012] present a state-of-the-art approach that pre-processes the recorded sound to remove unsounded periods where no insect call is detected, that maps the raw frequencies to the mel scale, which better represents human hearing; then it converts it back into a pseudo-time domain, called the *cepstrum*, by calculating a number of mel frequency cepstral coefficients (MFCC), that are used as features for the HMM classification with just one state per species.

The use of automatic acoustic recognition is particularly appealing in the case of the New Forest cicada, since this in-

sect has a particularly loud high-pitched mating song which, while being close to the upper frequency limit of a normal adult hearing range, can easily be detected by conventional microphones. However, the use of fixed sensors to collect these recordings for later analysis is less compelling. The New Forest covers 600 km<sup>2</sup>, and, to exhaustively survey it for potential cicada breeding sites, would require tens of thousands of sensors. Therefore, in our work, we are pursuing a different approach, and are seeking to exploit the 13 million day visits to the New Forest that occur each year by the general public to crowdsource the search for the New Forest cicada. In particular, in this paper, we describe a HMM-based detection algorithm that runs within a smartphone app, to be used by these visitors, that can automatically detect the mating call of the male cicada, identify it as such to the user in real-time, and prompt the user to make a recording that can be uploaded for further analysis once the smartphone has an acceptable Internet connection. A similar approach to crowdsourcing for sustainability problems using mobile phones was employed by Quinn *et al.* [2011], however it has not yet been applied to crowdsourcing biodiversity using acoustics.

The requirements of the automatic detection algorithm that will run on these smartphones are somewhat different to those that have been previously proposed for post recording analysis of insect calls. Firstly, we must be aware of the limited computational resources available on some lower-end mobile devices. The algorithms described above are typically designed to run on high-end servers, and thus, are relatively unconstrained by both memory and processor speed. Secondly, we note that the algorithms above are designed to run in batch-mode on existing recordings, whereas in our application, we require that the algorithms provide real-time feedback to the user as to the identification of the insect being heard; firstly, so that if it is not a cicada we do not force the user to upload unnecessary recordings<sup>1</sup>, and conversely, if a cicada is detected, so that we can prompt the user to collect the best quality recording possible and keep them engaged in the search. Finally, the pre-processing approaches described above may actually impair detection in certain cases. In particular, while often insects can easily be classified by differences in the frequencies of their song, we demonstrate that these methods fail to distinguish between the song of the New Forest cicada (which sings continuously at 14 kHz), and that of the dark bush cricket, common in the New Forest (that chirps intermittently at 14 kHz). This is because the conversion of the raw frequency domain data into mel frequency cepstral coefficients fails to generate any distinguishing features in the frequency-domain (since the mel frequency conversion has poor resolution at high frequencies, being intended for the much lower frequencies which constitute typical human speech), while the automatic removal of un-sounded periods from the recording also removes the time-domain features which would have differentiated them.

Thus, to address these shortcomings, in this paper we present an algorithm specifically intended for real-time de-

<sup>1</sup>A 60s mono recording at 44,100 samples per second, is about 5MB; a significant file to upload in areas with poor mobile phone reception where connection rates may be down to 100kbps or less.

tection and recognition of insects (and specifically, the New Forest cicada) on computationally constrained smartphones. Rather than calculating a number of mel frequency cepstral coefficients, as above, we use the Goertzel algorithm — an efficient method for approximating individual terms of a discrete Fourier transform (DFT) [Goertzel, 1958] — to calculate the magnitude of two specific frequency bands; one centred at 14 kHz, corresponding the central frequency of both the insects’ calls, and one centred at 8 kHz, which is far from both general background noise and the insects’ call. We use the ratio of these magnitudes as a single feature, which identifies the song of either the bush cricket or the New Forest cicada. Then, we use a four-state hidden Markov model that explicitly represents both the idle, un-sounded period between insect calls, and also the short pauses between the chirps of the bush cricket’s song. Hence, rather than attempting to recover the time domain information lost while removing un-sounded periods through heuristic methods, we explicitly capture this in the HMM, as this approach can be readily extended to cover insect calls of more complexity, all within the same principled framework. We then use the Viterbi algorithm to identify the most likely sequence of insect calls at any point in time, as in Sun *et al.* [2009].

We evaluate our approach using recordings made by the authors with an Apple iPhone 4S of both the New Forest cicada (recorded in Slovenia where the same species is still abundant) and the dark bush cricket (recorded in the New Forest). Unlike standard library recordings, our trial dataset represents the quality of crowdsourced data that we are likely to encounter, exhibiting significant noise (including, among others, handling noise, background road traffic, human voice and noise generated by the wind), and insect calls of varying amplitude depending on the proximity of the recording device to the specimen.

We show how our approach is capable of identifying the cicada call in normal environmental noise more accurately than the state-of-the-art batch classification algorithms described earlier. In particular, we achieve an  $F_1$  score of 0.955 for the detection of the cicada on a large data set of over 30 calls, recorded with a smartphone, while the method proposed by Chaves *et al.* [2012] only scores  $F_1 = 0.126$  due to the confusion with the cricket’s calls. Our efficient feature extraction procedure is robust to noise and decreases the computational complexity of the detection process, providing the capability for real time classification.

The remainder of this paper is organised as follows. In Section 2, we describe our proposed approach, highlighting the different techniques used. In Section 3 we analyse its performance in comparison to a state-of-the-art model for batch classification, providing the relevant accuracy metrics. We conclude in Section 4 along with an overview of future work.

## 2 Real-Time Insect Detection Using Hidden Markov Models

We now give a description of our proposed approach for real-time insect detection. We first describe a method by which we efficiently extract individual terms of a DFT from the raw audio recordings using the Goertzel algorithm. We then de-

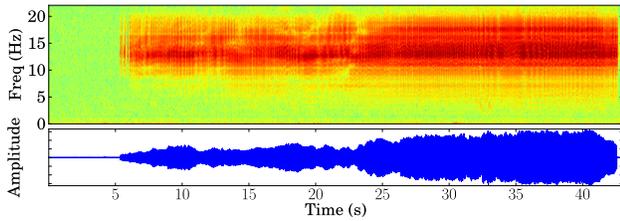


Figure 2: Spectrogram and waveform of a New Forest cicada call (recording by Jim Grant, 1971 and courtesy of the British Library, wildlife sounds collection).

scribe how two of these terms can be combined to produce a feature that is robust to noise. Last, we show how this feature is used to classify periods of a recording to a particular insect using a four-state HMM.

### 2.1 Feature Extraction Using Goertzel Algorithm

For the purposes of our system, it was observed that the call of the New Forest cicada displays a strong frequency component centred around 14 kHz (see Figure 2). This frequency is sufficiently distant from any common background noise, such as wind noise, road traffic or people speaking, to be a reliable identifier of the presence of the cicada. An efficient approximation of the magnitude of this given frequency can be calculated using the Goertzel algorithm, a method that evaluates individual terms of a DFT, implemented as a second order infinite impulse response (IRR) filter.

An efficient implementation of the Goertzel algorithm requires two steps. The first step produces a coefficient that can be pre-computed and cached to reduce CPU cycles:

$$c = 2 \cos \left( \frac{2\pi f}{f_s} \right) \quad (1)$$

where  $f$  is the central frequency in question and  $f_s$  the sampling rate of the recording.

The second step consists of iteratively updating the values of a temporary sequence  $y$  with any incoming sample  $s$  such that:

$$y_n = \text{hamming}(s) + (c \cdot y_{n-1}) - y_{n-2} \quad (2)$$

where the samples are passed through a *Hamming* filter, given by:

$$\text{hamming}(s) = 0.54 - 0.46 \cos \left( \frac{2\pi s}{N-1} \right) \quad (3)$$

and the length of the sequence of samples  $N$  determines the bandwidth  $B$  of the Goertzel filter, such that:

$$B = 4 \frac{f_s}{N} \quad (4)$$

A sequence length  $N$  yields larger bandwidth, at the cost of a noisier output. In practice, we use multiples of 64 samples to match a typical smartphone’s buffer size. For example, a block size  $N = 128$  samples gives a bandwidth of just under 1.4 kHz. The magnitude  $m$  of the frequency band centred at  $f$  and with bandwidth  $B$  is then given by:

$$m_f = \sqrt{y_N^2 + y_{N-1}^2 - c \cdot y_N \cdot y_{N-1}} \quad (5)$$

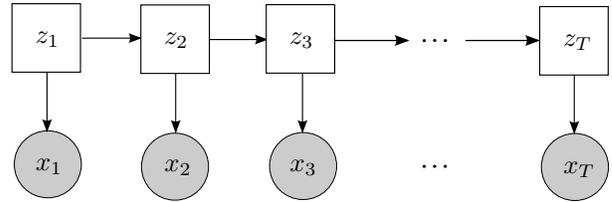


Figure 3: A hidden Markov model. Unshaded square nodes represent observed discrete variables, while shaded circular nodes represent hidden continuous variables.

In terms of computational complexity, this approach shows a considerable benefit compared to the single-bin DFT. An efficient algorithm to compute the latter, the fast Fourier transform (FFT), has a complexity of  $O(N \log N)$ , while the Goertzel algorithm is only computed in order  $O(N)$ , where  $N$  is the number of samples per window. Moreover, the sample update described in equation 5 can be processed in real-time, eliminating the need for an independent background thread on the smartphone app.

### 2.2 Feature Combination Using Filter Ratio

The magnitude of the frequency component at 14 kHz is a good indicator of the presence of a New Forest cicada, robust against generic background noise. However, it may be sensitive to white noise that covers the entire frequency spectrum, such as handling noise. Therefore, in order to reduce this sensitivity, we divide the magnitude of this feature by the magnitude observed around 8 kHz, also computed with the Goertzel algorithm described above. This band is outside the range of both the cicada call and environmental noise. Hence, this ratio will be high in the presence of a cicada and tend to zero when either no sound is detected in the cicada range or if sound is present across both bands. The ratio of the sequences of these two terms  $m_{f,1}, \dots, m_{f,T}$ , computed over time  $T$ , results in the feature vector  $\mathbf{x} = x_1, \dots, x_T$  such that:

$$\mathbf{x} = \frac{\mathbf{m}_{14}}{\mathbf{m}_8} \quad (6)$$

Once the ratio between the 14 kHz and 8 kHz frequencies has been calculated, this can be used as a single feature vector,  $\mathbf{x}$ , for our classification model. In order to obtain real-time computationally efficient insect identification, we adopt a HMM based approach to classification.

### 2.3 Classification Using Four-State HMM

A HMM consists of a Markov chain of discrete latent variables and a sequence of continuous observed variables, each of which is dependent upon one discrete variable’s state [Blasiak and Rangwala, 2011]. Figure 3 shows the graphical structure of a HMM, where the discrete, hidden variables (the singing states) are represented by the sequence  $z_1, \dots, z_T$ , and the continuous, observed variables (the song feature) are represented by the sequence  $x_1, \dots, x_T$ . The value of each discrete variable  $z_t$  corresponds to one of  $K$  states, while each continuous variable can take on the value of any real number.

The behaviour of a hidden Markov model is completely defined by the following three parameters. First, the probability

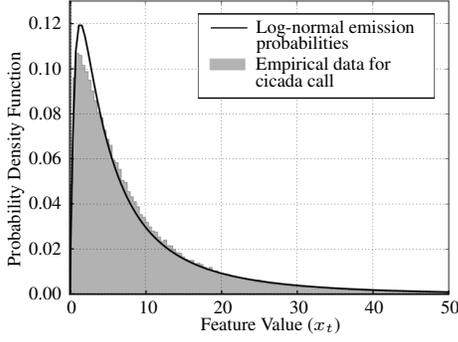


Figure 4: Log-normal distribution of the extracted feature for the cicada call

of each state of the hidden variable at  $t = 1$  is represented by the vector  $\pi$  such that:

$$\pi_k = p(z_1 = k) \quad (7)$$

Second, the transition probabilities from state  $i$  at  $t - 1$  to state  $j$  at  $t$  are represented by the matrix  $\mathbf{A}$  such that:

$$A_{i,j} = p(z_t = j | z_{t-1} = i) \quad (8)$$

Third, the emission probabilities that describe the observed feature,  $\mathbf{x}$ , given parameters  $\phi$ , follow a log-normal distribution such that:

$$x_t | z_t, \phi \sim \ln \mathcal{N}(\mu_{z_t}, \sigma_{z_t}^2) \quad (9)$$

where  $\phi = \{\mu, \sigma^2\}$ , and  $\mu_{z_t}$  and  $\sigma_{z_t}^2$  are the mean and variance of the Gaussian distribution for state  $z_t$ . Figure 4 shows a histogram of data generated by a cicada’s song, along with our log-normal distribution fitted to the data. Despite the distribution’s long tail, it still has poor support for data of unusually high magnitude, as are often generated by handling noise. In order to prevent the model from strongly favouring a certain state when a data point is in the extreme of the log-normal distribution, we add a uniform term over all emission probabilities, with small probability, to capture cases where our data are likely to be poorly represented.

Equations 7, 8 and 9 can then be used to calculate the joint likelihood of a hidden Markov model:

$$p(\mathbf{x}, \mathbf{z} | \theta) = p(z_1 | \pi) \prod_{t=2}^T p(z_t | z_{t-1}, \mathbf{A}) \prod_{t=1}^T p(x_t | z_t, \phi) \quad (10)$$

where the model parameters are defined by  $\theta = \{\pi, \mathbf{A}, \phi\}$ .

We use the Viterbi algorithm to infer the most likely sequence of hidden states given the features described. Despite the fact that the number of possible paths grows exponentially with the length of the chain, this algorithm efficiently finds the most probable sequence by maximising Equation 10, with a cost that grows only linearly with the length of the chain.

## 2.4 Finite State Model of Insect Call

We propose a four-state HMM for cicada detection, in which the states consist of: an *idle* state in which no insect is singing ( $I$ ), a *cicada* singing state ( $C$ ), a state where the *dark bush*

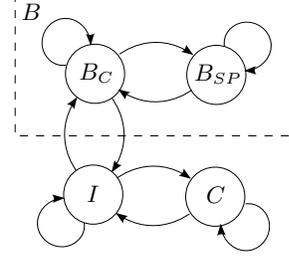


Figure 5: Four-state finite state machine.  $B_C$  represents the bush cricket’s chirp,  $B_{SP}$  represents the short pause between chirps,  $I$  represents the idle state and  $C$  represents the cicada’s song.

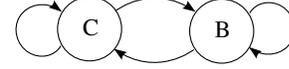


Figure 6: Two-state finite state machine.  $C$  represents the cicada’s song and  $B$  represents the cricket’s chirp.

*cricket* is chirping ( $B_C$ ) and a *short pause* in between the dark bush cricket’s chirps ( $B_{SP}$ ). The model parameters  $\mu$  and  $\sigma^2$  are learned empirically for each state  $k$  using:

$$\mu_k = \frac{\sum_{t=1}^T \ln(x_t)}{T}, \sigma_k^2 = \frac{\sum_{t=1}^T (\ln(x_t) - \mu)^2}{T} \quad (11)$$

The transition matrices describing the dynamics of a Markovian process can be represented graphically using finite state machines. Figure 5 shows the four states described above and all possible transitions, where those with non-zero probability are represented by arrows connecting two states. Our model explicitly represents the silence between the dark bush cricket song, which is essential information for distinguishing between the two insects’ songs. This is in contrast to the model used by Chaves *et al.* [2012], in which each insect is represented by a single state, as shown by Figure 6.

Having presented our model, we now go on to compare its performance to the more complex system for batch processing proposed by Chaves *et al.* [2012]. This system mimics a standard speech recognition model based on HMMs.

## 3 Empirical Evaluation Using Smartphone Recordings

We compare our approach, as described in Section 2, to the state-of-the-art approach used by [Chaves *et al.*, 2012]. In the latter, the signal is firstly stripped of un-sounded areas and segmented to extract individual calls. It is then pre-processed by removing the DC offset, dividing it into frames, emphasising high frequencies, and passing it through a windowing function. The windows are then run through a FFT and converted into the mel frequency scale, from which the mel frequency cepstral coefficients are generated. These are used as individual features for a simple HMM. For the recording in analysis, this consists of two states, one for the cicada and one

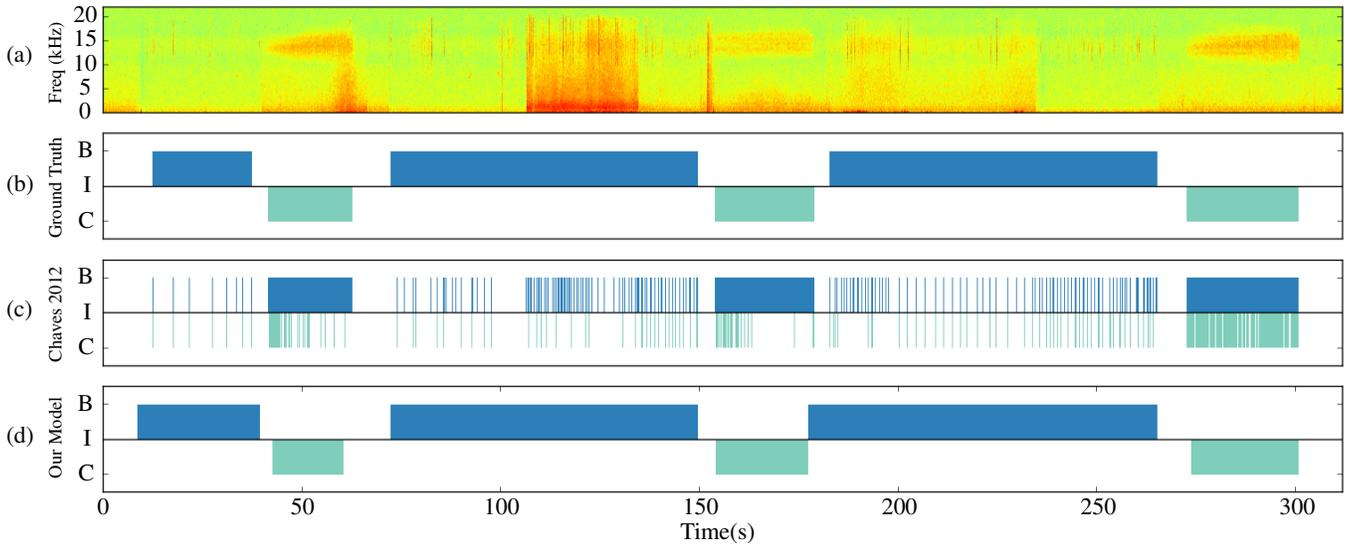


Figure 7: The proposed model, run on a recording with several dark bush cricket’s calls and three cicada songs. *I*, *C* and *B* represent the idle, cicada and cricket states respectively, as in Figure 5. *B* encompasses both the cricket’s chirping ( $B_C$ ) and short pause ( $B_{SP}$ ) states.

for the bush cricket, with a feature vector of 24 cepstral coefficients, each assumed to be normally distributed. No state for silence is considered, as this has been removed during the pre-processing stage.

To evaluate the accuracy of our approach, we collected recordings of the New Forest cicada from a known habitat in Slovenia and the dark bush cricket from the New Forest using an Apple iPhone 4S. In contrast to existing recording libraries, this data set represents the quality of crowdsourced data that we are likely to encounter, exhibiting significant noise (including, among others, handling noise, background road traffic, human voice and noise generated by the wind), and insect calls of varying amplitude depending on the proximity of the recording device to the specimen.

Figure 7 shows a comparison of the two approaches using a concatenation of three cicada calls and several instances of the dark bush cricket call intertwined. Figure 7a shows a spectrogram with the time domain on the  $x$ -axis, and the frequency domain on the  $y$ -axis, with the magnitude of the frequency bins varying with the colour of the plot. The three cicada calls can be identified as the prolonged strong component in the high frequency band. The chirping calls are visible as thin vertical bars on the top half of the spectrum. Note that the different recordings, merged together into this data set, have varying background noise, identifiable particularly as high magnitude components at the bottom of the spectrum. Figure 7b shows the ground truth, labelled manually, i.e. the correct classification of the different insects. The states are labelled as in Figure 5, where *I* represents the un-sounded idle state, *C* represents the cicada’s song and *B* represents both the bush cricket’s chirping and short pause states. Figure 7c shows the output of the model from Chaves *et al.* [2012]. For this approach, areas identified as idle have been removed from the feature by the pre-processing stage, but have been

reintroduced in the output for the sake of comparison. On the plot they are marked as idle, although the model itself does not account for an idle state. Since the comparison is focused on the discernment of the two insects rather than the detection of sounded and un-sounded areas we manually label the sounded and un-sounded areas. Finally, Figure 7d shows the output of the model proposed in this paper. The two states used to identify the dark bush cricket’s call are merged into one, again as represented in Figure 5. It is immediately apparent how closely our proposed approach matches the ground truth in comparison to Chaves *et al.* [2012].

It emerges clearly that removing silence between calls also removes the time domain features crucial at discerning these two insects. The output of the HMM in Figure 7c, displays confusion between the chirping call and the prolonged call and is unable to identify them correctly. The visual intuition is confirmed by the accuracy measures described below and reported in Table 1. On the contrary, our proposed model is able to take advantage of the clear time-domain feature and, despite the emission probabilities of the two sounded and the two un-sounded states being identical, the transition probabilities ensure that prolonged periods of silence are classified as the idle state. To this extent, the backward pass of the Viterbi algorithm ensures that any mistakes due to a state having the highest local probability are corrected to provide the most likely overall path. Furthermore, this approach can be readily extended to calls of more complexity by further increasing the number of sub-states attributed to each insect.

We assess the accuracy by which each approach can correctly classify the cicada using the standard precision, recall and  $F_1$  score metrics. The precision represents the fraction of time slices in which the approach detected the cicada as singing when it was in fact singing, while the recall represents the fraction of time slices in which the cicada was singing that

Approach	Precision	Recall	$F_1$ -score
Our approach	1.000	0.914	0.955
Chaves <i>et al.</i> [2012]	0.563	0.071	0.126

Table 1: Accuracy metrics of cicada detection

were correctly detected. Precision and recall are defined as:

$$precision = \frac{tp}{tp + fp} \quad (12)$$

$$recall = \frac{tp}{tp + fn} \quad (13)$$

where  $tp$  represents the number of correct cicada song detections,  $fp$  represents the number of cicada song detections when it was actually not singing, and  $fn$  represents the number of cicada songs which were not detected. In this work, we are not concerned by the accuracy of the cricket’s detection. We also use the  $F_1$  score, which represents a weighted combination of precision and recall, defined as:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (14)$$

Table 1 shows the precision, recall and  $F_1$  score metrics both for the approach described in this paper and that used by Chaves *et al.* 2012 over a much larger data set of over 30 different cicada songs. It is clear that the approach proposed by Chaves *et al.* [2012] fails to distinguish between the cicada’s song and the bush cricket’s chirp, resulting in poor precision and recall statistics. Conversely, both the precision and recall metrics for our proposed approach are close to 1, as a result of our model’s ability to use the periods between the bush cricket’s chirps to differentiate between the two songs. Furthermore, the vastly greater precision and recall metrics for our proposed approach have resulted in a greater  $F_1$  score. Our approach’s  $F_1$  score can be interpreted as a suitable trade off between false detections and missed detections.

It is also worth comparing the computational efficiency of the approach used by Chaves *et al.* [2012] to the approach described in this paper. In the Chaves *et al.* [2012] model, the two most costly operations, namely the sound detection algorithm and the computation of the cepstral coefficients, both require an order  $O(N \log N)$  to compute, with  $N$  being the number of samples in the recording. In comparison, the entire feature extraction process in our proposed model only requires  $O(N)$  operations. This complexity corresponds to a computation time of 537s for the Chaves *et al.* [2012] approach, while our approach takes 45s to process the recording of length 311s, shown in Figure 7. Since the Chaves *et al.* [2012] method takes longer to run than the length of the recording, clearly it is not efficient enough to run in real-time. In comparison, our approach processed the whole recording in one seventh of the recording time, and therefore is suitable to run in real-time. These values, although dependent on implementation details, corroborate the hypothesis that the former model has a considerably higher computational complexity, as shown in Section 2. This, together with the increased robustness to noise shown by the accuracy metrics, allows us to conclude that our model is better suited to real-time detection than the state of the art for insect classification. The



Figure 8: Prototype app interface showing the real-time feedback presented to the user during a 60 second survey, indicating the detection of a cicada song, and the electronic cicada that replicates the sound of the cicada, used for user testing.

execution times of both approaches were evaluated on a mid-range modern computer (Intel Core 2 Duo CPU, 2.4 GHz, 8 GB RAM), with the software entirely written in Python.

## 4 Conclusions

In this paper we have presented a novel automated insect detection algorithm that, to the best of our knowledge, is the first targeted at real-time identification of selected species. We have shown that with a careful analysis of the call to be detected, key features can be extracted at minimal cost, greatly simplifying the identification process. We compared our approach with a state-of-the-art technique, and identified scenarios where such a technique would fail to distinguish two given calls in a smartphone recording.

Our results show that the proposed system achieves an accuracy of  $F_1 = 0.955$  on a data set of recordings taken with an Apple iPhone 4S at 44,100 kHz in their original soundscape, which includes various forms of background noise and different animal calls, as well as human voice, interfering with the signal. Rather than focusing on batch processing of large data sets of species, our approach aims at identifying a small number of species in real time.

With the development of the robust acoustic classifier complete, we have integrated the technology into the prototype smartphone app shown by Figure 8. We are currently carrying out user testing of the app through the use of an electronic cicada, a stand-alone circuit board and speaker that replicates the sound of the New Forest cicada’s song. The project will launch publicly in June 2013, and will constitute the first live crowdsourced system to search for an endangered species of insect of national importance.

## Acknowledgements

This research was supported by an EPSRC Doctoral Training Centre grant (EP/G03690X/1) and by the ORCHID Project, www.orchid.ac.uk (EP/I011587/1). Many thanks to Dr Tomi Trilar and Prof Matija Gogala for their field guidance in Slovenia and to Dr David Chesmore for his original suggestion and ongoing support.

## References

- [Blasiak and Rangwala, 2011] S. Blasiak and H. Rangwala. A hidden markov model variant for sequence classification. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 1192–1197. Barcelona, Catalonia, Spain, 2011.
- [Chaves *et al.*, 2012] V. A. E. Chaves, C. M. Travieso, A. Camacho, and J. B. Alonso. Katydid acoustic classification on verification approach based on MFCC and HMM. *Proceedings of the 16th IEEE International Conference on Intelligent Engineering Systems (INES)*, pages 561–566, 2012.
- [Chesmore and Ohya, 2004] E. D. Chesmore and E. Ohya. Automated identification of field-recorded songs of four British grasshoppers using bioacoustic signal recognition. *Bulletin of Entomological Research*, 94(04):319–330, 2004.
- [Chesmore, 2004] E. D. Chesmore. Automated bioacoustic identification of species. *Anais da Academia Brasileira de Ciências*, 76(2):436–440, 2004.
- [Goertzel, 1958] G. Goertzel. An algorithm for the evaluation of finite trigonometric series. *The American Mathematical Monthly*, 65(1):34–35, 1958.
- [Joint Nature Conservation Committee, 2010] Joint Nature Conservation Committee. UK priority species pages *Cicadetta montana* (New Forest Cicada). Technical report, 2010.
- [Leqing and Zhen, 2010] Z. Leqing and Z. Zhen. Insect Sound Recognition Based on SBC and HMM. In *International Conference on Intelligent Computation Technology and Automation (ICICTA)*, volume 2, pages 544–548, 2010.
- [MacLeod, 2007] Norman MacLeod. *Automated Taxon Identification in Systematics: Theory, Approaches and Applications*. CRC Press, 2007.
- [Pinchen and Ward, 2002] B. J. Pinchen and L. K. Ward. The history, ecology and conservation of the New Forest Cicada. *British Wildlife*, 13(4):258–266, 2002.
- [Quinn *et al.*, 2011] J. Quinn, K. Leyton-Brown, and E. Mwebaze. Modeling and monitoring crop disease in developing countries. In *Proceedings of the Twenty-Fifth Conference of the Association for the Advancement of Artificial Intelligence*. San Francisco, CA, USA, 2011.
- [Sun *et al.*, 2009] X. Sun, T. Matsuzaki, D. Okanohara, and J. Tsujii. Latent variable perceptron algorithm for structured classification. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*, pages 1236–1242. Pasadena, California, USA, 2009.