A ROBUST AUTOMATIC BIRD PHRASE CLASSIFIER USING DYNAMIC TIME-WARPING WITH PROMINENT REGION IDENTIFICATION

Kantapon Kaewtip[1], Lee Ngee Tan[1], Abeer Alwan[1], Charles E.Taylor[2]

[1]Department of Electrical Engineering, [2]Department of Ecology and Evolutionary Biology,
University of California, Los Angeles, California, USA

ABSTRACT

In this paper, we present a novel approach to birdsong phase classification using template-based techniques suitable even for limited training data and noisy environments. The algorithm utilizes dynamic time-warping and prominent (high-energy) time-frequency regions of training spectrograms to derive templates. The algorithm is evaluated on 32 classes of Cassin's Vireo bird phrases. Using only three training examples per class, our algorithm yields a phrase accuracy of 96.23%, outperforming other classifiers (e.g. 85.21% classification accuracy of SVM). In the presence of additive noise (10 dB SNR degradation), the proposed classifier does not degrade significantly, compared to others.

Index Terms— bird phrase classification, limited data, dynamic time-warping, noise-robust, template-based

1. INTRODUCTION

Birdsongs typically comprise a sequence of smaller units, termed phrases, separated from one another by longer pauses. Automatic recognition systems of bird sounds are needed, among other things, to annotate large amounts of birdsong recording data [1]. Automatic bird-phrase recognition is challenging due to within-class variability, limited training data, and noisy environments [2]. Two spectrograms with identical class labels may look different due to time misalignment and frequency variation. In real recording environments such as in a tropical forest, the data can be corrupted by background interference, such as rain, wind, other animals or even other birds vocalizing. A noise-robust classifier needs to handle such conditions.

Techniques such as support vector machines [3-4], sparse representation [2], HMMs [5-6], and dynamic time-warping (DTW) [6-7] have been used for automatic birdsong classification. Correlation-optimized DTW has been used in other fields [18-19]. Studies in [6] show that, under noisy recording conditions, "good performance of the DTW based-techniques requires careful selection of templates that may demand expert knowledge", while HMMs need "many more training examples than DTW templates." Some algorithms have been designed to reduce noise in bird songs [8-9] based on signal enhancement techniques, such as spectral subtraction [10]. Another noise-robust processing technique, commonly used in speech processing, is mask-based [11-12]. Generally, a mask is estimated from testing samples and used for enhancing the test features. In [13], a mask is obtained during both training and testing and is used as a feature for species classification. Another related idea is the glimpsing model of speech where the speech energy is sparse in the time -frequency space [14]. The glimpsing model can be valid for bird vocalization whose frequency coverage, in general, ranges from 1 kHz – 20 kHz but only a few ranges of hundred Hz contain significant energy at a particular time. This prominent time-frequency region is abbreviated as *prominent region* throughout this paper. In speech processing, a similar approach employs a set of spectro-temporal rectangular patches for discriminative word-spotting [15].

Template-based classifiers are appealing as time-alignment can be integrated with noise-robust processing. In our methodology, the Spectrogram-Fusion Algorithm derives a prominent region from training samples using DTW, in an iterative fashion. A contribution here is that our training procedure automatically derives a good template, bypassing manual selection. To achieve this, the algorithm aligns all training spectrograms with respect to one another and attempts to extract a reliable template using the prominent regions. In our classifier architecture, each class has one template, which comprises three entities: a spectrogram, a prominent region description, and a weighting function. A weighting function assigns more weights to reliable frames based on short-time correlation. In our testing procedure, these three entities are used by a DTW scheme to measure the similarity between a given test sample and a class template. The class template that achieves maximum similarity is identified as the classification output.

Section 2 briefly presents the database used, while Section 3 elaborates on the implementation of the proposed classifier. Sections 4-6 describe the experimental framework and present results along with a discussion and ideas for future work.

2. SOUND DATA

All experiments in this paper use the audio recordings described in [2]. Song fragments (phrases) for classification were obtained from recordings of Cassin's Vireo (*Vireo cassinii*). All recordings were obtained in 2010, in a mixed conifer-oak forest, near Volcano, California, at a sampling rate of 44.1 kHz. Manual annotation was performed using Praat (http://www.fon.hum.uva.nl/praat) to note the phrase class. The annotations have been updated to account for mislabeling. In this paper, the subset used for phrase classification consists of the same 32 classes as in [2].

3. PROPOSED ALGORITHM

3.1. Spectrographic extraction

First, the sampling rate is reduced from 44.1 kHz to 20 kHz because the energy above 10 kHz is relatively small. A highpass filter at 1 kHz cutoff is applied to the signal to eliminate irrelevant

signals because the energy of the signals for these birds below 1 kHz is absent. The range of energy can be specified according to the species being classified. The short-time 512-point FFT was performed using a frame length of 9 ms and a frames shift of 3 ms; then the magnitude of the Fourier transform is obtained while the phase information is discarded, resulting in a spectrogram.

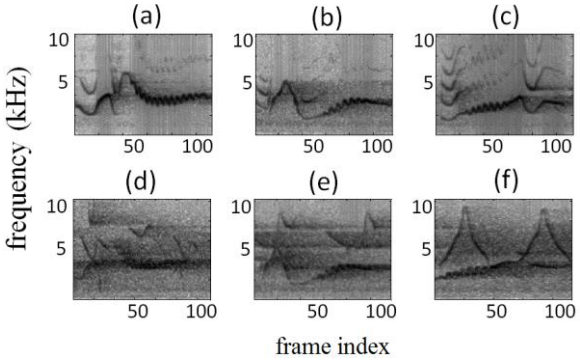

Fig. 1: Spectrograms of clean (a - c) and noisy samples (d - f). Spectrograms in the same columns (e.g. (a) and (d)) have the same class labels.

3.2. Prominent time-frequency regions

When a birdsong recording is corrupted by background interference, the accuracy of classifiers may degrade significantly. Fig. 1 shows examples of spectrographic mismatch for some random phrases extracted from a real recording. Spectrograms of the same columns have the same class labels (i.e. Fig. 1a and Fig. 1d are from the same class). The top images represent clean spectrograms and the bottom images are spectrograms of the same phrase class as above but corrupted by background interference.

High-energy regions in both clean and noisy spectrograms form a distinctive feature of a given class, as these regions are somewhat invariant when corrupted by noise. A low energy region, on the other hand, is not a reliable discriminative cue for classification. For example, the region above 5 kHz in Fig. 1b has low energy while this region apparently has high energy in Fig. 1e resulting in a spectrographic mismatch. However, if we reduce the scope of attention to a portion of the spectrogram image (rather than the entire image), the mismatch can be reduced. In this example, Fig. 1b and Fig. 1e are more similar if only the region below 5 kHz is considered.

In our algorithm, we use a better representative region rather than a rectangular patch. For example, the region enclosed by the dotted boundary in Fig. 2d represents the prominent region of the spectrogram in Fig. 2a. In this paper, we denote the prominent region of a spectrogram S as $R = \phi(S)$. Let S be a spectrogram and S_i denote the i^{th} column vector of S or simply the vector representing the spectrum at frame i. To derive $R = \phi(S)$, for each frame spectrum S_i, we first determine the maximum amplitude $\lambda_i = \max(S_i)$, and assign a value 1 to $R_i(k)$ if $R_i(k)$ is greater than a threshold $0.2\lambda_i$, where k is the frequency index. Then we expand this interval by 0.5 kHz. A more sophisticated algorithm to derive the prominent region can be explored in subsequent studies; the focus of this paper is to present the effectiveness of the prominent region rather than studying the optimal region derivation. Fig. 2e and Fig. 2f illustrate the pixels of the spectrogram from Fig. 2b and Fig. 2c, respectively, supported by the prominent region shown in Fig. 2d. The process of deriving a prominent region is performed only for the training template; we do not derive the prominent region of the test data.

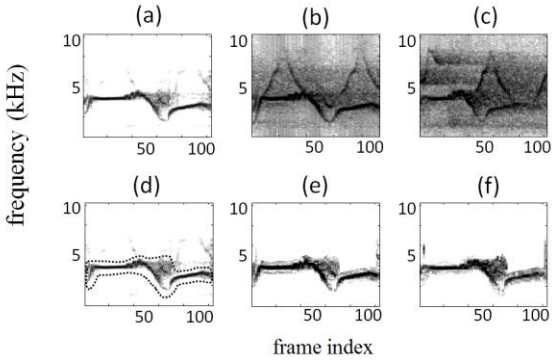

Fig. 2: Illustration of prominent regions. For a reference spectrogram (a), the prominent region is the region enclosed by the dotted boundary in (d). For spectrograms (b) and (c), Figs. (e) and (f) shows the pixels in the corresponding prominent regions, respectively.

3.3. Dynamic time-warping (Procedure I)

Two spectrograms, $S^{(1)}$ and $S^{(2)}$, of the same phrase may have different durations that cannot be aligned by a simple shift so a dynamic time warping (DTW) [16][20] is incorporated into our framework. In [17], the cosine similarity is shown to be a good metric for a DTW scheme. Let us define a notation $\theta(u,v) = \frac{u^T v}{|u||v|}$ as the cosine similarity degree between vectors u and v. The value of the cosine similarity is always in the range of [0,1] if all elements of u and v are non-negative; the closer to 1, the more similar u and v are. In our algorithm, the cosine similarity is used in our DTW scheme to measure the overall similarity of two spectrograms.

Procedure I: Dynamic time-warping (p,X',c) = DTW(M,R,w,X)

- i and j are the time indices of the reference M (with N_M frames) and test X (with N_X frames), respectively.
- C(i,j) is the cosine similarity between the i^{th} frame of M and the j^{th} frame of X.
- P(i,j) is the intermediate cumulative score.
- The operator $\odot$ is the element-wise multiplication.
- c is the vector of frame-wise cosine similarities of M and X'.

1) $C(i,j) = \theta(M_i \odot R_i, X_j \odot R_i)$
2) $P(1,j) = C(1,j)$ for $j \le \text{floor}(0.1N_X)$
3) $w_i = w_i/(w_1 + w_2 + ... + w_{N_M})$
   $P(2,j) = \max\{P(1,j)+w_2C(2,j), P(1,j-1) + w_2C(2,j)\}$ for $j>1$
   Recursive step

$$P(i,j) = \max \begin{cases} P(i-1,j-2) + 0.5w_iC(i,j-1) + 0.5w_iC(i,j) \\ P(i-1,j-1) + w_iC(i,j) \\ P(i-2,j-1) + w_{i-1}C(i-1,j) + w_iC(i,j) \end{cases}$$

4) $p = \max\{P(N_M,j)\}$, $\text{floor}(0.9N_X) \le j \le N_X$
5) Backtrack the optimal path and obtain X' accordingly.
   $c_i = \theta(M_i \odot R_i, X_i' \odot R_i)$

DTW is used to find the optimal time warping function between a test spectrogram X and a reference spectrogram M so that the resulting spectrogram X' will have the same number of frames as M. Our DTW scheme is described in Procedure I and explained step by step as follows. 1) The local score C(i,j) of the DTW is the frame-wise cosine similarity between the i^{th} frame of M and j^{th} frame of X. The cosine similarity is not computed over the entire frequency range, but only on the range determined by the

prominent region of the reference frame R_i. 2) The optimal warping function is constrained to begin within the first 10% of the test frames. 3) A given reference frame is allowed to align with up to two test frames and vice versa; for this reason we employ DTW type I [16]. In computing the cumulative score, each reference frame is weighted differently based on the frame weight input vector w of the DTW such that the weights sum to 1 ($\sum_{i=1}^{N_M} w_i=1$). Depending on situations, the weight vector w can be determined in several ways, some of which will be described in Section 3.4. 4) The optimal path is backtracked ensuring that at least 80% of the test frames are accounted for. 5) Along with the average similarity p, the DTW also outputs the aligned spectrogram X' and the corresponding vector of frame-wise similarities c. All 3 outputs (p,X',c) are needed for the training process while only the overall similarity p is needed for testing.

3.4. Training procedure (Procedure II)

It is important to design an algorithm that extracts common features from training samples and discards noise components, resulting in a good template. A *template* is defined as a collection of three *attributes*: a spectrogram reference M, a prominent region R and a weight function w. Our spectrogram fusion algorithm takes N training spectrograms per phrase class, $T = \{T^{(1)}, T^{(2)}, \ldots, T^{(N)}\}$, and outputs a template model $(\hat{S},\hat{R},\hat{w})$ that represents common features among the training samples in each case. This procedure is performed individually for each class.

Procedure II: Spectrogram Fusion $(\hat{S},\hat{R},\hat{w}) = \Psi(T)$

for trial n = 1:N

1) $S = T^{(n)}$, $R = \phi(S)$, $w_i = \max(S_i)$ where S_i is the i^{th} frame of S

2) Recursive step: repeat the following blocks for 3 times

for m = 1 to N

$(p^{(m)}, \tilde{T}^{(m)}, c^{(m)}) = \mathrm{DTW}(S,R,w, T^{(m)})$

end

$S_i(k) = \mathrm{median}(\tilde{T}_i^{(1)}(k), \ldots, \tilde{T}_i^{(N)}(k))$ for each i and k

$R = \phi(S)$, $c_{i,ave} = \frac{1}{N}\sum_m c_i^{(m)}$, $w_i = c_{i,ave}^{10}$

3) $p_{ave}^{(n)} = \frac{1}{N}\sum_m p^{(m)}$, $S^{(n)} = S$, $R^{(n)} = R$.

end

$\hat{n} = \mathrm{argmax}_n p_{ave}^{(n)}$

$\hat{S} = S^{(\hat{n})}$, $\hat{R} = R^{(\hat{n})}$, $\hat{w} = w^{(\hat{n})}$

Procedure II is explained as follows. 1) The reference template (S,R,w) is initialized with one training sample, say $T^{(n)}$. Specifically, $S = T^{(n)}$, $R = \phi(T^{(n)})$, and w is determined by the frame amplitudes. 2) This template is used as a reference in the DTW (Section 3.3) and each training sample is used as a test. In other words, we perform $(p^{(m)}, \tilde{T}^{(m)}, c^{(m)}) = \mathrm{DTW}(S,R,w, T^{(m)})$ for all m = 1,2, …, N. Then the each i^{th} frame of the updated spectrogram $S_i(k)$ is taken to be the median values of $\tilde{T}_i^{(1)}(k)$, $\tilde{T}_i^{(2)}(k)$, … , $\tilde{T}_i^{(N)}(k)$. The purpose of this operation is to align invariant components and to discard outliers contributed from noise or within-class variability. The updated prominent region R is derived accordingly from the new S. The weight should be assigned to the frames that have high similarities, so we use c_i to compute the new weight w_i. This new template (S,T,w) is then used as a reference in the DTW to generate another new template by the same procedure. We found that using only 3 iterations is sufficient for this data set.

3) After the final iteration, we compute the average similarity $p_{ave}^{(n)}$ between the derived template and each training sample to measure the effectiveness of the final template. If an unreliable (e.g. noisy) spectrogram happens to be the initial template, the resulting model maybe unreliable. For this reason, the *Spectrogram-Fusion Algorithm* performs N trials with different initial templates from the same class. Finally, the algorithm selects the template from the trial whose average similarity p_{ave} is the highest. The template $(\hat{S},\hat{R},\hat{w})$ generated from this trial is assigned for that particular phrase class.

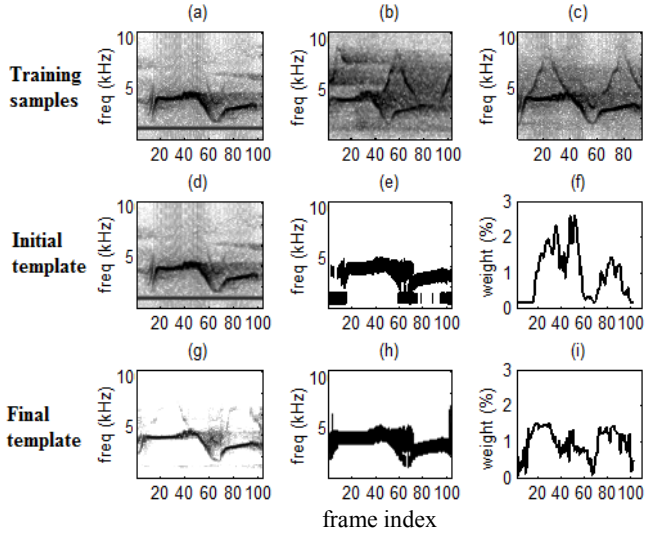

Fig. 3: Illustration of Procedure II. Training samples are in the first row. The initial and final templates are shown in the second and third row, respectively. For the first iteration, the template is based on only the information of sample (a) while templates of each subsequent iteration are based on all samples (a-c). Figs. 3d and 3g are template spectrograms, Figs. 3e and 3h are prominent regions, and Figs. 3f and 3i are frame weights.

Fig. 3 illustrates Procedure II that uses spectrograms (a), (b), and (c) as a training set (T = {a,b,c}). Consider Trial 1 where Fig. 3a is selected as the initial reference template. The template attributes are derived as S = a, R = ϕ(a), and w is determined by the frame amplitudes of S. These initial spectrogram, prominent region, and weight function are shown in Figs. 3d, 3e, and 3f respectively. After 3 iterations (including (b) and (c)), the final template attributes are shown in Figs. 3g, 3h, and 3i. In fact, the final template of this trial (Trial 1) yields the highest p_{ave} among all 3 trials, so this final template is essentially the output of Procedure II. Note that although all training spectrograms are corrupted by noise, the Spectrogram-Fusion Algorithm is able to capture most of the reliable content. In Fig. 3g, although some noise components remain in the final spectrogram template, they will not affect the performance of the DTW if they are not in prominent regions.

3.5. Testing Procedure

For a given segment, the spectrogram is derived as described in Section 3.1. Then the spectrogram is used to compute the similarity with each class template as described in Section 3.3. The overall similarity between a template and a test is in the range of [0,1]. The class that gives the highest similarity is identified to be the classification output.

4. EXPERIMENTAL SETUP AND EVALUATION FRAMEWORK

4.1. Comparison classifiers

Comparison algorithms for automatic birdsong classification are based on support vector machine (SVM) and sparse representations (SR). Both have been shown to be effective for limited training data [2]. In this paper, we compare the performance of the proposed classifier to those of SVM and SR. In [2], it has been shown that SR and SVM perform better with a higher feature dimension. Therefore, we use the maximum feature dimension for the SR and also use that number for the SVM classifier. The implementations are the same as described in [2].

4.2. Evaluation framework

The evaluation framework is the same as described in [2]. In short, we evaluated the performance of the classifiers under limited data conditions. We varied the number of training samples from 3 to 7 (for each of the 32 classes) and used the remaining samples (800-1000) for testing. In each case, we conducted 5 sub-experiments (training samples randomly selected) and averaged their results.

4.3. Testing conditions

We conducted two experiments to evaluate the performance of the proposed algorithm. In Experiment 1, we used the same recordings as in [2] (see Section 2). Experiment 2 evaluated all 3 classifiers in the presence of noise. The background noise was recorded from the ambient environments in the same location as when the birdsong occurred. There are total of 7 noise files (20 minutes long). These files contain birdsongs from other species as well as ambient noise. For a given phrase segment, the noise file ID and time location were selected randomly. Then the noise portion is scaled to generate a pseudo signal-to-noise ratio of 10 dB. Note that we cannot conclude the SNR is exactly 10 dB because the original files might contain noise energy as seen previously in Fig. 1. We found that enhancing the signal by spectral subtraction improves the performance of the SR and SVM classifiers. Therefore, for a fair comparison, we use the standard spectral subtraction from VOICEBOX [21] as a noise-robust processing tool before generating features for the SR and SVM classifiers.

5. RESULTS AND DISCUSSION

Figure 4 shows the results of Experiment 1 (original recordings) and Experiment 2 (noise added). Overall, the proposed algorithm always outperforms SR, while SR always outperforms SVM. The accuracy of SR and SVM generally increases when more training samples are used. Our template-based classifier is not highly sensitive to the number of training examples. Using only 3 training samples, it achieves an accuracy of 96.23% and 79.23% in fairly clean and noisy conditions, respectively. In Experiment 2, the proposed algorithm significantly outperforms both SR and SVM. There are several possibilities that account for such degradation in SVM and SR. First, the time alignment of the testing spectrogram and the SVM and SR models might not be accurate while DTW in the proposed algorithm reduced the time misalignment effect. Second, SVM and SR have certain optimal parameters and training conditions. For example, the kernel, the feature dimension and the cross-validation configuration might be sensitive factors to SVM. Third, noise estimation used in spectral subtraction is not sufficiently accurate. Template-based classifiers, on the other hand, can exclude irrelevant signal components based on the given template (e.g. using the prominent region). Note that pilot experiments using DTW with no prominent region identification showed worse performance than the proposed algorithm.

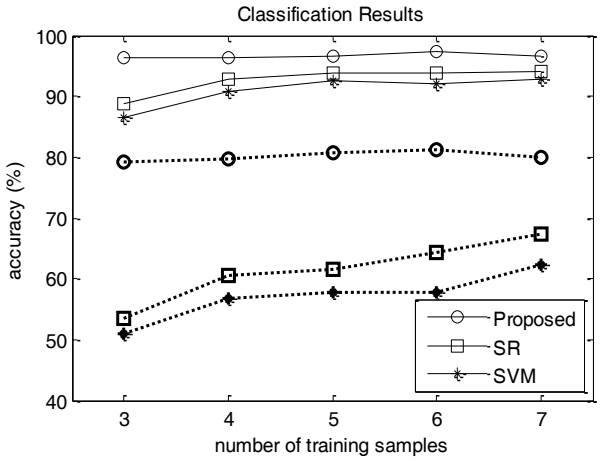

Fig. 4: Experimental results: the solid lines are for Experiment 1 and dotted lines, for Experiment 2.

6. RELATION TO PREVIOUS WORK

In this paper, we develop a birdsong phrase classifier that is robust to 1) duration variability, 2) limitedness of training data, and 3) noisy environments. For birdsong classification, DTW-based approaches [6-7] and SVM-based classifiers [3-4] can deal with limited training and duration variability, but require careful selection of training samples. Here, we propose a novel Spectrogram Fusion Algorithm that bypasses manual selection of training samples. Another novelty of this work is the way we use prominent time-frequency regions for achieving noise robustness in training as well as testing. Our technique is related to the "ordered spectro-temporal patch features" for keyword spotting [15], but employs a more systematic and knowledge-based approach for deriving the prominent regions.

7. CONCLUSIONS

A template-based algorithm for birdsong phrase classification is proposed. In a 32-class bird-phrase database, the proposed classifier obtains the highest classification accuracies compared to the SR and SVM classifiers. Using fairly clean recordings, our proposed algorithm achieves 96.23% outperforming SR (87.66%) and SVM (85.21%), with only three training examples per class. Using noisy recordings where the signals are degraded by approximately 10 dB, the performance of the SVM and SR degrades dramatically compared to the proposed algorithm. Our future work will include evaluations of the proposed classifier with other databases and different noise conditions. We are also interested in modifying our algorithm for keyword spotting in noise-robust ASR.

8. ACKNOWLEDGEMENTS

This study was supported in part by NSF Award No.0410438 and IIS-1125423. We thank George Kossan for his assistance with phrase identification.

9. REFERENCES

[1] T. Scott Brandes, “Automated sound recording and analysis techniques for bird surveys and conservation,” *Bird Conservation International*, vol. 18 , pp. S163–S173, 2008.

[2] L. N. Tan, K. Kaewtip, M. L. Cody, C. E. Taylor, and A. Alwan, "Evaluation of a Sparse Representation-Based Classifier For Bird Phrase Classification Under Limited Data Conditions," *Interspeech*, 2012.

[3] Seppo Fagerlund, "Bird species recognition using support vector machines," *EURASIP Journal on Advances in Signal Processing*, 2007, 2007.

[4] Miguel A. Acevedoa, Carlos J. Corrada-Bravoc, Héctor Corrada-Bravob, Luis J. Villanueva-Riverad, and T. Mitchell Aidea, “Automated classification of bird and amphibian calls using machine learning: A comparison of methods,” *Ecological Informatics*, Vol. 4, pp. 206–214, 2009

[5] Vlad M. Trifa, Alexander N. G. Kirschel, Charles E. Taylor, and Edgar E. Vallejo, “Automated species recognition of antbirds in a Mexican rainforest using hidden Markov models,” *The Journal of the Acoustical Society of America (JASA)*, vol. 123, 2424–2431, 2008.

[6] Joseph A. Kogan and Daniel Margoliash , “Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden Markov models: A comparative study,” *Journal of the Acoustical Society of America (JASA)*, vol. 103, pp. 2185–2196, 1998.

[7] Ken Ito, Koich Mori, and Shin-ichi Iwasaki, ‘‘Application of dynamic programming matching to classification of budgerigar contact calls,’’ *Journal of the Acoustical Society of America (JASA)*, vol. 100, 3947–3956, 1996.

[8] Forrest Briggs, Fern Xiaoli, and Raich Raviv. "Technical Report (Not Peer Reviewed): Acoustic Classification of Bird Species from Syllables: an Empirical Study."

[9] Wei Chu and Daniel T. Blumstein, “Noise robust bird song detection using syllable pattern-based hidden Markov models,” *IEEE ICASSP*, pp. 345–348, 2011.

[10] S. Boll. “Suppression of acoustic noise in speech using spectral subtraction,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, 27(2):113–120, 1979.

[11] Julien van Hout and Abeer Alwan, "A novel approach to soft-mask estimation and Log-Spectral enhancement for robust speech recognition." In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4105-4108. IEEE, 2012.

[12] Bhiksha Raj and Richard M. Stern. "Missing-feature approaches in speech recognition." *Signal Processing Magazine, IEEE* 22.5 (2005): 101-116.

[13]Forrest Briggs, Balaji Lakshminarayanan, Lawrence Neal, Xiaoli Z. Fern, and Raviv Raich, Sarah J. K. Hadley, Adam S. Hadley, and Matthew G. Betts, "Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach," *Acoustical Society of America Journal* 131 (2012): 4640.

[14] Martin Cooke, "A glimpsing model of speech perception in noise," *The Journal of the Acoustical Society of America,* 119 (2006): 1562.

[15] Tony Ezzat and Poggio Tomaso, "Discriminative word-spotting using ordered spectro-temporal patch features," *Proc. SAPA* (2008).

[16] Cory Myers, Lawrence R. Rabiner, and Aaron E. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on* , vol.28, no.6, pp. 623- 635, Dec 1980

[17] Brian King, Paris Smaragdis, and Gautham J.Mysore, "Noise-robust dynamic time warping using PLCA features," *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, vol., no., pp.1973-1976, 25-30 March 2012

[18] Giorgio Tomasi, Frans van den Berg,and Claus Andersson, "Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data," *Journal of Chemometrics* 18, no. 5 (2004): 231-241.

[19] Niels-Peter Vest Nielsena, Jens Michael Carstensenb, and Jørn Smedsgaarda, "Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping," *Journal of Chromatography A* 805, no. 1 (1998): 17-35.

[20] Hiroaki Sakoe and Chiba Seibi "Dynamic programming algorithm optimization for spoken word recognition." *Acoustics, Speech and Signal Processing, IEEE Transactions on* 26, no. 1 (1978): 43-49.

[21] M. Brookes, “Voicebox, Speech Processing Toolbox for MATLAB”, Department of EE, Imperial College, London, www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html