# A robust automatic birdsong phrase classification: A template-based approach[a)]

Kantapon Kaewtip[b)] and Abeer Alwan
*Department of Electrical Engineering, University of California, Los Angeles, 56-125B Engineering IV Building, Box 951594, Los Angeles, California 90095, USA*

Colm O'Reilly
*Sigmedia, Department of Electronic and Electrical Engineering, Trinity College, Dublin, Ireland*

Charles E. Taylor
*Department of Ecology and Evolutionary Biology, University of California, Los Angeles, 621 Charles Young Drive South, Los Angeles, California 90095, USA*

(Received 5 February 2016; revised 23 August 2016; accepted 18 October 2016; published online 15 November 2016)

Automatic phrase detection systems of bird sounds are useful in several applications as they reduce the need for manual annotations. However, birdphrase detection is challenging due to limited training data and background noise. Limited data occur because of limited recordings or the existence of rare phrases. Background noise interference occurs because of the intrinsic nature of the recording environment such as wind or other animals. This paper presents a different approach to birdsong phrase classification using template-based techniques suitable even for limited training data and noisy environments. The algorithm utilizes dynamic time-warping (DTW) and prominent (high-energy) time-frequency regions of training spectrograms to derive templates. The performance of the proposed algorithm is compared with the traditional DTW and hidden Markov models (HMMs) methods under several training and test conditions. DTW works well when the data are limited, while HMMs do better when more data are available, yet they both suffer when the background noise is severe. The proposed algorithm outperforms DTW and HMMs in most training and testing conditions, usually with a high margin when the background noise level is high. The innovation of this work is that the proposed algorithm is robust to both limited training data and background noise.
 [http://dx.doi.org/10.1121/1.4966592]

[MLD] Pages: 3691–3701

## I. INTRODUCTION

Studies of animal biodiversity and song syntax would benefit greatly from an ability to identify species and classify phrase types automatically.[1–4] We focus here on automatic classification of songs by birds. These vary dramatically from simple notes to complex sequences with thousands of different song types.[5] One common structure of birdsongs is for several closely spaced "notes" to be grouped into "phrases," separated from one another by short time intervals, thus, long sequences of phrases comprise a "song." For example, Cassin's Vireos (*Vireo cassinii*; CAVI) have songs made up of phrases, short bursts of notes, <0.7 s in duration, separated by 1 s of silence or more. Each bird will typically have 40–60 phrase types[6] that can be reliably distinguished manually by an observer; doing so automatically is challenging due to within-class variability, limited training data, and noisy environments. This problem shares many features with speech processing in human, while presenting new challenges of its own.

Birdsongs become especially challenging when the song repertoire is diverse: some species have thousands of distinct phrases in their lexicons. Two spectrograms with identical class labels may look different due to time misalignment and frequency variation. The frequency distribution of birdsong elements often resembles a Zipf-Mandelbrot distribution where some phrases appear many times, while others appear sparingly. Thus, it is important to have an automated system that can correctly classify birdsongs and can be trained with only a few samples per phrase. Furthermore, the amount of available training data may be limited by the logistics of the recording procedure. The lack of human annotation may also limit the amount of training labels even when more recordings are available. The manual annotation labor can be reduced if an automatic classifier is able to correctly identify phrase types while requiring few training data.

Another challenge of automatic phrase classification is background noise. In real recording environments, the data can be corrupted by background interference such as rain, wind, other animals, or even other birds vocalizing. Automatic birdsong systems may suffer from detecting non-target segments or segments that contain both the target phrases, as well as unwanted noise components. Most systems are sensitive to noise and demand "a low-clutter, low noise environment."[7] A noise-robust classifier needs to handle such adverse conditions that may be present in the training data and also in the actual deployment data.

[a)]Portions of this work were presented in "A robust automatic bird phrase classifier using dynamic time-warping with prominent region identification," *Proceedings of ICASSP*, Vancouver, Canada, May 2013 and "Bird-phrase segmentation and verification: A noise-robust template-based approach," *Proceedings of ICASSP*, April 2015.
[b)]Electronic mail: kkaewtip@ucla.edu

Techniques such as support vector machines (SVMs), sparse representation, hidden Markov models (HMMs), and dynamic time-warping (DTW) have been used for automatic birdsong classification.[8–14] Template-based approaches such as DTW are appealing because the segmentation can be performed by discarding speech frames that are not similar to the template. However, the limitation of DTW is that it would require numerous templates to capture the speech variability. To solve this problem, templates may be grouped using several techniques such as clustering.[15] HMMs, on the other hand, treat a speech or audio signal as a sequence of observations generated by a state machine. HMMs are described as "generative models" as the models learn the statistical distribution of acoustic features. HMMs employ the maximum likelihood (ML) criterion, which requires their estimated probability models to represent the actual distribution of data. However, this requirement is difficult to achieve, resulting in high performance degradation of speech recognition in mismatched conditions such as noisy environments or speaker variability since it is impossible to include all of those conditions when training HMMs.

Studies in Ref. 13 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 birdsongs based on signal enhancement techniques, such as spectral subtraction.[16–18] Another noise-robust processing technique, commonly used in speech processing, is mask based.[19,20] Generally, a mask is estimated from testing samples and used for enhancing the test features. In Ref. 21, 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.[22] The glimpsing model can be valid for bird vocalization whose frequency coverage, in general, ranges from 1 kHz to 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.

Template-based classifiers are appealing as time alignment can be integrated with noise-robust processing. In our methodology, the SFA derives, iteratively, a prominent region from training samples using DTW. 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 correlations. 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 II briefly presents the database used, while Sec. III elaborates on the implementation of the proposed classifier. Sections IV and V describe the experimental framework and present results along with a discussion and ideas for future work. In this paper, we study the performance of the proposed system, traditional DTW, and HMMs under several training and test conditions. The number of training samples is varied to investigate the algorithm's robustness to limited training data. Three training conditions—clean, multiconditional, and adverse training—are used to see how noise level affects mismatch. Adverse training simulates situations where most of the recordings are severely corrupted by background noise so the data available to train the system are mostly unreliable.

## II. THE CAVI DATABASE

The CAVI species is found commonly in many coniferous and mixed-forest bird communities in far western North America. Birdsong phrases for classification were obtained from song recordings of male CAVIs because only the males of this species give full songs. Their songs have been described as "…a jerky series of burry phrases, separated by pauses of $\geq$1 s. Each phrase is made up of 2 to 4 notes [syllables], with song often alternating between ascending and descending phrases…." The "song [is] repeated tirelessly, particularly when [the singing male is] unpaired…."[23] Figure 1 shows the spectrogram of a CAVI song segment containing two different phrases, each consisting of two syllables.

Manual phrase annotation was obtained by human expert annotators. Phrase identity and time boundaries of each phrase in the song were annotated based on visual spectrogram inspection using Praat software.[24] Phrase types were categorized based on their frequency trajectories on spectrograms, and the label of the phrase was assigned to the subjectively matching spectrogram in the CAVI phrase catalogue. Whenever a phase segment with a subjectively different frequency trajectory from the existing spectrograms was found, its spectrogram was added to the catalogue, and a new phrase label was created.

The recordings were obtained in a mixed conifer-oak forest at ~800 m elevation (38°29′04″N, 120°38′04″W), near the city of Volcano, California, USA. Each sound file was recorded in WAV-format, 16-bit, mono, 44.1 kHz sampling rate. Each file contains songs from a single targeted CAVI, with occasional vocalizations of other species in the background. One or more files were recorded per CAVI individual. More information about CAVI and the recording setup

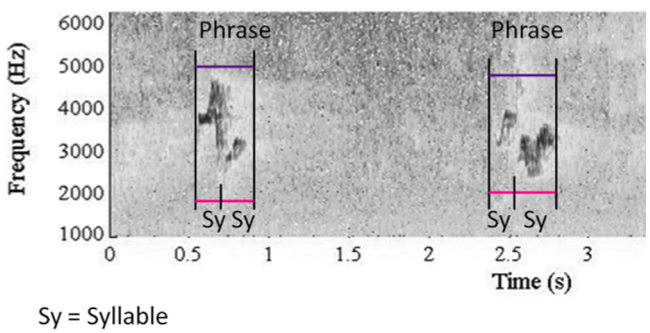


FIG. 1. (Color online) A spectrogram of a CAVI song segment. The spectrogram contains three phrases. Each phrase contains two syllables.

can be found in Refs. 6 and 11. All recordings with some metadata and the phrase catalogue are available online.[29]

## III. PROPOSED ROBUST TEMPLATE-BASED ALGORITHM

The proposed algorithm includes spectrogram generation (Sec. III A), prominent region identification (Sec. III B), noise-robust DTW (Sec. III C), and a SFA (Sec. III D). Only SFA is used in training, while all the others are involved in both training and testing.

### A. Spectrogram generation

Each sound file is first downsampled from 44.1 kHz to 20 kHz because the energy above 10 kHz is relatively weak. 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 fast Fourier transform (FFT) was performed using a frame length of 10 ms and a frame shift of 5 ms; then the magnitude of the Fourier transform is obtained while the phase information is discarded, resulting in a spectrogram.

### B. Prominent region identification

When a birdsong recording is corrupted by background interference, the accuracy of classifiers may degrade significantly. Figure 2 shows examples of spectrographic mismatch of some random phrases extracted from a real recording. Spectrograms of the same columns have the same class labels [i.e., Figures 2(a) and 2(d) 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. 2(b) has low energy while this region apparently has high energy in Fig. 2(e), 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, Figs. 2(b) and 2(e) are more similar if only the region below 5 kHz is considered.

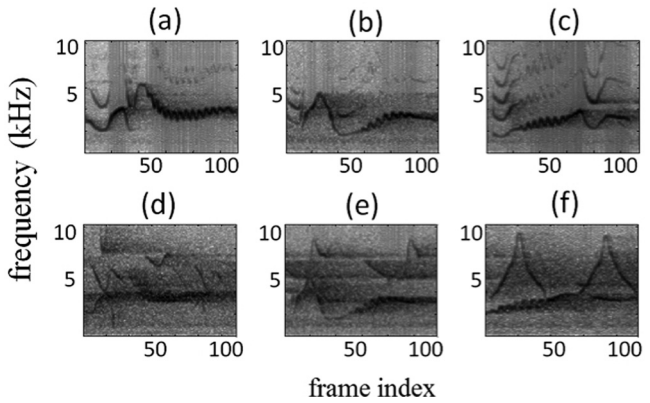


FIG. 2. Spectrograms of clean (a)–(c) and noisy samples (d)–(f). Spectrograms in the same columns [e.g., (a) and (d)] have the same class labels.

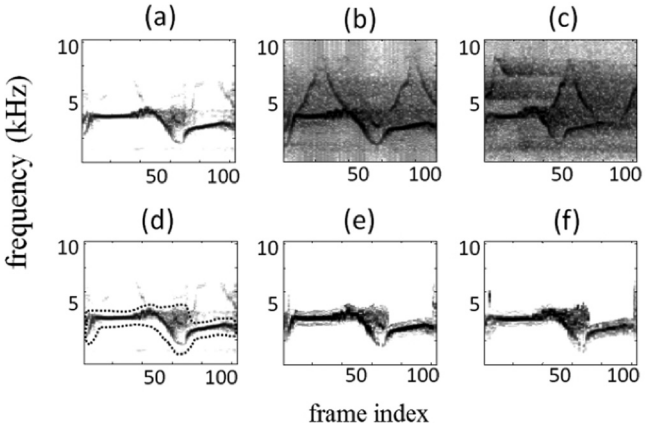


FIG. 3. Illustration of prominent regions. For a reference spectrogram (a), the prominent region is the region enclosed by the dotted boundary in (d). For spectrograms (b) and (c), (e) and (f) show the pixels in the corresponding prominent regions, respectively.

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. 3(d) represents the prominent region of the spectrogram in Fig. 3(a). 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 of 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 also be used as described in Sec. III D. However, the focus of this section is to present the effectiveness of the prominent region rather than to study the optimal region derivation method. Figures 3(e) and 3(f) illustrate the pixels of the spectrogram from Figs. 3(b) and 3(c), respectively, which are selected based on the prominent region shown in Fig. 3(d). 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.

### C. Noise-robust DTW

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 DTW is incorporated into our framework.[11,14] The dynamic time aligning component is shown to be essential to this classification task.[11] The cosine similarity is shown to be a good metric for a DTW scheme.[11,25] Let us define a notation $\theta(u, v) = u^T v/|u||v|$ as the cosine similarity degree between vectors $u$ and $v$ where $|\bullet|$ represents the $l_2$ vector norm. In our algorithm, the cosine similarity is used to measure the similarity of the spectra between a frame of a given spectrogram and a frame of a template. For noise-robustness, the cosine similarity is only computed within the prominent region.

**Procedure I: Robust DTW $(p, X', c) = \text{RDTW}(Y, R, w, X)$.**

Notations

- $i$ and $j$ are the time indices of the reference $Y$ (with $N_Y$ frames) and test $X$ (with $N_X$ frames), respectively.
- $w_i$ is the weight (importance values) of frame $i$
- $R_i$ is the prominent region of frame $i$
- $C[i,j]$ is the cosine similarity between the $i$th frame of $Y$ 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 $Y$ and $X'$
- $p$ is the overall similarity between $Y$ and $X'$

Summary of procedure

1. $C[i,j] = \theta(Y_i \odot R_i, X_j \odot R_i)$
2. $P[1,j] = C[1,j]$ for $j \leq \text{floor}(0.5N_X)$
3. $w_i = \dfrac{w_i}{w_1 + w_2 + \cdots + w_{N_Y}}$

$$P[2,j] = \max \begin{cases} P[1,j] + w_2 C[2,j] \\ P[1,j-1] + w_2 C[2,j] \end{cases} \tag{1}$$

4. Recursive step for $i \geq 3$

$$P[i,j] = \max \begin{cases} P[i-1,j-2] + 0.5w_i C[i,j-1] + 0.5w_i C[i,j], & \text{path 1} \\ P[i-1,j-1] + w_i C[i,j], & \text{path 2} \\ P[i-2,j-1] + w_{i-1} C[i-1,j] + w_i C[i,j], & \text{path 3} \end{cases} \tag{2}$$

$p = \max(P[N_Y,j], \text{floor}(0.5N_X)) \leq j \leq N_x$. Backtrack the optimal path and obtain $X'$ accordingly. $c_i = \theta(Y_i \odot R_i, X'_i \odot R_i)$

DTW is used to find the optimal time-warping function between a test spectrogram $X$ and a reference spectrogram $Y$ so that the resulting spectrogram $X'$ will have the same number of frames as $Y$ and also properly align with the template $M$. 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 $Y$ 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 $R_i$. These prominent regions are determined during the training process (procedure II in Sec. III D)

(2) The optimal warping function is constrained to begin within the first 10% of the test frames. The initial cumulative scores are taken from the cosine similarity scores of the first frame of the template and the allowed frames of the test spectrogram.

(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.[26] In Eq. (2), the cosine similarity values are weighted by 0.5 for path 1 so as not to double count the similarity score with the same frame of the reference spectrogram, i.e., $Y_i$. This makes $p$, the final cumulative score of the optimal path (between the reference and the test spectrogram), comparable across all testing spectrograms or samples. 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 Sec. III D.

(4) The optimal path is backtracked ensuring that at least 80% of the test frames are accounted for. Along with the average similarity $p$, the DTW also outputs the aligned spectrogram $X'$ and the corresponding vector of frame-wise similarities $c$. To obtain the time-warped test spectrogram that aligns with the template $X' = [x'_1, x'_2, \ldots, x'_{N_M}]$, the optimal path is backtracked as shown

$$x'_i = \max \begin{cases} \dfrac{1}{2}(x_j + x_{j-1}), & \text{path 1} \\ x_j, & \text{path 2} \\ x_j = x'_{i-1}, & \text{path 3.} \end{cases} \tag{3}$$

All 3 outputs $(p, X', c)$ are needed for the training process, while only the overall similarity $p$ is needed for testing.

### D. SFA

It is important to design an algorithm that extracts common features from training samples and discards noise components, resulting in a good template that represents the characteristics of the phrase class. A *template* $T$ is defined as a collection of three *attributes*: a spectrogram reference $Y$, a prominent region $R$, and a weight function $w$. Our SFA (see Fig. 4) takes $N$ training spectrograms per phrase class, $S = \{S^1, S^2, \ldots, S^{(N)}\}$, and outputs a template model $(\hat{Y}, \hat{R}, \hat{w})$ that represents common features among the training samples in each case. This procedure is performed individually for each class.

**Procedure II: Spectrogram fusion**

$$(Y, R, w) = \text{SFA}(S^1, S^2, \ldots, S^{(N)})$$

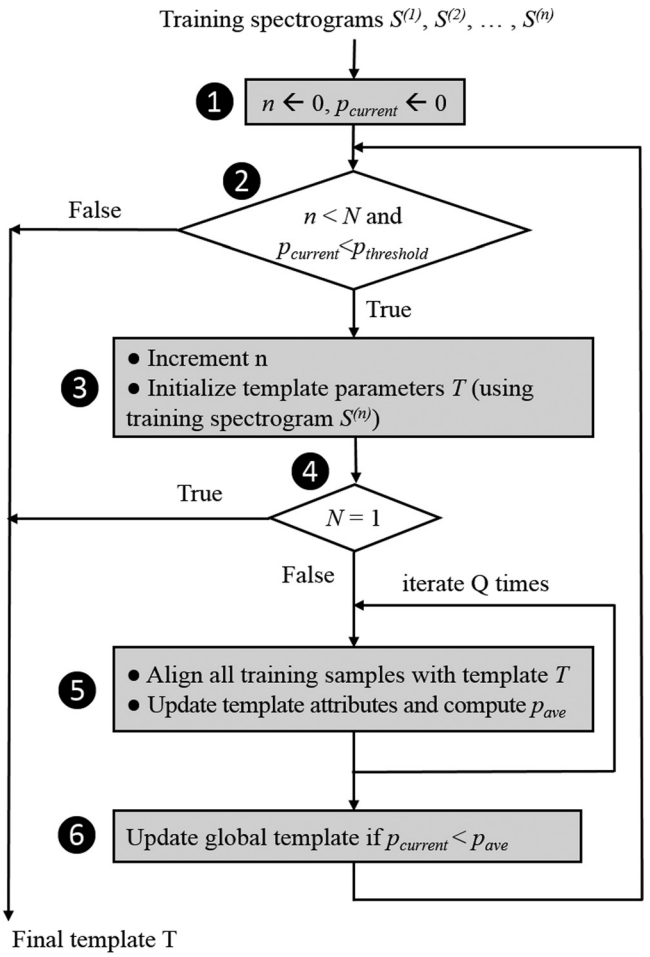


FIG. 4. An overview of the SFA. The input of the algorithm is a set of training spectrograms and the output is the template representing those spectrograms.

Procedure II is explained as follows.

This procedure derives a template $(Y,R,w)$ from a set of spectrograms. Let $N$ be the number of spectrograms. As an overview of the procedure, the algorithm picks a spectrogram from the same training set to construct an initial template. The template is then updated through several iterations. This procedure is repeated in a similar fashion in which some other spectrograms are used as initial templates. The best template, among all trials, is stored.

Steps 1 and 2: The variable $p_{\text{current}}$ keeps track of how well the current template represents all $N$ training spectrograms. For efficiency, the user can set $p_{\text{threshold}}$ as a satisfying score so that the procedure will terminate and return the current template when $p_{\text{current}} > p_{\text{threshold}}$.

Steps 2 and 3: The variable $n$ keeps track of the number of training samples that have been used to initialize a template. For each trial, a new training spectrogram is selected from the remaining list. In other words, spectrogram $S^{(n)}$ is selected at the $n$th trial and used as the initial template. Specifically, the spectrogram reference $Y$ is simply the selected spectrogram itself $[Y = S^{(n)}]$ and the prominent region is derived accordingly $R = \phi(Y)$.

Steps 4 and 5: If there is only one training sample, this initial template becomes the final template as there is no fusion to be carried out. If $N > 1$, the initial template is then used as a reference in the DTW (Sec. III C) and each training sample is used as a test. In other words, we perform $(p^{(m)}, \tilde{S}^{(m)}, c^{(m)}) = \text{RDTW}(Y, R, w, S^{(m)})$ for all $m = 1,2,\ldots,N$. The spectrograms should now be aligned with the template $Y$, i.e., frame $i$ should have similar spectral characteristics among $\tilde{S}^{(1)}, \tilde{S}^{(2)}, \ldots, \tilde{S}^{(N)}$. Then for each frame $i$ of each $\tilde{S}^{(m)}$, the spectrum is normalized so that the frame magnitude is 1 to make the algorithm invariant to energy level. The prominent region is then determined for each normalized spectrogram.

Now the template is updated as follows. For each time and frequency index $[k,i]$, the spectrogram reference is taken to be the median values of $\tilde{S}^{(1)}[k,i], \tilde{S}^{(2)}[k,i], \ldots, \tilde{S}^{(N)}[k,i]$. The purpose of this operation is to align invariant components and to discard outliers contributed from noise or within-class variability. The median value operation is robust when the noise level is excessive in some samples. For each time and frequency index $[k, i]$, the updated prominent region $R[k,i]$ is taken from the majority vote from $\tilde{R}^{(1)}[k,i], \tilde{R}^{(2)}[k,i], \ldots, \tilde{R}^{(N)}[k,i]$. This new template $(Y, R, w)$ is then used as a reference in the RDTW to generate another new template by the same procedure. We found that using only five iterations is sufficient for any type of data. If an unreliable (e.g., noisy) spectrogram happens to be the initial template, the resulting model may be unreliable. For this reason, the SFA performs several trials with different initial templates from the same class. At the end of the final iteration or at the end of each trial, the average similarity ($p_{\text{ave}}$) is compared with the highest similarity stored from the previous trials $p_{\text{current}}$. If the average similarity exceeds ($p_{\text{current}}$), the template is then updated to be the template of this trial.

Step 6: The procedure is repeated on until $p_{\text{current}}$ meets $p_{\text{threshold}}$ or the number of trials reaches $N$. (All training samples have already been used for initializations.) Finally, the algorithm selects the template from the trial whose average similarity ($p_{\text{ave}}$) is the highest. The template $(Y,R,w)$ generated from this trial is assigned to that particular phrase class.

**Example 1**: $\boldsymbol{N = 1}$. Suppose the training contains only spectrogram $a$ [Fig. 5(a)]. The template attributes are derived as $Y = a$, $R = \phi(a)$, and $w$ is determined by the frame amplitudes of $Y$ (Step 3). These initial spectrogram, prominent region, and weight function are shown in Figs. 5(d), 5(e), and 5(f), respectively. Since $N = 1$, the procedure terminates at Step 4 without going through Steps 5 and 6. Therefore, the final template attributes for this case are $Y = d$, $R = e$, and $w = f$ [Figs. 5(d), 5(e), and 5(f), respectively].

**Example 2**: $\boldsymbol{N = 3}$, $\boldsymbol{Q = 5}$. Suppose the training contains three spectrogram $S = \{a, b, c\}$ [Figs. 5(a), 5(b), and 5(c), respectively]. For the first trial ($n = 1$), the first training spectrogram ($a$) is selected to be the initial reference. Similar to example 1, the template attributes are then derived as $Y = a$, $R = \phi(a)$, and $w$ is determined by the frame amplitudes of $Y$ (Step 3). These initial spectrogram, prominent region, and weight function are again shown in Figs. 5(d), 5(e), and 5(f), respectively. However, since $N = 3$, the algorithm executes Step 5. That is, training spectrograms $a$, $b$, and $c$ are aligned with $Y$ and the new $Y$, $R$,

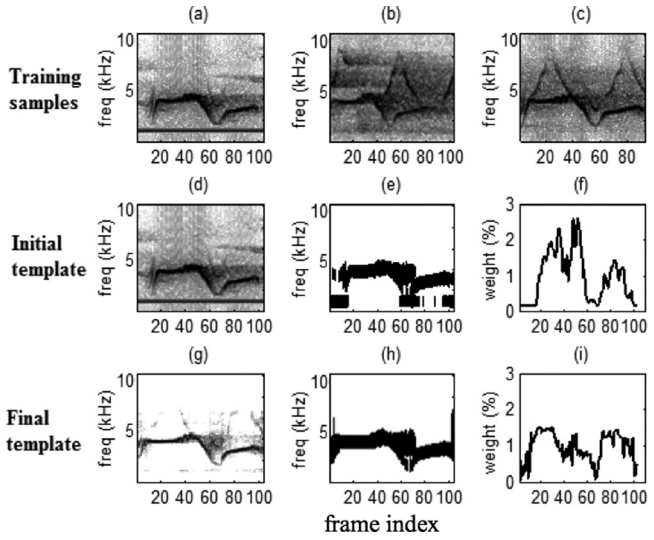


FIG. 5. Illustration of procedure II; Training samples {a,b,c}. In trial 1, sample (a) is selected as the initial reference [i.e., (d) is the same as (a)]. Next, the initial prominent region (e) and weight function (f) are derived from the spectrogram reference (d). After the initial step, training samples (b) and (c) are also used, iteratively, to refine the current template. The final template attributes are shown in (g), (h), and (i). Note that (d) and (g) are template spectrograms. (e) and (h) are prominent regions and (f) and (i) are frame weights.

$w$ are computed according to the alignment (Step 5). This procedure is repeated five times ($Q = 5$) and the last template and $p_{ave}$ are then stored as the global template and the current score, respectively. The global template attributes are shown in the last row where Figs. 5(g), 5(h), and 5(i) are the current global spectrogram reference, prominent region, and weight function, respectively. For the second trial ($n = 2$), $b$ is selected to be the initial reference and the procedure is carried on until Step 6. However, if the $p_{ave}$ from trial 2 happens to be less then $p_{ave}$ (from trial 1), the global template is not updated. After all training sets are used as initial templates, the final template is taken as the current global template. If the last template of trial 1 yields the highest $p_{ave}$ among all three trials, this template from trial 1 is essentially the output of procedure II.

### E. Phrase classification

For a given phrase, the spectrogram is derived as described in Sec. III A. Then the spectrogram is used to compute the similarity with each class template as described in Secs. III B and III C. 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. Note that the SFA is not in classification or testing.

## IV. EXPERIMENTAL SETUP AND EVALUATION FRAMEWORK

### A. Database

The training set is obtained from CAVI2013 (recorded in 2013), while the test set uses CAVI2014 (recorded in 2014). In 2013, CAVI songs were recorded in April, May, and June, resulting in 198 audio tracks of 4 h and 50 min recording. In 2014, CAVI songs were recorded in May and June, resulting in 438 audio tracks. The phrases that have at least 32 tokens in CAVI2013 and 10 tokens in CAVI2014 were selected for all experiments in this paper. There are 75 phrase classes that meet the criteria, and 32 samples are randomly selected from CAVI2013 for each phrase. Therefore training data comprise 2400 samples in total, while the test data comprise the same 75 phrases, each of which has 10 samples (750 total samples).

#### 1. Additive noise

To evaluate noise-robustness, we simulated noisy birdsongs by adding background noise at various signal-to-noise ratios (SNRs). The background noise was recorded in the same environment, when the target bird species was not singing. For a given recording segment, the time location was selected randomly to match the length of the recording. The noise portion is scaled to generate a pseudo SNR of a given SNR value. Note that this SNR represents the upper bound of the true SNR because the original files are not always completely noise free. The true quality of the signal may be worse than the SNR indicated.

#### 2. Train and test conditions

Clean and multiconditional training were included. For the clean training condition, the original recording (without noise added) was used to train each algorithm. For multiconditional training, each phrase segment was added with additive noise at 20 dB, 15 dB, 10 dB, and 5 dB. In other words, four other new recordings were generated from the original test recording. These four copies, together with the original signals, are used for multiconditional training.

Additionally, we included *adverse-condition training* where the training data are corrupted at 0 dB SNR. This condition simulates a scenario where most training data are severely corrupted by background noise possibly due to poor quality recording or adverse environments that strongly interfere with the vocalizing signal of the target species. The objective of this experiment is to evaluate the robustness of the algorithms if the only training data available are both limited and corrupted.

Another variable to investigate, beside the mismatched effect, is the number of training samples. Recall that $N$ is the number of training samples. Under each training condition, a different number of samples was used to train each phrase class: $N = 1$, 2, 4, 8, 16, and 32. Therefore, 18 sets of experiments (3 noise-level setups and 6 sample-size training conditions) were investigated in this paper.

Each experiment set was tested on six SNR conditions, i.e., 20 dB, 15 dB, 10 dB, 5 dB, 0 dB, and the clean condition. For a given test segment, each algorithm classifies which one of the 75 phrase types the segment belongs to. The average classification accuracy is observed from the 750 test samples for each SNR. In summary, each algorithm is tested in 108 train-test conditions ($3 \times 6 \times 6$).

### B. Comparative algorithms

Comparative algorithms for automatic birdsong classification are based on the two main frameworks of the generative learning approach—DTW and HMMs. Note that, most of the experiments involve limited training data (less than ten training samples per class) so deep-neural-network classifiers (which generally require more training data than the Gaussian Mixture Model–HMM framework) are not suitable for this study.

#### 1. DTW

For controlled components, the sample features, similarity score metric, and path configuration are identical to those of the proposed algorithm. However, the similarity scores are computed over the entire frequency range (unlike in the proposed algorithm, which computes the similarity only within the prominent regions). In addition, there is no spectrogram fusion. For a given test segment, the similarity score between its spectrogram and each training spectrogram is computed. The training sample that yields the highest similarity score is then used to map to the class label, resulting in phrase class identification.

#### 2. HMMs

HMMs were executed using the Hidden Markov Model Toolkit (HTK). We model 75 phrase types with 17 states per model, and each state is modeled using 1, 2, or 4 Gaussian mixtures (whichever gives the highest accuracy for each $N$) as this combination is observed to give the best results in a validation subset. Each model is left to right. The covariance matrices were diagonal. The pruning option $t$ of HRest set to 250.0 150.0 1000.0 as in the standard HTK benchmark.[27,28] Mel-frequency cepstral coefficients (MFCCs) were used as front-end features for HTK with standard parameters (25 ms frame size, 10 ms frame shift, 26 Mel filterbanks, 39 cepstral coefficients, including the first 2 derivatives).

## V. RESULTS AND DISCUSSION

In this section, classification accuracies (Acc.) of each classifier are presented. Three factors are analyzed for each classifier: the number of training data samples (1,2,4,8,16,32), the training condition (clean, multi-condition, and adverse), and the level of background noise of the test data set (0, 0 dB, 5 dB, 10 dB, 15 dB, 20 dB, and clean). The accuracies of comparative algorithms in 126 train-test conditions are presented in Figs. 6 and 7. Figures 6 and 7 share the same information but present different perspectives. Figure 6 presents only clean and 0 dB-SNR test conditions to illustrate the effect of noise in test data. Figure 7 presents the results when the systems are trained with $N = 32$ and 4 to illustrate the effect of limited training data.

Figure 6 shows the classification accuracies of comparative algorithms under six subsets of experiments. Each column is a set of experiments in different training conditions; clean [Figs. 6(a) and 6(b)], adverse-condition [Figs. 6(c) and 6(d)], and multicondition [Figs. 6(e) and 6(f)], respectively.

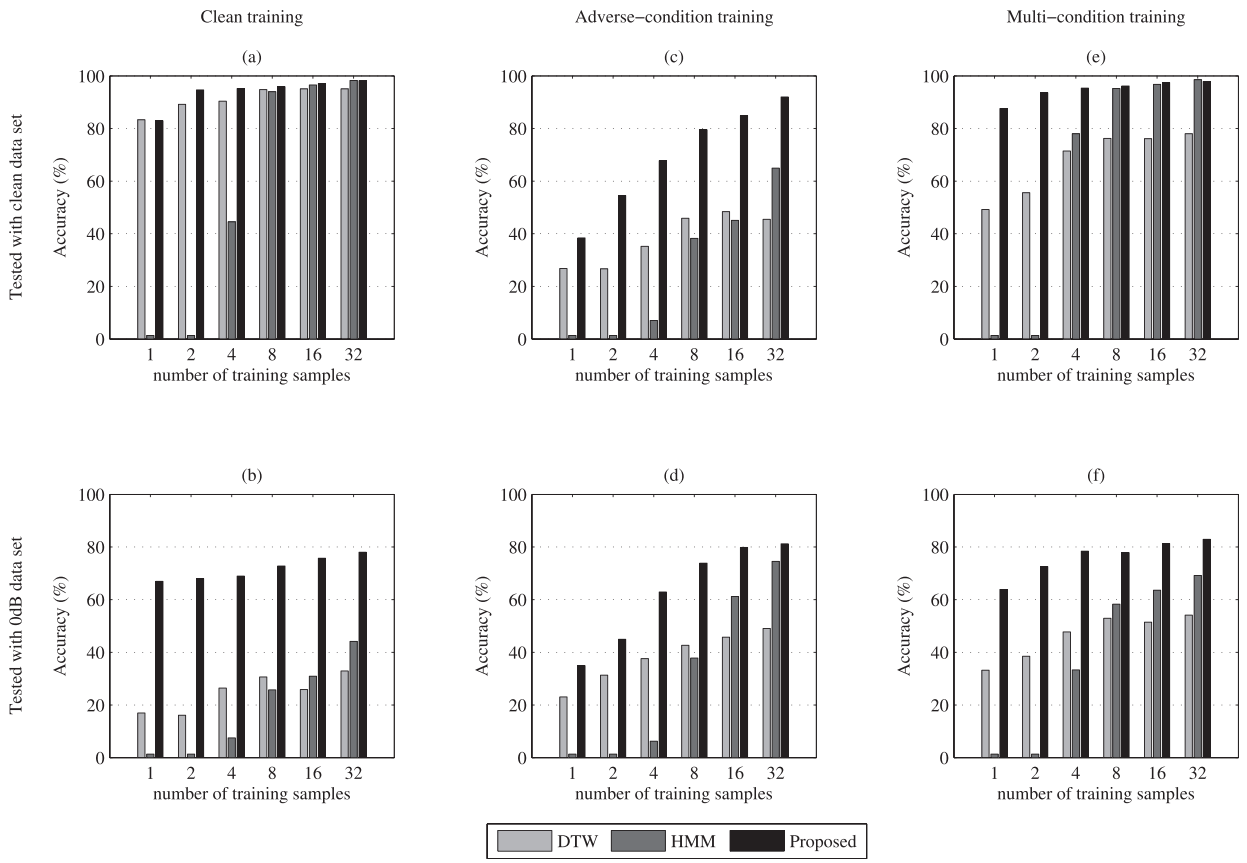


FIG. 6. Classification accuracies under different training and test conditions. The top and bottom rows show performance when testing with clean and 0 dB-SNR data, respectively.

The top plots [Figs. 6(a), 6(c), and 6(e)] and the bottom plots [Figs. 6(b), 6(d), and 6(f)] show the experiments when the systems are evaluated on the clean data set and the 0 dB data set, respectively. Within each subplot, each bar group is evaluated on the same noise condition and trained with the same training condition (clean, adverse-condition, or multi-condition), but with a different number of training samples.

Figure 7 shows the classification accuracies of comparative algorithms under six subsets of experiments. Same as that of Fig. 6, each column is a set of experiments in different training conditions; clean [Figs. 7(a) and 7(b)], adverse-condition [Figs. 7(c) and 7(d)], and multicondition [Figs. 7(e) and 7(f)], respectively. The top plots [Figs. 7(a), 7(c), and 7(e)] and the bottom plots [Figs. 7(b), 7(d), and 7(f)] show the experiments when the systems, however, are trained with 32 and 4 samples, respectively. Within each subplot, each bar group is evaluated on the same training condition but with a different SNR for the test data sets.

### A. Limited data

In this section, we analyze the relationship between the number of training data and the classification performance for comparative algorithms. Across all experiments, the performance of each algorithm generally increases as more training data are available but with a different rate of improvement and performance behavior. In most conditions, the performance of DTW starts with a decent performance at $N = 1$, generally goes up as the number of training samples increases but with a slow rate of improvement. The HMM classifier, on the other hand, yields poor classification when $N$ is lower than 8 (limited data) but starts to catch up and outperforms DTW when $N$ is 16 or more.

In the clean test-train condition [Fig. 6(a)], when the number of training samples is only one, DTW and the proposed algorithm yield reasonable performances of 83.33% and 83.93% Acc, respectively. The HTK setup results in a pure guess for the output ($1/75 = 1.33\%$ Acc.) due to the limitation of the statistical nature of the HMM algorithm. When the number of training samples increases to eight and above, the performance of HMMs increases significantly, while that of DTW increases slightly and plateaus around 95.7%. The proposed algorithm also consistently improves and its performance is comparable with HMMs when the number of training samples is high.

It can be observed that the accuracy of DTW may decrease even when the number of training data increase. In Fig. 6(b), when the number of training data is 8, 16, and 32, the accuracies go back down—30.67%, 25.87%, and 32.93%, respectively. This trend is also observed in the multiconditional training condition tested with 0 dB SNR data set [Fig. 6(f)]. The algorithm does not necessarily benefit from more training data especially in a mismatched testing condition. Misclassification may occur when test data are

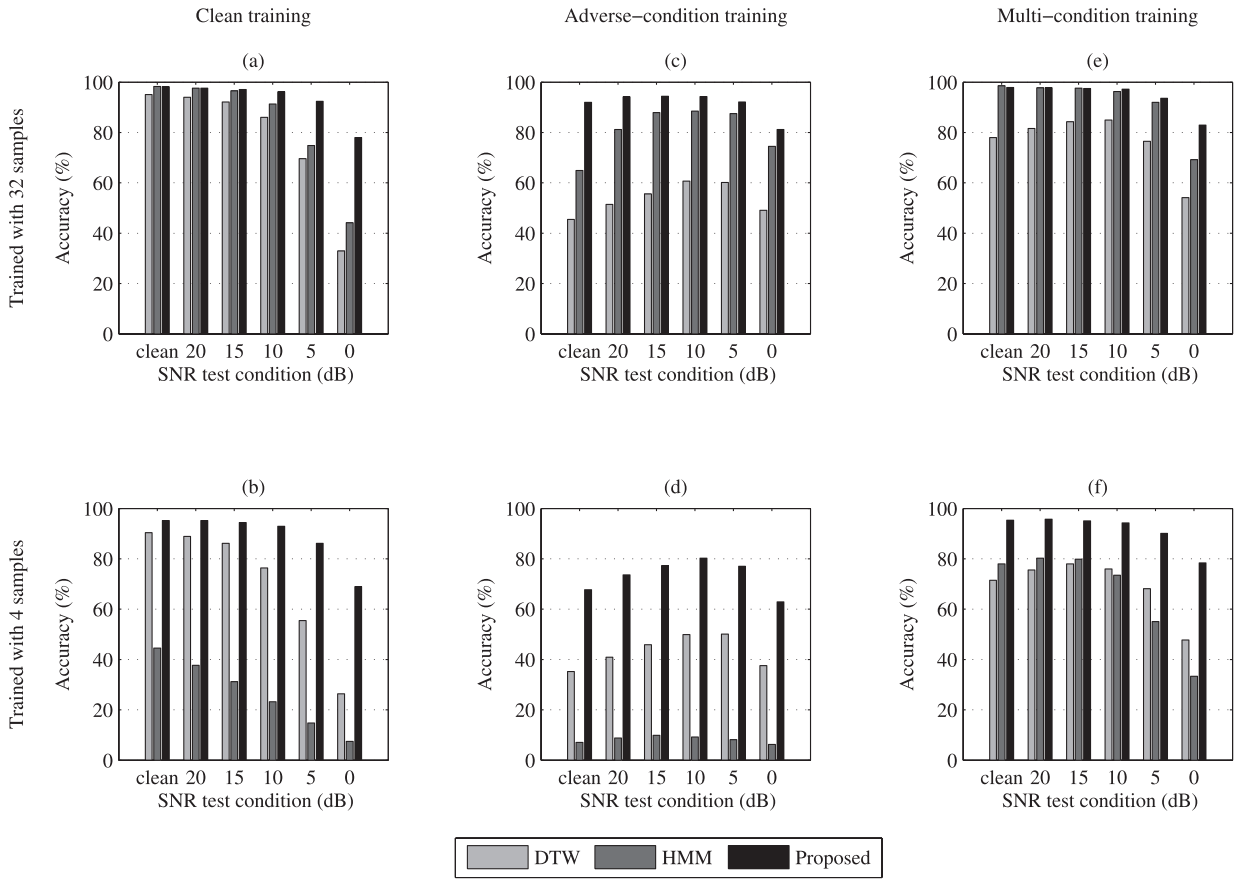


FIG. 7. Classification accuracies under different training and test conditions. The top and bottom rows show performance when training with 32 and 4 samples, respectively.

corrupted by background noise resulting in a signal that is similar to a certain class label, which is not the actual phrase class.

Figure 7 juxtaposes the performance of comparative algorithms using 32 and 4 training samples. When $N = 32$, the HMM algorithm outperforms DTW in all training and testing conditions [Figs. 7(a), 7(c), and 7(e)]. When $N = 4$, however, the HMM algorithm underperforms DTW in most training and testing conditions [Figs. 7(b), 7(d), and 7(f)]. Clearly, the strength of DTW is observed when the training data are limited, while the HMMs yield better performance when more training data are available.

In general, the proposed algorithm outperforms DTW and HMMs. The only case that slightly underperforms DTW is the clean train-test condition with one training sample. In this case, the only differences between DTW and the proposed algorithm are the prominent region and weight function, which may not be well estimated when the number of training sample is one. The proposed algorithm has a simple procedure to derive the prominent region and weight function when $N = 1$ (Sec. III D). When the training data are more available, however, the proposed algorithm is significantly more robust than DTW. There are a few cases where the HMMs outperform the proposed algorithm when $N = 32$ and tested with the clean data sets [Figs. 7(a) and 7(e)]. This shows that HMMs work well when the number of training data is high and the test condition is somewhat clean.

The performance trends of HMMs reflect the statistical nature of the algorithm. When the data are limited, the algorithm fails to capture the statistical model or to reliably estimate the parameters. However, if more data are available, the model generally represents the variation of the data more accurately. The DTW algorithm, on the other hand, has an advantage when the data are limited since there is virtually no parameter to estimate. However, the disadvantage is that its marginal improvement is minimal when more data become available. If the additional training data is corrupted, the traditional DTW may misclassify when there is a high similarity of the corrupted training sample and the test sample (due to noise distortion). Moreover, the computation required for DTW classification increases with the number of training samples. These two characteristics make DTW less appealing when the number of training data is high. The proposed algorithm is robust to limited training data but its models also improve when more data are available because the model is derived in a robust fashion (in the SFA). The computing time in classification for the proposed algorithm is essentially the same as HMMs, which is approximately $N$ times faster then DTW's computing time.

### B. Noise robustness

Background noise can interfere with both test and training data. Figure 6(b) shows the accuracies of the comparative algorithms with the same training conditions as Fig. 6(a). The only difference is that the testing condition is at 0 dB instead of the clean condition. The overall performance for each algorithm drops significantly. The dramatic degradation is observed in DTW and HMMs especially when the number of training data is limited. Using only one training sample per class, the performance of DTW drops from 83.33% (tested in clean) to 16.93% (tested with 0 dB-SNR data set). The performance of the proposed algorithm also drops but with less degradation from 82.93% to 66.93%. Such difference validates the importance of the prominent region because both DTW and the proposed algorithm essentially have the same spectrogram reference when $N = 1$ (Sec. III D). The performance of the proposed algorithm stands out in all cases. As previously discussed in limited training data, the model improves when more training data become available but the main strength of this algorithm is the noise-robust component.

With four training samples, the accuracy of HMMs drops by 83% (44.53% when tested in clean to 7.47% when tested with 0 dB data set). For the 0 dB-SNR test set, the performance of HMMs improves when more data are available but the accuracy is still at 44.13% even when the number of training samples is 32. Such dramatic degradation of over one factor (98.13% when tested with the clean data set to 44.13 when test with the 0 dB-SNR data set) shows that the HMM framework with MFCCs is not a noise-robust system in a mismatch condition even though the same $N$ (32) has been shown to be sufficient in the clean condition.

This trend is also observed in the multiconditional training [Figs. 6(e) and 6(f)]. The performance of HMMs improves when the data are more available but the accuracy is still 69.2% even when $N = 32$. In Fig. 6(f), HMMs may eventually catch up the proposed algorithm in multiconditional training but very large amounts of data may be required for the models to learn all the noise variations. Nevertheless, in the 0 dB testing condition, the degree of signal degradation is so high that the HMMs may fail to recognize the actual underlying clean component of the signal.

In multiconditional training and clean testing [Fig. 7(e)], the performance trend of DTW has a unique characteristic: the accuracy seems to increase for a higher SNR but it falls down eventually in spite of a better signal quality. One reason is that there are generally more training data at a moderate SNR. Therefore, a test sample at this SNR range can match with a training sample while a test segment at extremely high SNR (clean) or low SNR (0 dB) may have difficulty matching the training samples. Because the number of training data is low, it is impossible to generate all possible clean + noise combinations that reflect all the variations of signals that are effected by additive noise.

Adverse training condition demonstrates the scenario where the recording condition is extremely adverse, hence, most of training data are corrupted. Figure 6(c) shows the accuracies of the comparative algorithms evaluated on the clean data sets. All systems are trained in 0 dB condition but with varying number of training samples. Compared to the clean training condition [Fig. 6(a)], the performance of all algorithms drops significantly. The proposed algorithm has the least degradation and outperforms DTW and HMMs in all $N$'s (numbers of training samples). Under the same $N$ and algorithm, the performance trend exhibits an interesting behavior especially in adverse-condition training (and multiconditional training). In Figs. 7(c) and 7(d), all algorithms

seem to reach their best performance at a moderate SNR (5–15 dB) rather than the extreme ones (clean or 0 dB). This is because most of training samples, if not all, are highly corrupted. The model will not predict a clean sample very well due to the mismatch. However, when the test data are at 0 dB SNR the signal quality is still too poor to get an accurate prediction even though the model is trained at 0 dB SNR. With a moderate SNR (5–15 dB) test sample, the model characteristic is not far from the test sample, and the signal quality is not severely corrupted, yielding the best performance of all SNR levels.

## VI. SUMMARY AND CONCLUSION

A robust template-based classification framework has been proposed. The algorithm introduces "prominent region," which is an essential component for noise-robust classification of birdsongs. In addition, the proposed algorithm is designed to be robust when the number of training data is limited (four samples or less). The representation of signals is the simple time-frequency spectrogram. During the training process, the algorithm extracts reliable information from training samples in an iterative fashion called SFA. At the end of the process, a template is derived for each phrase class. Each template has three attributes—reference spectrogram, the prominent region, and frame weighting function. During classification, a given test spectrogram is matched with each template in a dynamic programming fashion. The attributes of each template make this process much more robust than the traditional DTW.

The phrases used in this study are extracted from songs of the CAVI. The training set contains a wide range of training samples (from 1 to 32) per phrase class from a few bird individuals. The data were generated in three training conditions—clean, multiconditional, and adverse-conditional training, resulting in 18 training conditions (6 numbers of training samples each). The models or systems derived from each training condition were then tested at 6 SNR levels, resulting in 108 train-test conditions.

Experimental results show that the proposed algorithm outperforms DTW and HMMs in most conditions. Among 108 train-test conditions, the best algorithm in each condition is as follows: the proposed algorithm in 3 conditions, HMMs in 103 conditions, DTW in 1 condition, and 1 tie case of HMMs and the proposed algorithm. The 4 scenarios where HMMs outperform the proposed algorithm is in clean and multiconditional training with 32 samples that tested with somewhat clean data (15 dB or above). The only scenario where DTW outperforms the proposed algorithm is when the system is trained in clean with one sample and tested in clean data set. However, in the five cases where the proposed algorithm underperforms either HMMs or DTW, the margin is small ($<1\%$). For the rest of the train-test conditions, the proposed algorithm usually outperforms by large margins especially when the test condition has extremely low quality (low SNRs). The behaviors of DTW and HMMs under each training and test conditions are also analyzed. In limited-data training conditions (four samples or less), DTW outperforms HMMs in all cases except in multiconditional training that tested with high-SNR data sets (15 dB and above). When the number of training data is 32, HMMs outperform DTW in all cases. This observation confirms that the HMM framework prefers a large amount of training data while DTW works reasonably well in limited training data but fails to improve when more data are available. Both algorithms suffer in mismatched conditions due to background noise that may be present in training or test data. The proposed algorithm is robust to limited training data and noise.

Future work will include development of a better system that takes advantage when the data are more available for some phrase classes but are still limited for others. In addition, we will develop a noise-robust HMM framework by integrating the concepts of prominent regions and spectrogram fusion in HMM training and decoding. We also plan to extend this framework to fully automated phrase recognition where presegmentation is not needed for classification. The algorithm can be extended to species classification where techniques such as *k*-means and other clustering algorithms can be used.

## ACKNOWLEDGMENTS

This research was supported in part by National Science Foundation Award Number 1125423.

[1]A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, R. Planqué, A. Rauber, S. Palazzo, B. Fisher, and H. Müller, "LifeCLEF 2015: Multimedia life species identification challenges," in *Experimental IR Meets Multilinguality, Mutlimodality, and Interaction*, Vol. 9283 of Lecture Notes in Computer Sciences (Springer, 2015), pp. 462–483, available at http://link.springer.com/chapter/10.1007%2F978-3-319-24027-5_46.

[2]P. J. Clemins, M. T. Johnson, K. M. Leong, and A. Savage, "Automatic classification and speaker identification of African elephant (*Loxodonta africana*) vocalizations," J. Acoust. Soc. Am. **117**(2), 956–963 (2005).

[3]D. T. Blumstein, D. J. Mennill, P. Clemins, L. Girod, K. Yao, G. Patricelli, J. L. Deppe, A. H. Krakauer, C. Clark, K. A. Cortopassi, S. F. Hanser, B. McCowan, A. M. Ali, and A. N. G. Kirschel, "Acoustic monitoring in terrestrial environments using microphone arrays: Applications, technological considerations and prospectus," J. Appl. Ecol. **48**(3), 758–767 (2011).

[4]A. Kershenbaum, D. T. Blumstein, M. A. Roch, Ç. Akçay, G. Backus, M. A. Bee, K. Bohn, Y. Cao, G. Carter, C. Cäsar, M. Coen, S. L. DeRuiter, L. Doyle, S. Edelman, R. Ferrer-i-Cancho, T. M. Freeberg, E. C. Garland, M. Gustison, H. E. Harley, C. Huetz, M. Hughes, J. Hylan Bruno, A. Ilany, D. Z. Jin, M. Johnson, C. Ju, J. Karnowski, B. Lohr, M. B. Manser, B. McCowan, E. Mercado, P. M. Narins, A. Piel, M. Rice, R. Salmi, K. Sasahara, L. Sayigh, Y. Shiu, C. Taylor, E. E. Vallejo, S. Waller, and V. Zamora-Gutierrez, "Acoustic sequences in non-human animals: A tutorial review and prospectus," Biol. Rev. **91**(1), 13–52 (2016).

[5]C. K. Catchpole and P. J. B. Slater, *Bird Song: Biological Themes and Variations* (Cambridge University Press, Cambridge, 1995).

[6]R. W. Hedley, "Composition and sequential organization of song repertoires in Cassin's Vireo (*Vireo cassinii*)," J. Ornithol. **157**(1), 13–22 (2015).

[7]S. E. Anderson, A. S. Dave, and D. Margoliash, "Template-based automatic recognition of birdsong syllables from continuous recordings," J. Acoust. Soc. Am. **100**(2), 1209–1219 (1996).

[8]S. Fagerlund, "Bird species recognition using support vector machines," EURASIP J. Adv. Signal. Proc. **2007**(1), 64 (2007).

[9]M. A. Acevedo, C. J. Corrada-Bravo, H. Corrada-Bravo, L. J. Villanueva-Rivera, and T. M. Aide, "Automated classification of bird and amphibian calls using machine learning: A comparison of methods," Ecol. Inf. **4**(4), 206–214 (2009).

[10]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," in *INTERSPEECH* (2012), pp. 2522–2525.

[11]L. N. Tan, A. Alwan, G. Kossan, M. L. Cody, and C. E. Taylor, "Dynamic time warping and sparse representation classification for birdsong phrase classification using limited training data," J. Acoust. Soc. Am. **137**(3), 1069–1080 (2015).
[12]V. M. Trifa, A. N. Kirschel, C. E. Taylor, and E. E. Vallejo, "Automated species recognition of antbirds in a Mexican rainforest using hidden Markov models," J. Acoust. Soc. Am. **123**(4), 2424–2431 (2008).
[13]J. A. Kogan and D. Margoliash, "Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden Markov models: A comparative study," J. Acoust. Soc. Am. **103**(4), 2185–2196 (1998).
[14]K. Ito, K. Mori, and S.-i. Iwasaki, "Application of dynamic programming matching to classification of budgerigar contact calls," J. Acoust. Soc. Am. **100**(6), 3947–3956 (1996).
[15]O. Dufour, T. Artieres, H. Glotin, and P. Giraudet, "Clusterized Mel filter cepstral coefficients and support vector machines for bird song identification," in *Proc. of 1st Workshop on Machine Learning for Bioacoustics*, Vol. 951 (2013), pp. 89–93.
[16]F. Briggs, X. Fern, and R. Raich, "Acoustic classification of bird species from syllables: An empirical study," Oregon State University Technical Report, 174 182-183 (2009).
[17]W. Chu and D. T. Blumstein, "Noise robust bird song detection using syllable pattern-based hidden Markov models," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2011), pp. 345–348.
[18]S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," IEEE Trans. Acoust., Speech, Signal Process. **27**(2), 113–120 (1979).
[19]J. Van Hout and A. Alwan, "A novel approach to soft-mask estimation and log-spectral enhancement for robust speech recognition," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2012), pp. 4105–4108.
[20]B. Raj and R. M. Stern, "Missing-feature approaches in speech recognition," IEEE Signal Process. Mag. **22**(5), 101–116 (2005).
[21]F. Briggs, B. Lakshminarayanan, L. Neal, X. Z. Fern, R. Raich, S. J. Hadley, A. S. Hadley, and M. G. Betts, "Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach," J. Acoust. Soc. Am. **131**(6), 4640–4650 (2012).
[22]M. Cooke, "A glimpsing model of speech perception in noise," J. Acoust. Soc. Am. **119**(3), 1562–1573 (2006).
[23]C. B. Goguen and D. R. Curson, "Cassin's Vireo (*Vireo cassinii*), The birds of North America online)," edited by A. Poole (Cornell Lab of Ornithology, Ithaca, NY), available at http://bna.birds.cornell.edu/bna/species/615 (Last viewed 7/18/2012).
[24]P. Boersma and D. Weenink, "Praat: Doing phonetics by computer (version 5.2.22) [computer program]," (2010), http://www.praat.org (Last viewed 4/15/2011).
[25]K. Kaewtip, L. N. Tan, A. Alwan, and C. E. Taylor, "A robust automatic bird phrase classifier using dynamic time-warping with prominent region identification," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP*) (2013), pp. 768–772.
[26]H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," IEEE Trans. Acoust., Speech, Signal Process. **26**(1), 43–49 (1978).
[27]S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, and D. Povey, *The HTK Book* (Entropic Cambridge Research Laboratory, Cambridge, 1997), Vol. 2.
[28]H.-G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ASR2000—Automatic Speech Recognition: Challenges for the New Millenium ISCA Tutorial and Research Workshop (ITRW*) (2000).
[29]J. Arriaga, M. L. Cody, E. E. Vallejo, and C. E. Taylor, "Bird-db database for annotated bird song sequences," http://taylor0.biology.ucla.edu/birdDBQuery/ (Last viewed 10/26/2016).