AN ABSTRACT OF THE THESIS OF

Lawrence W. Neal for the degree of Honors Baccalaureate of Science in Computer Science presented on May 30, 2012.

Title: Detection and Segmentation of Bird Song in Noisy Environments

Abstract approved: _____

Xiaoli Z. Fern

Recent work in machine learning concerns the detection and identification of bird species from audio recordings of their vocalizations. Such analysis can yield valuable ecological information concerning the activity and distribution of species in the wild. Current species-identification methods require individual syllables of bird audio as input, but field-collected audio contains noise and simultaneous vocalizations. This thesis presents two supervised learning methods for identifying and separating individual syllables of bird vocalizations from field-recorded audio. The segments output by this process can be input into species classification algorithms, to yield useful ecological data.

Detection and Segmentation of Bird Song in Noisy Environments

by

Lawrence W. Neal

A PROJECT

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science

Presented May 30, 2012
Commencement June 2012

Honors Baccalaureate of Science in Computer Science thesis of Lawrence W. Neal presented on May 30, 2012

APPROVED:

Xiaoli Z. Fern, School of Electrical Engineering and Computer Science

---

Raviv Raich, School of Electrical Engineering and Computer Science

---

Julia Jones, Department of Geosciences

Daniel J. Arp, University Honors College

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Lawrence W. Neal, Author

ACKNOWLEDGEMENTS

I am indebted to Xiaoli Fern, Raviv Raich, Forrest Briggs, Julia Jones, Jed Irvine, Matthew Betts, Sarah Frey, Adam Hadley, and all the others who have made possible the research undertaken by the OSU Bioacoustics group.

I would like to thank my family for their constant support, in this thesis and all of my efforts here at Oregon State.

This work was partly supported by NSF award 1055113, and grants from the National Science Foundation to the HJ Andrews Long-Term Ecological Research (LTER) program and the Ecosystem Informatics IGERT at Oregon State University.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

DETECTION AND SEGMENTATION OF BIRD SONG IN NOISY ENVIRONMENTS

1. INTRODUCTION AND BACKGROUND

Recent advancements in machine learning and related technologies have made possible automated data collection and analysis at greater scope and resolution than ever before. Machine analysis of text, image, video, and audio now yield useful data in many applications for science, industry, and commerce. Among recent applications of machine learning for data collection is the analysis of bioacoustic audio for ecological studies. Such bioacoustic data collection systems apply learning algorithms to recorded audio to identify species and individuals of animal populations under study, to generate useful data about species distribution and activity.

This thesis describes my work with the Oregon State Bioacoustics Group, an interdisciplinary collaboration with the goal of developing a system that will analyze large sets of automatically-recorded audio data to better understand the behavior of bird populations. This process involves several sequential steps: the collection of digital audio, the extraction of bird sound from that audio, the identification of the species of each recognized bird sound, and the mapping of species counts back to the location of recorders in the forest. The focus of this thesis is the implementation of the second step in this process: the detection and extraction of bird sounds from field-collected audio. Sections 3 and 4 will describe two methods of segmenting distinct syllables of bird sound from field recordings using supervised machine learning. Each of the methods involves decomposing input audio to a time-frequency spectrogram, then applying supervised techniques

to segment pixels of the spectrogram in a way analagous to image segmentation. The segments output by these methods can be input to a species classification algorithm, such as those described in [8] and [7], to generate species count data useful for ecological research.

1.1. Organization of this thesis

The remainder of Section 1 will explain the motivation and ecological use of an automated recording system, and give background information on the data collection performed at the H. J. Andrews Experimental Forest. Section 2 will explain the formulation of the audio segmentation problem, the definitions and terminology used in the thesis, and the relationship of segmentation to the rest of the species identification system. Section 3 will describe the ”Per-Pixel Time-Frequency” segmentation method first developed to solve the segmentation task. Section 4 will desctibe the later-developed ”Superpixel Merger” method. Section 5 will introduce evaluation metrics, and will evaluate both proposed segmentation methods alongside a naive non-learning alternative.

1.2. Motivation and Ecological Importance

Birds are among the most visible and easily identifiable of vertebrate species. As such, bird population and activity are widely used in ecological studies as markers for overall environmental capacity[1]. Traditional methods of surveying bird activity, however, are labor-intensive and low-resolution. Typical studies of bird species activity involve trained human observers watching and listening for birds at dozens of pre-determined points in the environment, and taking counts of each species detected. This species count data can be analyzed to reveal meaningful ecological patterns with respect to time, loca-

tion, moisture gradient, tree density, and other environmental characteristics[1][2]. Despite the ecological importance of such data, several issues limit the scope and resolution of species counts gathered with traditional methods.

The most pressing limit on manual data collection for bird activity is the requirement that an observer physically move to each site or location. This limits the number of locations and the amount of time that can be spent at each one. Because each observer cannot be present in multiple sites at once, some method of sampling must be performed in which each site is observed at different times. Because the observers must physically move from site to site, there may be biases in the collected data introduced by the time of sampling. If the bird activity being measured changes over the course of several hours, then a site observed early in the morning may not yield the same data when sampled later in the day. Because of the travel time involved, sampling a single site multiple times in one day may be prohibitively expensive. The requirement for physical presence forces a tradeoff for data collection: more sites can be sampled each day by shortening the duration of observation at each site, or fewer sites can be sampled to gain a more accurate count at each site.

A second factor limiting accurate data collection is the difference between observers. Because each observer may have a different skill level or a different visibility or volume threshold for counting each bird call, there may be bias introduced. To reduce this bias, observers can be randomly assigned to different locations, and common training can be undertaken by all observers[1]. Random assignment reduces bias, but as observers are individuals, logistical constraints can prevent a study from randomly assigning all data-gathering personnel. Likewise, observers who gathered data in previous years or in physically distant locations cannot always be brought in to repeat their observation. While common training procedures can reduce the difference in data collected by different individuals, training cannot remove the differences entirely. The perfor-

mance of individuals in detecting and identifying birds varies not only with training, but also age, experience, motivation, hearing acuity, eyesight, physical health, and fatigue level[3]. Even with training, subtle differences between species can be recognized differently among data collectors, and one data collector may count a distant or less-noticable bird that another does not.

Other issues that arise in the manual collection of bird counts include observers' effects on the behavior of birds, changing environmental variables such as wind velocity, precipitation, cloud cover, and light intensity, and differences in the detectability of each bird species[3]. A number of statistically-motivated techniques have been applied to reduce the effect of these biases. These approaches include double sampling, distance sampling, removal model estimation, and double-observer point counts[5]. These techniques can increase the consistency and reliability of gathered data, but at the cost of sample size. Techniques based on double-observer point counts, for example, can significantly increase reliability by providing an estimate of the rate of false negative counts (birds that were noticed by one observer but not another). The double-observer technique still assumes equivalent performance by each observer, however, and reduces by a factor of two the spatial or temporal scope of data collection.

Many of these issues that limit the scope of bird count data collection can be ameliorated by applying uniform, machine-automated bird species recognition to each site in an area of study. This is the goal of the ongoing bioacoustics research based on data from the H. J. Andrews Experimental Forest. By applying a single predetermined computer analysis to all data collected, differences between observers are negated. By collecting audio simultaneously at all sites, biases in data collection related to time of day or travel time to sites are also negated. Because automatic audio recorders are limited only by battery life and memory capacity, they can record for much longer than humans could, yielding a larger data set.

Although automated recorders provide an alternative to human data collection, an automated system introduces limitations of its own. Audio-only systems will never account for birds that do not produce sound, and will capture loud bird song at a larger radius than soft. By combining automated recording with human-collected data, greater accuracy and resolution can be achieved.

1.3. Data Collection and Experimental Site

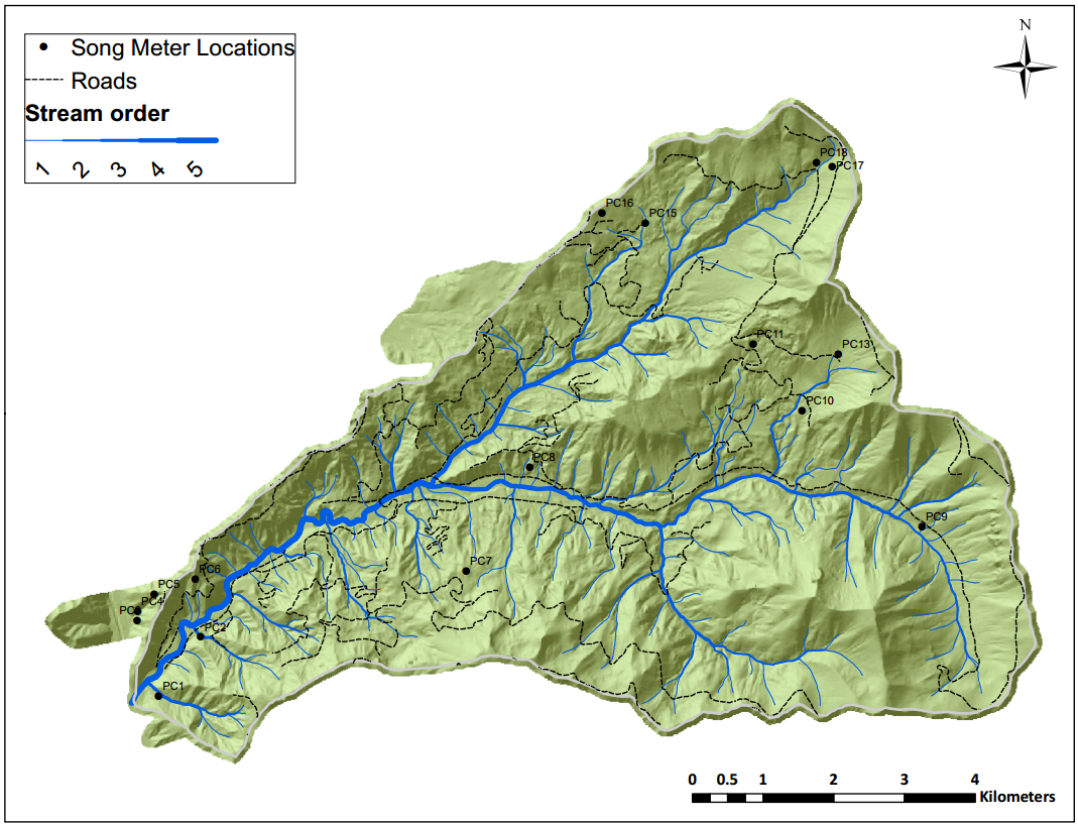


FIGURE 1.1: A map of the H. J. Andrews Experimental forest, with data collection sites labeled

The audio data collected from H.J. Andrews is gathered by thirteen Wildlife Acoustics Song Meter SM2 recording devices, each placed in a separate location in the forest. The devices are referenced by site number, from site PC1 to site PC18 (some sites were

discontinued), placed so as to cover a range of elevations and habitats. Each device consists of a battery power source, stereo microphones and associated electronics, with a removable flash memory card. Devices are placed and set to a daily schedule, then left for approximately one week between battery and memory card replacements. The majority of recorded audio is in uncompressed 16kHz stereo PCM WAV format.

Audio has been recorded at each site on regular schedules between May and August of each year, beginning in 2009. Only 13 of the 16 recorders operated during the 2009 season, and not all recorders operated each week. For the first half of the 2009 season, audio was recorded 20 minutes per hour, 24 hours per day, at each site. Later adjustments left the recorders collecting data only during hours of peak bird activity, approximately 5am-12pm. Although each year's collected data is incomplete, the gathered data permits year-to-year comparisons for several weeks in each site.

For a visual guide to the songs of species commonly recorded in the H.J. Andrews dataset, refer to Appendix A. This document contains several examples of spectrogram output containing syllables of song from each of the 13 species most commonly recorded in the dataset. These species make up the vast majority of bird syllables detected in the collected audio.

FIGURE 1.2: Left: A Song Meter recording device in H.J. Andrews. Right: An alternative microphone configuration.

2. AUDIO SEGMENTATION

2.1. Overview of Audio Segmentation

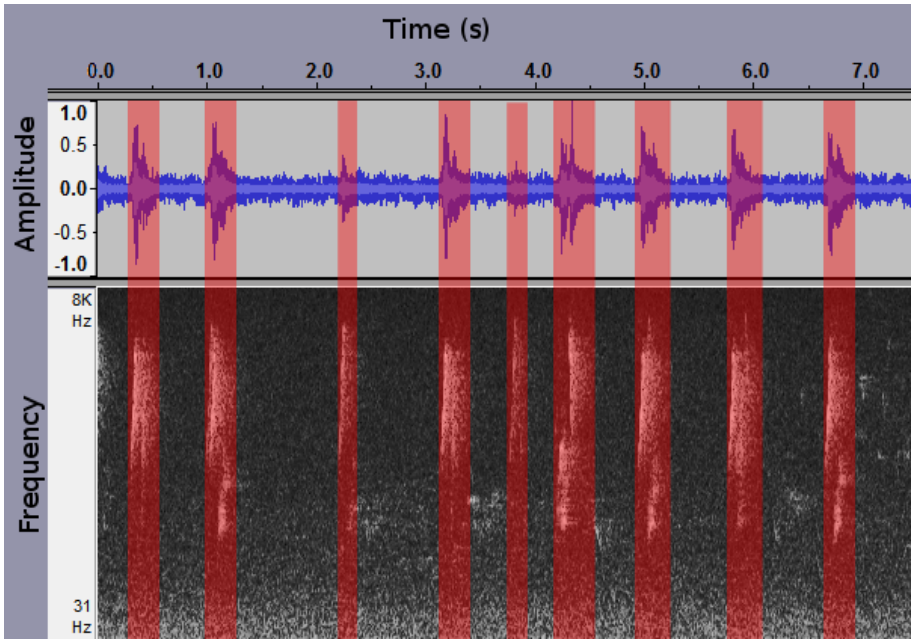


FIGURE 2.1: An example time-domain segmentation of audio collected from H.J. Andrews. Top: Audio waveform. Bottom: Spectrogram

The term ”segmentation” is applied in the context of audio processing to mean the separation of one audio recording into multiple recordings. Most applications of automatic audio segmentation involve dividing a length of audio by splitting it at instants in time. For example, work has been carried out in segmenting recorded audio of radio or television broadcasts, by detecting instants in time that correspond to transitions between scenes or commercials This type of segmentation, which will be referred to as ”time-domain” segmentation, works well in domains such as television audio and interactive voice response systems. Under certain criteria, time-domain segmentation can work well in separating bioacoustic sounds, such as bird calls, from silence or non-

interesting background noise. In problems where a small number of ”interesting” sound events are spread out sparsely, where few or no sounds overlap in time, and where background noise is constant, identifying beginning and ending timestamps is sufficient to isolate each event.

Many audio processing problems must deal with signals that do not meet these criteria. In processing field-collected audio from the recorders in H.J. Andrews, recordings must be segmented that contain multiple simultaneous vocalizing birds, at varying amplitudes and against varying background noise. In order to separate individual sounds against varying background signals, a different type of segmentation is required, one that separates sounds that occur simultaneously.

2.2. Time-Frequency Segmentation

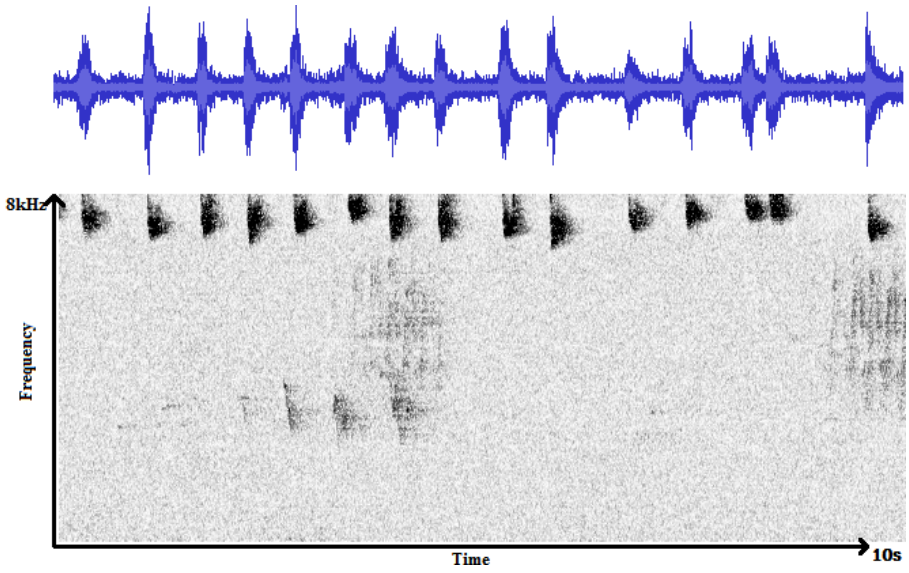


FIGURE 2.2: An example waveform and corresponding spectrogram of a ten-second audio clip containing syllables from three species. Darkened areas indicate higher audio energy.

The data collected from recorders in the HJ Andrews forest is not amenable to

simple time-domain segmentation for two main reasons. First, the background sound that a syllable is heard against can vary with time of day, weather, and other conditions such as the flow rate of nearby running water. This means that two time-domain audio segments of the same syllable may look very different to an automated identification system, due to interference from different background noises. Thus, the output of a time-domain segmentation method may not allow accurate species classification. The second, and crucial problem is that multiple birds will often vocalize simultaneously, especially during peak morning hours. If two birds call in two different frequency ranges, but overlapping in time, there is no meaningful way for time-domain segmentation to extract two segments that each represent the vocalization of one bird.

Because of the shortcomings of purely time-domain segmentation, we take advantage of the structure exhibited by bird calls in the time-frequency plane. By decomposing a one-dimensional function of time, such as an audio signal, using a discrete Fourier transform, we can generate a two-dimensional spectrogram. We represent spectrograms as bitmap images in which the horizontal axis corresponds to time, while the vertical axis corresponds to frequency. We visualize spectrograms in monochrome or color relief, where the brightness or color of each pixel in the spectrogram represents the amplitude of audio energy at a given interval of time and within a given interval of frequency.

2.2.1 Computational Auditory Scene Analysis

The ability of a human listener to concentrate selectively on one sound in a mixture of many, a phenomenon known as the ”cocktail party problem”, has long been studied by researchers interested in the analysis of audio. The field of ”auditory scene analysis” studies the psychoacoustic process by which human listeners separate an audio signal into the multiple sound sources that comprise it [13]. Advancements in the understanding of the process have allowed work in emulating human auditory scene analysis through computer models, a field of work known as ”computational auditory scene anal-

ysis" [4] Sound separation methods based on CASA generally involve decomposition of audio signals into the time-frequency domain, followed by segmentation of the audio into regions represented by binary masks over the spectrogram [12]. The segmentation methods proposed here work on the same principles; it is assumed that each discrete sound (in this case, a syllable of bird call) can be represented by a binary mask over its spectrogram.

2.2.2 Parallels with Image Segmentation

The segmentation of a spectrogram into contiguous regions that each represent a meaningful sound is a similar problem to the segmentation of an image into regions that represent objects. In the most common formulation of image segmentation, each pixel is assigned to a segment, and each segment is is contiguous group of pixels that overlay a part of the same object or texture. Spectrogram segmentation can be approached in the same way, except that each pixel represents a time-frequency range rather than a position. Image segmentation is often applied based on computed features describing the texture of local regions of pixels. Analogously, a time-frequency texture can be computed from a patch of pixels within a spectrogram.

Despite the clear similarities, there are a few important differences between spectrogram segmentation and image segmentation. While the horizontal and vertical spatial dimensions are usually treated as equivalent in an image, the two are very different in a spectrogram. The dimensions of a spectrogram are not fixed; different parameters for the Fourier transform that generates the spectrogram can arbitrarily stretch the resulting bitmap in the horizontal or vertical directions. Because vertical distance and horizontal distance have distinct meanings in the time-frequency plane, many image segmentation techniques, such as the use of scale-invariant and rotation-invariant [14] [15] features to describe image texture, are not applicable. Segmentation of spectrograms must also deal with transparency, a phenomenon not commonly accounted for in image segmentation.

Two sounds that overlap in both the time and frequency ranges will create an additive mixture, rather than one occluding the other.

2.3. Problem Formulation

The purpose of the two computational methods outlined in this thesis is to identify contiguous time-frequency regions (segments) within input spectrograms, that correspond to individual bird vocalizations. We define a ”syllable” to be a distinct unit of sound produced by a bird. Bird song exhibits organizational structure at multiple levels; each syllable may consist of multiple repetitions of a single pattern, and likewise, multiple syllables may combine in pre-determined patterns to form songs [6]. In our analysis, no distinction is made between songs and calls, or other bioacoustic bird sounds. Instead, the objective of the methods is to match each syllable to one segment, where a segment is defined as a contiguous and identifiable region within the time-frequency plane.

By defining each segment as contiguous, we mean that each syllable must form a connected component in the spectrogram. A bird song consisting of alternating low notes followed by high notes could be segmented as multiple syllables, one per note. Alternatively, it could be segmented as a single component containing the low and high notes, and the connecting spectrogram pixels in-between.

In order for syllable segments to be useful for a species classifier, each syllable must also be identifiable, which is to say it must share characteristics with other repetitions of the same vocalization. This does not necessarily mean that each vocalization correponds to one syllable. Each bird song may consist of multiple syllables, and any single level of segmentation may break a single syllable into multiple segments. Because no single level of segmentation can capture the full structure of sounds in a spectrogram, a system of hierarchical organization would be necessary to fully describe a complex auditory environ-

ment [4]. In order to simplify the problem of species classification, however, the methods described here assume a one-to-one correspondance between syllables and contiguous spectrogram segments. This formulation of the problem provides limited ability to count individual vocalizing birds. However, it is sufficient for determining the presence or absence of species within the intervals of time sampled for spectrograms (typically 10 to 15 seconds in our work).

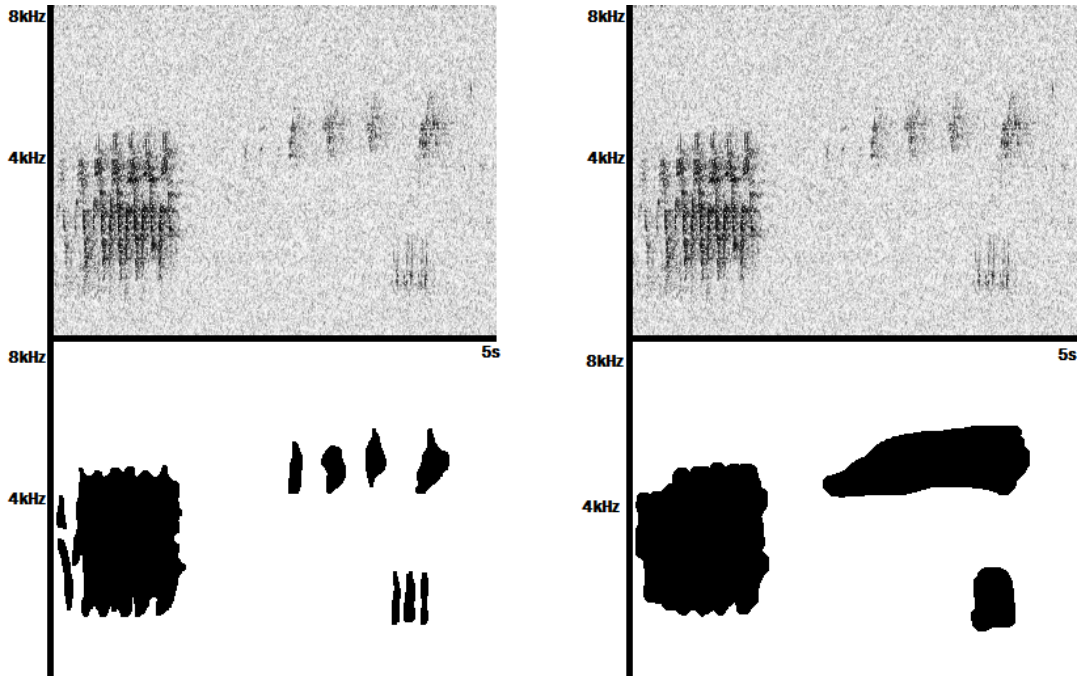


FIGURE 2.3: Two possible binary masks forming segmentations over a spectrogram.

2.4. Segmentation Output

Accurate segmentation of bird syllables is essential for successful automatic species classification[10]. Because the objective of the system is to classify the species of birds, it is essential that each audible bird call be represented by at least one segment, and that calls by multiple birds not be merged into single segments. Each of the two segmenta-

tion methods proposed here outputs contiguous two-dimensional monochrome image regions, in which the value of each pixel is the noise-filtered spectrogram coefficient. Each output spectrogram segment is clipped horizontally, so information about the segment's position in time is not saved. However, the segment's vertical spectrogram position, corresponding to frequency, is not clipped, as the absolute frequency of a syllable is an important input to features that may describe it, and is useful in species classification.

Classification of species using the output segments involves computing some descriptive feature vector for each segment, using features that will capture the distinct shapes and textures of bird species' syllables. These segment feature vectors can be used as individual inputs to a species classification scheme, in which a classifier is trained on feature vector/species label pairs. Alternatively, as in [7], multiple segments can be aggregated over each time period, and using Multi-Instance Multi-Label learning, a classifier can be trained to output presence/absence for each time period. The benefit of a MIML approach is the relaxation of requirements for training data- rather than a requirement that each syllable be labeled by species, the MIML classifier can be trained using binary vectors of species presence/absence per time period.

3. PER-PIXEL TIME-FREQUENCY METHOD

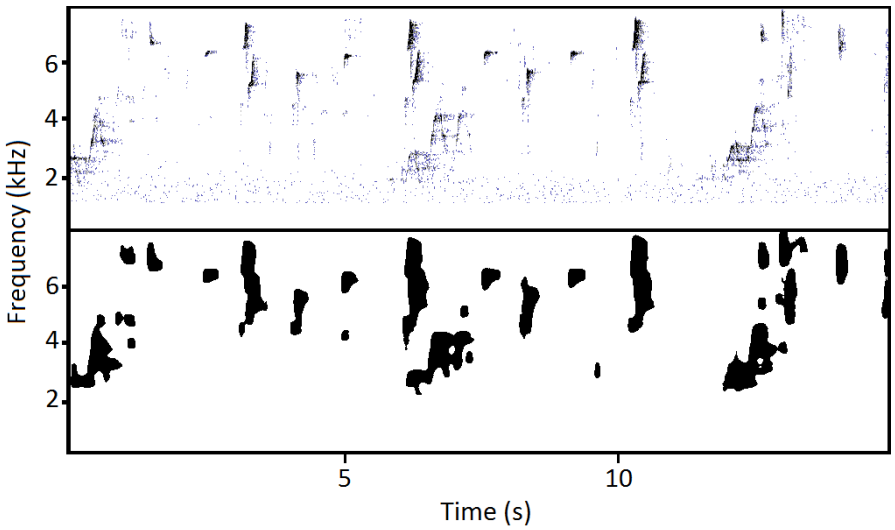


FIGURE 3.1: Above: A noise-reduced spectrogram of a Swainson's Thrush and a Pacific-Slope Flycatcher. Below: The binary mask generated by the proposed method. Each darkened region corresponds to a detected syllable of bird song.

3.1. Motivation and Approach

The Per-Pixel Time-Frequency segmentation method is based on the idea that each pixel in a spectrogram can be classified as either part of a bird syllable or part of the background, based on the amplitude of spectrogram pixels in a local window around it. The method classifies each pixel in an input spectrogram as positive (bird syllable) or negative (background), and then extracts contiguous regions of positive-labeled pixels as segments. This allows the extraction of syllables with arbitrary shape and size in the spectrogram, and allows the separation of syllables that overlap in either time or frequency (but not both).

First, a spectrogram is generated for each input audio clip. Each spectrogram is

noise-reduced and contrast-boosted. The method involves training a Random Forest decision tree ensemble classifier to label individual pixels of a spectrogram, using feature vectors that represent a patch of the spectrogram, centered on each pixel. The classifier requires as training input a set of audio clips with associated human-generated training labels. The training labels are two-dimensional monochrome bitmaps that form a mask over the spectrogram of each audio clip. Zero-valued pixels in the training labels indicate background or non-bird noise regions in the spectrogram, while nonzero pixels indicate bird call. Pixels are randomly sampled from input training data, and for each pixel, a feature vector is constructed based on the values of pixels in a window centered on it. The training label assigned to the pixel is positive or negative based on the value of the corresponding pixel in the training mask.

After training, the random forest classifier is applied to each pixel in the spectrograms of input audio. The output of the classifier at this step is a mask over the spectrogram, where the value of each pixel is the output of the random forest algorithm for that pixel: a real value in the range 0 to 1 representing the proportion of trees in the forest that 'voted' for a positive label. To compensate for noise from outlier examples, a Gaussian blur is applied to this per-pixel probability mask. To translate from probability mask to a set of spectrogram segments, a threshold θ is applied. Each contiguous region of pixels with labels greater than θ is extracted as a syllable.

3.2. The Random Forest Algorithm

The per-pixel segmentation method is based on supervised classification using a Random Forest classifier. Random Forest (RF) is an ensemble classifier consisting of a collection of decision trees [9]. Given a set of training examples $\mathcal{T}$, each tree h_i in the RF classifier is independently built from a bootstrap sample selected randomly with replace-

ment from $\mathcal{T}$. Trees are constructed by recursively applying the following procedure:

- Take as input a set of examples T, where each $T_i = (x, y)$, x is a feature vector, and y is the corresponding class label.
- If all labels y are the same, create a leaf node with the value y.
- Select a random subset $\mathcal{F}$ of $log_2(k) + 1$ features, where k is the number of features in x.
- For each feature $d \in \mathcal{F}$, sort T on d and find the threshold value θ_d that splits T into two sets T_{left} and T_{right}, such that the Gini index $G(T_{left}, T_{right})$ is maximized.
- Choose the feature and threshold (d, θ_d) such that G is maximized. If all possible values of G are equal, then make a leaf node with the majority label. Otherwise, create two child nodes by recursively applying the procedure using T_{left} and T_{right} as input.

Each interior node of an RF tree corresponds to a test of the form $x_d < \theta$. Traversing the tree with any input vector x will lead to a leaf node, which contains a single class label y. When classifying an input x, each decision tree in the RF classifier casts a vote. The output label for x is equal to the proportion of trees that voted for y.

3.3. Segmentation Process

3.3.1 Preprocessing

In each input audio file, a Hamming window is first applied to each frame. A short-time FFT is then applied with a frame size of 512 samples, and an overlap of 256 samples between each subsequent frame, transforming the signal into a time-frequency spectrogram. A whitening filter is subsequently applied to the spectrogram, to normalize

the level of environmental noise at each frequency. Frequency ranges below 1kHz contain little or no bird call [11], so a band-pass filter removes frequencies under 1kHz.

3.3.2 Random Forest Training

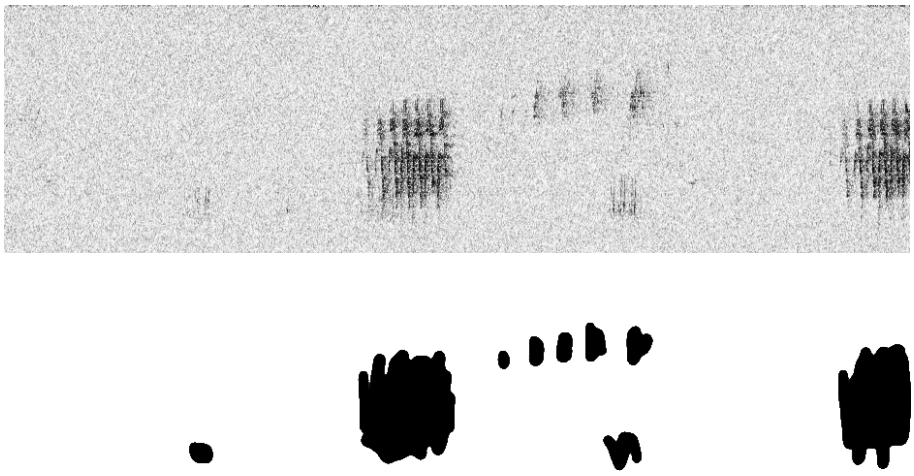


FIGURE 3.2: Above: Spectrogram of an example training input. Below: The human-generated binary mask used for training.

The method requires as input a training set of audio files with corresponding binary masks (see 3.2). Each audio file is converted to a spectrogram, using the same parameters as the input data. Time-frequency units covered by the mask are used as positive examples of bird sound when training the classifier. All other elements, including silence, static noise, and non-bird sound, are used as negative examples. The masks in this set were created manually by visual and auditory examination of each spectrogram and its corresponding audio. For the purposes of training and evaluation, these masks are assumed to be ideal binary masks corresponding to audible bird call.

3.3.3 Classification

In training and classification, a feature vector $x_{t,f}$ is extracted for each time-frequency unit $S(t, f)$ in the spectrogram. The vector $x_{t,f}$ describes the spectral characteristics of a rectangular window surrounding (t, f), and is defined by the following:

- The frequency value f
- The values within a rectangular window surrounding (t, f)

  $$S(i,j), i \in [t - t_w, t + t_w],\ j \in [f - f_w, f + f_w]$$

  centered at (t, f), where $2t_w + 1$ is the size of the window in the time dimension and $2f_w + 1$ is its size in the frequency dimension.
- The variance σ^2 of the units in this window

  $$\sigma^2 = \frac{1}{(2t_w+1)(2f_w+1)} \sum_{i=t-t_w}^{t+t_w} \sum_{j=f-f_w}^{f+f_w} (S_{i,j} - \mu)^2$$

  where μ is the mean value in this window.

We use a t_w value of 6 T-F units and a f_w value of 12 units, yielding a window spanning 192ms by 750hz in the T-F domain. In the classification process, a probability mask M_p is generated by the outputs of the RF classifier, in which each value $M_p(i, j)$ corresponds to the fraction of RF trees that labeled $S_{i,j}$ as bird call.

3.4. Output and Analysis

After classification, a Gaussian convolution is applied to create a smoothed probability mask M_s.

$$M_s = M_p \star g, \text{ where } g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

This convolution is applied with a square kernel of 17x17 time-frequency units, and $\sigma = 3.0$. After smoothing, the probability mask is converted to a binary mask M_b by applying a threshold

$$M_b(x,y) = 1 \text{ if } M_s \geq \theta, \text{ or } 0 \text{ otherwise}$$

where $0 \leq \theta \leq 1$. The value θ controls the trade-off between false positive and false negative. A larger value leads to lower false positive rate but higher false negative rate. The

smallest time-frequency regions identifiable as bird syllables typically have a duration of approximately 160ms and a frequency range of approximately 300hz. Any regions in the binary mask less than 90% of this size are discarded from the final segmentation.

FIGURE 3.3: Example spectrogram segments output by the per-pixel method, grouped by species.

4. SUPERPIXEL MERGER METHOD

4.1. Motivation

The ”Superpixel Merger” method is a heuristic-driven method of segmenting bird syllables that borrows concepts from computer vision. Although it requires more parameter tuning than the per-pixel random forest, its greater complexity yields better accuracy for a given amount of computation, especially if segmentation run-time must be restricted to real-time or faster. The superpixel method has another advantage in that it contains a mechanism for separating syllables that overlap in the time-frequency plane, a situation that always results in under-segmentation for the per-pixel method. Although the unintended merging of syllables is still a common problem, further development of the superpixel method could conceivably improve performance significantly from the per-pixel method.

The Superpixel Merger method involves an initial pre-segmentation of all pixels into small, homogeneously-sized regions, formed such that boundaries of the regions separate segments of bird call. These regions, analagous to ”superpixels” in image segmentation, are subsequently clustered together to build a segment over each detected syllable. After the initial pre-segmentation, two supervised Random Forest classifiers are used: one to discriminate foreground (bird call) from background, and another to separate adjacent sets of superpixels into individual segments.

Classifying individual pixels requires a feature vector to be constructed for each pixel. Due to the large number of pixels, the size of this vector and the computation required to generate it must be strictly limited in order to achieve an acceptable training and classification speed. The motivation behind the superpixel-based method is to reduce the number of elements to be learned or classified, and thus increase classification

speed. Instead of classifying hundred of thousands of individual pixels, the superpixel merger method splits each input spectrogram into a few hundred regions, then classifies those regions. This allows more computation per feature vector, and thus a more detailed characterization of each region of the spectrogram.

4.2. Superpixel Segmentation Process

4.2.1 Preprocessing

The segmentation process begins with the same spectrogram-generating process undertaken in the per-pixel method. Each input audio file is transformed into a spectrogram using the FFT algorithm with Hamming-windowed frames of 512 samples each, and an overlap of 256 frames. Resulting spectrograms are contrast-boosted for visibility by taking the square root of amplitude values. A whitening noise filter is applied to reduce the effect of static background noise.

4.2.2 SLIC Algorithm

We use a method based on the Simple Linear Iterative Clustering (SLIC) algorithm to pre-segment each input spectrogram into a set of evenly-sized regions, shaped around potentially interesting features in the spectrogram. SLIC is an image segmentation algorithm that clusters pixels in a combined five-dimensional RGB color and XY image plane space to efficiently generate compact, nearly uniform "superpixel" regions [16]. The algorithm involves transforming the R, G, B color channels into the L, a, b channels of the CIELAB color space, then applying localized K-means clustering with a distance measure as follows:

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$$

$$D_s = d_{lab} + \frac{m}{S} d_{xy}$$

where (l_k, a_k, b_k) represents the CIELAB color of each pixel, (x_k, y_k) the spatial position of each pixel, and m an arbitrarily-chosen spatial weight variable controlling the regularity of superpixel shapes. The value D_s represents the computed distance metric between pixel k and cluster centroid i. Initial centroids for the K clusters are initially distributed evenly in low-gradient positions across the X, Y plane of the image, and each pixel is compared only to those clusters within a set spatial (L1) distance of $2S$, where $S = \sqrt{N/K}$ and N is the number of pixels in the image.

Applied to an image for a given number of superpixels K, the SLIC algorithm segments all pixels in the image into K connected superpixel regions, with inter-region borders coincident with edges and changes of texture in the image. Because superpixel segmentation algorithms such as SLIC are fast and require no training data and little parameter tuning, they can be used in image segmentation to apply an initial over-segmentation, followed by a selection and merging process to output components of adjacent superpixels. We use the same concept of oversegmentation followed by merging to first split each input spectrogram into superpixels, then identify connected sets of superpixels that make up syllables of bird call.

4.2.3 Modifications to SLIC

Although superpixel clustering is a useful way to reduce the number of entities to be classified for segmentation, it must be adapted in several ways for use in a spectrogram. Syllables of bird call are not regions of constant texture bounded by distinct edges, but instead are peaks of high energy separated by low-energy regions. Spectrograms generated in the preprocessing step are single-valued per pixel, and do not carry separate information in the R, G, and B color channels. So, rather than the (L, a, b) color

values of each pixel, we use a 5-valued vector comprised of computed per-pixel values that separate pixels into regions centering on syllable peaks in the spectrogram. The five computed values used are as follows:

- Blurred Spectral Energy

  The noise-filtered spectrogram **F** is convolved with a Gaussian kernel to generate a blurred spectral energy map B:

  $$\mathbf{B} = \mathbf{F} \star g, \text{ where } g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

  Blurred pixel energy for the spectrogram segmentation acts in a similar way to the L value used in SLIC segmentation. Distancing pixels with different energy values tends to cluster foreground pixels together, and using a Gaussian blur promotes connectedness of pixels close to each high-energy peak. A sigma value of $\sigma = 2.0$ is used.

- Weighted Variance of Spectral Energy

  The variance of spectrogram pixels within a window around each pixel is used as a second element in the superpixel clustering feature vector. A Gaussian function is again applied here, to weigh more highly pixels closer to the center of the window.

  $$\mathbf{V}(x, y) = \sum_{i=x-x_w}^{x+x_w} \sum_{j=y-y_w}^{j+y_w} g(i-x, j-y)(\mathbf{F}(i, j) - \mu)$$
  $$\text{where } g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

  Here, μ is the mean of pixel intensities in the window around (x, y), and x_w, y_w represent the window size. The inclusion of pixel variance as a feature in the pre-segmentation helps by promoting superpixel boundaries at the edges of high-energy regions, where syllables meet background. A window radius of $x_w = y_w =$ 5 pixels and a sigma value of $\sigma = 3.0$ are used.

- Horizontal Sobel Gradient, Vertical Sobel Gradient

  The distance measure used to cluster pixels includes the horizontal and vertical gradient values, obtained by convolving the Gaussian blurred spectrogram generated earlier with the Sobel gradient filter:

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{B} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{B}$$

  Including the two-dimensional gradient allows the superpixel segmentation to cluster together pixels on the rising or falling edges of a syllable. It promotes superpixel boundaries along the contours between high-energy peaks within each syllable, allowing each syllable to be broken up into meaningful sub-units.

- Nearest-Peak Time, Nearest-Peak Frequency

  A 'peak-finding' function is applied to each pixel, after which each pixel is assigned to a local maximum (or 'peak') close to it in the spectrogram. For each pixel, the weighted nearest peak is identified as the x,y position that satisfies the following constraints:

$$\mathbf{P_x}(x,y) = \arg\max_i \ \mathbf{F}(i,j)g(i-x, j-y)$$
$$\mathbf{P_y}(x,y) = \arg\max_j \ \mathbf{F}(i,j)g(i-x, j-y)$$

$$\text{where } i \in [x-x_w, x+x_w],\ j \in [y-y_w, y+y_w],\ g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

  The peak-finding function essentially finds, for each pixel, the position of the 'highest peak' close to that window, multiplied by a weight value that decays exponenitally with distance from the pixel. By applying this function, each peak is assigned

an area of pixels centered on itself with an average radius that increases with the log of the peak's height. This prevents rectangular-window artifacts and prevents large peaks from overwhelming smaller peaks within their window.

Including this feature in the superpixel generation assists in spatial locality, and promotes superpixel boundaries in regions between high energy peaks within syllables that contain multiple peaks.

Whereas the SLIC algorithm uses a single m parameter to control the weight given to spatial locality, the superpixel merger segmentation method uses a vector w of 5 weight factors. This provides us with a new distance measure for the superpixel clustering of spectrogram pixels:

$$d_b = \mathbf{B}(x_k, x_y) - B_i$$

$$d_v = \mathbf{V}(x_k, x_y) - V_i$$

$$d_g = \sqrt{(\mathbf{G}_x(x_k, y_k) - G_{xi})^2 + (\mathbf{G}_y(x_k, y_k) - G_{yi})^2}$$

$$d_p = \sqrt{(\mathbf{P}_x(x_k, y_k) - P_{xi})^2 + (\mathbf{P}_y(x_k, y_k) - P_{yi})^2}$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$$

$$D_s = w_0 d_b + w_1 d_v + w_2 d_g + w_3 d_p + w_4 d_{xy}$$

Here, the vector of values $[B_i, V_i, G_{xi}, G_{yi}, P_{xi}, P_{yi}, x_i, y_i]$ represents the value of centroid i in the K-means clustering. In evaluation, the weight vector $[5.0, 1.0, 1.0, 1.0, 1.0]$ provided sufficient results with a K value of 600. A more finely-tuned weight vector would fit superpixels more closely to meaningful features within the image, decreasing the number of superpixels required. The localized pixel clustering itself is run, with the defined distance measure, for 10 iterations. Initial centroids are evenly spaced throughout the spectrogram. The resulting segmentation is filtered to enforce spatial connectivity

of each superpixel segment, and superpixels of size less than a threshold (100 pixels) are merged into their nearest neighbor.

4.2.4 Foreground-Background Superpixel Classifier

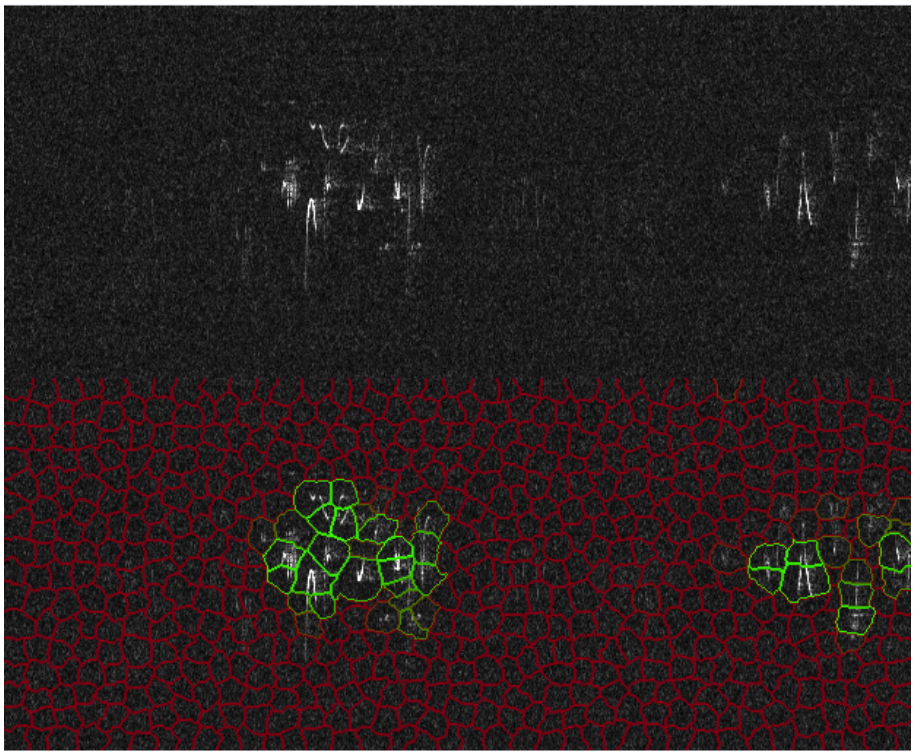


FIGURE 4.1: Example output of the superpixel classifier, foreground labeled superpixels outlined in green and background in red.

The Superpixel Merger method relies on two trained classifiers, the first of which is the foreground-background classifier. This is implemented as a Random Forest classifier, which takes feature vectors including the following values:

- Pixel Energy: The mean and the variance of all spectrogram pixels in **F** belonging to the superpixel to be classified are included in the feature vector.
- Gaussian Blurred Pixel Energy: The mean and variance of all spectrogram pixels in the Gaussian blurred mask, **B** are included.

- Nearest-Peak Frequency values: The mean and variance of the frequency (but not time) values of the peaks nearest to each pixel, ie. $\mathbf{P}_y(x,y)$ for all (x,y) belonging to the superpixel, are included in the spectrogram.

- Histogram of Oriented Gradients: A vector of 8 values is calculated from the distribution of edge directions in the pixels belonging to each superpixel. An angle $\mathbf{\Theta} = \text{atan2}\left(\mathbf{G}_y(x,y)\mathbf{G}_x(x,y)\right)$ is calculated for each pixel (x,y) that satisfies the condition $\sqrt{\mathbf{G}_y(x,y)^2 + \mathbf{G}_x(x,y)^2} > \epsilon$, for a small epsilon value. The gradient angle value is discretized into one of the HOG vector's 8 bins. An epsilon value of 0.01 was used for evaluations given in this thesis.

The classifier is trained by applying the superpixel segmentation to a training spectrogram, then labeling as foreground all superpixels that overlap more than 10% of their area with positive-labeled regions in the training mask. All superpixels in each training spectrogram are included in the training set.

4.2.5 Superpixel Merger Classifier

The second classifier in the merger method classifies edges between adjacent superpixels. Adjacent superpixels whose edges are classified as positive are merged together. The training process is as follows:

- An input spectrogram is filtered and segmented into a set of superpixels.

- An adjacency matrix is computed, indicating which superpixels share borders in the time-frequency plane.

- For each pair (i,j) of adjacent superpixels such that at least one of i or j is labeled foreground by the ground-truth mask, a feature vector is computed for (i,j)

- Each pair (i,j) of adjacent superpixels is labeled positive if any positive-labeled region in the training mask overlaps with the pixel border between i and j.

Edges between background superpixels are not considered part of the training set. The feature vector constructed to describe a pair of superpixels contains a concatenation of the features of each of the two superpixels' foreground classifier features, as well as the relative horizontal distance between the centers of the bounding boxes of the two superpixels. This results in a total feature vector size of $2N + 1$, where N is the dimensionality of the foreground/background superpixel feature vector. During training, the classifier is trained with two equivalently-labeled inputs per edge in the training set, one directed each way. During testing, the outputs of the classifier for each pair (i, j) and (j, i) are averaged to determine the edge weight.

4.3. Output and Analysis

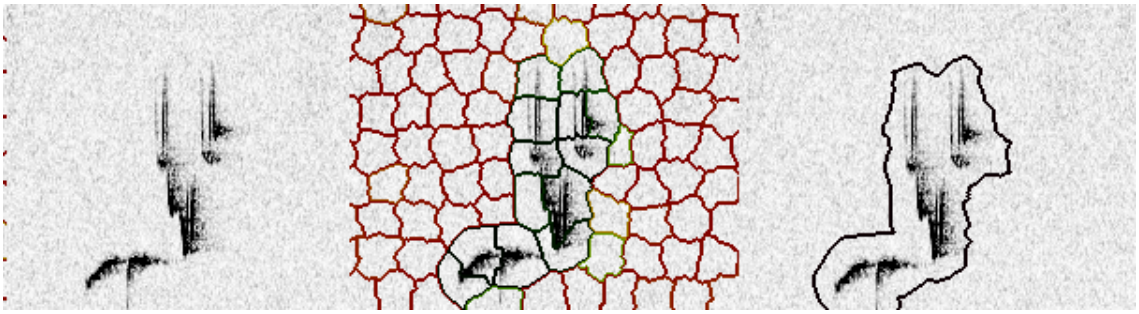


FIGURE 4.2: An example of the superpixel pre-segmentation of a syllable, followed by foreground/background filter and merging of superpixels

After input audio data are converted to spectrogram form, noise-filtered, and pre-segmented into the initial superpixel set, the foreground/background classifier is applied to each superpixel, to generate a probability value p_i for the superpixel, analagous to the probability values output by the per-pixel method. A threshold parameter θ separates foreground and background by discarding all superpixels for which $p_i < \theta$.

After only foreground superpixels remain, the merger classifier is applied to each adjacent pair (i, j) of remaining superpixels, generating a probability value m_{ij} for the

pair. Note that the classifier is applied twice for each pair, once for each of the two possible orderings. A second threshold parameter δ is applied, and all adjacent superpixels for which $m_{ij} + m_{ji} \geq 2\delta$ are labeled as the same segment. Each labeled segment is subsequently extracted from the spectrogram and output in the same fashion as the per-pixel classifier.

5. EVALUATION

5.1. Data Sets

Two data sets were used in the evaluation of the two proposed methods. The first consists of an annotated set of 625 audio segments, each 15 seconds, collected in 16kHz PCM format. The audio segments are selected, two per hour, from a 24 hour recording at each of 13 sites across the H.J. Andrews Experimental Forest. These data were recorded between May and July 2009. The manual ground-truth labels in this set tend to include a larger area around each syllable, and have more positive examples of mergers between adjacent superpixels.

The second data set consists of 166 annotated audio segments, 10 seconds each, collected in the same format. The audio segments are paired when possible, one from each of the years 2009 and 2010, with two segments selected from each of several days in the season, ranging from May 13 to August 4. Recordings from thirteen sites are used. For each site/day/year, at least two segments are included: one each from the early morning (4:30am to 6:00am) and one from the late morning (6:00am to 8:00am). The manual labels for this set cover less area around each syllable, and include more divisions between syllables.

Due to the constraints of training and testing time, two-fold cross-validation is used for each of the evaluations given. Each evaluation metric is tested once for each of the two methods, for each of the two data sets.

5.2. Evaluation Metrics

5.2.1 Time-Frequency Area Metric

This error measure is concerned only with the number of spectrogram pixels correctly classified, aggregated over the full set of spectrograms. Each pixel in every input spectrogram is classified as either a True Positive, False Positive, True Negative, or False Negative. First, a threshold parameter θ is selected, and pixels in the output mask with a foreground probability less than this value are zeroed from the probability mask. The manual human-labeled mask over the spectrogram is loaded, and all positive-labeled pixels in it are classified as True Positive, if their element in the probability mask is nonzero, or False Negative otherwise. All negative-labeled pixels in the manual mask are labeled True Negative if their correpsonding output probability is zero (below threshold), and False Positive otherwise.

The parameter θ is varied, to plot a curve of values ranging from all-negative to all-positive classification. For each θ value, the True Positive Rate and False Positive Rate are calculated:

$$TPR = \frac{TP}{TP+FN} \quad FPR = \frac{FP}{FP+TN}$$

Each calculated (FPR, TPR) pair is plotted on a Receiver Operating Characteristic (ROC) curve, to show the tradeoff between precision and recall at varying θ values.

5.2.2 Segment Recall Metric

This metric simply measures the number of connected regions of positive-labeled pixels (ie, syllables) marked by the manual segmentation that were labeled as true by the segmentation method, divided by the total number. Each positive-labeled region is counted as a recall if any part of its area forms part of a segment output by the segmentation process. To compare varying methods equivalently, segment recall is plotted against the area metric's True Positive Rate, which ranges from 0 to 1 for the evaluations of each

method. This displays the number of segments sucessfuly recalled for a given number of pixels. This metric will penalize a method that maximizes the recall of pixels from large syllables, at the expense of smaller ones. Note that for a given TPR, the FPR will vary, so the segment recall graph should be referenced against the ROC curve of time-frequency area.

5.2.3 Segment Merger Error

The merger error metric measures the proportion of separate positive-labeled syllable regions in the manual labels that are joined together by at least one positive-labeled syllable region in the output segmentation. A merger error of 0 indicates that no two syllables are joined by an output segment. A merger error of 1 indicates that every syllable is joined to at least one other. This metric penalizes methods that indiscriminately merge adjacent syllables into single segments. This is a drawback particularly of the per-pixel time-frequency method. Merger error is also plotted against TPR from the previous ROC metric, and should be referenced against it.

5.3. Energy Threshold Method

To compare both methods to a non-learning segmentation scheme, an energy thresholding segmentation is evaluated against the same data sets. This segmentation involves the following steps:

- Spectrograms are generated from input audio and noise-reduced using the same parameters as both learning methods.
- A Gaussian blur with $\sigma = 6.0$ is applied to the noise-reduced spectrograms.
- Each spectrogram is normalized such that its maximum value is 1.0

- A threshold θ is applied to the blurred spectrogram, and all contiguous regions above that threshold are extracted as syllables.
- Syllables with area less than a threshold of 100 pixels are removed

5.4. Results

Results are given for each of the three defined error measures, based on the output of each of the three given classifiers, on two data sets.

5.4.1 HJA 625-Spectrogram Set

The ROC curve shows the superpixel merger method out-performing the per-pixel method at false positive rates below about 10%, although the per-pixel method provides slightly higher recall with lower precision.

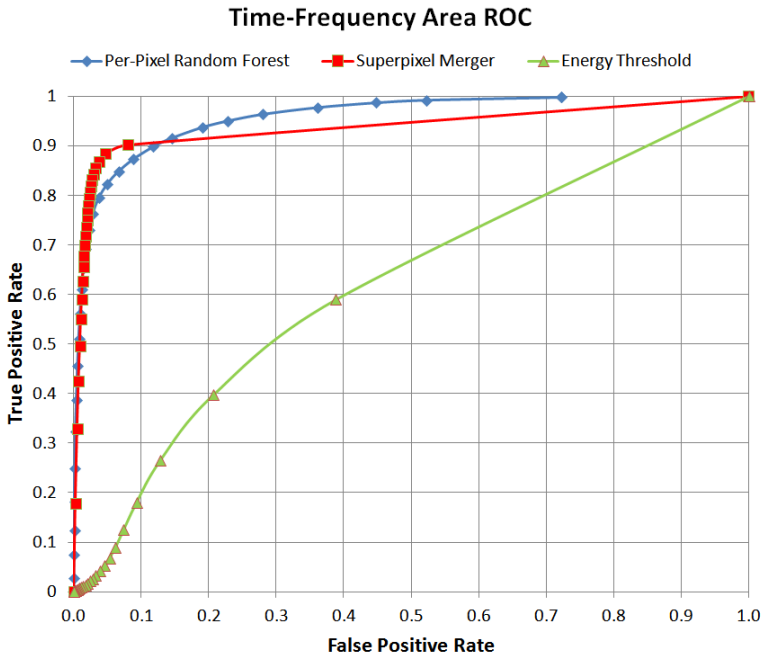


FIGURE 5.1: ROC curve for time-frequency area, tested on the 625-spectrogram HJA dataset

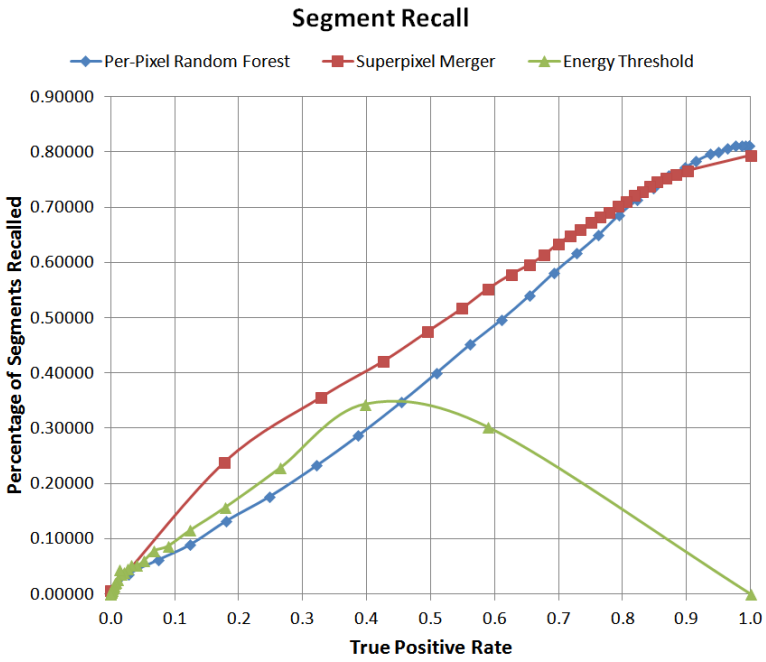


FIGURE 5.2: Segment recall, tested on the 625-spectrogram HJA dataset

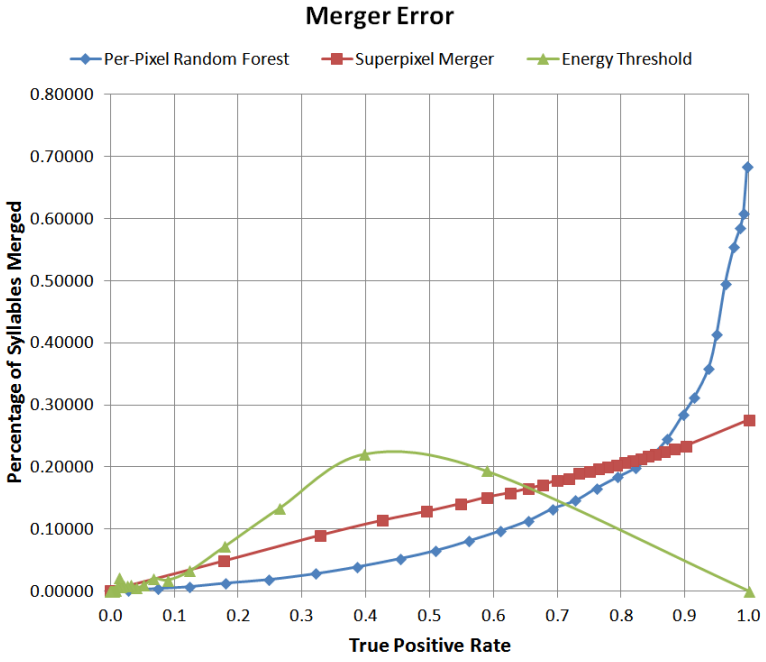


FIGURE 5.3: Merger error for each method, tested on the 625-spectrogram HJA dataset

5.4.2 "Set A" 166-Spectrogram Set

The ROC curve for pixel accuracy shows the per-pixel method slightly outperforming the superpixel method at lower false-positive rates, with the superpixel method taking the lead for FPR of greater than about 0.04. The merger error, however, shows a problem with the per-pixel method: at θ levels low enough to capture more than 90% of the true bird syllables, the per-pixel method merges many adjacent segments together. This is more apparent in the Set A dataset due to the greater separation between syllables in the manual masks that segmentations are compared to.

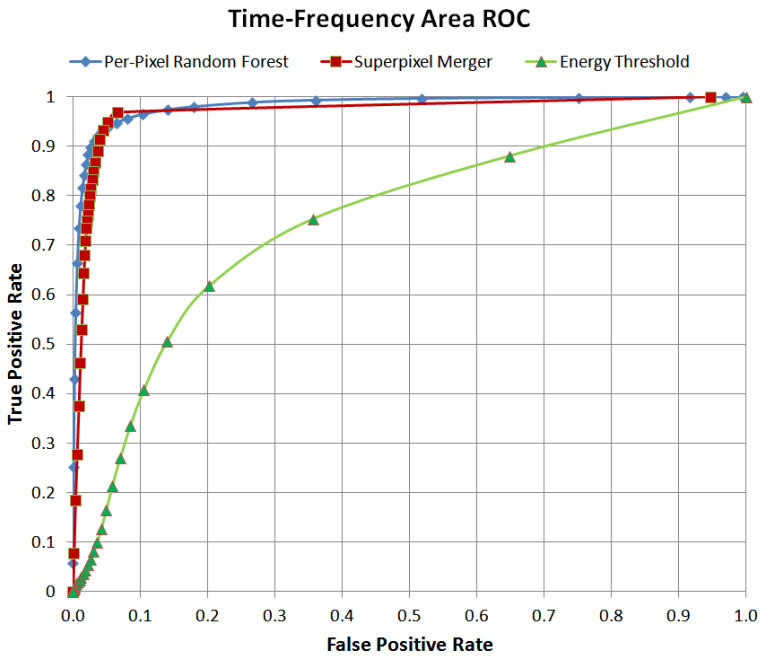


FIGURE 5.4: ROC curve for time-frequency area, tested on the 166-spectrogram Set A dataset

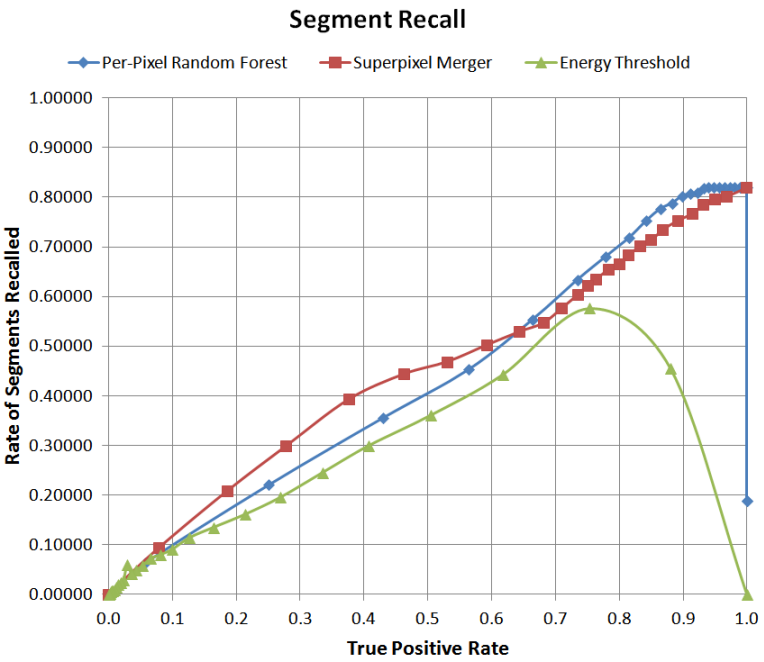


FIGURE 5.5: Segment recall, tested on the 166-spectrogram Set A dataset

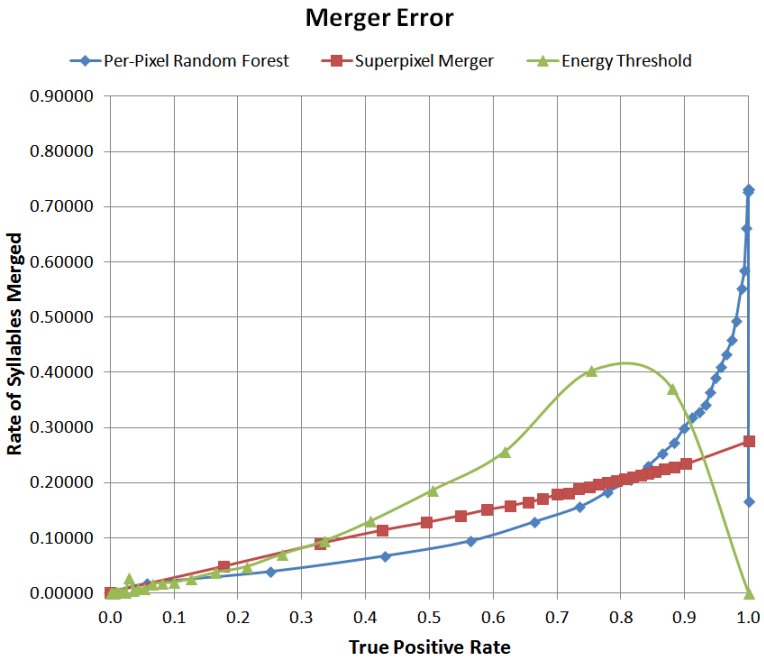


FIGURE 5.6: Merger error for each method, tested on the 166-spectrogram Set A dataset

6. CONCLUSION AND FUTURE WORK

This thesis has explained the importance of time-frequency syllable segmentation as part of a bird species classification system. Two methods were proposed for the extraction of bird call syllables from the time-frequency spectrogram. Evaluation of the two methods shows that each achieves higher recall and precision for certain inputs.

Although it is not noted in the evaluation metrics, the running time for superpixel segmentation is characteristically much less than the per-pixel method. Restricting the number of Random Forest trees or the size of feature vectors to achieve a real-time or faster speed results in slightly degraded performance for the superpixel method, but highly degraded performance for the per-pixel.

Further improvements to the superpixel method could include the use of a more sophisticated graph cut algorithm to separate sets of superpixels. Additional features that describe the joint characteristics of adjacent pairs of superpixels could also improve the superpixel method. Further additions, such as hierarchical merging of superpixels or the use of an ensemble of superpixel pre-segmentations with different weights could also improve the superpixel method's accuracy at the expense of runtime.

BIBLIOGRAPHY

1. Gilbert, Frederick F.; Allwine, Rochelle. 1991. *Spring bird communities in the Oregon Cascade Range*. In: Ruggiero, Leonard F.; Aubry, Keith B.; Carey, Andrew B.; Huff, Mark H., tech. eds. Wildlife and vegetation of unmanaged Douglas-fir forests. Gen. Tech. Rep. PNW-285. Portland, OR: U.S. Department of Agriculture, Forest Service, Pacific Northwest Research Station: 145-158.

2. Hansen, Andrew J.; McComb, William C.; Vega, Robyn; Raphael, Martin G.; Hunter, Matthew. 1995. *Bird habitat relationships in natural and managed forests in the west Cascades of Oregon*. Ecological Applications. 5(3): 555-569.

3. Rosenstock, Steven S.; David R. Anderson, Kenneth M. Giesen, Tony Leukering and Michael F. Carter *Landbird Counting Techniques: Current Practices and an Alternative* The Auk, Vol. 119, No. 1 (Jan., 2002), pp. 46-53

4. Ellis, D.P.W. *Prediction-driven computational auditory scene analysis* Ph.D. thesis, Dept. of Elec. Eng & Comp. Sci., M.I.T., June 1996.

5. Forcey, Greg M.; James T. Anderson, Frank K. Ammer and Robert C. Whitmore. *Comparison of Two Double-Observer Point-Count Approaches for Estimating Breeding Bird Abundance* Greg M. Forcey, James T. Anderson, Frank K. Ammer and Robert C. Whitmore The Journal of Wildlife Management, Vol. 70, No. 6 (Dec., 2006), pp. 1674-1681

6. Briggs, Forrest; Raviv Raich, Xiaoli Z. Fern. *Audio Classification of Bird Species: a Statistical Manifold Approach* Proc. International Conference on Data Mining, ICDM 2009

7. Briggs, Forrest; Balaji Lakshminarayanan, Lawrence Neal, Xiaoli Fern, Raviv Raich, Matthew G. Betts, Sarah Frey, and Adam Hadley. *Acoustic classification of multiple simultaneous bird species: a multi-instance multi-label approach.* Accepted pending revision, Journal of the Acoustical Society of America, 2012.

8. Lakshminarayanan, B; R. Raich, and X. Fern. *A syllable-level probabilistic framework for bird species identification.* Proc. IEEE International Conference on Machine Learning and Applications, 2009.

9. Breiman, Leo *Random Forests* Machine Learning, Vol. 45, No. 1. January 2001, pp.5-32

10. Fagerlund, Seppo. *Bird species recognition using support vector machines* EURASIP Journal on Applied Signal Processing. January 2007, p. 64

11. Harma, A. *Automatic identification of bird species based on sinusoidal modeling of syllables* IEEE International Conference on Acoustics Speech and Signal Processing. April 2003, pp. 545-548

12. Li, Yipeng; DeLiang Wang *On the optimality of ideal binary time-frequency masks* IEEE International Conference on Acoustics Speech and Signal Processing. 2008, pp. 3501-3504

13. Brown, Guy J.; Martin Cooke. *Computational auditory scene analysis*. Computer Speech and Language (1994) 8, 297-336.

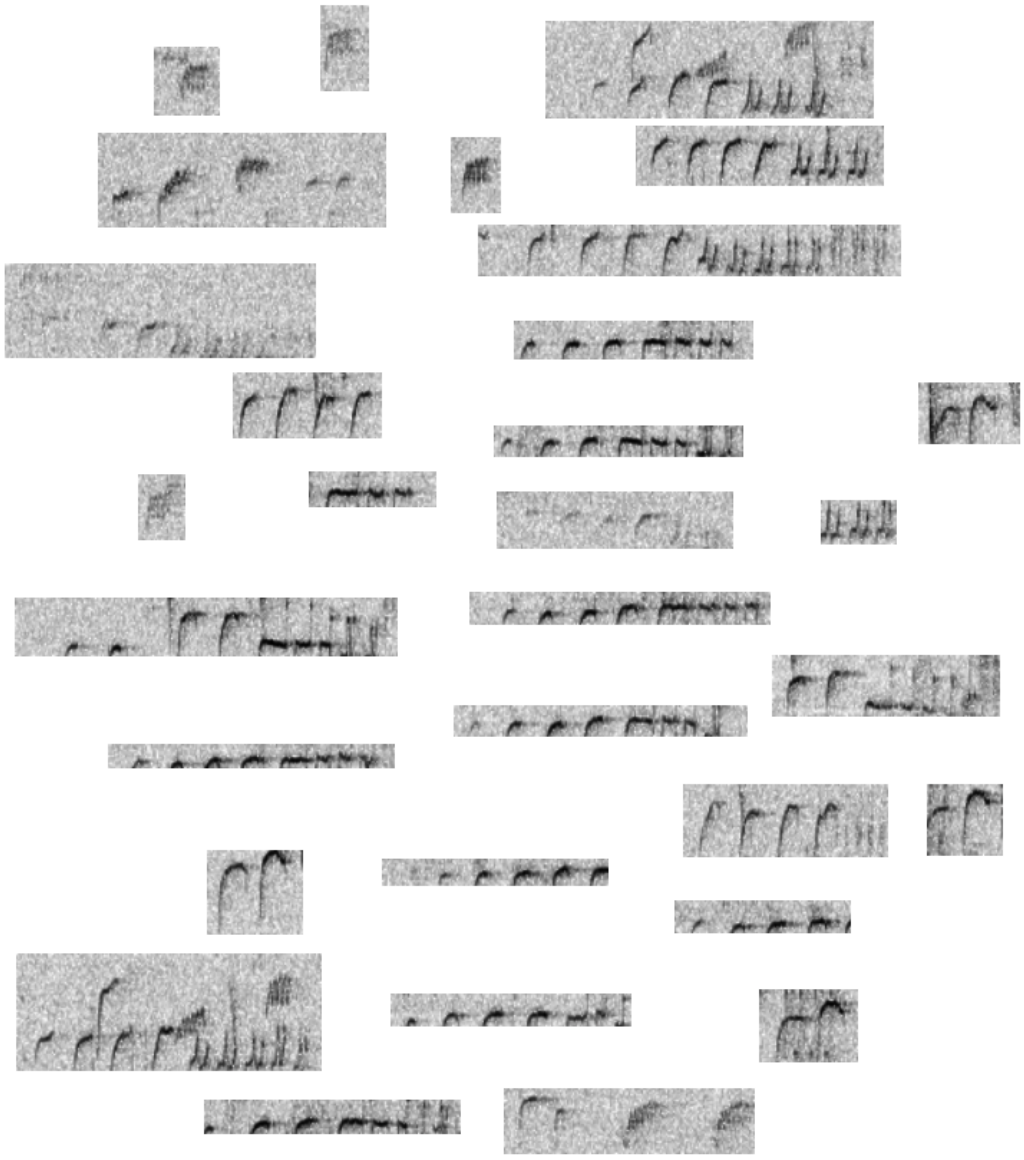
14. Li, Yan; David M.J. Tax, Marco Loog *Supervised Image Segmentation with Scale Invariance* ASCI - IPA - SIKS tracks, ICT.OPEN, Veldhoven, November 14-15, 2011

15. Lo, Edward H. S; Mark R. Pickering, Michael R. Frater and John F. Arnold *Image Segmentation using Invariant Texture Features from the Double Dyadic Dual-Tree Complex Wavelet Transform* IEEE International Conference on Acoustics Speech and Signal Processing, 2007.

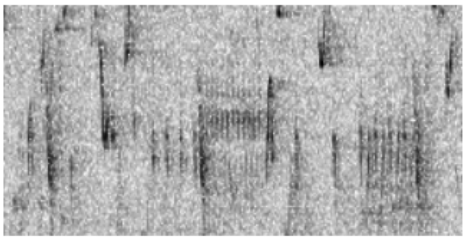
16. Achanta, Radhakrishna; Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. *SLIC Superpixels* EPFL Technical Report no. 149300, June 2010.

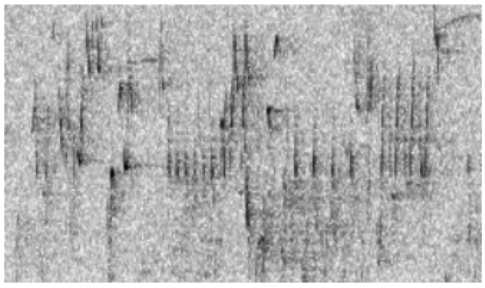
APPENDIX

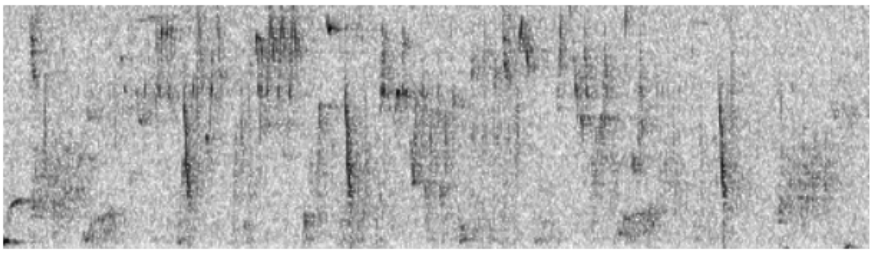
BRCR – Brown Creeper

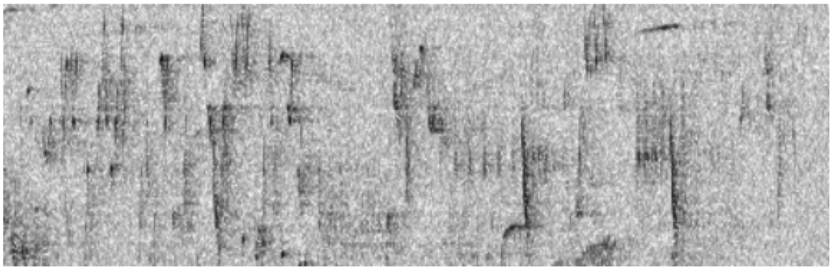


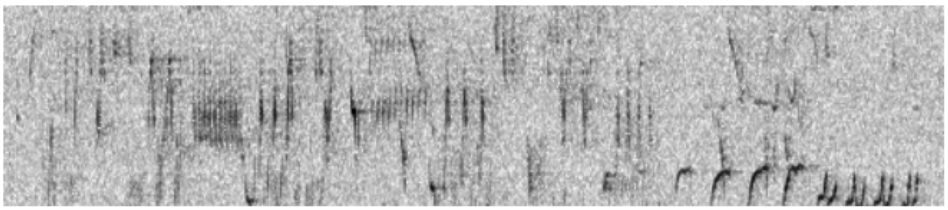
WIWR – Winter Wren



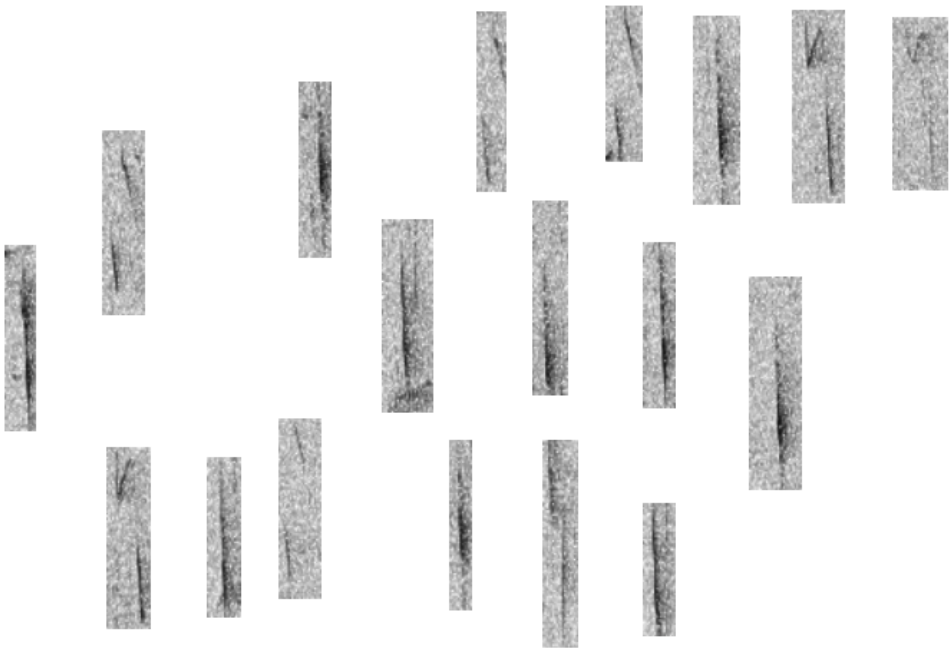




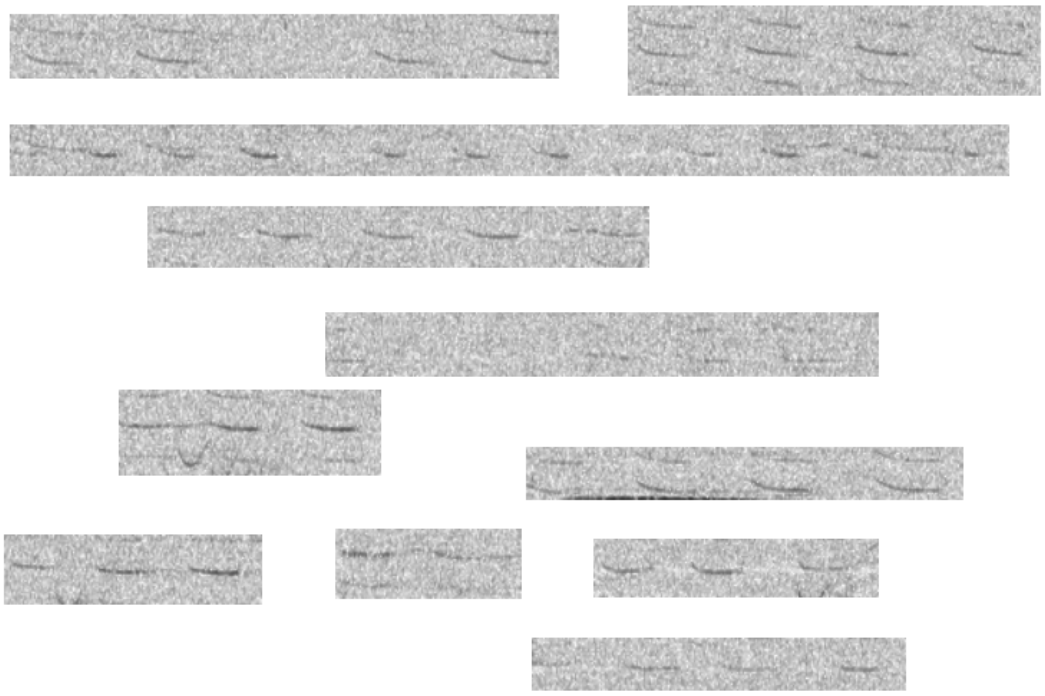




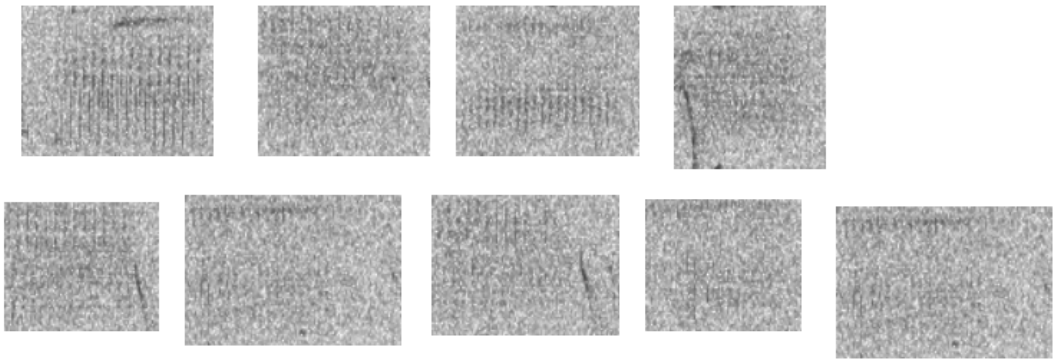
PSFL – Pacific-slope Flycatcher



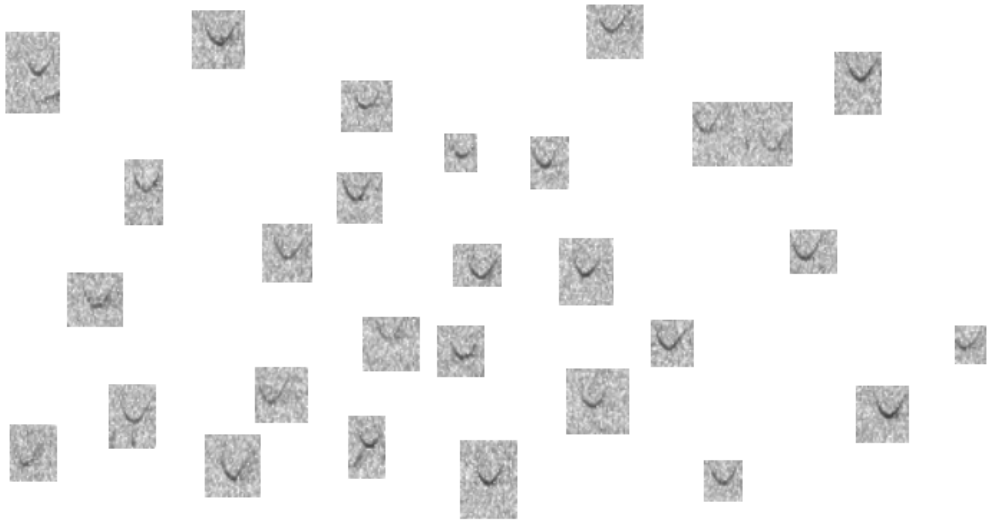
RBNU – Red-breasted Nuthatch



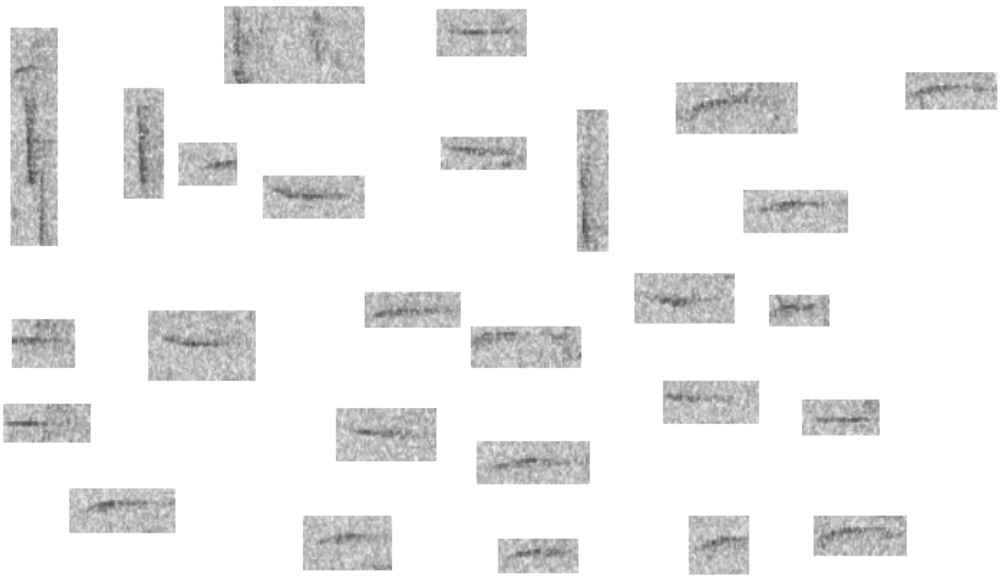
DEJU – Dark-eyed Junco



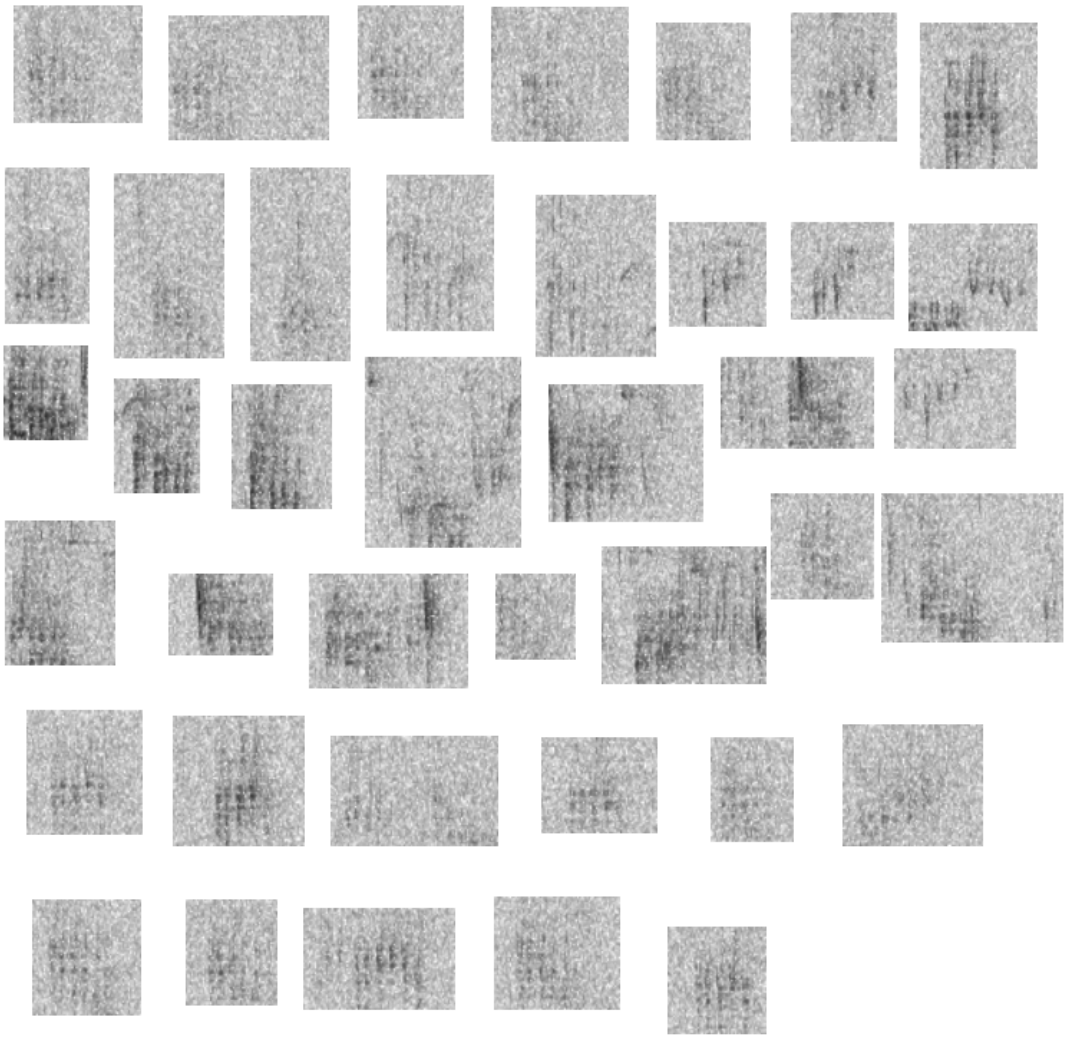
OSFL – Olive-sided Flycatcher



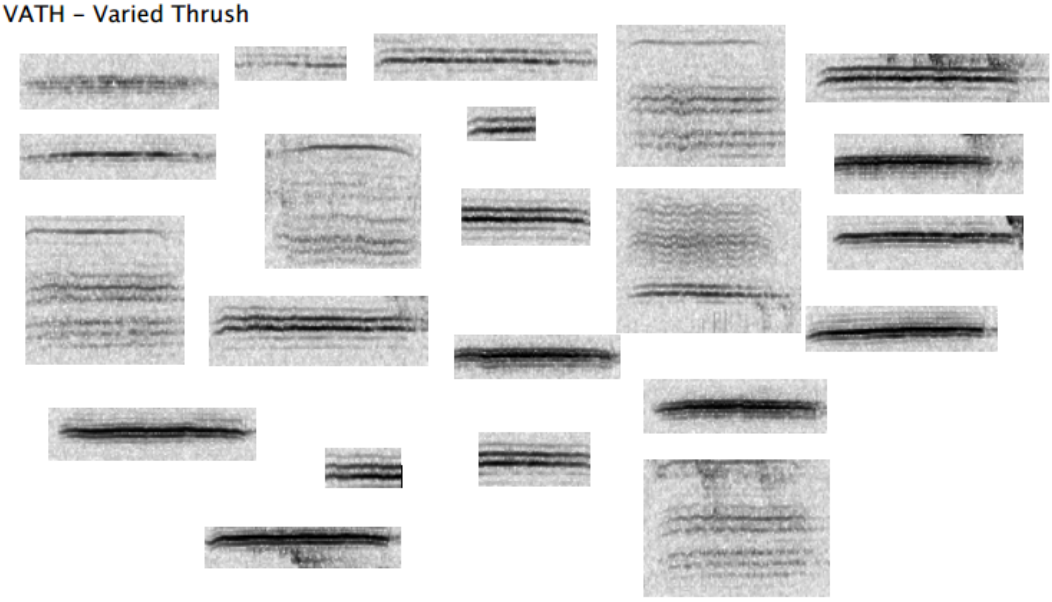
HETH – Hermit Thrush



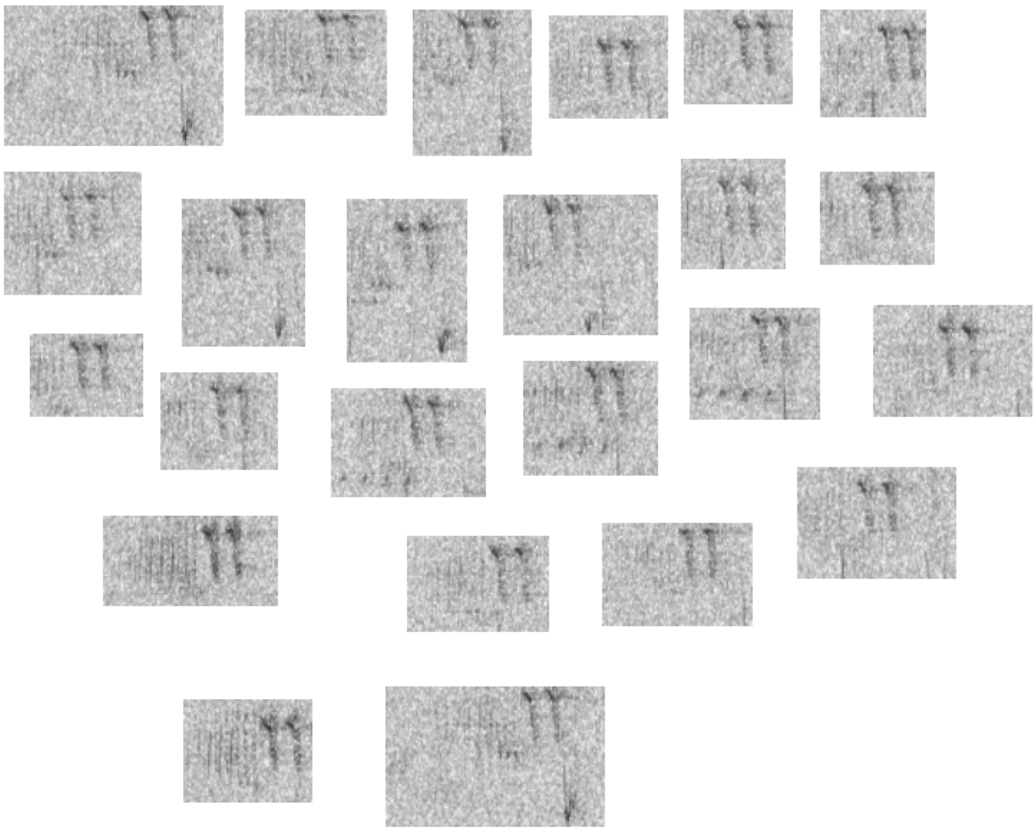
CBCH – Chestnut-backed Chickadee



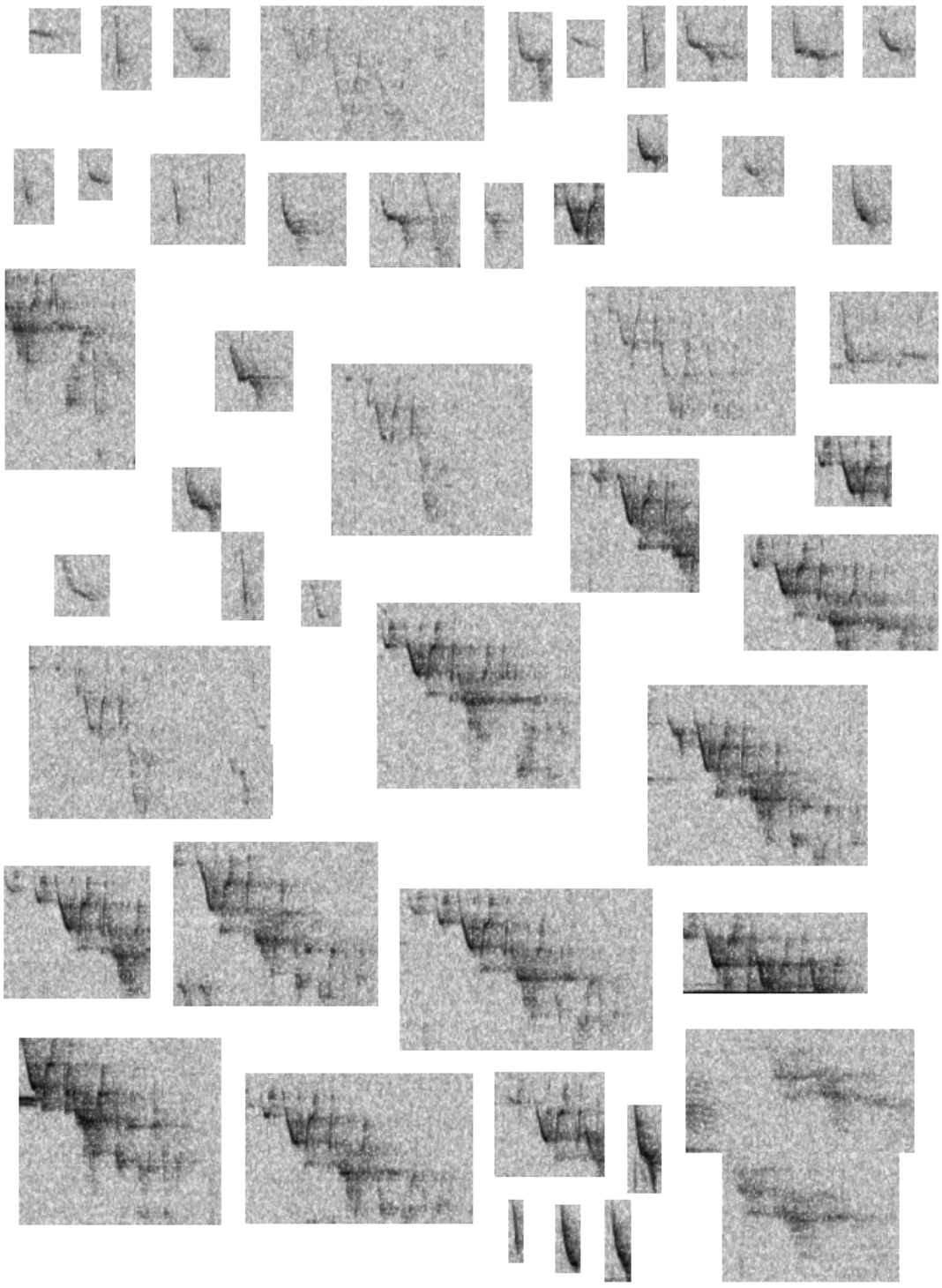
VATH – Varied Thrush



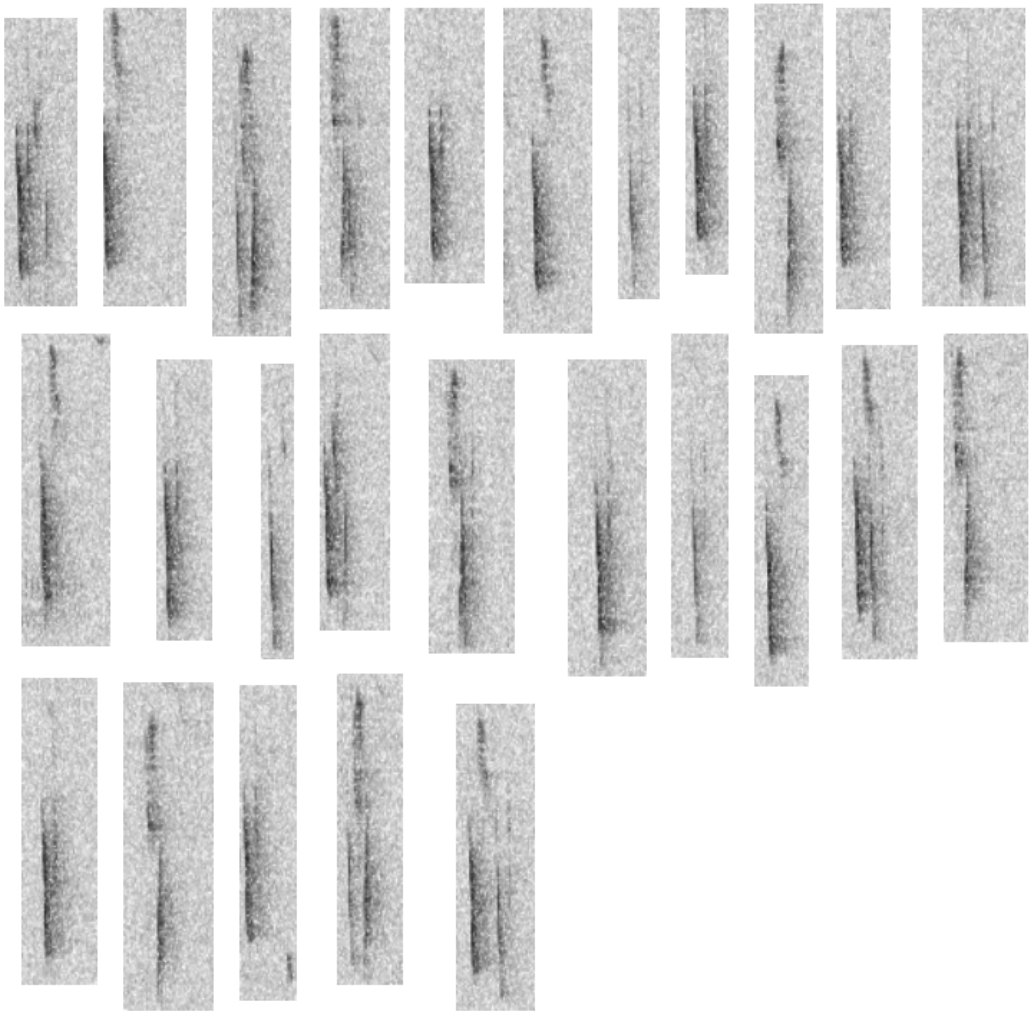
HEWA – Hermit Warbler



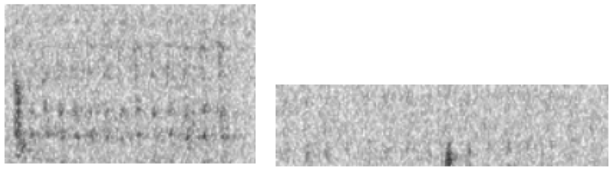
SWTH – Swainson's Thrush



HAFL – Hammond's Flycatcher



STJA – Steller's Jay



WETA – Western Tanager