Bird Song Recognition through Spectrogram Processing and Labeling

Katie Wolf
University of Minnesota - Twin Cities
wolfx265@umn.edu

Abstract

In this work, we present solutions to two problems on a project developing an end-to-end system for collecting and analyzing bioacoustic recordings, in particular bird songs. This system will conduct automatic bird species survey, which involves gathering the presence and absence of information as well as an abundance of information of different bird species. The two problems discussed in this work are that (1) there needs to be an effective way to obtain manually collected data for the system to initially learn and train from and (2) a noise reduction algorithm is needed to allow for effective gathering and labeling of information because the recordings from the environment tend to be noisy.

1 Introduction

Collecting data from wild animals in outdoor environments and analyzing them has been a tedious task that some field workers have little enthusiasm for, considering it difficult, dull and old-fashioned work [2]. However, modern technology has enabled scientists to accomplish this task in a way that was not possible years ago. The spectrogram was a turning point in bird song research. It made it possible to analyze, measure, classify and recognize the different sounds a bird makes. Today, with digital and computer-based technology, the power and speed of the latest generation machines allows for easier storage of data as well as being able to manipulate and synthesize for experimental purposes [2].

While data is more easily available, the vast amount of continuous sound recordings often take, at a minimum, equivalent time to analyze as to acquire. The use of automated recognition is a developing area of great opportunity for analyzing continuous data [3]. However, even to use a machine learning method, a training set of properly labeled data is necessary, but labeling data is a time consuming process. The project concentrates on the problem of cleaning the data and facilitating human labeling for use with machine learning algorithms.

In this paper, Section 2.1 will provide an overview of bioacoustics and past research involving automated recognition and bioacoustics. In Section 2.2, an explanation of spectrograms will be discussed and how they were used in this project. Section 3 will give an overview of the project. The program to assist in labeling data will be discussed in Section 4, and Section 5 will explain the algorithms used for cleaning the data. A final summary will be discussed in Section 6.

2 Background

2.1 Bioacoustics

Bioacoustics pertain to the sounds that animals make and can often provide insight to their behavior. Bioacoustics research and tools can aid in monitoring and managing species, which is vital to the conservation and preservation of diversity. Currently, capturing and monitoring of bird species through marking individuals with radio monitoring devices or visual tags is sometimes necessary for biosurveys. The survival and successful reproduction of these captured individuals may be affected. Using bioacoustics may be one way in which to expand conservation efforts and shun this type of handling [3].

Using sound to identify certain species is not a new idea. T. A. Parker recorded the dawn choruses of bird in the Bolivian Amazon, and within 7 days he found he had captured 85% of the regional species on tape. In that same region, seven experienced ornithologists took 54 days to inventory the birds using a capture and release technique [3].

The basis of the project is to study and use bioacoustical recordings from birds to create a machine learning algorithm that, taking in an audio stream, will be used as an automated species recognition tool. Automated recognition of bioacoustics signals has been re-

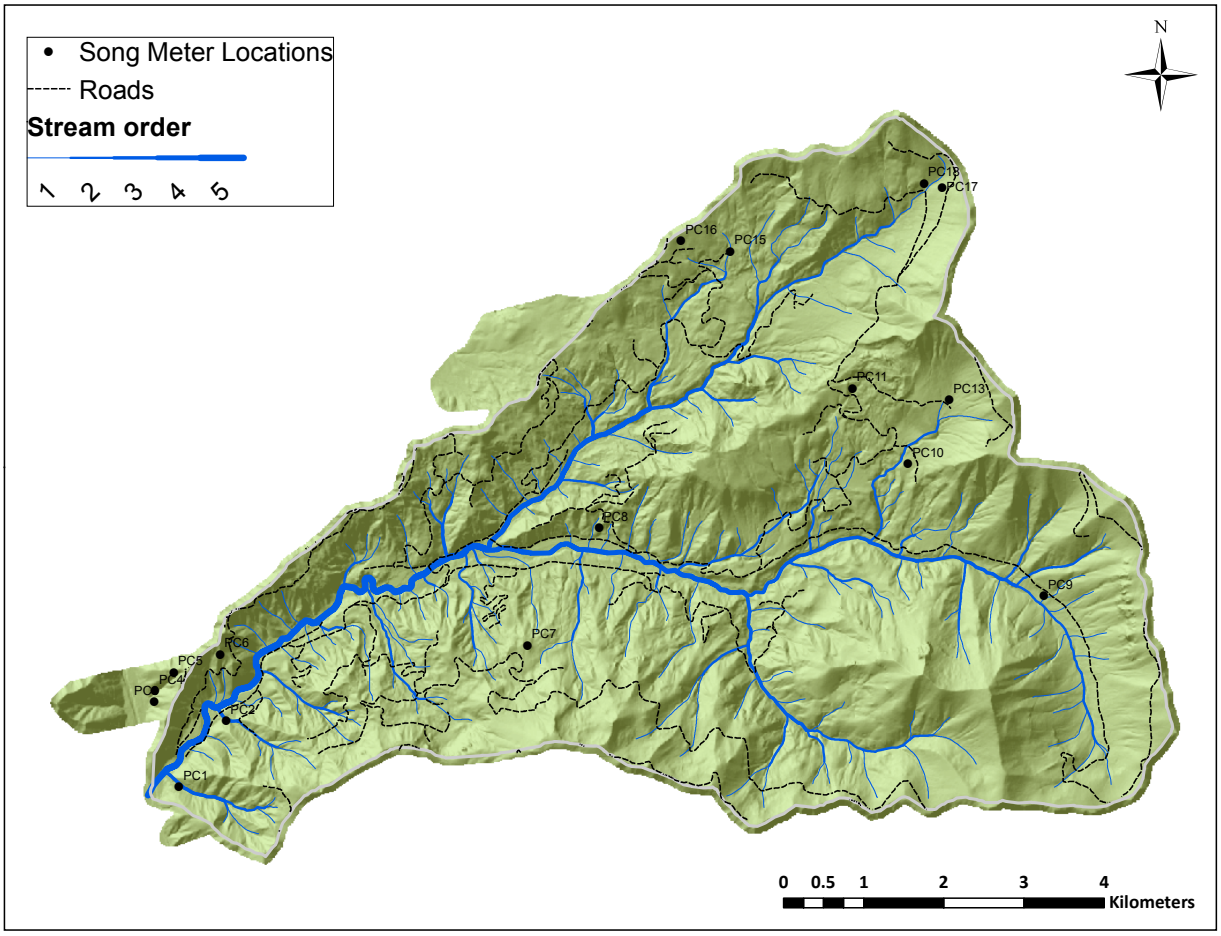

Figure 1. Map of the H.J. Andrews Research Forest. Each dot represents a Song Meter recording data, while the lines running through the valleys represent the water sources.

ported with encouraging results in a variety of animals including bats, birds, frogs and Orthoptera (grasshoppers, crickets and locusts) [1]. Birds species recognition in particular has been studied using support vector machines [4], sinusoidal modeling [5], hidden Markov models [6], and dynamic time warping [6]. Using pre-recorded and pre-labeled data from the Cornell Maculauy Sound Library was a way in which to create a database of sounds and species labels for the machine. However, there were many challenges to the project when faced with the raw data that was collected from the H.J. Andrews Research Forest including noisy audio files and unlabeled data.

2.2 Spectrograms

Spectrograms are graphical representations of audio files, with time on the horizontal axis and frequency (going from low frequency at the top of the image and high at the bottom) on the vertical axis. The images of the graphs are what we used on the project and are generated from the audio files. For each of the audio files, we divide the sound of the signal into frames and compute the spectrum for each of the frames. A spectrum represents the intensity of the signal as a function of frequency. The spectrogram is then created as a graph of the spectra of each frame in the sound. In our case to create the spectra we divided the signals into frames with a *frame-size* = 1024 samples. With each frame starting a *frame-step* = 256 samples after the previous frame, we have a 75% overlap between

frames. Then for each of the frames, we take the fast-Fourier transform (FFT) complex coefficients c_i for i = 1,...,m where m = *frame-size*/2 = 512. One way in which we remove the noise in the lower frequency range is by dropping the lowest 13 FFT coefficients. This leaves us with n =499 elements in each spectrum which are used to create the spectrogram.

Figure 2. The song Meter in its location in the forest.

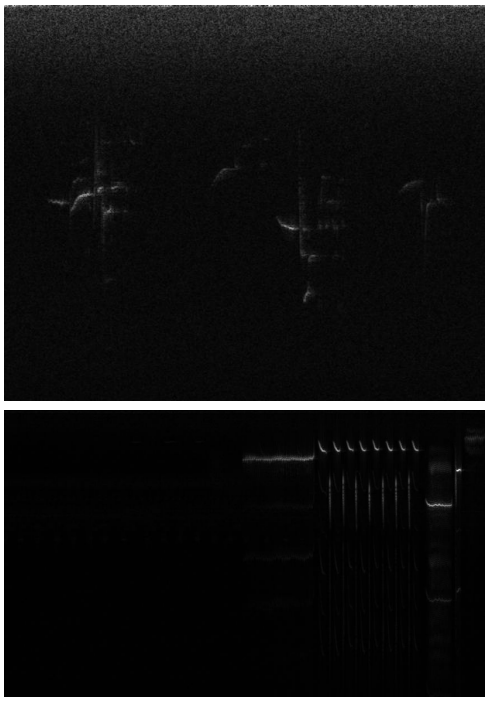

Figure 3. A clairity comparison can be seen between the spectrogram created with an audio file from the H.J. Andrews Research Forest (top) versus a spectrogram generated from an audio file from the Cornell Maculauy Sound Library(bottom)

3 Project

Within the H.J. Andrews Research forest, Song Meters (Figure 2) used to record data were placed at 16 locations, and, as can be seen in Figure 1, most of the surrounding area is near moving water. Each recording has some noise or echo of a noise from the rivers and streams that run through the forest. Figure 3 shows the difference in clarity between a raw image from the forest and an image from the sound library. On the top image there is a high concentration of noise near the low frequencies not found in the bottom image, which represents the noise that is coming from the water. These bright areas that are not a part of the data we are trying to collect, also called the 'noise', tend to dim the actual bird songs or 'signals' which we are trying to preserve.

One of the goals of our research is to create an algorithm, or a series of algorithms that will remove these areas or 'de-noise' the image, so that the signal is clear to one reviewing the spectrograms. There are a few reasons why we need to visibly see the signals within the image. One reason is that for the machine learning algorithms to successfully train and categorize data they will need the cleanest images possible. Another is that for our initial labeling of the raw data we needed ecologists to go through each recording and manually label the signals. In order for them to correctly see and label the data, it has to be clearly displayed. This brings us to the other part of our research: creating a user interface that will allow labeling spectrograms by species in order to create a training set of data for the machine learning algorithms.

4 Annotation

Our first part of the research was to create a program that would take in the audio wave, compute the FTT coefficients in order to create a spectrogram, and from there use a GUI program that would allow a user to label parts of the spectrogram. The initial requirements for the program were to have it:

- Break up a large audio file into smaller ones so that the spectrograms would fit on the screen. This was done by prompting for the size (in seconds) of the smaller segments.
- Play the audio files for the small segments.

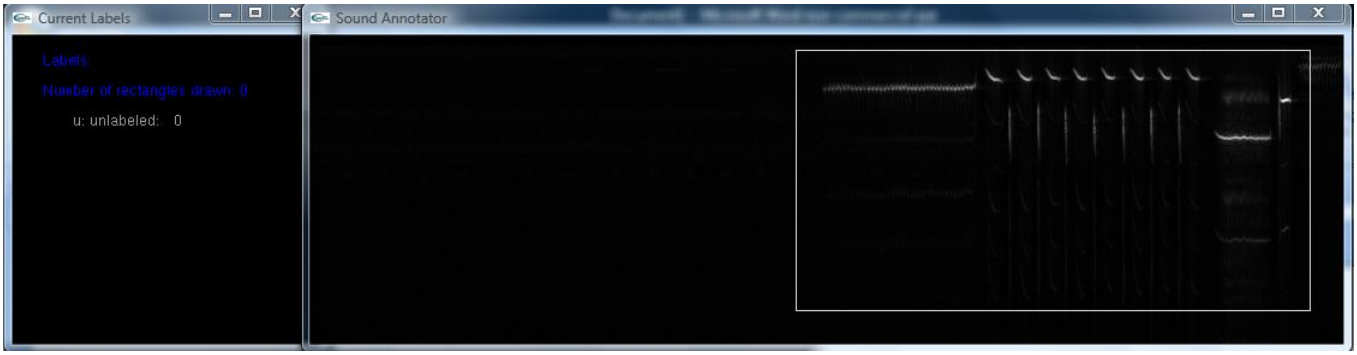

Figure 4. The annotator program with the left window representing the current labels and the right window represents the annotation window where the users label the spectrograms

- Flip through the segmented spectrograms (moving both forward and backward while still preserving the labels that were assigned.

- Contain a labeling system to tell the difference between types of species.

- Save a text file for each of the individual segments with information about the spectrogram (including frame-size and frame-step) and information on the labels (including the coordinates for opposite points of the annotated rectangle and the label that was assigned to that annotation).

The initial program met these requirements and had all of the basic tools needed to label the data. After working with the program we noticed that while the labeling system worked well, another window was needed in order to view past labels while still labeling the current spectrograms. This would not only keep the labeling more consistent, but would allow the user to notice any incorrect labels immediately. A second window would also be able to keep track of which key represented which labels and give more options to the user as they did there annotations on the main window. Figure 4 depicts the basic display of the two windows. For more on the program's functionality, view the user manual [11].

In order to test the program we were able to go through and manually label a day of data which consisted of 20 minute segments of sound for each hour of the day. Figure 5 is a graph of the number of bird calls that were labeled per hour for each hour. As can be seen the results show that the bird calls are mostly concentrated during the early morning between 5:00 AM and 11:00 AM. This is consistent with the well known fact that birds have a peak of singing activity in the early morning, called the dawn chorus [2], and also consistent with the information given from the ecologists who work within the research forest. Between 5:00 AM and 9:00 AM they conduct point counts which consist of going from one site to another every 10 minutes and counting the number of each bird species. We have yet to compare their counted data with data from the program that was taken at the exact site and time.

5 Noise Reduction

In this section we specify algorithms used to clean the images so that the signals are easier to identify.

5.1 Wiener Filter

Our attenuated Wiener Filter algorithm used to estimate the noise from the signal and remove it is composed of the following steps:

1. First, boost the image contrast of by setting each pixel in the image to the square root of its value (the pixel values range from 0 (black) to 1 (white)).

2. Run a low band pass filter that removes 8% of the pixels with the lowest frequency (those that are closest to the top of the image) by setting their value to 0.

3. For each frame (columns of pixels) compute the sum of all of the bins (rows of pixels) in that frame.

4. For each of the bins, compute the bin sum for the lowest 20% of the frame sums (found in 2) and then for each pixel set the value to the value divided by the bin sum (of the lowest 20% found previously) of the bin that the pixel is in.

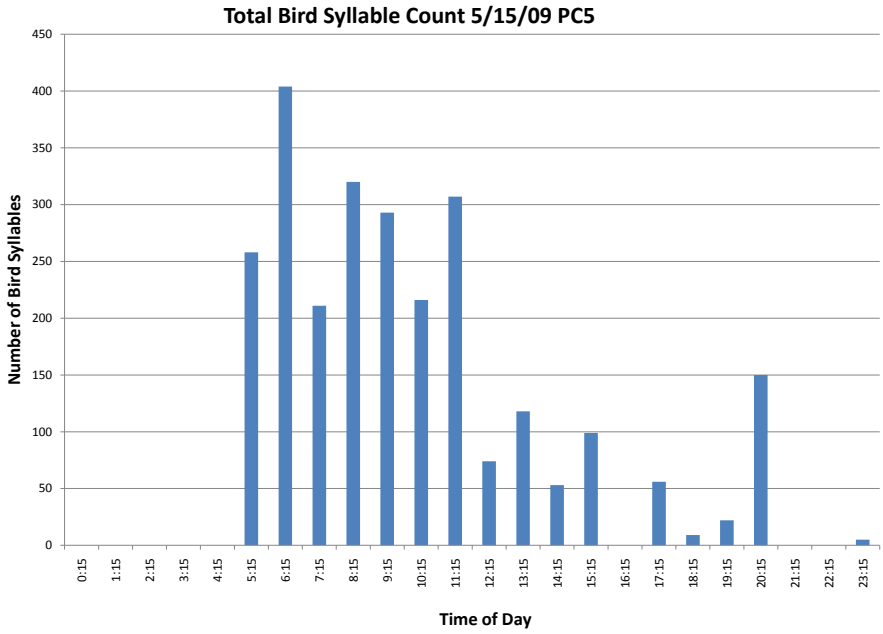

Figure 5. Graph of the number of bird songs annotated for a day of data (20 minutes of each hour starting at the time listed).

5. Apply 3 and 4 again on the modified values of the pixels.

To minimize the least mean squared error between the filter output and the desired signal, Wiener filter coefficients are calculated. In a basic Wiener filter it is assumed that the signals are stationary processes. Because in our process we are dealing with a variable signal but a stationary noise, we use an application of the Wiener filter for additive noise reduction [10].

In our signals, we know the noise (from the moving water in the forest) is located predominantly at the low frequencies of the spectrograms and that they also stay consistent throughout the frames. In our algorithm above, by first computing the sum of the bins for each frame, we are using the lowest sums as estimates of where the noise is alone without a signal. Then by taking a sum of these estimates for each bin and dividing each pixel in the bin by that sum we are removing our estimation of the 'noise' from the image. Once we have completed this we apply steps 3 and 4 again to smooth out the bins. The top three images in Figure 6 illustrates this algorithm.

5.2 Smoothing Filters

Once we have removed most of the constant low frequency noise from the spectrogram, we want to clarify the image further by applying smoothing filters to remove the speckling in the images. This section will discuss our 3 algorithms that we implemented with promising results: Gaussian Blur, Median Filter and Hybrid Median Filter.

5.2.1 Gaussian Blur

The Gaussian filter is applied so that for each pixel in the image, a weighted average is calculated such that the central pixels contribute more considerably to the result than the pixels on the edge [8]. The Gaussian filter (2D smoothing operator) $G(x, y)$ is given below:

- $G(x, y) = e^{-(x^2+y^2/2\sigma^2)}$ where (x, y) are the image co-ordinates and sigma (standard deviation) is the only parameter of the filter [9].

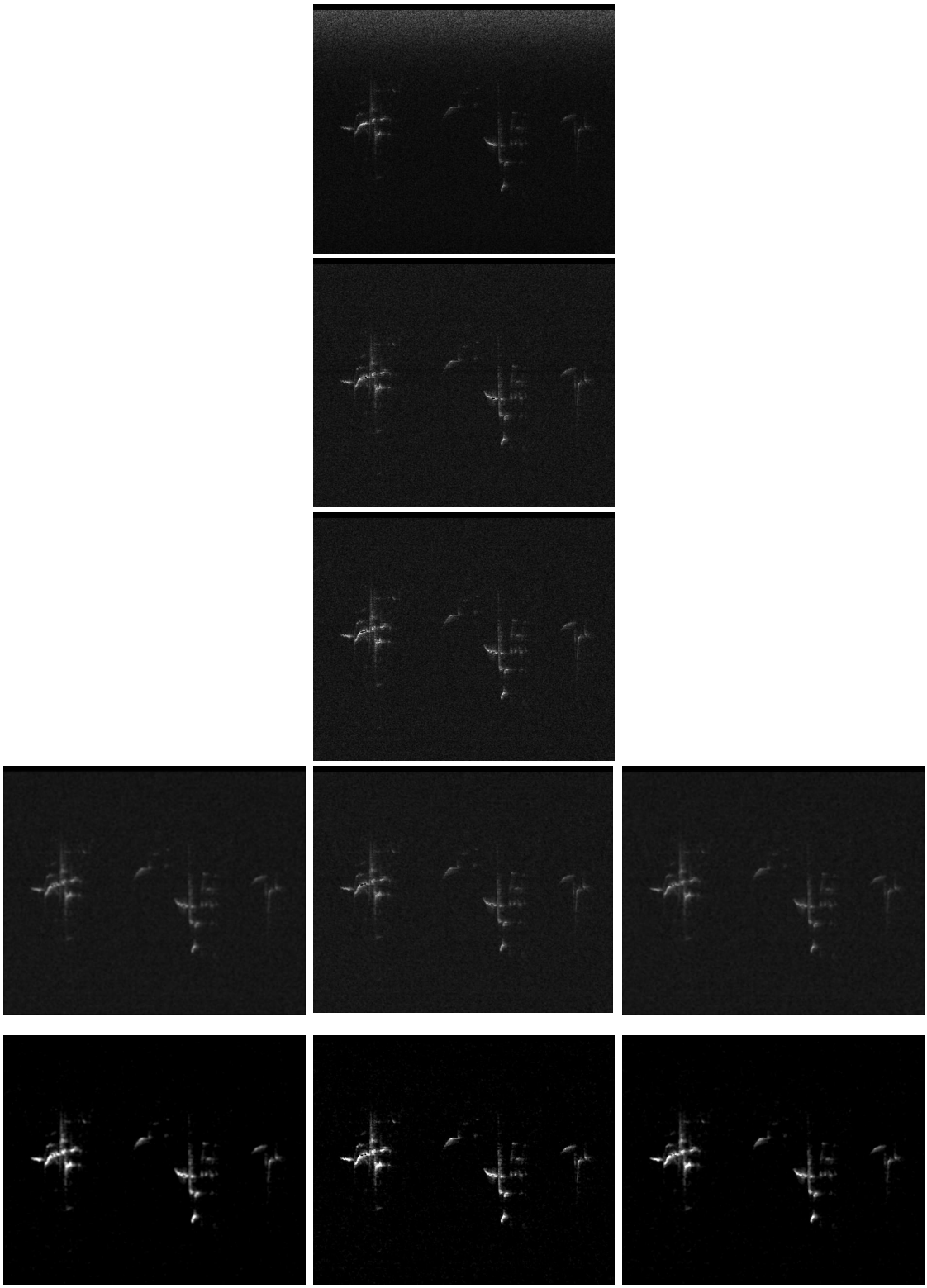

Figure 6. The top image represent a raw image taken from the forest. The next two images are applications of the attenuated Weiner Filter. From there we applied 3 different smoothing techniques along with a brightness and contrast algorithm (listed from left to right) Gaussian Blur, Median Filter, and Hybrid Median Filter.

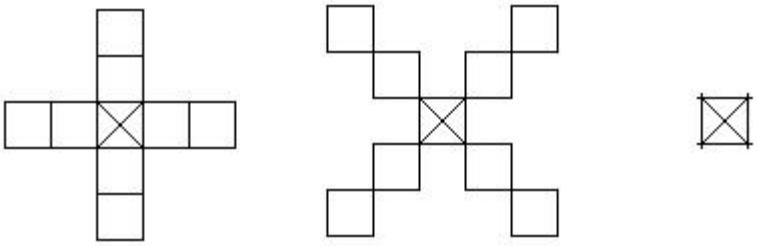

Figure 7. The left image represents the median of the horizontal/vertical pixels, the center image represents the median of the cross pixels, and the right pixel represents the center pixel. The Hybrid Median filter takes the median of these 3 and sets the value of the center pixel to it.

These coefficients of the size k by k filter mask have only the one parameter, sigma, which should be chosen as (2w + 1) / 2 where w is the scale of the diameter of the blur. Then the size k should be chosen such that k > 2w + 1 [8]. In our algorithm we estimated w to be 2 so then we chose k = 7 (which is greater than 5) and sigma = 3 (rounded up from 5/2).

5.2.2 Median Filter

The algorithm to smooth an image using the Median filter is as follows:

- For each pixel, replace the pixel with the median of the k by k neighborhood of pixels.

While conserving the sharp edges of the image, the median filter is used to reduce speckle noise. In the filter there is only one parameter k, which is the filter length. Ideally, k should be chosen with at least k = 2w + 1, where w is the noise feature representing the estimated width of the speckles (in pixels) [8]. In our algorithm we used k = 5, since the speckles averaged a width of 2 pixels. The main disadvantage of the median filter is that by taking the rectangular neighborhood of the pixels, this can damage thin lines and sharp corners of the image. One way to solve this is to use the Hybrid Median Filter described in the next section.

5.2.3 Hybrid Median Filter

Like the Median filter, the hybrid takes the median of pixels and sets the center value to that median. However, the Hybrid Median filter takes the median of only 3 values as can be seen by the algorithm below:

- Replace the center pixel of a k by k neighborhood with the median of: (1) the median of the neighboring center vertical and center horizontal (2) the median of the neighboring cross pixels, and (3) the center pixel.

As in the Median filter, the parameter k represents the filter length and was chosen as 5 in our algorithm for the same reason as above. The Figure 7 represents the 3 pixels of which the median will replace the center pixel. The first image represents the median of the 9 pixels in the horizontal/vertical lines, the second represents the median of the 9 cross pixels, and the last represents the center pixel [7].

5.3 Brightness and Contrast Filters

Lastly, using a grey-scale transformation, the brightness and contrast will be adjusted in order to clarify the image further. Transformations are used typically when humans are analyzing images, because it is easier to interpret if the contrast is enhanced [9]. Given the normalized value of a pixel which is between 0 and 1, we apply the algorithm below:

- For each pixel, take the value and multiply it by the brightness. Subtract a faction from that and multiply it by the contrast and add a fraction. Finally, apply a clamp that sets all values above one to one (white), and negative values to 0 (black).

Using this after the previous smoothing filters reduces the noise between signals and also enhances the clarity of the signals themselves.

5.4 Algorithm Results

Figure 6 displays the full noise reduction algorithm sequence applied to an image from the research forest. As can be seen each of the three smoothing algorithms,

in addition to adjusting the brightness and contrast, effectually eliminates the additional noise after the attenuated Weiner Filter is applied.

6 Summary

This paper discussed two important implementations that will contribute to the overall project of using machine learning techniques to automate bird species recognition. Having designed a program to assist in manually labeling data efficiently, we will now be able to study and use that information as a training set for machine learning algorithms. The next step will be to test this program by comparing results of species found during the manual point counts done by ecologists in the field to the count of species found with the labeling program collected from audio at the same times and locations as the point counts. Additionally, sets of labeled data found with this program will act as the expected results when testing the accuracy of machine learning algorithms.

By reducing background noise for the raw collected data through applying various signal processing algorithms, we have images with more defined signals. This not only contributes to the accuracy of manually labeling data, but will also create a clearer distinction between the noise and the signal for more precise automation in future algorithms.

7 Acknowledgements

This work was supported by DREU the Distributed Research Experience for Undergraduates and Oregon State University where this research was conducted. The author is grateful to Dr. Xiaoli Fern, Dr. Raviv Raich, and Ph.D. Student Forrest Briggs for their insights and contributions to this work.

References

[1] T. S. Brandes, P. Naskrecki, and H. K. Figueroa. Using image processing to detect and classify narrow-band cricket and frog calls. *Journal of the Acoustical Society of America*, 120(5):2950–2957, 2006.

[2] C. Catchpole and P. Slater. *Bird song : biological themes and variations*. Cambridge University Press, 1995.

[3] J. R. Clemmons, R. Buchholz, and A. B. Society. *Behavioral approaches to conservation in the wild / edited by Janine R. Clemmons and Richard Buchholz*. Cambridge University Press, Cambridge ; New York :, 1997.

[4] S. Fagerlund. Bird species recognition using support vector machines. *EURASIP J. Appl. Signal Process.*, 2007(1):64–64, 2007.

[5] A. Harma. Automatic identification of bird species based on sinusoidal modeling of syllables. *International Conference on Acoustic Speech Signal Processing*, 5:545–548, 2003.

[6] 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. *The Journal of the Acoustical Society of America*, 103(4):2185–2196, 1998.

[7] C. Reiter. With j: image processing 1: smoothing filters. *SIGAPL APL Quote Quad*, 34(2):9–15, 2004.

[8] M. Seul, L. O'Gorman, and M. J. Sammon. *Practacal algorithms for image analysis description, examples, and code*. Cambridge University Press, 2000.

[9] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis, and machine vision*. Brooks/Cole Publishing Company, 1999.

[10] S. V. Vaseghi. *Advanced signal processing and digital noise reduction*. John Wiley & Sons and B. G. Teubner Publishers, 1996.

[11] K. Wolf. *Sound Annotator User Guide*, November 2009. Can be found at: `http://www.cra.org/Activities/craw_archive/dmp/awards/2009/wolf/projects/user_manual/annontator_user_manual.pdf`.