Birdzam: App for Acoustic Bird Species Recognition

Final Project Report, COMS E6998

Talha Ansari (tja2117), Columbia University - Spring 2014

I. Introduction

The advent of big data and machine learning has brought on new world order. We are pushing the limits of computing, to solve problems which were once deemed unsolvable and, in some cases, only solvable by humans. One such problem is the identification of birds using their songs. This is a problem which humans have been solving, on a limited scale, without the aide of computers for years. There exists ornithologists who can recognize birds from their sounds. However, these ornithologists have to go through years of training and experience, a requirement often unavailable to many. It, therefore, makes sense that one of our challenges that we outsource to computers is of acoustic bird specie recognition.



There have been many attempts, both academic and practical, to recognize birds using their sounds. Several academic papers have explored the performance of various data mining techniques and features. Vilches et. al look at three common birds species in Mexico, segment the bird sounds into calls and pulses, and use of the decision trees ID3 and J4.8 and Naive Bayes to report a cross-validation accuracy above 90% [1]. However, their study is limited in scope and results affected by a skewed dataset. Recently, many studies explore Hidden Markov Models (HMM) in finding otherwise elusive patterns in bird songs [2] [3]. Support Vector Machines (SVM) is another increasingly popular classifier, with its accuracy sometimes beating the best among other classifiers. With regards to feature extraction, Mel-frequency Cepstral Coefficients (MFCC) and Linear Predictive Coding (LPC) are two most common set of features widely used, and debated on. Trifa and Vlad find that MFCC features outperform LPC features considerably [4]. Fagurland uses MFCC features with SVM to report cross-validation accuracies crossing 90% [5]. Furthermore, research also claims that convolutional deep belief networks (CDBN) and restricted Boltzmann machines (RBM) can

improve the results of MFCC by about 5-10% [6][8]. Convolutional DBN is particularly interesting, as it is time-invariant, and does not require a fixed number of frames [7].

On the consumer side, there have been attempts to build apps to solve problem. Berres and his team at University of Madison-Wisconsin have built WeBird [9], a smartphone app which aims to tell a bird using its sound. However, the app is not available on any popular app market places. Another app [10], available on Apple's App Store, boasts an accuracy at rank k=3 of 85%. Again, the functionality of the limited to the birds of British Isles and only auto recognizes 36 species. Clearly, advancements in academic research in solving the problem of acoustic bird specie recognition have not translated into consumer products.

This project, which has been carried out in collaboration with Jingwei Zhang, has been inspired by the problem presented by Prof. Peter Belhumeur and Thomas Berg of Columbia University. The aim was to build a consumer focused mobile app which can effectively aid people in recognizing birds from their sounds and can eventually be an important part of not only bird watching expeditions but also everyday life. We feel that there is enough tools and technology available to give this problem a serious attempt. Given that not a lot of consumer focused work has been done in this regard, our expectations of the project were limited from the start. It is worth acknowledging that our app has also been inspired by the popular music identification app, Shazam. Hence, so is our name Birdzam.

The rest of the report is organized in seven sections. Section II to V describe our work in developing the classifier, including data acquisition, pre-processing, feature extraction, the classifier and its results. Section VI explains the Birdzam app, explaining in detail the structure of the app, the functionality, and the user-interface. Section VII describes the logistics of the project, focusing on the division of work between team members. Finally, we present our conclusions and plans of our future work in Section VIII.

II. Data Acquisition & Pre-processing

There are a few popular bird sounds databases which support majority of the bird sound research. Data for our project came from one of those resources, the Xeno-canto library. Our dataset included recordings of 542 north American bird species, with a total of 22,107

sound files. Each sound recording was from a few seconds to a few minutes long. The distribution of sound recordings was uneven. *House Wren* had the maximum number of recordings of 445, while twenty one species had less than 5 recordings. The quality of the recordings also varied a lot across and within species. Table 1 shows some important statistical metrics regarding the data used in our project.

TABLE 1. Statistics of the number of sound recordings used in our project	
Total number of sound recordings	22,107
Average number of recordings per specie	41
Mode of number of recordings per specie	8
Median of number of recordings per specie	31
Maximum number of recordings of a specie	445

Prior to extracting features, the sound recordings were pre-processed. Each sound was resampled to a sampling rate of 30 KHz, and then split into segments. Segmentation was carried by converting the sound into a power spectrum, which was then converted to a 2D image type structure, and the image was used to find contrasting regions of different power distributions. An example of a sound recording with detected segments in shown in Figure 1. As can be seen, segmentation not only allowed separation of different sources of sounds in the recording, but also helped exclude regions of no or unimportant sounds.

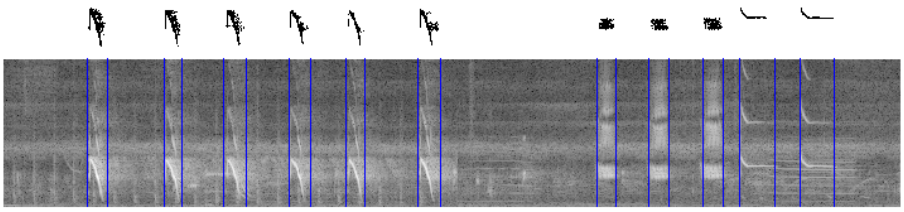


Figure 1. Spectrogram of a sample recording showing the detected segments.

The number of segments of each sound file depended on the composition of the sound and, therefore, varied. The distribution of sound different metrics of sound segment is shown in Figure 2. The average number of segments was around 18 per file, 800 per specie, and the average length of segments was about 1 second.

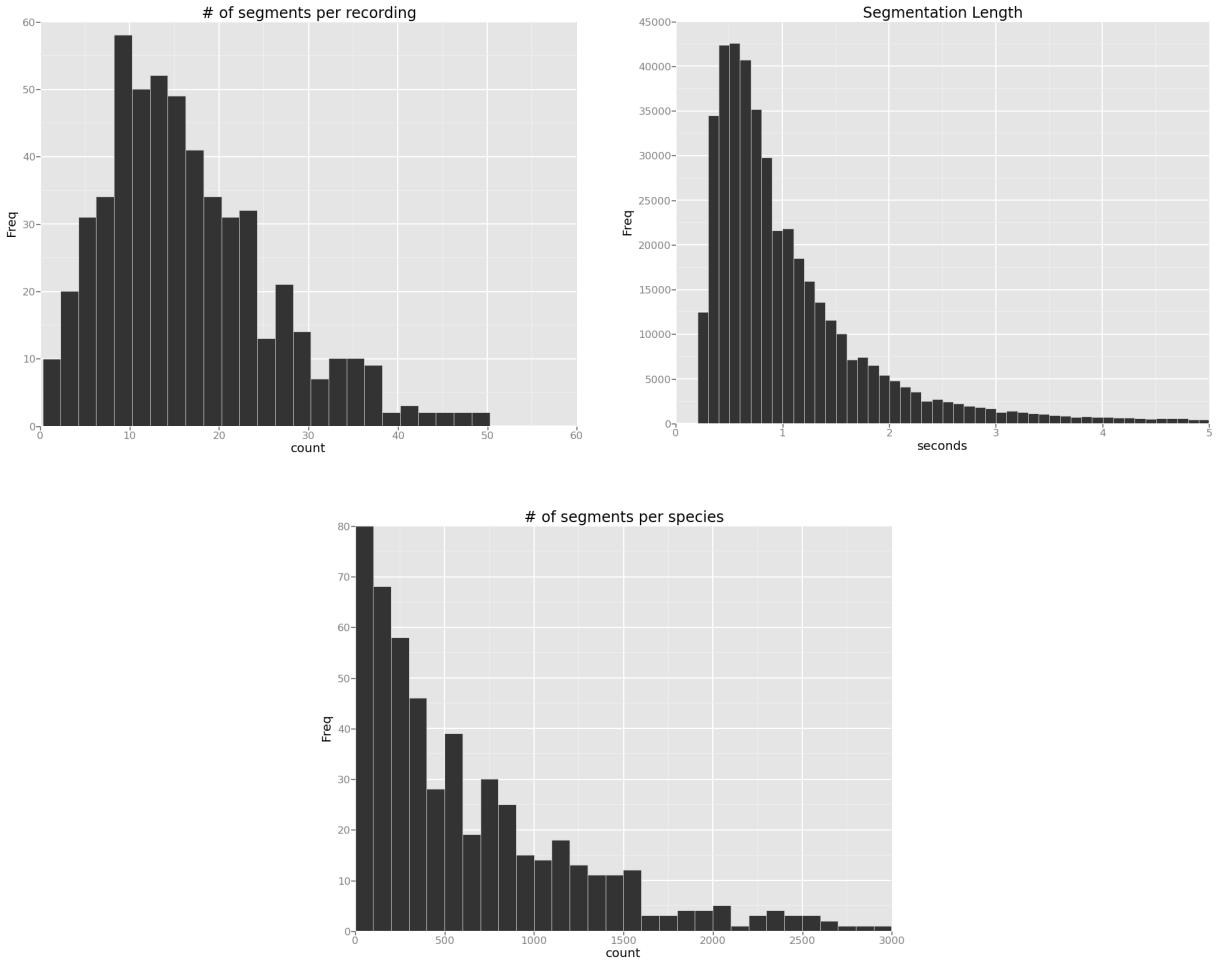


Figure 2. Histograms showing the distribution of three metrics of the sound segments

III. Feature Extraction

Mel-frequency coefficients (MFCC) are widely used in the domain of sound recognition, and have shown compelling results. The MFC coefficients of each segment were extracted using the Rastamat library provided by LabROSA at Columbia University [11]. The order of the MFCC (number of cepstra) was kept at thirteen, a value usually used for speaker recognition, while forty warped spectral bands were used during MFCC calculation. The band edges of MFCC were also limited between 150 Hz and 11,250 Hz.

The final feature set consisted of multiple summary statistics (e.g. mean, variance, max, min, first and second difference) of MFCC, center of mass of auditory spectrum and MFCC, maximum position of auditory spectrum, and normalized 4-bin histogram of auditory spectrum, resulting in a feature vector of length 69. Figure 3 shows a 2D projection of MFCC features of five randomly selected species us t-Distributed Stochastic-Neighbor Embedding (tSNE). As can be seen, species are well separated even in the 2D subspace.

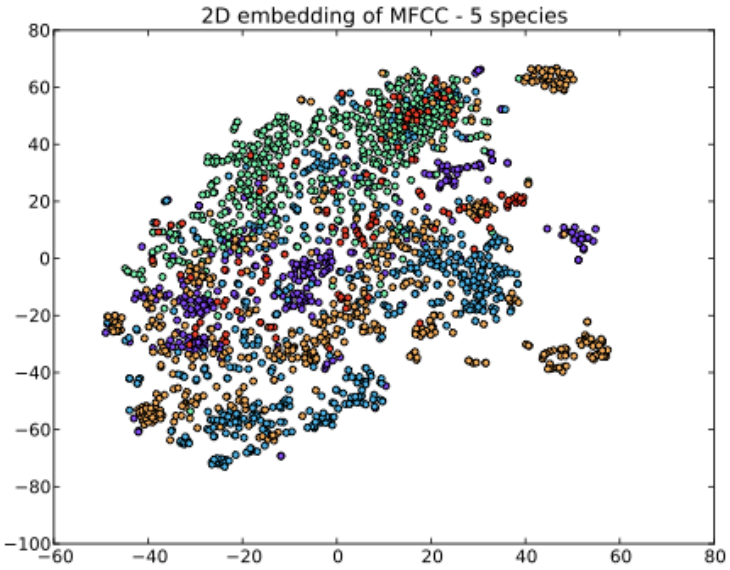


Figure 3. 2-D projection of the MFCC of five randomly selected bird species

IV. Classifier

We tested several classifiers, including Extra-Trees, Random-Forest, K-Neighbor, Logistic Regression, deep learning, and SVM. However, due to time constraints, we decided to explore CDBN and SVMs in more detail. As it turned out, using deep learning required massive amounts of computing resources. Training a five-specie classifier took several hours to train. Consequently, we halted our experiments with CDBNs and decided to proceed with SVMs for the rest of our project. Our final classifier is an SVM with a radial (rbf) kernel and a one-vs-one scheme. SVM model was generated using the popular *sklearn* library for Python [12].

V. Results

Performance of the classifier is represented in the Figure 4. The graph shows the accuracy at rank *k* plot, where the accuracy is the number of times the correct species was included in the *k* most likely species in a prediction by the SVM classifier. As can be seen, the accuracy of our final classifier, the one with 542 species, manages to cross the 70%.

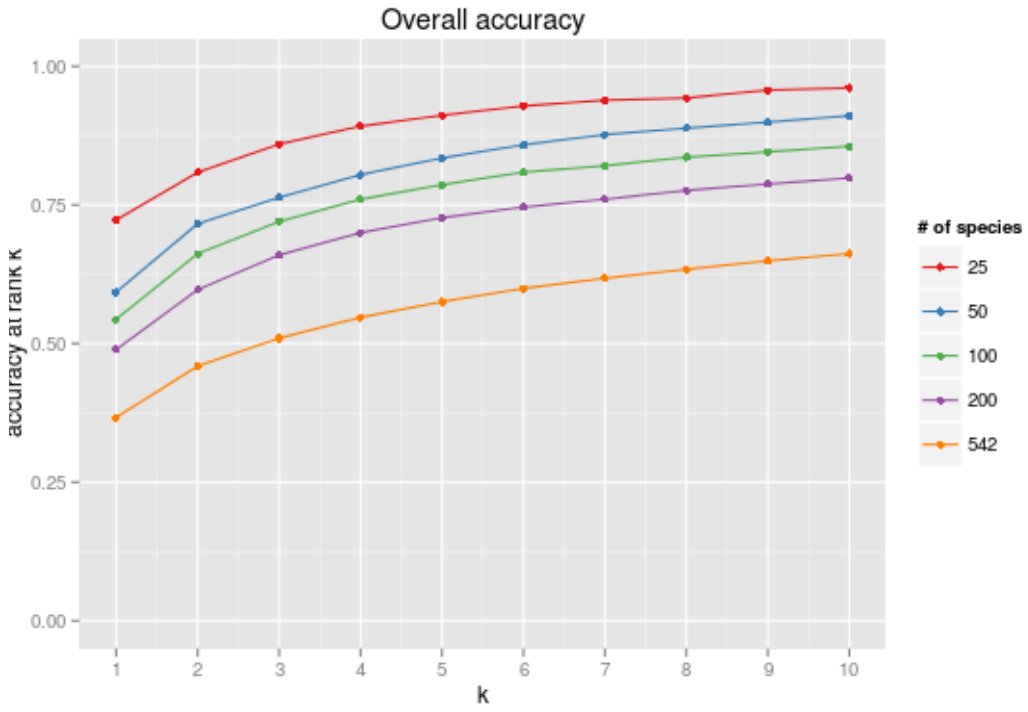


Figure 4. Accuracy at rank *k* plot of the SVM classifier used in our project

VI. Birdzam - The App

Background

Choosing between iOS and Android, fortunately, was an easy decision. Apple has an advantage over Android when it comes to quality of apps. Given that our project was focused on North American birds, Apple's edge over Android in terms of it's market share is even more in this region. Therefore, we decided to go ahead with iOS, focusing on the more portable iPhone and not the iPad. Screenshots from the latest version of the app are shown in Appendix A.

Functionality

The app development process required eight main stages. Using the app functionality stories, they can be represented as: 1) App should have a client-side which can record sound using iPhone's native hardware, 2) App should have a client-side which can send recorded sounds to a remote server, 3) App should have a remote server which can receive a sound recording from the client-side, 4) App should have a remote server which can pre-process a sound recording, extract features, and produce specie predictions using a pre-built classifier, 5) App should have a remote server which can send predictions to the client-side, 6) App should have client-side which can display the predictions sent by the remote server, and 7) App should be able to display details about each of the predicted species, 8) App should have a client-side which looks user-friendly and is, at least, not a turn-off for users. As can been seen in the app functionality stories above, the app has two main structural parts, the iPhone client-side and the remote server. Both parts come with their own challenges and require different skill sets.

The Client and the Server

The client-side of app has been built on Apple's dedicated app development framework called Xcode (version 5). Apple's AV Framework has been used to record and play sound files. The recorded sound files are stored in a *wav* format and then sent to the server using Objective-C's NSURLConnection with method POST.

The server, on the other hand, has been built on Python's Flask framework. Flask is a micro framework that allows lot of development flexibility while at the same time makes development easy by providing external modules and standardizations. The server listens to POST requests from the client, receives the sound files and saves them locally. The feature extraction stage at the server has two parts. First, the Flask server calls on Matlab functions to segment the sound file and then generate its MFC coefficients. The MFC coefficients are saved in a *.mat* file. Second, the Flask server calls on external Python functions to read the *.mat* file, generate the required MFCC features, evaluate the features through the pre-built SVM classifier, and generate the prediction values. Once the server has the prediction values available, it looks for the ten species with most likelihood, and sends them to the client-side. The client-side then displays the returned species in order of their likelihood. Clicking on any

of the predicted species takes the user to the third view, showing the enlarged image of the bird.

User-interface and User Interaction

Building the functionality of the app was important and challenging, but equally challenging was to design the user-interface of the client-side. With no prior experience in either iOS or Objective-C, I had to go through a steep learning curve. It was decided that an app with a maximum of three views was ideal, as it would limit the number of steps users would need to use the app but will allow enough organizational space to carry out the required functionality and display the required information. The structure of the app is nicely represented in the screenshot of the Xcode's storyboard mode, shown in Figure 5.

There are three views controllers in Birdzam. The first (main) view provides the functionality to record and send the recording to the server. the second view displays the 10 most likely species, and the third view has the ability to display more information and pictures about each of the ten species. The UI of the main view was inspired from Shazam's main view, which includes a single button in the centre which carries out most of the user-interaction. The background image of the main view of the app, the launch image and logo were designed with an intention to keep the front-end simple and free of excess colors. As a result, it provides users with a pristine experience.

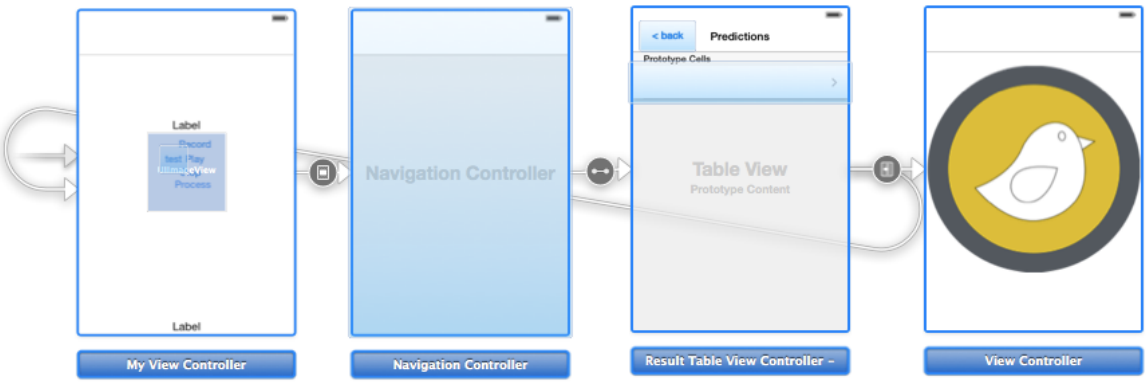


Figure 5. Structure of the Birdzam app, as represented by Xcode's storyboard mode

VII. Logistics

The project was completed in collaboration with Jingwei Zhang. The project work was divided differently during the different parts of the semester. It needs to be noted that the success of our project depended entirely on the performance of the machine learning algorithms that we would have ended us up sing. Therefore, prior to the spring break, both Jingwei and I were trying out different classifiers (using the MFCC coefficients). While Jingwei dedicated his time into exploring deep learning, I was more focused on decision trees, KNN and SVM. After spring break, as the class started finalizing its projects, we compartmentalized our work in a stricter manner. Given Jingwei's stronger background in machine learning, we decided to let him work on building the classifier. I, on the other hand, took responsibility of the iOS Birdzam app - an area I had no prior experience in.

Two things need to be noted. First, even while I had taken the responsibility of the app side of project, I was continuously working on testing the feature extraction code. Second, many of the features of the app were highly dependent on a having a classifier, a testing version of which was finished very late in the semester.

VIII. Future work and Conclusions

The Birdzam project was constrained by limited time and computing resources available to us. This is especially true for the app side, because I had only about two weeks to work on the app beyond its classification functionality. As such, there were many functions which we wanted to incorporate in the app but couldn't to so. However, these ideas are still in line and should be worked on in the future.

The Birdzam app can be improved in several ways. First, it will be nice to have a continuous upload stream functionality in Birdzam. Currently, the app records a sound using the user designated start and stop points, and then sends the recorded sound file to the server. This is not ideal because there is no way to figure out if the recorded sound has enough information for good classification. Having an upload stream will let Birdzam continuously send audio data to the server until the server makes confident predictions.

Second, Birdzam currently displays pictures of the predicted species. However, in most of the situations in which the app is useful, the user does not have a clear image of the bird. Therefore, we should provide users with a sample sound of each of the predicted species, in order for the user to make a comparison. Third, the detail view of the predicted species can be massively improved by providing more data regarding the birds. Apart from the image and the sample sound, it will also be nice to include a Wikipedia description about the bird, as is the case in the Bird Snap app. Finally, an ability for the app to keep a track of the geo-locations of the predictions will be crucial in building a bird map, which can eventually give insights into bird migrations.

We started the project with uncertain expectations. We did not know how well the classifiers would perform on the problem and, therefore, were always excited to try new ideas. This uncertainty turned out to be a major motivation behind our continued work on the project. In the end, Birdzam may not be ready to be launched in the App Store, but it certainly has validated the concept that such a consumer focused tool can be created. If the improvements suggested above are implemented, and if the classifier is made more robust, there is no reason why Birdzam can not tap into the enthusiasm of bird-lovers throughout the country.

IX. References

[1] Vilches, Erika, et al. "Data mining applied to acoustic bird species recognition."*Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*. Vol. 3. IEEE, 2006.

[2] Kogan, Joseph A., and Daniel 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 (1998): 2185-2196.

[3] Somervuo, Panu, Aki Harma, and Seppo Fagerlund. "Parametric representations of bird sounds for automatic species recognition." *Audio, Speech, and Language Processing, IEEE Transactions on* 14.6 (2006): 2252-2263.

[4] Trifa, Vlad M., et al. "Automated species recognition of antbirds in a Mexican rainforest using hidden Markov models." *The Journal of the Acoustical Society of America* 123.4 (2008): 2424-2431.

[5] Fagerlund, Seppo. "Bird Species Recognition Using Support Vector Machines."*EURASIP Journal on Advances in Signal Processing* 2007.1 (2007): 038637.

[6] Hamel, Philippe and Eck, Douglas. “Learning features from music audio with deep belief networks.” *ISMIR on* 339344. Utrecht, The Nether- lands, 2010. 00047.

[7] Lee, Honglak, Grosse, Roger, Ranganath, Rajesh and Ng, Andrew. “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations.” *26th Annual International Conference on Machine Learning on* 609616. ACM, 2009. 00440

[8] Lee, Honglak, Largman, Yan, Pham, Peter and Ng., Andrew. “Unsupervised feature learning for audio classification using convolutional deep belief networks.” 00143.

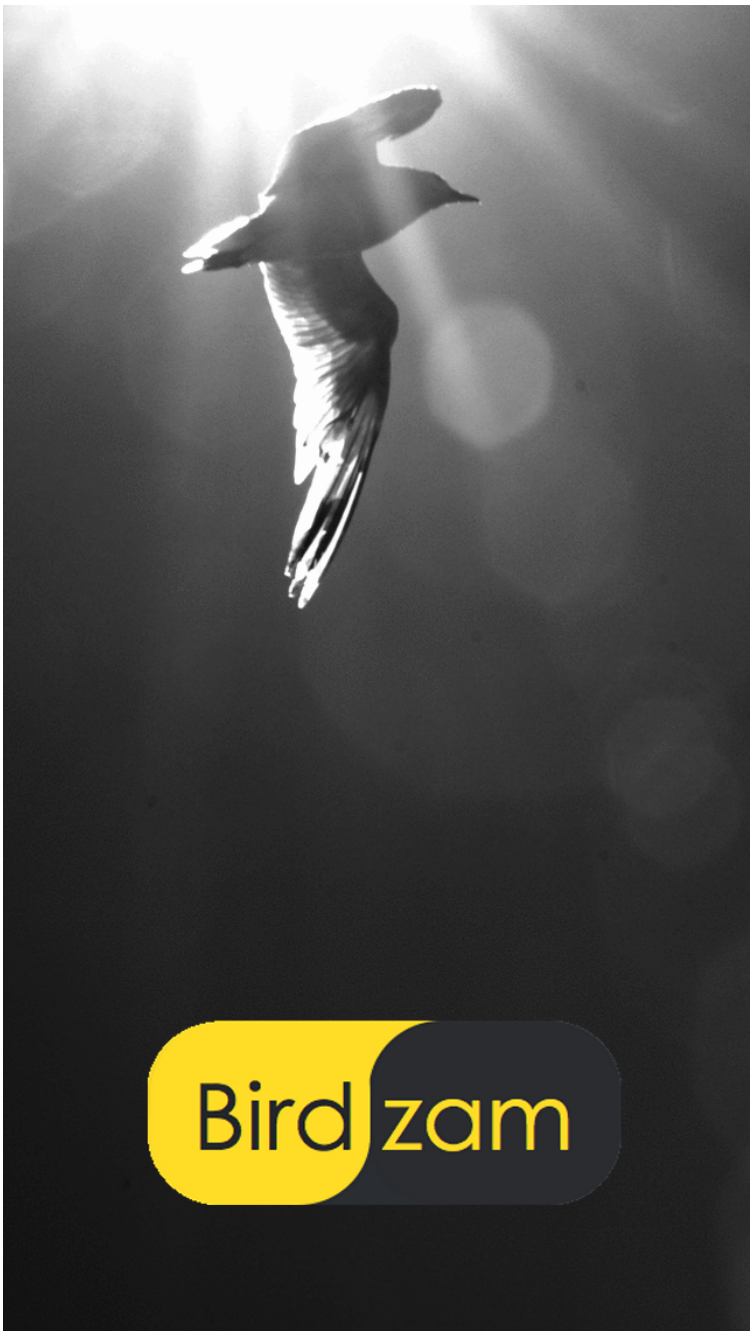
[9] Berres, Mark. “WeBird (Wisconsin Electronic Bird Identification Resource Database).” *University of Wisconsin Madison*

[10] Isoperla. “Bird Song Id Automatic Recognition & Reference - Birds of the British Isles.” Available at https://itunes.apple.com/gb/app/bird-song-id-automatic-recognition/id601362210

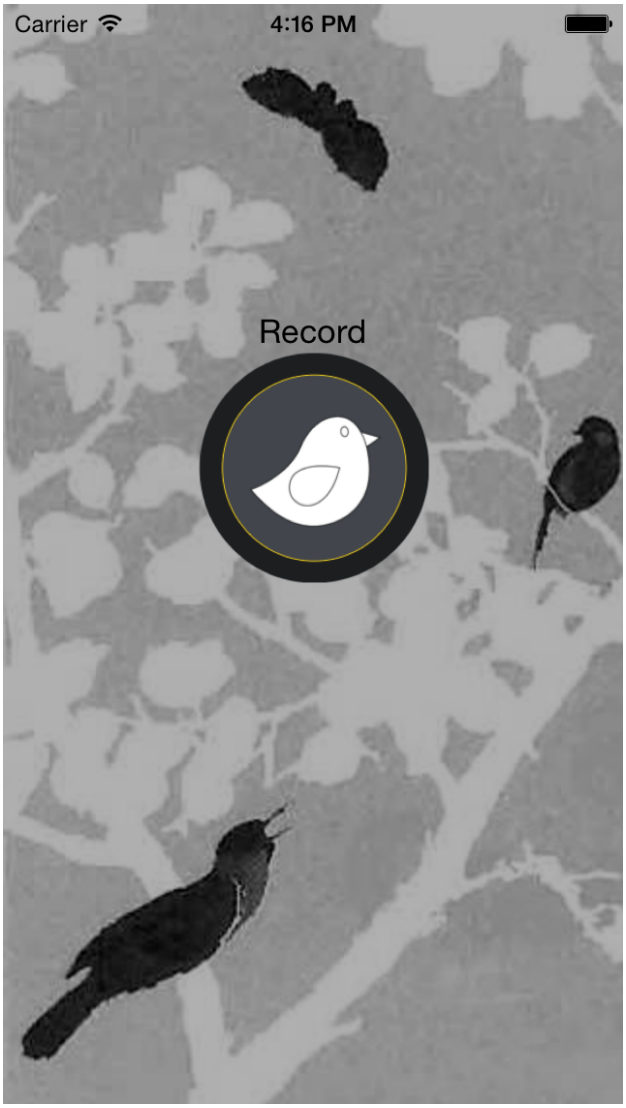
[11] LabROSA. “Laboratory for the Recognition and Organization of Speech and Audio.” *Columbia University, New York, U.S.*

[12] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

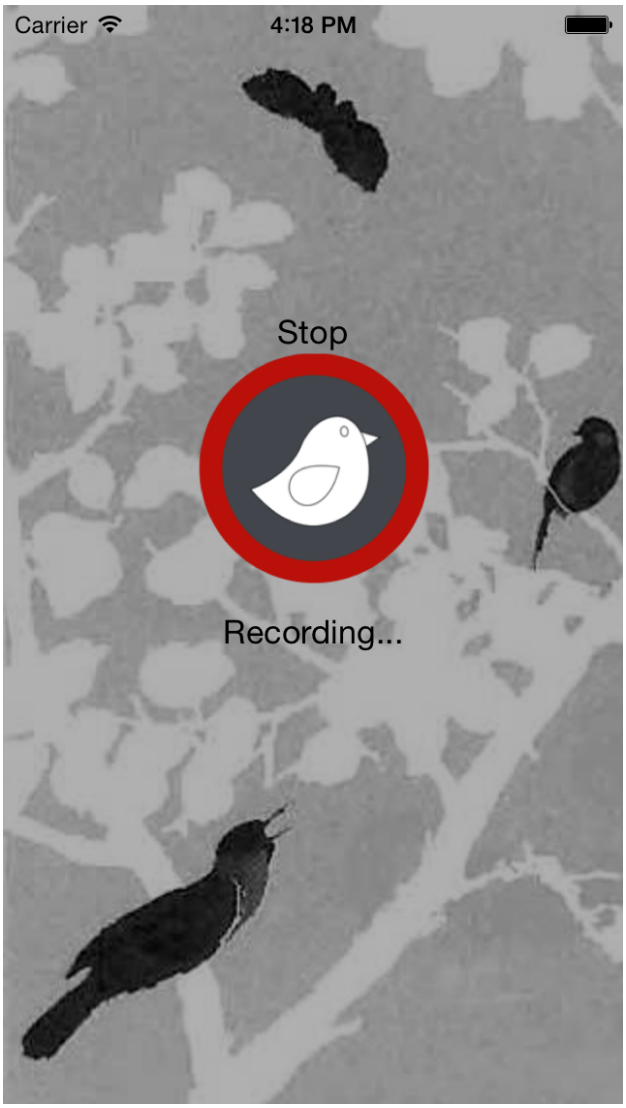
Appendix A - Birdzam App Demo



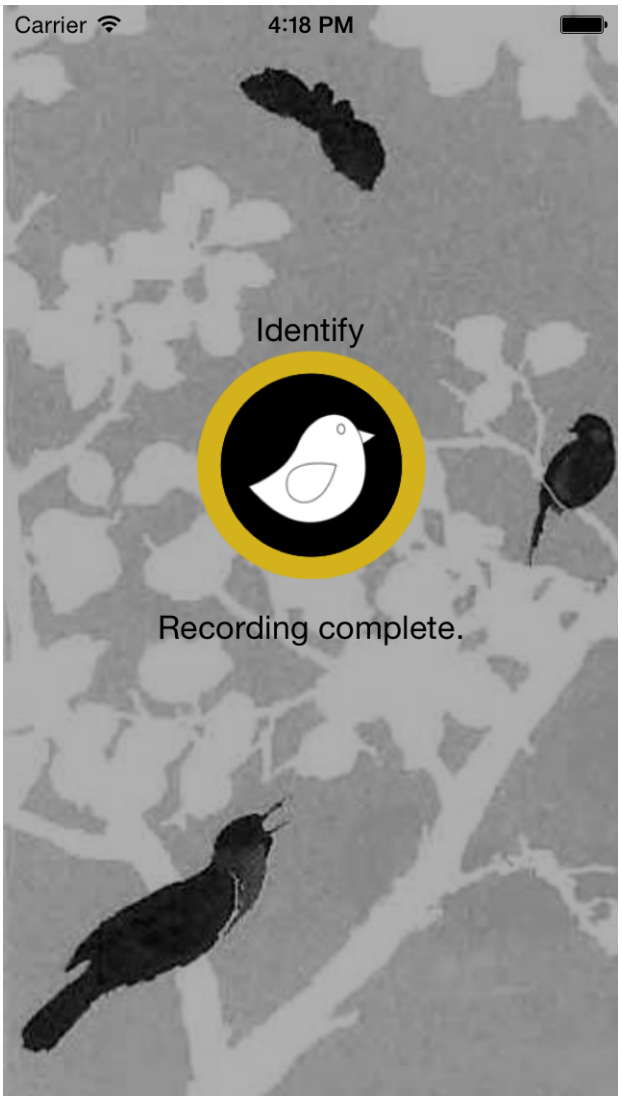
1



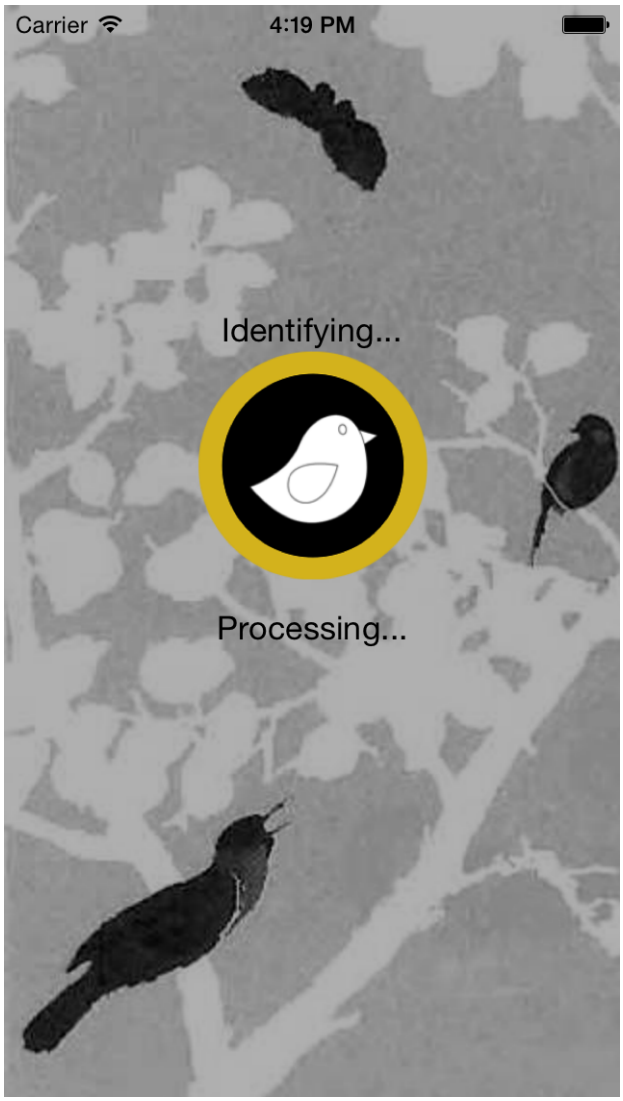
2



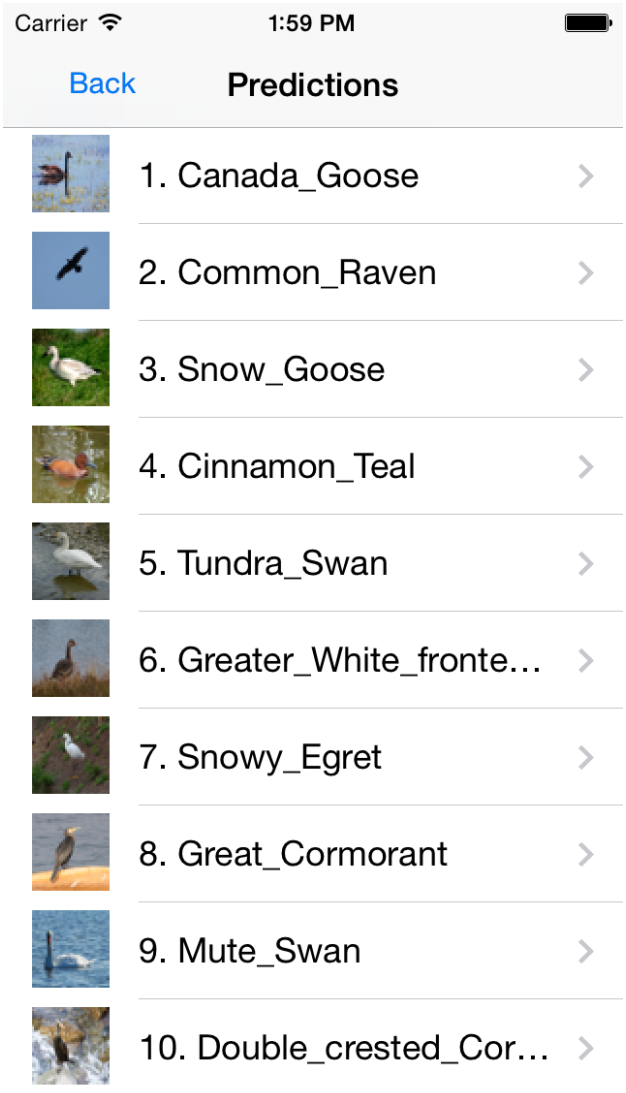
3


Carrier
4:18 PM
Identify
Recording complete.

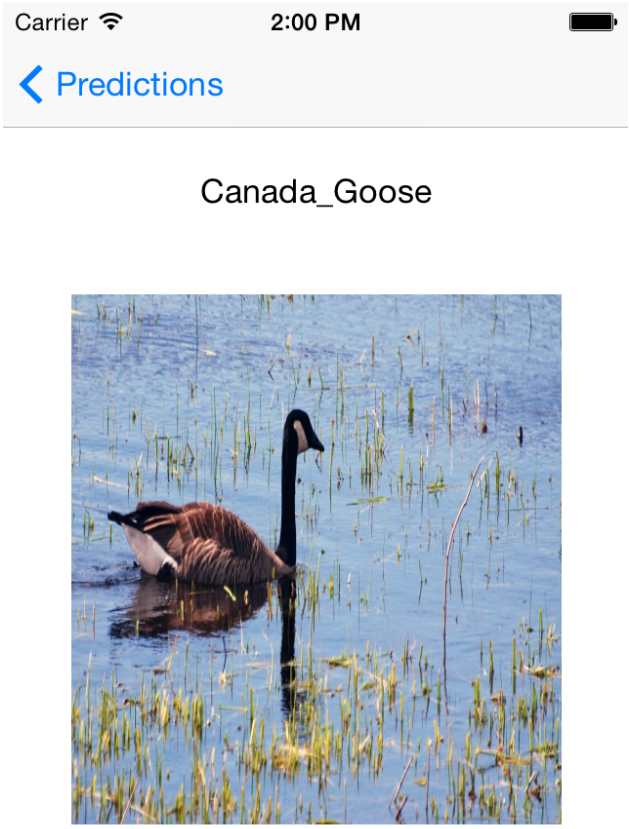
4


Carrier
4:19 PM
Identifying...
Processing...

5


Carrier
1:59 PM
Back
Predictions
1. Canada_Goose
2. Common_Raven
3. Snow_Goose
4. Cinnamon_Teal
5. Tundra_Swan
6. Greater_White_fronte…
7. Snowy_Egret
8. Great_Cormorant
9. Mute_Swan
10. Double_crested_Cor…

6


Carrier
2:00 PM
Predictions
Canada_Goose