A-BiRD

Automated Bird Recognition Device

A tool to collect continuous and objective data for ornithologists

Abstract

A-BiRD (Automated Bird Recognition Device) is a tool to collect continuous and objective data for ornithologists. Scientists all around the world are observing changes in bird populations and a decline in bird species. In their attempt to find answers, the scientific community relies on bird observations made almost entirely by ordinary citizens who volunteer their interest and time to help collect important data. Unfortunately for the professionals, common people tend to be inconsistent sources who make mistakes or misinterpret data.

The engineering solution described in this project solves these issues. **A-BiRD** consists of a microphone array that is hooked up to a Raspberry Pi. The computer is programmed to listen to ambient sounds, record bird songs, show audio curves, and send all data to BirdNET at the Cornell Lab of Ornithology for analysis in order to determine what kind of bird is present in a specific location. **A-BiRD** uses the same data from the microphone array to determine the angular direction of a bird. This allows it to distinguish between two different birds, or two birds of the same species singing at the same time in different locations.

A-BiRD works seamlessly. It reads and processes all audio files well. The correlation algorithm successfully determines the angle direction of birds. Data files are uploaded to BirdNET smoothly and with 100% accuracy. As a result, ornithologists can make valuable predictions regarding bird species, inferred nesting locations, preferred habitat, and migration patterns. **A-BiRD** is a successful tool to track the rise and fall of bird populations by species more easily. Since the system can operate 24 hours a day, seven days a week, the continuous coverage improves the accuracy and continuity of bird data. **A-BiRD** can be employed globally for objective data collection without relying on human intervention.

Research and Background Information (1)

Global bird populations are declining. Environmental changes such as agricultural intensification, pollution, habitat destruction, insecticides, urbanization, and climate change are all driving changes in bird populations. Since the 1970s, 2.9 billion birds have been lost in North America alone, or 29% of the total population. Ornithologist are working to save bird species and have a better understanding of bird populations and migrations.

BirdNET was launched by the Cornell Lab of Ornithology in Ithaca, New York, as a research platform that aims at recognizing birds by sound at scale. The system relies on ordinary citizens sending in manually collected data on bird types and their location. When BirdNET was programmed, researchers used more than 5 million samples of bird sounds as a database. BirdNET splits every submitted recording into one-second chunks and makes a species prediction for every chunk. Each prediction contains probability values as well. The species with the highest probability in all chunks is the species most likely to be present in the recording. Cornell studies and tracks over 10,000 bird species around the world. Ornithologists currently depend on hundreds of thousands of people like you and me to manually track and report what they're seeing in backyards, neighborhoods, and wild locations around the world.

A-BiRD is programmed on a Raspberry Pi using Python programming language. Python has been used successfully throughout the computer world as it is easy to learn. Scientists and engineers are often using it to solve specific complex technical problems. The extensive mount of bird song data that is recorded by the microphone array of the device has to be isolated and analyzed before it can be sent to globally operating bird surveys for research purposes.

Research and Background Information (2)

Python coding language makes the mathematics to handle large amounts of data and 2-D analysis easier. Its mathematical tools and automatic data processing capabilities simplify the analysis of signals from the device's microphone array and isolate bird song from ambient noise.

The state of Arizona is considered a world-class birding destination and, depending on migrations, is home to hundreds of species throughout the year. It is a perfect location to design, engineer, and test **A-BiRD**. The data collected by the device can be easily shared with researchers at Cornell, "Project FeederWatch", "eBird" or "World Bird Count". All of these platforms welcome as much bird data as possible for their research efforts.

In eBird's first 10 years of existence, bird watchers contributed 100 million observations. It took only 2 more years to reach the next 100 million.

A-BiRD

(**A**utomated **Bi**rd **R**ecognition **D**evice)

↓

Automatically detects and records sound waves of bird songs via microphone array and Raspberry Pi

↓

Determines the angular direction of bird song arrival

↓

Sends audio files to BirdNET for identification

↓

Data is used to increases the probability values for native and migratory bird species in its location across the seasons

Requirements, Criteria, and Constraints/Limitations

Requirements and Criteria

- A computer processor: low-power, low-cost, stable
- Two+ microphones on a fixed mount to pick up the bird songs and determine their direction
- Cables for signals and power, power source
- Wireless connection to internet to reach BirdNET
- Analog to Digital Converters
- Code to record bird songs, determine the direction, communicate with BirdNET, and to save the results
- Tolerant of weather which includes: rain, hail, snow; wind, freezing temperatures, heat; and sand or dust
- A human interface to interact with the computer

Constraints / Limitations

- Weather will not be tested in this phase due to prototype development
- The audio sample rate and quality is determined by the ability of the prototype to record and store it
- The sample rate and microphone spacing determines the angle resolution of the device
- The amount of memory constrains the sample rate and quality of the data
- Python may not be able to process the data fast enough
- System requires set up in reasonably remote locations to avoid loud ambient noises such as excessive wind, human speech, or extensive road traffic

Materials for the A-BiRD Prototype

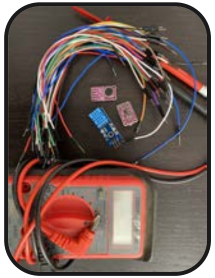
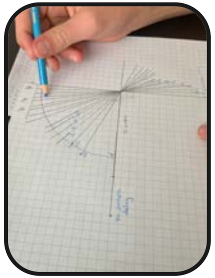
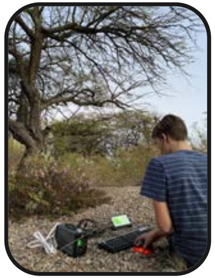
- Power Supply, Raspberry Pi, HDMI Monitor, Keyboard, Mouse
- General Purpose Input Output (GPIO) Bus Extension Ribbon
- Assorted Dupont Wire Cables, GPIO Breakout Board and Breadboard
- Waveshare WM8960 Audio Hat for Raspberry Pi [Multi-Channel Analog Audio Inputs, Analog-to-Digital Converters (ADC) and Digital-to-Analog Converter (DAC)]
- Microphone Mounting Fixture and Weather Housing (Optional)
- Material for Reducing Wind Noise (Optional)
- Portable LiON Power Supply for Field Work (Optional)

Test Materials list:

- Digital Multimeter (Voltage, Current and Resistance)
- Multi-Channel Oscilloscope with Probes, Sound Sample Files for Testing, Research Materials

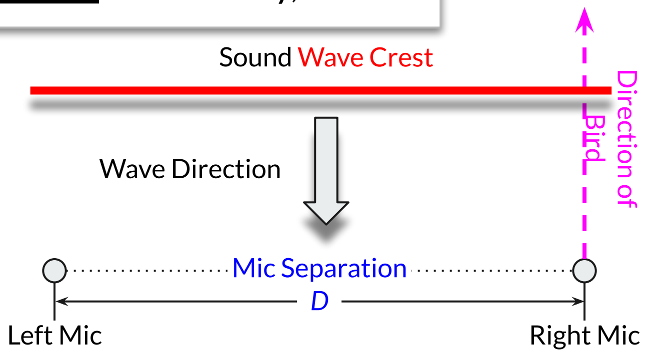




Execution: Prototype Design and Methodology – Geometry (1)

Case 1: Zero Delay, $\theta = 0°$

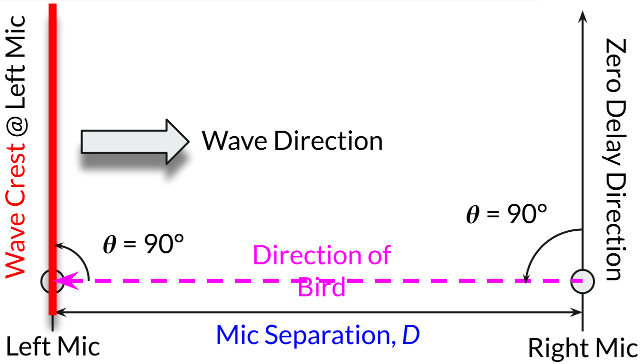


Wave Travels at Speed of Sound, $c \cong 334$ m/s

Wave Crest is Parallel to Mic Separation Line

Wave Reaches Both Mics at the Same Time

Case 2: Maximum Delay, $\theta = 90°$



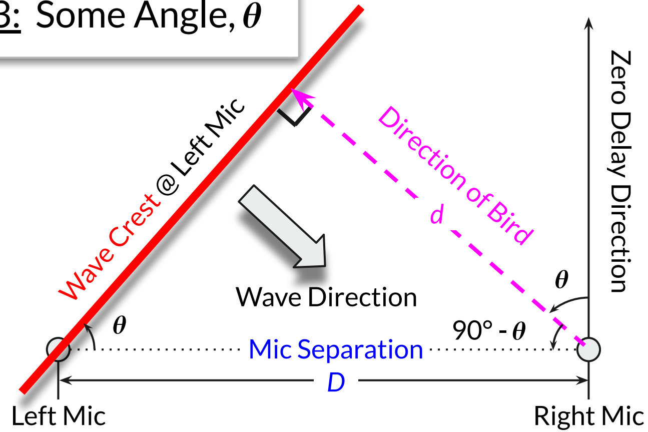
Wave Crest is at Right Angles to Mic Separation Line

There is a Maximum Time Delay for the Wave to Pass Between Both Mics

T-max = *D* / *c*

Wave will Reach Right Mic *T-max* seconds Later

Case 3: Some Angle, θ



1. Direction of Bird Always at Right Angles to Wave Crest
2. Right Triangle: Direction of Bird, Wave Crest, Mic Separation
3. Right Angle Makes Mic Separation the Hypotenuse
4. Wave Strikes Left Mic at Angle θ to Mic Separation Line
5. Let *d* be Unknown Distance from Right Mic to Wave Crest
6. Let *T-delay* be Time Delay Until Wave Crest Reaches Right Mic (Difference in Times: Right - Left)

T-delay = *d* / *c*

Sin(θ) = Direction of Bird / Mic Separation
= *d* / *D* = *T-delay* / *T-Max*

θ = ArcSin(*T-delay* / *T-Max*)

Goal:
Measure *T-delay*

Execution: Prototype Design and Methodology – Geometry (2)

Case 4: Mirror Image of Cases 2 & 3

Direction of Bird is to the Right
WaveCrest Arrives at Right Mic First
T-delay is Negative (Right minus Left)

Converting *D* and *d*, and *T-max* and *T-delay*, to Sample Intervals

Each Microphone Collects *Rs* Samples per Second

Time Between Samples: $Ts = 1 / Rs$

Maximum No. of Samples of Delay:
N-max = *T-max* / *Ts* = *D Rs* / *c*

Measured No. of Samples of Delay:
N-delay = *T-delay* / *Ts* = *d Rs* / *c*

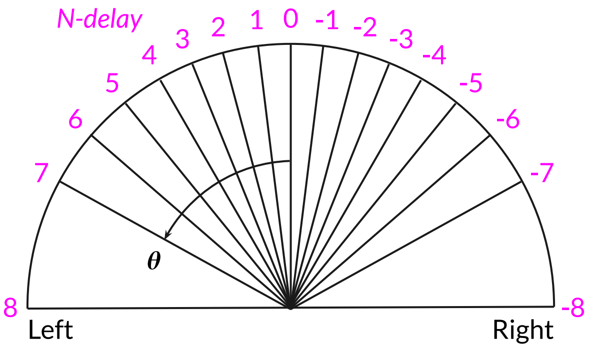
θ = ArcSin(*T-delay* / *T-Max*)
= ArcSin(*N-delay* / *N-max*), *where*
-(*N-max*) ≤ *N-delay* ≤ *N-max*

	N-delay	Sin(θ)	θ
[Far]	-8	-1.0	-90.0°
	-7	-0.875	-61.0°
	-6	-0.75	-48.6°
Right	-5	-0.625	-38.7°
Side	-4	-0.5	-30.0°
	-3	-0.375	-22.0°
[Near]	-2	-0.25	-14.5°
	-1	-0.125	-7.2°
Center	0	0.0	0.0°
	1	0.125	7.2°
[Near]	2	0.25	14.5°
	3	0.375	22.0°
Left	4	0.5	30.0°
Side	5	0.625	38.7°
	6	0.75	48.6°
	7	0.875	61.0°
[Far]	8	1.0	90.0°

A-BiRD Prototype has a Microphone Spacing of
D = 60.6 mm
CD Quality Sampling of each Mic at
Rs = 44,100 Samples per Second

Therefore,
N-max = *D Rs* / *c*
= (0.0606) (44,100) / 334 ≅ 8 samples

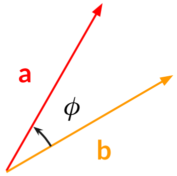
So,
-8 ≤ *N-delay* ≤ 8



Angular Direction θ as a Function of *N-delay*, the Delay in Samples of the Right Signal versus Left

Execution: Prototype Design and Methodology–Correlation

Vector Geometry: Dot Product



Vectors a and b separated by the angle ϕ

Dot product is the sum of the term-by-term products of two vectors

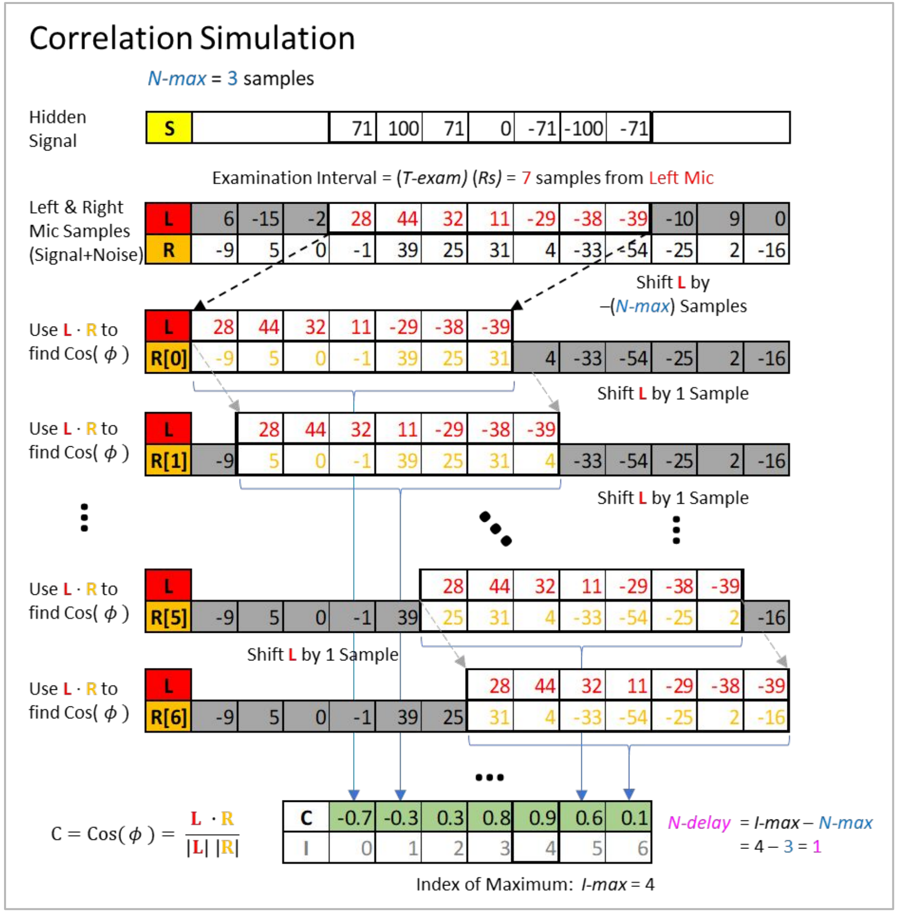
a = [$a1$, $a2$, …] and
b = [$b1$, $b2$, …]

a · b = ($a1$) ($b1$) + ($a2$) ($b2$) + …
= |a| |b| Cos(ϕ), *where*
|·| *is length of enclosed vector*

ϕ = 0° → Cos(ϕ) = 1 a and b lie on top of one another and point in the same direction	ϕ = 90° → Cos(ϕ) = 0 a and b are mis-matched and have nothing in common	ϕ = 180° → Cos(ϕ) = -1 a and b are aligned but point in opposite directions

Problem: Which birds are present is not known, nor are the songs or variations they may be singing.
Solution: While actual bird song signal isn't known, both mics recorded it.

- Samples from Left Mic used to "find" signal in samples from Right Mic
- Sample vector of *T-exam* seconds from the Left Mic is dotted with a corresponding slice from the Right Mic
- Maximum delay is -(*N-max*) to *N-max* samples:
  - Left is shifted from -(*N-max*) to *N-max*, one sample interval at a time
  - Dot product taken each time and divided by magnitudes of Left & Right to give Cos(ϕ)[*i*]
- Where Cos(ϕ)[*i*] is maximum, report the index *i*

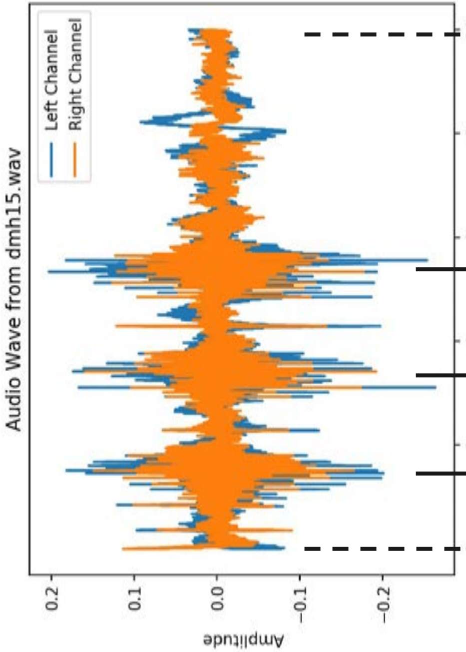


Data Collection: Audio Wave, Correlation Output and BirdNet Results

Three chirps from Costa’s Hummingbird sitting in a tree
– Other sounds were water in an artificial stream, leaves and people

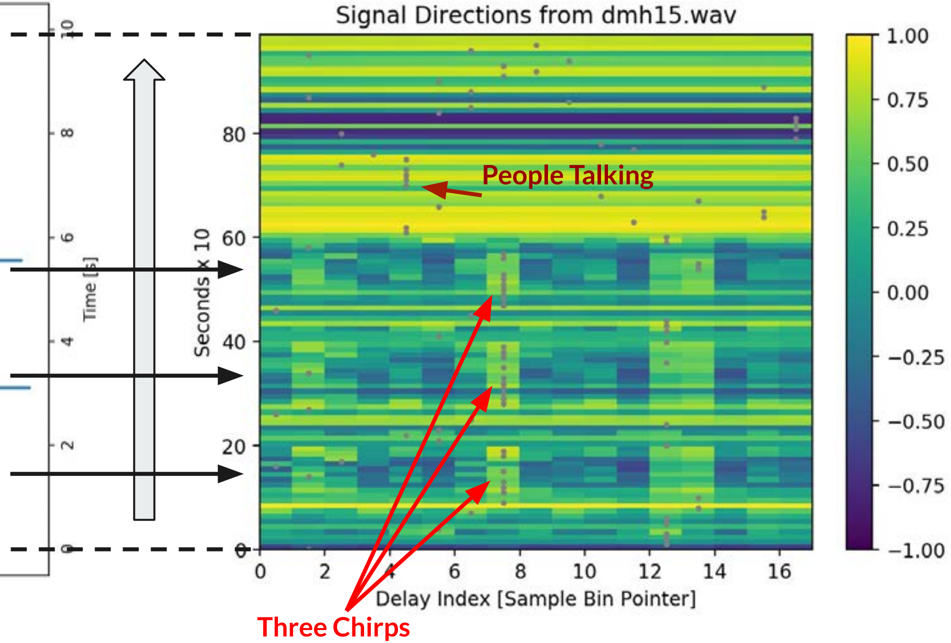
A-BiRD Parameters:
- *N-max* = 8 Samples
- *Rs* = 44,100 Samples / Second
- *T-exam* = 0.1 seconds

Recorded Audio Wave



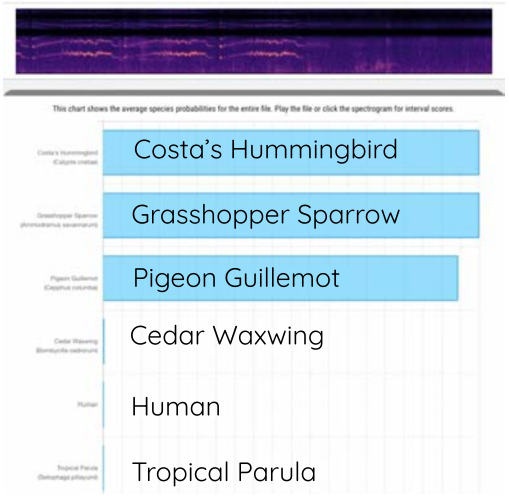
Three hummingbird chirps (Large Amplitudes), Plus People Talking and Artificial Brook in the Background

Correlation/Delay



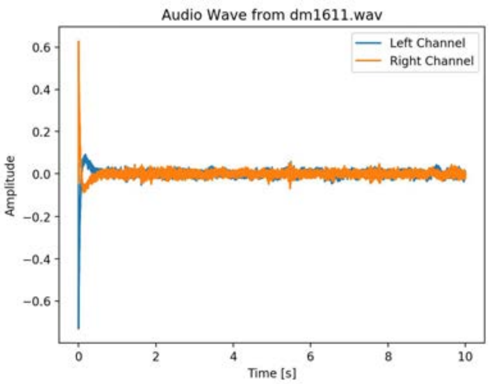
Dots show delay bin indices with peak correlation
Consistent delay output when the bird chirped

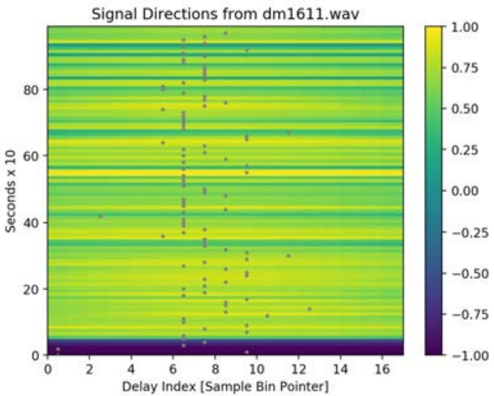
BirdNET Analysis Of Audio File

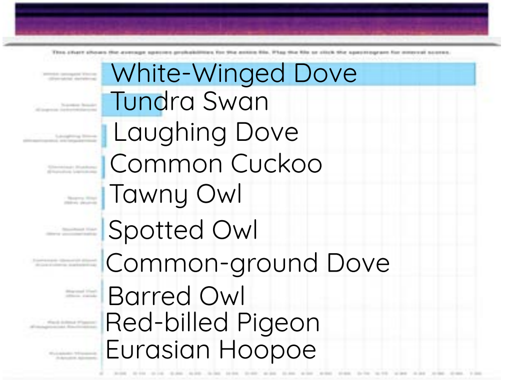


Spectrogram of audio waveform and possible bird species from most likely to least likely from BirdNET AI Engine

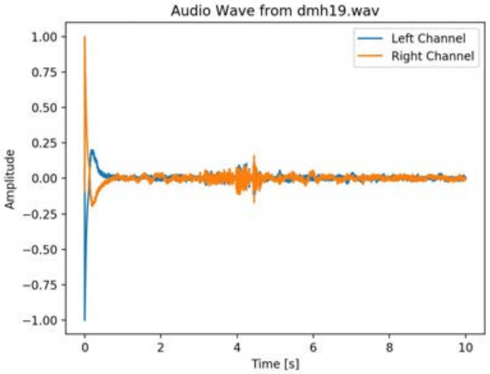
Data Collection: Audio Wave, Correlation Output and BirdNet Results

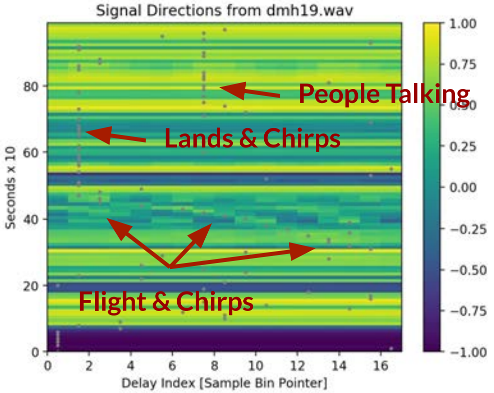






Several doves in trees cooing as they start to roost

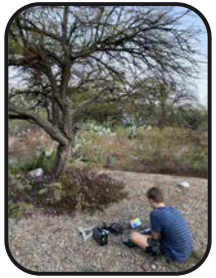




Costa's Hummingbird flying and chirping past the microphones at 3-5 seconds

Encountered Problems - Reflection - Redesign

Encountered Problems: (1) Due to the winter season, bird population in Southern Arizona was sparse. Therefore, testing had to take place in an aviary at the "Arizona-Sonora Desert Museum" in order to reliably have bird songs to record. Due to the public location there was more background noise in the recordings. (2) The code to locate a bird and figure out what kind of bird it was frequently needed to change. For example, the examination window went from 1 s to 0.2 s before settling on every 0.1s. Every time that happened the changes had to be manually debugged. (3) Python data types were an issue in the code, and mixing arrays and iterables had to be manually fixed. (4) Before testing, simulations to troubleshoot **A-BiRD** were made to anticipate functionality problems while testing in the field.



Reflection - Redesign: (1) Different microphones should be tested to optimize functionality of the system. (2) Adjusting the separations of the microphones should be considered in order to optimize the angle resolution. (3) **A-BiRD** should be optimized to withstand extreme weather conditions encountered in Southern Arizona, especially in the hot summer months with high UV. (4) The utilization of the system should be expanded. Extended field studies should be planned to track bird migrations and nesting throughout the year for continuous periods of time. (5) An Arduino could be added for audio recording and some of the data processing. The timing of the audio samples is very important in order to get accurate delay measurements. An added Arduino would free up the Raspberry Pi to analyze the recording for angle locations, handle sending the samples to BirdNET, and process the results. (6) BirdNET has code to preprocess audio files. Integration with this should be looked at.

Results

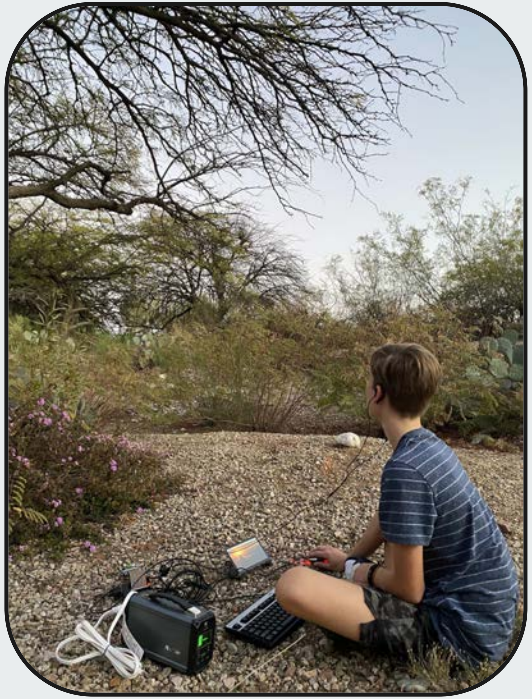
- The prototype device recorded the bird songs from both microphones at CD quality (44,100 samples per second, 16-bits per sample per microphone, left and right channels)
- The device read and processed the audio files well with only a few adjustments or manual debugging. Many different audio files were needed to get plenty of recordings with different sound environments.
- Once the code was figured out, the Raspberry Pi processed the audio files perfectly and quickly.
- The correlation algorithm worked on all of the audio files and determined the right delay from one microphone to the next. For example, it accurately found the angle location of the Costa's Hummingbird in the aviary, and it could handle several different doves cooing together.
- The audio files uploaded to BirdNET very smoothly without any problems.
- BirdNET gave back the results super quickly (10-15 sec which includes uploading the file and ~0.24 sec for the prediction). It was 100% accurate with all of the audio files recorded and determined the type of bird that sang.
- Sometimes the audio file doesn't have enough distinguishing information. In these cases the results came with suggested birds and their probabilities ranked from most likely at the top, to least likely at the bottom. Working with these probabilities is an area of future design.
- The prototype worked very well in the environment with water sounds, people and the birds.

Conclusion

A-BiRD was successfully designed and engineered as a tool to collect continuous and objective data for ornithologists.

The system works seamlessly. It reads and processes all audio files well. The correlation algorithm successfully determines the angular direction of birds. Data files are uploaded to BirdNET smoothly and with 100% accuracy. As a result, ornithologists can make valuable predictions regarding bird species, inferred nesting locations, preferred habitat, and migration patterns.

The Automated Bird Recognition Device provides consistent data to help track the rise and fall of bird populations by species. Since the system can operate 24 hours a day, seven days a week, the continuous coverage improves the accuracy and continuity of bird data. A-BiRD can be employed globally for objective data collection without relying on human intervention.



Sources

Pieplow, Nathan. “Peterson Field Guide to Bird Sounds of Western North America”, Houghton Harcourt Publishing Company, 2019

Bader, Dan. “Python Tricks: The Book”, Self-Published, 2021

Molloy, Derek. “Exploring Raspberry Pi”, John Wiley & Sons, Inc., 2016

Cornell Lab of Ornithology, 2022, https://birds.cornell.edu, September, 2021 - January, 2022

Cornell Lab of Ornithology. “Project FeederWatch”. 2022, https://feederwatch.org/about/project-overview. Accessed September 2021 - January 2022

Python Organization, “Python 3.10.2 documentation”. 2022, https://docs.python.org. Accessed November 2021 - January 2022

Koidan, Kateryna. “Arrays vs. Lists in Python:” Vertabelo SA, 2022, https://learnpython.com/blog/python-array-vs-list/. Accessed December 2021 - January 2022

NumPy Organization, “NumPy”. 2021, https://numpy.org. Accessed November 2021 - January 2022

SciPy Organization, “SciPy”. 2021, https://scipy.org. Accessed November 2021 - January 2022

MathWorks, “Matplotlib: Visualization with Python”. 2021, https://matplotlib.org. Accessed December 2021 - January 2022

WaveShare, “WaveShare”. 2022, https://www.waveshare.com. Accessed October 2021 - January 2022

Wolfram, “Wolfram Language & System”. 2022, https://reference.wolfram.com, Accessed December 2021 - January 2022

National Audubon Society, “Costa’s Hummingbird”, 2022, https://www.audubon.org/field-guide/bird/costas-hummingbird#, Accessed January, 2022