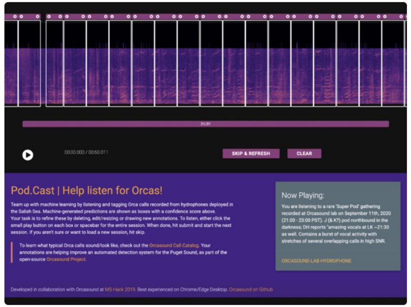
Computing on the Edge of the Sea

A hydrophone, a Jetson-Nano, and multiple ResNets

Presentation by Val Veirs & Scott Veirs (Beam Reach, SPC; WA, USA)

Meridian "Winter Webinar" on Wednesday, January 13, 2021

Artificial Intelligence for orcas



Pod.Cast | Help listen for Orcas!

Team up with machine learning by listening and tagging Orca calls recorded from hydrophones deployed in the Salish Sea. Machine-generated predictions are shown as boxes with a confidence score above. Your task is to refine these by deleting, edit/resizing or drawing new annotations. To listen, either click the small play button on each box or spacebar for the entire session. When done, hit submit and start the next session. If you aren't sure or want to load a new session, hit skip.

> To learn what typical Orca calls sound/look like, check out the Orcasound Call Catalog. Your annotations are helping improve an automated detection system for the Puget Sound, as part of the open-source Orcasound Project.

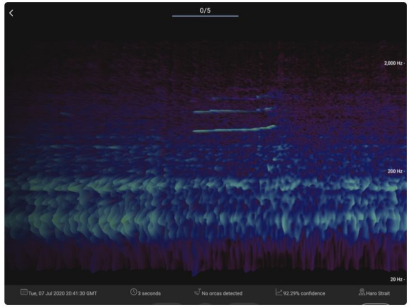
Now Playing:

You are listening to a rare 'Super Pod' gathering recorded at Orcasound lab on September 11th, 2020 (21:00 - 23:00 PST). J (& K?) pod northbound in the darkness; DH reports "amazing vocals at LK ~21:30 as well. Contains a burst of vocal activity with stretches of several overlapping calls in high SNR.

ORCASOUND LAB HYDROPHONE

Developed in collaboration with Orcasound at MS Hack 2019. Best experienced on Chrome/Edge Desktop. Orcasound on Github

POD.CAST ANNOTATION SYSTEM



ORCA AL – SRKW

ML – ON THE EDGE

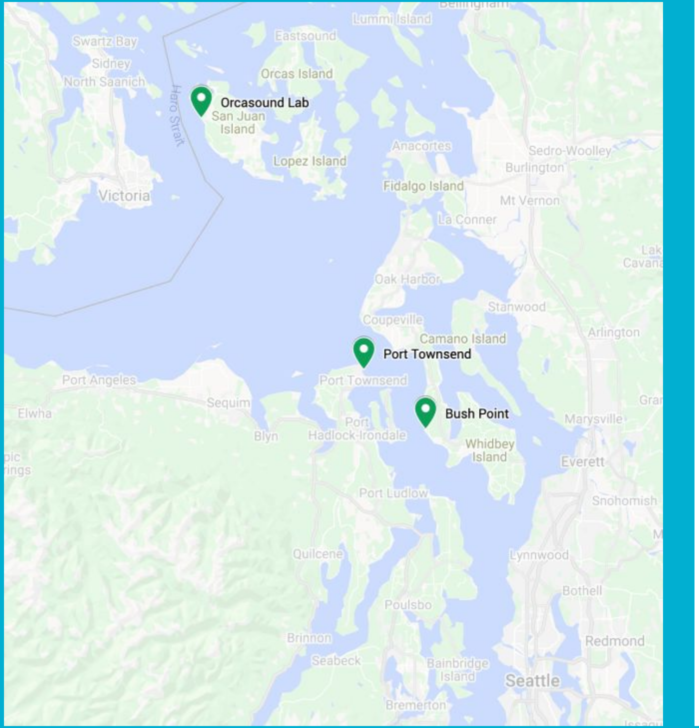
AI for orcas (#ai4orcas) -- ai4orcas.net

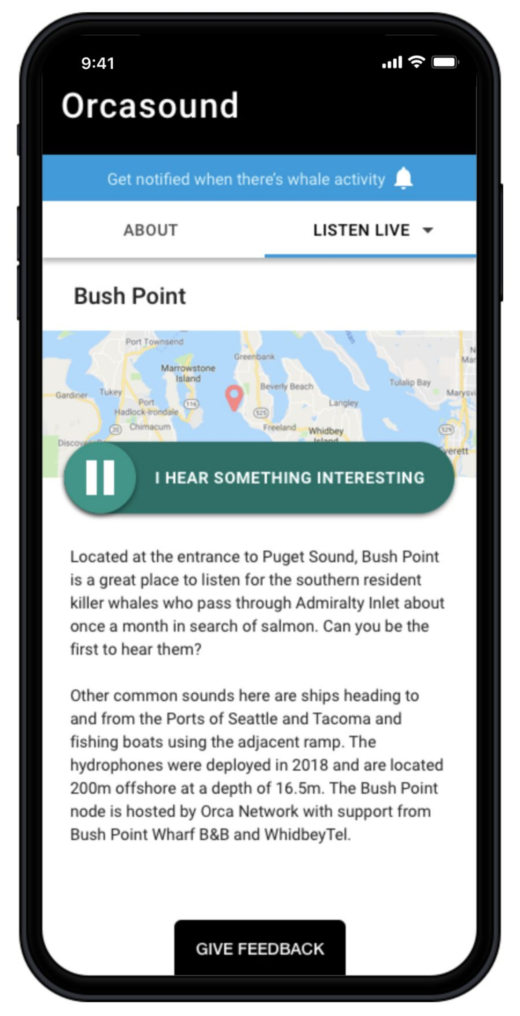
towards (more) open (marine) bioacoustic data science...

ORCASOUND

A hydrophone network (WA, USA) & open-source software community

- 3 cabled nearshore sites streaming 24/7 in 2020
- Citizen scientists detect orcas in real-time via web app -- live.orcasound.net
- Real-time inference in the cloud (deployed Sep 2020 at 3 locations) & on the “edge” at Orcasound Lab
- Github repositories of open-source code





9:41

Orcasound

Get notified when there's whale activity

ABOUT

LISTEN LIVE

Bush Point

I HEAR SOMETHING INTERESTING

Located at the entrance to Puget Sound, Bush Point is a great place to listen for the southern resident killer whales who pass through Admiralty Inlet about once a month in search of salmon. Can you be the first to hear them?

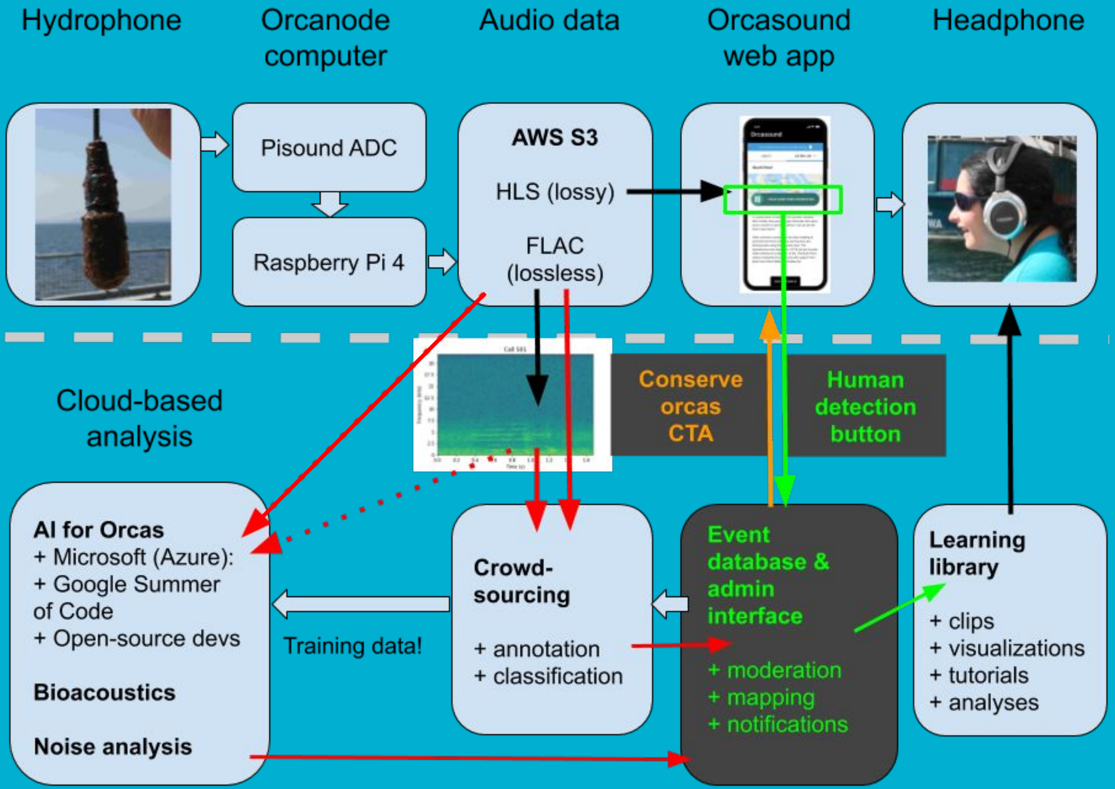
Other common sounds here are ships heading to and from the Ports of Seattle and Tacoma and fishing boats using the adjacent ramp. The hydrophones were deployed in 2018 and are located 200m offshore at a depth of 16.5m. The Bush Point node is hosted by Orca Network with support from Bush Point Wharf B&B and WhidbeyTel.

GIVE FEEDBACK

Orcasound data flow & ML opportunities

- Currently, real-time inference on Orcasound data happens in the **cloud** (Microsoft hackathon volunteer effort)

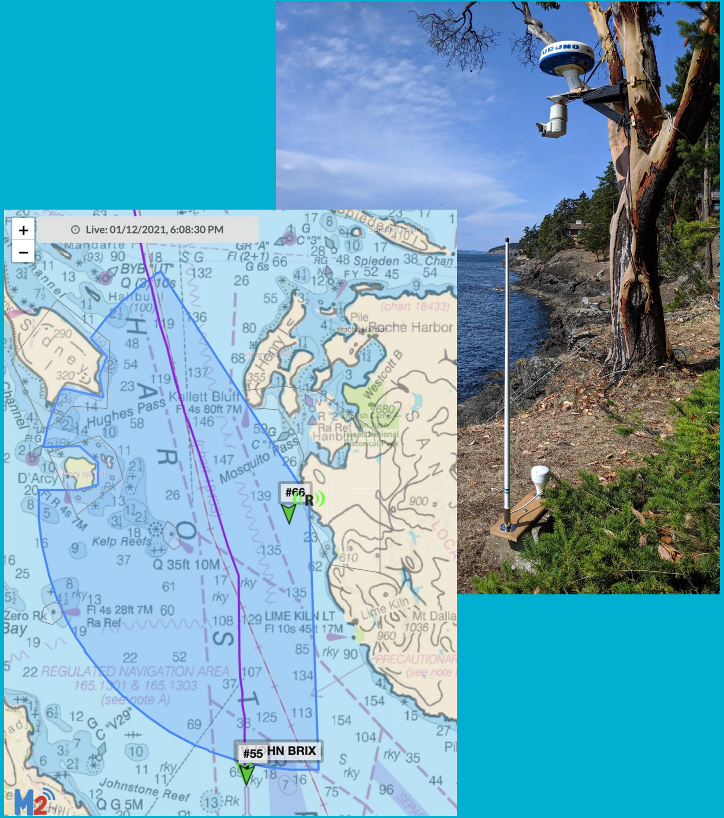
- As training/test data sets grow & model performance becomes reliable, **edge**-computing can help nodes like Val's that are bandwidth-limited



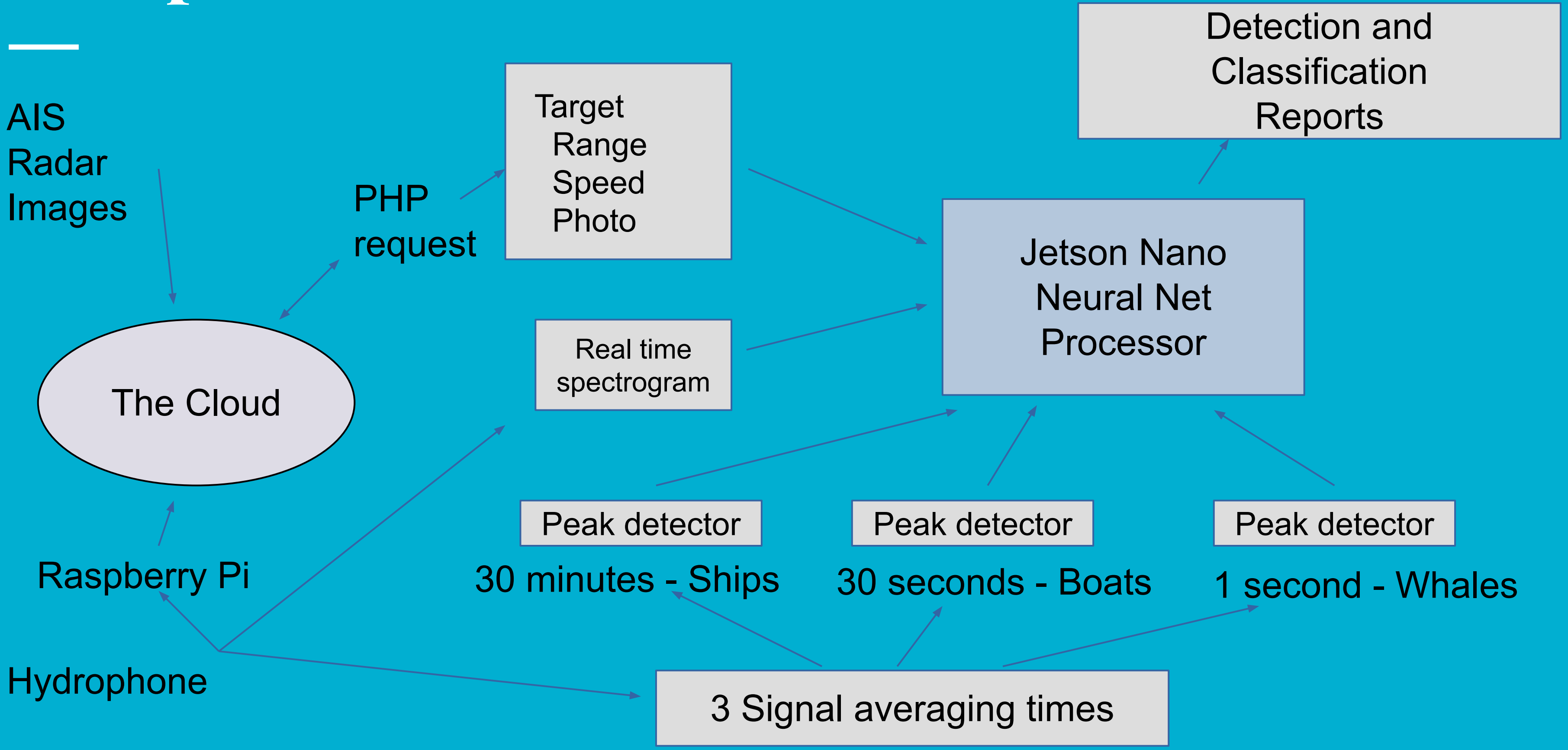
M2 system – Tracking vessels while listening

- M2 = Marine Monitor
- Partnership with Protected Seas (CA, USA)
- 24/7 real-time monitoring system:
  - AIS tracking for vessels near Orcasound Lab
  - Radar detection & tracking
  - Pan/tilt/zoom camera photographs targets
  - Data archived locally & in the cloud

Allows validation of acoustic signals from vessels: "boat" or "ship" classifications



Computational Network



AIS
Radar
Images

PHP
request

Target
Range
Speed
Photo

Detection and Classification Reports

Jetson Nano Neural Net Processor

The Cloud

Real time spectrogram

Peak detector

Peak detector

Peak detector

Raspberry Pi

30 minutes - Ships

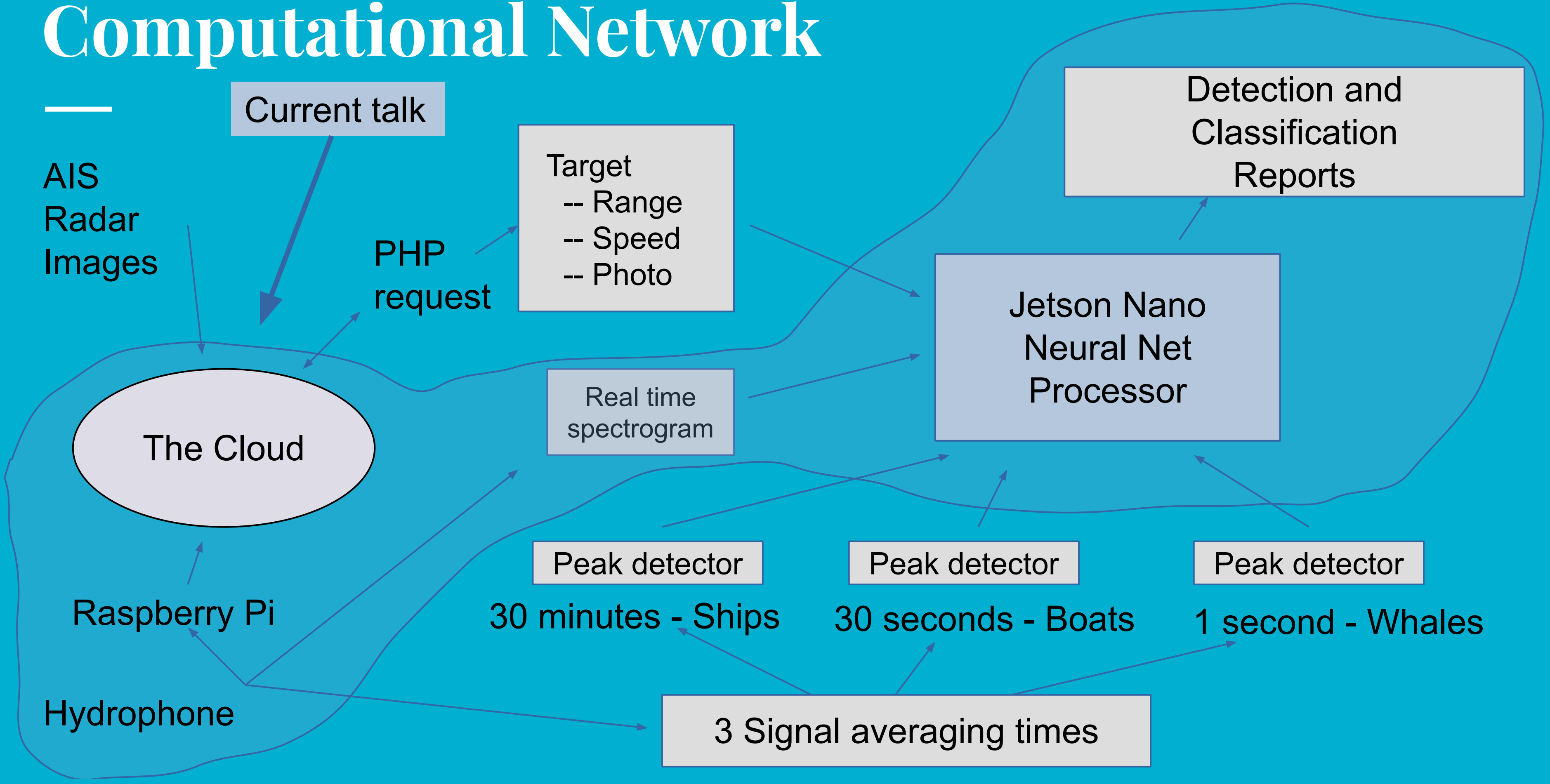
30 seconds - Boats

1 second - Whales

Hydrophone

3 Signal averaging times

Computational Network



Jetson Nano

Jetson Nano developer kit (US $99)

- 128-core NVIDIA GPU
- 4 CPUs running at ~20% load
- ~7 Watts power consumption
- Launched 2019 along w/RPi4

Digitizes audio, calculates PSDs, runs ResNet model

PyCharm (Python programming IDE)

Neural Net Development

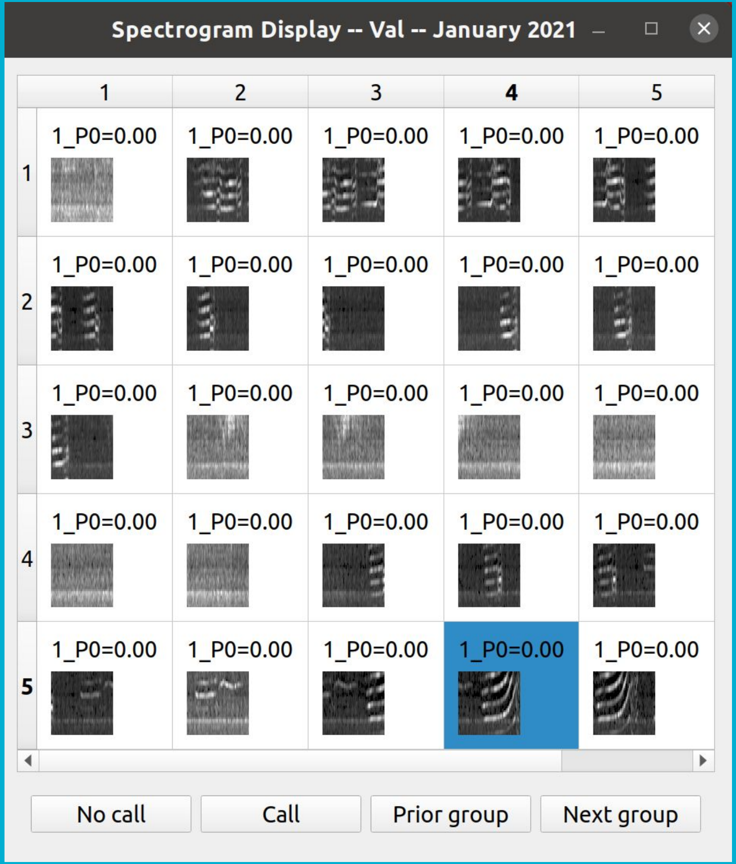
Status as of Jan 2021

- Meridian’s Ketos front end to Tensorflow and ResNet (using default parameters)
- Programming (via pyCharm) & neural net training done mostly on laptop
- Trained ResNet runs on Jetson Nano taking advantage of NVIDIA neural net processor
- Jetson Nano calculates power spectra from hydrophone feed and preprocesses spectrograms
- Jetson Nano issues notifications (spectrogram + sound clip) via secure ftp

Spectrogram preprocessing

- Hydrophone signals are pre-processed into spectrograms:
  - Power spectral density (PSD) calculated for 100 30-ms windows
  - The frequency spectrum is integrated into 100 equal-width bands between 600 & 6000 Hz
  - Each spectrogram is normalized to its maximum and minimum value
- A long-term (10-minute) running average of the PSD is used for background subtraction & the result is re-normalized between 0 and 1
- For purposes of visual display, though not for the neural net, the normalized PSD values are square-rooted twice to raise the lower intensity PSD peaks relative to the maximum (which has been normalized to 1).

Initial labeling by expert



	1	2	3	4	5
1	1_P0=0.00	1_P0=0.00	1_P0=0.00	1_P0=0.00	1_P0=0.00
2	1_P0=0.00	1_P0=0.00	1_P0=0.00	1_P0=0.00	1_P0=0.00
3	1_P0=0.00	1_P0=0.00	1_P0=0.00	1_P0=0.00	1_P0=0.00
4	1_P0=0.00	1_P0=0.00	1_P0=0.00	1_P0=0.00	1_P0=0.00
5	1_P0=0.00	1_P0=0.00	1_P0=0.00	1_P0=0.00	1_P0=0.00

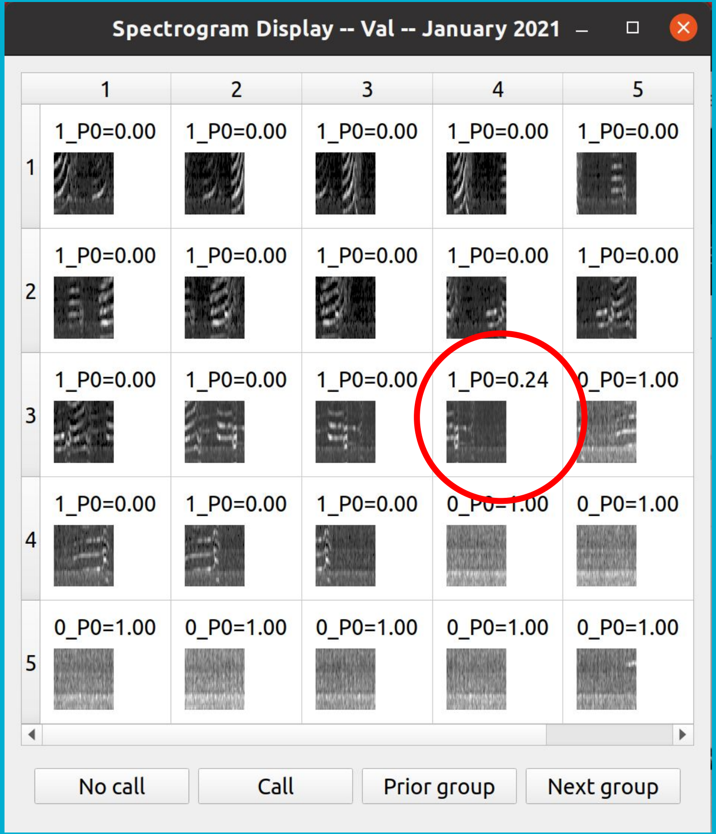
These are typical spectrograms prior to binary classification (0 = No call; 1 = Call)

1. Generate 3-second spectrograms
   - start with call-rich recording
   - 3-sec window every ~1 second
   - display as table

2. Label each sample
   - visually inspect
   - click to listen if uncertain visually
   - select “No-call” or “Call” button

3. As this was a call-rich recording, the default label was set to “Call” so I only had to re-label samples that had no calls.

Neural net development



	1	2	3	4	5
1	1_P0=0.00	1_P0=0.00	1_P0=0.00	1_P0=0.00	1_P0=0.00
2	1_P0=0.00	1_P0=0.00	1_P0=0.00	1_P0=0.00	1_P0=0.00
3	1_P0=0.00	1_P0=0.00	1_P0=0.00	1_P0=0.24	0_P0=1.00
4	1_P0=0.00	1_P0=0.00	1_P0=0.00	0_P0=1.00	0_P0=1.00
5	0_P0=1.00	0_P0=1.00	0_P0=1.00	0_P0=1.00	0_P0=1.00

1. Use expert-labeled set to train a ResNet via Ketos and Tensorflow.

2. Examine predictions for a new recording (P<0.5 = “No call” and “0” label)

3. Use this program to correct predictions
   a. Does a **partial call** get a “Call” label?
   b. ~5 minutes for a 120x10 grid
   c. <10% correction rate

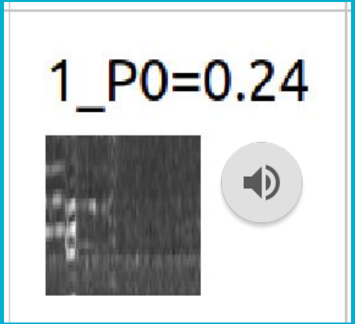
4. Retrain model with new round of training data and iterate from step 2.

New labeled SRKW call data

Current labeled training set:

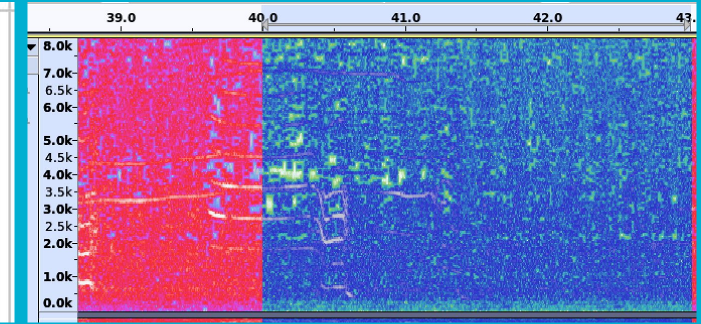
- ~5000 samples
- ~½ SRKW calls
- Only ~2 hours of recordings so far
- Labels from single expert (Val)

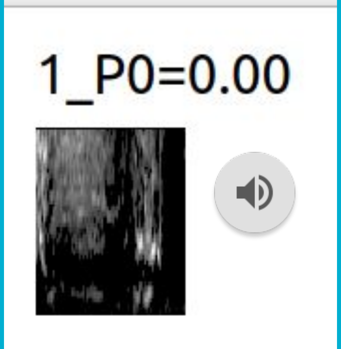
Examples of true/false positives:



1_P0=0.24

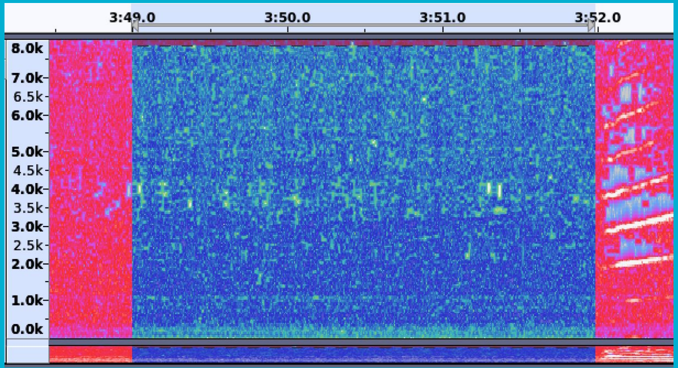
Tail end of a call



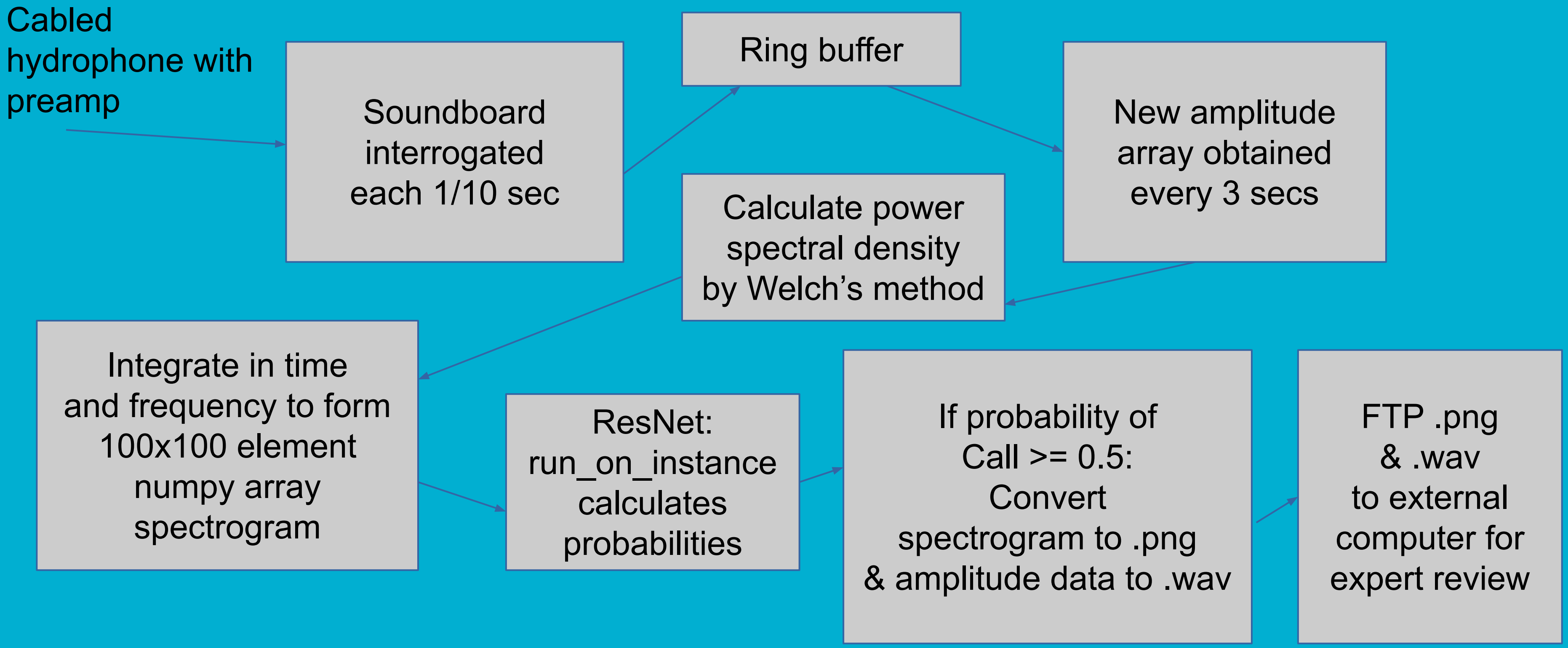


1_P0=0.00

False positive



Realtime inference on the Jetson Nano



Cabled hydrophone with preamp

Soundboard interrogated each 1/10 sec

Ring buffer

New amplitude array obtained every 3 secs

Calculate power spectral density by Welch’s method

Integrate in time and frequency to form 100x100 element numpy array spectrogram

ResNet: run_on_instance calculates probabilities

If probability of Call >= 0.5: Convert spectrogram to .png & amplitude data to .wav

FTP .png & .wav to external computer for expert review

User parameters:

- Residual Neural Network (vs e.g recurrent NN)
- Mostly use default pre-processing parameter values:
  - Nfft = 512 *# Number of PSD frequency bins will be one half of Nfft*
  - **n_slices** = 100 *# Number of bins along time axis*
  - **n_bands** = 100   *# Number of bins along frequency axis*
  - logFreq = False *# If true, n_bands are a log frequency axis (compressing high freq bands)*
  - **f_low** = 400
  - **f_high** = 6000
  - samplerate = 24000 *# sample rate for sound board*
  - **deltaT** = 3 *# number if seconds in a single spectrogram array*
  - stepFrac = 1 / 3 *# advance this fraction of deltaT for each new spectrogram*
  - secsSkip = 0 *# skip this many secs between successive stepFrac groups of spectrograms*

Where we go from here

Call – No call Neural Net: (currently operational)

- Run Jetson Nano & Resnet model on live hydrophone data
- Use false positives to retrain neural net (in external computer)
- Automatically update Jetson Nano with retrained model via sftp

Implement multi-class neural net (Hydrophone + M2 reports):

- Killer whale calls (and whistles)
- AIS vessel (ship or smaller boat that transmits AIS reports)
- Speedboat
- Harbor seal (“heavy-breather”)
- Humpback call
- Echolocation clicks

Beam Reach acknowledgements & links

Thanks to supporters & colleagues!

- The Orcasound open-source community's volunteer hackers!
- Protected Seas
- Anthropocene Institute
- $10k Amazon start-up credits to Beam Reach
- Project HALLO (2020-2022)

Join us!

ai4orcas.net