Real-time detection and classification of underwater soundscape signals via a

convolutional autoencoder

Val Veirs and Scott Veirs

Orcasound.net

vveirs and sveirs @ gmail

Our orcas: the southern salmon seekers

Southern Resident Killer Whales (SRKWs) are:

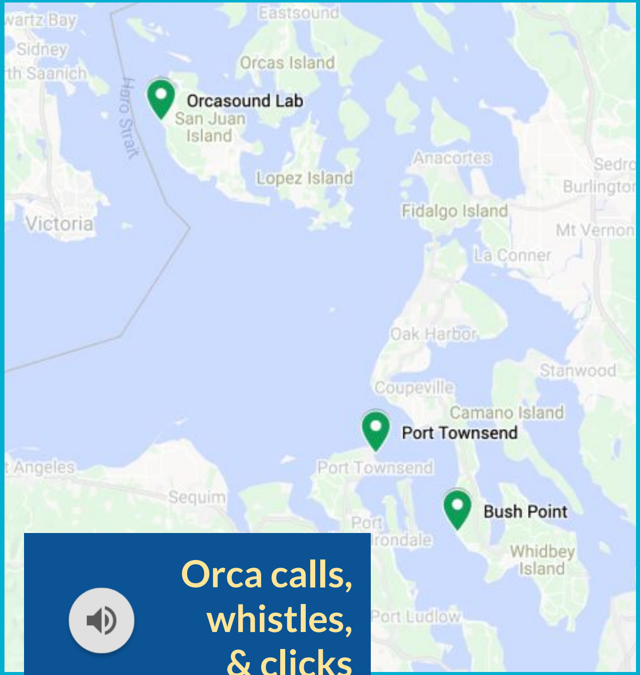
- **Southern** = ranging from northern California to SE Alaska
- **Resident** = historically re-occurring within the Salish Sea (inland waters of WA and BC)
- **Killer** = apex predators, salmon specialists
- **Whales** = cultural icons, both historically & as modern “charismatic megafauna”

ORCASOUND

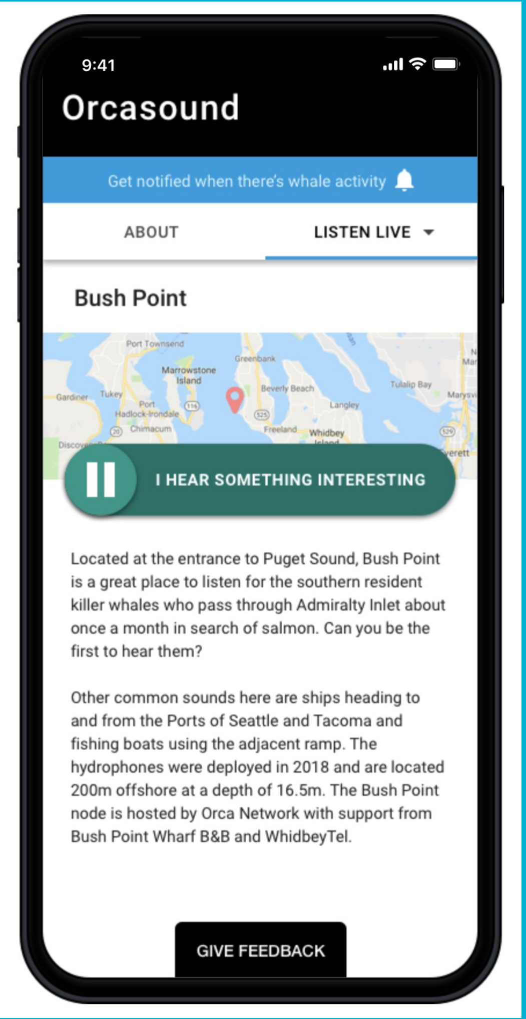
A hydrophone network (Puget Sound, WA, USA)

- 3 cabled nearshore sites streaming 24/7 in 2021
- Community scientists detect orca sounds in real-time via a web app -- live.orcasound.net
- AI is starting to listen, too -- ai4orcas.net

How can AI & human listeners understand soundscapes and advance marine bioacoustics?



Orca calls, whistles, & clicks



ORCASOUND

Can we automatically characterize the “Puget Soundscape” ?

- Humpback non-song sounds: e.g. creaks, whoops, moans
- Speed boats (~minutes)
- Ships (~hour)

Humpbacks

Ships

Boats

How can AI detect and classify this wide range of signals and noise?

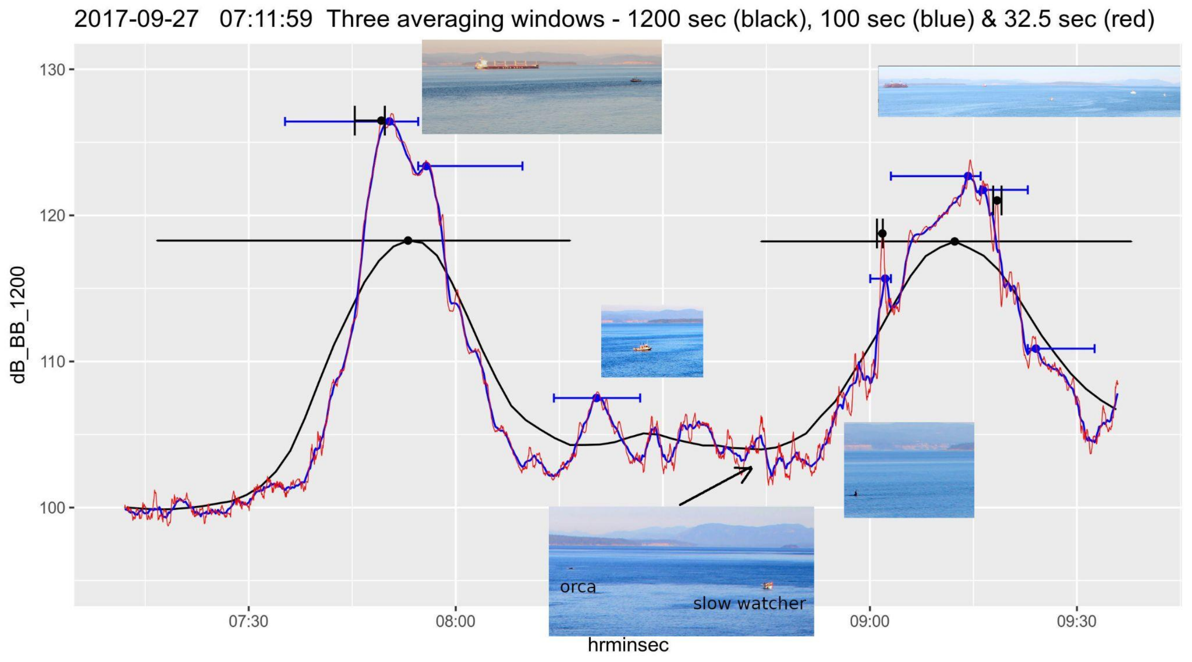
ORCASOUND

Averaging schemes for the time domain

- Motivation for temporal averaging experiments

- Averaging windows (factors of 8)
    - 3, 24, 200, 1500 secs
    - 4-hr windows

- Common ML approach is fixed arrays (e.g. 3 seconds into a 128x128)



Motivations for an autoencoder (1)

Current Machine Learning applications here under the Salish Sea center on the detection of signals in categories that have been determined by trained listeners.

Large quantities of labeled data are used to train neural networks to detect signals such as orca calls.

To fully assess the soundscape, such a ML system would need labeled data for myriad sound categories from microsecond-long clicks to hour-long ship passages, etc.

Motivations for an autoencoder (2)

An autoencoder is a type of artificial neural network that extracts low dimensional ‘features’ from high dimensional signals.

These nonlinear features are measures of correlation between nearby regions in spectrograms (frequency vs time plots).

The features extracted from any spectrogram contain enough information to reconstruct the spectrogram to a reasonable degree.

These features can then be used as input to either supervised or non-supervised classifiers.

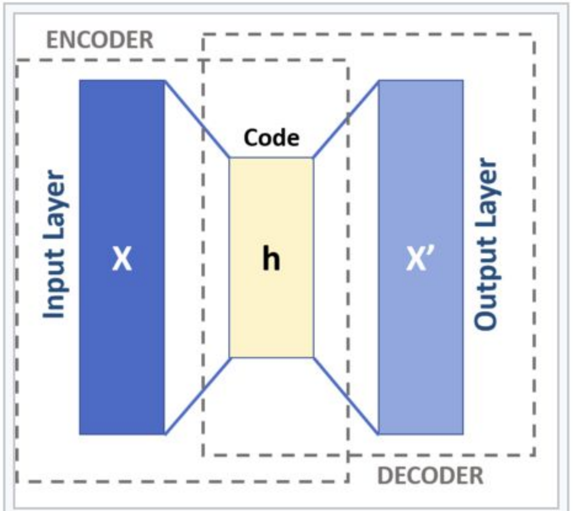
Motivations for an autoencoder

Autoencoder feature extraction is robust to variations in the input signal.

Since the autoencoder representation contains many fewer bits of information in any spectrogram, similar features are triggered by variations in the input signals.

This robust feature detection makes similar signals close together in feature space.

Extracting and Composing Robust Features with Denoising Autoencoders
http://dx.doi.org/10.1145/1390156.1390294



Schema of a basic Autoencoder

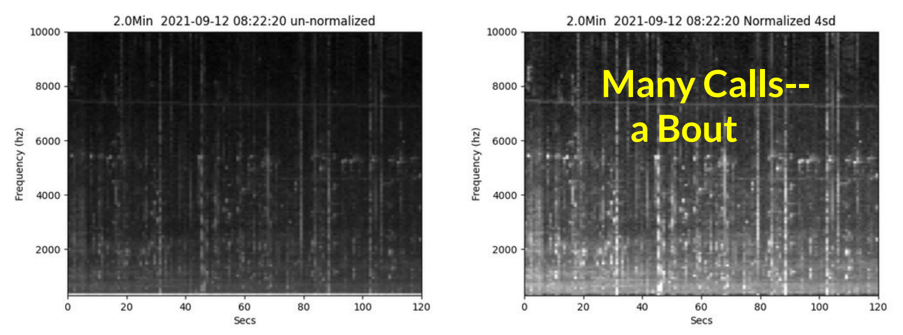
Wikipedia: Autoencoder

Methods: **pre-processing**

Fixed size arrays of power spectral density vs time - 128 x 128 psds

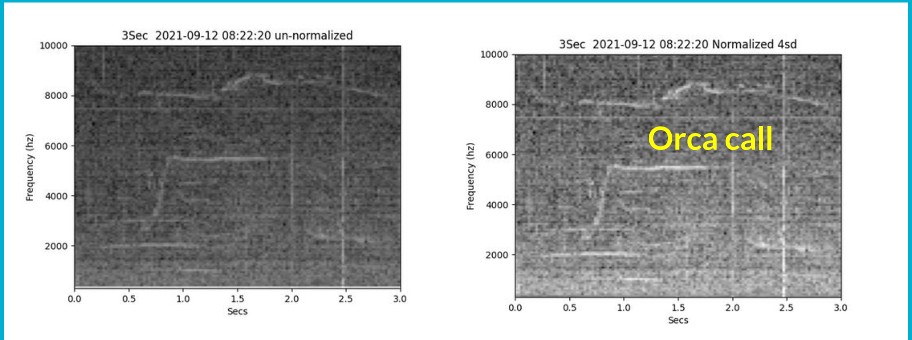
Various averaging times

2 minute spectrograms

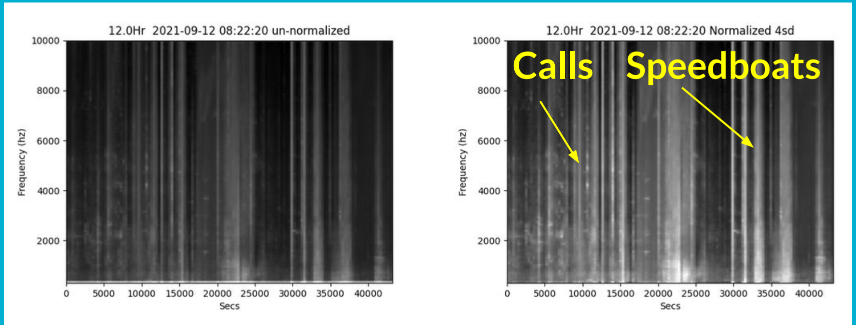


3 second spectrograms

Un-normalized psd Normalized to mean += 3 stdev



1 hour spectrograms



Methods: autoencoder description

- Acoustic pattern detector
- Spectrogram input (128x128)
- Choke point layer (8x8x32)
- Layer sequence is reversed until the shape gets back to the original 128 x 128 array.
- Enough information is ‘detected’ in the choke point that the original can be reconstructed!

Spectrogram: 16,384 values

Encoder Section - Keras Tensorflow

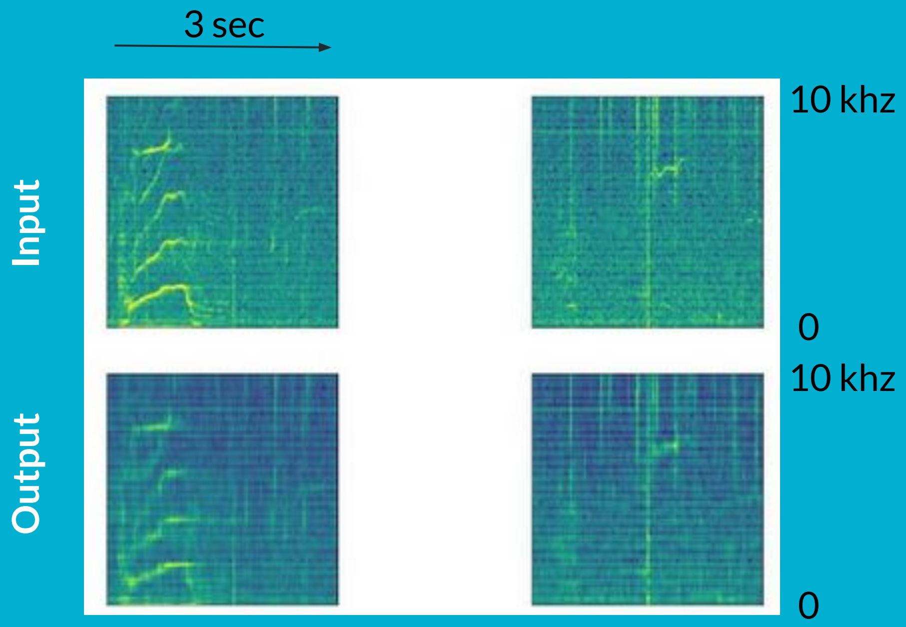
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 128, 128, 1)]	0
conv2d (Conv2D)	(None, 64, 64, 128)	1152
batch_normalization	(None, 64, 64, 128)	512
re_lu (ReLU)	(None, 64, 64, 128)	0
conv2d_1 (Conv2D)	(None, 32, 32, 64)	73728
batch_normalization_1	(None, 32, 32, 64)	256
re_lu_1 (ReLU)	(None, 32, 32, 64)	0
conv2d_2 (Conv2D)	(None, 16, 16, 32)	18432
batch_normalization_2	(None, 16, 16, 32)	128
re_lu_2 (ReLU)	(None, 16, 16, 32)	0
conv2d_3 (Conv2D)	(None, 8, 8, 32)	9216
batch_normalization_3	(None, 8, 8, 32)	128
re_lu_3 (ReLU)	(None, 8, 8, 32)	0

Total params: 103,552
Trainable params: 103,040

Features: 2048 values

Methods: autoencoder training

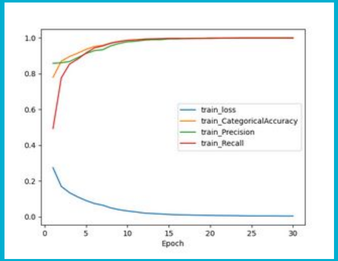
- The autoencoder was trained on 15,000 spectrograms with averaging times of 3-sec to 1-hr.
- Training is accomplished via Tensorflow (Google opensource) learning to create outputs that are as close as possible to the corresponding inputs.
- No expert classes needed!

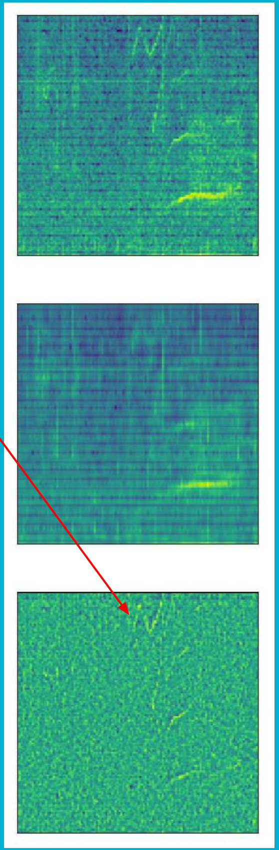


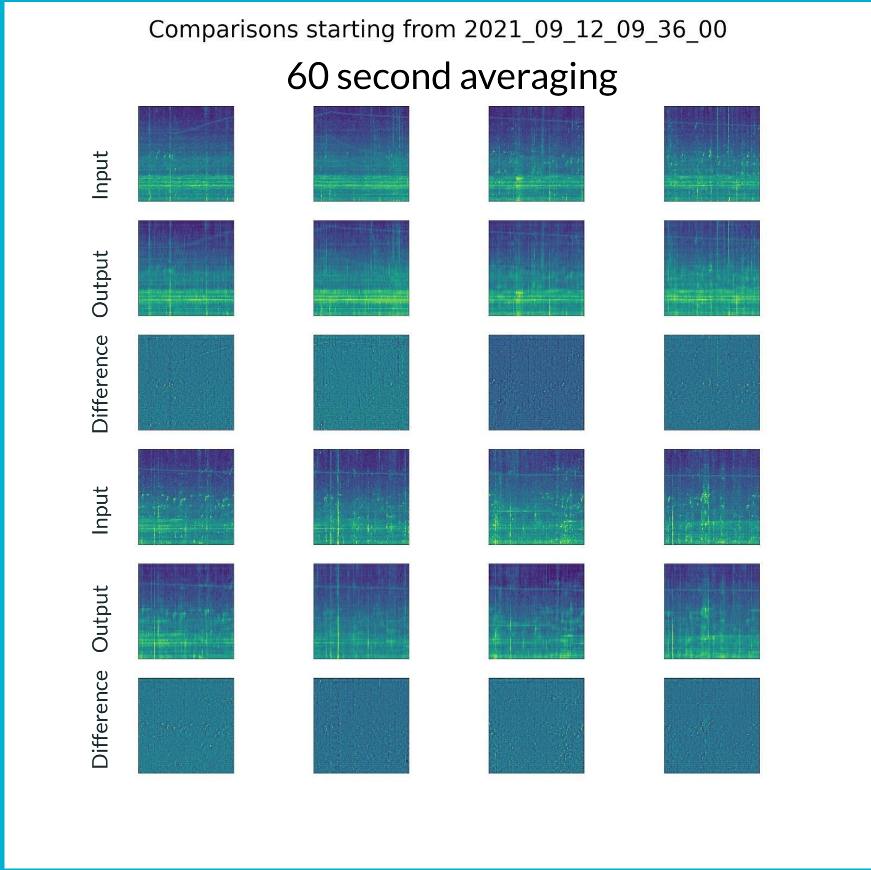
Results: low-loss signal encoding

SRKW call autoencoding

- Samples are de-noised
- Residuals are minimal
- Perhaps more training will reduce these residuals?





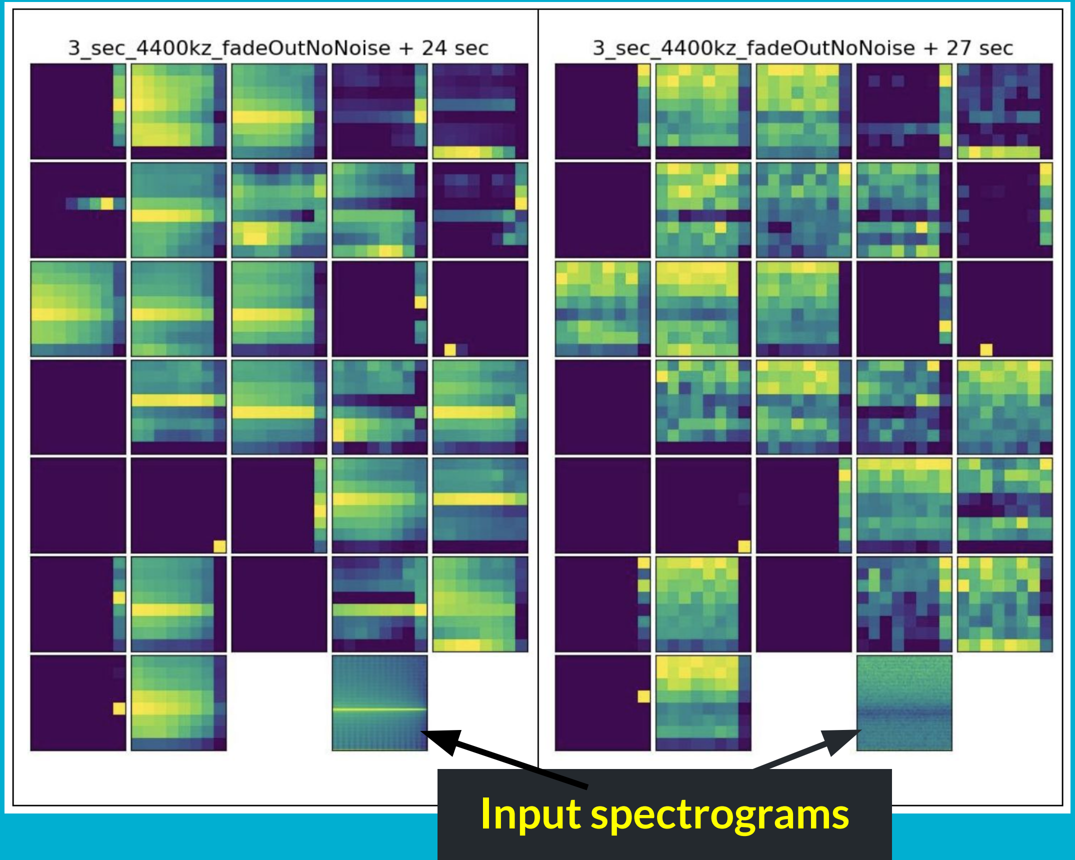


Encoder: synthetic signal vs no-signal

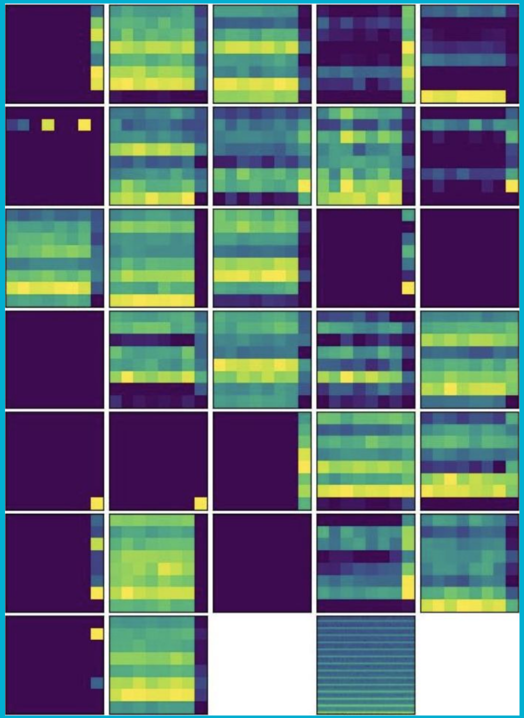
Bottleneck visualization:

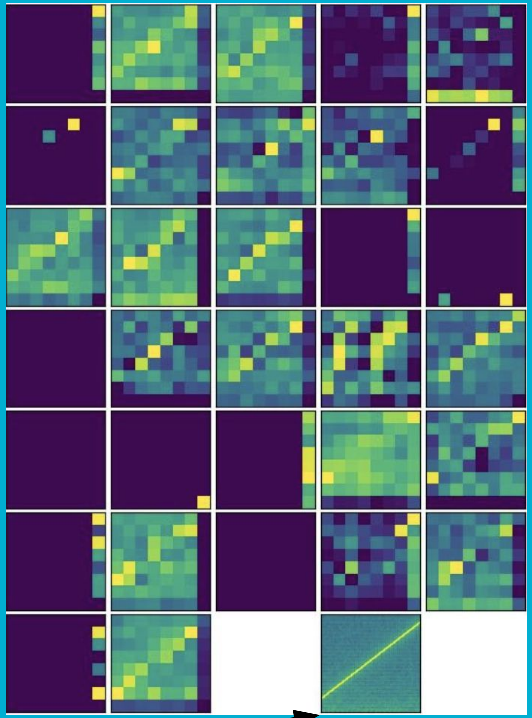
- 4.4 kHZ tone
- Amplitude fading to zero

- No noise
- Features have compressed the PSD data by a factor of 8!



Encoder: synthetic signals



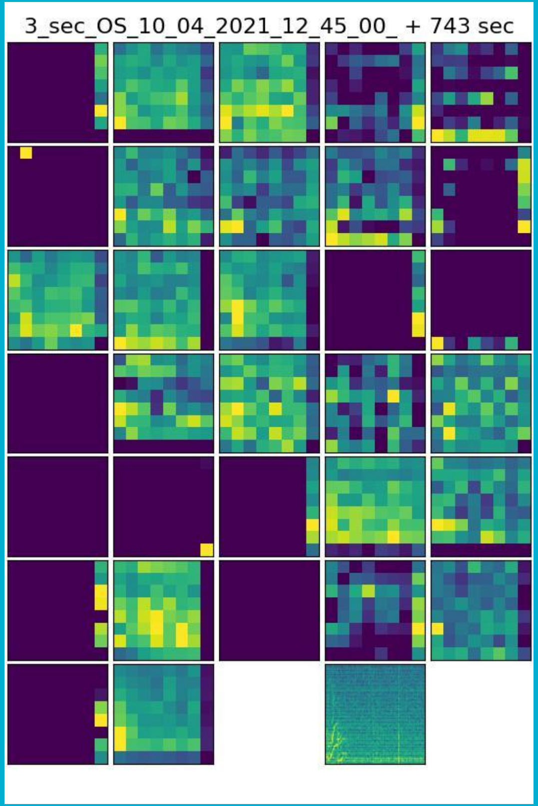


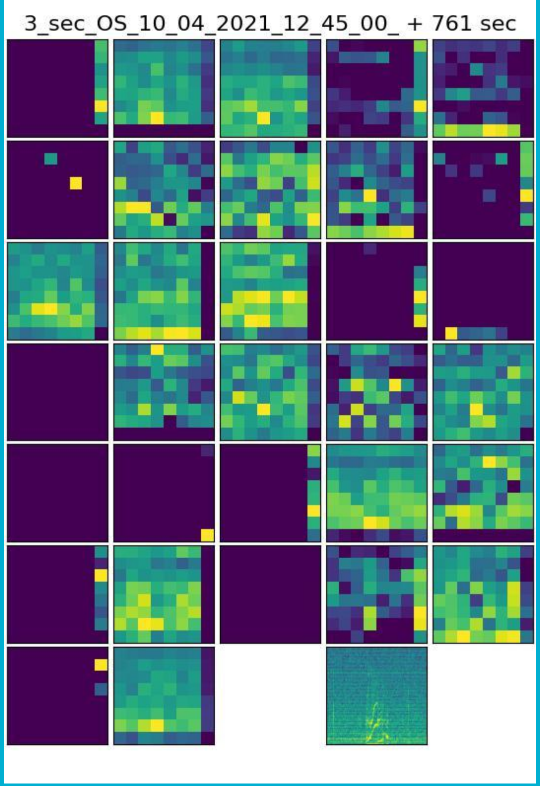
Input spectrograms

350 Hz, square, Brownian noise

Frequency ramp, steep slope

Features for some orca calls

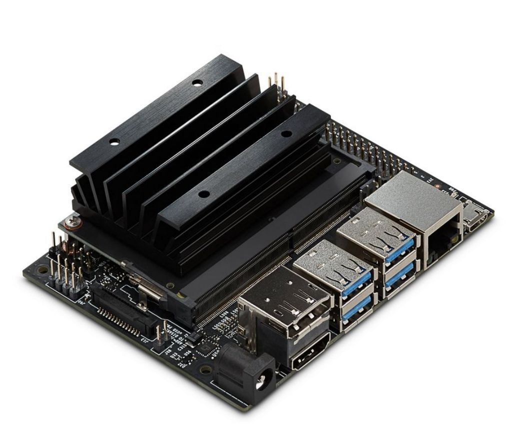




Next steps and applications:

ORCASOUND

- Train autoencoder on a larger dataset
- Implement clustering:
    - Supervised . . . CNN -> expert classes
    - Unsupervised . . . e.g. Random Forest
- Compare results of direct spectrogram inputs vs the features extracted for each spectrogram.
- Implement system on Jetson Nano running at the hydrophone feed at Orcasound Lab archiving and reportin cluster detections by cluster class.



Acknowledgements & links

Give orcas a voice!

live.orcasound.net

Thanks to all our collaborators!

- The Orcasound open source community's volunteer hackers
- The many NGOs & volunteers who maintain the hydrophones at each node
- Oliver Kirsebom & Fábio Frazão et al. at Meridian teaching us about open source deep learning

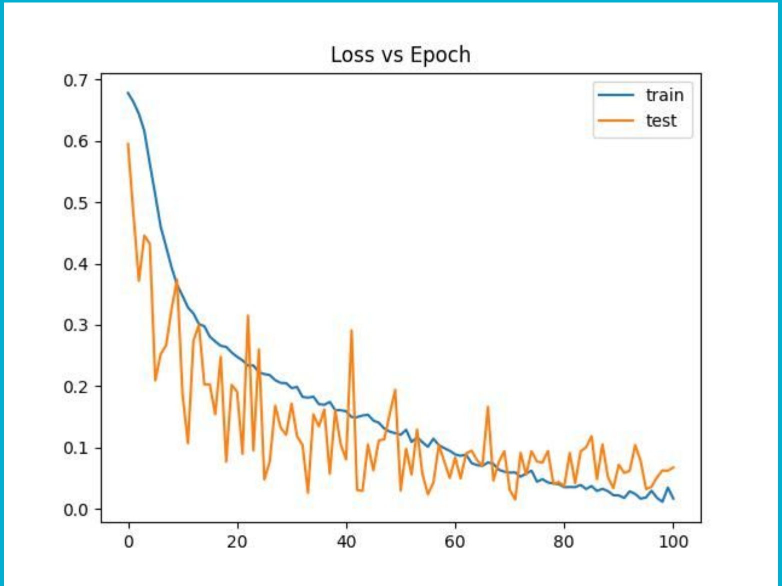
Explore the soundscape, AI, & code :

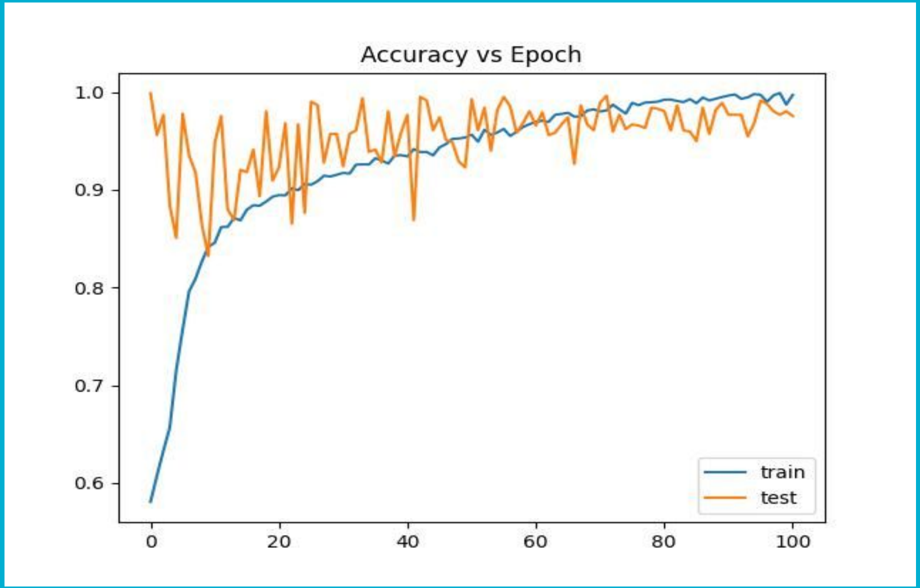
- orcasound.net/learn
- ai4orcas.net
- github.com/orcasound

Preliminary classifier results...

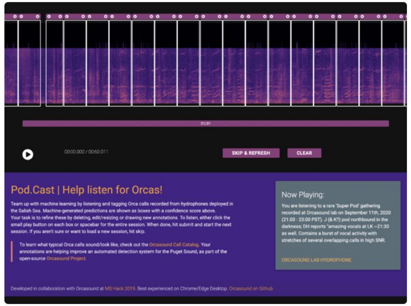
Pod.cast annotations of SRKW calls used to train binary classifier

Inputs to classifier are features from the autoencoder's encoder section.

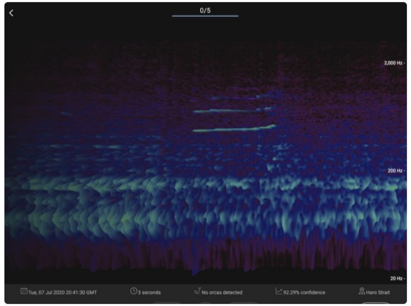




Artificial Intelligence for orcas



POD.CAST ANNOTATION SYSTEM



ORCA AL – SRKW

ML – ON THE EDGE

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

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