Lessons learned from MERIDIAN.
Open science principles for underwater acoustic detectors and classifiers

Fabio Frazao

Institute for Big Data Analytics/MERIDIAN
@ Dalhousie University

MERIDIAN Winter webinar series - December 15th, 2020

Outline

- Data
  - Dataset size, quality and representativeness
  - Annotation levels and compatibility
- Methods
  - Well-documented metrics, models, and processing algorithms
- Software
  - Open-source code
  - Adaptability, reusability and collaboration principles

Dataset size, quality and representativeness

Dataset size, quality and representativeness

Desired dataset:

- Many samples (1000s)

- Represents a great variety of conditions (reflecting the use case/goal)
  - Variations of the target signals
  - Variations background sound/ non-target signals
  - Collected with a variety of instruments

Dataset size, quality and representativeness

Desired dataset:

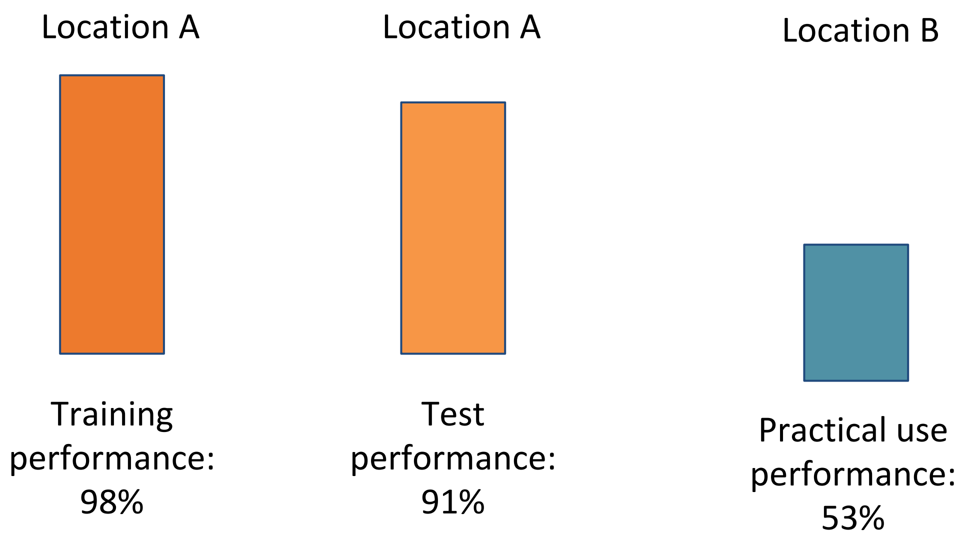
- Many samples (10,000s to 100,000s)

- Represents a great variety of conditions (reflecting the use case/goal)
    - Variations of the target signals
    - Variations background sound/ non-target signals
    - Collected with a variety of instruments

Rarely the case!

Dataset size, quality and representativeness

- Limited representation of the conditions in which the model will be used.



Example: NARW detection

Dataset A (Gulf of Saint Lawrence, surface)

2078 samples

Dataset B (Gulf of Saint Lawrence, bottom)

3892 samples

Example: NARW detection

3s clips

Dataset A (Gulf of Saint Lawrence, surface)

2078 samples

Dataset B (Gulf of Saint Lawrence, bottom)

3892 samples

\+ 50 x 30 min files, fully annotated (for tests)

Positive and negative labels came from validating another detector.

Example: NARW detection

3s clips

Dataset A (Gulf of Saint Lawrence, surface)

2078 samples

Dataset B (Gulf of Saint Lawrence, bottom)

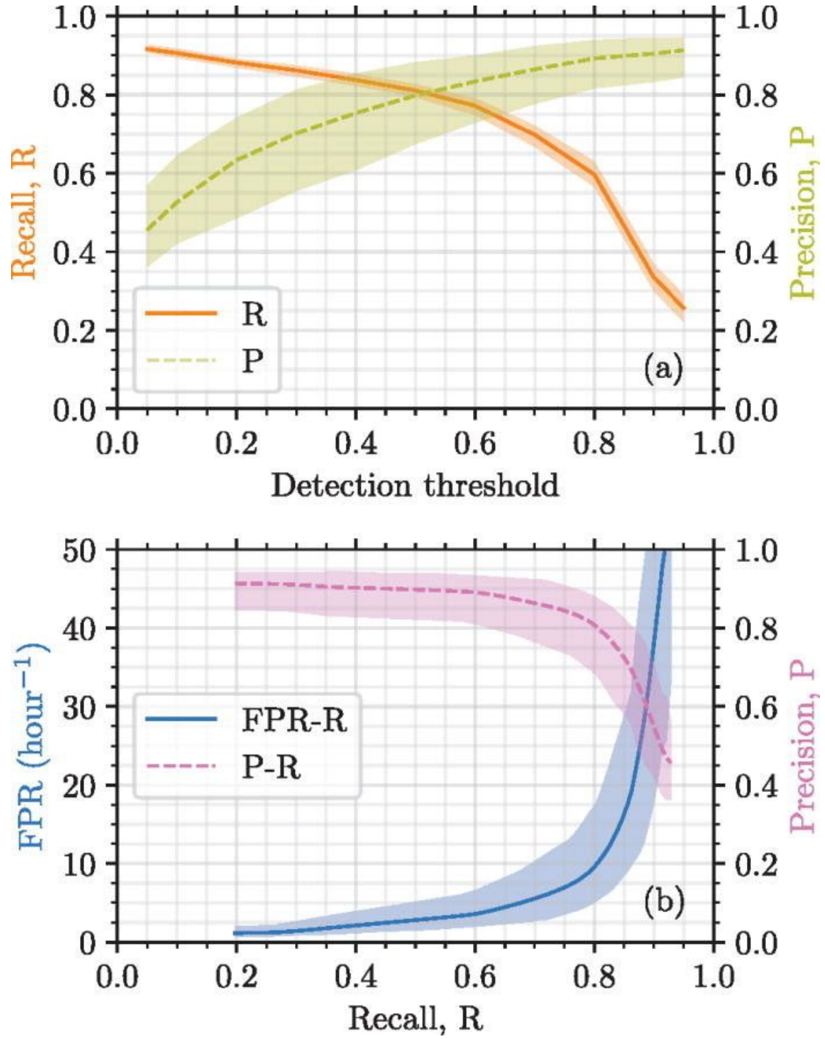
3892 samples

\+ 50 x 30 min files, fully annotated (for tests)

Dataset C (Gulf of Maine, surface)

3000 samples

Example: NARW detection

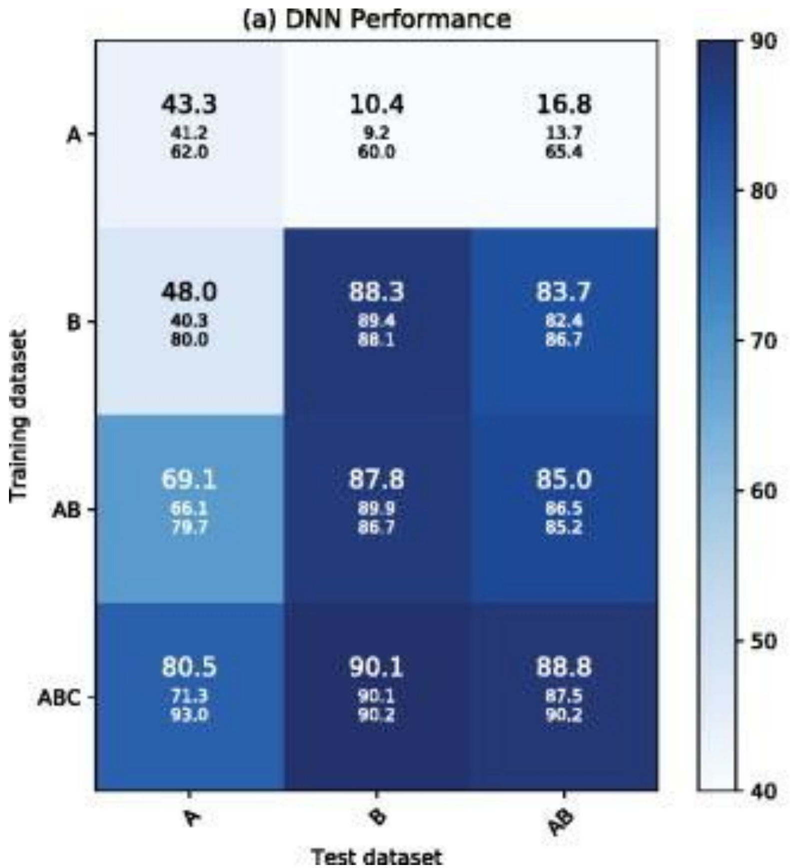


Results on continuous test set

Best model:

Precision=90%
Recall=80%
FPR/h=5

Example: NARW detection



Results on A,B, AB test sets (3s clips)

Performance of a deep neural network at detecting North Atlantic right whale upcalls

The Journal of the Acoustical Society of America **147**, 2636 (2020); https://doi.org/10.1121/10.0001132

Oliver S. Kirsebom[1,b)], Fabio Frazao[1], Yvan Simard[2,c)], Nathalie Roy[3], Stan Matwin[1,d)], *and* Samuel Giard[3]

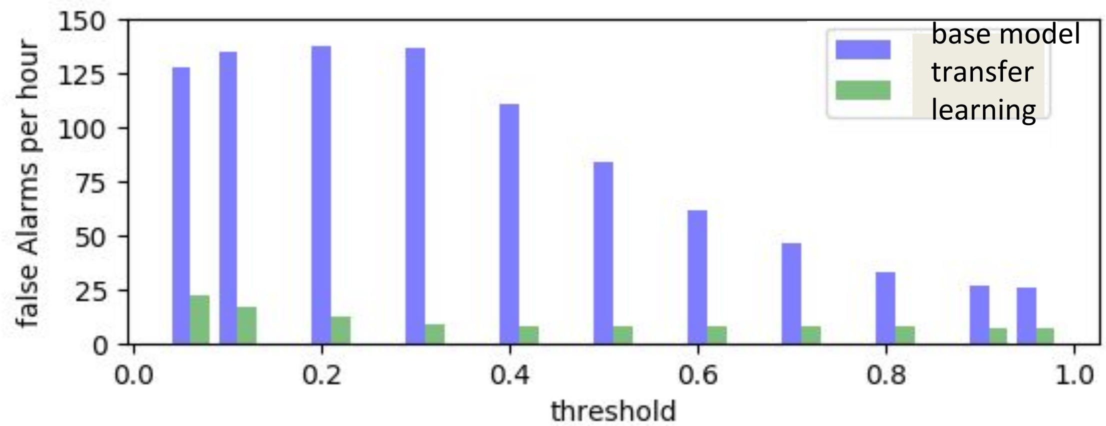
Example: NARW detection

Important: This is not THE universal NARW detector (not the goal)

Example: NARW detection

Important: this is not THE universal NARW detector (not the goal)

But it might still be a good starting point:



Adapting the Gulf of Saint Lawrence model to the Emerald Basin (lots of seismic noise)

Dataset size, quality and representativeness

Public and benchmark datasets

Dataset size, quality and representativeness

(Useful benchmark dataset for image classification)

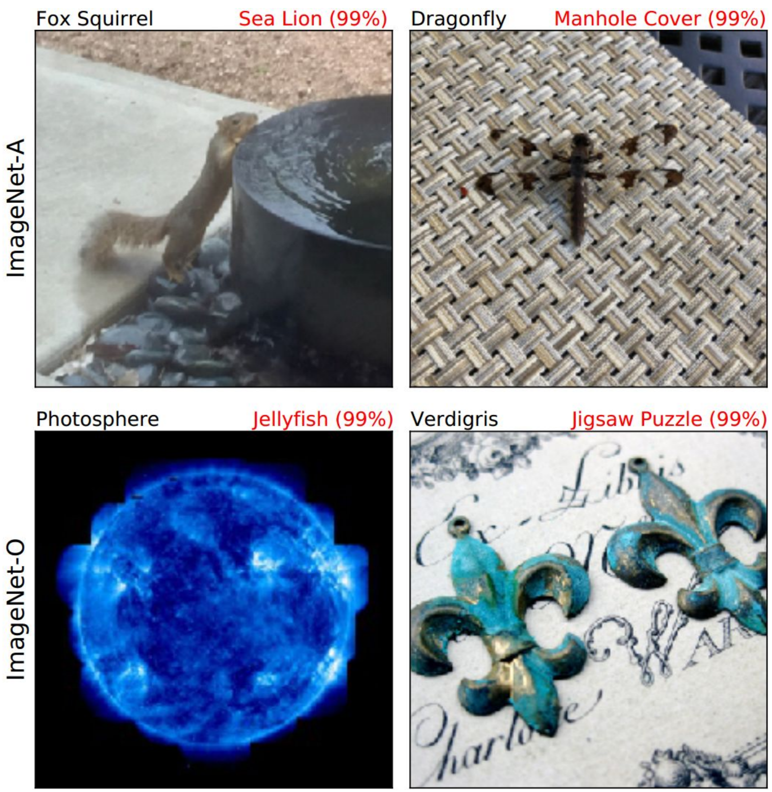
IMAGENET

images of lighters in the IMAGENET dataset

Dataset size, quality and representativeness

Still has plenty of problems

Natural adversarial examples to the IMAGENET dataset



from “Natural Adversarial Examples, Bassart et al 2020. Pre-print: arxiv.org/pdf/1907.07174.pdf

Dataset size, quality and representativeness

NEW BEDFORD WHALING MUSEUM Woods Hole Oceanographic Institution

Watkins Marine Mammal Sound Database

https://cis.whoi.edu/science/B/whalesounds/index.cfm

OCEAN NETWORKS CANADA

Recycling data: An annotated marine acoustic data set that is publicly available for use in classifier development and marine mammal research

The Journal of the Acoustical Society of America **148**, 2595 (2020); https://doi.org/10.1121/1.5147208

Kristen Kanes

Welcome to the orcadata wiki!

This is a place to share and collaborate, especially regarding bioacoustic analysis of real-time and archived audio data related to the Orcasound open source project. Here you can learn more about Orcasound: machine learning resources related to orcas and access to Orcasound data -- both archived training and testing data, and real-time audio streams.

https://github.com/orcasound/orcadata/wiki

Dataset size, quality and representativeness

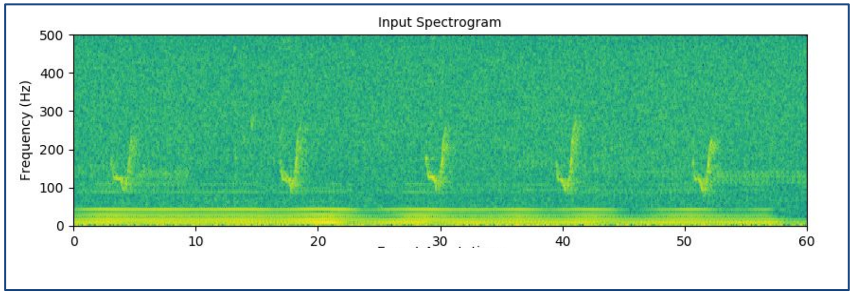
CetaSound (ENSTA-BRETAGNE + MERIDIAN)

- Multiple species, instruments, locations
- Multiple defined tasks and benchmarks
- Pre-trained models

Annotation levels and compatibility

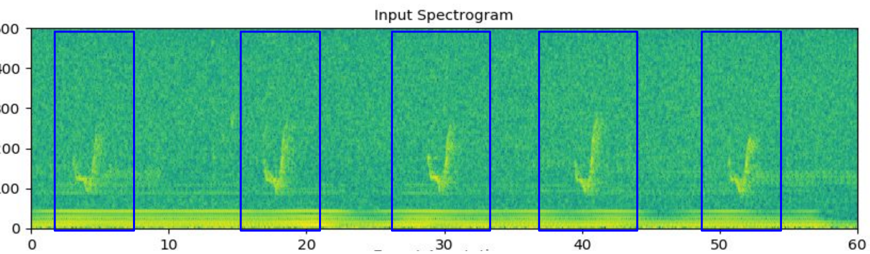
Annotation levels and compatibility

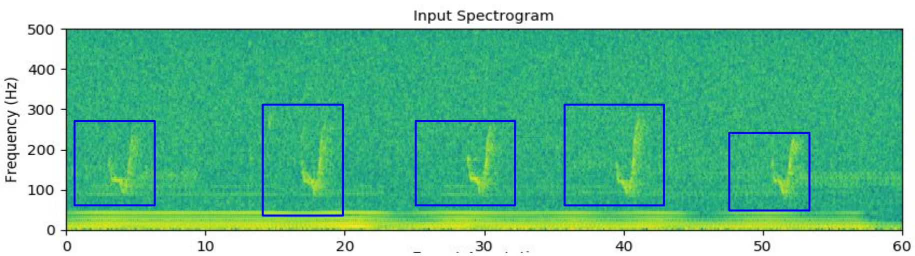
File/event level annotation



Signal of interest is present

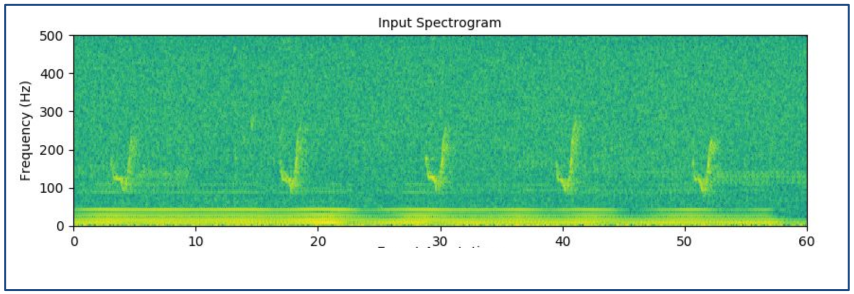
Call level annotation





Annotation levels and compatibility

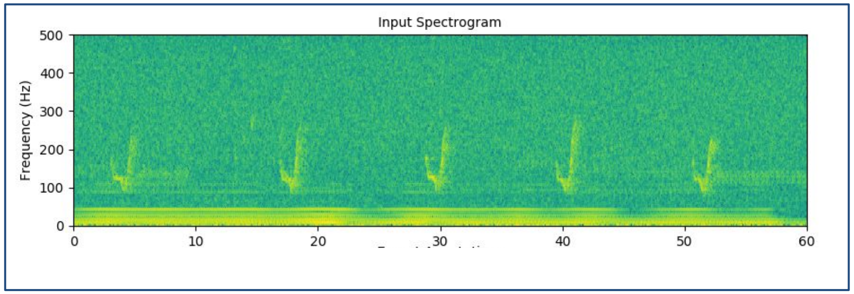
Clip classification



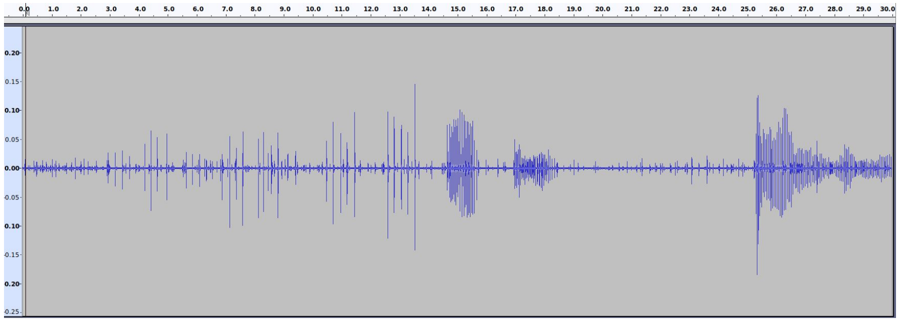
Output: “Species A detected in this 1min clip”

Annotation levels and compatibility

Clip classification

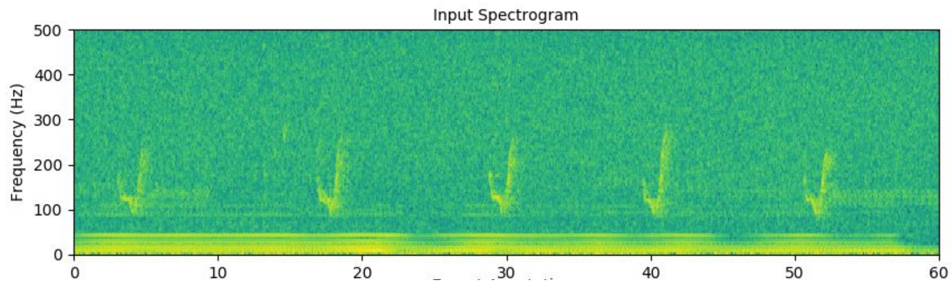


Output: “Species A detected in this 1min clip”

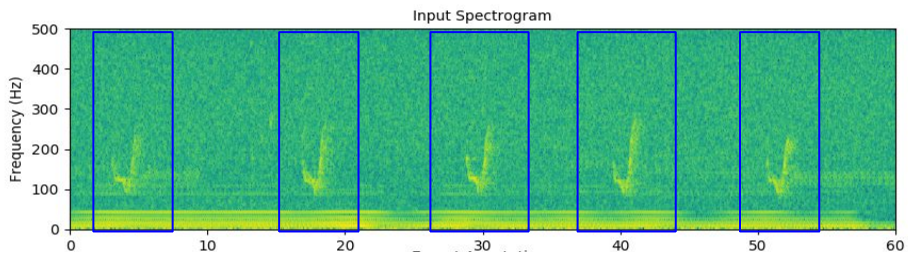


Different goals, different models

Temporal Call detection

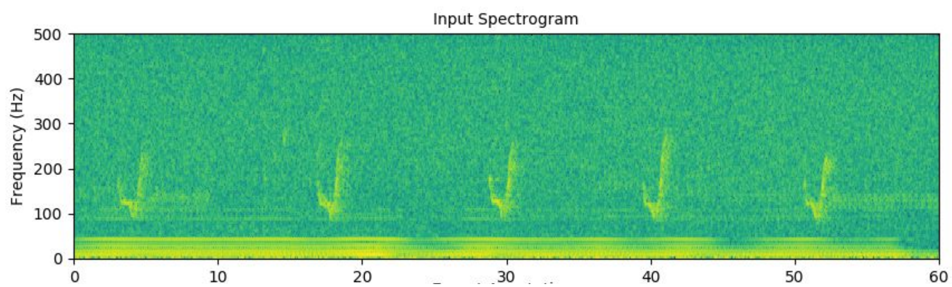


Output:

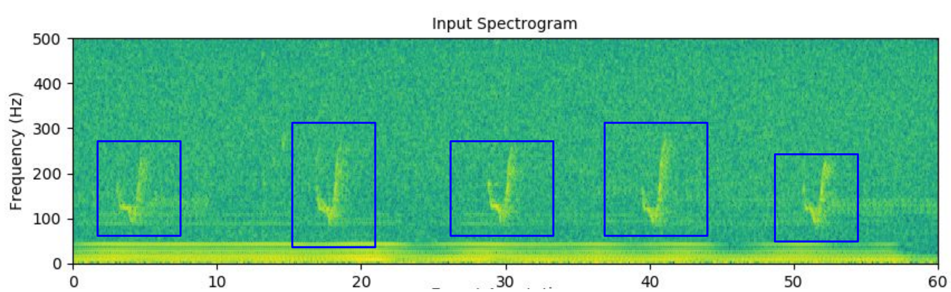


Annotation levels and compatibility

Time-frequency Call detection



Output:

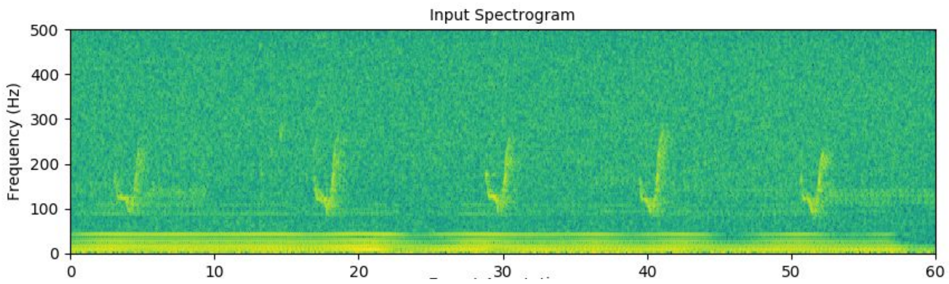


Annotations:

- Start/end time + min/max freq. for target signal(s) eg:”species A call”
- What’s in between
  - “Everything else”
  - “Species B call”, “Boat”, “environmental background”
  - More specific classes helpful to reduce misclassification, but also require bounding boxes

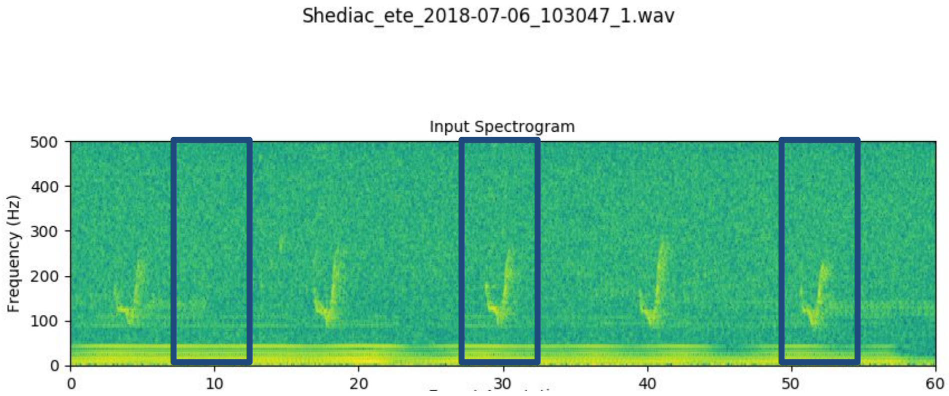
Annotation levels and compatibility

Shediac_ete_2018-07-06_103047_1.wav



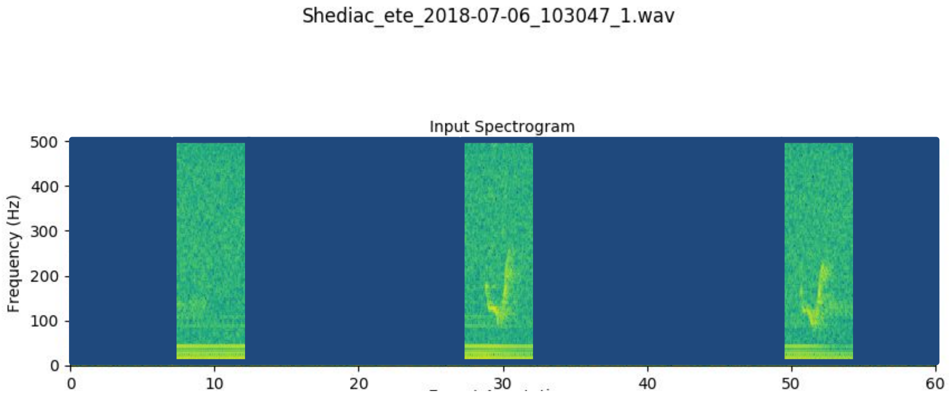
Annotation levels and compatibility

Detector 1



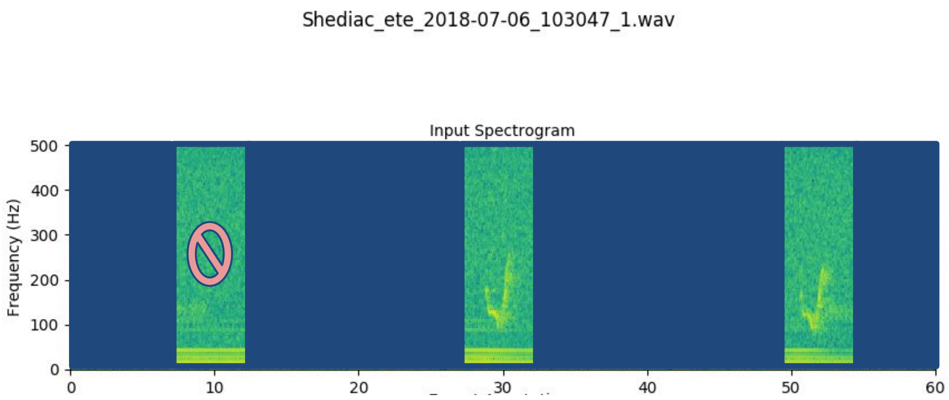
Annotation levels and compatibility

Expert validation



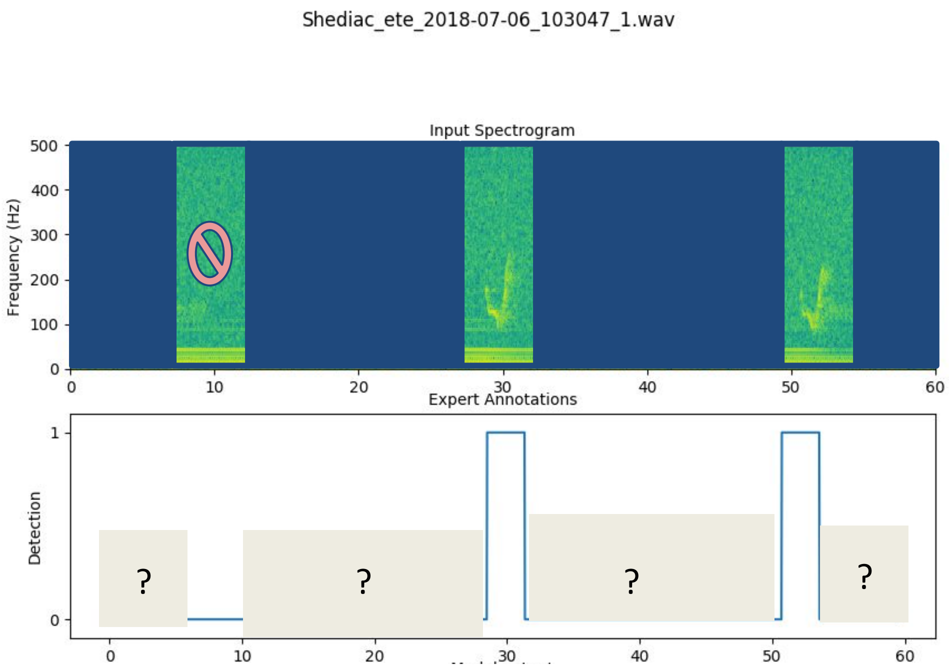
Annotation levels and compatibility

Expert validation



Annotation levels and compatibility

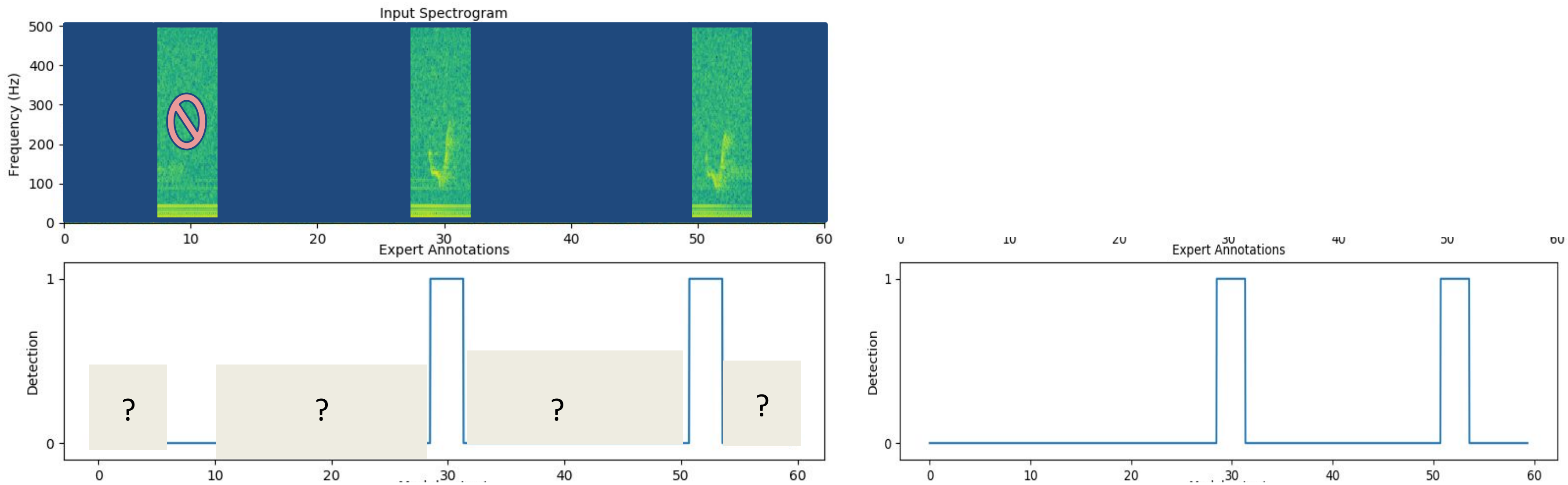
Expert validation



Annotation levels and compatibility

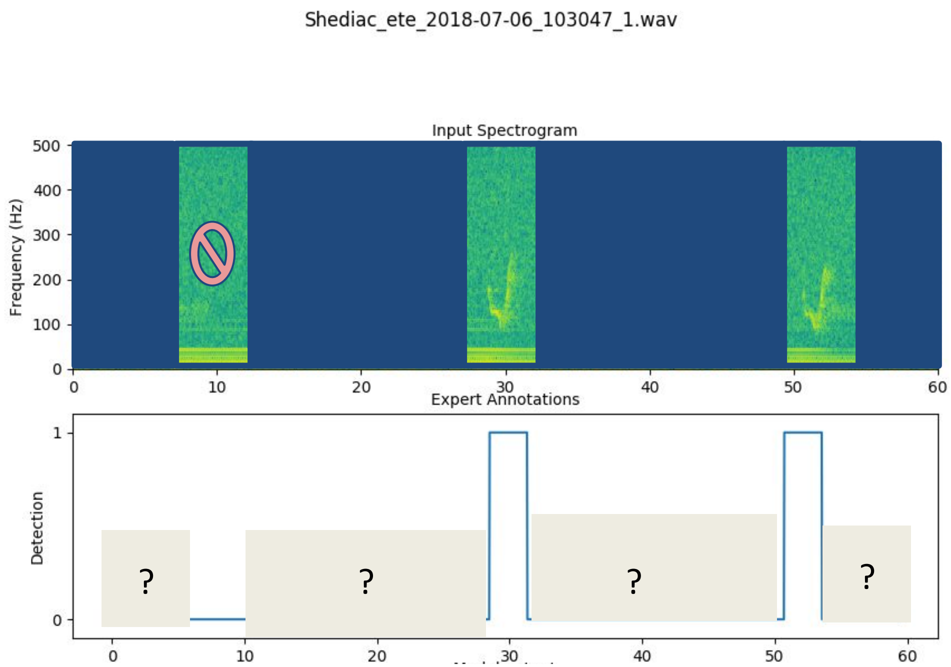
Expert validation

Shediac_ete_2018-07-06_103047_1.wav



Annotation levels and compatibility

Expert validation

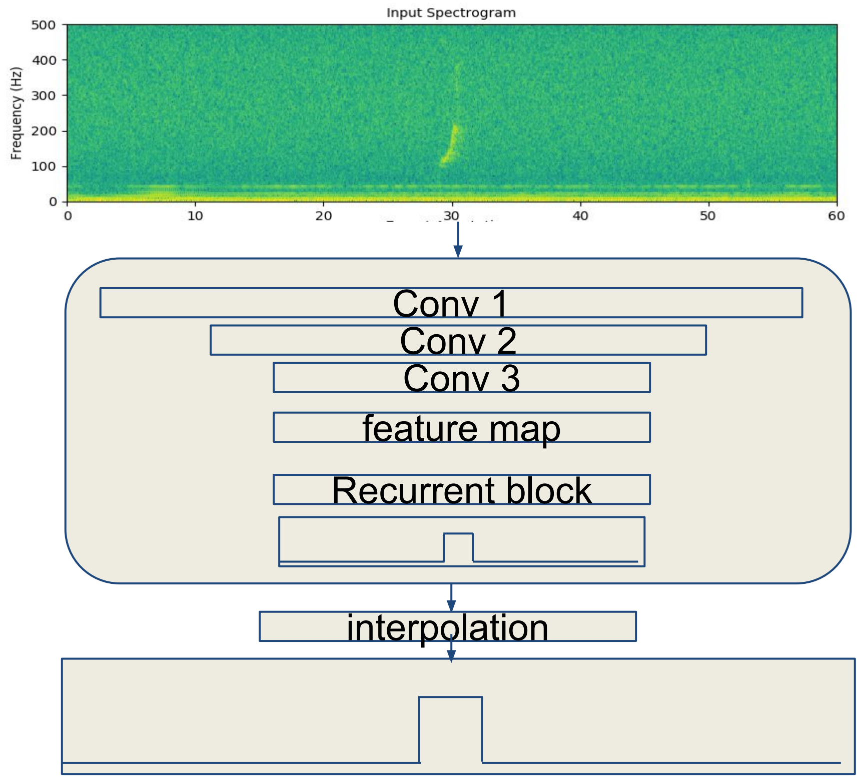


These annotations:

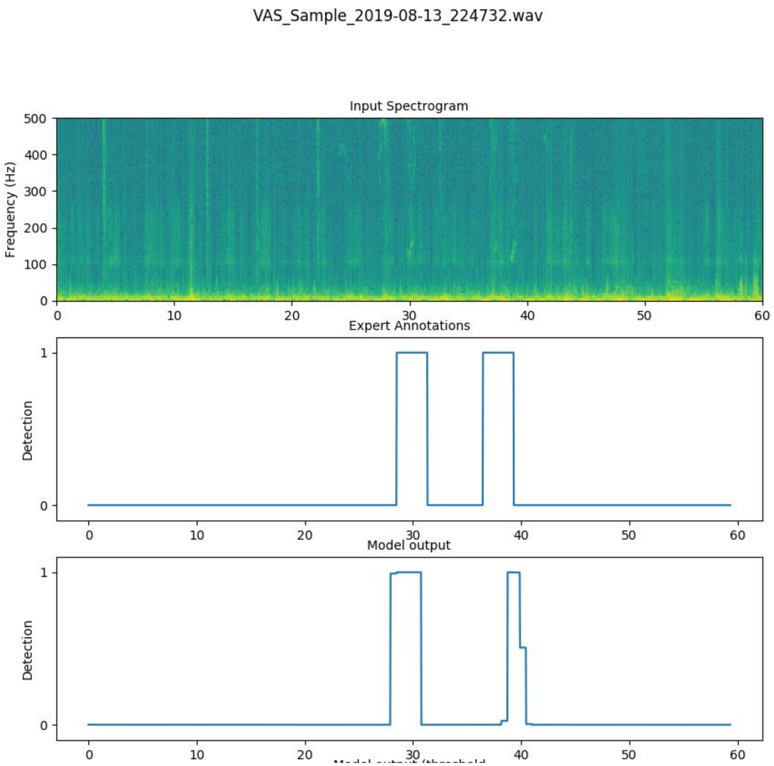
- Reduce the amount of data available
- Carry the biases/limitations from the first detector
- Reduce the feature space for the model, often presenting an unrealistic scenario

Annotation levels and compatibility

A different kind of model: sequence to sequence (Recurrent Convolutional Network)

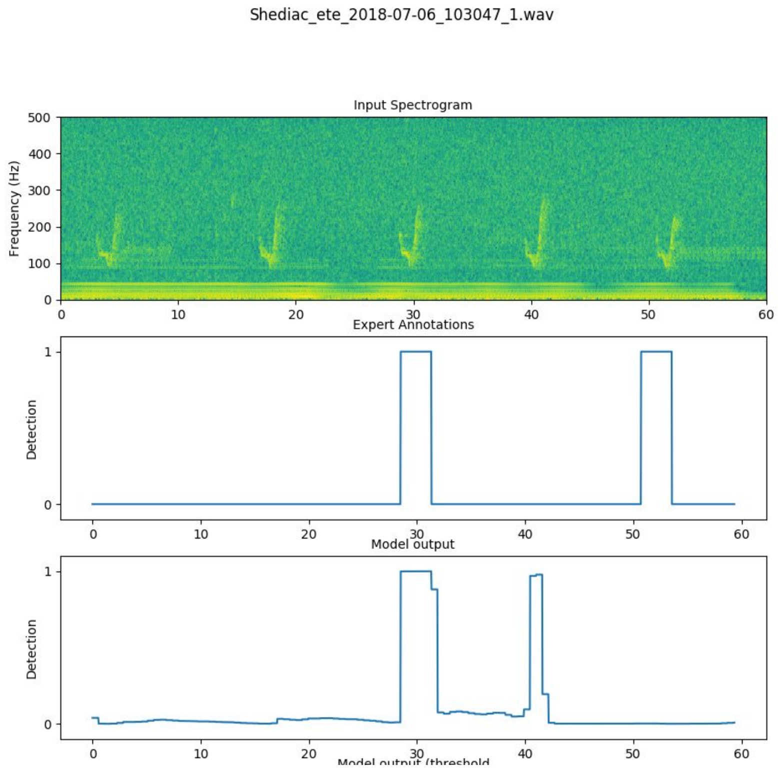


Annotation levels and compatibility



Annotation levels and compatibility

Annotations not well-suited for this kind of model



Well-documented metrics,
methods and models

Well-documented metrics, methods and models

"Explicit is better than implicit" - from the zen of python

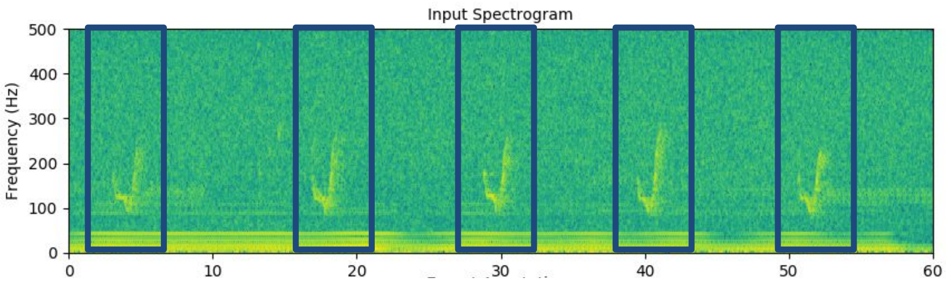
Well-documented metrics

Well-documented metrics, methods and models

For example, common metrics might differ depending on the criteria used to establish a true positive

$$\text{Precision} = \frac{\text{True Positives}}{\text{Total } \textit{Predicted} \text{ Positives}}$$

Shediac_ete_2018-07-06_103047_1.wav



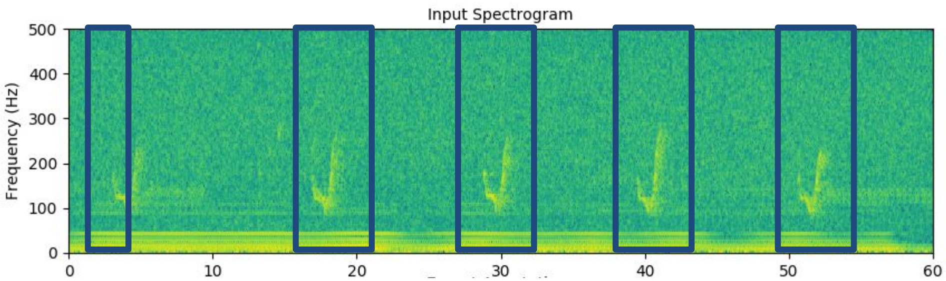
Precision= 5/5 =1

Well-documented metrics, methods and models

For example, common metrics might differ depending on the criteria used to establish a true positive

$$\text{Precision} = \frac{\text{True Positives}}{\text{Total } \textit{Predicted} \text{ Positives}}$$

Shediac_ete_2018-07-06_103047_1.wav



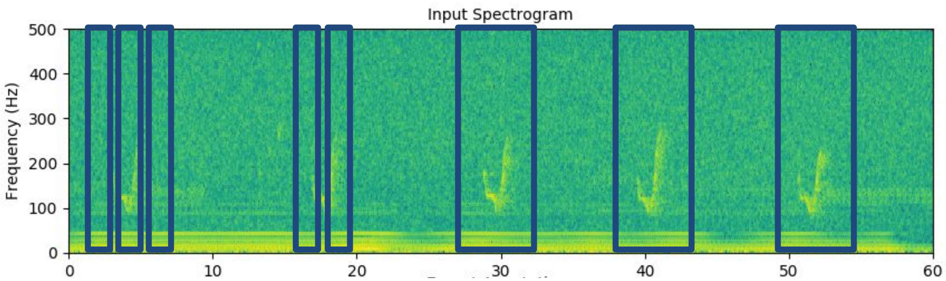
Precision= 5/5 =1
or
Precision= 4/5 = 0.8?

Well-documented metrics, methods and models

For example, common metrics might differ depending on the criteria used to establish a true positive

$$\text{Precision} = \frac{\text{True Positives}}{\text{Total } \textit{Predicted} \text{ Positives}}$$

Shediac_ete_2018-07-06_103047_1.wav



Precision= ?/8 =?

Well-documented methods
(for pre and post processing)

Well-documented metrics, methods and models

A good example of well-documented signal processing methods:

OSmOSE report *No* 1

Theory-plus-code documentation of the DEPAM workflow for soundscape description

Dorian Cazau, Paul Nguyen (2019) arxiv.org/pdf/1902.06659.pdf

Well-documented metrics, methods and models

MERIDIAN

4.1 PSD (Power Spectral density)

4.1.1 Theory

The Discrete Fourier Transform (DFT) of the m^{th} segment $X^{(m)}(f)$ is given by

$$X^{(m)}(f) = \sum_{n=0}^{N-1} xin_{win}^{(m)}[n]e^{\frac{-i2\pi fn}{N}} \tag{4.1}$$

The power spectrum is computed from the DFT, and corresponds to the square of the amplitude spectrum (DFT divided by N), which for the m^{th} segment is given by

$$P^{(m)}(f) = |\frac{X^{(m)}(f)}{N}|^2 \tag{4.2}$$

where $P^{(m)}(f)$ stands for the power spectrum. For real sampled signals, the power spectrum is symmetrical around the Nyquist frequency, $Fs/2$, which is the highest frequency which can be measured for a given Fs. The frequencies above $Fs/2$ can therefore be discarded and the power in the remaining frequency bins are doubled, yielding the single-sided power spectrum

$$P^{(m)}(f') = 2.P^{(m)}(f') \tag{4.3}$$

where $0 < f' < fs/2$. This correction ensures that the amount of energy in the power spectrum is equivalent to the amount of energy (in this case the sum of the squared pressure) in the time series. This method of scaling, known as Parseval's theorem, ensures that measurements in the frequency and time domain are comparable. The power spectral density PSD (also called mean-square sound-pressure spectral density) is defined by:

$$PSD(f',m) = \frac{P^{(m)}(f')}{B\Delta f} \quad [\mu\text{Pa}^2\ /\text{Hz}] \tag{4.4}$$

where $\Delta f = fs/2N$ is the width of the frequency bins, and B is the noise power bandwidth of the window function, which corrects for the energy added through spectral leakage:

$$B = \frac{1}{N}\sum_{n=0}^{N-1}(\frac{w[n]}{\alpha})^2 \tag{4.5}$$

Note that a spectral density is any quantity expressed as a contribution per unit of bandwidth. A spectral density level is ten times the logarithm to the base 10 of the ratio of the spectral density of a quantity per unit bandwidth, to a reference value. Here the power spectral density level would be expressed in units of dB re 1 μPa^2 /Hz.

Discussion This section has been integrally drawn from (Merchant et al., 2015, Supplementary Material) without any modifications.

4.1.2 Matlab code

Correspondences with theory Eq. 4.1 is performed at lines 6-7. Eq. 4.2 is performed at lines 8. Eq. 4.3 is performed at lines 9.

```
1  if (mod(nfft, 2) == 0)
2    spectrumSize = nfft/2 + 1;
3  else
4      spectrumSize = nfft/2;
5  end
6  twoSidedSpectrum = fft(windowedSignal, nfft);
7  oneSidedSpectrum = twoSidedSpectrum(1 : spectrumSize, :);
8  powerSpectrum = abs(oneSidedSpectrum) .^ 2;
9  powerSpectrum(2 : spectrumSize-1, :) = powerSpectrum(2 : spectrumSize-1, :) .* 2;
10 psdNormFactor = 1.0 / (fs * sum(windowFunction .^ 2));
11 powerSpectralDensity = powerSpectrum * psdNormFactor;
12 welch = mean(powerSpectralDensity, 2);
```

Discussion Drawn from the function pwelch.m in Matlab 2014a.

4.1.3 Python code

Correspondences with theory Eq. 4.1 is performed at lines 1-3. Eq. 4.2 is performed at lines 4-7. Eq. 4.3 is performed at lines 8-13. Eq. 4.4 is performed at lines 14-16.

```
1 rawFFT = np.fft.rfft(windowedSignal, nfft)
2 vFFT = rawFFT * np.sqrt(1.0 / windowFunction.sum() ** 2)
3 periodograms = np.abs(rawFFT) ** 2
4 vPSD = periodograms / (fs * (windowFunction ** 2).sum())
5 vWelch = np.mean(vPSD, axis=0)
```

Discussion Adapted from the function spectrogram in scipy, with modifications only done to make this code suitable for our variable names.

Well-documented models

Well-documented metrics, methods and models

"What exactly do you mean by ResNet50?"

-What is described in He et al 2015
-That, but with a different input size.
-Whatever I get with

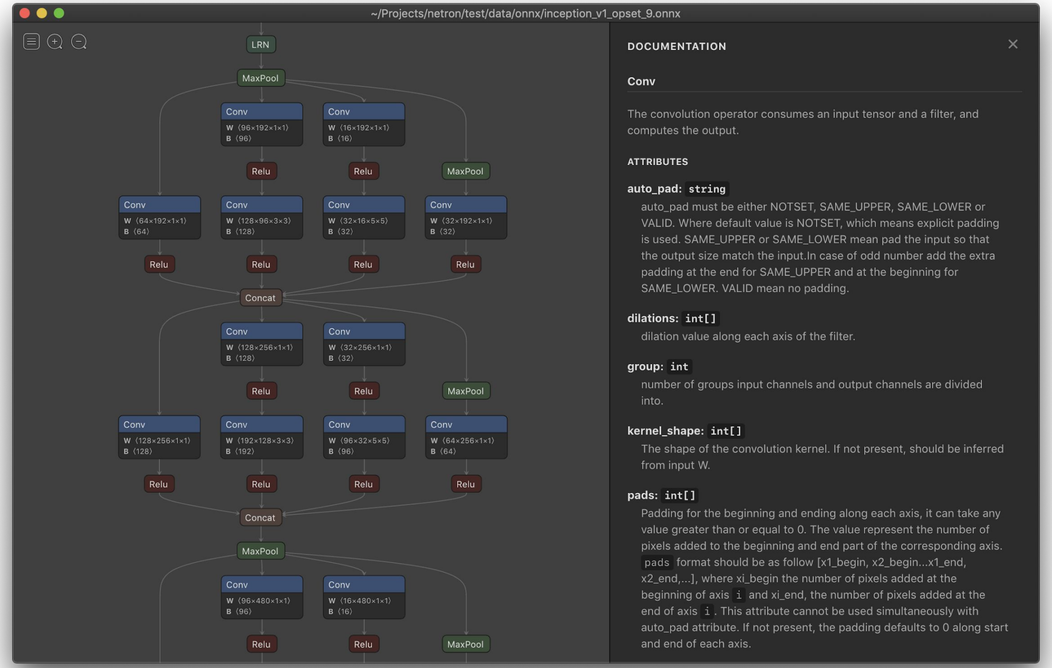
```
import torch
model = torch.hub.load('pytorch/vision:v0.6.0', 'resnet50', pretrained=True)
```

-Some other variation that's still a residual network with 50 layers

Well-documented metrics, methods and models

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 128, 128, 3)	0	
conv2d (Conv2D)	(None, 128, 128, 32)	896	input_layer[0][0]
max_pooling2d (MaxPooling2D)	(None, 64, 64, 32)	0	conv2d[0][0]
conv2d_1 (Conv2D)	(None, 64, 64, 32)	9248	max_pooling2d[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 32)	0	conv2d_1[0][0]
flatten (Flatten)	(None, 32768)	0	max_pooling2d_1[0][0]
dense (Dense)	(None, 128)	4194432	flatten[0][0]
dropout (Dropout)	(None, 128)	0	dense[0][0]
dense_1 (Dense)	(None, 128)	16512	dropout[0][0]
dropout_1 (Dropout)	(None, 128)	0	dense_1[0][0]
weather (Dense)	(None, 4)	516	dropout_1[0][0]
ground (Dense)	(None, 13)	1677	dropout_1[0][0]

Total params: 4,223,281
Trainable params: 4,223,281
Non-trainable params: 0



For sharing models:

Open source software with adaptability, reusability and collaboration principles in mind

Open source software: adaptability, reusability and collaboration

- Well-written code is probably the most accurate documentation for methods, metrics and models

Open source software: adaptability, reusability and collaboration

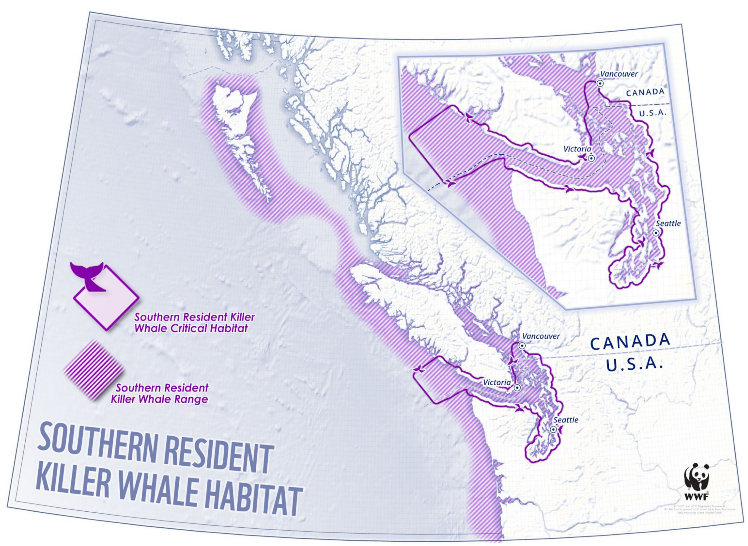
Sharing code for verification is good.
Others can reproduce the steps and get the same results.

Sharing code for adaptation is even better.
Others can not only reproduce the steps, but can reuse parts of the code and adapt it achieve things the original authors didn't even anticipate

Open source software: adaptability, reusability and collaboration

HALLO
Humans and ALgorithms Listening for Orcas

- Collaboration between SFU, DFO, Dalhousie, Carleton and OrcaSound
- Aim to produce effective models for detection and classification of whales (focus on southern resident orcas)
- Develop reusable and adaptable frameworks for collaborative annotations, data access and model adaptat



wwf.ca/species/southern-resident-killer-whales/

Key messages

- Share as much as possible (data, annotations, trained model, code, benchmark results)
- Document everything
- Build software that is adaptable and promote collaboration

Thank you!