Segmentation, Classification, and Visualization of Orca Calls using Deep Learning

Hendrik Schröter, Elmar Nöth, Andreas Maier, Rachael Cheng, Volker Barth, **Christian Bergler**
Pattern Recognition Lab, Friedrich-Alexander University Erlangen-Nürnberg
IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)
May 12th – 17th 2019, Brighton, United Kingdom (UK)

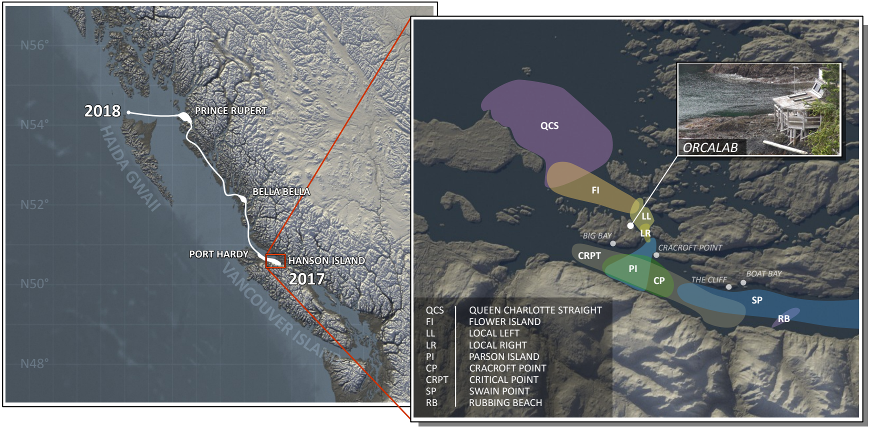
Motivation: Killer whale research

The Killer Whale *(Orcinus orca)* [1]

Motivation: Killer whale research

OrcaLab [2]

Motivation: Killer whale research



Covered recording area by the DeepAL [1] expedition and the fixed installed OrcaLab [2] hydrophones

Motivation: Killer whale research

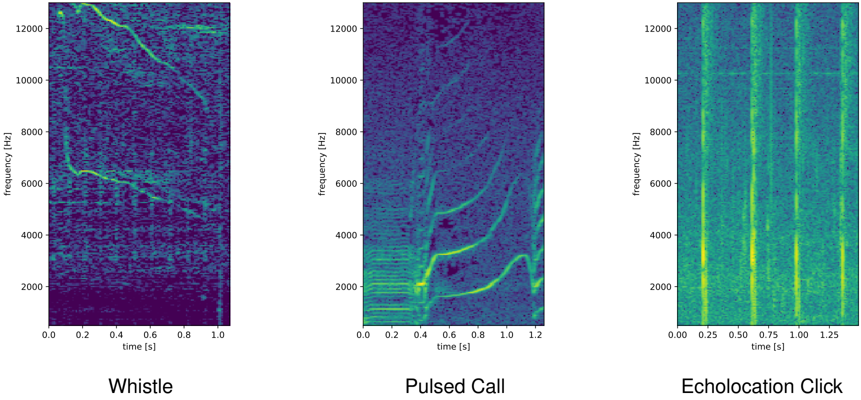
The Orchive [3]

- collected by the OrcaLab [2] and Steven Ness [3]
- 20,000 hours of underwater recordings by using 6 stationary hydrophones (1985–2010)
- 23,511 digitized audio tapes each $\sim$45 min.
- Orchive Annotation Catalog (OAC) [3] comprises 15,480 orca/noise labels

DeepAL Fieldwork Data (DLFD) 2017/2018 [1]

- collected via a 15-meter research trimaran
- 1,007 hours of multi-channel underwater recordings
- 89 hours video footage about behavioral data

Example killer whale vocalizations



Spectrograms from three characteristic killer whale sounds.

Outline

Data Corpora and Preprocessing

Data Corpora – Orca/Noise Segmentation

Corpora

dataset \ split		training		validation		test	
		samples	% orca	samples	% orca	samples	% orca
OAC[1]	**11,504**	8,042	**84.9**	1,711	83.3	1,751	82.4
AEOTD[2]	**17,995**	14,424	8.9	1,787	15.4	1,784	5.7
DLFD[3]	**31,928**	23,891	14.2	4,125	30.1	3,912	28.3
SUM	**61,427**	46,357	**24.8**	7,623	38.6	7,447	35.6

[1] Orchive Annotation Catalog (OAC) [2]

[2] Automatic Extracted Orchive tape data (AEOTD) [3]

[3] DeepAL Fieldwork Data (DLFD) [1]

Data Corpora – Call Type Classification

Corpora

dataset \ split		training samples	training %	validation samples	validation %	test samples	test %
CCS[1]	**138**	102	73.9	19	13.8	17	12.3
CCN[2]	**286**	198	69.2	41	14.4	47	16.4
EXT[3]	**90**	63	70.0	12	13.3	15	16.7
SUM	**514**	363	70.6	72	14.0	79	15.4

[1] Call Catalog Symonds (CCS) [2]
[2] Call Catalog Ness (CCS) [3]
[3] Orchive Extension Catalog (EXT)

Data Preprocessing

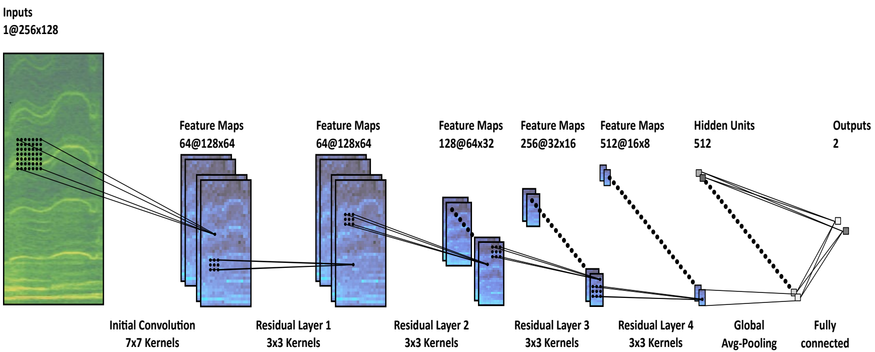
Preprocessing and Augmentation

- Power-Spectrogram
- Augmentation
  - Amplitude scaling
  - Frequency shift
  - Time stretch
  - Addition of noise spectrograms
  - Trimming / Padding to fixed length
- dB-Normalization

Segmentation – Network Architecture, Training, and Results

Network Architecture and Training

Architecture

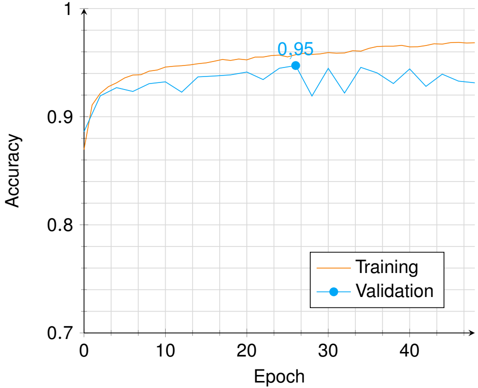


ResNet18-based Convolutional Neural Network (CNN) without max-pooling in the first residual layer for a binary classification problem

Network Results

Results

- **Test accuracy** of **95.0 %** (TPR = 93.8 %, FPR = 4.3 %)

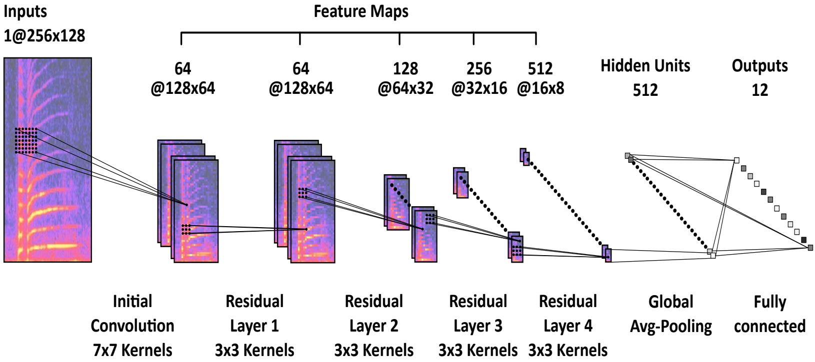


Training and validation accuracy of the segmentation model.

Call Type Classification – Network Architecture, Training, and Results

Network Architecture and Training

Architecture

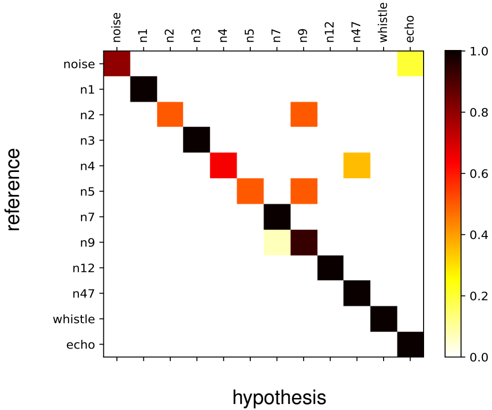


ResNet18-based Convolutional Neural Network (CNN) without max-pooling in the first residual layer for a 12-class problem

Network Results

Results

- **Mean test accuracy** of **87.0 %**

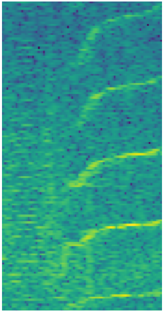


Confusion matrix from the call type classifier.

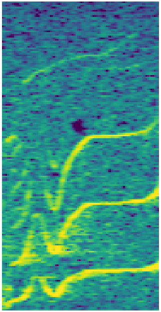
Network Results

Misclassifications

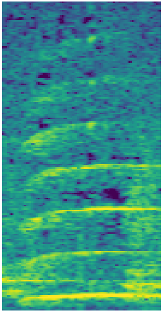
Reference — **Wrong predictions** —



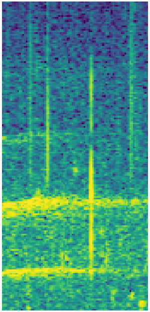
N9



N2 as N9



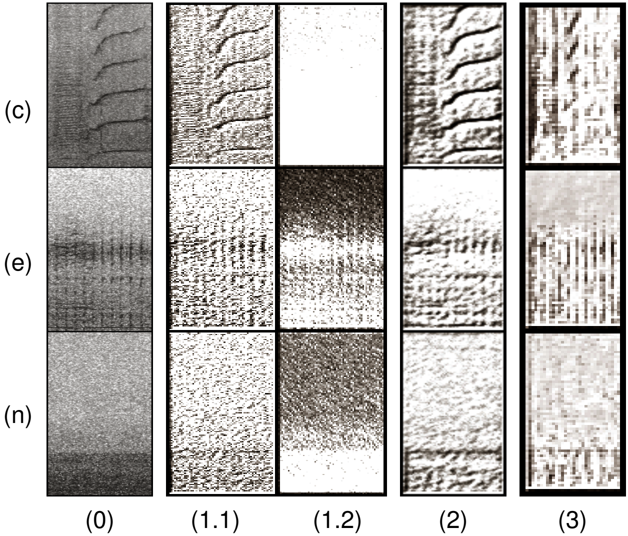
N5 as N9



N9 as N7

Visualization – Call Type Features

Call Type Feature Visualization



Conclusion

Conclusion

- Two-stage approach for robust segmentation and classification
- Applicable on any semi-labeled database
- Real-time factor of 1/25 (NVIDIA GTX 1050) enables on-the-fly detection in the field
- Automatically segment large data corpora followed by a subsequent call type classification
- Direct comparison to other work is difficult (different data corpora and/or approaches) (Steven Ness [3])
- Training call type classifier with only few call type labels
- Increase training data to be more robust against signal variety of real-world data

Thank you for your attention.

Questions?

References I

[1] C. Bergler, *Deepal fieldwork data 2017/2018 (dlfd)*, `https://www5.cs.fau.de/research/data/` (April 2019).

[2] ORCALAB, *Orcalab - a whale research station on hanson island*, http://orcalab.org (September 2018).

[3] S. Ness, “The orchive : a system for semi-automatic annotation and analysis of a large collection of bioacoustic recordings”, PhD thesis (Department of Computer Science, University of Victoria, 3800 Finnerty Road, Victoria, British Columbia, Canada, V8P 5C2, 2013), p. 228.

[4] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch”, in Nips 2017 workshop (2017).

Data Distribution

Call Type Label Distribution

Orca Call Type/ Corpus	N01	N02	N03	N04	N05	N07	N09	N12	N47	echo	whistles	noise	SUM
CCS [2]	33	10	—	21	14	18	26	16	—	—	—	—	**138**
CCN [3]	36	—	56	60	—	31	70	—	33	—	—	—	**286**
EXT	—	—	—	—	—	—	—	—	—	30	30	30	**90**
SUM	**69**	**10**	**56**	**81**	**14**	**49**	**96**	**16**	**33**	**30**	**30**	**30**	**514**

Orca call type, echolocation, whistle, and noise label distribution of the CCS, CCN, and EXT data corpus

Data Preprocessing

Preprocessing and Augmentation

Data: Training Input Audio $\mathcal{A}_{inp}$

Result: Trainable Spectrogram $\mathcal{S}_{train}$

$\mathcal{S}_{inp} \leftarrow 10 \cdot log_{10}(|\mathcal{FFT}(resamp(mono(\mathcal{A}_{inp}), 44.1\,\mathrm{kHz}), \mathrm{ffts} = 4096, \mathrm{hop} = 441)|^2)$

$\mathcal{S}_{train} \leftarrow scaleAmplitude(\mathcal{S}_{inp}, \alpha_{dB} = sample([-6\,\mathrm{dB}, 3\,\mathrm{dB}]))$

$\mathcal{S}_{train} \leftarrow shiftPitch(\mathcal{S}_{train}, \alpha = sample([0.5, 1.5]))$

$\mathcal{S}_{train} \leftarrow stretchTime(\mathcal{S}_{train}, \alpha = sample([0.5, 2]))$

$\mathcal{S}_{train} \leftarrow compressFrequencies(\mathcal{S}_{train}, \mathrm{f}_{min} = 500\mathrm{Hz}, \mathrm{f}_{max} = 10\,000\,\mathrm{Hz}, \mathrm{bins} = 256)$

$\mathcal{S}_{train} \leftarrow addNoise(\mathcal{S}_{train}, sample(\mathcal{S}_{noise}), \mathrm{SNR} = sample([12\,\mathrm{dB}, -3\,\mathrm{dB}]))$

$\mathcal{S}_{train} \leftarrow normalize(\mathcal{S}_{train}, \mathrm{dB}_{min} = -100\,\mathrm{dB}, \mathrm{dB}_{ref} = 20\,\mathrm{dB})$

$\mathcal{S}_{train} \leftarrow trimPad(\mathcal{S}_{train}, \mathrm{length} = sample(128))$

return $\mathcal{S}_{train}$

Segmentation Model – Network Training

Training

- implemented and trained using PyTorch [4]
- Adam optimizer ($\mathrm{lr}_{init} = 10^{-5}$, $\beta_1 = 0.5$, $\beta_2 = 0.999$)
- learning rate decayed by a factor of 0.5 if there was no improvement on the validation accuracy for 4 epochs
- training stopped if there was no improvement on the validation accuracy for 10 epochs
- batch size = 32

Classification Model – Network Training

Training

- implemented and trained using PyTorch [4]
- Adam optimizer ($lr_{init} = 10^{-5}$, $\beta_1 = 0.5$, $\beta_2 = 0.999$)
- learning rate decayed by a factor of 0.5 if there was no improvement on the validation accuracy for 4 epochs
- training stopped if there was no improvement on the validation accuracy for 10 epochs
- batch size $= 4$

Comparison with previous work: Segmentation

Name	Segment. type	Dataset size	Accuracy	AUC
Ness [3]	Orca	11 041	92.12 %	–
Ours	Orca	61 427	94.97 %	98.17%

Comparison with previous work: Classification

Ness [3]

- Classification of 12 pulsed calls
- Mean accuracy of 76 %
- Per class accuracies between 60 % to 92 %

Ours

- Classification of 9 pulsed calls, whistle, echolocation and noise
- Mean test accuracy of 87 %
- Per class accuracy between 50 % to 100 %