Multimodal deep learning for Cetacean Distribution Modeling.

Case study of fin whales (*Balaenoptera physalus*) in the western Mediterranean Sea.

Cazau, D. (1) , Nguyen, P. (2) , Fablet, R. (1)

(1) Lab-STICC, Institut Mines Télécom Atlantique
(2) Institut d'Alembert, Sorbonne University

Context

Cetacean Distribution Modeling (CDM)

- **Use**: prediction tool for conservation biogeography and environmental change forecasts [1], with applications in ocean management, preservation of rare and/or endangered species, measurement of climate change on species;
- **Principle**: CDM models are driven by the idea that core biological activities of whales are strongly related to combined biogeochemical/hydrodynamical covariates distributed over the area;
- **Method**: inferring the spatial distribution (abundance or a presence index) of a cetacean species based on a matrix of ocean variables (e.g. temperature, chlorophyll-a) at a given location and time period (see example on the right graph of Fig. 1).

Project tech4whales

- **Long-term project objective**: proposing a global multimodal observational framework for CDM (left graph of Fig. 1) that would be able to predict whale occurrence whatever its behavior (only feeding habitat currently, see right graph of Fig. 1);
- **Core ideas**: using multimodal deep learning frameworks to leverage the potential of Big Ocean Data for CDM.

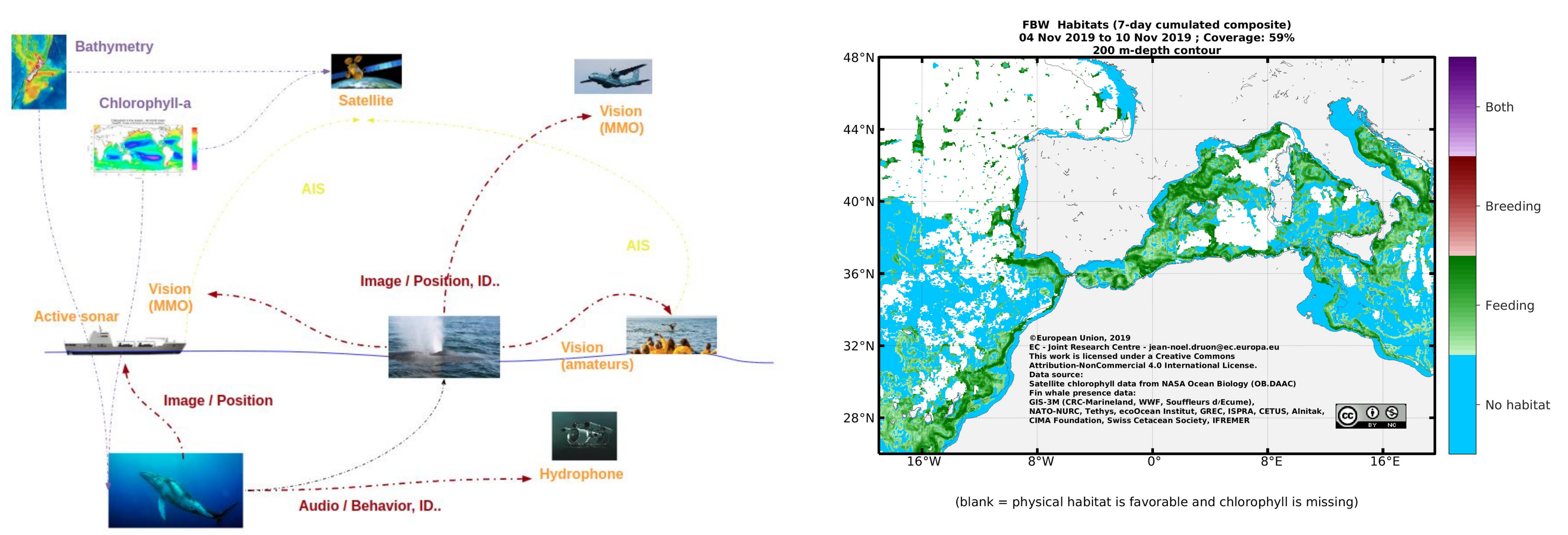


Fig. 1: On the left, schematic view of a multimodal observational framework of whale occurrences. On the right, example of a Feeding Habitat Occurrence (FHO) map for fin whales (*Balaenoptera physalus*) using [2]'s model.

Study objectives

- **Exploring** different data-driven learning schemes to capture generic environmental patterns revealing fin whale occurrence in the Mediterranean Sea during the feeding/foraging season;
- **Quantifying** performance of these models to predict fin whale occurrence data w.r.t the Feeding Habitat Occurrence (FHO) model [2], considered both as a state-of-the-art baseline and as a knowledge base for model training.

Models

- **FHO** (from [2]): this model provides a daily unitary index of the fin whale's preferred feeding habitat based on the distribution of horizontal chlorophyll-a gradients (orange histogram in Fig. 2) at the species' locations. It follows the expert rule that "feeding habitat is mostly related to the occurrence of chlorophyll-a fronts that are detected by satellite sensors of ocean colour";

$$\tilde{I} = [(\mathbb{C} > CHL_{min}) AND (\mathbb{C} < CHL_{max}) \\ AND (\Delta\mathbb{C} > gradCHL_{min}) AND (\mathbb{B} > minB)] \\ \times (gradCHL_{int} \times \Delta\mathbb{C}) OR (\Delta\mathbb{C} > gradCHL_{int})$$

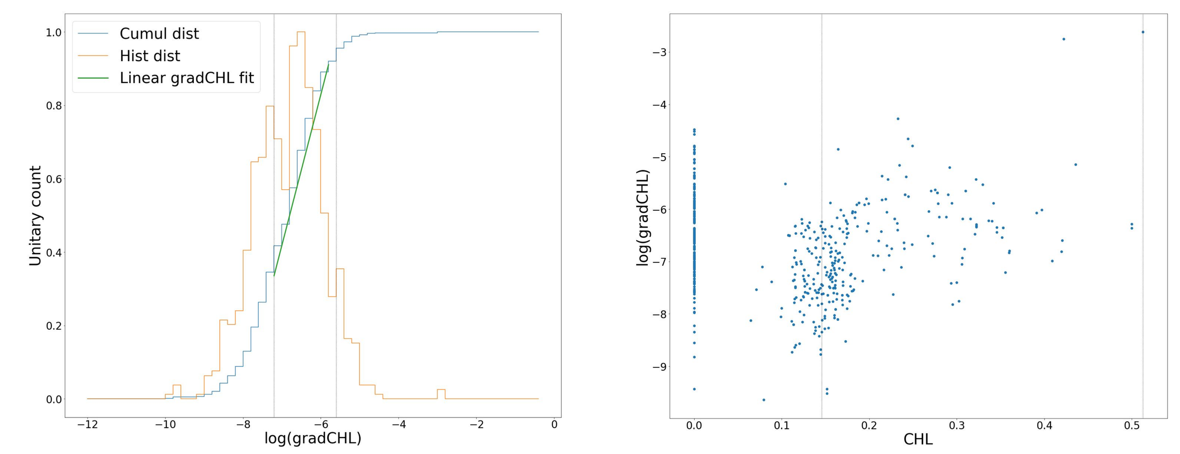


Fig.2 : On top, equation of the FHO model stated as a logical network. On the right, histograms and scattering plots used to estimate numerical values of the FHO model parameters (i.e., min and max of CHL and gradCHL).

- **FHO_MLP_SP**: this first model is a simple Multi Layer Perceptron that performs a regression task to predict a single pixel (i.e. central pixel of 24x24 pixel patches) of FHO values. It consists of a simple stack of 2 fully connected layers with relu activations, each having 16 hidden units, and followed by a third single-unit layer with a sigmoid activation as FHO values are unitary. Mean-squared error (mse) is used as a loss function;
- **FHO_AECNN_IP24**: this model is a deep autoencoder-like convolutional network that performs a multiple regression task on a 24x24 pixel to predict FHO input patches. As detailed in Fig. 3, it consists of a stack of alternated 2D convolutional layers (Conv2D) with relu activations and max pooling layers, as well as dropout layers used for regularization and residual connections to favor multi-scale representations. Mean absolute error (mae) is used as a loss function;
- **FWO_MLP_SP**: this model performs a binary classification task to predict a FWO probability. It uses the exact same MLP architecture as FHO_MLP_SP except that now binary cross-entropy is used as a loss function;
- **FHO-FWO_AECNN-MLP_IP24**: this model is a case of transfer learning where we used FHO_CNNAE_IP24 as a convolutional base pre-trained on a FHO regression task to perform the binary classification task on FWO data. In this model, we first added a fully connected classifier on top of FHO_AECNN_IP24 after flattening its output. Second, we fine tuned the last convolutional block (conv2D_9) of FHO_CNNAE_IP24, i.e. jointly training the new added classifier and this block while freezing the other layers.

All models have been developed using Keras functional API and Tensorflow backend [3]. Adam optimizer has been used for all models.

```
Layer (type)                     Output Shape          Param #     Connected to
==================================================================================================
input_1 (InputLayer)             (None, 24, 24, 3)     0
__________________________________________________________________________________________________
conv2d_1 (Conv2D)                (None, 24, 24, 16)    448         input_1[0][0]
__________________________________________________________________________________________________
conv2d_2 (Conv2D)                (None, 24, 24, 16)    2320        conv2d_1[0][0]
__________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)   (None, 12, 12, 16)    0           conv2d_2[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)              (None, 12, 12, 16)    0           max_pooling2d_1[0][0]
__________________________________________________________________________________________________
conv2d_3 (Conv2D)                (None, 12, 12, 32)    4640        dropout_1[0][0]
__________________________________________________________________________________________________
conv2d_4 (Conv2D)                (None, 12, 12, 32)    9248        conv2d_3[0][0]
__________________________________________________________________________________________________
max_pooling2d_2 (MaxPooling2D)   (None, 6, 6, 32)      0           conv2d_4[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)              (None, 6, 6, 32)      0           max_pooling2d_2[0][0]
__________________________________________________________________________________________________
conv2d_5 (Conv2D)                (None, 6, 6, 64)      18496       dropout_2[0][0]
__________________________________________________________________________________________________
conv2d_6 (Conv2D)                (None, 6, 6, 64)      36928       conv2d_5[0][0]
__________________________________________________________________________________________________
conv2d_transpose_1 (Conv2DTrans (None, 12, 12, 32)    18464       conv2d_6[0][0]
__________________________________________________________________________________________________
dropout_3 (Dropout)              (None, 12, 12, 32)    0           conv2d_transpose_1[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)      (None, 12, 12, 64)    0           conv2d_4[0][0]
                                                                   dropout_3[0][0]
__________________________________________________________________________________________________
conv2d_7 (Conv2D)                (None, 12, 12, 32)    18464       concatenate_1[0][0]
__________________________________________________________________________________________________
conv2d_8 (Conv2D)                (None, 12, 12, 32)    9248        conv2d_7[0][0]
__________________________________________________________________________________________________
conv2d_transpose_2 (Conv2DTrans (None, 24, 24, 16)    4624        conv2d_8[0][0]
__________________________________________________________________________________________________
dropout_4 (Dropout)              (None, 24, 24, 16)    0           conv2d_transpose_2[0][0]
__________________________________________________________________________________________________
concatenate_2 (Concatenate)      (None, 24, 24, 32)    0           conv2d_2[0][0]
                                                                   dropout_4[0][0]
__________________________________________________________________________________________________
conv2d_9 (Conv2D)                (None, 24, 24, 1)     289         concatenate_2[0][0]
==================================================================================================
Total params: 123,169
Trainable params: 123,169
Non-trainable params: 0
__________________________________________________________________________________________________
```

Fig. 3: Model summary of FHO_AECNN_IP24

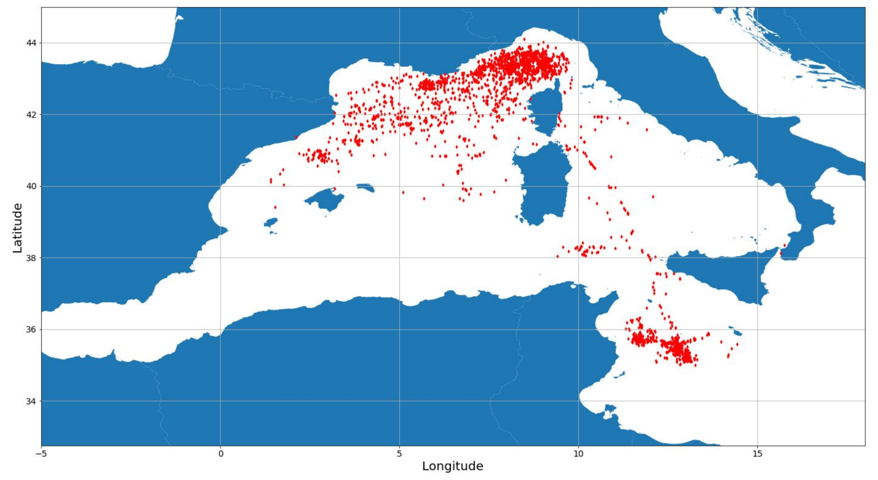
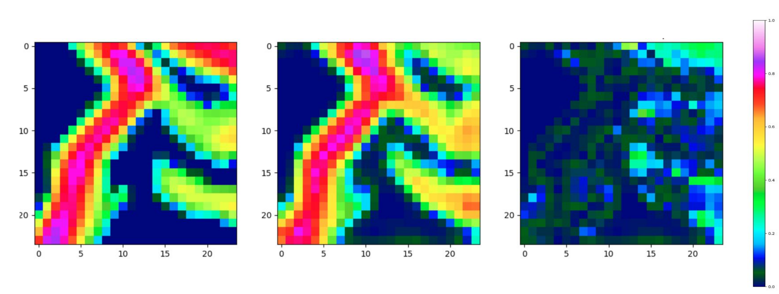


Fig. 4: Geographical locations of fin whale occurrence data, represented as red dots.

Experiments

- **Data**
  - **Fin Whale Occurrence (FWO, n = 2415)**: cetacean sightings (from marine mammal observers during shipboard and aerial surveys) and e-tagged whales;
  - **Pseudo-absence data (n = 2415)**: generated through random selection of points from the whole study area except from the occurrence localities of FWO;
  - **Environmental Context (n = 10,000)**: geographic rasters of bathymetry (spatial resolution of 1 arc-minute, from GEBCO), chlorophyll-a content and gradient (daily images with spatial resolution 4.6 km from 2013 to 2018, from MODIS-Aqua);
  - **Knowledge base (FHO, n = 10,000)**: feeding habitat maps from [2];
- **Training, test and evaluation procedure**
  - **FHO_* models for regression** : 8000 maps for training / 2000 for validation / 2000 for test;
  - ***FWO_* models for binary classification** : 1449 for training / 483 for validation / 483 for test, using balanced training and validation sets for the two classes (i.e. FWO and pseudo-absence), but a test set built only with FWO. FHO_* models are also used for this task by applying a threshold optimized during the validation phase;
  - **Iterated 5-folds cross-validation** have been performed, using as evaluation metrics the **pixel-wise rmse** for regression task and the **number of missed events** for the binary classification (only unbiased evaluation metrics with presence-only data).
- **Results**
  - **Regression on FHO**: FHO_MLP_SP achieves a rmse accuracy of 0.11 in reproducing the unitary FHO maps, while FHO_AECNN_IP24 is highly accurate, with a rmse below 0.02 (illustrated in Fig. 5);
  - **Prediction of FWO probability**: unsurprisingly, FHO model has the worst performance in predicting FWO, with a number of missed events around 278 (over 483). This number is considerably decreased by the FWO_MLP_SP and FHO-FWO_AECNN-MLP_IP24 models, with respectively 101 and 93 missed events on average for central pixels.



Models	Central pixel	Median patch 5×5
FHO	278 (± 29)	267 (± 11)
FHO_MLP_SP	234 (± 22)	243 (± 15)
FHO_AECNN_IP24	249 (± 34)	211 (± 46)
FWO_MLP_SP	101 (± 14)	90 (± 9)
FHO-FWO_AECNN-MLP_IP24	93 (± 21)	78 (± 17)

Fig. 5: On the left, the graphs from left to right represent the original FHO map, the predicted one with FHO_AECNN_IP24 and the pixel-wise rmse map. On the right, average number (and standard deviation) of missed events for the different models (over 483).

Wrapping up

- Although our FWO dataset has been mostly sampled in an area and time period favorable for fin whale feeding behavior, the knowledge base from FHO model [2] alone performs poorly in predicting no-learned FWO data, showing the weak generalization capacity of this rule-based model;
- We presented first results showing that data-driven deep representations with a joint training of FHO and FWO, and including larger contextual environmental data, could be a promising avenue to learn more generalizing environmental patterns associated with whale occurrence;
- In a near future, we wish to integrate new modalities such as passive acoustics data and explore other network architectures to model temporal dynamics (e.g. RNN) for our tasks.

[1] Franklin, J. (2010). Diversity and Distributions, 16, 321–330.
[2] Druon, J.N. (2017). Technical report JRC science for policy report.
[3] https://keras.io/ , https://www.tensorflow.org/
[4] Druon, J.N. (2019). Scientific Reports Nature.