# Deep Learning–Based Detection of *Nephrops norvegicus* Burrows

Oscar Papini*, Enrico Cecapolli†, Filippo Domenichetti†, Gabriele Pieri*,
Marco Reggiannini*, Lorenzo Zacchetti†,‡ and Michela Martinelli†
*Institute for Information Science and Technologies "A. Faedo" (ISTI)
National Research Council of Italy, Pisa, Italy
{oscar.papini, gabriele.pieri, marco.reggiannini}@isti.cnr.it
†Institute for Marine Biological Resources and Biotechnology (IRBIM)
National Research Council of Italy, Ancona, Italy
{enrico.cecapolli,lorenzo.zacchetti}@irbim.cnr.it
{filippo.domenichetti,michela.martinelli}@cnr.it
‡Department of Biological, Geological, and Environmental Sciences
Alma Mater Studiorum – University of Bologna, Bologna, Italy

***Abstract*—This work presents a methodology to support the assessment of a benthic species of great commercial importance (*Nephrops norvegicus*) taking advantage of a combination of machine learning and computer vision methods. Up to the present, abundance indices based on the density of this species are evaluated by visual inspection of underwater imagery and through manual counting of the observed burrows. A novel approach is proposed, based on the integration in the processing pipeline of a supervised learning model in charge of detecting the burrows. The model is trained exploiting underwater videos that experts annotate by identifying the frames where burrows are present and specifying the related features. To pursue such a goal, the proposed automated procedure must cope with several environmental issues, such as high underwater turbidity, uneven illumination, heavy colour distortions, as well as complexities arising from the presence of ambiguous objects and morphological features that may affect the misclassification rate. The proposed method was developed on video material collected in a specific area, but has the potential to be applied throughout the species' distribution range. Preliminary results concerning the analysis of data captured in the central Adriatic Sea are presented and discussed.**

***Index Terms*—Deep learning, *Nephrops norvegicus*, Underwater Television**

## I. Introduction

*Nephrops norvegicus* (Norway lobster) is a burrowing decapod crustacean inhabiting the continental shelf of the north-eastern Atlantic Ocean and the Mediterranean Sea, typically at depths between 20 and 800 metres. This species plays a key role in European fisheries, with annual landings approaching 60 000 tonnes, however currently showing signs of decline in various regions [1].

Conventional stock assessment methods for *N. norvegicus* involve fishery-dependent sampling and underwater video cameras. In many European countries, towed Underwater Television (UWTV) surveys are the preferred method of monitoring this species. UWTV is particularly well-suited to *N. norvegicus*, as traditional fishery-dependent approaches are often inadequate due to the species' burrowing behaviour [2]. The UWTV methodology involves visual inspection of a defined seabed area to identify and count the characteristic burrows of *N. norvegicus*. Currently, this operation is performed manually by trained analysts reviewing hours of video footage [3], a time-consuming process prone to human error. These burrow counts, expressed as burrows per square metre, serve as a proxy for estimating population abundance [2]. In the Adriatic Sea (Central Mediterranean basin), UWTV surveys were conducted consistently from 2009 to 2019 in the Pomo Pits area, which represents a Vulnerable Marine Ecosystem and an Essential Fish Habitat for various commercially and ecologically important species [4], [5]. The analysis of the footage and the identification and quantification of *N. norvegicus* burrows were performed following the International Council for the Exploration of the Sea (ICES) guidelines [6].

Recent advancements in machine learning (ML) have already demonstrated promising potential for automating these tasks [6], [7], [8]. The main goal of this study is to leverage state-of-the-art machine learning and computer vision techniques to improve the assessment of *N. norvegicus* populations.

This paper is structured as follows. Section II outlines the process of creating an annotated dataset suitable to be used in the context of machine learning algorithms; Section III describes an example of a deep learning model that can exploit the dataset and presents the first promising results; Section IV concludes the paper with a small discussion on these results and future perspectives.

## II. Materials

### A. Annotation Guidelines

As mentioned above, the UWTV approach is currently based on the inspection of a known seafloor surface by trained scientists to count the number of *N. norvegicus* burrows, distinguishing them from other animals' burrows or artefacts. This procedure relies on the direct visual assessment of human

This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101000825 – NAUTILOS Project (https://www.nautilos-h2020.eu).

operators who are possibly influenced by subjectivity. Several workshops organised by ICES were held to standardise video reading methodologies and quantify related uncertainties [3]. Eventually, the population densities are returned as the average of the individual estimates performed by multiple readers. To reduce uncertainty, the ICES training is based on the usage of reference footage for each specific area, and on statistical tests to assess the concordance of the multiple readings. Lin's CCC test is commonly used to evaluate readers' performance during training and to determine whether they are ready to proceed with actual counting. It is also used during readings to determine the level of agreement between readers, whether certain minutes need to be revised, or even whether certain sets of readings should be excluded from the calculation of the average number of density per minute. Lin's CCC can also be used retrospectively to review time series if it has not been applied previously [9].

The introduction of a machine learning block in this task entails new peculiarities in the processing pipeline. In particular, the training stage of a supervised algorithm requires the availability of a conspicuous amount of classification examples. This implies the careful analysis of selected UWTV videos to identify and tag the frames where the presence of burrows is ascertained. Whenever detected, a burrow is annotated by an expert through manual specification of a minimum enclosing area, so as to enrich the annotation with details related to position and size. This represents a fundamental step that must be carried out by expert annotators with great precision and accuracy to ensure the achievement of a high-quality training dataset. In fact, the learning algorithm must be trained with certain and trustworthy information to avoid propagation of errors and suboptimal performance at inference time. Moreover, given the effort required to annotate large amounts of video material, annotators may also benefit from assisting tools, such as the one presented in [8], which automatically expands a starting set of annotations by means of a template matching algorithm.

To promote a correct approach in the annotation task, a list of good practices has been defined and summarised in [10].

### B. Annotation Tool

Among existing tools for annotating videos, a selection was made based on various aspects and suitability to the objective of being able to apply the aforementioned guidelines. The tool should be open source, allow for offline usage, and have robust tagging capabilities and an easy-to-use front-end. After extended research, the choice fell on *Computer Vision Annotation Tool* (CVAT) [11]. CVAT is an open-source tool with strong community support, providing both online and offline functionality. Although offline installation is slightly complex, its robust tagging capabilities make it a preferable choice for large-scale annotation tasks.

The expert users were able to use CVAT to perform the tagging activity and provide ground truth data connected with selected videos, which could then be processed by the deep learning algorithms to function as a decision-support tool in the higher-level burrow counting problem.

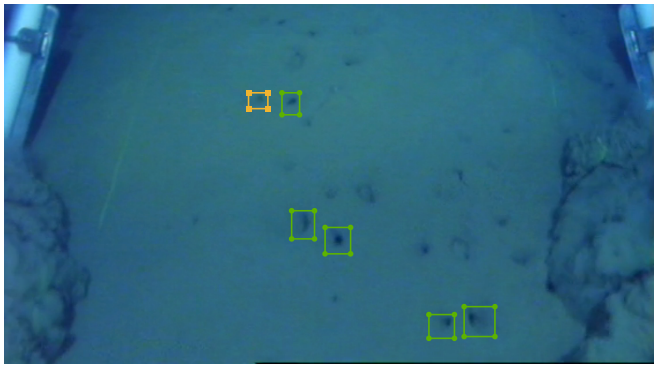

Figure 1. Examples of annotated objects. Green boxes with circle corners denote NEPHROPS objects; yellow boxes with square corners denote AMBIGUOUS objects.

### C. Dataset

To create a suitable dataset for the subsequent analysis, experts have reviewed video footage collected during UWTV surveys carried out in the Pomo Pits area, annotating with a bounding box the following objects (see Figure 1):

- openings in the sea floor that, in the experts' opinion, correspond to entrances of *N. norvegicus* burrows;
- openings in the sea floor that could be associated with such entrances, but lack some features that allow a certain attribution (see also [10]).

In the following, these two classes are endowed with the labels NEPHROPS and AMBIGUOUS respectively.

Several other objects in the surveyed scene may be mistaken for *N. norvegicus* entrances (other species' entrances, roundish shadows, etc.). These have been detected through an adaptive segmentation method and tagged with the additional label NON-NEPHROPS.

To ease the manual work of the annotators, videos have been reviewed and tagged at a frame rate of 2 FPS. At the time of writing, 1110 frames have been checked and annotated; the total number of objects is 557 NEPHROPS, 160 AMBIGUOUS and 1240 NON-NEPHROPS. In the rest of the paper this dataset will be referred to as "NephFrames". Due to strict regulations concerning both the preservation of fragile marine ecosystems and economic agreements concerning fishing, related sensitive information potentially contained in this dataset cannot be shared without the necessary preprocessing steps, e.g. data anonymization. In the future data and code will be released, provided that the mentioned procedures are applied.

## III. METHODS

In order to both explore the capabilities of an automatic deep learning framework that aims to detect and classify *N. norvegicus* burrows and assess the suitability of the NephFrames dataset illustrated in Section II-C, a custom deep neural network system has been defined and subsequently trained and evaluated. To keep things simple, the choice fell on a small architecture

Table I
TYPES OF LAYERS USED IN FIGS. 3 AND 4

| Name | Description |
|---|---|
| CONV2D | applies a 2-dimensional convolution operator, with $C$ output channels, $(k \times k)$-dimensional kernel, stride $s$ and padding $p$ in both dimensions |
| CONVTR2D | applies a 2-dimensional transposed convolution operator, with $C$ output channels, $(k \times k)$-dimensional kernel, stride $s$ and (input) padding $p$ in both dimensions |
| LINEAR | applies a linear (affine) transformation to an $n$-dimensional input to obtain an $m$-dimensional output, i.e. implements a map $x \mapsto Wx + b$ where $x \in \mathbb{R}^n$ and the trainable parameters $W$ and $b$ are an $(m \times n)$-matrix and a vector in $\mathbb{R}^m$ respectively |
| LEAKYRELU | applies the leaky rectified linear unit function, i.e. the map $x \mapsto \max\{0, x\} + 0.01 \min\{0, x\}$ for $x \in \mathbb{R}$, component-wise |
| SIGMOID | applies the sigmoid function, i.e. the map $x \mapsto 1/(1 + e^{-x})$ for $x \in \mathbb{R}$, component-wise |
| SOFTMAX | applies the softmax function to its input, i.e. the function $\mathbb{R}^n \to \mathbb{R}^n$ that maps $(x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_n)$ with $y_i = e^{x_i} / \sum_{j=1}^{n} e^{x_j}$ for $i = 1, \ldots, n$, so that the resulting vector satisfies $0 < y_i < 1$ for all $i = 1, \ldots, n$ and $\sum y_i = 1$ |
| FLATTEN | flattens its input, i.e. reshapes a multi-dimensional tensor into a 1-dimensional one by stacking its entries |
| UNFLATTEN | reshapes a 1-dimensional tensors rearranging its entries into the shape specified as output |
| UPSAMPLE | transforms an $h \times w$ tensor into an $nh \times nw$ one, replacing each value with an $n \times n$ block filled with that value |

with a few convolutional layers to extract meaningful features followed by a few linear (i.e., fully connected) layers to perform the actual classification task. Considering the available data, this approach poses two main issues:

1) it requires images that present a single label for the whole image, instead of labelled bounding boxes;
2) it requires a large amount of images to provide interesting and useful results.

To address the first issue, a new dataset has been derived from NephFrames. First, after applying a perspective transform to take into account the camera inclination, each frame has been divided in $60 \times 60$ px square tiles, with 30 px overlap; second, a label has been assigned to each tile by considering the set $\mathcal{L}$ of labels of the bounding boxes such that 70% of their area lies within the tile and checking the following conditions in this order:

1) if $\mathcal{L} = \emptyset$ (i.e. no bounding box satisfies the condition), then the tile has label BACKGROUND;
2) if NEPHROPS $\in \mathcal{L}$, then the tile has label NEPHROPS;
3) otherwise, if AMBIGUOUS $\in \mathcal{L}$, then the tile has label AMBIGUOUS;
4) otherwise (i.e. $\mathcal{L} = \{$NON-NEPHROPS$\}$), then the tile has label NON-NEPHROPS.

In the end, the new dataset, called "NephTiles" from now on, contains 231 990 tiles, divided in 1188 NEPHROPS, 368 AMBIGUOUS, 3777 NON-NEPHROPS and 226 657 BACKGROUND.

As far as the second issue is concerned, the chosen approach was to separate the feature extraction step from the classification step, and train those parts separately. In particular, the first step can be performed using a type of neural networks known as *autoencoders*, whose main task is to learn an efficient representation, or *encoding*, of data, usually belonging to a lower-dimensional domain (sometimes called *latent space*). The interested reader can find a good introduction on this topic, for example, in [12].

In summary, the overall training process can be split into two phases (see Figure 2):

1) train an autoencoder using a large number of (unlabelled) tiles and freeze its weights;

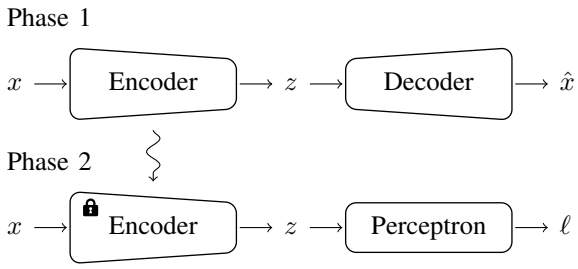


Figure 2. Concept of training phases. Here $x$ is the original tile, $z$ is the tile encoding, $\hat{x}$ is the reconstructed tile, and $\ell$ is the predicted label.

2) train a multi-layer perceptron (i.e. a stack of linear layers) using as inputs the *encodings* of the labelled tiles with respect to the autoencoder.

Since no label is required during the first phase, it is possible to create a very large training dataset of tiles by using, for example, frames of the original videos that have not been seen by the annotators.

### A. Experimental Setup

In the experiments outlined in this work, the two classes NEPHROPS and AMBIGUOUS have been merged into a single class—other methodologies that can exploit the presence of a label representing uncertainty will be explored in future works. In particular, each tile of NephTiles is given a new label:

- 0, if it is a BACKGROUND tile;
- 1, if it is a NEPHROPS or AMBIGUOUS tile;
- 2, if it is a NON-NEPHROPS tile.

Therefore, the output of the perceptron is a 3-dimensional vector $\ell = (\ell_0, \ell_1, \ell_2)$, with $0 < \ell_i < 1$ and $\sum \ell_i = 1$, such that $\ell_i$ is interpreted as the probability that the input tile has label $i$.

All the experiments have been carried out within a PyTorch (version 2.6.0) framework. Figure 3 details the architectures of the networks involved in this study. The reader can refer to Table I for a detailed description of the layers appearing in the figure. Consistently with PyTorch's conventions, 3-dimensional tensors are interpreted as (*channels*, *height*, *width*).

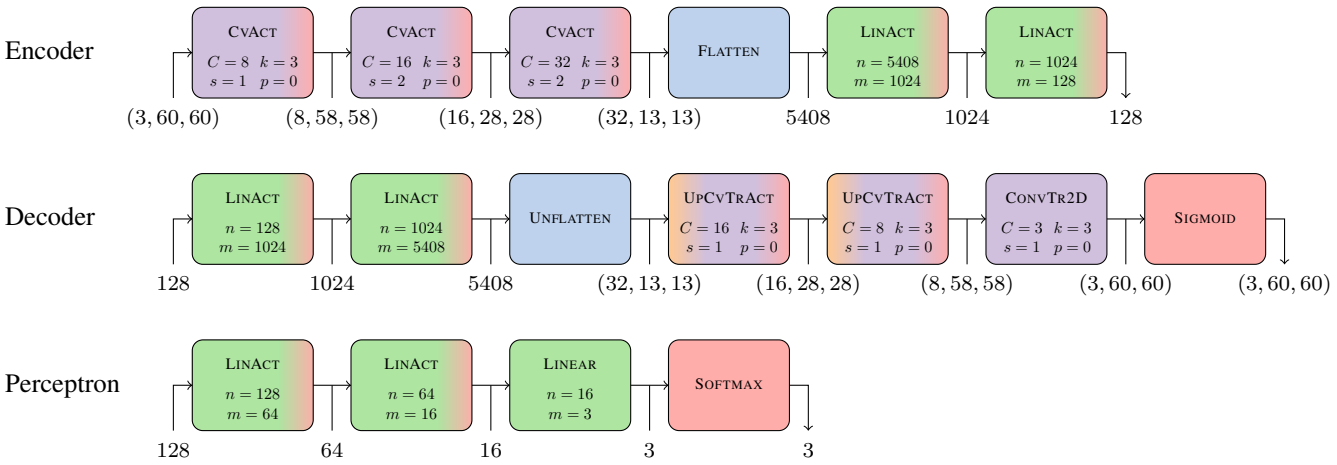


Figure 3. Architectures of the networks used in the experiments. The shape of the tensors through the networks is reported between each connected block.

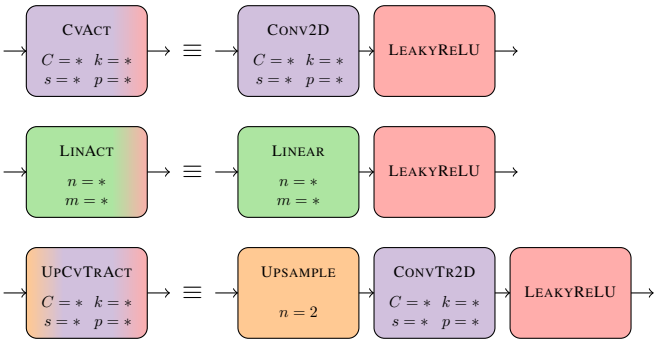


Figure 4. Shorthand blocks used in Figure 3.

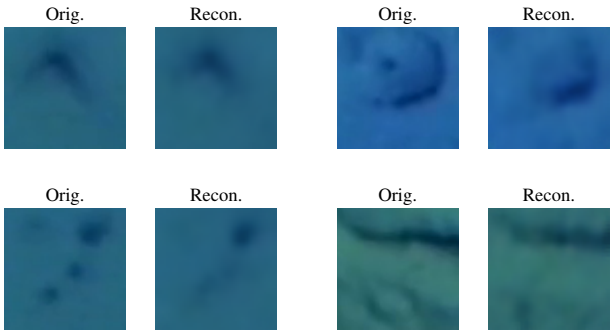


Figure 5. Examples of autoencoder reconstructions. In each pair, the original tile is displayed on the left and the reconstructed one on the right. None of the tiles belongs to either the training or the validation datasets used during the autoencoder training phase.

For improved readability, some blocks that appear multiple times have been condensed (see Figure 4).

### *B. Training Results*

For the first phase, the autoencoder is trained on a dataset consisting of 16 384 tiles, randomly selected among all the ones that can be extracted from the original video. To increase variability, the standard deviation $\sigma$ of the pixel values of the tiles' greyscale version (as integers between 0 and 255) has been considered during the selection process; in the end, about 75% of the chosen tiles has $\sigma \geq 4$.

From now on, tiles are assumed to belong to the Euclidean space $[0, 1]^{3\times 60\times 60}$. Training is implemented using the Adam optimiser, with default parameters, that targets the mean square error loss function:

$$L_{\text{MSE}}(x, \hat{x}) = \|x - \hat{x}\|^2$$

where $x$ is the original tile, $\hat{x}$ is its reconstruction, and $\|\cdot\|$ is the Euclidean norm. A validation dataset, consisting of 4096 tiles with the same variability ratio as the training dataset, has been used to determine the training stopping point; in particular, training ends if, after the first 15 epochs, the average loss for the tiles of the validation set does not decrease for 10 consecutive epochs, and the last model before those epochs is kept. Figure 5 shows some examples of reconstructed tiles.

In the second phase, the perceptron is trained using the cross-entropy as loss function:

$$L_{\text{CE}}(y, \ell) = -\ln(\ell_y)$$

where $y \in \{0, 1, 2\}$ is the ground truth label and $\ell = (\ell_0, \ell_1, \ell_2)$ is the perceptron output. Once again, the learning process is managed by the Adam optimiser. The training dataset is a subset of NephTiles with an equal number of tiles with labels 0, 1 and 2. In particular a 5-fold cross-validation scheme is implemented: five sets of 768 tiles, each containing 256 tiles for each label, are extracted from NephTiles and five models are trained, using one of the folds as validation set and the union of the others as training set. In each training session, the validation set is used to determine the training stopping point using the same policy as the one for the autoencoder training phase. A further set of 480 tiles, with the same number of tiles for each label, is extracted from NephTiles to be used as a test set after the training of each model.

A second experiment has been carried out, where the only difference is the dataset used to train the perceptron (in

Table II
CONFUSION MATRICES

Experiment 1

| | Prediction | | |
|---|---|---|---|
| GT | 0 | 1 | 2 |
| 0 | 145.4 | 7.6 | 7 |
| 1 | 6.4 | 112.6 | 41 |
| 2 | 21.4 | 54.4 | 84.2 |

Experiment 2

| | Prediction | | |
|---|---|---|---|
| GT | 0 | 1 | 2 |
| 0 | 1157.6 | 39.8 | 82.6 |
| 1 | 27.6 | 763.6 | 488.8 |
| 2 | 51.6 | 351.6 | 876.8 |

Table III
PERCEPTRON PERFORMANCES

| | Training | | Validation | | **Test** | |
|---|---|---|---|---|---|---|
| | $p$ | $r$ | $p$ | $r$ | $p$ | $r$ |
| Experiment 1 | 0.671 | 0.713 | 0.622 | 0.661 | **0.649** | **0.704** |
| Experiment 2 | 0.707 | 0.646 | 0.664 | 0.611 | **0.661** | **0.597** |

All values are rounded to the third decimal digit.

particular, the autoencoder has not been retrained). In this setup the training, validation and test datasets have been obtained by applying data augmentation to the ones used in the previous experiment. More specifically, all the planar symmetries of a square have been applied to each tile, so the final numerosity of the datasets is increased by a factor of 8.

Tables II and III show the results of the two experiments. In the confusion matrices in Table II, the element $n_{ij}$, for $i, j \in \{0, 1, 2\}$, represents the number of tiles of the *test* sets that have ground truth (GT) label $i$ and such that the perceptron assigned the label $j$ to them, averaged over the five folds. In Table III, the perceptron performance is evaluated using the precision and recall metrics computed for training, validation and test datasets limited to the label 1. In other words, the precision $p$ and recall $r$ for each dataset are determined by computing for each fold the values

$$p = \frac{n_{11}}{\sum_{i=0}^{2} n_{i1}}, \qquad r = \frac{n_{11}}{\sum_{j=0}^{2} n_{1j}}$$

and then averaging the results over the five folds.

While examining the results, the following limitations to the approach presented should be taken into consideration. First of all, precision and recall are evaluated at *tile* level, but notice that, due to the way in which the dataset NephTiles has been constructed, the same entrance can appear in multiple tiles, so it may be more meaningful to compute precision and recall at *entrance* level, for example by considering a positive detection of an entrance if at least one tile, among the ones containing it, is labelled NEPHROPS. Second, this model has been developed mainly to test the effectiveness of a dataset annotated following the directives outlined in Section II-A, so it can be considered more as a concept than a full-fledged *N. norvegicus* burrows detector—further work is necessary and will be carried out in the future.

### C. Comparison Experiments

To the best of the authors' knowledge, no other custom deep neural network architectures have been proposed in the literature specifically for detecting and classifying *N. norvegicus* burrows—other studies (e.g. [7]) use transfer learning techniques to retrain general-purpose detectors. Therefore a direct comparison between the mere values of the performance metrics would not be indicative of the actual strength of the models. Nonetheless, it may be interesting to see how already available models behave on the dataset described in Section II-C.

Table IV
DATASET SUBDIVISION FOR YOLOV8N TEST

| | No. frames | No. objects | | |
|---|---|---|---|---|
| | | NEP. | AMB. | NON-NEP. |
| Fold 1 | 135 | 90 | 24 | 203 |
| Fold 2 | 135 | 108 | 26 | 199 |
| Fold 3 | 135 | 89 | 30 | 226 |
| Fold 4 | 135 | 84 | 27 | 216 |
| Fold 5 | 135 | 95 | 26 | 204 |
| Test | 118 | 91 | 27 | 192 |

Table V
YOLOV8N PERFORMANCES

| Training | | Validation | | Test | |
|---|---|---|---|---|---|
| $p$ | $r$ | $p$ | $r$ | $p$ | $r$ |
| 0.642 | 0.437 | 0.595 | 0.393 | 0.528 | 0.412 |

All values are rounded to the third decimal digit.

A popular state-of-the-art model has been chosen for this experiment, namely YOLOv8 [13] in its "nano" variant (YOLOv8n). In this case the NephFrames dataset has been used; more precisely, its 793 non-empty frames have been divided into six random subsets, roughly maintaining the ratio between the classes in each subset (see Table IV): one is kept as a test set, and the others are used in a 5-fold validation scheme (i.e., five identical models are trained using four of the subsets as training data, and the last one as a validation set for early stopping; then the performances of the models are averaged). As in Section III-B, the NEPHROPS and AMBIGUOUS classes have been merged, so the YOLOv8n model is trained to detect and classify objects belonging to two classes (NEPHROPS+AMBIGUOUS and NON-NEPHROPS). Performances are evaluated by computing precision and recall relative to all predicted boxes assigned to the combined class, with an Intersection over Union (IoU) threshold of 0.5 to determine whether a box is a true or false positive.

Table V reports the results of the YOLOv8n training. The numbers show that even a state-of-the-art deep network, designed for general-purpose object detection and classification problems, struggles and performs poorly when dealing with a task as peculiar and complex as the one tackled by this work. In the future, one line of research will be devoted to possible tweaks of this kind of architectures, that will be fine-tuned for the detection of *N. norvegicus* burrows.

## IV. Conclusions

This paper presented some methodologies oriented to support the assessment of the *N. norvegicus* population through Computer Vision and Machine Learning. First, algorithms and guidelines specifically tailored to create properly annotated ground truth datasets have been discussed. Secondly, a deep learning model, specialised in detecting *N. norvegicus* entrances in UWTV videos, has been described. Interesting aspects emerge from preliminary tests. Table III reports the performance when the model is exploited to detect all the three possible classes. It is worth noting that the performance computed by merging classes 1 and 2 (targets potentially generated by animal action) and using the model to detect the resulting class versus class 0 (background) is considerably higher. Following this observation, future work will address the development of a cascaded sequence, with a first stage in charge of detecting potential targets of interest, followed by a second stage dedicated to the recognition, within the first selected candidates, of the *N. norvegicus* targets. In terms of precision, the classifier performance benefits from data augmentation, though at the expense of a proportional decrease in recall. This may suggest the onset of an overfitting regime triggered by augmentation. Indeed, the augmented dataset may be very similar to the starting one, and therefore prevent the network to achieve effective generalization.

An additional perspective concerns the enhancement of the model's ability to replicate the behaviour of human annotators by incorporating spatial context into the classification process. Specifically, this will involve training a second autoencoder to encode larger contextual tiles (e.g., $180 \times 180$ px) surrounding each candidate entrance. The perceptron will be fed with the concatenated encodings of both the original and contextual tiles, aiming at the improvement of the model's sensitivity to spatial relationships and relative positioning among burrow entrances.

Further analyses will concern the NephFrames dataset. As mentioned in Section II-C, currently only a few number of frames has been annotated, and all of them have been collected in a single geographical area. In the future more diverse data will be integrated in the dataset to improve the generalization capability of the models that will be trained with it.

## Acknowledgments

The authors would like to thank the IRBIM-CNR staff who contributed to collect UWTV footage, in particular Andrea Belardinelli, Pierluigi Penna and Paolo Scarpini. The authors would also like to thank Giulio Del Corso (ISTI-CNR) for the inspirational conversations.

## Copyright Notice



## References

[1] J. Aguzzi, D. Chatzievangelou, N. J. Robinson, N. Bahamon, A. Berry, M. Carreras, J. B. Company, C. Costa, J. del Rio Fernandez, A. Falahzadeh, S. Fifas, S. Flögel, J. Grinyó, J. P. Jónasson, P. Jonsson, C. Lordan, M. Lundy, S. Marini, M. Martinelli, I. Masmitja, L. Mirimin, A. Naseer, J. Navarro, N. Palomeras, G. Picardi, C. Silva, S. Stefanni, M. Vigo, Y. Vila, A. Weetman, and J. Doyle, "Advancing fishery-independent stock assessments for the Norway lobster (*Nephrops norvegicus*) with new monitoring technologies," *Front. Mar. Sci.*, vol. 9, 2022.

[2] J. Aguzzi, N. Bahamon, J. Doyle, C. Lordan, I. D. Tuck, M. Chiarini, M. Martinelli, and J. B. Company, "Burrow emergence rhythms of *Nephrops norvegicus* by UWTV and surveying biases," *Sci. Rep.*, vol. 11, 2021.

[3] ICES, "Report of the workshop on *Nephrops* burrow counting (WKNEPS)," ICES CM 2018/EOSG:25, 2019.

[4] M. Martinelli, E. B. Morello, I. Isajlović, A. Belardinelli, A. Lucchetti, A. Santojanni, R. J. A. Atkinson, N. Vrgoč, and E. Arneri, "Towed underwater television towards the quantification of Norway lobster, squat lobsters and sea pens in the Adriatic Sea," *Acta Adriat.*, vol. 54, no. 1, pp. 3–12, 2013.

[5] M. Martinelli, L. Zacchetti, A. Belardinelli, F. Domenichetti, P. Scarpini, P. Penna, D. Medvešek, I. Isajlović, and N. Vrgoč, "Changes in abundance and distribution of the sea pen, *Funiculina quadrangularis*, in the central Adriatic Sea (Mediterranean basin) in response to variations in trawling intensity," *Fishes*, vol. 8, no. 7, 2023.

[6] ICES, "Working group on *Nephrops* surveys (WGNEPS; outputs from 2022 meeting)," ICES Sci. Rep. 5:26, 2023.

[7] A. Naseer, "Advanced detections of Norway lobster (*Nephrops norvegicus*) burrows using deep learning techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 3, pp. 493–499, 2023.

[8] M. Reggiannini, M. Martinelli, O. Papini, L. Zacchetti, F. Domenichetti, and G. Pieri, "Machine learning for the evaluation of the *Nephrops norvegicus* population," in *Machine Learning, Optimization, and Data Science. Proceedings of the 10th International Conference LOD 2024, Revised Selected Papers, Part II*, ser. LNCS, vol. 15509. Springer, 2025, pp. 282–295.

[9] ICES, "Working group on *Nephrops* surveys (WGNEPS; outputs from 2019 meeting)," ICES Sci. Rep. 2:16, 2020.

[10] O. Papini, E. Cecapolli, F. Domenichetti, M. Martinelli, G. Pieri, M. Reggiannini, and L. Zacchetti. (2025) Guidelines for the annotation of *Nephrops norvegicus* UWTV videos. [Online]. Available: https://doi.org/10.5281/zenodo.14973160

[11] CVAT.ai Corporation, "Computer Vision Annotation Tool (CVAT) 2.32.0," Apr. 2025. [Online]. Available: https://github.com/cvat-ai/cvat

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[13] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO 8.2.60," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics