Very Small Neural Networks for Optical Classification of Fish Images and Videos

Marius Paraschiv[‡], Ricardo Padrino[‡], Paolo Casari[§], Antonio Fernández Anta[‡]

[‡]IMDEA Networks Institute, 28918 Madrid, Spain
[§]Department of Information Engineering and Computer Science, University of Trento, 38123 Povo (TN), Italy

***Abstract*—The task of visual classification, done until not long ago by specialists through direct observation, has recently benefited from advancements in the field of computer vision, specifically due to statistical optimization algorithms, such as deep neural networks. In spite of their many advantages, these algorithms require a considerable amount of training data to produce meaningful results. Another downside is that neural networks are usually *computationally demanding* algorithms, with millions (if not tens of millions) of parameters, which restricts their deployment on low-power embedded field equipment.**

In this paper, we address the classification of multiple species of pelagic fish by using small convolutional networks to process images as well as videos frames. We show that such networks, even with little more than 12,000 parameters and trained on small datasets, provide relatively high accuracy (almost 42% for six fish species) in the classification task. Moreover, if the fish images come from videos, we deploy a simple object tracking algorithm to augment the data, increasing the accuracy to almost 49% for six fish species. The small size of our convolutional networks enables their deployment on relatively limited devices.

***Index Terms*—Underwater images; fish classification; neural networks; optimized architectures; accuracy; sea trial**

I. Introduction

The study of the marine ecosystem often requires long-term observations and statistical analyses, performed on a discrete set of species, in order to study variations in overall populations, migration patterns, or human impact. While these tasks are often carried out manually, recent advances in statistical image recognition algorithms (especially related to deep learning) introduce substantial automation opportunities. Deep learning is the sub-field of machine learning that focuses on the design and use of deep neural networks for pattern recognition and prediction tasks (such as classification and regression). This technique has seen breakthrough applications in medicine [1], [2], [3], autonomous vehicles [4], object detection [5], speech synthesis [6], automatic translation [7], speech-to-text [8], agriculture [9] and energy consumption optimization [10], among many others. A detailed overview of the field can be found in [11], [12].

In this paper we describe our experience with the design and optimization of deep learning for the classification (i.e., the identification of the species) of fish images captured by submerged camera arrays. Our work is framed as part of the Symbiosis project,[1] concerned with the detection and classification of pelagic fish in the Mediterranean Sea and the Atlantic Ocean. In this project, an autonomous system is lowered beneath the sea surface in a mooring. The system has two sets of camera arrays, as well as sonars and other sensors, and records in real time the type of each individual fish within a given radius. Real-time statistics are computed and aggregated results are sent to an external observation station.

As the requirements of Symbiosis are to detect and classify a specimen in real time, a number of constraints need to be considered. Firstly, the range of each camera is of approximately 10 meters, and the target pelagic fishes are fast. Hence, the inference time must be reduced to a bare minimum. Secondly, as the size and position of the specimen varies with respect to the angle at which it approaches the camera and its distance from it, images also need to be cropped and scaled before being sent for classification. These are preprocessing steps undertaken on the same device that performs inference for the classification model, imposing restrictions on the size of the deep neural network and the models that can be used.

In this project we target six species of pelagic fish (see Section III), and we want to differentiate i) images showing to fish of these species from images of other species (*binary classification*), and ii) tell apart the images of target Symbiosis species from one another (*multi-class classification*). Prior to being deployed on the final hardware, the classifiers need to be trained. Unfortunately, producing relevant datasets of the six species of pelagic fish we target is a considerable challenge. Firstly, there exist a limited number of images and videos of these species, since they are usually found if deep waters. Secondly, available images are either taken above the sea level (which we chose not to use, as they do not represent Symbiosis's operational environment), or are present in observation videos, from which many images can be extracted, but typically showing a small set of specimens that repeat across video frames. This has heavy implications on the training of our deep neural networks. Specifically, in order to counter the lack of variety of individuals, the training and test phases are performed on images extracted from different videos whenever possible. We consider this situation to provide the most credible estimate for our system's performance.

Due to the fact that the models need to be very small (the target is deep neural networks with 10,000 to 20,000 param-

[1] http://symbiosis.networks.imdea.org/

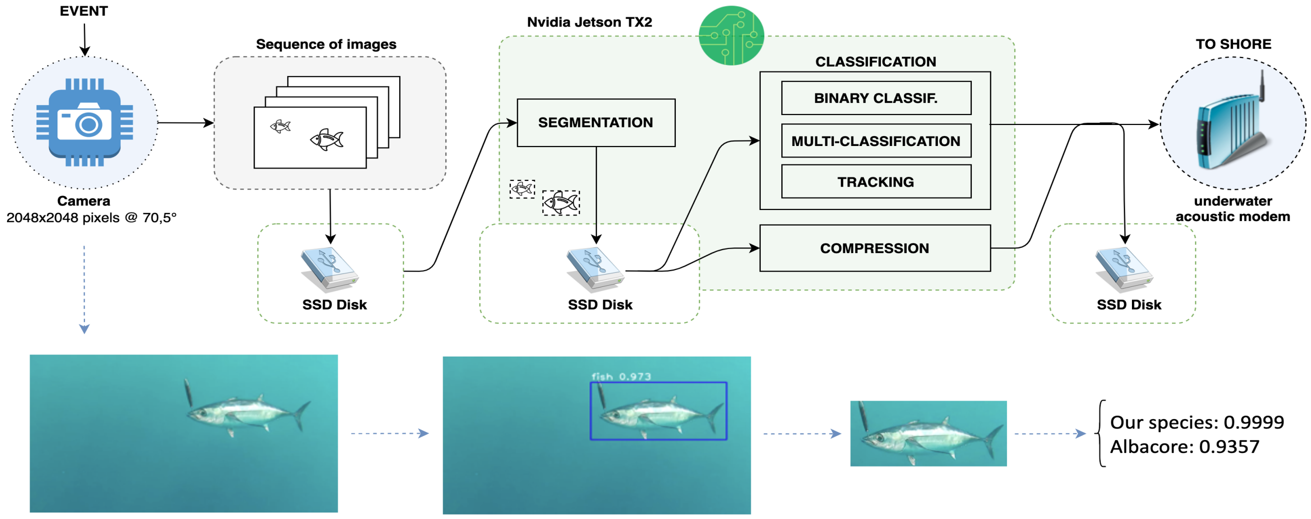

Fig. 1. The optical data pipeline, including similarity filtering, classification, and object tracking.

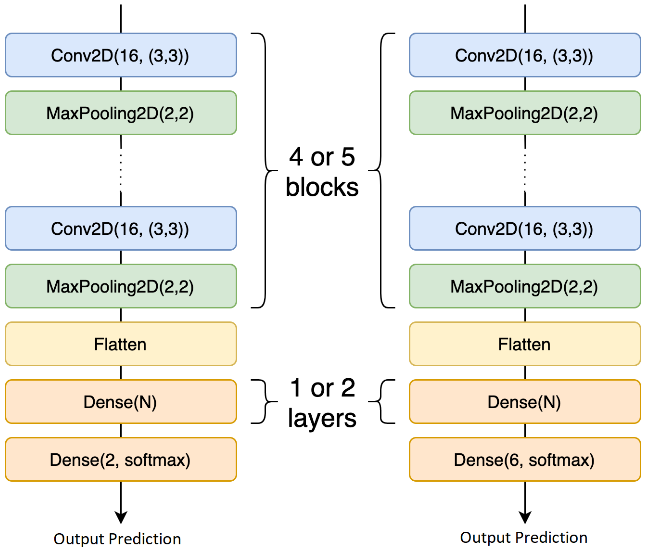

Fig. 2. Architectures used for binary and multi-class classification.

eters), their representation capacity is reduced and, combined with their training on relatively small datasets, they could suffer from poor performance. To remedy this, and taking advantage of the fact that the images come from videos, we insert an additional step, between preprocessing and classification, where we perform object tracking, presenting the classifier with multiple images of the same object, with the scope of increasing the amount of information available.

The main contributions of the paper are: (1) presenting a series of very small neural networks that can be implemented with very limited hardware resources, providing a training method based on small and imbalanced datasets, and (2) using object tracking to assist the classifier in its task.

The paper structure is as follows: in Section II we discuss various theoretical aspects, such as a brief overview of the neural architectures that have been used, relevant literature as well as object tracking and the components of our data pipeline. In Section III we give a detailed description of the actual system and its sensors, as well as the various hardware on which we performed our tests. This is followed by Section IV, where we present experimental results for both binary classification (distinguishing between our six classes grouped into two subsets, with the aim of trying out the proposed architectures) and multi-class classification (identifying each of the six classes of interest). The final experiments present a comparison between the cases where object tracking has been used and those where it has not been used.

II. Theoretical Aspects and Related Work

Of particular significance in the field of image processing is a type of neural layer known as a convolutional layer. Drawing inspiration from the human visual cortex, convolutional layers are able to exploit a series of invariances, such as invariance to local deformation or translation, in order to perform efficient feature aggregation and also drastically reduce the number of parameters, as opposed to the classical multi-layer perceptron (or fully-connected) layers [13].

The optical processing in the Symbiosis project (see Figure 1) is focused on the detection and automatic classification of fishes, seen in front of one of the submerged cameras and captured in a sequence of video frames. This process makes use of convolutional neural networks to solve two problems. First, image detection and segmentation, extracting relevant segments from entire frames [14]. Second, image classification, which is the focus of this paper, whose purpose is selecting the segments representing one of the six relevant species of pelagic fish. This involves differentiating them first from segments portraying fishes and artefacts that are irrelevant to our problem. The detection and segmentation task uses the neural network RetinaNET [15], whereas for the image classification tasks we have evaluated popular convolutional architectures, like VGG16 [16] and François Chollet's simple architecture [17], and compared them to our own custom networks (see Figure 2).

Our final goal is the classification of an object (detected as a fish), which may appear in several frames, into one of seven classes (one class for each the six species of interest in Symbiosis, and a seventh class for *other species*). Feeding the sequence of frames into a convolutional network, and relying exclusively on the classification results to identify the fish species would be problematic, as the same fish may disappear and reappear intermittently across subsequent frames. To circumvent this problem, we resorted to performing object tracking, detecting whether or not the same fish appears in a consecutive sequence of frames, after the frame itself has been classified. For the purpose of tracking, we use the SORT [18] algorithm.

Our approach, namely using convolutional neural networks for detection, segmentation, and classification, integrates well with previous work in the field. In what follows, we provide a brief description of a number of works, related to object detection in underwater scenarios. In [19], the authors present a method for fish localization in underwater images. The applied method consists of denoising and cleaning the images in a preprocessing step, and then splitting them into regions using the mean shift algorithm. Various statistical estimation techniques are then used to combine the regions into objects. This is a rather direct and hard-coded method, but it can be used as a baseline for the following results.

Another effort, using classical machine learning algorithms, is presented in [20]. The authors explore the limitations of analyzing input from stereo video systems and propose a two-step process for fish detection. First, they use a dimensionality reduction technique, such as PCA, to model pixel-level knowledge about the shape of the fish. Once the shape is identified, the method requires that a significant region of the fish's body, usually the head, be segmented, and yields sub-pixel accuracy in the object's localization.

Other efforts focus strictly on deep learning and on the capacity of convolutional neural networks to complete a variety of vision-related tasks. In [21], a CNN-based algorithm is used (called R-CNN, or Regions with CNN). Fast R-CNN is a version of the algorithm that initially generates subsegments from many candidate regions, then uses a greedy algorithm to combine similar segments, and finally combines the generated segments into final candidate proposals. The authors apply this technique to a dataset of approximately 24,000 images of fishes from 12 different species, and report a mean average precision (mAP) of 81.4%.

Another approach is provided in [22], where two supervised learning methods are proposed. The first method extracts object features using the Histogram of Oriented Gradients (HoG) technique, and then uses a classical machine learning classifier, such as an SVM, to differentiate the fish. The second proposed method is an end-to-end convolutional network.

As we have seen, deep learning algorithms have long surpassed traditional hard-coded approaches for vision tasks. Whether used in combination with some classical techniques (specially for feature extraction, denoising, or more general preprocessing, before a learning component performs the classification task) or in an end-to-end manner (where a convolutional network is the one responsible for extracting its features and performing the classification task), almost all identified works, including our own, use deep learning due to its ease of domain adaptation and relatively low implementation complexity.

We perform two benchmarking tasks, one being binary classification between our species of interest and others (in this paper we simply use the whole dataset of our six species grouped into two subsets), and one multi-class classification, distinguishing among our particular species. As the hardware resources are limited, small architectures are chosen, each having less than 20,000 trainable parameters. They are variations on the same general pattern, depicted in Figure 2.

As depicted in Figure 1, the classification procedure consists of a two-step process: image classification and object tracking. Tracking identifies a set of images, representing the same fish, which are classified separately. Each classifier returns a vector of probabilities, describing the representative class of the particular image. A voting system then decides on the accepted result, and this is assigned to the detected object.

III. Experimental Setup

The Symbiosis system is an opto-acoustic system whose objective is to detect, track, monitor, and classify pelagic fishes. The system is to be submerged into the sea, attached to a mooring cable. The system (see Figure 3) includes acoustic and optical subsystems. The acoustic system is a new advanced sonar designed and built ad hoc, with capabilities to detect, localize and track marine faune within a range of 500 m. The optical system is composed of two camera arrays, one in the top and other in the bottom of the system, separated 15 meters between them. The camera array has six camera units in a circle with a 10.5 degrees of overlap in the field of view on either side. Each camera unit consists mainly in the camera itself and an NVIDIA Jetson TX2 with an embedded GPU to process the data using neural networks. The system structure has longitudinal cylindrical distribution. A large waterproof case hosts a central computational unit (which is another NVIDIA Jetson TX2) and batteries. Sonar elements are distributed in several sub-units along the whole structure (e.g., the three cylinders facing left in the right-most photo of Figure 3). The system is designed to work autonomously under water, 20-40 meters deep, attached to a mooring cable with a surface buoy. This surface buoy provides connectivity with the shore. In the design terms, the energy autonomy of the system is one month without recharging the battery.

Energy consumption is an important constraint in the system. Due to that, acoustic and optical detection subsystems do not work simultaneously. Instead, the acoustic system operates continuously in low-energy mode, until a possible fish is detected in the detection range (10-500 meters). When that

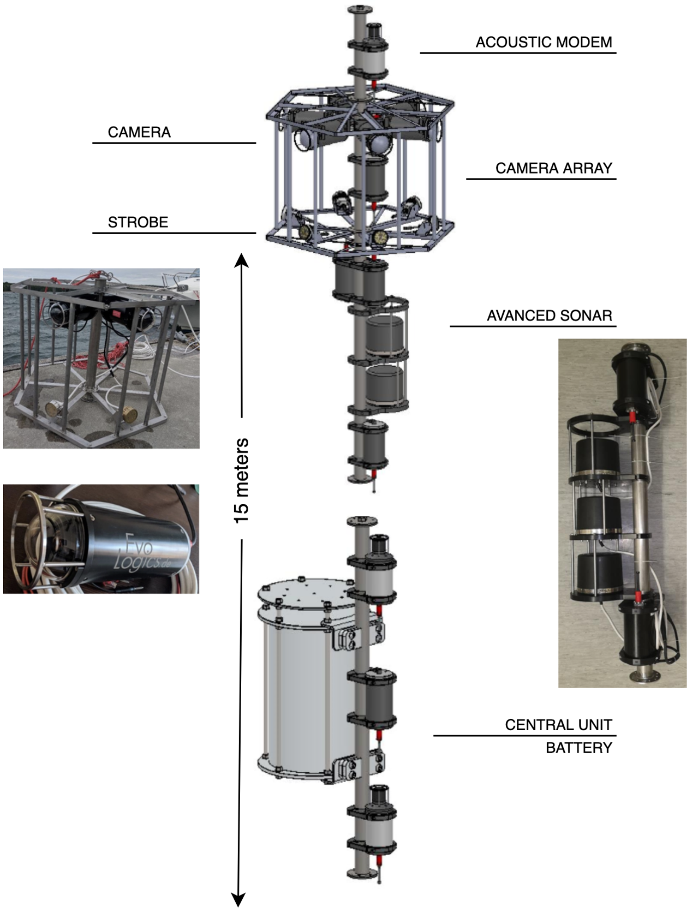

Fig. 3. Schematic representation of the acoustic and optical Symbiosis system and its main components (only the upper camera array is shown).

happens, the advanced sonar takes control to localize and track the targets. All this information is analyzed to classify the targets as one of the species of interest and estimate the biomass according to the acoustic data. If a fish trajectory can enter the detection vision range (1-10 meter radius, depending on water and light conditions) of one or more cameras, the optical pipeline is activated. The camera units will capture a sequence of images or frames, and perform the optical detection and classification of the possible fish candidates. The results of the cameras will be aggregated in the central unit. This data and the acoustic data will be processed and merged to generate the final report to be sent to shore for each event, to be further analyzed by marine biologists.

The Symbiosis prototype focuses on six species of bony fishes (see Figure 4), selected based on their high commercial importance, distinct behaviour or appearance, and occurrence at either or both study locations of this project – the Eastern Mediterranean Sea and the Canary Islands:

- Albacore tuna (*Thunnus alalunga*);
- Dorado (dolphinfish, *Coryphaena hippurus*);
- Atlantic mackerel (*Scomber scombrus*);
- Mediterranean horse mackerel (*Trachurus mediterraneus*);
- Greater amberjack (*Seriola dumerili*);
- Swordfish (*Xiphias gladius*).

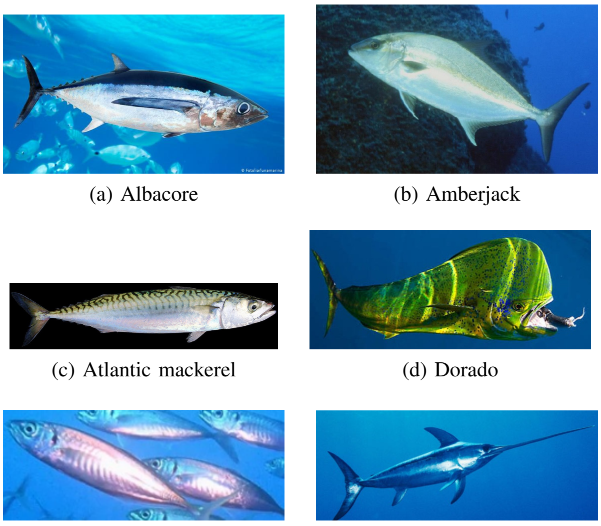
(a) Albacore
(b) Amberjack
(c) Atlantic mackerel
(d) Dorado
(e) Mediterranean mackerel
(f) Swordfish

Fig. 4. Example of the studied species in Symbiosis project.

The interested reader is referred to the corresponding Symbiosis project report [23] for more information about these species.

IV. Results

It is generally considered that neural networks are data-greedy classification algorithms, requiring a consistent sample of training examples, corresponding to the different classes, for efficient training. Otherwise, due to the high number of trainable parameters (weights), there is a high risk of overfitting when using small datasets. Hence, for the purpose of the optical processing tasks of Symbiosis, the compilation of a reasonably large and curated training dataset was a necessary step. This requires collecting, cleaning and further aggregating in a homogeneous form a large quantity of images and their corresponding relevant meta-data: labels, bounding boxes (regions of interest, or ROI), etc. This often represents the most time-consuming data analysis sub-task.

The raw data used in this paper was provided by the Biology team of the Symbiosis project, from the University of Haifa, who performed the collection task from a multitude of public and private data sources. They collected a number of photographs and videos that contained images of the 6 species of pelagic fish targeted in the optical classification process.

The data extraction and preprocessing of the raw data provided by the Biology team has gone through a series of steps. As the data contained both image and video files (of very different lengths), frame extraction was performed on the latter. Then, a preliminary version of the detection and segmentation algorithm [14] was applied on the resulting individual frames by the Optics team from the University of Haifa. This produced more than 1.5 million image segments (containing ROI with potential fish). As the algorithm was

TABLE I

Distribution of images subset for training. Total number of images: 51,260.

Species	Training	Validation	Testing
Albacore	$4,100$	$1,393$	$1,353$
Amberjack	$8,563$	$2,931$	$2,803$
Atlantic mackerel	$2,679$	840	812
Dorado	$7,631$	$2,560$	$2,576$
Mediterranean mackerel	$1,884$	654	680
Swordfish	$5,789$	$2,035$	$1,977$
TOTAL	$30,646$	$10,413$	$10,201$

still being optimized at the time, we inspected the images manually to discard those that did not contain a fish, or where the size or quality of the fish image was too low. This filtering process was based on various characteristics such as image quality, minimum size, full or partial coverage of the fish body, detection accuracy, etc. While a small part of this process was crowdsourced, the bulk of this cleaning was finally done manually, resulting in 51,260 segments.

The datasets used to perform both binary classification (distinguishing between our species of interest, grouped into two classes) and multi-species classification (distinguishing among our six species of interest), have been further split into train, test, and validation subsets, with a distribution of the sizes of (80%, 10%, 10%) over each species. The segments from the same video were kept together in this process. The size of this dataset appears in Table I. As an be seen, while the total number of images is not small, these are very unevenly distributed among the six species.

As mentioned above, the Symbiosis optical detection and classification process consists of a pipeline with two separate stages, both using convolutional neural networks. The first stage, the detection and segmentation task, developed by the Optical team from the University of Haifa [14], uses the network RetinaNET [15] to extract regions of interest (ROI) or image segments, corresponding to areas where a fish has been identified in a raw image or video frame. The second stage of the optical process, which is the focus of this paper, deals with segment classification, in which each image segment extracted is processed to determine if it corresponds to a fish of one of the six species of interest, and in case it is, which species. (See Figure 1 for a scheme of the Symbiosis optical pipeline and the role of these two stages in it.)

In fact, the classification stage is further split into two separate tasks: the binary classifier, which discriminates our species of interest from other species of fish, and the multi-species/multi-class classifier, providing a representative label from one of our six classes/species.

One of our main tasks has been to find extremely light architectures that would provide results comparable to well-known networks (such as VGG16 [16] or Chollet [17]). Some of the proposed models are just under $11,000$ parameters, and provide comparable performance to their considerably larger counterparts, in both binary and multi-class classification. All architectures discussed in the present paper are listed, together with their number of trainable parameters, in Table II. As can be seen, the architectures proposed in this paper are all formed by a sequence of pairs of layers, each pair formed by a convolutional layer and a max pooling layer (denoted C_{16} and M, respectively), followed by a sequence of dense layers (denoted D_x, $x \in \{8, 16, 32\}$), and a final classification layer (D_y, $y \in \{2, 6\}$, see Figure 2). We remark that VGG16 is a pre-trained architecture, for which only the final layer has been retrained on our particular dataset.

TABLE II

Total number of trainable parameters for the architectures considered in the present paper. The final dense layer D_y is either D_2 or D_6, for binary and multi-class classification, respectively. VGG16 is pre-trained and only the final layer has been re-trained with our dataset.

Architecture	Number of Parameters
$4(C_{16}M)D_8D_{16}D_y$	$18,030$
$4(C_{16}M)D_{32}D_{32}D_y$	$50,166$
$5(C_{16}M)D_8D_y$	$10,942$
$5(C_{16}M)D_{32}D_y$	$14,566$
$5(C_{16}M)D_{16}D_8D_y$	$12,238$
VGG16	$15,124,518$
Chollet	$1,667,494$

The tracking component makes use of SORT [24], [25], [26], an open-source tracking algorithm which shows good performance with regards to typical fish movements. As opposed to a deep learning model, SORT is not very computationally-expensive, and this facilitates its deployment on small systems, where memory and computational power are limited. The input to the tracker is formed by the image segments coordinates (pixel-wise), from every frame, in consecutive order. The algorithm then provides an artefact identifier and the predicted position of the object, in the next frame. In our algorithm, the predicted coordinates are then compared with the current coordinates, aiming at maximizing the intersection over union (IoU), and link the identifier with the image segment. A particular remark is that the tracker needs to be updated frame-by-frame, even in situations where the object itself is missing, in order to keep a consistent history. This is useful and brings strength if the detection stage could have a false negative for an intermediate frame, in particular, in cases with low frame rate.

For benchmarking a series of neural network architectures, we have constructed two datasets, one in which the six species of interest have been grouped into two classes, and a second dataset in which each species represents its own class. In the binary dataset, the images from amberjack and dorado are placed in one class, while the rest of species are placed in the other class. This results in two sets of results, for binary and multi-class classification, respectively. The setup in both cases is identical: segmented images are first sent to a classifier, in order to establish the fish species. A group of segmented images, and their assigned classes, are combined with the tracking procedure, to determine if the images represent the same object. The classification results are thus placed into a vector, if the tracking algorithm decides they represent the same fish. We then have three methods of extracting an accuracy value from the vector of predictions: by taking the

TABLE III

Results for the binary classification task with and without the tracking component. The mean, max *and* voting *columns refer to the type of decision scheme used when the tracking mechanism is applied.*

Architecture	Test Accuracy (NO TRACKING)	Test Accuracy (Acc. Mean)	Test Accuracy (Acc. Max)	Test Accuracy (Acc. Voting)
$4(C_{16}M)D_8D_{16}D_2$	52.73 %	61.05 %	61.67 %	61.08 %
$4(C_{16}M)D_{32}D_{32}D_2$	53.20 %	60.51 %	60.53 %	60.71 %
$5(C_{16}M)D_8D_2$	63.78 %	64.85 %	63.80 %	64.39 %
$5(C_{16}M)D_{32}D_2$	62.08 %	63.43 %	62.24 %	63.52 %
$5(C_{16}M)D_{16}D_8D_2$	57.68 %	60.47 %	60.20 %	60.46 %
VGG16	74.37 %	80.47 %	73.86 %	79.95 %
Chollet	69.09 %	65.42 %	68.39 %	65.01 %

TABLE IV

Results for the multi-class classification task with and without the tracking component. The mean, max *and* voting *columns refer to the type of decision scheme used when the tracking mechanism is applied.*

Architecture	Test Accuracy (NO TRACKING)	Test Accuracy (Acc. Mean)	Test Accuracy (Acc. Max)	Test Accuracy (Acc. Voting)
$4(C_{16}M)D_8D_{16}D_6$	37.55 %	38.09 %	41.24 %	36.10 %
$4(C_{16}M)D_{32}D_{32}D_6$	38.38 %	54.21 %	53.42 %	53.66 %
$5(C_{16}M)D_8D_6$	35.30 %	47.53 %	48.32 %	45.64 %
$5(C_{16}M)D_{32}D_6$	37.39 %	54.43 %	51.42 %	52.89 %
$5(C_{16}M)D_{16}D_8D_6$	41.54 %	45.11 %	47.55 %	43.96 %
VGG16	52.73 %	62.79 %	59.71 %	61.48 %
Chollet	28.94 %	48.21 %	46.54 %	48.61 %

mean of all values (denoted as Acc. Mean), by taking the maximum value (Acc. Max) or by using a majority voting scheme (Acc. Voting).

We present the results for the binary classification in Table III. As can be seen VGG16 is the architecture with the largest accuracy, with more that 74% without tracking. However, this comes as a cost of using more than 15 millions of parameters. On the other hand, architecture $5(C_{16}M)D_8D_2$ reaches almost 64% accuracy with less than $11,000$ parameters. A second observation from Table III is that tracking increases the accuracy in all cases (except for the Chollet architecture), in some cases significatively (more than 8%).

For the second task, namely multi-class classification among our species of interest, we observe a relatively lower accuracy when the convolutional classifier alone is used. In Table IV, we show the multi-class classification results for the different architectures, with and without the final object tracker. Again VGG16 is the architecture with highest accuracy (more that 52% without tracking and more than 62% with tracking), but much simpler architectures with roughly $12,000$ parameters have also comparable accuracy, of more than 41% without tracking and more than 47% with tracking. Again, we observe that tracking increases the accuracy significantly.

V. Conclusions

Performing underwater object detection with autonomous probes poses a series of technical limitations on computational power, memory availability, and image quality, among others. Working within these constraints, we have proposed a series of neural network models that are light-weight, in terms of the number of trainable parameters, but nevertheless produce results similar to much larger and widely used architectures, such as VGG16. To further augment their predictive capability, we combine the output of the image classifiers with an object tracking algorithm, SORT, that is computationally inexpensive and does provide a consistent boost in classification performance.

A difficult task for this work has been acquiring and preparing a dataset. In general, the amount of relevant images available (for the classes of interest) have been limited. As such, we found it necessary to perform frame extraction from videos. This, in turn, raised the issue of frame similarity, and the need to avoid having nearly identical images in both the train and validation/test sets. In addition to the size of the training data, its balance also posed a challenge. As seen in Table I, the number of images from each species is very different. This may be a reason that may have caused a lower accuracy. For binary classification, the six classes have been split into two categories, by maintaining a similar number of images per class. Future work includes collecting more images to balance the dataset, and performing more experiments with our current dataset.

Acknowledgment

This work has received support from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no. 773753 (SYMBIOSIS). We also gratefully acknowledge the support of NVIDIA Corporation for the donation of the Jetson TX-2 board used to develop the real-time version of our algorithm.

References

[1] T. Shaikhina and N. A. Khovanova, "Handling limited datasets with neural networks in medical applications: A small-data approach," *Artificial intelligence in medicine*, vol. 75, pp. 51–63, 2017.

[2] V. L. Patel, E. H. Shortliffe, M. Stefanelli, P. Szolovits, M. R. Berthold, R. Bellazzi, and A. Abu-Hanna, "The coming of age of artificial intelligence in medicine," *Artificial intelligence in medicine*, vol. 46 1, pp. 5–17, 2009.

[3] I. Kononenko, "Machine learning for medical diagnosis: history, state of the art and perspective," *Artificial intelligence in medicine*, vol. 23 1, pp. 89–109, 2001.

[4] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. D. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," *ArXiv*, vol. abs/1704.07911, 2017.

[5] Z.-Q. Zhao, P. Zheng, S. tao Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, pp. 3212–3232, 2019.

[6] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *ArXiv*, vol. abs/1609.03499, 2016.

[7] S. P. Singh, A. Kumar, H. Darbari, L. Singh, A. Rastogi, and S. Jain, "Machine translation using deep learning: An overview," *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, pp. 162–167, 2017.

[8] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, A. Y. Hannun, B. Jun, T. Han, P. LeGresley, X. Li, L. Lin, S. Narang, A. Y. Ng, S. Ozair, R. Prenger, S. Qian, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, A. Sriram, C. Wang, Y. Wang, Z. Wang, B. S. Xiao, Y. Xie, D. Yogatama, J. Zhan, and Z. Zhu, "Deep speech 2 : End-to-end speech recognition in english and mandarin," *ArXiv*, vol. abs/1512.02595, 2016.

[9] P. A. Dias, A. Tabb, and H. Medeiros, "Apple flower detection using deep convolutional networks," *ArXiv*, vol. abs/1809.06357, 2018.

[10] R. F. Berriel, A. T. Lopes, A. Rodrigues, F. M. Varejão, and T. Oliveira-Santos, "Monthly energy consumption forecast: A deep learning approach," *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 4283–4290, 2017.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[12] R. Vargas, A. Mosavi, and R. Ruiz, "Deep learning: a review," *Advances in Intelligent Systems and Computing*, 2017.

[13] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Computation*, vol. 29, pp. 2352–2449, 2017.

[14] D. Levy, Y. Belfer, E. Osherov, E. Bigal, A. P. Scheinin, H. Nativ, D. Tchernov, and T. Treibitz, "Automated analysis of marine video with limited data," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1466–14 668, 2018.

[15] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 318–327, 2020.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2015.

[17] F. Chollet, "Building powerful image classification models using very little data," https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html, June 2016, accessed: 2020-08-12.

[18] A. Bewley, Z. Ge, L. Ott, F. T. Ramos, and B. Upcroft, "Simple online and realtime tracking," *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, 2016.

[19] M. Boudhane and B. Nsiri, "Underwater image processing method for fish localization and detection in submarine environment," *J. Vis. Commun. Image Represent.*, vol. 39, pp. 226–238, 2016.

[20] M. Ravanbakhsh, M. R. Shortis, F. Shafait, A. S. Mian, E. S. Harvey, and J. W. Seager, "Automated fish detection in underwater images using shape?based level sets," *Photogrammetric Record*, vol. 30, pp. 46–62, 2015.

[21] X. Li, M. Shang, H. Qin, and L. Chen, "Fast accurate fish detection and recognition of underwater images with fast R-CNN," *OCEANS 2015 - MTS/IEEE Washington*, pp. 1–5, 2015.

[22] S. Villon, M. Chaumont, G. Subsol, S. Villéger, T. Claverie, and D. Mouillot, "Coral reef fish detection and recognition in underwater videos by supervised machine learning: Comparison between deep learning and hog+svm methods," in *ACIVS*, 2016.

[23] E. Bigal, A. Scheinin, S. Einbinder, and D. Tchernov, "Symbiosis biology report," http://symbiosis.networks.imdea.org/sites/default/files/2018-09/Symbiosis\%20Biology\%20Report-2.pdf, May 2018, accessed: 2020-08-24.

[24] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649, 2017.

[25] A. Bewley, "SORT algorithm source," https://github.com/6o6o/sort/, accessed: 2020-08-12.

[26] "SMOT challenge 2015 (multi-target tracking)," https://motchallenge.net/, accessed: 2020-08-12.