

ARIANNA: A novel deep learning-based system for fin contours analysis in individual recognition of dolphins

Rosalia Maglietta^{a,*}, Arianna Bussola^b, Roberto Carlucci^c, Carmelo Fanizza^d, Giovanni Dimauro^b

^a Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing - National Research Council, via Amendola 122/D-O, 70126 Bari, Italy

^b Department of Computer Science, University of Bari, Via Orabona 4, 70125 Bari, Italy

^c Department of Biology, University of Bari, Via Orabona 4, 70125 Bari, Italy

^d Jonian Dolphin Conservation, Viale Virgilio 102, 74121 Taranto, Italy

ARTICLE INFO

Keywords:

Machine learning
Deep learning
Edge and feature detection
Photo identification
Cetaceans

ABSTRACT

Starting from a digital image that represents the dolphin's body, distinctive features are extracted and used to find the identity of the unknown dolphin in a set of known individuals. This process is called photo identification, used by experts to monitor dolphins, providing relevant data to preserve the environment and its biodiversity. In this work, we show how semantic segmentation can be used to automatically extract a dolphin's fin contour starting from a cropped photo of the fin, and how this contour can be used for individual identification. A novel contour-based system, called ARIANNA, for the automated cetacean photo identification was designed, developed and tested. The novelty of this system is the adoption of two original modules. The first one, which takes as input a new cropped fin image of unknown dolphin, is devoted to the extraction of a mask that depicts the outline of the unknown fin; the core of this module is a trained neural network, specialized in semantic segmentation of images. The second module is designed to compare the outline of the unknown fin with the outlines of all known dolphins, collected in a referring catalogue, returning a ranked list of the best matches where to search the dolphin identity. The experiments were conducted on images collected between 2013 and 2020 in the Northern Ionian Sea (Central-eastern Mediterranean Sea), which presented cropped fins of Risso's dolphin *Grampus griseus*, one of the least-known cetacean species on a global and Mediterranean scale. The results suggest that ARIANNA provides advances over the state-of-the-art methods, can efficiently assist researchers in the photo identification of dolphins and can be a starting point for further studies on the photo identification of different species based on semantic segmentation.

1. Introduction

Nowadays, studies about the continuous changes in our planet are of vital importance to understand how the ecosystems are altering over years and make decisions to preserve them. The marine environment is one of the most affected, so we must examine the habitat use, migration and habits of the marine species populating it to analyse how they react to those changes. To that regard, in the past, invasive techniques were used to identify different specimens, such as geolocation devices applied on animals' bodies to keep track of their movement. This approach may affect animal behaviour, leading to bias in studies and ethical problems. The advent of the technologies introduced new methods or boosting up existing methods for the identification of specimens, such as photo

identification, used to identify an individual by analysing its digital images. Thus, a huge amount of precious information on animals can be collected without directly affecting them. In this work, we focus on a particular species, Risso's dolphin *Grampus griseus*, which is widespread in tropical and temperate waters at high depths. However, information and estimates on its distribution are scarce and insufficient, and the Mediterranean subpopulation is ranked as Data Deficient by the IUCN Red List Kiszka et al. (2018). The particularity of this species is the presence of long lasting identifiable natural marks on its body, such as scars, notches and variations in the shape of the fin, caused by fights with prey and other specimens (see Fig. 1). These features can be exploited as distinctive characters for their photo identification Maglietta et al. (2020), Maglietta, Renò, et al. (2018), Maglietta, Bruno, et al. (2018).

* Corresponding author.

E-mail address: rosalia.maglietta@cnr.it (R. Maglietta).



Fig. 1. A Risso's dolphin specimen.

In the literature, there are software programs that support researchers in the photo identification of cetaceans, as widely discussed in Maglietta et al. (2022). One of the oldest ones is DARWIN Stanley (1995), a semi-automated software that helps researchers identify bottlenose dolphins *Tursiops truncatus* and other species Paschoalini and Santos (2020), Frau et al. (2021), de Mello et al. (2019), Dinis et al. (2016), Degrati et al. (2018), Estrade and Dulau (2020), Degrati et al. (2020), Samuelson et al. (2021) by supporting them while tracing the outline of the dolphin's fin and minimizing subjectivity in the comparison of the unknown specimen with a catalogue of previously identified dolphins. Users must initially provide a catalogue of dolphin photos that are categorized as known specimens: for each photo, the user must roughly trace a general outline of the dolphin's fin with the cursor, so that the software can progressively adjust it by moving the points from their initial locations to the actual edge of the fin. During the photo identification process, DARWIN extrapolates some salient feature points on the fin, i.e., the beginning of the leading edge, the end of the leading edge, the tip of the dorsal fin, the most prominent notch on the trailing edge and the end of the trailing edge. Once the catalogue of the known specimens is completed, the user can use DARWIN to identify new dolphins. Starting from a photo depicting the unknown specimen, the fin outline will be extracted from it as described before, and the obtained contour of the unknown fin is compared with all available outlines in the referred catalogue to find the best matching ones. The DARWIN output is a ranked list of possible matches between the unknown dolphins and those collected in the catalogue. A downside of this software is that the user must always directly interact with the system, especially during the contour extraction phase.

Most recently, a fully automated photo identification software, finFindR Thompson et al. (2021), was presented in the literature, which can be considered the-state-of-the-art of contour-based algorithms for cetacean automated photo identification. After extracting dolphin fins from the input images, it tracks their outline and produces a ranked list with the top 50 most likely matching identities, which enables the users to view side-by-side image pairs and make the final identity determinations. In detail, mainly based on the application of neural networks, the finFindR workflow consists of four steps: fin isolation, isolation of the trailing edge, computation of a score based on distinguishing features and computation of the proximity of the query images score to the scores of other fins in the catalogue. FinFindR placed the correct individual among the 10 top-ranked matches in 94% of tests and among the 50 top-ranked matches in 97% of tests. However, these results were obtained using only high-quality photos, which makes this assessment poorly consistent with a real use scenario Maglietta et al. (2022). In fact, the authors in Thompson et al. (2021) scored each image on photography quality and dorsal fin distinctiveness using established criteria Friday et al. (2000), Urian et al. (1999, 2014). Only images that were

ranked as average or better photography quality were included in the analysis. No poor-quality images were included in the test.

The main goals of this paper are to design, develop, and test a new fully automated cetaceans photo identification system, able to process real-life scenario images, without any kind of photos selection based on images quality. With this aim, we present a new photo identification system, called ARIANNA, whose novelty is the adoption of two main modules: the *contour extraction module* and the *matching module* (see Fig. 3). Starting from the original photos, acquired during marine expeditions, the cropped images of dolphins' fins have been manually obtained, and collected in a catalogue. For each cropped image, a human-provided label is used, consisting of a manually extracted binary mask, where fin outline is highlighted. This labelled catalogue is used as input to ARIANNA. In the contour extraction module, the system uses a neural network Zhao et al. (2019), Goodfellow et al. (2016), trained on the labelled catalogue, specialized in semantic segmentation of images, that takes a new cropped photo of a dolphin's fin (new photo means that it is not collected in the labelled catalogue) and returns a mask that depicts the outline of its fin (see Fig. 2). Other tasks implemented in the contour extraction module, are oriented to find the fin tip and homogenise all images, meaning that the fin tip must be always found on the left side of all the photos (see Fig. 7). When fin tip is originally found on the right side, the orientation of the fin is reversed by the operations listed in the module. This operation is essential because it makes comparable the contours, regardless of which side of the fin was photographed. The design implemented in this module is original and never adopted before in the literature.

Once fin outline of the new dolphin has been obtained, the matching module is designed to compare the unknown fin with all models collected the referring catalogue, returning an ordered list of the best matches. The comparison between two fins is based on the calculation of the overlap error between their outlines, which defines a measure of dissimilarity between the fins. Valid methods to compare fins, already used in Stanley (1995), were tested and exploited in the matching module of ARIANNA methodology. Hence, basic metrics employed in this module are not original, however the effort done in the comparison of the advantages in their applications is original and add something new to the current knowledge on the field of application.

The experiments were conducted on images depicting known specimens of Risso's dolphins, which were collected between 2013 and 2020 in the Gulf of Taranto (North Ionian Sea, Central-eastern Mediterranean Sea) during marine research expeditions, and the outcomes of these experiments are reported in section 3. Finally, the performances of ARIANNA and finFindR were compared on these real-life scenario images, using all the photos acquired during the expeditions and containing a dolphin dorsal fin, where no selection of photos, based on image quality, was done.

2. Materials and methods

2.1. Survey area and data collection

The Gulf of Taranto is situated in the North Ionian Sea (Central-eastern Mediterranean Sea) and extends from Santa Maria di Leuca to Punta Alice, covering an area of 14000 km². The data collection used in this work contains manually cropped photos of dolphin fins from full-frame images acquired in the area of study between 2013 and 2020 by experts, and consists of the following:

- Catalogue A: 1745 cropped fin photos of 84 different known specimens, each detected by an identifying name (see Table 1), captured from 2013 to 2019. A binary mask that highlights the outline of the fin has been manually extracted for each photo of the catalogue: in these masks if a pixel corresponds to the fin outline, it is coloured in white; otherwise, it is coloured in black (see Fig. 2).



Fig. 2. (a) Original cropped photo of a dolphin fin. (b) Binary mask extracted from the fin in the panel (a).

Table 1
Number of Photos Provided for Each Dolphin in Catalogue A.

Alessandro	17	Frangetta	104	Peroni	8
Alex	13	Fulmine	12	Pi	9
Alt	47	Galassia	34	Pinna	15
Ariete	11	Gap	19	Prezzemolo	17
Arrow	36	Gianluca	13	Puntino	12
Ben	13	Golf	11	Quattro	11
Black	20	Graffio	61	Reflex	6
Caos	33	Grigio	13	Ric	11
Carl	23	Happy	7	Saturo	5
Carlucci	9	Hugo	39	Scivolo	16
Carmelo	28	Ildritto	5	Sette	16
Charlie	6	Ipsilon	10	Seven	6
Cometa	24	Jax	43	Smile	14
Crocifisso	18	Jhonatan	23	Smilzo	9
Cupido	19	Jova	8	Spada	10
Dalmata	39	Kappa	10	Spritz	10
Dali'	37	Kateto	42	Squalo	14
Delta	24	Luna	5	Surf	13
Divergo	6	Macete	12	Svirgolo	16
Dubbio	17	Mario	13	Thor	20
Duepunti	62	Massimo	6	Ti	17
Ele	19	Meno	7	Tigre	12
Elica	32	Meteorite	12	Tris	11
Emme	21	Morse	8	Vito	28
Erard	39	Nerino	6	Wave	51
Ez	5	Niente	13	Wolverine	19
Falco	11	Peggy	8	Zante	28
Foglia	150	Peroncino	10	Zebra	8

Table 2
Number of Photos Provided for Each Dolphin in Catalogue B.

Cometa	11	Meno	1	Smile	9
Crocifisso	4	Morse	9	Smilzo	2
Ele	2	Nerino	2	Vito	3
Falco	1	Niente	12		
Gap	11	Reflex	7		

- Catalogue B: 74 cropped fin photos of 13 different specimens, captured in the same study area in 2020 (see Table 2).

2.2. Methodology

The proposed methodology has been summarized in Fig. 3, and all the algorithms were built using MATLAB (MathWorks, Natick, MA). ARIANNA consists of two original modules: the contour extraction module and the matching module, which will be described in the following. The user of this system must provide a referring catalogue K of labelled cropped fin photos of known dolphins and a catalogue U of cropped fin photos of the unknown specimens that he wants to identify.

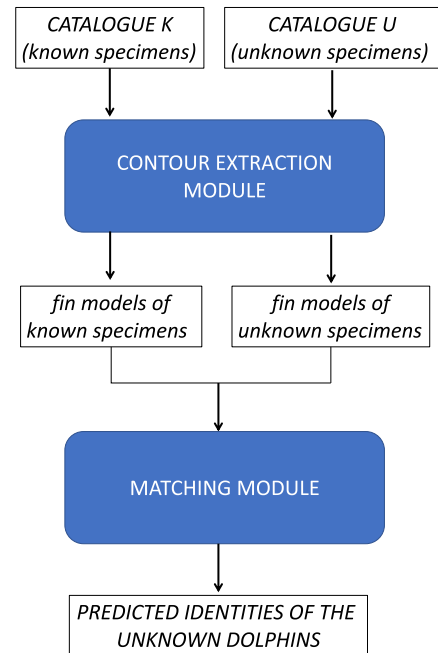


Fig. 3. Flow chart of ARIANNA photo identification system.

2.2.1. Contour extraction module

The catalogues K and U are given as the input to the contour extraction module which performs two different tasks. Firstly, every cropped photo passes through a neural network to create a binary mask that highlights all pixels corresponding to the fin contour, for each original image. Then, the software uses this mask to create a model of the fin outline for the matching phase. The models created for the fins in catalogues K and U are presented in input into the matching module.

The contour extraction module can be divided into the following 4 steps:

- step 1: Extraction of the fin outline with neural networks
- step 2: Building of the fin outline model
- step 3: Find the fin tip
- step 4: Rotate right-oriented fins

A description of each step follows.

Step 1: Extraction of the fin outline with neural networks

The user must feed into the system a cropped photo of a dolphin fin, as shown in Fig. 2. In the first step of this module, the original photo of

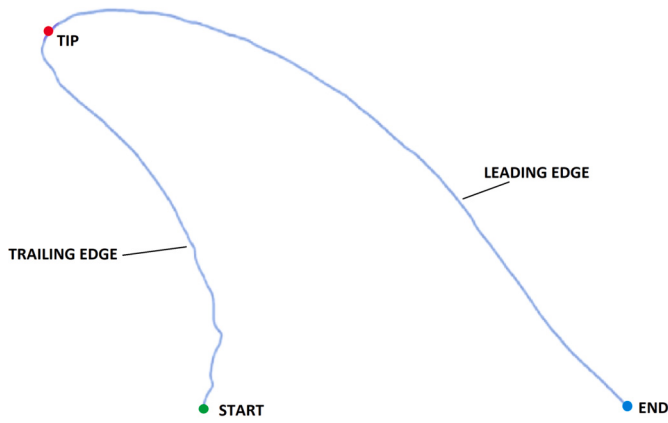


Fig. 4. Example of a fin outline with the start point, tip and end point used in the proposed model.

the fin is input into a neural network specialized in extracting of the fin outline of Risso's dolphins using semantic segmentation. This algorithm processes images to create masks, where pixels are sorted into different classes. We used RefineNet Lin et al. (2017), a learning network that uses a sequential architecture to exploit features from images at multiple level of abstraction for high-resolution semantic segmentation. We used 75% of the photos provided by catalogue A as the training set and the remaining 25% for the validation set with their corresponding binary mask. The classes for the semantic segmentation were *contour* for pixels corresponding to the fin outline (white) and *background* for all other pixels (black).

Step 2: Building of the fin outline model

Starting from the binary mask of the fin outline, we extracted the coordinates of each white pixel in the mask to create a processable model of the fin outline in the matching phase. This model will be a vector of ordered coordinates: the first item (*startpoint*) contains the coordinates of the pixel that correspond to the beginning of the trailing edge of the fin; the last item (*endpoint*) contains the coordinates of the pixel that correspond to the end of the leading edge of the fin, as shown in Fig. 4. In the model, the pixel that corresponds to the fin tip is also identified. The trailing edge of a dolphin dorsal fin goes from the back of the fin to its tip, and the leading edge goes from the tip to the front of the fin.

The algorithm that builds the ordered vector is called *getCoordinates* and can be divided into 5 phases:

1. we isolate the largest white section in the mask using the MATLAB-function *bwareafilt* MATLAB (2023a), to avoid considering white pixels scattered in the image (see Fig. 5a);
2. the remaining white section is thinned using the MATLAB function *bwmorph* MATLAB (2023b) to create a single line of white pixels corresponding to the fin outline (see Fig. 5b);
3. using the MATLAB function *find* MATLAB (2023c) we obtain a vector that contains the coordinates of all white pixels in the mask, randomly ordered;
4. to order those coordinates, we start by searching for the start point: it is the pixel with the minor value of the y coordinate in the left section of the mask;
5. once we have the start point, we can start building the ordered vector: the first item i is the *startpoint*, and item $i + 1$ is the pixel in the vector that is the closest to item i in terms of coordinates x, y going up to the fin tip and then down till the *endpoint* (see Fig. 5c). If there are still pixels to analyse in the original vector but it is impossible to find a pixel closer than 5 pixels to the current point it means that the mask contains a gap, so it is discarded.

Step 3: Find the fin tip

After obtaining the ordered vector of the coordinates of the fin outline, we must identify the one of that corresponds to the fin tip. For this purpose, we use another algorithm called *FinTip*, for the upper side of the mask which is input to *getCoordinates* to obtain an ordered vector of N white pixels in this section of the mask. We can divide the *FinTip* process into 4 phases:

1. three items have been selected in the ordered vector:

- A: first item
- B: item in position $pos(A) + N/\alpha$
- C: item in position $pos(A) + 2N/\alpha$

where the coefficient α was empirically chosen equal to 7. The idea is dividing the ordered vector of N white pixels into α equal parts, and to select the points B and C as previously written. If α is small, i.e. equal to 2 or 3, then N/α is large and items B and C may be too far from A, with the risk that B cross the fin tip; otherwise, if α is large, i.e. equal to 20, then N/α is small and items A, B and C may be too close together. The chosen value of $\alpha = 7$ is a compromise between the two extreme conditions just described. Thus, we ensure that the offset between the three points is proportional to the number of points in the vector (see Fig. 6a).

2. we calculate and store the value of

$$x = |AC - AB| + |AC - BC|, \quad (1)$$

where, AC, AB and BC are scalar quantities, each one corresponding to the distance between two points, i.e. AC is the distance between point A and point C. When we find a smaller value of x , point B is on the tip of the fin (see Fig. 6b).

3. we update the position of the three points: the new point A is the point after the current A, so the positions of points B e C are updated.
4. we repeat step 2 and 3 until we analyse all points in the vector, keeping track of the configuration in which the value x is smaller. Finally, point B of the configuration that gives the lower value of x is returned as the fin's tip.

Step 4: Rotate right-oriented fins

To homogenise all data, we must be sure that all photos are oriented with the fin tip on the left: for this reason, we must check if the fin we are working on is correctly oriented with the fin tip on the left, so we can overlap it with all fins of catalogue K without problems. After obtaining the ordered vector and the fin tip, we check the fin orientation: first, we calculate m , the average between the coordinate x of the first point of the contour and the coordinate x of the last one; if the coordinate x of the fin tip is less than m it means that the fin is correctly left-oriented; otherwise if the coordinate x of the fin tip is greater than m , it means that the fin is right-oriented so we must automatically rotate it. We can see two examples in Fig. 7: in the plots the green line marks the x coordinate of the first point of the contour, the blue line marks the x coordinate of the last point of the contour, the yellow line marks the m value and the red line marks the x coordinate of the fin tip. In the first case (Fig. 7a) the red line is to the left of the yellow line, and the actual fin tip is on the left of the image; in the second case (Fig. 7b), the red line is on the right of the yellow one, the tip is actually on the right, and the fin must be rotated.

2.2.2. Matching module

In the matching module, the outline of every unknown fin from catalogue U is compared with every fin outline model from catalogue K. The comparison is based on the calculation of the overlap error between the two outlines, which will define a measure of dissimilarity between the fins. Finally, the software uses these errors to create an ordered list of specimens that best match the fin of the unknown dolphin.

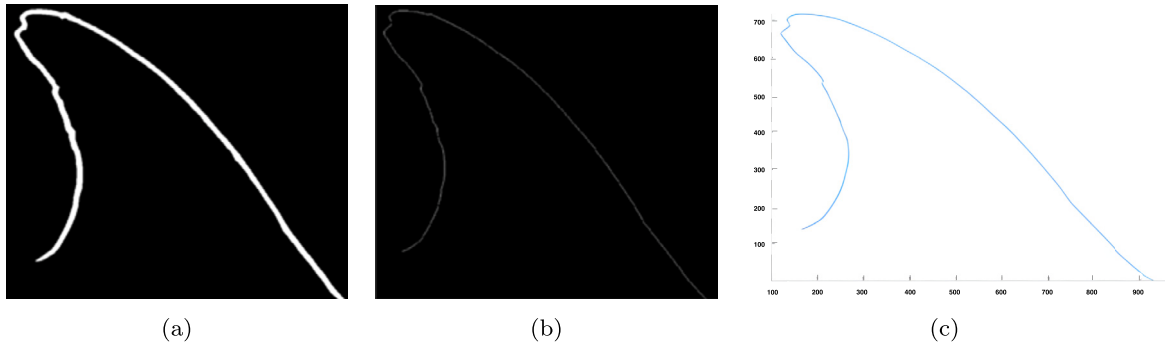


Fig. 5. (a) Original binary mask of the dolphin fin. (b) Mask after the thinning phase. (c) Plot of the extracted ordered vector.

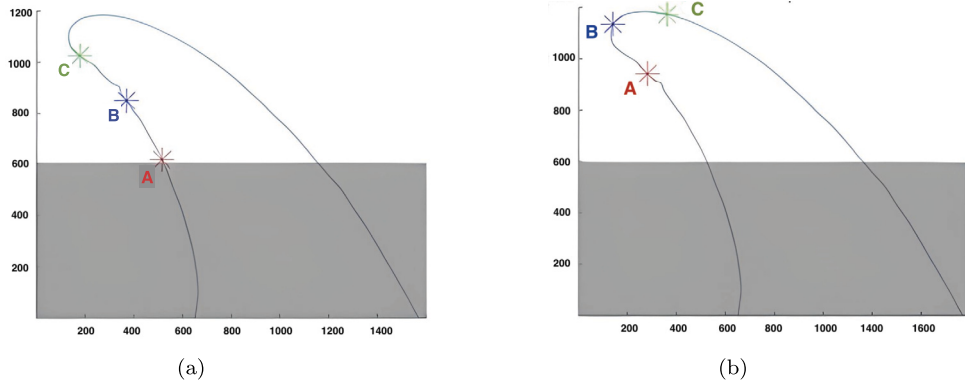


Fig. 6. (a) First points A, B and C. (b) Configuration in which point B corresponds to the fin tip.

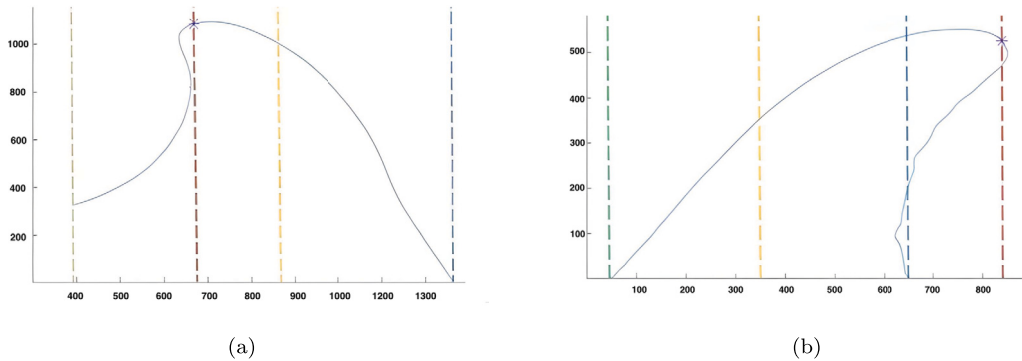


Fig. 7. (a) Fin correctly oriented with the tip on the left side. (b) Fin incorrectly oriented with the tip on the right side.

For the matching phase of ARIANNA some methods, which are inspired by methods used in the literature Stanley (1995), have been used and tested, leading to a matching process divided into 3 steps. The first step is devoted to a gradual reduction of the length of the trailing edges of the two fins, in order to find their best overlap; the second step serves to the mapping of the unknown fin outline on the outline of the known fin; in the last step the dissimilarity, as an overlap error, is computed. In the following a detailed description of the 3 steps is provided.

Step 1: Gradually reduce the length of the trailing edges

The first step takes its cue from the *TRIM FIXED PERCENT* method, which was proposed in Stanley (1995): it establishes that the error between two fins must be calculated multiple times to gradually reduce the length of the trailing edges of the two fins by 5% of their length, in order to find the best overlap. Thus, it is also possible to identify fins when the photo quality is low, e.g., when the fins are cropped or covered by waves, splashes of water or other disturbing elements. We adapted this idea to our methodology by executing steps 2 and 3 for 7 different times to return smaller overlap error.

Step 2: Map the unknown fin on the catalogue fin

Before calculating the overlap error between two fins, we must map the unknown fin outline on the outline of the catalogue fin to obtain their best overlap. The method *mapContour* chooses 3 points on the fins and exploits them to create two linear systems that are used in the mapping Stanley (1995). The 3 chosen points are: the point that corresponds to the most prominent notch on the trailing edge; the first point of the trailing edge (the start point in *getCoordinates*) and the last point of the leading edge (the end point in *getCoordinates*). In our work we decided to use the fin tip instead of the most prominent notch, since the fin tip is much easier to find and leads to a good overlap too. Thus, the mapping between the unknown fin outline (U) and the known fin outline (K) is performed by using these 3 points:

- $tip(F)$: tip of the fin F
- $start(F)$: first point of the trailing edge of fin F
- $end(F)$: last point of the leading edge of fin F



Fig. 8. Two different fins before (a) and after (b) mapping.

The linear systems are:

1.

$$tip(K)_x = A * tip(U)_x + B * tip(U)_y + C \quad (2)$$

$$start(K)_x = A * start(U)_x + B * start(U)_y + C \quad (3)$$

$$end(K)_x = A * end(U)_x + B * end(U)_y + C \quad (4)$$

2.

$$tip(K)_y = D * tip(U)_x + E * tip(U)_y + F \quad (5)$$

$$start(K)_y = D * start(U)_x + E * start(U)_y + F \quad (6)$$

$$end(K)_y = D * end(U)_x + E * end(U)_y + F \quad (7)$$

These systems are solved with the Gauss-Jordan elimination method to find six coefficients (A, B, C, D, E, F) to recalculate the coordinates x' and y' for every point (x, y) of the unknown fin outline:

$$x' = A * x + B * y + C \quad (8)$$

$$y' = D * x + E * y + F \quad (9)$$

Fig. 8 shows an example of two mapped fins.

Step 3: Calculate the dissimilarity

Once the unknown fin outline has been mapped on the outline of the catalogue fin, their dissimilarity can be estimated. This step exploits the method *meanSquaredErrorBetweenOutlineSegments* proposed in Stanley (1995) with a minor change: the dissimilarity is computed as an overlap error on the entire fin contour (trailing edge and leading edge); here, only the trailing edge is considered. This choice was made for the following reasons:

- the trailing edge contains much more distinctive features than the leading edge;
- in most cases, the leading edge is very similar among different specimens, so its consideration may be useless and time-consuming;
- is very difficult to have a perfect overlapping, so even if two leading edges are identical, they may not be perfectly overlapped and the error will increase.

The first phase of the *meanSquaredErrorBetweenOutlineSegments* method requires calculating a middle outline between two trailing edges: every point of this middle outline is obtained by scanning every segment of the two fins and choosing a middle point between them. This middle outline is used to find the pairs of points on the two analysed outlines between which the overlap error is calculated. From now on, we use U to denote the trailing edge of the unknown fin and with K to denote the trailing edge of the known fin from the catalogue. At every iteration we

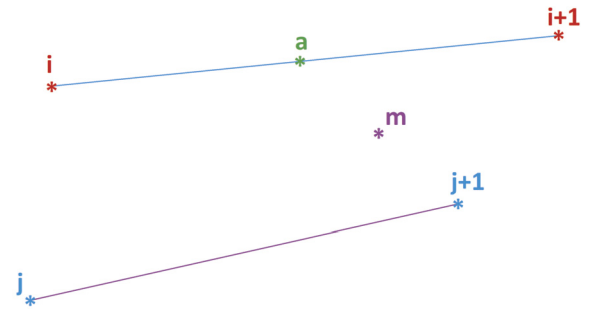


Fig. 9. Graphic illustration of one step of the process of creating the middle outline.

consider: the segment between point i and point $i+1$ on contour U , and the segment between point j and point $j+1$ on contour K . Indices i and j start from 1 and increase until the end of the trailing edges. The aim is to find point a on the current segment of U to obtain a new point m , whose coordinates are the average between the coordinates of point a and the coordinates of point $j+1$, which is part of the medium outline (see Fig. 9).

The coordinates of point a are:

$$x = x_i + [s * (x_{i+1} - x_i)] \quad (10)$$

$$y = y_i + [s * (y_{i+1} - y_i)] \quad (11)$$

In these equations the value s is calculated as the ratio between the length of the analysed contour portion and the length of the current segment on contour U . The process of creating the middle contour continues until we reach the end of one outlines.

Fig. 10 shows a section of two analysed contours: U in blue (trailing edge of the unknown fin) and K in orange (trailing edge of the known fin from the catalogue). If we consider the segment of U between points $u1$ and $u2$ and the segment of K between points $k1$ and $k2$, we can see the yellow starred point m , whose coordinates are the average between the coordinates of points a and $k2$.

Once the middle outline is obtained, it is used to find n pairs of points on the two original outlines between which we will calculate the overlap error between two original fins. The entire process is divided into multiple iterations that simultaneously loop on the three outlines (unknown fin, known fin and middle outline).

Let us consider the following indices:

- m , which loops through the points of the previously calculated middle outline;
- i , which loops through the points of the trailing edge of the unknown fin;
- j , which loops through the points of the trailing edge of the known fin from the catalogue.

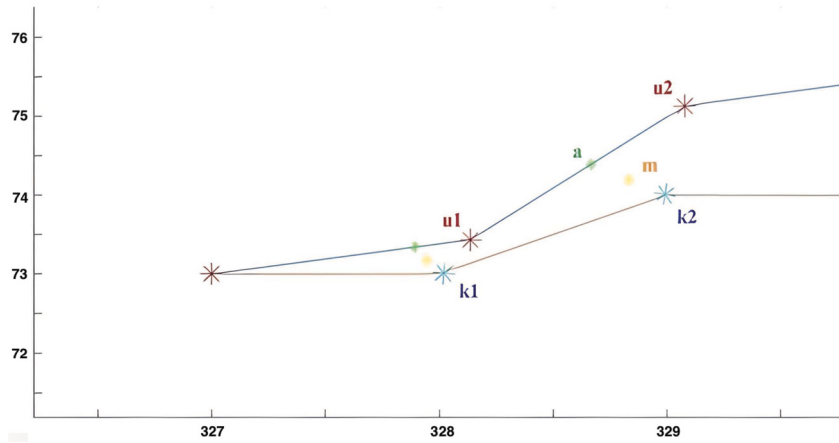


Fig. 10. Section of two contours and their middle outline.

In every iteration, we will search a pair of points (u, k) , where u is a point on the trailing edge of the unknown fin between indices i and $i - 1$, and k is a point on the trailing edge of the known fin between indices j and $j - 1$. The coordinates of these two points are calculated so that at the end, it is possible to trace a straight line between points u , k and m . At the end of every iteration, we increase the number of pairs found $(n + 1)$ and add to the sum of errors the distance calculated between points u and k . After retrieving all pairs, the measure of dissimilarity between the two fins analysed is the ratio between the sum of errors and the number of pairs found (mean squared error). To find point u on the trailing edge of the unknown fin between indices i and $i - 1$, we calculate:

$$dot1 = (x_{i-1} - x_m) * mdx + (y_{i-1} - y_m) * mdy \quad (12)$$

$$dot2 = (x_i - x_m) * mdx + [(y_i - y_m) * mdy] \quad (13)$$

The variables mdx and mdy are the distances between the coordinates x and y of points $m + 1$ and $m - 1$, respectively.

The values $dot1$ and $dot2$ will tell us which segment of the unknown fin we must find point u : if they are both positive, we are considering a segment of the unknown fin that is too far ahead of point m , so we must decrease index i until we reach the beginning; if they are both negative, we are considering a segment of the unknown fin that is too far behind point m , so we must increase index i until we reach the end. There is correspondence between point m and segment between points i and $i - 1$ only if the signs of $dot1$ and $dot2$ are opposite, so we can continue and find point u , whose coordinates are:

$$x_u = (beta * dx) + x_{i-1} \quad (14)$$

$$y_u = (beta * dy) + y_{i-1} \quad (15)$$

Variables dx and dy are the distances between the coordinates x and y of point i and point $i - 1$, respectively; $beta$ is calculated as:

$$beta = - \frac{mdx * (x_{i-1} - x_m) + mdy * (y_{i-1} - y_m)}{mdx * dx + mdy * dy} \quad (16)$$

Once point u is obtained, the same process is executed to find point k on the trailing edge of the known fin from the catalogue, iterating on index j instead of index i .

Fig. 11 shows a graphic illustration of the process: the trailing edge of the unknown fin is in blue, the middle outline is in light blue, the trailing edge of the known fin from the catalogue is in orange, and all pairs (u, k) are starred in red. Thus, we can clearly see the straight lines that can be traced between each triplet of point u , k and m .

3. Experiments and results

All data were analysed using MATLAB (MathWorks, Natick, MA).

3.1. Neural network training

To train a neural network, which provides a binary mask of the original cropped photo of the dolphin fin, we exploited the learning algorithm provided by RefineNet: starting from a *training set* and a *validation set* provided by the user, this algorithm generates n different networks at 5 epochs from each other trained on the *training set*, and evaluates their performance on the *validation set*, which is provided by calculating the value of *intersection over union*, that is the ratio of the overlapping area of the ground truth and predicted area to the total area. We gave RefineNet 1345 photos from catalogue A (approximately 75% of the entire catalogue) as *training set* and the remaining 400 photos as *validation set*, and stopped the learning algorithm at $n = 150$. Among the 30 neural networks obtained, the one generated at epoch 80 was the best, since it provided the best value for the *IoU* metric (0,894) (see Fig. 12).

3.2. Results of ARIANNA matching phase

To evaluate the performance of the ARIANNA system, a test was performed using catalogue A as the dataset of photos of known specimens and catalogue B as the dataset of photos of dolphins to identify. Thus, we can see how the software reacts when the unknown fins are slightly different from the known ones, since fins can change over time and have slightly different features in the outline. Considering the measures of dissimilarity calculated for each unknown fin, we used 4 different metrics to evaluate the final prediction of the software:

- *metric 1*: average of the measures of dissimilarity obtained with all photos provided for each known specimen;
- *metric 2*: median of the measures of dissimilarity obtained with all photos provided for each known specimen;
- *metric 3*: average of 3 fewer measures of dissimilarity obtained with the photos provided for each known specimen;
- *metric 4*: average of 5 fewer measures of dissimilarity obtained with the photos provided for each known specimen.

Each metric provided a list of predictions for every unknown specimen ordered by the metric value: thus, the lowest values of the metrics correspond to the best prediction.

In Table 3, we can see the percentage of times in which the system returned the correct identity of the unknown dolphin as the 1st item of the ordered list in the top 3 positions, top 5 and top 10 using the 4 different metrics.

The success rates significantly improve when the output list is ordered by metric 3, i.e., the average of the less 3 measures of dissimilarity obtained with the photos provided for each known specimen.

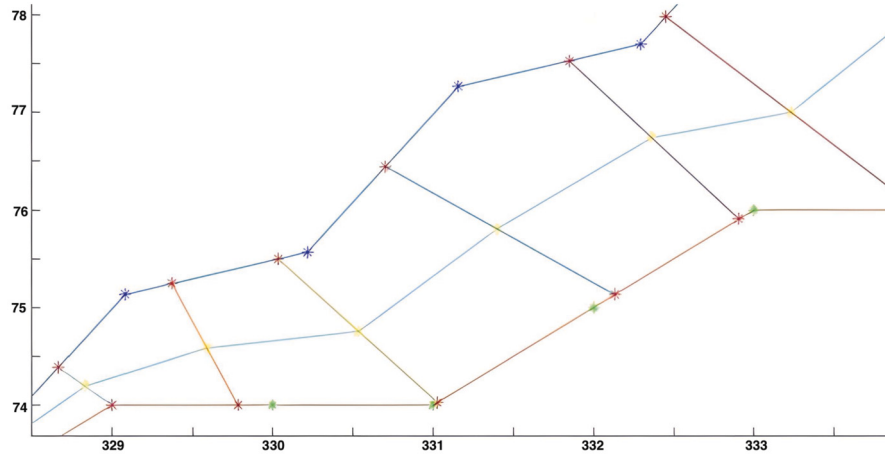


Fig. 11. Graphic illustration of the pairs of points found on two contours.

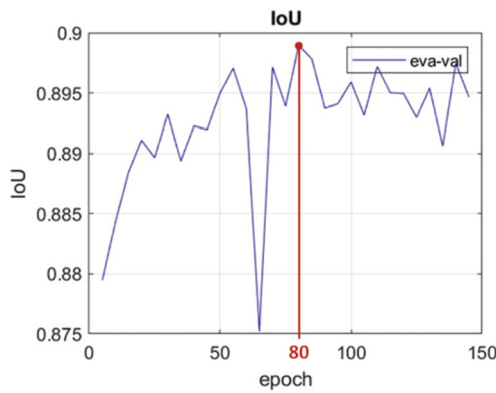


Fig. 12. Value of the intersection over union for the neural networks generated by RefineNet while trained on catalogue A.

Generally, we can see how the system ensures with a medium-high range of probabilities that the correct identity of the unknown dolphin is returned in the first 10 position of the lists. However, the success rates significantly decrease if we ask the system to return the correct identity in an higher position, especially in the 1st place. The reason is that despite the distinctive marks on the fin, many specimens have very similar trailing edges, so if we add this problem to the inherent margin of error due to the fully automatic process, specimens with less pronounced features can easily be mistaken with other specimens, so the correct identity can be exceeded by similar dolphins and have lower ranks. In addition to these problems related to the natural fin shape and automatic extraction of the contour, there are problems related to the quality of the photos in the process, which can often be blurry, have a distant and distorted perspective and be covered by other animals, boat parts, waves and splashes of water.

After analysing the results of various tests, we notice that similar to the test shown here, the 3rd metric always tends to lead to the best prediction, so it has been chosen as the final metric of the system for the output list of predicted identities.

3.3. Comparison with finFindR

To compare the performance of our system with an existing one, we chose to test one of the most used platforms, specialised in dolphin photo identification using fin contour, finFindR Thompson et al. (2021), FinFindR (2018), on the same catalogues used in chapter 3.2 (catalogue A as known specimens and catalogue B as specimens for identification).

The operation of finFindR is similar to the one described in this paper: it finds the trailing edge of dolphin fins in photos, calculates a score

Table 3

Success Rates of the Test of ARIANNA system.

Metric 1	
1 st place	7.34%
Top 3	21.10%
Top 5	33.94%
Top 10	54.13%
Metric 2	
1 st place	17.43%
Top 3	31.19%
Top 5	42.20%
Top 10	68.81%
Metric 3	
1 st place	34.86%
Top 3	56.88%
Top 5	66.97%
Top 10	91.74%
Metric 4	
1 st place	31.19%
Top 3	53.21%
Top 5	68.81%
Top 10	88.07%

in relation to the found features and produces a sorted list of likely identities from a catalogue of known individuals based on similar scores. The ARIANNA system provides a list of dolphin names after analysing all individual photos from the catalogue of known specimens, whereas the list provided by finFindR directly contains the file names of the photos from the catalogue of known specimens that have obtained the best match.

In Table 4 we can see the success rates in comparison: finFindR has slightly higher accuracy (+ 10%) when we request the correct identity in the first place of the ordered list, however its performances are significantly lower than ours when considering the first 3, 5 and 10 positions of the lists.

In more detail, in a real-life scenario, the ARIANNA system could return in 91.74% of cases the correct identity of the unknown dolphin in the top 10 places on the list, while in this case, the success rate of finFindR was only 67.57%. Moreover, finFindR could not recognise dolphin fins in 864 of 1745 photos of catalogue A, and it discarded them.

We should consider that the accuracy of 45.95%, achieved in the first place of the ordered list of predictions made by finFindR, is not applicable in photo identification studies, introducing a high error of predictions. Hence, it is preferable to opt for an ordered list of the first 10 predictions, such that provided by ARIANNA, where the user is almost certain, with an accuracy of 91.74%, of finding the correct identity

Table 4
Success Rates of finFindR and ARIANNA systems.

	ARIANNA	finFindR
1 st place	34.86%	45.95%
Top 3	56.88%	52.70%
Top 5	66.97%	56.76%
Top 10	91.74%	67.57%

of the dolphin. In this way, he can easily inspect the ordered list and assign the correct photo identification of the dolphin.

4. Conclusions

This work aimed to design, develop and test an automatic system to support researchers during the noninvasive photo identification of cetaceans. In the proposed system, an unknown dolphin is identified based on the automatic extraction of distinctive features on the edge of its dorsal fin, which are subsequently used to match with a set of known specimens. The experiments were performed on Risso's dolphins since in addition to the particular marks on the edge of their fin, these dolphins are an endangered species that must be monitored. ARIANNA uses as input the cropped dolphin fin images, which are obtained by original full-frame photos acquired during marine expedition. This is an essential step for all cetacean photo-ID algorithms Maglietta et al. (2022) and it is generally manually performed with a time-consuming process, especially in case of large dataset. However, a system devoted to the automated cropping of dolphin dorsal fin has been presented in the recent literature: it combines an image pre-processing algorithm with a Convolutional Neural Network classifier, obtaining an accuracy of 92% Reno et al. (2020). This system can be effectively combined with ARIANNA, obtaining a fully automated pipeline for the cetacean photo-ID starting from the full-frame images. The fin outline was extracted using semantic segmentation techniques applied to cropped photos of the dolphin fins. Starting from this contour, we tested and exploited several methods to match the unknown dolphin's fin with a catalogue of known specimens, based on the calculation of the overlap error among different outlines. The final step of the photo identification system provides an ordered list of the specimens that best match the unknown fin and guarantees with medium-high percentages (80% - 90%) that the correct identity of the unknown dolphin is in the first 10 positions of this list. The presented work is a milestone for future developments of photo identification techniques based on the extraction of contour features with semantic segmentation. Future studies should be focused on the application of ARIANNA to other marine specimens such as other dolphin species, sharks and whales. A weakness of Arianna and finfindR is that they are unable to predict dolphin identity in the first place of the ordered list of predictions, with good accuracy. At the best of our knowledge, in the literature this goal has not been achieved by any fin contour-based photo identification algorithm, and further studies should be devoted to this aim. Moreover, a limitation of ARIANNA is that it uses only the information contained in the fin outline of dolphins and it does not consider evaluable information contained over the fin. For example, in 2018 a fully automated tool for the photo identification of Risso's dolphin, called SPIR Maglietta, Renò, et al. (2018), has been published. It uses distinctive marks on the dorsal fin of Risso's dolphin, whose natural scars are particularly evident in adult individuals. Further development should be focused on the integration of SPIR and ARIANNA, aiming to provide a novel method for the Risso's dolphin photo identification based both on natural marks over the fin and over the fin outline. This method could be very useful also for other species of marine mammals.

CRediT authorship contribution statement

Rosalia Maglietta: Conceptualization, Methodology, Supervision, Writing – review & editing. **Arianna Bussola:** Data curation, Software, Validation, Writing – original draft. **Roberto Carlucci:** Writing – review & editing. **Carmelo Fanizza:** Data curation, Resources. **Giovanni Dimauro:** Conceptualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors thanks to Michele Attolico for technical assistance and Emanuele Seller for valuable discussions on experimental aspects of the paper.

References

- de Mello, A., Molina, J., Kajin, M., & Santos, M. (2019). Abundance estimates of guiana dolphins (*sotalia guianensis*; van b  n  den, 1864) inhabiting an estuarine system in southeastern brazil. *Aquatic Mammals*, 45, 56–65.
- Degrati, M., Coscarella, M., Crespo, E., & Dans, S. (2018). Dusky dolphin group dynamics and association patterns in peninsula valdes, argentina. *Marine Mammal Science*, 35, 416–433.
- Degrati, M., Loizaga, R., Coscarella, M., Sueyro, N., Crespo, E., & Dans, S. (2020). Integrating multiple techniques to estimate population size of an impacted dusky dolphin's population in patagonia, argentina. *Frontiers in Marine Science*, 7, 289.
- Dinis, A., Alves, F., Nicolau, C., Ribeiro, C., Kaufmann, M., Ca  nadas, A., & Freitas, L. (2016). Bottlenose dolphin *tursiops truncatus* group dynamics, site fidelity, residency and movement patterns in the madeira archipelago (north-east atlantic). *African Journal of Marine Science*, 38, 1–10.
- Estrade, V., & Dulau, V. (2020). Abundance and site fidelity of bottlenose dolphins off a remote oceanic island (reunion island, southwest indian ocean). *Marine Mammal Science*, 36, 871–896.
- FinFindR (2018). Retrieved from <https://github.com/haimeh/finFindR>.
- Frau, S., Ronchetti, F., Perretti, F., Addis, A., Ceccherelli, G., & Manna, G. L. (2021). The influence of fish farm activity on the social structure of the common bottlenose dolphin in sardinia (italy). *PeerJ Computer Science*, 9, Article e10960.
- Friday, N., Smith, T., Stevick, P., & Allen, J. (2000). Measurement of photographic quality and individual distinctiveness for the photographic identification of humpback whales, megaptera novaeangliae. *Marine Mammal Science*, 16, 355–374.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. Retrieved from <http://www.deeplearningbook.org>.
- Kiszka, J., Braulik, G., Reeves, R., & Lowry, L. (2018). *Grampus griseus*, iucn red list of threatened species, assessment information in detail. In *The IUCN Red List of Threatened Species* (pp. 1–13).
- Lin, G., Milan, A., Shen, C., & Reid, I. (2017). Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1–10).
- Maglietta, R., Bruno, A., Reno, V., Dimauro, G., Stella, E., Fanizza, C., Bellomo, S., Cipriano, G., Tursi, A., & Carlucci, R. (2018). The promise of machine learning in the Risso's dolphin *Grampus griseus* photo-identification. In *IEEE International Workshop on Metrology for the Sea*.
- Maglietta, R., Carlucci, R., Fanizza, C., & Dimauro, G. (2022). Machine learning and image processing methods for cetacean photo identification: A systematic review. *IEEE Access*, 10, 80195–80207.
- Maglietta, R., Ren  , V., Caccioppoli, R., Seller, E., Bellomo, S., Santacesaria, F., Colella, R., Cipriano, G., Stella, E., Hartman, K., Fanizza, C., Dimauro, G., & Carlucci, R. (2020). Convolutional neural networks for risso's dolphins identification. *IEEE Access*, 8, 80195–80206.
- Maglietta, R., Ren  , V., Cipriano, G., Fanizza, C., Milella, A., Stella, E., & Carlucci, R. (2018). Dolfin: an innovative digital platform for studying risso's dolphins in the northern ionian sea (north-eastern central mediterranean). *Scientific Reports*, 8, Article 17185.
- Matlab (2023a). Extract objects from binary image by size (bwareafilt). MathWorks.
- Matlab (2023b). Morphological operations on binary image (bwmorph). MathWorks.
- Matlab (2023c). Find indices and values of nonzero elements (find). MathWorks.

- Paschoalini, V., & Santos, M. (2020). Movements and habitat use of bottlenose dolphins, *tursiops truncatus*, in south-eastern brazil. *Journal of the Marine Biological Association of the UK*, 100(4), 1–12.
- Reno, V., Losapio, G., Forenza, F., Politi, T., Stella, E., Fanizza, C., Hartman, K., Carlucci, R., Dimauro, G., & Maglietta, R. (2020). Combined color semantics and deep learning for the automatic detection of dolphin dorsal fins. *Electronics*, 9(5), 758.
- Samuelson, M., Fujiwara, M., Pulis, E., Pitchford, J., Howard, V., & Solangi, M. (2021). Comprehensive evaluation of survival and population growth for common bottlenose dolphins (*tursiops truncatus*) in the mississippi sound, usa, following the deepwater horizon oil spill. *Southeastern Naturalist*, 20, 178–191.
- Stanley, R. (1995). Darwin: Identifying dolphins from dorsal fin images. Senior Thesis Eckerd College.
- Thompson, J., Zero, V., Schwacke, L., Speakman, T., Quigley, B., Morey, J., & McDonald, T. (2021). finfindr: Automated recognition and identification of marine mammal dorsal fins using residual convolutional neural networks. *Marine Mammal Science*, 1–12.
- Urian, K., Hohn, A., & Hansen, L. (1999). Status of the photo-identification catalog of coastal. bottlenose dolphins of the western north atlantic: Report of a workshop of catalog contributors. NOAA Administrative Report NMFS-SEFSC-425, (p. 22).
- Urian, K., Waples, D., Tyson, R., Hodge, L., & Read, A. (2014). Abundance of bottlenose dolphins (*tursiops truncatus*) in estuarine and near-shore waters of north carolina, usa. *Journal of North Carolina Academy of Science*, 129(4), 165–171.
- Zhao, Z., Zheng, P., Xu, S., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3212–3232. <https://doi.org/10.1109/TNNLS.2018.2876865>.