Applied Soft Computing 141 (2023) 110315

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Review article

Semantic segmentation based on Deep learning for the detection of Cyanobacterial Harmful Algal Blooms (CyanoHABs) using synthetic images

Fredy Barrientos-Espillco [a,*], Esther Gascó [b], Clara I. López-González [b], María J. Gómez-Silva [a], Gonzalo Pajares [c]

[a] Department of Computer Architecture and Automation, University Complutense of Madrid, 28040 Madrid, Spain
[b] Department of Software Engineering and Artificial Intelligence, University Complutense of Madrid, 28040 Madrid, Spain
[c] Institute for Knowledge Technology, University Complutense of Madrid, 28040 Madrid, Spain

ARTICLE INFO

Article history:
Received 18 August 2022
Received in revised form 29 March 2023
Accepted 10 April 2023
Available online 18 April 2023

Keywords:
Cyanobacterial harmful algal blooms
Semantic segmentation
Generative adversarial network
Neural style transfer
Convolutional neural networks
Deep learning
Autonomous surface vehicles

ABSTRACT

Cyanobacterial Harmful Algal Blooms (CyanoHABs) in lakes and reservoirs have increased substantially in recent decades due to different environmental factors. Its early detection is a crucial issue to minimize health effects, particularly in potential drinking and recreational water bodies. The use of Autonomous Surface Vehicles (ASVs) equipped with machine vision systems (cameras) onboard, represents a useful alternative at this time. In this regard, we propose an image Semantic Segmentation approach based on Deep Learning with Convolutional Neural Networks (CNNs) for the early detection of CyanoHABs considering an ASV perspective. The use of these models is justified by the fact that with their convolutional architecture, it is possible to capture both, spectral and textural information considering the context of a pixel and its neighbors. To train these models it is necessary to have data, but the acquisition of real images is a difficult task, due to the capricious appearance of the algae on water surfaces sporadically and intermittently over time and after long periods of time, requiring even years and the permanent installation of the image capture system. This justifies the generation of synthetic data so that sufficiently trained models are required to detect CyanoHABs patches when they emerge on the water surface. The data generation for training and the use of the semantic segmentation models to capture contextual information determine the need for the proposal, as well as its novelty and contribution.

Three datasets of images containing CyanoHABs patches are generated: (a) the first contains real patches of CyanoHABs as foreground and images of lakes and reservoirs as background, but with a limited number of examples; (b) the second, contains synthetic patches of CyanoHABs generated with state-of-the-art Style-based Generative Adversarial Network Adaptive Discriminator Augmentation (StyleGAN2-ADA) and Neural Style Transfer as foreground and images of lakes and reservoirs as background, and (c) the third set, is the combination of the previous two. Four model architectures for semantic segmentation (UNet++, FPN, PSPNet, and DeepLabV3+), with two encoders as backbone (ResNet50 and EfficientNet-b6), are evaluated from each dataset on real test images and different distributions. The results show the feasibility of the approach and that the UNet++ model with EfficientNet-b6, trained on the third dataset, achieves good generalization and performance for the real test images.

Contents

* Corresponding author.
E-mail addresses: fredybar@ucm.es (F. Barrientos-Espillco), egasco@ucm.es (E. Gascó), claraisl@ucm.es (C.I. López-González), mgomez77@ucm.es (M.J. Gómez-Silva), pajares@ucm.es (G. Pajares).

https://doi.org/10.1016/j.asoc.2023.110315

1. Introduction

In recent years, Cyanobacterial Harmful Algal Blooms (CyanoHABs) have become a worldwide concern since they pose a threat to human health, animals, and aquatic ecosystems [1,2], causing also economic damages [3]. The occurrence of CyanoHABs is due to different environmental factors, such as cyanobacterial and cyanotoxin diversity, nutrient concentration, buoyancy regulation, light level, water temperature, and hydrological and meteorological conditions [1]. In recent decades CyanoHABs have rapidly increased in tropical/subtropical lakes and worldwide reservoirs [4].

Currently, there are different approaches and techniques for monitoring CyanoHABs, such as microscopy-based technique, chemistry, flow cytometry, enzyme-based assays (ELISA), DNA, and remote detection [5,6]. Each of these methods differs in its ability to monitor different targets (e.g., cell vs. toxin detection, or detection vs. identification) and also possesses different limitations during the detection phase [5]. Indeed, they are expensive, requiring a lot of time, labor, and knowledgeable professionals in the field, and with the proliferation of CyanoHABs in lakes and reservoirs around the world, they are no longer sufficient for monitoring.

Vision-based remote-sensing monitoring studies of CyanoHABs use satellite with hyperspectral imagery [7–11]. The study in [12] mentions that the most commonly used remote sensing spectral indices to identify CyanoHABs are: band ratio index, normalized difference vegetation index (NDVI), maximum chlorophyll index (MCI), and floating algae index (FAI). However, these types of methods present certain problems: satellite data are not easily accessible, they are expensive and only historical data can be obtained. Added to this is the problem that CyanoHABs emerge at the water surface unpredictably, depending on many out-of-control factors, as well as its capability to emerge and submerge several times a day. Therefore, there is a need to develop new accessible and inexpensive approaches for efficient monitoring of CyanoHABs that are adapted to the dynamic nature of these algae, and that are capable of detecting them when they appear on the water surface.

The use of Autonomous Surface Vehicles (ASVs) to measure water quality and track the proliferation of CyanoHABs in lakes and reservoirs is increasing. They can collect spatio-temporal dynamic information and allow a better understanding of CyanoHABs [13–15]. In addition, their detection helps ASVs to navigate toward Regions of Interest (RoIs) where blooms are located. This leads to unnecessary energy consumption and runs the risk of discharging the ASVs batteries (with a limited capability) very fast, before having completed the mission of tracking the CyanoHABs [16]. Machine Vision Systems (MVSs), onboard ASVs, are excellent tools to analyze images from the scene and to determine the presence of harmful algae or not. Images of RoIs with algae exhibit both, spectral and textural information, which is extracted by considering the spatial context of each pixel and its neighbors. Deep learning-based semantic segmentation methods, due to their convolutional nature, can simultaneously capture both types of information, which together with their high performance in multiple domains, make them good candidates to detect possible CyanoHABs patches in images from the ASV perspective. Textural differentiation represents a substantial contribution to the exclusive use of the chromatic indices mentioned above, representing an important contribution and novelty in this domain.

To validate the proposed semantic segmentation models, a powerful and representative dataset is always required. Two problems arise in this regard. On one hand, to the best of our knowledge, there is no publicly available dataset of images with CyanoHABs captured with MVSs onboard ASVs. On the other hand, collecting sufficient images using ASVs in reasonable time periods is a very difficult task. This is due to the unforeseen dynamic behavior of the algae. Indeed, their occurrence on the water surface over time and in certain time periods cannot be guaranteed. Therefore, even if ASVs equipped with MSVs are available, it is not possible to obtain sufficient data to train and validate semantic segmentation models in reasonable time periods, and it is even necessary to wait several years, with no guarantee of being able to do so.

In order to have as many CyanoHABs patches in images as possible, without needing to wait for them to appear in a more or less capricious fashion and for years, we got inspired by [17], which indicates that carefully artificially created data can provide results that are almost comparable to real data, and by applications of data augmentation in medical imaging [18–21] to propose the creation of several datasets of synthetic images in a suitably structured way to have deep-learning models available without having to wait for blooms to appear on the water surface.

The main contribution and innovation are the generation and comparison of three types of synthetic image datasets for evaluating semantic segmentation models based on Convolutional Neural Networks (CNNs):

(a) R-CyanoHABs, which is built by assembling real patches (foreground) on real lake and reservoir images (background), from aquatic environments. The real patches are extracted from real images, which are not part of this dataset or of real test images, containing CyanoHABs and manipulated by image augmentation strategies, including rotation and translation, before fusing them to the background images.

(b) S-CyanoHABs, which is built by assembling synthetic patches (foreground) on real lake and reservoir images (background) coming from aquatic environments. The synthetic patches are generated with Style-based Generative Adversarial Network Adaptive Discriminator Augmentation (StyleGAN2-ADA) [22] and refined with Neural Style Transfer [23,24]. We evaluate the quality of the synthetic patches using two different approaches: the Fréchet Inception Distance (FID, [25]) for quantitative analysis and the best-performing model on real test images for qualitative analysis.

(c) RS-CyanoHABs, which is inspired by [19,20,26,27] and consists of the combination of the whole previous two datasets obtaining a greater variety of data.

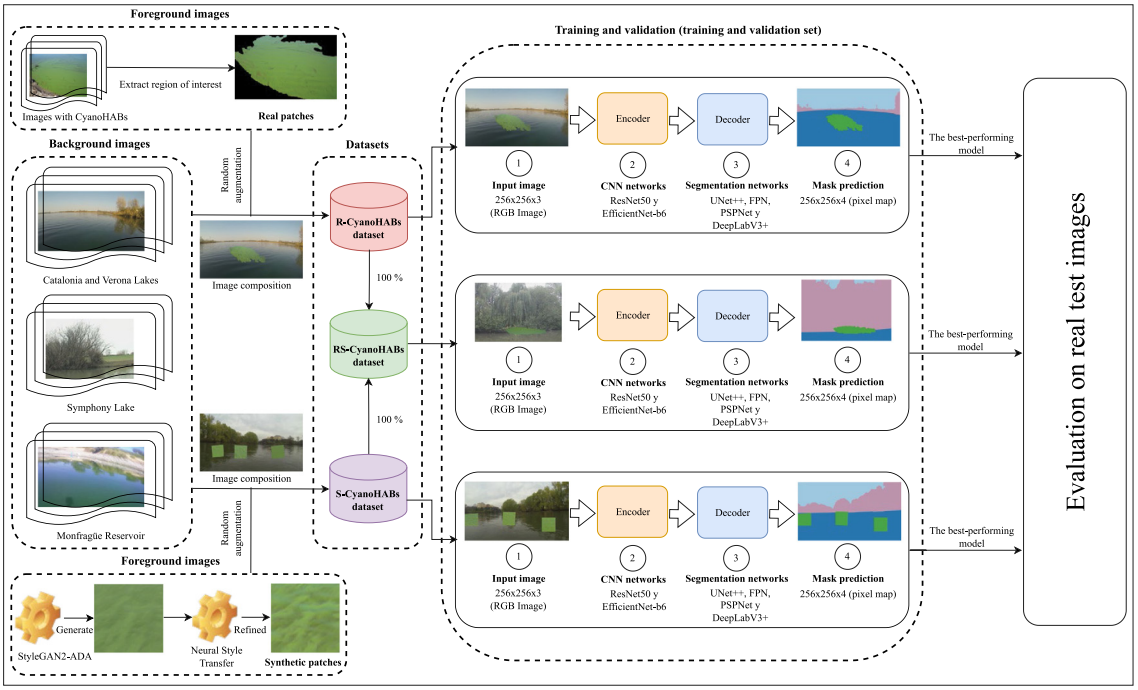


Fig. 1. Summary of the CyanoHABs detection approach from the ASV perspective using deep learning-based semantic segmentation methods.

Fig. 1 outlines this dataset generation process. In the middle part, the three datasets are shown, while in the upper and lower left-hand parts the extraction of real and synthetic patches respectively is represented. The background images, for the composition of new images with real and synthetic patches are displayed in the central part on the left, coming from two different lakes (Catalonia-Verona and Symphony) and a reservoir (Monfragüe).

Once the three datasets have been built, they are supplied to three different segmentation models for training and validation. An Encoder–Decoder approach is the strategy applied, where different specific models, based on CNNs, are defined as the backbone for the down-sampling Encoder and specific segmentation networks for the up-sampling Decoder, including mask prediction. The convolutional approach captures both, spectral and textural spatial contextual information, existing between a pixel and its neighbors in images with CyanoHABs patches and transmits and propagates it over the different resolution levels of Encoder–Decoder models, as indicated later in the description of methods, Section 2.3. This is the second innovation for the proposed approach. Which, together with the synthetic generation of data, constitute the heart and reason for the proposal.

Methods such as K-Means [28] or Fuzzy c-Means [29] are clustering techniques in which a data set is grouped into K or c clusters with every pixel in the dataset. They try to minimize variances or distances from points to the center of clusters assuming equal or similar significance of the spectral information (color) in the involved variables, which is too strong an assumption in the case of CyanoHABs, as can be seen in the illustrative example shown in Fig. 3(c) below, where the bloom is only identified considering at least two clusters, instead of one, as would be desirable. This represents another handicap since both methods require several clusters to be set, and this number is different depending on the incoming image to be segmented. In this regard, we want to obtain the widest possible generalization of the spatial–spectral and contextual characteristics of pixels in the patches, at the expense, obviously, of increasing the number of training samples considerably, which is the main objective of this work. These considerations can be extended to clustering methods other than the above, but which also essentially do not capture the above-mentioned spatial–spectral and contextual information. The best-performing model in each dataset has been evaluated on real test images.

In addition to the proposed strategy for the generation of the three datasets, which is the main contribution and novelty, the following additional contributions are also noteworthy:

- CyanoHABs detection from the ASV perspective using deep learning-based semantic segmentation methods in lakes and reservoirs.
- We demonstrate the feasibility of generating sufficiently realistic, diverse, and high-quality synthetic CyanoHABs patches using StyleGAN2-ADA and refined with Neural Style Transfer.
- We use the synthetic patches to obtain composite images and then train model architectures for semantic segmentation on these images, where, thanks to the convolutional concept, they can extract spatial spectral and contextual information. The generalization ability on real test images of the best model is almost comparable to that of the best model trained on images containing real patches.
- We compare the effect of two data augmentation techniques (basic augmentation and advanced augmentation based on StyleGAN2-ADA and Neural Style Transfer), where StyleGAN2-ADA and Neural Style Transfer show remarkable improvements in generalization performance.
- We perform the comparative analysis of state-of-the-art semantic segmentation model architectures with encoders on the three validation datasets. In addition, we evaluate the best-performing model from each validation set on real test images coming from different distributions.

The rest of the sections are mainly organized as follows. Section 2 presents the process for data generation, and the methods used for training and evaluation of the semantic segmentation models. Section 3 describes and discusses the experimental results obtained by the data generation and detection methods. Finally, Section 4 draws the main conclusions of this study.

Fig. 2. Images of lakes and reservoirs (background images), (a) Verona Lake (top) and Catalonia lake (bottom), (b) Symphony lake, (c) Monfragüe reservoir, shore (top), and open water (bottom).

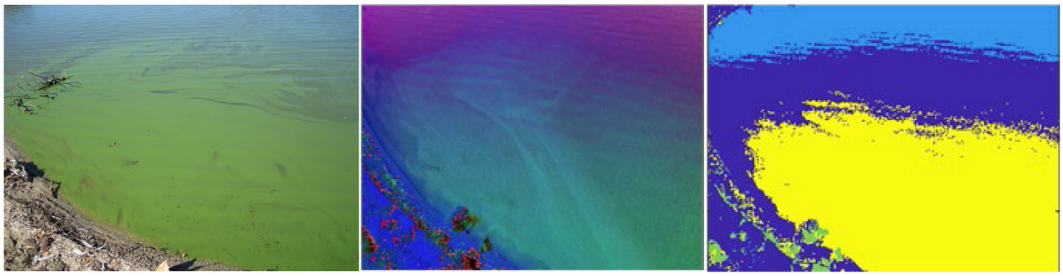


Fig. 3. Computing clusters centers and standard deviations, (a-b) RGB and HSV images, and (c) four clusters obtained with K-Means.

2. Data and methods

The full image segmentation process consists of two phases, which are described below, namely: (a) data preparation, where the three synthetic image datasets are generated and (b) definition of the semantic network models, including training, validation, and testing.

2.1. Data preparation

The generation of new images is based on the insertion of patches of CyanoHABs, which constitute what we call foreground images, on other real base images, which we call background images. Thus, the background images are real, and the foreground images are either real or synthetic. Real images, background, and foreground patches are conveniently selected and processed, and synthetic images (synthetic patches) are built with the generative model StyleGAN2-ADA and refined with Neural Style Transfer. Foreground and background images are conveniently combined to obtain new composite images.

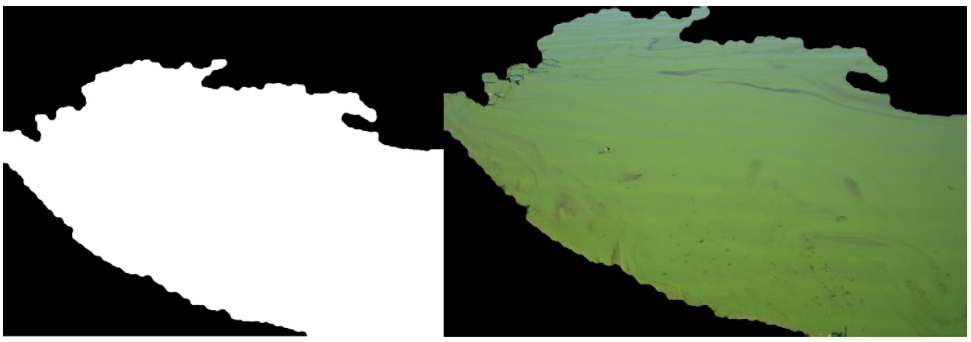
(*a*) *Background images*

They are real images of lakes and reservoirs captured by considering the image perspective projection and field of view of a camera onboard the ASV. These images serve as the basis for inserting CyanoHABs patches. They contain a broad variety of different scenes of aquatic environments coming from three lakes and one reservoir, collected at different times.

The IntCatch Vision dataset [30] is the first subset of background images. It contains several video sequences and sensor data from different lakes and rivers captured from an ASV. The IntCatch Vision project drives a paradigm shift in surface water quality monitoring and management. We select two video sequences belonging to Catalonia and Verona lakes respectively. The selected video sequences were captured at 60 Frames Per Second (FPS) and each frame has a width and height of 1920 × 1080 pixels. We extract randomly 200 frames, in their original sizes, from each video sequence. Fig. 2(a) shows two representative images from the first subset of background images.

The Symphony dataset [31] is the second subset of background images. It contains images of the shore of Symphony Lake, in Metz France. These images were captured by an ASV for more than three years between 2014 to 2017. We chose 400 images dated between January and December 2017, 32 images from each month approximately. The selected images contain different aquatic scenes of the shore during the four seasons of the year. Fig. 2(b) shows two representative images from the second subset of background images.

The third subset of background images comes from a reservoir located inside the national park of Monfragüe, Province of Cáceres, Spain (UTM 39° 49′03.0035″ N, 5° 56′, 14.3223″ W). We mounted a camera on an ASV to capture two video sequences from the shore and the open water respectively. Then, we randomly extract 200 frames from each video sequence. Finally, we crop the bottom part to exclude irrelevant information, such as the date and time of capture. The final size of the images is 2494 × 1410 pixels. Fig. 2(c) shows two representative images of this subset.



(a) (b)

Fig. 4. Real patch extraction, (a) image mask, and (b) cropped patch.

Table 1
Color boundaries in HSV space to obtain binary masks and patches from real images.

Parameter	Hue (H) ($m_H \pm \sigma_H$)	Saturation (S) ($m_S \pm \sigma_S$)	Value (V) ($m_V \pm \sigma_V$)
Lower	30	5	10
Upper	92	255	255

The final dataset of real background images is built by putting together the three subsets of background images, obtaining a total of 1200 images from different aquatic environments.

(*b*) *Foreground images*

The foreground images are the CyanoHABs patches and belong to two groups: real and synthetic patches. In the following, we detail the generation process of each type of patches.

(*b*.1) *Real patch extraction*

They are obtained from real images (one per image) containing accumulation of CyanoHABs located on the water surface of lakes or reservoirs. Fig. 3(a) displays a bloom that emerged and was pushed by the wind toward the shore. The extraction process is outlined as follows:

(1) Transform the original RGB (*Red*, *Green*, *Blue*) images to the HSV (*Hue*, *Saturation*, *Value*) color space, Fig. 3(a–b).
(2) Use the K-Means++ clustering method [28] to obtain four ($K = 4$) clusters on the HSV images, Fig. 3(c), and their corresponding cluster centers.
(3) Select all cluster centers associated with CyanoHABs patches. Compute the average value (m_H, m_S, m_V) of these centers and the standard deviations ($\sigma_H, \sigma_S, \sigma_V$) per channel ($H, S, V$) in the HSV images. Determine lower and upper limits for each channel by subtracting (lower) and adding (upper) two times the corresponding standard deviations for each channel, Table 1.
(4) Binarize each HSV image, so that a logical value of 1 (white) is assigned to pixels with values within the interval defined by lower and upper limits, and logical values of 0 (black) otherwise. Apply the morphological operations of opening, closing, and erosion (with kernel size 5×5) to remove undesired small holes and spots in the binarized areas.
(5) From each binarized image, extract the black binary region with the largest area, Fig. 4(a). This region is cropped by considering its bounding box (mask), Fig. 4(b).
(6) With each mask and its location, extract the real patch from the corresponding original RGB image. A total of 92 real patches are obtained, which are to be inserted into the background images.

It is worth mentioning that although K-Means is ineffective for segmenting bloom clusters in images globally, as explained above, it is sufficiently valid, with the relaxation of boundaries (lower, upper) for patch extraction, which are those that are inserted into the background images.

(*b*.2) *Synthetic patch generation*

The need to generate synthetic patches arises from the limited number of real patches of CyanoHABs available. Inspired in [21,32], we develop a data advanced augmentation technique supported by Generative Adversarial Networks (GANs) and Neural Style Transfer to generate realistic, diverse, and high-quality synthetic patches.

GANs [33] were derived from the game theory in 2014 to generate synthetic images from real images, used as training. This generative model consists of two networks that are trained together: the first network is a generator that creates false images, and the second network is a discriminator to classify between real and false images. Based on this approach, several networks have emerged with impressive performances, including the style-based Generative Adversarial Networks (StyleGANs) that improve image quality, training speed, and network model stability [34].

Despite their performance, the images generated by StyleGANs contain undesired artifacts. In [35] StyleGAN2 is proposed to improve image quality. Its training requires large data sets to avoid overfitting in the discriminator, and this lack of images is an important problem. StyleGAN2-ADA model is proposed in [22], with an adaptive discriminator augmentation mechanism that stabilizes the training with a limited dataset, which justifies its use in this approach. The process is as follows:

(1) Split the real patches into tiles (small patches) of size 256×256 pixels as shown in Fig. 5. Tiles that do not contain CyanoHABs are ignored. The reason for splitting into mosaics is because the number of real patches extracted and available is low (92 in total), as indicated above, and the generative model needs thousands of images for training. The size chosen for these tiles considers that they will be inserted in the background images to find a trade-off between the following two facts: larger patches will produce less data, while smaller patches will not have enough pixels to be later inserted within the high-resolution background images.
(2) Train StyleGAN2-ADA with a total of 3114 small real patches by applying transfer learning from a model previously trained with the Flickr-Faces-HQ (FFHQ) dataset (a high-quality human face dataset) [34]. During the training process, the momentum parameters β_1 and β_2 of the

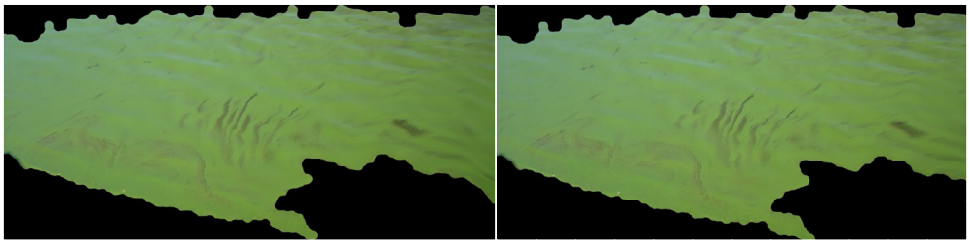


Fig. 5. Splitting real patches into tiles of 256 × 256 pixels, (a) original real patch, and (b) real patch split to tiles.

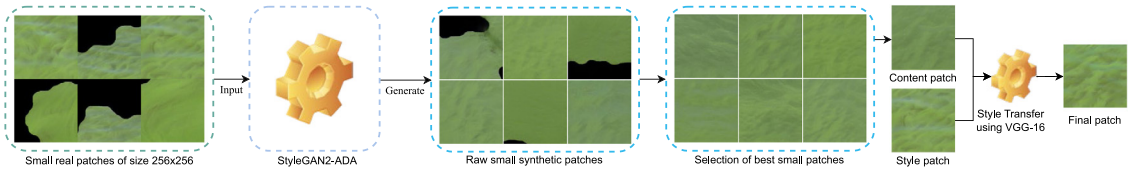


Fig. 6. Small synthetic patch generation process using StyleGAN2-ADA and Neural Style Transfer.

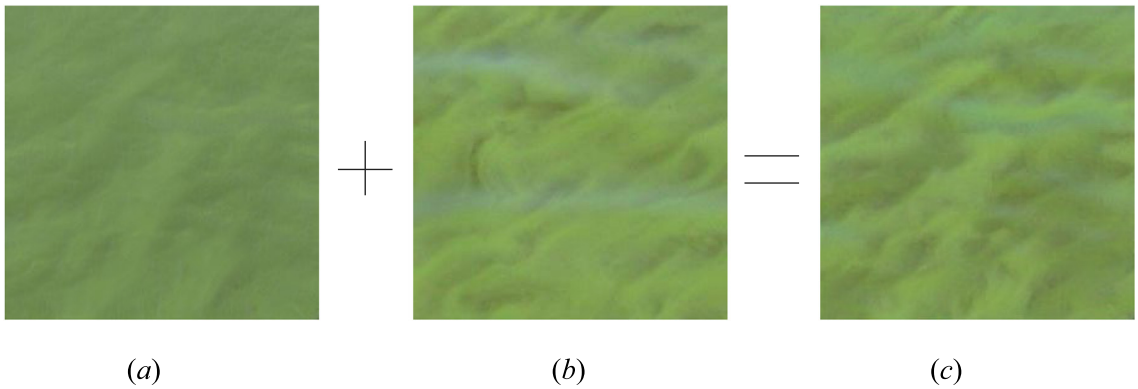


Fig. 7. Small synthetic patch created on the base of content and style patches: (*a*) small patch generated by StyleGAN2-ADA (content patch); (*b*) small real patch (style patch); (*c*) final synthetic patch.

generator and discriminator network are set to 0 and 0.99, respectively, while the initial learning rate is set to 0.0025 and the R_1 regularization weight γ to 0.5, after trial and error. The training phase was completed in approximately 72 h since after that time there were no noticeable improvements, achieving a performance of 42.56 FID (Fréchet Inception Distance).

(3) Once the model is trained, generates 800 small synthetic patches of size 256 × 256 pixels. By visual inspection, chose the best 515 small patches, the most realistic and without black background according to their best FID values. These patches are refined with Neural Style Transfer [23,24]. Neural Style Transfer in GANs uses deep learning-based algorithms, by applying a process consisting of separating and recombining the content of one image with the style of another to produce a new image. In [23] firstly a deep learning-based algorithm is used to create high-quality artistic images, which was late improved by the authors of [24] by varying some hyperparameters in the original model. This improvement has inspired our approach. The full process of small synthetic patches generation is shown in Fig. 6.

(4) Use Neural Style Transfer with VGG-16 [36] to refine the small synthetic patches generated by StyleGAN2-ADA, using the generated patches as content patches, and the tiled real patches as style patches. 75% of the resulting refined patches received texture and color transfer, and the remaining 25% received texture with color preservation (following the strategy presented in [37]). Fig. 7 displays a representative example of refinement with Neural Style Transfer. The final synthetic patch of size 256 × 256 pixels has no black parts, the CyanoHABs cover the full image patch, Fig. 7(c). So, it does not need delimitation and is inserted as it is on the water surface of the background image, specifically 3 synthetic patches are inserted.

(*c*) *Image generation: background and foreground composition*

The composition process to insert the foreground patches (real and synthetic) into the background images is summarized as follows and sketched in Fig. 8 as an illustrative example:

- Apply random augmentation to the foreground image with the transformation operations listed in Table 2. We use these transformations under the assumption that their application generally involves interpolation operations that modify the spectral characteristics to increase the spectral variability of the original patches. Moreover, as these patches are to be fused with the background images, different orientations and sizes are suitable to cover the widest possible range of poses and orientations on different parts of the water surface, on which different spectral shades also appear.

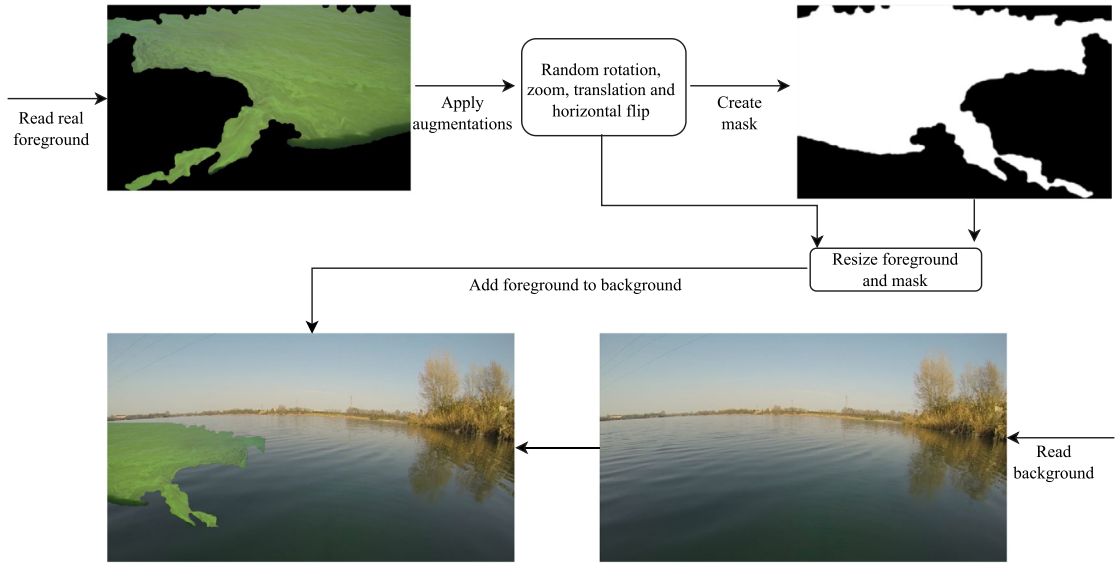


Fig. 8. Synthesis of the image creation process.

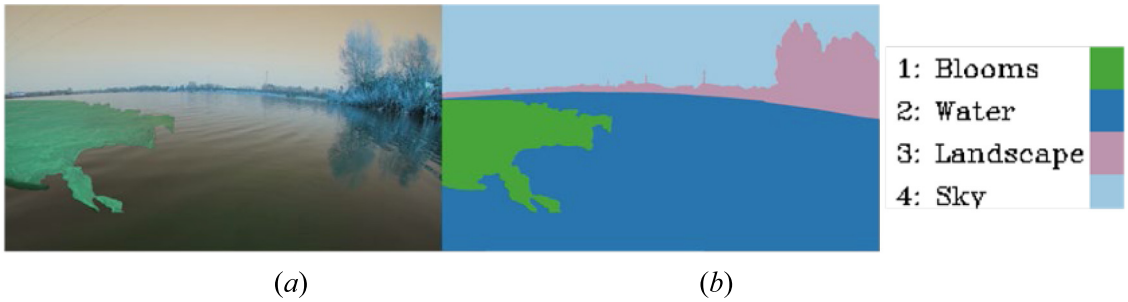


Fig. 9. Annotation example: (a) composite image; (b) annotated regions of interest.

Table 2
Transformations apply to the foreground image.

Transformation	Parameters
Rotation	An angle of 10° to 20°
Zoom	In the range of 0.8 to 1.2
Translation	10 pixels offset along the x and y-axis
Horizontal flip	Horizontal flip around the y-axis

- Extract the segmentation mask from the augmented foreground image.
- Resize the augmented foreground image and the mask without distortion if its heights are greater than 900 pixels, as is the case for some real patches.
- For each subset of background images, a representative image is chosen in order to determine the boundary values of the X and Y coordinates that correspond exclusively to the water class, and then intermediate values are set, e.g., for X every 200 pixels. A random selection of these X and Y coordinates is then made to insert the foreground patch on the water surface in the background images.
- Apply a general weighted average, i.e. a fusion-based approach [38] between the foreground area and the corresponding overlapped area in the background image, so that spectral features of both are mixed (fused). This operation is carried out for each spectral channel separately. The weights assigned to the foreground channels range between [0.80, 0.90], and for each pixel position they are randomly selected.

With the image creation process described above, the following datasets for training and validation of the segmentation models are obtained, all with real images from lakes and reservoirs as background.

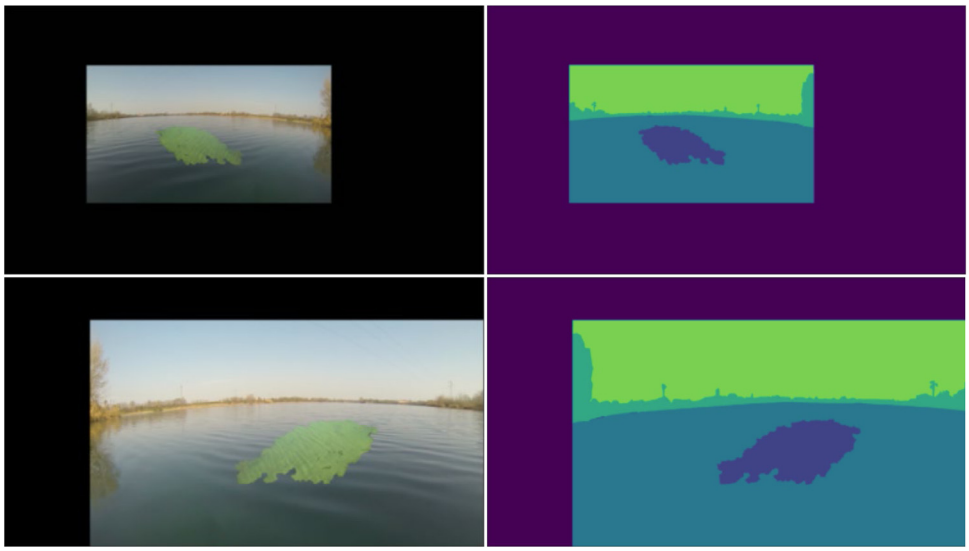
- R-CyanoHABs with 1200 images with real patches.
- S-CyanoHABs contain 1200 images with synthetic patches.
- RS-CyanoHABs, a combination of the above two, with a total of 2400 images. This dataset finally represents the data augmentation based on StyleGAN2-ADA and refined with Neural Style Transfer.

2.2. Data preparation for semantic segmentation

Once the three datasets are available, they are conveniently arranged to be used in the semantic segmentation models. In this section, we describe the layout of the data for training, validation, and testing, which are then made available for the different models described in Section 2.3.

We manually annotate each image, including the ones for testing, using the Labelme tool [39]. It is an open-source tool that allows the drawing of polygons, circles, rectangles, and lines by dragging. Four different regions of interest are annotated, to distinguish between Blooms, Water, Landscape, and Sky at the pixel level, as displayed in Fig. 9. It should be noted that the labels corresponding to the patches are verified taking into account that they have been previously generated and fused and therefore their size and position after fusion is known.

The three datasets are randomly split to contain the distribution summarized in Table 3.



(a) (b)

Fig. 10. Some examples after data augmentation, (a) RGB images, and (b) annotated images.

Table 3
Datasets for training and validation.

Dataset	Training set (80%)	Validation set (20%)
R-CyanoHABs	960 images	240
S-CyanoHABs	960 images	240
RS-CyanoHABs	1920 images	480

For testing, a set of real test images is selected, not used for training nor in validation [40]. With such purpose, 12 have been captured with a mobile device Galaxy A5 from real environments aquatic scenarios and 6 are selected from the all-public-accessible web images. There is not information available on how these images were captured, although they come from conventional, non-remote sensing, capturing devices, and all of them contain regions with cyanobacterial blooms.

A common and useful practice in deep learning applications is the application of image augmentation [40], to increase performance during training [41]. The more diverse the data, the better results [42], avoiding overfitting and facilitating convergence. This is particularly advantageous when data are not abundant enough, as in the case of the R-CyanoHABs and S-CyanoHABs sets that contain only 1200 images each, thus we apply the basic augmentation consisting of the application of geometric and radiometric operations to the input images. While the set RS-CyanoHABs, which is the combination of the other two sets, contains 2400 images already augmented based on StyleGAN2-ADA and Neural Style Transfer (advanced augmentation), no other type of augmentation is applied.

We apply different image transformations from Albumentations [43], a fast and flexible open-source library for classical data augmentation. The transformations applied to the two training datasets (R-CyanoHABS y S-CyanoHABs) in the training process are controlled by an augmentation parameter p defined by each process as displayed in Table 4. This parameter indicates the possibility that the process will actually be applied, expressed in terms of percentage (i.e., $p = 0.3$ expresses a possibility of 30%). Importantly, all augmented images, obtained are normalized as part of the training process because the encoder models are pre-trained with the ImageNet dataset [44]. The first four transformations in Table 4 are geometric and the remaining radiometric operations. All the transformations modify the spectral components through the parameter values. The first group through the applied interpolation process, in the same way as in the generation of the synthetic patches, explained above, and the second ones by direct modifications of the specified modifications. The normalization applies re-centering and re-scaling so that for each pixel, and in each spectral channel, the mean value ($\boldsymbol{\mu}$) of the dataset of images used for training is subtracted and divided by the standard deviation ($\boldsymbol{\sigma}$). These values are the ones displayed at entry 9 in Table 4 and are limited to the range [0,1]. According to [45] this kind of normalization speeds up convergence during training even when data are decorrelated. The data augmentation process described here is called basic, which also includes the fusion of foreground and background images, to distinguish it from what we call advanced data augmentation based on StyleGAN2-ADA.

Some representative examples after data augmentation are displayed in Fig. 10.

An important issue concerning all outdoor aquatic images and specifically the ones containing cyanobacterial blooms, is the reflection effect on the water surface causing intensity saturation. In this regard, the Brightness Contrast operation is responsible of its minimization.

In summary, due to the small number of images available in the datasets, we apply basic data augmentation, with the operations displayed in Table 4, to the training set R-CyanoHABs and S-CyanoHABs. The number of images generated by the augmentation process is controlled by the parameter p, so this number is not known a priori. While advanced data augmentation is based on StyleGAN2-ADA and Neural Style Transfer we apply it to the RS-CyanoHABs dataset.

2.3. Methods for semantic segmentation

Most model architectures based on Deep Learning for semantic segmentation in common use are based on end-to-end trainable encoder–decoder structures. Essentially, the encoder gradually reduces the feature maps and captures higher semantic information, and the decoder gradually recovers the spatial information. The deep of such architectures requires a good selection of images for training to achieve acceptable performances. This justifies

Table 4
Transformations applied to the training sets R-CyanoHABs and S-CyanoHABs.

Nº	Transformation/operations	Parameters
1	Horizontal flip	$p = 0.5$
2	Shift scale rotate	$p = 0.5$, scale limit = 0.5, rotate limit = 0, shift limit = 0.1, border_mode = 0
3	Resize	$p = 1$, height = 256, width = 256.
4	Perspective	$p = 0.5$
5	Gauss noise	$p = 0.2$
6	Random brightness contrast	$p = 0.9$
7	Random gamma	$p = 0.9$
8	Blur	$p = 0.5$ blur limit = 3
9	Normalize	$p = 1$, μ = [0.485, 0.456, 0.406]; σ = [0.229, 0.224, 0.225]

the data generation/augmentation as an essential part of the evaluation process of these models in the domain of CyanoHABs detection. The Encoder–Decoder structures chosen in this study are: UNet++, PSPNet, DeepLabV3+, and FPN. On the other hand, the encoders of these models can be built with convolutional architectures (i.e. (CNN)) used specifically for image classification such as ResNet and EfficientNet variants, used in this work, acting as the backbone. All these models are evaluated with the metrics defined below.

In recent years, these encoder–decoder based models have received much attention because of their performance against conventional models in terms of speed, accuracy, and convergence during training and inference [46]. In this regard, K-Means and Fuzzy c-Means, introduced previously, do not achieve enough efficient performance in the context of CyanoHABs detection, as expressed above. Moreover, the application domains of encoder–decoder structures are very diverse, such as autonomous driving, robotic navigation, industrial inspection, remote sensing, cognitive and computational sciences, medical sciences, agriculture, and many others, so its application in the field of CyanoHABs looks promising.

(*a*) *Encoder–Decoder based models:* UNet++, PSPNet, DeepLabV3+ and FPN

In [47] a comparison between UNet [48] and UNet++ [49] architectures is highlighted, concluding that UNet++ outperforms UNet. Also, in [49] is reported that UNet++ outperforms UNet with an average IoU of 3.9. The Encoder part evolves towards a compressed representation of the information, from the input image, so that through different convolutional layers, combined with ReLU and max-pooling operations, feature maps with decreasing resolutions are obtained. On the other hand, the Decoder part gradually up-samples the previous layers, achieving high-resolution feature maps, where the information embedded in these maps is combined with the spatial–spectral and contextual information coming from the Encoder, at the same level of resolution, via skip connections. In UNet++ the skip connections consist of a dense convolution block with concatenated convolution layers where the information flows in the feature maps until it is fused with the corresponding up-sampled block of the same resolution in the decoder. Such dense connections have the advantage of propagating the semantic information of the feature maps from the encoder to the decoder at the levels the decoder is awaiting, with high efficiency in recovering fine-grained details, as the ones existing in the CyanoHABs patches.

PSPNet (*pyramid scene parsing network*) [50] took first place in the competition for ImageNet Scene Parsing Challenge 2016 [51]. It is a model designed as a pyramid parsing approach that exploits global contextual information by different region-based context aggregation. This model contains a pyramid pooling module that separates the information into feature maps of different scales to capture different contextual levels in the original image. These levels are up-sampled according to their resolution and merged with the first feature map obtained after a first convolution to form the high-level feature map, on which another convolution is applied to generate the final per-pixel classification.

DeepLabV3+ [52] outperforms PSPNet and its predecessor DeepLabV3 [53], being an extension of this last one. The Encoder module in DeepLabV3+ applies atrous convolution to extract and package the information at different levels of resolution (scales) into the corresponding feature maps. The Decoder refines the segmentation results along object boundaries, and consequently on the boundaries separating the different categories of pixels in the images, including those defined by the cyanoHABs patches.

The Feature Pyramid Network (FPN) [54] is an extension for lighter segmentation (in terms of compute and memory) and produces higher resolution features [55]. FPN consists of a top-down (encoder–decoder) architecture, with convolutional feature maps obtained at different pyramidal levels of resolution (hierarchy), on both encoder and decoder and skip connections joining levels of the same scale. The model's predictions are made at all levels of the pyramid by exploiting the semantic information existing at those levels.

(*b*) *Encoder architectures:* ResNet and EfficientNet

CNNs have become popular in different computer vision tasks, such as image classification, object detection, image generation, semantic segmentation, and so on. To increase the efficiency and accuracy of these networks, scaling methods were developed, by arbitrarily increasing the depth or width of the CNN or using a larger input image resolution for training and evaluation [56]. ResNet (ResNet-18 to ResNet-150) [57] applies this type of scaling, increasing the number of layers and making the model become very deep with relevant results (it won first place in the ILSVRC 2015 classification task [44]).

On the other hand, there are methods based on uniform scaling of the network dimensions, such as depth, width, and image resolution. The EfficientNet family of architectures (EfficientNet-b0 to EfficientNet-b7) performs this type of scaling achieving state-of-the-art performances in 2020 in both, ImageNet and other tasks [56]. Moreover, this model is considered one of the most efficient models, as it requires fewer FLOPS (FLoating point Operations Per Second) for inference than other existing models [56].

Therefore, the weights involved in the encoders (backbone) of ResNet50 and EfficientNet-b6 in UNet++, FPN, PSPNet, and DeepLabV3+ segmentation models are initialized with the results obtained after a pre-training process with the ImageNet dataset. More specifically, ResNet50 is used by applying arbitrary scaling and EfficientNet-b6 with uniform scaling. These encoders with pre-trained weights allow better results in terms of accuracy and faster convergence. Results coming from combinations of encoders, based on ResNet50 and EfficientNet-b6, with the above-mentioned segmentation models, are displayed in Tables 7 to 9.

Pre-training is carried out with the well-known ImageNet dataset [44], taking advantage of the knowledge embedded in this dataset. Thus, the initialization exploits the information of this

Table 5
Summary of methods with references, description, and use.

	Method	Description	Use
1	Image augmentation [40]	New images from the existing	R-CyanoHABs generation
2	StyleGAN2-ADA [22]	Generative model-discriminator	S-CyanoHABs generation
3	Neural style transfer [23,24]	Combine contrast with style	S-CyanoHABs generation
4	Fréchet Inception Distance (FID), [25]	Similarity measurement	S-CyanoHABs generation
4	K-Means [28], Fuzzy c-Means [29]	Clustering methods	Justification for semantic methods
5	Albumentations [43]	Library for data augmentation	R-CyanoHABs and S-CyanoHABs generation
6	Imagenet [44]	Dataset	Pre-train CNN models
7	UNet++ [49], PSPNet [50], DeepLabV3+ [52], FPN [54]	Encoder-Decoder models	Semantic segmentation
8	ResNet [57], EfficientNet [56]	CNN-based models	Encoder backbone architecture
9	Overall Accuracy (OA) [46]	Evaluation metric	Assessment of semantic models
10	Intersection over Union (IoU) [46,59]	Evaluation metric	Assessment of semantic models

dataset in the first part of the network where relevant features are to be extracted, at least in the first layers, such as boundaries. Because the domain represented by ImageNet and the aquatic where CyanoHABs are to be detected are very different, a re-training is required to achieve a fine-tuning of the weights valid for CyanoHABs identification, improving the convergence, as expressed above. On the other hand, the initialization of the decoder part is based on the initializer proposed in [58], which uses samples weights from the normal distribution with zero mean and variance 2/N where N is the number of nodes in the previous layer. Unlike the encoder, this random initialization is justified on the basis that the decoder captures more global information from the images in the dataset, and therefore it does not need initialization with the underlying information.

(*c*) *Evaluation metrics*

Quantitative evaluation of semantic segmentation models can be performed using pixel-based and overlap-based measures [46]. From the confusion matrix, the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) pixels are obtained to calculate metrics such as Overall Accuracy (OA), Precision, Recall (also known as sensitivity) and F1 Score. Alternatively, commonly used overlap-based metrics are the Dice coefficient and Intersection over Union (IoU) [46,59], the latter also known as Jaccard Index. Both metrics can also be calculated from the confusion matrix, as shown in Eq. (2).

In this study, we choose overall accuracy and IoU to measure the performance of semantic segmentation models. These metrics are defined as follows:

$$Overall\ Accuracy\ (OA) = (TP + TN)/(TP + TN + FP + FN) \quad (1)$$

$$IoU = TP/(TP + FP + FN) \quad (2)$$

Regarding this choice, it is worth mentioning that *OA* measures the percentage of pixels correctly classified averaged across all classes (four in this approach). It evaluates both, pixels correctly classified as belonging to the correct class (*TP*) and pixels correctly identified as not belonging to a selected class (*TN*) against all pixels. Although *OA* is a global metric, when the representation of a given class is small or relatively small compared to the overall size of the image and the rest of the classes, it is possible that this measure may lead to certain biases as it is including in the measure how well negative cases (*TN*) are identified (along with the positive ones). This is sometimes the case with the identification of the class to which cyanobacteria belong. For this reason, it was thought appropriate to try to consider this bias by using the *IoU* measurement coefficient, which excludes *TN*. Under this consideration, these two metrics cover the expectations for the evaluation of the classes present in the images, while the other metrics, mentioned above, do not contribute anything relevant in this respect.

3. Experimental results and discussion

For clarity and better tracking the process and experimental results, Table 5 summarizes the different methods applied, together with a brief description and their use in the different parts of the proposed approach.

In the following, we present the experimental results and discussion. Firstly, the training and evaluation performance of model architectures for semantic segmentation with pre-trained encoders is introduced. Secondly, the best-performing model is evaluated from each dataset on real test images. Finally, the results are discussed.

3.1. Training and evaluation

The training process is critical in semantic segmentation involving deep learning. It requires a special analysis to establish specific conditions and settings, as a preliminary step to select the best model (i.e. UNet++, PSPNet, DeepLabV3+, FPN). After training, the next step is the performance analysis for each model, especially considering the use of CNN-based encoders (ResNet, EfficientNet) with their ability to take advantage of transfer learning. First, we provide the details of the training prior (to find the appropriate loss function and optimizer) and of the training itself. Then, we show the performance results of the selected models and architectures for semantic segmentation with pre-trained encoders on the three validation sets.

(*a*) *Previous considerations and parameter settings*

A major issue arising from the three datasets (R-CyanoHABs, S-CyanoHABs, and RS-CyanoHABs) is the well-known class imbalance, deriving from the inherent nature of the problem. This happens because the CyanoHABs patches are at a numerical disadvantage concerning the other textures present in the images. Fig. 11 displays the lower number of pixels in that class concerning the rest, computed over the images in the three training sets. Training with such images often leads to the trained network being biased towards the larger regions and trapped in local minima [60]. Therefore, to address this problem, and to minimize its effect for better performance, some preliminary experiments are required. We train FPN with Efficient-b6 considering the 80% of the R-CyanoHABs dataset (and the remaining 20% for validation) varying the loss function (to mitigate class imbalance), the optimizer, and the number of epochs while keeping the batch size ($bs = 4$) and the learning rate ($\alpha = 0.001$). The choice of this setting is because in other studies, such as in [61], it achieves an intermediate performance compared to other models and values. Table 6 provides details about the different settings with six experiments and Fig. 12 displays the optimization learning curves calculated on the loss.

Experiments 1 to 3 show that the loss value is high for both the training and validation curves displayed in Fig. 12, which

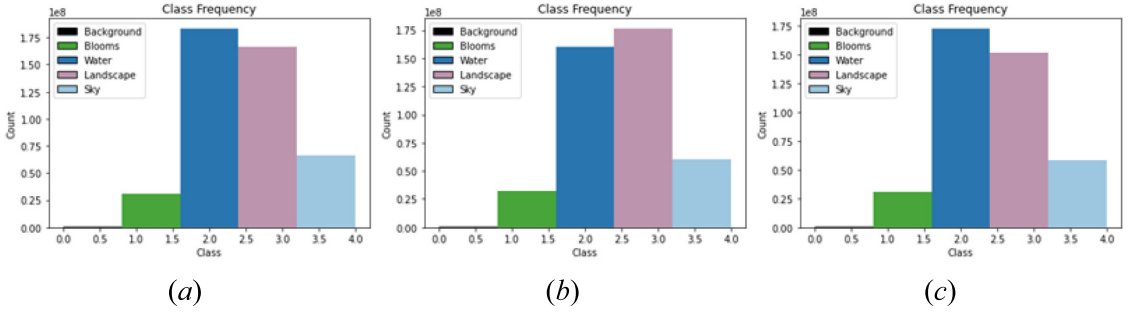


Fig. 11. Distribution of pixels per class in the three training sets, (a) R-CyanoHABs, (b) S-CyanoHABs, and (c) RS-CyanoHABs.

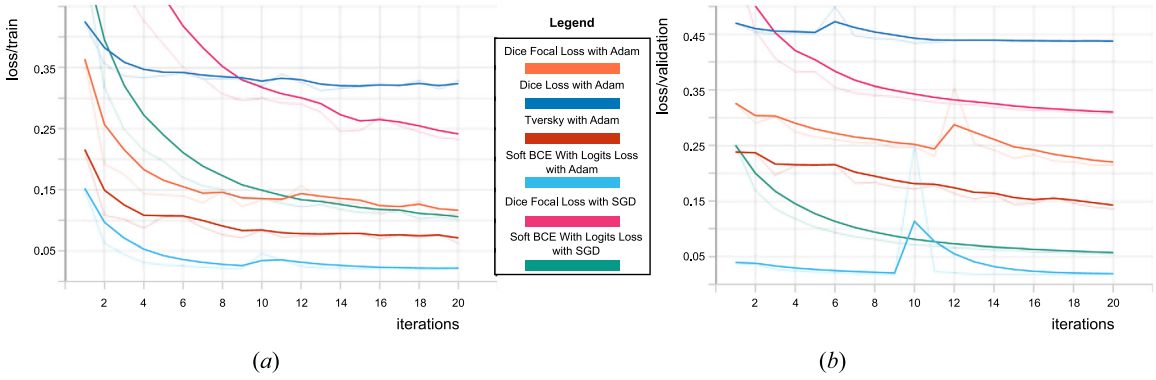


Fig. 12. Optimization learning curves from the loss function, (*a*) training, and (*b*) validation.

Table 6
Results of previous training to find a suitable configuration ($\alpha = 0.001$, bs $= 4$).

Nº	Loss function	Optimizer	# Epochs	Training loss (%)	Validation loss (%)
1	Dice focal	Adam	20	11.28	21.50
2	Dice	Adam	20	32.82	43.70
3	Tversky	Adam	20	6.37	13.61
4	Soft BCE with logits	Adam	20	2.27	1.85
5	Dice focal	SGD (momentum $= 0.9$)	50	23.28	30.79
6	Soft BCE with logits	SGD (momentum $= 0.9$)	50	10.10	5.54

Table 7
Performance comparison of model architectures for semantic segmentation (UNet++, FPN, PSPNet, and DeepLabV3+) with encoders (ResNet50 and EfficientNet-b6) without data augmentation on the two validation sets (R-CyanoHABs and S-CyanoHABs) using the OA and IoU metrics.

Encoder	Architecture	R-CyanoHABs		S-CyanoHABs	
		OA (%)	IoU (%)	OA (%)	IoU (%)
ResNet50	UNet++	97.33	93.80	98.17	95.21
	FPN	96.98	92.29	97.87	94.30
	PSPNet	95.82	89.61	97.34	93.18
	DeepLabV3+	97.27	92.81	98.01	94.48
EfficientNet-b6	UNet++	**97.66**	**94.71**	**98.47**	**95.75**
	FPN	97.60	94.28	98.27	95.26
	PSPNet	97.12	92.79	97.81	93.71
	DeepLabV3+	97.56	94.17	98.20	94.90

indicates that the model is biased and does not capture relevant information, leading to underfitting. Furthermore, they all converge to roughly stationary value, achieving a certain stabilization once a certain number of iterations has been reached. Ideally, they should all follow the downward tendency until they are close the ideal value of zero, but this does not happen, which is why these configurations are discarded. Note that as the different loss functions measure different aspects of the training and validation phase of each model, they are not directly comparable. Nevertheless, their tendencies allow to determine the performance of each model and reject those that do not have overall good behavior.

In experiment 4, the loss value is minimal for both training and validation curves, Fig. 12, and this indicates that the learning algorithm captures all the richness from the data and models properly in both, the training data, and the new data well.

Softmax Binary Cross-Entropy (BCE) With Logits Loss, is the replacement of BCE With Logits Loss with some extra additions, numerically more stable, is also the variation of the original BCE loss function [62] and it works correctly for unbalanced datasets.

In experiments 5 and 6, which use some of the same loss functions, optimized with SGD instead of Adam, the training and validation loss functions decay slowly, Fig. 12, requiring a larger number of epochs to reach a convergence point. Thus, the

Table 8
Performance comparison of model architectures for semantic segmentation (UNet++, FPN, PSPNet, and DeepLabV3+) with encoders (ResNet50 and EfficientNet-b6) with data augmentation on the three validation sets (R-CyanoHABs, S-CyanoHABs, and RS-CyanoHABs) using the OA and IoU metrics.

Encoder	Architecture	R-CyanoHABs		S-CyanoHABs		RS-CyanoHABs	
		OA (%)	IoU (%)	OA (%)	IoU (%)	OA (%)	IoU (%)
ResNet50	UNet++	98.02	94.76	98.33	96.45	98.90	97.44
	FPN	97.28	92.84	98.08	95.57	98.36	96.26
	PSPNet	97.17	92.41	97.65	94.35	98.47	95.95
	DeepLabV3+	97.70	93.87	98.21	96.00	98.71	96.71
EfficientNet-b6	UNet++	**98.47**	**95.95**	**98.62**	**97.02**	**99.09**	**97.97**
	FPN	98.13	94.89	98.47	96.56	98.80	96.97
	PSPNet	97.81	93.87	98.06	95.30	98.62	96.32
	DeepLabV3+	98.13	94.99	98.39	96.38	98.89	97.22

Table 9
Performance comparison of model architectures for semantic segmentation (UNet++, FPN, PSPNet, and DeepLabV3+) with encoders (ResNet50 and EfficientNet-b6) without (WO) and with (W) data augmentation on the three validation sets (R-CyanoHABs, S-CyanoHABs, and RS-CyanoHABs) using the IoU averaged metric per class.

Encoder	Architecture	Blooms (%)		Water (%)		Landscape (%)		Sky (%)	
		WO[a]	W[b]	WO[a]	W[b]	WO[a]	W[b]	WO[a]	W[b]
Results on the validation set R-CyanoHABs									
ResNet50	UNet++	91.30	92.11	96.32	97.39	94.38	95.15	93.20	94.39
	FPN	87.56	88.10	95.90	96.36	92.92	93.65	92.79	93.23
	PSPNet	81.37	87.36	94.29	96.40	91.98	93.51	90.81	92.38
	DeepLabV3+	89.48	90.50	96.18	97.05	93.31	94.53	92.27	93.38
EfficientNet-b6	UNet++	**93.27**	**94.72**	96.76	**98.03**	95.00	**96.06**	93.81	**95.00**
	FPN	91.61	92.07	96.62	97.62	94.85	95.51	94.02	94.35
	PSPNet	88.23	89.95	96.10	97.36	94.06	94.92	92.76	93.23
	DeepLabV3+	91.59	92.75	96.58	97.65	94.73	95.39	93.76	94.15
Results on the validation set S-CyanoHABs									
ResNet50	UNet++	94.45	96.61	97.11	98.25	95.27	96.03	93.99	94.91
	FPN	91.37	94.07	96.78	97.94	94.87	95.78	94.16	94.50
	PSPNet	89.05	91.39	96.49	97.44	94.20	95.10	92.96	93.46
	DeepLabV3+	91.84	95.22	96.80	98.09	94.90	95.93	94.37	94.74
EfficientNet-b6	UNet++	**95.53**	**97.02**	**97.46**	**98.61**	95.05	**96.74**	94.94	**95.70**
	FPN	93.54	95.73	97.16	98.41	95.43	96.60	94.90	95.48
	PSPNet	90.08	92.80	96.64	97.96	94.55	95.98	93.57	94.47
	DeepLabV3+	92.27	95.64	97.01	98.32	95.42	96.37	94.91	95.20
Results on the validation set RS-CyanoHABs									
ResNet50	UNet++	96.63		98.88		97.42		96.81	
	FPN	94.52		98.21		96.60		95.72	
	PSPNet	93.09		98.44		96.70		95.57	
	DeepLabV3+	94.77		98.67		97.12		96.26	
EfficientNet-b6	UNet++	**97.57**		**99.11**		**97.89**		**97.32**	
	FPN	95.13		98.79		97.39		96.55	
	PSPNet	93.66		98.61		97.05		95.95	
	DeepLabV3+	95.77		98.88		97.49		96.74	

[a]Without data augmentation.
[b]With data augmentation.

best results are obtained in experiment 4, i.e., it determines the hyperparameters and settings used for the experiments reported below.

(*b*) *Performance evaluation*

In this section, we report the quantitative results of the performance of model architectures for semantic segmentation on the three validation sets, and of course, identify the best-performing model on each set.

The model architectures used in the experiments are summarized in Tables 7 and 8 and implemented with the Segmentation Models Pytorch (SMP) library presented in [63], which is a high-level API built over PyTorch, an open-source framework for machine learning projects. Besides, the training, validation, and testing of all these model architectures for semantic segmentation, including those carried out during the preliminary experiment, are performed independently, using the Google Colab service that provides an NVIDIA Tesla P100 GPU and 16 GB of memory. Table 7 shows the results without data augmentation on the two validation sets R-CyanoHABs and S-CyanoHABs, and Table 8 shows the results with data augmentation on the three validation sets R-CyanoHABs, S-CyanoHABs, and RS-CyanoHABs, indicating: (1) the backbone CNN-based used in the encoder, (2) the full architecture, and (3) results on the validation sets. The results are based on the overall pixel accuracy (OA, percentage of correctly labeled pixels in the validation set) and the mean *IoU* score (i.e., the mean of the class-wise intersection-over-union score). The *IoU* is selected because it excludes the *TN* to avoid the negative effect of relatively small CyanoHABs patches, as described above.

In Table 7, all model architectures for semantic segmentation obtain acceptable but not sufficient performance, e.g., in the R-CyanoHABs set the difference between the model with a high mean IoU score and the worst one is 5.10%, and in the S-CyanoHABs set 2.57%. The results of the models in the S-CyanoHABs set are slightly better than in the R-CyanoHABs set, e.g., the difference between the best model in each set is 1.04%. UNet++ with EfficientNet-b6 achieves a high mean IoU score in the two validation sets.

In the following, we highlight the remarkable points of the results presented in Table 8.

- The performance of all model architectures for semantic segmentation on the R-CyanoHABs and S-CyanoHABs validation sets is slightly better than that presented in Table 7, and on the RS-CyanoHABs validation set the improvement is noticeable.
- In the three validation sets, all the model architectures for semantic segmentation combined with different encoders perform well, the difference between the best and the worst mean IoU score is 5.56%. This indicates that all models generalize very well on the validation data, thanks to the data augmentation (basic augmentation in the R-CyanoHABs and S-CyanoHABs sets, and advanced augmentation based on StyleGAN2-ADA and Neural Style Transfer in the RS-CyanoHABs set) and the strength of the pre-trained weights.
- UNet++ in combination with EfficientNet-b6 and ResNet50 outperforms other architectures, followed by DeepLabV3+, FPN, and PSPNet. However, the EfficienteNet-b6 encoder allows UNet++ to score high on the mean IoU in all three validation sets. Furthermore, the margin between the two encoders is relatively small (e.g., in the RS-CyanoHABs set, it is 0.53%). It is clear that UNet++ with its dense skip connections, along with the encoder depth (EfficientNet-b6) outperforms in terms of accuracy (OA, IoU) the other models and architectures, i.e., this extracts conveniently the underlying information in the images.
- All the architectures combined with EfficientNet-b6 perform better than with the ResNet50 encoder. ResNet50 contains 48 convolutional layers, one max pool, and one average pool and EfficientNet-b6 has 668 layers, including 139 convolutional layers. The difference in the number of layers is significant. It is well known that deeper CNNs can capture richer and more complex features and generalize better (an advance over the limited number of images available for training). This greatly enhances the capture of spatial–spectral and contextual information embedded in both real (R-CyanoHABs) and synthetic (S-CyanoHABs) images and especially when both are used (RS-CyanoHABs). In terms of accuracy, this demonstrates the great ability of EfficientNet-b6 to extract features from any data set. This confirms the use of EfficientNet-b6 in different applications to improve accuracy and efficiency as it can scale all dimensions of CNN width and depth. Deeper can capture richer and more complex features with good generalization and wider can capture well fine-grained features. All these properties are present on images containing CyanoHABs patches.
- UNet++ with EfficientNet-b6 has the higher score in mean IoU on the RS-CyanoHABs set concerning other sets, followed by S-CyanoHABs and R-CyanoHABs. The difference with the second is 0.95%, and with the third 2.02%. Moreover, the other architectures (FPN, PSPNet, and DeepLabV3+), regardless of the encoder, obtain better results on the RS-CyanoHABs set, followed by S-CyanoHABs and R-CyanoHABs. Thus, it is proved that model architectures for semantic segmentation trained with advanced data augmentation based on StyleGAN2-ADA and Neural Style Transfer outperform basic augmentation, as they bring more variability to the dataset to further improve the training process. In addition to the better performance of EfficientNet, it is clear that the structure of UNet++ with its dense skip connections, at all levels of resolution, contributes to the better performance of this architecture as a whole, and therefore, the capture of spectral and contextual information in relation to the CyanoHABs patches is guaranteed. Although there are not enough real data available, which has motivated the approach of generating synthetic data and combining them with real data, the results obtained allow the prediction that the proposed scheme will provide satisfactory results when real data are used exclusively.

The results without (WO) and with (W) data augmentation regarding the comparison of averaged IoU scores per class in the three validation sets are shown in Table 9, indicating in the first two columns the encoder and model architecture for semantic segmentation, and in the last four columns the classes of interest.

In Table 9, the UNet++ model with EfficientNet-b6 trained with data augmentation (basic or advanced) obtains the best results for the four classes (Blooms, Water, Landscape, and Sky) in all three validation sets. Moreover, all the classes obtain their highest score in the RS-CyanoHABs set (advanced augmentation), followed by S-CyanoHABs (basic augmentation) and R-CyanoHABs (basic augmentation). For example, for the Blooms class, the difference with the second one is 0.55% and with the third one 2.85%.

To further study the effect of water reflections, which cause intensity saturation in the images, several experiments have been carried out by varying the parameter p in the basic augmentation operation corresponding to Brightness Contrast, Table 4. In some, we have left the possibility of applying this operation to the maximum ($p = 1$), in others without the possibility of applying it ($p = 0$), and in the rest with an intermediate possibility ($p = 0.5$). With respect to the results of the previous table, no noteworthy improvements have been observed, since the best case has meant an improvement of 0.05% in the Water class with $p = 1$, always with the architecture UNet++ and EffcienteNet-b6. In other classes, such as Blooms, the results have worsened. With $p = 0$ and 0.5 the results have also worsened. This leads to the conclusion that the value of $p = 0.9$ (Table 4) is appropriate for such models.

The highest scoring class of the UNet++ model with EfficientNet-b6 in the RS-CyanoHABs validation set is for the Water class, followed by Landscape, Blooms, and Sky. While in the S-CyanoHABs set the Blooms class ranks second (only surpassed by Water), followed by Landscape and Sky, and in the R-CyanoHABs set, the highest scoring class is Water, followed by Landscape, Sky, and Blooms. Again, as previously indicated, the use of very deep models in conjunction with skip dense connections achieves good results. This allows us to hypothesize that it is very likely that the behavior of these architectures with any real data will be equally efficient. The following section is a confirmatory test of this hypothesis, verified with the real data available at this moment.

The benefits of using synthetic imagery to validate models are further discussed in Section 3.3.

3.2. Testing

In this section, we report the quantitative and qualitative results of the best-performing model from each dataset on real test images, always with the encoder EfficientNet-b6 as backbone and the UNet++ architecture, as the best-performing model from each dataset, as indicated above.

Table 10 shows the quantitative results without and with data augmentation (basic and advanced depending on the augmented dataset) for each dataset, indicating in the first four columns the classes of interest, and in the last two columns, the mean IoU and the averaged OA.

Table 10 shows the poor generalization performance on real test images of the model trained without data augmentation and improved generalization performance on the same images

Table 10
Quantitative results without and with data augmentation of the best-performing model from each dataset on real test images.

	Blooms (%)	Water (%)	Landscape (%)	Sky (%)	Mean IoU (%)	OA (%)
Without data augmentation	**R-CyanoHABs**					
	40.54	52.69	68.47	98.68	65.10	71.55
	S-CyanoHABs					
	39.40	53.02	66.34	97.21	63.99	69.07
With data augmentation	**R-CyanoHABs**					
	82.94	79.69	86.83	99.84	87.33	90.19
	S-CyanoHABs					
	84.12	78.32	85.72	98.28	86.61	89.53
	RS-CyanoHABs					
	90.81	79.91	88.07	98.65	89.36	92.76

of the models trained with data augmentation. In particular, the best-performing model in R-CyanoHABs without data augmentation barely reaches 40.54% in predicting the Blooms class, while the best-performing model in S-CyanoHABs without data augmentation only reaches 39.40% for the Blooms class.

Regarding the results with data augmentation, the most interesting results for analysis are those concerning the Blooms, as it is the region (foreground image) of the fused image that received the augmentation based on StyleGAN2-ADA and Neural Style Transfer in the S-CyanoHABs and RS-CyanoHABs sets. Regarding the other regions corresponding to the background image, (i.e. water, landscape, and sky) only minor modifications were made concerning data augmentation (e.g., horizontal flipping to the background image at the time of construction of the S-CyanoHABs dataset). In this regard, we can infer that the best-performing model in the RS-CyanoHABs set for the Blooms class obtains a high score against other models, for example, the difference with the best-performing model in the R-CyanoHABs set is 7.87%, and with the best-performing model in the S-CyanoHABs set is 6.69%. These results indicate that data augmentation based on StyleGAN2-ADA and Neural Style Transfer is a promising method that improves model performance significantly over basic data augmentation.

On the other hand, in relation to the effect of reflection, the same experiments as in the case of the validation, mentioned in the previous section, have been repeated for the case of the basic augmentation Brightness Contrast operation. The conclusion is the same, i.e. no substantial improvement has been found when the saturation is corrected. Only in the Water class, a slight improvement appears with the UNet++ and EfficientNet-b6 architecture.

Continuing with the results with data augmentation, the difference between the best-performing model on the R-CyanoHABs set and the best-performing model on the S-CyanoHABs set is not much, only 0.72% (mean *IoU*), indicating that the best model trained on images containing synthetic patches obtains results almost comparable to those of the best model trained on images containing real patches. It is easy to infer, from this reasoning, that if the number of real images is (considerably) increased, when sufficient images are available, the best performance is guaranteed with the models with dense connections (UNet++) and high depth (EfficentNet-b6).

For illustrative purposes, Fig. 13 shows qualitative results of the segmentation performed on four representative images of the real test dataset with the best-performing model obtained for each dataset (Table 8). Note that we have not applied any preprocessing to these test images.

The original images in rows 1 to 3 are images from the internet and are licensed under Creative Commons. The one in row 1 [64] is licensed under CCO Public Domain, and those in rows 2 [65] and 3 [66] under CC BY-SA 3.0. The input image in row four is an image captured in its natural form with a mobile device, Samsung Galaxy A5. The regions that compose them are similar (at least to the human eye) to those of images used for training and validation.

The model (UNet++ with EfficientNet-b6) with the best performance in R-CyanoHABs predicts cyanobacterial blooms quite well in terms of density and area coverage, especially in images one, two, and four. Similarly, the model (UNet++ with EfficientNet-b6) with the best performance on S-CyanoHABs predicts almost in a similar way to the previous model and even manages to outperform in image number 3. On the other hand, the model (UNet++ with EfficientNet-b6) with better performance on RS-CyanoHAB predicts overall better than the previous two, indicating that the data augmentation based on StyleGAN2-ADA and Neural Style Transfer has been beneficial. This also allows indirectly to infer that the generation of real and synthetic patch data is in turn beneficial with this approach, as RS-CyanoHABs contain both types of images. The use of more real data clearly guarantees the best performance.

3.3. Discussion

We have been hypothesizing about how to detect CyanoHABs from a machine vision system onboard an ASV using state-of-the-art deep learning-based semantic segmentation methods. In the absence of public or own available data, we have generated synthetic data which allows the validation of the proposed models for use in real-life scenarios, to have the models trained and ready for the moment when the cyanobacterial blooms occur. This is the core of this research, and the results show that the creation of datasets with this kind of images is an effective alternative when lacking of enough real-world images. More specifically, the following facts can be derived.

(1) Results in Tables 8 and 9 show the feasibility of the proposed strategy (data augmentation based on StyleGAN2-ADA and Neural Style Transfer), especially in the best-performing model (UNet++ with EfficientNet-b6) trained on the RS-CyanoHABs set (images with real and synthetic patches). Moreover, Table 10 (with data augmentation) and Fig. 13(d) show how well it can detect cyanobacterial blooms on images coming from two datasets with different data distributions (change of dataset [67]). This is important because generally real-world data are moderately different, i.e., from the ASV perspective, the colorations of water bodies vary depending on what the sky is like (clear, cloudy, etc.), the time of day, how polluted it is, the season, etc. Therefore, with these approaches, lakes, and reservoirs are characterized as non-stationary environments.

(2) Tables 7 and 8 display that the best-performing model (UNet++ with EfficientNet-b6) trained on S-CyanoHABs set (images with synthetic patches generated by StyleGAN2-ADA and refined with Neural Style Transfer) get a better result than the best model trained on R-CyanoHABs (images with real patches). Again, Table 10 and Fig. 13(c) show that

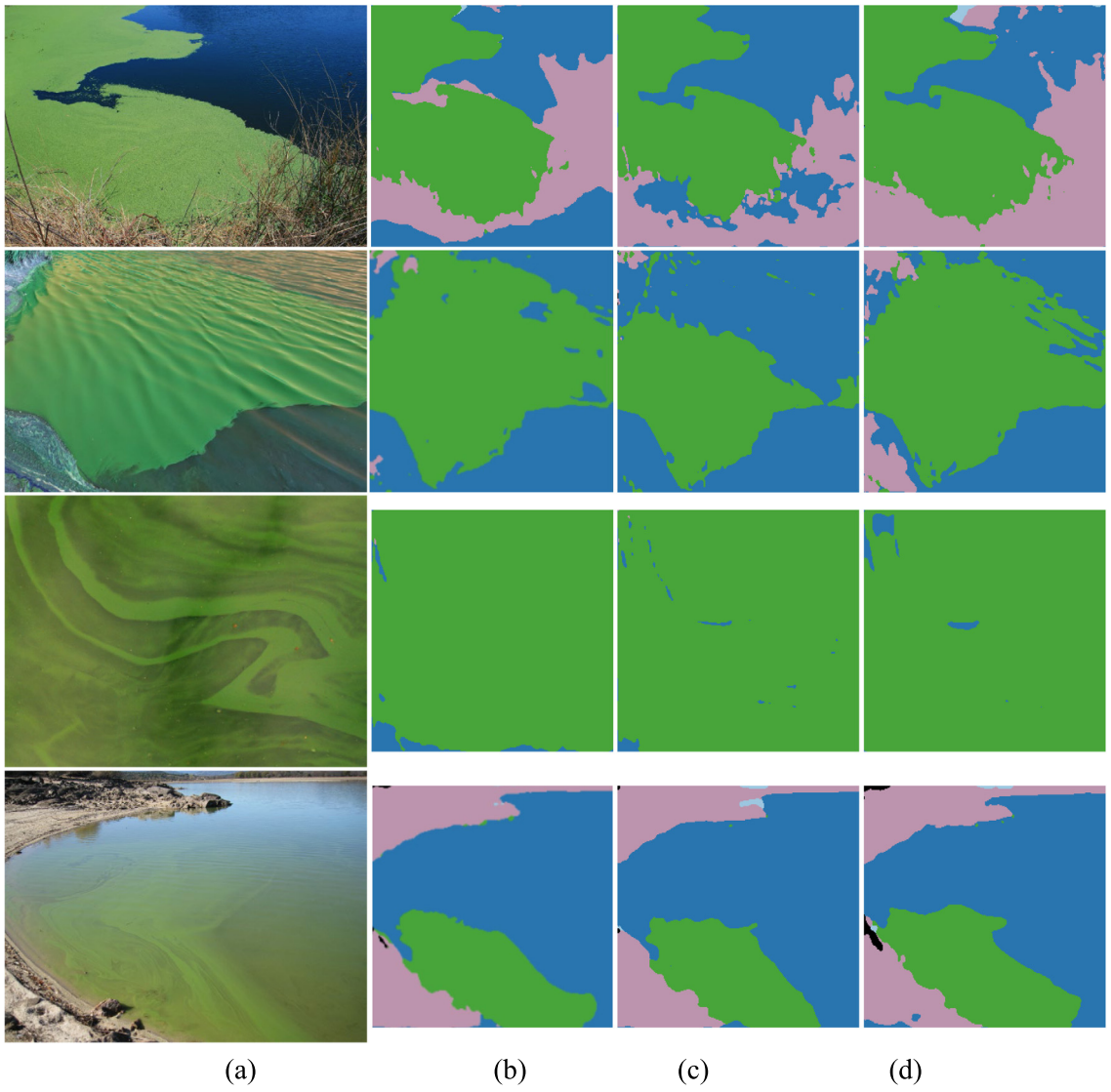


Fig. 13. Qualitative results with data augmentation of the best-performing model from each dataset on real test images, (a) original images, (b) UNet++ with EfficientNet-b6 (best-performing model on R-CyanoHABs), (c) UNet++ with EfficientNet-b6 (best-performing model on S-CyanoHABs), (d) UNet++ with EfficientNet-b6 (best-performing model on RS-CyanoHABs).

its generalization ability on real test images was almost similar to that of this model. From this result we can infer that the StyleGAN2-ADA model in conjunction with Neural Style Transfer has succeeded in generating cyanobacterial patches that are sufficiently realistic in terms of texture, color, variety, and quality.

(3) Table 10 and Fig. 13(d) show that data augmentation, based on StyleGAN2-ADA and Neural Style Transfer, provided the model with a remarkable improvement in generalization ability on real test images coming from an independent set. This indicates that the augmentation based on advanced techniques provides the model with a wider variety of data than basic data augmentation (horizontal flip, rotation, etc.).

(4) Tables 8 and 9 show that the models trained on dataset S-CyanoHABs (images with synthetic patches) have managed to obtain a high score against other models trained on R-CyanoHABs (images with real blooms). This makes us understand that models learn better from virtually generated data by a computer. It is worth keeping in mind, that a model trained on a simulated (not very realistic) dataset that scores high on a validation set does not necessarily will have good generalization on moderately different real-world data. Evidence of this assertion can be seen in Table 7 where UNet++ with EfficienteNet-b6 trained on the S-CyanoHABs set without data augmentation obtains on the validation set an acceptable value of 95.75% mean IoU, but the same configuration does not generalize well on real test images obtaining a value of 63.99% mean *IoU*, Table 10.

(5) Finally, it is worth mentioning, that models with dense connections (UNet++) and high depth encoder (EfficentNet-b6) are jointly able to properly handle and represent the underlying spectral and contextual information in images containing CyanoHABs, which provides sufficient clues to address this issue with deep models and dense interconnections.

4. Conclusions

Our findings show the viability of the approach of detecting CyanoHABs from the ASV perspective using deep learning-based semantic segmentation methods with synthetic images.

In the absence of sufficient real data, we have generated three synthetic image datasets using the image compositing technique.

This allows us to have validated models already available for when blooms appear on the water surface.

We trained and evaluated four semantic segmentation model architectures (UNet++, FPN, PSPNet, and DeepLabV3+) with two encoders as backbone (ResNet50 and EfficientNet-b6), both pre-trained on ImageNet. We apply basic data augmentation to two training sets (R-CyanoHABs and S-CyanoHABs) and data augmentation based on StyleGAN2-ADA and Neural Style Transfer to the RS-CyanoHABs set. In addition, we use transfer learning and fine-tuning for better and faster convergence.

StyleGAN2-ADA together with Neural Style Transfer (the latter has been used for refinement) have succeeded in generating cyanobacterial patches sufficiently realistic in terms of texture, color, variety, and quality. Augmentation based on these advanced methods provided the model with a remarkable improvement in terms of generalization performance in the validation set and on real test images.

The proposed approach can be applied in different domains involving outdoor images. The most obvious is in the use of ASVs for the detection of oil spills at sea. As in the case of cyanobacteria, there are not enough images to train semantic segmentation models for oil slicks or refined products, as it is necessary to wait for leaks of this nature to occur. Moreover, these slicks also exhibit similar characteristics as the cyanobacteria patches, in terms of spatial–spectral and textural information. Another clear area of application is in precision agriculture for detecting weeds in maize and cereal fields for site-specific treatments. Weed patches can be obtained to train the models before the weeds appear so that the MVS on board a tractor is ready when the patches appear. Again, weed patches exhibit spectral and textural information.

In all these scenarios, in general, and particularly in the aquatic ones, appears the surface reflection effect. In this study, different experiments have been carried out to analyze its influence, trying to minimize the intensity saturation caused by it. The conclusion is that the proposed network models achieve satisfactory results. This does not prevent, that in the future the use of polarizing filters can be considered as a physical component added to the MVS. Also, and without needing to exclude one from the other, additional image radiometric processes, such as homomorphic filtering, to reduce the lighting component while preserving the spatial textural characteristics, can be very useful. In any case, the UNet++ with EfficientNet, as backbone, remain as the most promising models.

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.

Acknowledgments

This work was supported by the Research Project IA-GES-BLOOM-CM (Y2020/TCS-6420) of the Synergic program of the Comunidad Autónoma de Madrid, Spain and by the Research Project AMPBAS (RTI2018-098962-BC21) of the National Societal Challenges program of the Spanish Ministry of Science, Innovation and Universities. The first author, Fredy Barrientos-Espillco, is supported a scholarship by PRONABEC, Ministry of Education of Peru. Clara I. López-González, is supported by a FPU Ph.D. scholarship from the Spanish Ministry of Universities. The authors thank the anonymous referees for their very valuable comments and suggestions.

References

[1] J.L. Graham, N.M. Dubrovsky, S.M. Eberts, Cyanobacterial harmful algal blooms and U.S., in: Geological Survey Science Capabilities, U.S., Geological Survey, Reston, VA, 2016, http://dx.doi.org/10.3133/ofr20161174.

[2] J. Huisman, G.A. Codd, H.W. Paerl, B.W. Ibelings, J.M.H. Verspagen, P.M. Visser, Cyanobacterial blooms, Nat. Rev. Microbiol. 16 (2018) 471–483, http://dx.doi.org/10.1038/s41579-018-0040-1.

[3] D.P. Hamilton, S.A. Wood, D.R. Dietrich, J. Puddick, Costs of harmful blooms of freshwater cyanobacteria, in: Cyanobacteria, John Wiley & Sons, Ltd, 2014, pp. 245–256, http://dx.doi.org/10.1002/9781118402238.ch15.

[4] J.R. Yang, H. Lv, A. Isabwe, L. Liu, X. Yu, H. Chen, J. Yang, Disturbance-induced phytoplankton regime shifts and recovery of cyanobacteria dominance in two subtropical reservoirs, Water Res. 120 (2017) 52–63, http://dx.doi.org/10.1016/j.watres.2017.04.062.

[5] S.M. Feist, R.F. Lance, Genetic detection of freshwater harmful algal blooms: A review focused on the use of environmental DNA (eDNA) in microcystis aeruginosa and prymnesium parvum, Harmful Algae 110 (2021) 102124, http://dx.doi.org/10.1016/j.hal.2021.102124.

[6] F. Tan, P. Xiao, J.R. Yang, H. Chen, L. Jin, Y. Yang, T.-F. Lin, A. Willis, J. Yang, Precision early detection of invasive and toxic cyanobacteria: A case study of raphidiopsis raciborskii, Harmful Algae 110 (2021) 102125, http://dx.doi.org/10.1016/j.hal.2021.102125.

[7] N. Chen, S. Wang, X. Zhang, S. Yang, A risk assessment method for remote sensing of cyanobacterial blooms in inland waters, Sci. Total Environ. 740 (2020) 140012, http://dx.doi.org/10.1016/j.scitotenv.2020.140012.

[8] J.P. Cannizzaro, B.B. Barnes, C. Hu, A.A. Corcoran, K.A. Hubbard, E. Muhlbach, W.C. Sharp, L.E. Brand, C.R. Kelble, Remote detection of cyanobacteria blooms in an optically shallow subtropical lagoonal estuary using MODIS data, Remote Sens. Environ. 231 (2019) 111227, http://dx.doi.org/10.1016/j.rse.2019.111227.

[9] C. Hu, A novel ocean color index to detect floating algae in the global oceans, Remote Sens. Environ. 113 (2009) 2118–2129, http://dx.doi.org/10.1016/j.rse.2009.05.012.

[10] T. Kutser, L. Metsamaa, N. Strömbeck, E. Vahtmäe, Monitoring cyanobacterial blooms by satellite remote sensing, Estuar. Coast. Shelf Sci. 67 (2006) 303–312, http://dx.doi.org/10.1016/j.ecss.2005.11.024.

[11] Y.-H. Ahn, P. Shanmugam, J.-H. Ryu, J.-C. Jeong, Satellite detection of harmful algal bloom occurrences in Korean waters, Harmful Algae 5 (2006) 213–231, http://dx.doi.org/10.1016/j.hal.2005.07.007.

[12] H. Cao, L. Han, L. Li, A deep learning method for cyanobacterial harmful algae blooms prediction in Taihu Lake, China, Harmful Algae 113 (2022) 102189, http://dx.doi.org/10.1016/j.hal.2022.102189.

[13] G. Hitz, F. Pomerleau, M.-E. Garneau, C. Pradalier, T. Posch, J. Pernthaler, R.Y. Siegwart, Autonomous inland water monitoring: Design and application of a surface vessel, IEEE Robot. Autom. Mag. 19 (2012) 62–72, http://dx.doi.org/10.1109/MRA.2011.2181771.

[14] E. Romero-Vivas, F.D.V. Borstel, C.J. Pérez-Estrada, D. Torres-Ariño, J.F. Villa-Medina, J. Gutiérrez, On-water remote monitoring robotic system for estimating the patch coverage of anabaena sp. filaments in shallow water, Environ. Sci. Process. Impacts 17 (2015) 1141–1149, http://dx.doi.org/10.1039/C5EM00097A.

[15] G. Hitz, F. Pomerleau, C. Pradalier, T. Posch, J. Pernthaler, R.Y. Siegwart, Lizhbeth: Toward autonomous toxic algae bloom monitoring, in: 2011 IEEE Conf. Intell. Robots Syst. Workshop Robot. Environ. Monit., San Francisco, USA, 2011, p. 5.

[16] W. Touzout, Y. Benmoussa, D. Benazzouz, E. Moreac, J.-P. Diguet, Unmanned surface vehicle energy consumption modelling under various realistic disturbances integrated into simulation environment, Ocean Eng. 222 (2021) 108560, http://dx.doi.org/10.1016/j.oceaneng.2020.108560.

[17] N. Patki, R. Wedge, K. Veeramachaneni, The synthetic data vault, in: 2016 IEEE Int. Conf. Data Sci. Adv. Anal. DSAA, 2016, pp. 399–410, http://dx.doi.org/10.1109/DSAA.2016.49.

[18] P.-T. Nguyen, T.-H. Tran, V.-H. Dao, H. Vu, Improving gastroesophageal reflux diseases classification diagnosis from endoscopic images using StyleGAN2-ADA, in: N.H.T. Dang, Y.-D. Zhang, J.M.R.S. Tavares, B.-H. Chen (Eds.), Artif. Intell. Data Big Data Process, Springer International Publishing, Cham, 2022, pp. 381–393, http://dx.doi.org/10.1007/978-3-030-97610-1_30.

[19] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, H. Greenspan, GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification, Neurocomputing 321 (2018) 321–331, http://dx.doi.org/10.1016/j.neucom.2018.09.013.

[20] V. Sandfort, K. Yan, P.J. Pickhardt, R.M. Summers, Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in CT segmentation tasks, Sci. Rep. 9 (2019) 16884, http://dx.doi.org/10.1038/s41598-019-52737-x.

[21] A. Mikołajczyk, M. Grochowski, Data augmentation for improving deep learning in image classification problem, in: 2018 Int. Interdiscip, PhD Workshop IIPhDW, 2018, pp. 117–122, http://dx.doi.org/10.1109/IIPHDW.2018.8388338.

[22] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, T. Aila, Training generative adversarial networks with limited data, Adv. Neural Inf. Process. Syst. 33 (2020) 12104–12114.
[23] L. Gatys, A. Ecker, M. Bethge, A neural algorithm of artistic style, J. Vis. 16 (2016) 326, http://dx.doi.org/10.1167/16.12.326.
[24] R. Novak, Y. Nikulin, Improving the neural algorithm of artistic style, 2016, http://dx.doi.org/10.48550/arXiv.1605.04603.
[25] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, GANs trained by a two time-scale update rule converge to a local Nash equilibrium, in: Adv. Neural Inf. Process. Syst., Curran Associates, Inc., 2017, https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html (accessed March 23, 2023).
[26] Z. Wu, C. Shen, A. van den Hengel, Wider or deeper: Revisiting the ResNet model for visual recognition, Pattern Recognit. 90 (2019) 119–133, http://dx.doi.org/10.1016/j.patcog.2019.01.006.
[27] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2018) 834–848, http://dx.doi.org/10.1109/TPAMI.2017.2699184.
[28] D. Arthur, S. Vassilvitskii, K-means++: The advantages of careful seeding, 2006, http://ilpubs.stanford.edu:8090/778/?ref=https://githubhelp.com (accessed March 23, 2023).
[29] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Springer US., Boston, MA, 1981.
[30] L. Steccanella, D. Bloisi, J. Blum, A. Farinelli, Deep learning waterline detection for low-cost autonomous boats, in: M. Strand, R. Dillmann, E. Menegatti, S. Ghidoni (Eds.), Intell. Auton. Syst., Vol. 15, Springer International Publishing, Cham, 2019, pp. 613–625, http://dx.doi.org/10.1007/978-3-030-01370-7_48.
[31] S. Griffith, G. Chahine, C. Pradalier, Symphony lake dataset, in: 2017 Int. J. Robot. Res. IJRR, 2017, https://dream.georgiatech-metz.fr/datasets/symphony-lake-dataset-2014/ (accessed March 23, 2023).
[32] D. Mukherkjee, P. Saha, D. Kaplun, A. Sinitca, R. Sarkar, Brain tumor image generation using an aggregation of GAN models with style transfer, Sci. Rep. 12 (2022) 9141, http://dx.doi.org/10.1038/s41598-022-12646-y.
[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Adv. Neural Inf. Process. Syst., Curran Associates, Inc., 2014, https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html (accessed March 23, 2023).
[34] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4401–4410, https://openaccess.thecvf.com/content_CVPR_2019/html/Karras_A_Style-Based_Generator_Architecture_for_Generative_Adversarial_Networks_CVPR_2019_paper.html (accessed March 23, 2023).
[35] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, Analyzing and improving the image quality of stylegan: 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR), in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2020, pp. 8107–8116, http://dx.doi.org/10.1109/CVPR42600.2020.00813.
[36] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2015, ArXiv14091556 Cs. http://arxiv.org/abs/1409.1556 (accessed March 23, 2023).
[37] L.A. Gatys, M. Bethge, A. Hertzmann, E. Shechtman, Preserving color in neural artistic style transfer, 2016, http://dx.doi.org/10.48550/arXiv.1606.05897.
[38] G. Pajares, J. Manuel de la Cruz, A wavelet-based image fusion tutorial, Pattern Recognit. 37 (2004) 1855–1872, http://dx.doi.org/10.1016/j.patcog.2004.03.010.
[39] K. Wada, Labelme: Image polygonal annotation with python, 2022, http://dx.doi.org/10.5281/zenodo.5711226.
[40] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, A. Haworth, A review of medical image data augmentation techniques for deep learning applications, J. Med. Imaging Radiat. Oncol. 65 (2021) 545–563, http://dx.doi.org/10.1111/1754-9485.13261.
[41] J. Wang, L. Perez, The effectiveness of data augmentation in image classification using deep learning, Convolutional Neural Netw. Vis. Recognit. 11 (2017) 1–8.
[42] H.-C. Shin, N.A. Tenenholtz, J.K. Rogers, C.G. Schwarz, M.L. Senjem, J.L. Gunter, K.P. Andriole, M. Michalski, Medical image synthesis for data augmentation and anonymization using generative adversarial networks, in: A. Gooya, O. Goksel, I. Oguz, N. Burgos (Eds.), Simul. Synth. Med. Imaging, Springer International Publishing, Cham, 2018, pp. 1–11, http://dx.doi.org/10.1007/978-3-030-00536-8_1.
[43] A. Buslaev, V.I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, A.A. Kalinin, Albumentations: Fast and flexible image augmentations, Information 11 (2020) 125, http://dx.doi.org/10.3390/info11020125.
[44] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (2015) 211–252, http://dx.doi.org/10.1007/s11263-015-0816-y.
[45] Y.A. LeCun, L. Bottou, G.B. Orr, K.-R. Müller, Efficient BackProp, in: G. Montavon, G.B. Orr, K.-R. Müller (Eds.), Neural Netw. Tricks Trade, second ed., Springer, Berlin, Heidelberg, 2012, pp. 9–48, http://dx.doi.org/10.1007/978-3-642-35289-8_3.
[46] S. Asgari Taghanaki, K. Abhishek, J.P. Cohen, J. Cohen-Adad, G. Hamarneh, Deep semantic segmentation of natural and medical images: a review, Artif. Intell. Rev. 54 (2021) 137–178, http://dx.doi.org/10.1007/s10462-020-09854-1.
[47] V. Zyuzin, T. Chumarnaya, Comparison of unet architectures for segmentation of the left ventricle endocardial border on two-dimensional ultrasound images, in: 2019 Ural Symp. Biomed. Eng. Radioelectron. Inf. Technol. USBEREIT, 2019, pp. 110–113, http://dx.doi.org/10.1109/USBEREIT.2019.8736616.
[48] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W.M. Wells, A.F. Frangi (Eds.), Med. Image Comput. Comput.-Assist. Interv. – MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241, http://dx.doi.org/10.1007/978-3-319-24574-4_28.
[49] Z. Zhou, M.M. Rahman Siddiquee, N. Tajbakhsh, J. Liang, Unet++: A nested U-net architecture for medical image segmentation, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. DLMIA ML-CDS 2018, in: Lecture Notes in Computer Science, vol. 11045, Springer, Cham, 2018, http://dx.doi.org/10.1007/978-3-030-00889-5_1.
[50] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Honolulu, HI, USA, 2017, pp. 6230–6239, http://dx.doi.org/10.1109/CVPR.2017.660.
[51] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, A. Torralba, Semantic understanding of scenes through the ade20k dataset, Int. J. Comput. Vis. 127 (2019) 302–321, http://dx.doi.org/10.1007/s11263-018-1140-0.
[52] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Comput. Vis. – ECCV 2018, Springer International Publishing, Cham, 2018, pp. 833–851, http://dx.doi.org/10.1007/978-3-030-01234-2_49.
[53] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, 2017, ArXiv170605587 Cs. http://arxiv.org/abs/1706.05587 (accessed March 23, 2023).
[54] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: 2017 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR, 2017, pp. 2117–2125, http://dx.doi.org/10.1109/CVPR.2017.106.
[55] A. Kirillov, R. Girshick, K. He, P. Dollar, Panoptic feature pyramid networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6399–6408, https://openaccess.thecvf.com/content_CVPR_2019/html/Kirillov_Panoptic_Feature_Pyramid_Networks_CVPR_2019_paper.html (accessed March 12, 2023).
[56] M. Tan, Q.V. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in: Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, 2019, pp. 9–15, http://proceedings.mlr.press/v97/tan19a.html.
[57] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778, https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html (accessed March 23, 2023).
[58] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034, https://openaccess.thecvf.com/content_iccv_2015/html/He_Delving_Deep_into_ICCV_2015_paper.html (accessed March 23, 2023).
[59] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, J. Garcia-Rodriguez, A survey on deep learning techniques for image and video semantic segmentation, Appl. Soft Comput. 70 (2018) 41–65.
[60] J. Merkow, A. Marsden, D. Kriegman, Z. Tu, Dense volume-to-volume vascular boundary detection, in: S. Ourselin, L. Joskowicz, M.R. Sabuncu, G. Unal, W. Wells (Eds.), Med. Image Comput. Comput.-Assist. Interv. - MICCAI 2016, Springer International Publishing, Cham, 2016, pp. 371–379, http://dx.doi.org/10.1007/978-3-319-46726-9_43.
[61] T.S. Sharan, S. Tripathi, S. Sharma, N. Sharma, Encoder modified U-net and feature pyramid network for multi-class segmentation of cardiac magnetic resonance images, IETE Tech. Rev. (2021) 1–13, http://dx.doi.org/10.1080/02564602.2021.1955760.

[62] S. Jadon, A survey of loss functions for semantic segmentation, in: 2020 IEEE Conf. Comput. Intell. Bioinforma. Comput. Biol. CIBCB, 2020, pp. 1–7, http://dx.doi.org/10.1109/CIBCB48159.2020.9277638.
[63] P. Iakubovskii, Qubvel/segmentation_models.pytorch, 2022, https://github.com/qubvel/segmentation_models.pytorch (accessed March 23, 2023).
[64] B. Ltd, Algae on the water free stock photo - public domain pictures, 2023, https://www.publicdomainpictures.net/en/view-image.php?image=103402&picture=algae-on-the-water (accessed March 28, 2023).
[65] C. Fischer, Bloom of cyanobacteria in a freshwater pond, 2014, https://commons.wikimedia.org/wiki/File:Cyanobacteria_Aggregation1.jpg (accessed March 28, 2023).
[66] Lamiot, Inflorescence planctonique. They are cyanophyceae (blue algae), 2009, https://commons.wikimedia.org/wiki/File:CyanobacteriaLamiot2009_07_26_237.jpg (accessed March 28, 2023).
[67] J.G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N.V. Chawla, F. Herrera, A unifying view on dataset shift in classification, Pattern Recognit. 45 (2012) 521–530, http://dx.doi.org/10.1016/j.patcog.2011.06.019.

Fredy Barrientos-Espillco is currently a Ph.D. student in Computer Science and Engineering at University Complutense of Madrid. He received his M.S. degree in Computer Engineering from University of Deusto. His research interests include deep learning, computer vision, generative models, and reinforcement learning.

Esther Gascó received in 2021 her master's degree in Systems Engineering and Automation at University Complutense (Madrid) and UNED. She is team leader at ANOVO developing customized solutions in computer systems. She is a Ph.D. candidate at University Complutense of Madrid. Her research focuses on pattern recognition and computer vision under deep learning.

Clara I. López-González in 2021 received her master's degree in mathematics and applications at Padova (Italy) and Bordeaux (France) universities under an Erasmus Mundus program (European Commission). She is supported by a FPU contract by the Spanish Ministry of Universities, with research interest in pattern recognition and computer vision based on deep learning.

María José Gómez-Silva is an Assistant Professor of Systems Engineering and Automation at University Complutense of Madrid. She holds a Ph.D. in Electrical Engineering, Electronics and Automation from the University Carlos III of Madrid with honors. Her research interests include computer vision, machine learning, deep learning, and robotics.

Gonzalo Pajares is a full professor of Computer Vision and Artificial Intelligence at University Complutense of Madrid. From 1990 till 2004 he worked at Indra Space and INTA in Remote Sensing. His research interests include machine vision and pattern recognition, including deep learning. He is an editorial board member of several indexed journals in these areas.