AI IN SUPPORT TO WATER QUALITY MONITORING

C. A. Biraghi [1*], M. Lotfian [1,2], D. Carrion [1], M. A. Brovelli [1]

[1] Department of Civil and Environmental Engineering, Politecnico di Milano – Lecco Campus, Via Gaetano previati 1/c, 23900 Lecco, Italy - (carloandrea.biraghi, maryam.lotfian, daniela.carrion, maria.brovelli)@polimi.it

[2] University of Applied Sciences and Arts Western Switzerland, Institute INSIT, 1400, Yverdon-les-Bains

Commission IV, WG IV/4

KEY WORDS: Artificial Intelligence; Convolutional Neural Networks; Citizen Science; Water monitoring;

ABSTRACT:

This study explores the possibility of using Artificial Intelligence (AI) as a means to support water monitoring. More precisely, it addresses the issue of the quality and reliability of Citizen Science data. The paper addresses the tools and data of the SIMILE (Informative System for the Integrated Monitoring of Insubric Lakes and their Ecosystems) project in order to develop an open pre-filtering system for Volunteer Geographic Information (VGI) of lake water monitoring at the global scale. The goal is to automatically determine the presence of harmful phenomena (algae and foams) in the images uploaded by citizen scientists to reduce the time required for a manual check of the contributions. The task is challenging because of the heterogeneity of the data that consist in geotagged pictures taken without specific instructions. For this purpose, different tools and deep learning techniques have been tested (Clarifai platform, a Convolutional Neural Network (CNN), and an object detection algorithm called faster Region-based CNN (R-CNN). The original dataset composed by the observations of SIMILE – Lake Monitoring application, has been integrated with the results of both keyword and image searches on web engines (Google, Bing, etc) and crawling Flickr data. The performances of the different algorithms are presented for their capability of detecting the presence and correctly labelling the phenomenon together with some possible strategies to improving them in the future.

1. INTRODUCTION

1.1 SIMILE project and its context

Waterbodies play a key role in the mitigation of the impact of climate change. They also represent an essential resource available to billions of people for multiple uses. The importance of waterbodies is further highlighted by their inclusion in the sustainable development agenda (SDGs 6, 14, UN.org, 2019). Lake ecosystems are highly exposed to the consequences of global warming and the impact of human activities (Adrian et al., 2009; Vincent, 2009). The quality of lake water needs to be preserved, and recent scientific and technological development could provide a significant support to the cause. SIMILE (Informative System for the Integrated Monitoring of Insubric Lakes and their Ecosystems) is a project that involves academia (Politecnico di Milano – Lecco Campus; Fondazione Politecnico; SUPSI - University of Applied Sciences and Arts of Southern Switzerland), research bodies (Water Research Institute - National Research Council) and institutions (Lombardy Region; Ticino Canton in Switzerland), which are cooperating partners in the preservation of water quality in the Lugano-, Maggiore- and Como lakes. The main strategy of this cooperation aims at integrating the existing monitoring protocols with data coming from recently developed geospatial tools and techniques, such as the processing of satellite images (Luciani et al., 2020), of high frequency *in situ* sensors and also Citizen Science (Brovelli et al., 2019).

SIMILE – Lake Monitoring (Biraghi et al., 2020; Pessina et al., 2020) is a recently released cross-platform, open-source, mobile application, which has been developed to support the activities of the project related to Citizen Science (CS). The mobile application enables private citizens to share their observations of the lake environment through geo-referenced images of algae, foams and litter, and the measurements of water parameters (transparency, temperature, pH, etc.). In addition, it promotes water-related events, it features a glossary and also a set of useful links that help improving the user's knowledge and awareness of the lake ecosystem. Registration to the app is optional and most of its functionalities can be used without registering. This means that everyone – including amateur and even malevolent users – can contribute freely to the project. This clearly implies the possible occurrence of irrelevant or inappropriate content.

Even though the mobile application is designed to facilitate the upload of contributions by users, it is not the optimal tool for their subsequent management. In fact, it only contains an agile map view where all the observations and measurements are displayed with the same marker and one needs to open each observation in order to explore their content. A web application (https://simile.como.polimi.it/SimileWebAdministrator/faces/index.xhtml) has been developed to allow the partners of the project, and ultimately also environmental agencies, to edit and better analyse the data uploaded through the app. Moreover, the web application offers the possibility to filter observations according to time, position and all the other attributes that the mobile application features. This is a powerful tool – complementary to the mobile application – that enriches the set of data available that environmental agencies can study. However, it may also be regarded as a burden and an additional task for the employees who will actually elaborate the data.

At the end of the project, the integration of new technologies (including the two apps) with the existing routines will be evaluated by the institutional partners. For this reason, the added value on the scientific level must be balanced with the additional workload weighing on the employees of the environmental agencies. As a consequence, the present study explores the development of a system for the pre-filtering of data which exploits Artificial Intelligence (AI) with the purpose of reducing

* Corresponding author

the need of a manual check of the observations performed by the employees, thus simplifying the general workflow.

1.2 Integration of AI and CS

Until recently, Artificial Intelligente (AI) and Citizen Science (CS) were considered distinctly and were used independently (McClure et al., 2020). Their integration, however, can be helpful in addressing some of the challenges existing in both areas. Indeed, CS can be the source that provides large amounts of labelled data in order to feed and train Machine Learning (ML) algorithms, whereas ML can support CS in automating the validation of data or in sustaining user participation by giving automatic feedback to the volunteers. Even though the inclusion of AI in CS projects is expanding to various areas such as astronomy (Beaumont et al. 2014) and neuroscience (Keshavan et al. 2019), the main focus of their cooperative exploitation has been on biodiversity studies (Green et al., 2020; Lotfian et al., 2019; Terry et al., 2020; Torney et al., 2019).
This article aims at integrating ML in the SIMILE project in order to introduce an automatic identification and the validation of observations regarding water quality. More precisely, the presence of foams and algae is considered, as it is the most relevant and reported issue in the Insubric lakes. The availability of studies on the use of AI in order to identify phenomena linked to water quality is limited, and the majority of them – to the best of our knowledge – is based on detecting litter on the water surface (Garcia-Garin et al., 2021; Wolf et al., 2020). AI is also used to detect harmful algal blooms (HAB) by analysing satellite images (Hill et al., 2020). The final goal of the present study is to implement a tool for the detection of algae and foams in the images uploaded by the users of the SIMILE – Lake Monitoring mobile application. The findings of this research have been compared to those of a similar study (Samantaray et al., 2018) which deals with the detection of HAB both from aerial and from ground surveys.
The following chapter details all the steps required to develop this tool, starting from a definition of the dataset needed. The preparation of the data that will feed the alternative ML algorithms presented will be subsequently assessed, then, in the final section of the chapter, their performance will be analysed.

2. METHOD

2.1 Dataset acquisition

The initial dataset consisted of the images uploaded by the users of the SIMILE – Lake Monitoring mobile application, integrated with archive images received from the partners of the project. This dataset featured 35 images of algae, 32 of foams and 14 of clean water. Even though the images were very precise and context specific, the dataset was limited if compared to the one needed to train a ML model. In order to enlarge it, a web search has been carried out by exploiting search engines Google and Bing and by using as keywords the two phenomena (algae; foams), their synonyms (algal bloom and scum; froth and spume) and the corresponding Italian words (alghe e fioriture algali; schiume). This manual research produced similar results both on Google and Bing, and it helped collect nearly one-hundred valid imagess for each phenomenon. As a means to acquire contents in other languages or unlabelled ones, a complementary search by image has been performed using 10 input images for each of the two phenomena on 5 search engines (Google, Bing, TinEye, StackPhoto, Shuttershock). Table 1 shows the number of valid images found using each input image on the search engines detailed above.

	ID	Google	Bing	TinEye	Stack Photo	Shutter shock
Algae	1	6	3	0	2	0
	2	0	0	0	0	0
	3	15	8	0	0	0
	4	0	0	0	0	0
	5	20	11	0	0	2
	6	24	30	0	0	2
	7	86	0	0	1	2
	8	7	0	0	0	0
	9	3	0	0	0	0
	10	8	1	0	0	1
	Tot	169	53	0	3	7
Foams	1	1	4	0	0	0
	2	1	1	0	0	0
	3	0	9	0	0	0
	4	0	0	0	0	0
	5	0	3	0	0	2
	6	4	0	0	0	0
	7	0	0	0	0	0
	8	4	2	0	1	0
	9	1	4	0	1	0
	10	2	3	0	0	1
	Tot	13	26	0	2	1

Table 1. Valid images for the different search engines used for the search by image

Thanks to this additional, manual search, 232 valid images of algae (73% from Google) and 49 valid images of foams (45% from Bing) have been collected. In very limited cases, images of algae have also been found by using images of foams as an input (5 images), and viceversa (2 images). A further research into the Flickr dataset has been performed, first by downloading all the images containing the hashtag “algae” or “foams”, then by manually checking all the results. Through this method, 82 valid images of algae have been gathered out of a total of 1012 downloaded images, and 101 images of foams out of 4094 downloaded images.
Some recurring elements have been noticed among the images that portrayed neither foams nor algae: from now on these will be labelled “false positives”. These elements generally trick the search engine as in some cases the images may look like the phenomena our study investigates. For instance, false positives for algae were landscapes with grass, water lilies, clean water surrounded by trees and vegetation, paintings and red metal powder. False positives for foams, instead, were cloudy satellite images, clouds in general, ice floating on water, snowflakes, glasses with drops of water, stones, waves and sun reflected on water. As the results will show later, the occurrence of certain false positives in the search by image could give some anticipations about the behaviour of the algorithm. More precisely, images containing the above-mentioned elements will have a high probability of being detected as false positives, especially as far as foams are concerned.
On the one hand, this process has considerably increased the number of images at our disposal. On the other, it has introduced elements of disturbance. As a matter of fact, algae and foams are unstable objects without fixed dimensions, whose shape, extension, colour and appearance (i.e. compact, dispersed, linear, scattered) may vary. They can be found in the sea, in lakes and smaller basins, and in rivers. Pictures can be taken from different observation points and with various inclinations of the camera. The diversity found in this larger dataset of images could not have been found by looking at the SIMILE project context alone, which takes into consideration only the Insubric Lakes.

Considering the global interest for these phenomena, including into the dataset a significant variety of their manifestations is clearly an added value for the pre-filtering system, because it helps increasing its usability beyond the framework of the SIMILE project.

2.2 Data preparation

After downloading the images, several pre-processing operations needed to be performed in order to elaborate the final dataset. The dataset was used to train two different algorithms: a Convolutional neural network (CNN) and an object-detection algorithm called faster Region-based CNN (R-CNN). It is important to notice how the two algorithms – which will be detailed in the following section – partially required different pre-processing operations, as it will be explained below.

Remove duplicate images: Considering that the images were obtained from multiple sources, some of them appeared twice or even more times, but were labelled under different names. In order to remove the duplicates a script was implemented so as to obtain the pixel values of an image. Using a hash function called MD5 (https://en.wikipedia.org/wiki/MD5), a hexadecimal code (an alphanumeric string such as 21933563820f99833ad07aac2e8 18a6b) was generated based on the image pixel value. The code generated was compared among the images: those with an identical value were considered duplicates, then the exceeding copies were removed from the dataset.

Label the images: When using the CNN algorithm, the images could be put in a folder labelled with the name of the phenomenon in order to generate the labels from the structure of the directory. This has been done by using the flow_from_directory function in Keras (https://keras.io/api/preprocessing/image/). In order to label the images using the object-detection algorithm, the phenomenon needed to be identified within each image by drawing one or more bounding boxes around it, and assigning a label corresponding to each phenomenon (algae or foam). In order to perform this action, a tool for image annotation written in Python, called LabelImg (https://github.com/tzutalin/labelImg), was used. The output consisted of the image and a homonymous XML file (Extensible Markup Language) containing the coordinates of the bounding boxes and the label name. The labelled images were then uploaded on a platform called Roboflow (https://roboflow.com/), which helped performing the following processing steps, needed only for the object-detection algorithm.

Augment the images: Once the images along with their XML files were added to Roboflow, it was possible to verify whether all of them were labelled, and to control and modify the location of bounding boxes in case they were outside the border of the image. In order to increase the amount of input data two augmentation steps were applied to all images. A horizontal flip and an image rotation of ±15 produced two augmented versions for each image. Figure 1 shows a sample of the dataset following data augmentation.

Export to the required output format: Once the dataset was ready, it could be exported to the format needed for the model. Roboflow provides the opportunity to choose the output format based on the algorithm one wishes to train. Then one can either download the dataset on a local machine or get the URL of the data. The format required in this study was TFRecord, a TensorFlow format that stores the sequence of binary records (https://www.tensorflow.org/tutorials/load_data/tfrecord).

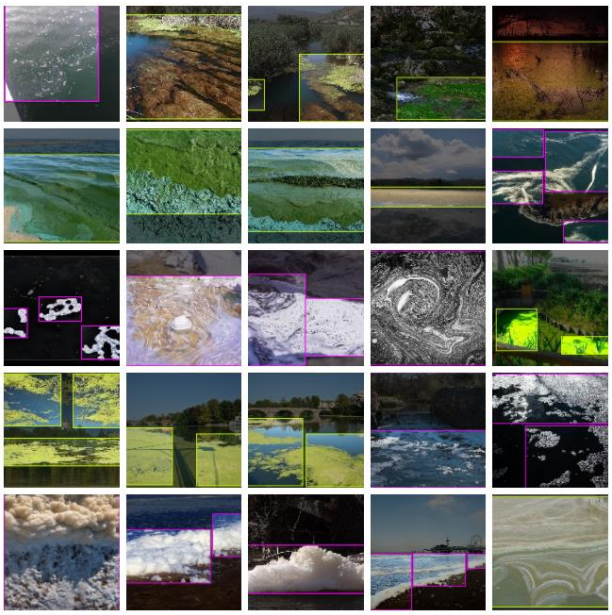

Figure 1. Sample of dataset images after data augmentation

2.3 Tools and approaches

In order to perform an automatic identification of water quality phenomena (here algae and foams), three approaches were considered, as explained below.

Clarifai: Clarifai (https://www.clarifai.com/) is an AI platform for computer vision, natural language processing and automatic speech recognition. It offers pre-trained models as well as the opportunity of training a model using a custom dataset. The services can be used through Clarifai API, which has a high-speed response return and can be integrated in AI-based mobile or web applications. Our first attempt was done by building a custom model using the initial dataset of 60 images of both foam and algae classes. Because of limitations concerning the number of free API calls (1000 free calls), the black box regarding the structure and parameters of the model and the peculiarity of the case studied, it was decided to construct and train a new CNN custom model. It is important to note that the pre-trained models available on Clarifai (https://www.clarifai.com/developers/pre-trained-models) are very efficient in the detection of objects in images. The pre-trained models include a general model detecting a variety of objects, as well as more specific models such as face recognition and the detection of humans and cars, to name a few. Even though this initial attempt with Clarifai was characterised by the availability of a limited dataset (30 images for algae and 27 for foams) and other limitations, it showed promising results that encouraged us to continue with a more detailed approach.

Convolutional neural network (CNN): CNN is a deep learning algorithm which initially has been used to study the brain's visual cortex and is now widely used to identify patterns for image processing and sound recognition (Albawi et al., 2018). Thanks to the availability of large amounts of data, as well as to the recent advances made in computing power, CNN models have achieved a high level of performance in the identification of patterns in complicated visual tasks, which sometimes is superior to the abilities of a human being (Gu et al., 2017). When using a CNN model, a kernel (filter) functioning as a moving window is

applied on the image pixels in order to identify and extract the various features in the image (e.g. edges). The convolutional layers (one or more) are responsible for capturing different features with varying levels of detail. By analysing a variety of layers, the algorithm can ultimately achieve a full understanding of the image taken into consideration. In order to extract dominant features and to downsample the images, a further pooling layer is added, which similar to the convolutional layer the kernel moves within the image and it returns a new set of pixel values depending on the type of kernel. There are two types of pooling – maximum pooling and average pooling – that return respectively the maximum and the average values found by the kernel (O'Shea and Nash 2015). Finally, the fully connected layer performs the duty of the traditional Artificial Neural Network (ANN with input layer, hidden layers, and the outer layer) and conducts the final classification and scoring. Figure 2 illustrates a simple architecture of a CNN model.

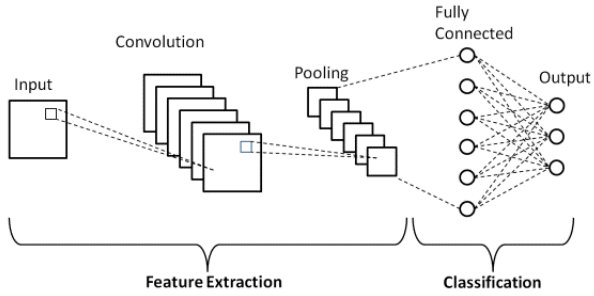

Figure 2. Representation of the Architecture of a simple CNN (Balaji, 2020)

In line with the purposes of the present study, a CNN model composed of three hidden convolutional layers and a fully connected layer has been trained using a two-dimensional max pooling. The fully connected layer also featured the ReLU activation function (https://en.wikipedia.org/wiki/Rectifier_(neural_networks)). A summary of the specific architecture of the model used is shown in Figure 3, which illustrates that each of the tree convolution layers used is followed by a pooling layer. The initial input image was resized to 150x150 pixels and, as Figure 3 clearly shows, its size was reduced gradually as a consequence both of the convolution and of the pooling, up to reaching a final size of 17x17 pixels. The output of the final pooling layer is flattened in order to produce a unidimensional vector of all the values that feed the fully connected layers (see ANN, the classification part as shown in Figure 2). The fully connected layer has 512 neurons, and since we are doing a binary classification (1: algae, 0: foam), the size of the output is 1.

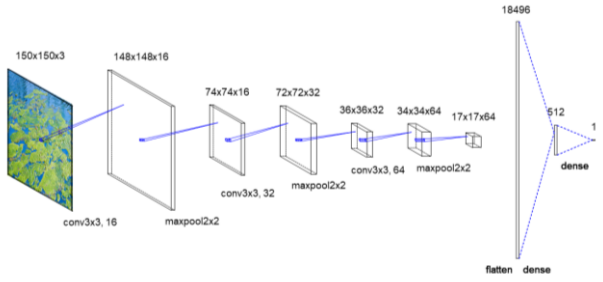

Figure 3. Architecture of the implemented CNN model

The first training of the model has been done by using uniquely the observations from the SIMILE mobile application, whereas the second training has exploited the extended dataset (755 images equally distributed between two classes). 90% of the data was used for training purposes and 10% to validate the performance of the model, taking into consideration validation accuracy and validation loss. TensorFlow (https://www.tensorflow.org/) and Keras (https://keras.io) were used in order to build, train and evaluate the model, whereas the computation was done on the Google Collaboratory platform (https://colab.research.google.com/), which allows free GPU access. Because of the heterogeneity of the phenomena analysed – variation in texture and colour, presence of false positives – it may be complex for the model to understand which type of phenomenon it is observing, especially if the dataset of images is limited. Moreover, CNN exclusively allowed to predict whether a particular image represented an algae or a foam. The model did not give any information on the location of the phenomenon within the image, neither on the possible presence of both phenomena in a single image. As a consequence, it was decided to train an object detection model which uses CNN to predict both the phenomenon and its precise location in an image.

Object detection: While image classification models work on the probability of an object to be present in an image, object detection algorithms predict the presence of objects as well as their location in an image (e.g. as bounding boxes). Object detection algorithms typically work by proposing the potential regions in an image where an object may exist, then by classifying the regions in connection with the object(s) of interest. One of the known object detection algorithms is R-CNN (Region-based CNN) developed by Ross Girshick et al. (2014). This approach uses an algorithm for image segmentation called "selective search" in order to determine the possible regions where an object may be located (approximately 2000 region proposals per image). The regions are then passed to a CNN model which generates a feature vector from each region proposal. Finally, a support vector machine (SVM) model performs a classification of the objects found and identifies the location of the objects in the image. Training this kind of model is computationally expensive and the test data take a long time to be predicted (approximately 49 seconds per image). For this reason, an enhancement of the algorithm – called fast R-CNN (Girshick, 2015) – was proposed in 2015. Fast R-CNN uses the same approach to determine the regions where an object may be located, then all region proposals are passed to CNN as one single input, rather than sending them one by one as with R-CNN. The performance of the model is improved in that the training and detection time are significantly reduced to about 2 seconds per image. Still, fast R-CNN proves to be computationally expensive as it uses a traditional image segmentation algorithm to propose regions (i.e. selective search algorithm). As a consequence, a yet additional version of R-CNN was proposed – faster R-CNN (Ren et al., 2015) – which used convolutional networks to propose regions, rather than an algorithm of external region proposal. Faster R-CNN requires both less training time and less time to detect test images (0.2 seconds per image), which makes it suitable for integration in applications of real-time object detection. For the purposes of the present study a customised faster R-CNN algorithm was trained using TensorFlow object detection API, which offers pre-trained weights. The API offers pre-trained models on a COCO (common objects in context) dataset which can be used and adapted on customised data. The "faster_rcnn_inception_v2" pre-trained model was used, and a training using 1000, 10'000, and 20'000 steps was performed, comparing computation time and model performance for each run. In order to evaluate the performance of the model, average recall and mean average precision (mAP) metrics were considered. Here follows a textual and visual (Figure 4) summary of the steps performed to train the custom faster R-CNN model:

1. perform image annotation by drawing bounding boxes around the object of interest in the image and label them with *algae* or *foams* tag;
2. apply data augmentation (e.g. image rotation, horizontal or vertical flip, etc.) to address the lack of input images;
3. finalise the dataset and extract the required data format for the model (in our case: TFRecords format);
4. select and configure the pre-trained model;
5. train the model using the updated dataset and monitor loss, mAP and recall while training;
6. evaluate model performance and adjust its parameters;
7. observe the results of predicted bounding boxes on the test dataset.

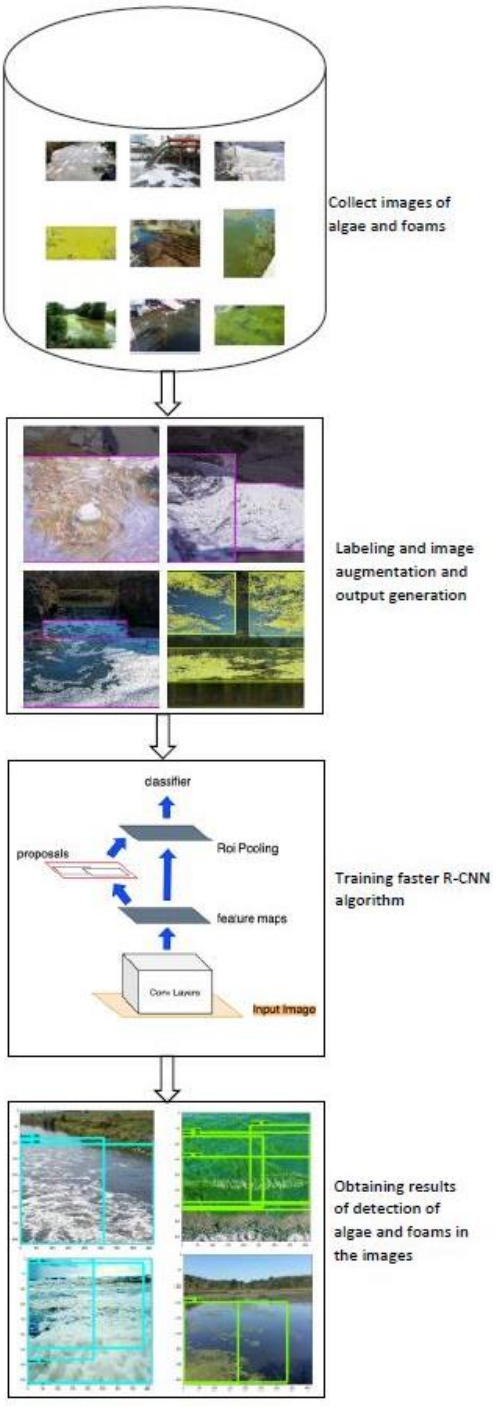

Figure 4. Training steps of R-CNN model

3. RESULTS AND DISCUSSION

3.1 Model performance

CNN: The results obtained by using CNN show that the model has learned relatively well on the training dataset. However, in the validation dataset we can observe some noises in the accuracy with values fluctuating between over 90% and 50% (Figure 5). Although after epoch 120 we observe more stability in the validation accuracy/loss, due to the observed gap between training and validation accuracy/loss it can be concluded that the model is suffering from high variance and thus overfitting. This behaviour could be due to the relatively small amount of data used to train the CNN or to the heterogeneity in the dataset. However, in order to have a better understanding of the cause of this fluctuation, changes in different parameters need to be tested in the future. A solution for the limited size of the dataset could be found by using transfer learning, that is, by using the weights of the pre-trained models on large datasets. The present study has not tested this solution, so it remains as a hint for future investigations.

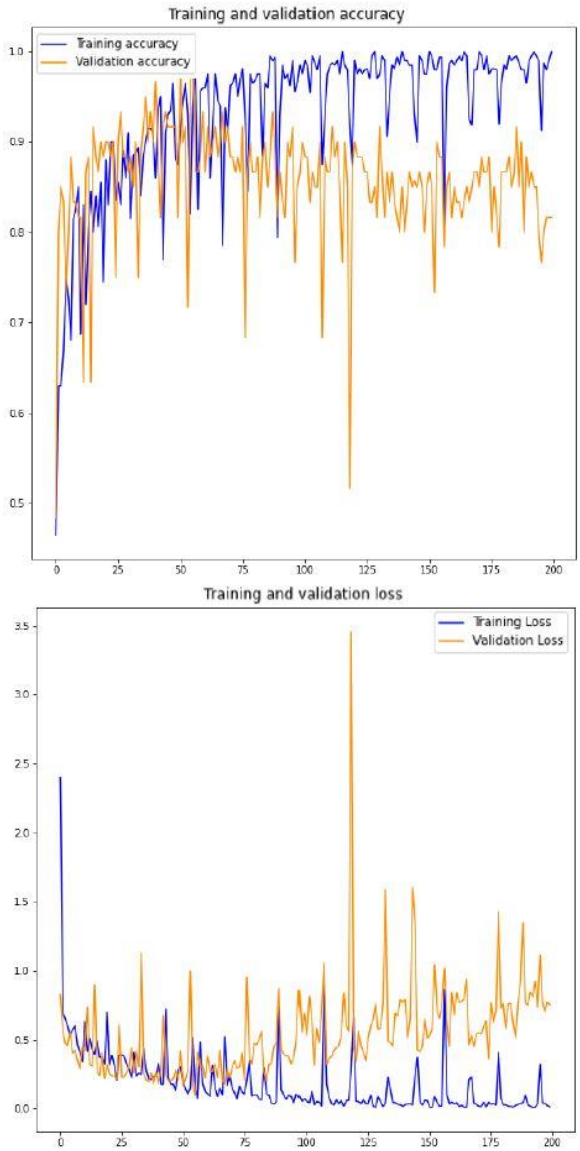

Figure 5. CNN performance. x axis: number of epochs, y axis Top: Accuracy; Bottom: Loss.

Object detection: the three runs were compared with regard to model performance in order to consider the ability of the model to classify the object and to define its location correctly. A possible index for model evaluation is a measurement of the overlap between the predicted bounding box and the ground truth bounding box, which is called IoU (Intersection over Union, see Figure 6).

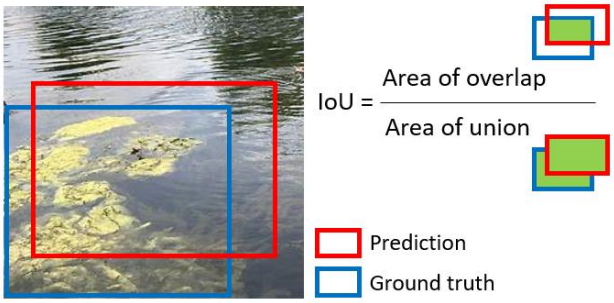

Figure 6. Intersect over Union (IoU), visual explanation

Therefore, IoU can be set as a threshold in order to consider whether a prediction is true or whether it is a false positive. For instance, setting IoU threshold to 0.5, if the IoU of the prediction is above this threshold, it is considered as true positive, if below it is considered as a not precise detection. The IoU threshold value was set to 0.5, then the average recall and mAP were compared.

Training steps	Average recall	mAP	Training time
1000	44.5 %	12.5 %	Approx. 30 min
10,000	41.6 %	17.6 %	Approx. 5 hrs
20,000	44.6 %	19.9 %	Approx. 8 hrs

Table 2. Indicators of model performance

The results shown in Table 2 highlight that with 20'000 steps the model performs better with a higher mAP. Even though the results of the recall performed with 1'000 steps are closer to those performed with 20'000 steps – and also higher than the recall with 10'000 steps – the mAP increased as we trained the model with more steps. That is to say, the model learns early to predict the phenomena, but the location of detected phenomena improves by increasing the number of steps. The results can be compared with the few existing studies on object detection of HAB (Kumar & Bhandarkar, 2017; Samantaray et al., 2018). Even though the recall and mAP for faster R-CNN are lower than what they achieved, considering that the object of identification were two phenomena, and also that the dataset was heterogeneous and small, the results are still promising for the detection of more than one phenomenon, given that the model is trained on a larger dataset. The results of predicted classes and bounding boxes on some of the test set images (using the model trained with 20'000 steps) are presented in support to the discussion of model performance (Figures 7-11). The model performs better in the detection of algae than foams. Several cases of false positives have been detected in which clouds, stones or light reflected on water were identified as foams. However, the positive aspect is that no false negative was detected for foams, and all the false positive bounding boxes featured in images that included also actual foams (Figure 8) or algae (Figure 9). It is interesting to note that no cases of false positives for algae were detected. However, there were cases where algae were not detected (n. 3 false negative), especially in those images where the phenomenon was particularly extended and covered the whole image. In those cases, a contrast with clean water or other elements was not available (Figure 10). The testing of the model with images of clean water showed that in no image the presence of algae was predicted (no false positives). On the contrary, as far as foams are concerned, most of the images containing the reflection of light on clean water were predicted as false positives, as expected (Figure 11). In conclusion, the model works well with algae, but it needs more investigation regarding foams. The model correctly predicts foams when foam is in the image, but it also predicts many false positives.

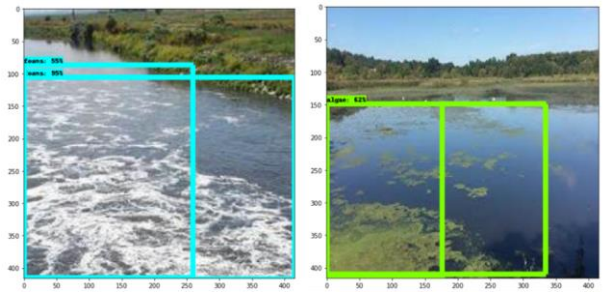

Figure 7. True positives for foams and algae (correctly detected and located)

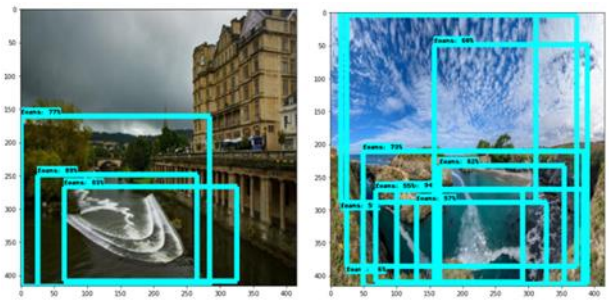

Figure 8. True and false positives (clouds, right) for foams

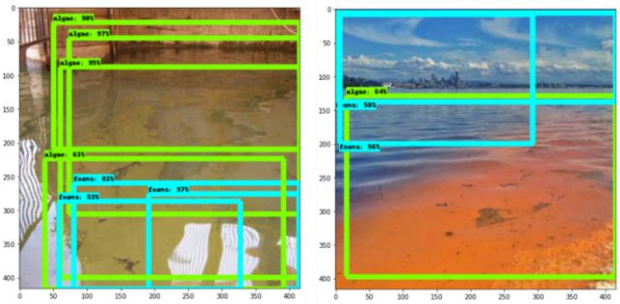

Figure 9. True positives for algae and false positive for foams (reflection of light, left and clouds, right)

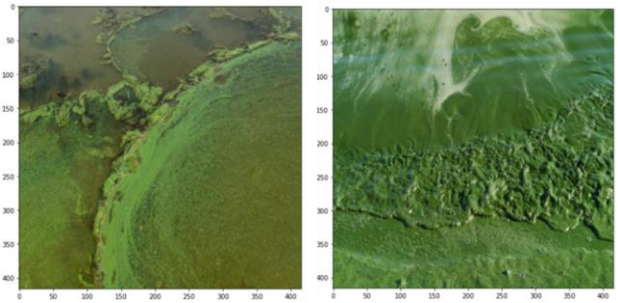

Figure 10. False negatives for algae (not detected)

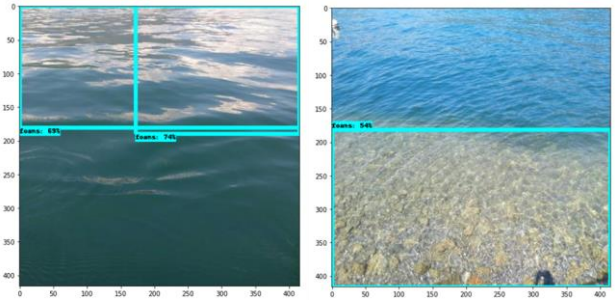

Figure 11. False positives for foams on clean water

3.2 Conclusions

Advances in ML and particularly in computer vision have resulted in many studies that integrate such techniques in their research. CS is among the areas whose integration with ML techniques has received attention in recent years. This article discussed how to integrate ML in a CS application focused on detecting harmful phenomena (i.e. algae and foams) that affect the quality of lake water. Various algorithms have been tested in order to address the aim of an automatic identification of algae and foams in images collected by citizen scientists. Among the various approaches tested, the faster R-CNN object detection algorithm proved to be more suitable, with a performance closer to what other, similar studies have achieved.

However, this research presents some aspects that would benefit from further investigation. For example, one could try the proposed approach using a larger dataset, as the one used here has proved to be too small when considering the complexity and peculiarity of the issue that this study investigated. A larger dataset could also allow for the introduction of more specific tags in order to better distinguish the phenomena analysed and their various manifestations (e.g., compact, linear or scattered algae or foams), hopefully reducing the number of false positives.

In addition to this, among the available algorithms for object detection, only faster R-CNN was trained. However, other algorithms such as SSD (Single Shot Detector), YOLO (You Only Look Once) or R-FCN (Region-based Fully Convolutional Networks) could be trained in order to have a clearer understanding of which performs best in the detection of harmful phenomena in lake water. Finally, another important aspect that can be explored is that of performing predictions considering the location of an image, and not only its content, in such a way that if a model predicts algae or foam in an area with a low probability of occurrence for such phenomena, this will negatively affect the reliability of the prediction.

At the current stage of development this tool may support environmental agencies, but it still presents some aspects that need to be improved, so it cannot fully substitute a manual check if total accuracy in the manifestation of these phenomena is required.

ACKNOWLEDGEMENTS

The research described in this paper is part of SIMILE project (ID: 523544), which has been funded with the support from the European Commission within the Interreg Italy-Switzerland 2014-2021 programme.

REFERENCES

Adrian, R., O'Reilly, C. M., Zagarese, H., Baines, S. B., Hessen, D. O., Keller, W., Livingstone, D. M., Sommaruga, R., Straile, D., Van Donk, E., Weyhenmeyer, G. A., & Winder, M. (2009). Lakes as sentinels of climate change. Limnology and Oceanography.https://doi.org/10.4319/lo.2009.54.6_part_2.2283

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2018). Understanding of a convolutional neural network. Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017, 2018-Janua, 1–6. https://doi.org/10.1109/ICEngTechnol.2017.8308186

Balaji, S. (2020). *Binary Image classifier CNN using TensorFlow*. https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697

Biraghi, C. A., Pessina, E., Carrion, D., & Brovelli, M. A. (2020). VGI Visualisation to support participatory lake monitoring: The case study of simile project. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, 43(B4), 237–244. https://doi.org/10.5194/isprs-archives-XLIII-B4-2020-237-2020

Brovelli, M. A., Cannata, M., & Rogora, M. (2019). SIMILE, a geospatial enabler of the monitoring of Sustainable Development Goal 6 (ensure availability and sustainability of water for all). ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-4/W20(4/W20), 3–10. https://doi.org/10.5194/isprs-archives-XLII-4-W20-3-2019

Garcia-Garin, O., Monleón-Getino, T., López-Brosa, P., Borrell, A., Aguilar, A., Borja-Robalino, R., Cardona, L., & Vighi, M. (2021). Automatic detection and quantification of floating marine macro-litter in aerial images: Introducing a novel deep learning approach connected to a web application in R. Environmental Pollution, 273. https://doi.org/10.1016/j.envpol.2021.116490

Girshick, R. (2015). Fast R-CNN. Retrieved April 17, 2021, from https://github.com/rbgirshick/

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5). Retrieved April 17, 2021, from http://www.cs.berkeley.edu/~rbg/rcnn.

Green, S. E., Rees, J. P., Stephens, P. A., Hill, R. A., & Giordano, A. J. (2020). Innovations in Camera Trapping Technology and Approaches: The Integration of Citizen Science and Artificial Intelligence. Animals, 10(1). https://doi.org/10.3390/ani10010132

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, L., Wang, G., Cai, J., & Chen, T. (2017). Recent Advances in Convolutional Neural Networks.

Hill, P. R., Kumar, A., Temimi, M., & Bull, D. R. (2020). HABNet: Machine Learning, Remote Sensing Based Detection of Harmful Algal Blooms. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. https://doi.org/10.1109/JSTARS.2020.3001445

Kumar, A. C., & Bhandarkar, S. M. (2017). A Deep Learning Paradigm for Detection of Harmful Algal Blooms.

Lotfian, M., Ingensand, J., Ertz, O., Oulevay, S., & Chassin, T. (2019). Auto-filtering validation in citizen science biodiversity

monitoring: a case study. Proceedings of the ICA, 2, 1–5. https://doi.org/10.5194/ica-proc-2-78-2019

Luciani, G., Bresciani, M., Biraghi, C. A., Ghirardi, N., Carrion, D., Rogora, M., & Brovelli, M. A. (2020). Satellite Monitoring system of Subalpine lakes with open source software: the case of SIMILE project. Baltic J. Modern Computing, Vol. xx, N, 1–3.

McClure, E. C., Sievers, M., Brown, C. J., Buelow, C. A., Ditria, E. M., Hayes, M. A., Pearson, R. M., Tulloch, V. J. D., Unsworth, R. K. F., & Connolly, R. M. (2020). Artificial Intelligence Meets Citizen Science to Supercharge Ecological Monitoring. Patterns, 1(7), 100109. https://doi.org/10.1016/j.patter.2020.100109

Pessina, E., Carrion, D., Biraghi, C. A., & Brovelli, M. A. (2020). Crowdsourcing water quality with the SIMILE app. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. http://image-net.org/challenges/LSVRC/2015/results Samantaray, A., Yang, B., Eric Dietz, J., & Min, B.-C. (2018). Algae Detection Using Computer Vision and Deep Learning.

Terry, J. C. D., Roy, H. E., & August, T. A. (2020). Thinking like a naturalist: Enhancing computer vision of citizen science images by harnessing contextual data. Methods in Ecology and Evolution, 11(2), 303–315. https://doi.org/https://doi.org/10.1111/2041-210X.13335

Torney, C. J., Lloyd-Jones, D. J., Chevallier, M., Moyer, D. C., Maliti, H. T., Mwita, M., Kohi, E. M., & Hopcraft, G. C. (2019). A comparison of deep learning and citizen science techniques for counting wildlife in aerial survey images. Methods in Ecology and Evolution, 10(6), 779–787. https://doi.org/https://doi.org/10.1111/2041-210X.13165

UN.org. (2019). About the Sustainable Development Goals - United Nations Sustainable Development. https://www.un.org/sustainabledevelopment/sustainable-development-goals/

Vincent, W. F. (2009). Effects of Climate Change on Lakes. In Encyclopedia of Inland Waters. https://doi.org/10.1016/B978-012370626-3.00233-7

Wolf, M., van den Berg, K., Garaba, S. P., Gnann, N., Sattler, K., Stahl, F., & Zielinski, O. (2020). Machine learning for aquatic plastic litter detection, classification and quantification (APLASTIC-Q). Environ. Res. Lett, 15, 114042. https://doi.org/10.1088/1748-9326/abbd01