Hand-Crafted Features for Floating Plastic Detection

Matija Sukno and Ivana Palunko

***Abstract*— Plastic waste is a global concern that has a negative impact on the oceans and wildlife health. This paper focuses on detection of floating plastics in aerial images taken from unmanned aerial vehicles (UAVs). It proposes a new method for plastic detection in marine environments, based on SIFT descriptor and color histograms for feature extraction, as an alternative to state-of-the-art object detectors based on convolutional neural networks (CNNs). Our approach is named SURFACE: "SIFT featURes For plAstiC dEtection". We investigate how different color-spaces and image resolutions impact the extraction of SIFT features and compare SURFACE to ResNet CNN. Also, we provide a detailed comparison with YOLO and Faster-RCNN object detection models and show that SURFACE achieves approximately the same accuracy while being faster and less memory consuming. The dataset acquired during this research will be publicly available.**

I. Introduction

Plastic pollution threatens food safety and quality, human health, coastal tourism, and contributes to climate change. Up to 80% of the waste that accumulates on land, ocean surface or seabed is plastic and it is one of the most pressing environmental challenges [1]. The authors in [2] estimate that 4.8 to 12.7 million tonnes of plastic waste entered the oceans in 2010. Moreover, [3] estimates that between 1.1 and 2.4 million tonnes of plastics enters the oceans by rivers every year. In order to efficiently monitor, detect and quantify plastic litter, advanced and automated strategies are needed.

An analysis of hyperspectral information reveals unique shortwave infrared (SWIR) spectral features common to plastics that can be used for aquatic plastic detection [4]. Some recent studies manage to detect floating plastics on hyperspectral images captured from the Sentinel-2A satellite and unmanned aerial vehicles (UAVs) [5], [6]. In [5], the authors select 6 bands of reflectance data (blue, green, red, red edge 2, near infrared, and short wave infrared 1) and two indices (NDVI and FDI) to develop the attribute sets and perform supervised (support vector regression and semi-supervised fuzzy c-means) and unsupervised (K-means and fuzzy C-means) classification. The main drawback of these approaches is that they require hyperspectral information that are obtainable only with high-cost hyperspectral cameras. The authors in [7] manage to successfully quantify floating macro-debris transport by monitoring the river surface with a digital video camera and using image processing techniques based on color difference of the floating macro debris. The algorithm proposed by [8] uses UAVs for beach monitoring and the segmentation threshold method to identify the litter that is then used for distribution assessment. Although techniques based on color difference and image thresholding show promising results in litter quantification, they might experience difficulties when a greater detection precision is required. Lately, most of the studies use UAVs for capturing high quality aerial imagery and state-of-the-art machine learning algorithms for image processing. In [9], the authors perform litter detection on a beach located at the Saudi Arabian Red Sea coastline. They use the sliding window method for extraction of candidate regions, histogram of oriented gradients (HoG) for region description and random forest method for classification. The authors in [10]–[13] train deep learning models for detection, classification and quantification of plastic litter in vegetation, beaches and water channels in dense urban areas. The algorithm proposed in [11] consists of two convolutional neural networks (PLD-CNN and PLQ-CNN). An airborne input image is partitioned into tiles and classification is performed on each tile. PLD-CNN manages to categorize the targets as water, sand, vegetation and plastic litter. PLQ-CNN further distinguishes and enumerates litter items into 11 sub-classes. Marine environments are often filled with hazards such as waves and strong winds that pose a potential risk to drone safety. Deploying low-budget UAVs in such environments is something worth considering. Deep learning models require a lot of processing power, which is not always available on UAVs. Also, in practice it is difficult to collect enough data to train such models.

This paper is focused on detecting plastic piles in marine environments when small amounts of training data are encountered. We set detection of floating plastics as a one-class object detection problem and propose a simpler architecture named SURFACE: "SIFT featURes For plAstiC dEtection", which is based on the SIFT features and SVM for recognition of plastics. Here, we show that objects in marine environments can be detected as local extrema in the image scale-space using difference-of-Gaussians blob detector, which serves as an object proposal module in our architecture. Also, we demonstrate that SURFACE achieves better results in detecting the presence of plastics in extracted regions than ResNet-50 CNN. Finally, throughout our experiments, we show that SURFACE is faster and less memory consuming then Faster-RCNN and YOLO without sacrificing any accuracy. We also provide a new dataset with 162 aerial images of floating plastic piles acquired with UAV.

The remainder of this paper is organized as follows. In Section 2, we define the problem of surface litter detection in

*This work is supported in part by project SeaClear, European Union's Horizon 2020 Research and Innovation Action under grant agreement No. 871295 and in part by project InnovaMare, Interreg IT-HR, European Regional and Development Fund under No.10248782.

Both authors are with Laboratory for Intelligent Autonomous Systems (LARIAT), Department of Electrical Engineering and Computing, University of Dubrovnik, Croatia matija.sukno@unidu.hr; ivana.palunko@unidu.hr

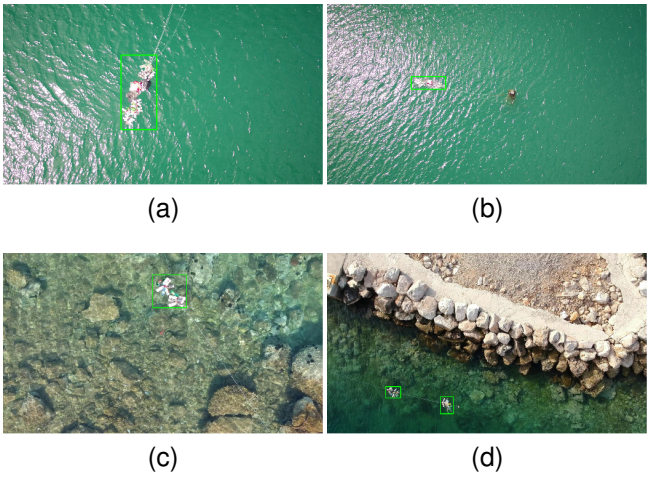

Fig. 1. Aerial images of floating plastics with the presence of glare and in the shallows: **(a)** and **(b)** images are taken from 9m and 20m altitude; **(c)** and **(d)** show floating plastics in shallow sea. Images are taken from 4 and 16 meters.

aerial images. Methodology and the components of proposed detection architecture are described in Section 3. The dataset description and results are given in Section 4. Section 5 concludes the paper.

II. Problem Statement

Aerial images captured during sunny conditions are strongly affected by a glare due to the sunlight reflection from the sea surface. Because of its high reflective index, floating plastics have very similar reflectance properties as glare. This similarity becomes even more prominent with an increase in the altitude from which the aerial image is taken. One of the main challenges is to distinguish between glare and floating plastics. Also, in some coastal areas due to the shallow sea, the seafloor is clearly visible making detection of plastic waste even more challenging (see Fig. 1). In this paper, the detection of plastics is set as an object detection problem where the main task is to draw a bounding box around the floating plastic piles despite the glare occurring in the image.

III. Methodology

In this section, we describe the details of the proposed architecture shown in Fig. 2.

A. Difference of Gaussians

In the field of computer vision, blobs refer to regions in an image which are either brighter or darker then their surroundings. Due to its reflectance index, floating plastics often appear as bright blobs in aerial images. The blob detection is usually performed throughout the computation of local extrema in the image scale-space. In this paper, we propose a method for extraction of candidate regions based on the difference-of-Gaussians (DoG) function [14], [15]. The scale-space of an image is defined as a function $L(x, y, \sigma)$ that is produced from the convolution of a variable-scale Gaussian, $G(x, y, \sigma)$, with an input image, $I(x, y)$:

$$G(x, y, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x^2+y^2)/2\sigma^2},$$
$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

where x is the distance from the origin in the horizontal direction, y is the distance from the origin in the vertical direction, σ is the standard deviation and $*$ is the convolution operation at (x, y) position. To efficiently detect extrema locations in the image scale-space, [14] uses the difference-of-Gaussians function $D(x, y, \sigma)$, which is computed from the difference of two nearby scales separated by a constant factor k:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$

In order to detect blobs as the local maxima and minima of $D(x, y, \sigma)$, each point in the difference-of-Gaussians image is compared to its neighbors in the current image and neighbors in the scale above and below. A blob is selected only if it is larger than all of these neighbors or smaller than all of them.

The blob size depends on the value of σ. In order to detect blobs of different sizes, we need to compute DoG at multiple scales. This increases computation time and slows down the extraction of candidate regions. To perform faster extraction, in the architecture proposed herein, we compute the difference of Gaussians only between two scales $L(x, y, 1.6\sigma)$ and $L(x, y, \sigma)$ separated by the factor $k = 1.6$. Also, to detect floating plastics despite various altitudes, changing the σ value is also necessary. We propose the following function f for choosing the σ value based on the altitude h:

$$\sigma = f(h) = \begin{cases} 6, & \text{if } 0 < h \leq 5, \\ 5, & \text{if } 5 < h \leq 10, \\ 4, & \text{if } 10 < h \leq 15, \\ 3, & \text{if } 15 < h \leq 20, \\ 2, & \text{otherwise.} \end{cases}$$

Therefore, together with the aerial image, the input to object proposal module is also the altitude from which the image was taken.

B. Feature extraction

There is a large number of descriptors already proposed based on color, texture and shape [14], [16]–[18]. Because of its speed, robustness and favorable results, we choose the SIFT descriptor to create gradient histograms. Besides the gradient histograms, we also compute a color histogram with 50 bins in all three channels of the input image. Both the gradient and color histogram are concatenated into one feature vector. Finally, to detect the presence of floating plastics in each region, we use the support vector machine (SVM) with a radial basis function.

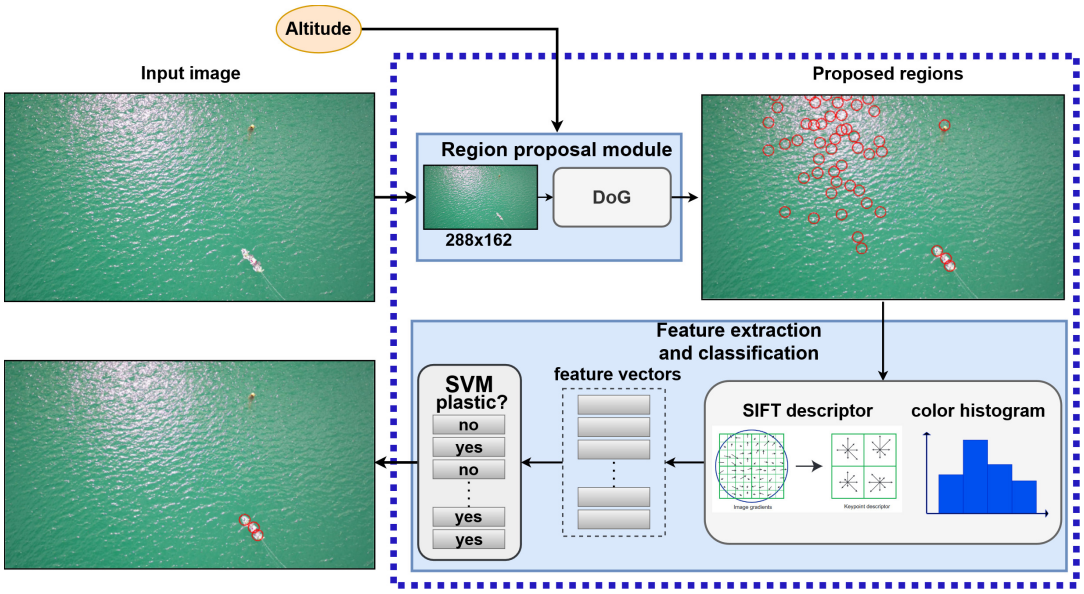

Fig. 2. SURFACE architecture schematics. Inputs to the model are aerial RGB image and altitude from which the image was taken. In order to speed up the computation, input image is resized to 288×162 resolution before applying the difference-of-Gaussians method. For each proposed region, feature vector is constructed by computing SIFT descriptor and color histogram in all three image channels. Finally, SVM is used to detect presence of floating plastics.

SIFT descriptor: SIFT descriptor is a scale and rotation invariant descriptor proposed by [14]. To compute the descriptor, first the image gradient magnitudes $m(x, y)$ and orientations $\theta(x, y)$ are sampled from the region of interest:

$$m(x,y) = \sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2},$$
$$\theta(x,y) = \tan^{-1}\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)},$$

where x is the distance from the origin in the horizontal direction and y is the distance from the origin in the vertical direction. A dominant orientation of the interest region is determined from orientations of the gradient vectors in this region. In order to achieve rotation invariance, the coordinates of the descriptor and gradient orientations are rotated relative to the computed dominant orientation vector. To obtain gradient orientation histograms with 8 orientation planes, the gradient image of the region is sampled over 4×4 grid of locations. Finally, $4\times4\times8 = 128$-dimensional feature vector is formed from the computed histograms.

IV. Results

A. Dataset

Collection: The recorded plastic litter was collected over a period of five months. It mostly consists of bottles (milk, yogurt, water and juice), bags, snack wraps, lids and plastic ropes. Most of the litter items got punctured and deformed in the collection process. In each of the data acquisitions performed on different locations and weather conditions, litter was placed on the surface. During the first data acquisition, which took place in the Mandrač bay, Dubrovnik, Croatia, aerial images were captured using a DJI Spark quadrotor with 1920×1080 video resolution. The second acquisition took place in the same bay also using a DJI Spark. This time we used bigger piles and different bay locations. On the third, longest acquisition, which took place in Bistrina, Croatia, images were captured in the morning and afternoon using a DJI Matrice 210 V2 quadrotor equipped with a Zenmuse X5S camera with 3840×2160 video resolution. For images sampled from the captured videos refer to [19].

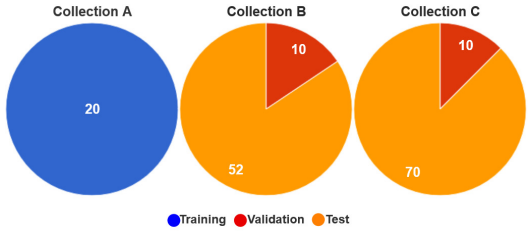

Fig. 3. Portion of images in training, validation and test sets. Training set contains only 20 images. Validation and test sets contain 20 and 122 images, respectively.

Training, validation and test sets: To form the training, validation and test sets, we first create collections A, B and C by sampling the videos from the first, second and third acquisition, respectively. From approximately 15 minutes of a video, we sample only 162 images to ensure there are no duplicate images. Also, to make the dataset more challenging, we use only collection A for training. The validation and test sets are formed from collections B and C (see Fig. 3).

B. Evaluation on different color-spaces and resolutions

Our architecture is a two-stage detection not capable of end-to-end training. So we first extract region proposals from the training set and then train SVM on extracted feature vectors from those regions. To extract regions used for performance evaluation, we use the region proposal

TABLE I

DISTRIBUTION OF EXTRACTED REGIONS FROM TRAINING, VALIDATION AND TEST IMAGES.

Set	Number of positive examples	Number of negative examples
Training	1248	2689
Validation	61	445
Test	319	2572

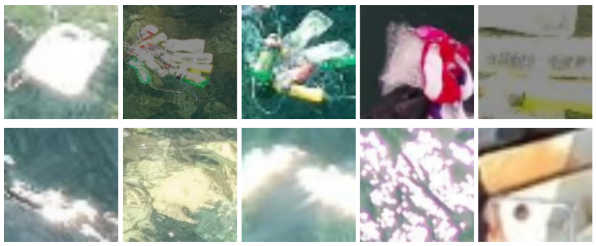

Fig. 4. Examples of positive (top) and negative (bottom) patches used for training and evaluation.

TABLE II

CLASSIFICATION ACCURACY WITH DIFFERENT COLOR-SPACES.

Color-space	Validation set accuracy	Test set accuracy
RGB	0.926	0.863
HSV	0.931	0.874
LAB	**0.978**	0.903
YUV	0.923	0.869
YCbCr	0.918	0.877

module for both validation and test sets. In order to create a significantly larger amount of train data, we use the data augmentation strategies (i.e., rotations, Gaussian blur, color jitter) to expand the training set and the difference-of-Gaussians method operating on multiple scales with different σ values for the extraction of regions (Table I, Fig. 4).

Color-space selection: The performance evaluation with different color-spaces was carried-out on low resolution images of size 480×270. Except the standard RGB color-space, we also consider the LAB, HSV, YCbCr and YUV color-spaces for descriptor and color histogram computation. Eventually for future evaluations, we choose the configuration with highest classification accuracy on the validation set.

Performance on higher resolution: Training and evaluating the same model on images with higher resolution results in a significant drop in validation accuracy (see Table III). The SIFT descriptor is using a 128-dimensional feature vector for description of local regions. With an increase in image quality, the size of candidate regions is also increasing and a higher dimensionality is required to summarize computed gradients. To overcome the drop in accuracy, we customize the existing descriptor by splitting the proposed region into three sub-regions and computing a SIFT descriptor for each. The computed descriptors are then merged into a $3 \times 128 = 384$-dimensional feature vector.

TABLE III

CLASSIFICATION ACCURACY ON DIFFERENT RESOLUTIONS. IN SURFACE (EXTENDED) MODEL, BASIC SIFT DESCRIPTOR IS REPLACED WITH AN EXTENDED VERSION.

Image resolution	Model	Validation accuracy	Test accuracy
480x270	SURFACE	0.978	0.903
1280x720	SURFACE	0.939	0.899
1920x1080	SURFACE	0.875	0.873
1280x720	SURFACE (extended)	0.977	0.902
1920x1080	SURFACE (extended)	0.977	0.922
480x270	ResNet-50	0.903	0.802
1280x720	ResNet-50	0.847	0.777

Comparison to ResNet-50: To get a better understanding of the results obtained in Table II, we compare SURFACE with ResNet-50 CNN [20]. To train the ResNet model on extracted regions, we use a fine-tuning approach by taking the pre-trained model on Image-Net and fine-tune only the last convolutional block and fully-connected layers. Results show that SURFACE achieves better classification accuracy on both high and low resolution images (Table III).

C. Comparison to the state-of-the art object detection architectures

State-of-the-art results in object detection are achieved by the architectures based on CNNs [21]–[24]. We can divide these architectures into two main types: one-stage and two-stage. One-stage architectures prioritize inference speed while two-stage prioritize detection accuracy. Authors in [25] compare Faster-RCNN, R-FCN and SSD architectures in terms of inference speed, detection accuracy and memory consumption. They observe that R-FCN and SSD are faster on average while Faster-RCNN tends to lead to slower but more accurate models. We choose Faster-RCNN as a representative of two-stage and YOLO as a representative of one-stage detection architectures and compare them with SURFACE in terms of detection accuracy, inference speed and memory consumption.

Replacing non-maximum suppression with greedy grouping: As already mentioned, object detection models typically extract thousands proposals per image to correctly localize objects of interest. After a model completes its forward pass, most of the proposed regions get classified as "background" and they get automatically discarded. The nondiscarded regions are considered likely to contain specific objects. Yet, in most cases there are many overlapping regions left that point to the same object. To further filter proposals and get better predictions, the object detection pipeline is usually using the non-maximum suppression algorithm [26]. However, in our case, models are trained to detect small bottle groups or individual items on a limited dataset. To detect bigger piles, instead of using the non-maximum suppression to eliminate overlapping regions, we

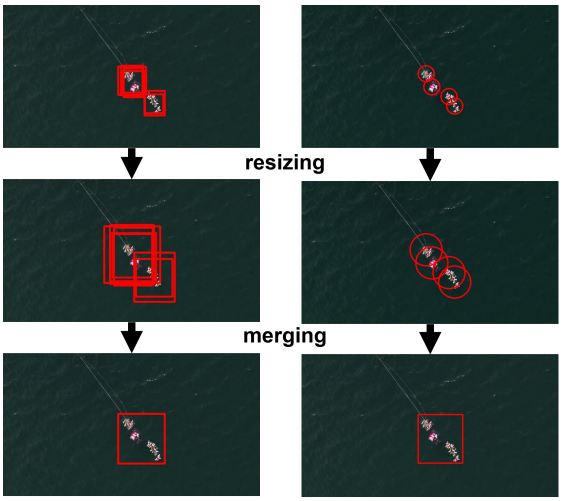

Fig. 5. Greedy grouping applied on YOLO (left) and SURFACE (right) to form the final bounding box predictions. Intersection check was done on resized regions that survived the forward pass. All intersecting regions are merged together to form the final bounding box.

use the greedy grouping algorithm to merge those regions. The algorithm iteratively eliminates regions that intersect by merging them until there are no more intersecting regions left (see Fig. 5).

Expanding training set with data augmentation: Data augmentation is one of the most popular techniques for enhancing the performance of deep learning models [27]. It is especially useful to prevent overfitting when we are dealing with insufficient amount of training data, which is often the case in marine robotics applications. Using geometric transformations (flipping, translation, noise injection), color jitter and mosaic augmentation, we expand the initial training set to 1000 images. This expanded set is used as the basis for training YOLO and Faster-RCNN in later experiments. The state-of-the-art models are usually trained and evaluated on a large-scale datasets such as PASCAL VOC and COCO for object detection or ImageNet for classification tasks. A common practice when training such models on a problem specific datasets is to use transfer learning by taking the model that has learned useful features on one dataset and adapting that model and its developed features to another dataset. Therefore, to train YOLO and Faster-RCNN on our expanded training set, we use the fine-tuning approach. Instead of training the whole model, we freeze the backbone and use it only as a feature extractor. Here, freezing implies that layer weights will not be changed during training.

Quantitative results: To obtain the detection results, we run multiple experiments by changing the amount of training data with different image resolutions. For each experiment, we report AP^{50} detection accuracy on the test set (Table IV). We also compare models in terms of inference speed and memory consumption on both high and low resolution images. All comparisons were done on a Intel(R) Core(TM) i7-9750H CPU @2.60 GHz.

Reducing the amount of training data results in a lower detection accuracy. When there is enough training data, all three models approximately give the same detection performance.

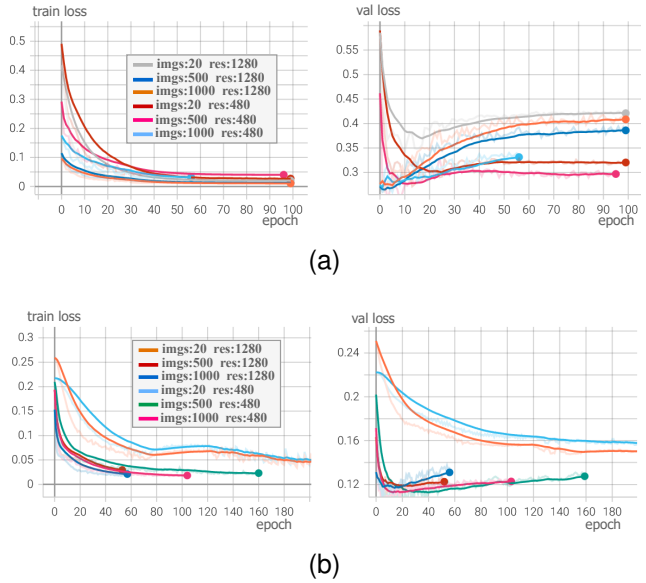

Fig. 6. Progress of loss functions during the training of Faster-RCNN (a) and YOLO (b). To reduce overfitting and obtain better results, we use early stopping as additional regularization method.

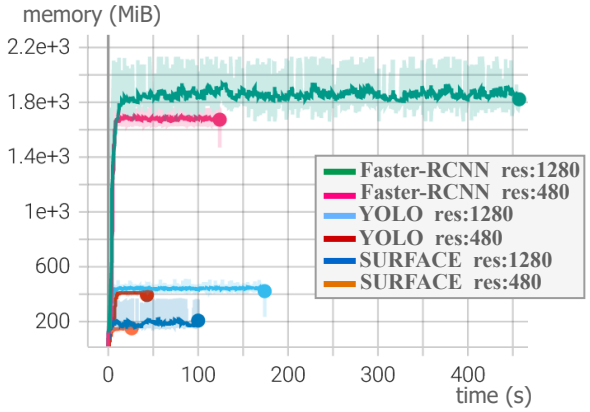

Fig. 7. Memory consumption as a function of time.

When no data augmentation is used, both Faster-RCNN and SURFACE outperform the YOLO model. Fig. 6 shows the progress of loss functions on training and validation sets for YOLO and Faster-RCNN during the training phase. It is obvious that in most cases both models start to overfit after 20 epochs (when trained on more images, this happens even sooner, since there are more training steps in each epoch). When training on higher resolutions, overfitting is even more pronounced, which results in a lower detection performance on test images. Table V shows that SURFACE is approximately 2 times faster than YOLO and 8 times faster then Faster-RCNN. It also needs less memory to run (see Fig. 7). For a video demonstration regarding online detection using the SURFACE model mounted on the UAV, refer to [19].

V. Conclusion

This paper presents a new method for detection of floating plastics based on hand-crafted features. We show that in marine environments the object proposal method can be im-

TABLE IV

DETECTION RESULTS. FOR FASTER-RCNN AND YOLO EXPANDED TRAINING SET CONTAINS 1000 IMAGES CREATED WITH DATA AUGMENTATION. FOR SURFACE IT CONTAINS 3937 REGIONS, EXTRACTED ALSO USING DATA AUGMENTATION (SEE TABLE I).

Amount of training data	**YOLO**		**Faster-RCNN**		**SURFACE**	
	480	**1280**	**480**	**1280**	**480**	**1280**
100% of expanded training set	82.18	79.34	83.10	82.54	82.62	81.56
50% of expanded training set	80.37	75.92	82.9	78.69	80.4	78.75
Original 20 images (no augmentation)	47.33	39.19	69.90	72.42	70.91	74.21

TABLE V

INFERENCE TIME AND MEMORY CONSUMPTION. INFERENCE TIME IMPLIES THE AVERAGE TIME ELAPSED DURING THE FORWARD PASS ON A SINGLE IMAGE. MEMORY CONSUMPTION IS MEASURED AS MAXIMUM PEAK IN MEMORY.

	YOLO		**Faster-RCNN**		**SURFACE**	
	480	**1280**	**480**	**1280**	**480**	**1280**
time [s]	0.19	1.23	0.84	3.53	0.09	0.51
memory [MiB]	421	515	1601	2155	161	384

plemented as a difference-of-Gaussians function to efficiently extract regions that could potentially contain plastic objects. By setting detection of plastics as a one-class detection problem, we successfully replace CNN with the SIFT descriptor for feature extraction. SURFACE achieves approximately the same detection accuracy as the state-of-the-art detectors when trained on sufficient amount of data, having lower inference time and memory consumption. When trained on low amounts of data, SURFACE outperforms both models.

Despite the fact that we had to limit our comparisons by running the experiments on CPUs only, SURFACE can be rewritten to run on GPUs, which we leave for future work. It is also worth noticing that, although we use SIFT as our base descriptor, any other descriptor can be easily integrated into an existing architecture to achieve potential boosts in speed and accuracy.

REFERENCES

[1] D. Barnes, F. Galgani, R. Thompson, and M. Barlaz, "Accumulation and fragmentation of plastic debris in global environments," *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, vol. 364, pp. 1985–1998, 2009.

[2] J. Jambeck, R. Geyer, C. Wilcox, T. Siegler, M. Perryman, A. Andrady, R. Narayan, and K. Law, "Marine pollution. plastic waste inputs from land into the ocean," *Science (New York, N.Y.)*, vol. 347, pp. 768–771, 2015.

[3] L. Lebreton, J. Van der Zwet, J.-W. Damsteeg, B. Slat, A. Andrady, and J. Reisser, "River plastic emissions to the world's oceans," *Nature Communications*, vol. 8, 2017.

[4] S. Garaba, J. Aitken, B. Slat, H. Dierssen, L. Lebreton, O. Zielinski, and J. Reisser, "Sensing ocean plastics with an airborne hyperspectral shortwave infrared imager," *Environmental Science & Technology*, vol. 52, p. 11699–11707, 2018.

[5] B. Basu, S. Sannigrahi, A. Basu, and F. Pilla, "Development of novel classification algorithms for detection of floating plastic debris in coastal waterbodies using multispectral sentinel-2 remote sensing imagery," *Remote Sensing*, vol. 13, 2021.

[6] K. Topouzelis, A. Papakonstantinou, and S. Garaba, "Detection of floating plastics from satellite and unmanned aerial systems (plastic litter project 2018)," *International Journal of Applied Earth Observation and Geoinformation*, vol. 79, pp. 175–186, 2019.

[7] T. Kataoka and Y. Nihei, "Quantification of floating riverine macrodebris transport using an image processing approach," *Scientific Reports*, vol. 10, 2020.

[8] Z. Bao, T. Sodango, E. Shifaw, X. Li, and J. Sha, "Monitoring of beach litter by automatic interpretation of unmanned aerial vehicle images using the segmentation threshold method," *Marine Pollution Bulletin*, vol. 137, pp. 388–398, 2018.

[9] C. Martin, S. Parkes, Q. Zhang, X. Zhang, M. F. McCabe, and C. M. Duarte, "Use of unmanned aerial vehicles for efficient beach litter monitoring," *Marine Pollution Bulletin*, vol. 131, pp. 662–673, 2018.

[10] S. Bak, D. Hwang, H. Kim, and H. Yoon, "Detection and monitoring of beach litter using UAV image and deep neural network," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-3/W8, pp. 55–58, 2019.

[11] M. Wolf, K. Berg, S. Garaba, N. Gnann, K. Sattler, F. Stahl, and O. Zielinski, "Machine learning for aquatic plastic litter detection, classification and quantification (aplastic-q)," *Environmental Research Letters*, vol. 15, p. 114042, 2020.

[12] M. Tharani, A. W. Amin, M. Maaz, and M. Taj, "Attention neural network for trash detection on water channels," *ArXiv*, vol. abs/2007.04639, 2020.

[13] K. Kylili, I. Kyriakides, A. Artusi, and C. Hadjistassou, "Identifying floating plastic marine debris using a deep learning approach," *Environmental Science and Pollution Research*, vol. 26, p. 17091–17099, 2019.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[15] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors. (2014) scikit-image: image processing in Python. [Online]. Available: https://bit.ly/334GZa8

[16] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, pp. 1615–1630, 2005.

[17] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine region," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, pp. 1265–1278, 2005.

[18] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[19] M. Sukno and I. Palunko. (2022) Floating plastic detection repository. [Online]. Available: https://1drv.ms/u/s!AqWDDDzUOxjbiEjc6vL8xl7kaZfE?e=u0sAdX

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[22] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *ArXiv*, vol. abs/2004.10934, 2020.

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Computer Vision – ECCV 2016*. Springer International Publishing, 2016, pp. 21–37.

[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[25] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3296–3297.

[26] J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 4507–4515.

[27] C. Shorten and T. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, 2019.