Boat detection

Chiara Guizzaro
Identity number: 2019293

***Abstract*—The implementation of autonomous systems for vessels traffic organization, naval security and surveillance is crucial for improving sea safety. Thus a boat detection method based on computer vision and machine learning techniques is proposed. In detail the algorithm is composed of two phases: boat localization by Selective Search and boat classification by means of a CNN. Then the approach is tested for different points of view, background elements and illumination, plus the presence of images both with single and multiple boats with several shapes.**

I. Introduction

Vessel traffic services (VTS) are maritime systems that can provide basic information such as position, identity and possible intentions of other ships, meteorological and waterway conditions, or even manage vessels traffic organization preventing dangerous situations or reducing possible inconveniences caused by boats traffic. Also of utmost importance are naval security and surveillance applications which increase sea safety fighting illegal and hostile activities. For these reasons having an automatic system for boat detection based on computer vision and machine learning can be vital.

The proposed approach is based on the usage of Selective Search [1] as regions proposal algorithm, instead a CNN is exploited for regions classification. In addition some image processing techniques are considered to improve the final results. Therefore for this report firstly in Section II there is a listing of some related works, then in Section III there is the presentation of the datasets used, in Section IV the methodology applied and in Section V the experiment results obtained. At the end in Section VI there are some final considerations regarding boat detection task.

II. Related Work

Maritime environment is a really challenging scenario for object detection because there can be different illumination conditions, several view points, a variety of ships types, sizes, shapes and high frequency background objects which can be wrongly classified as boats. This is also true for reflections and raindrops. Nevertheless some systems can be implemented for vessels detection and/or recognition. For example [2] proposed a boat detection algorithm based on VGG19 [3] for boat types classification in Tokyo. Instead in [4] they present a Sea-Land Segmentation-based Convolutional Neural Network (SLS-CNN) framework for boat detection which combines the SLS-CNN detector, saliency computation and corner features. Meanwhile [5] uses a median filter for denoising the input and then uses a CNN for ships detection on satellite images. On the other hand [6] implements an Haar-like classifier to solve the boat detection problem.

III. Dataset

In general an object detection problem such as boat detection has two components: object localization and object recognition. To perform the latter, dataset [7] is used only for training a chosen classifier. In detail for each object in an image, it is defined a ground truth bounding box whose top left (tf) and bottom right (br) corners have coordinates (x_{tl}, y_{tl}) and (x_{br}, y_{br}). Then for each ground truth bounding box a patch is cropped from the respective image and a sample is created. In addition at most other 10 possible samples are created such that the box

$$[x_{tl} + b, y_{tl} + c, x_{br} + b, y_{br} + c] \tag{1}$$

and ground truth box has an interception over union (IoU) greater or equal to 0.9 where b and c are random integers chosen in the range $[-20, 20]$. In this way the total number of boats samples (label 1) is 10777. Instead to create some negative samples (label 0) for each image a maximum of 15 random boxes is created such that the IoU is equal to zero for each true bounding box for that specific image. The total number of no-boats samples is 12641. Therefore the training set $Tr = \{(x_i, y_i)\}_{i=1}^{m}$ has $m = 23418$, where from now on x_i is the i-th sample and $y_i \in \{0, 1\}$ its true label. An example of patches created in such a way are in Figure 1. On the other hand for testing how good the proposed boat detector performs, two other different datasets [8] and [9] are used.

IV. Method

In general an object detector can be in one of the following categories: two-stage detectors or one-stage detectors. The first, it is more fitted for tasks where an high localization and object recognition accuracy are needed, instead the latter achieves an high inference speed. Since the domain of the boat detector in analysis is static images where the main goal is to achieve a high accuracy and the speed factor is secondary, it has been decided to use a two-stage approach based on Selective Search [1] as regions proposal method and the usage of a custom CNN for regions classification, with the addition of some image processing phases. In detail the proposed algorithm follows these steps:

1) Read an input image;
2) Apply a median and a bilateral filter;
3) Use Selective Search ("fast mode") to the filtered image for generate regions proposal;
4) Discard all regions that have the ratio between height and width (or vice versa) bellow a given threshold;
5) Each region proposal represent a patch from the original image that will be resized and classified thanks a CNN;
6) Apply Non-Maximum Suppression (NMS);

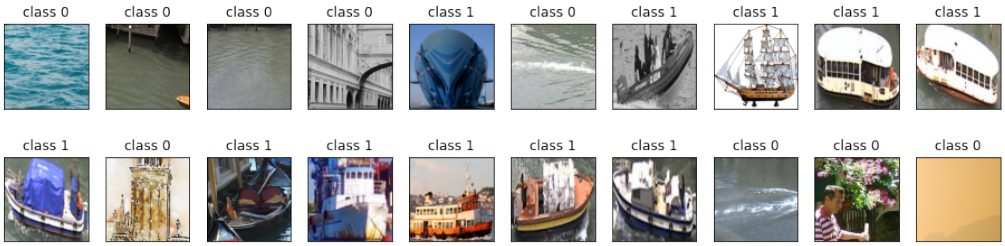


Figure 1: Examples of the patches used as training set for boat recognition.

A. *Selective Search*

Selective Search [1] takes the output of Felzenszwalb and Huttenlocher' [10] graph-based segmentation method as basis for a bottom-up grouping strategy. In specific starting from small regions it continues grouping locations until the whole image becomes a single region. This is done in a greedy way, using different color spaces and computing some similarity metrics between neighbouring regions: texture, size, shape and a final meta-similarity measure. Then the two most similar regions are grouped together. Therefore given all these characteristics Selective Search combines the advantages of both an exhaustive search and segmentation proving to be better than image pyramids and sliding window approaches which are very slow and sensitive to parameters choices. Obviously Selective Search is not the only region proposal algorithm, for example there are also Edge Boxes [11], Objectness [12], BING [13] etc..; but Selective Search remains a common choice because of its high recall results.

Its only remarkable disadvantage is that sometimes when an image has several elements it may require a lot of time for the computation. This was the case also for this boat detection task, thus the introduction of a median and bilateral filter before the application of the regions proposal algorithm. Indeed the filtering reduces greatly the number of regions given in output maintaining the final goal to detect ships. In particular the filters remove the effect that minor elements in the background and reflections of light in the water may have on the detection. An example can be seen at Figure 2. Additionally there is also a check on the ratio r between height h and width w of a given bounding box, such that

$$r = \frac{h}{w} < T_r \tag{2}$$

or

$$r = \frac{w}{h} < T_r \tag{3}$$

where T_r is a given threshold, then the box/region is discarded. This reduce further the number of regions to have as input in the classification phase.

B. *Classification*

The classification process is done thanks a CNN, implemented by scratch. In specific for each region a patch is extracted from the input image and then resized using bilinear interpolation in order to have a fixed size

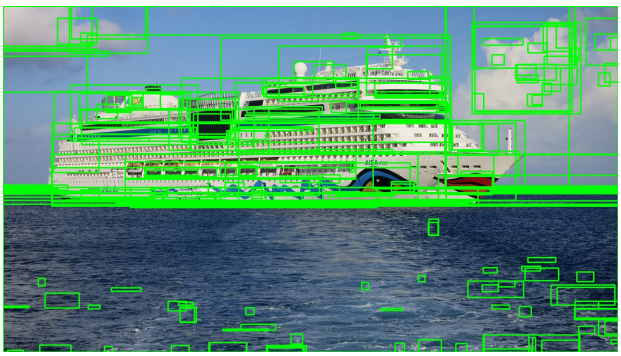


Figure 2: An example of Selective Search for boat detection after the application of a median and bilateral filter. For the sake of simplicity only 250 regions out of the many proposed are showed in the current image.

of 224x224. The output of the network is a score $\hat{y}$ that represents the probability that a certain region is a boat. This is similar to what is done in R-CNN [14], but instead to use the CNN for features extraction and then classify each region with a class-specific linear SVM, in the proposed approach the patch is directly classified using the CNN.

For training the CNN, binary cross-entropy has been used

$$J(h) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(\hat{y}_i) \tag{4}$$

where N is the number of samples, y_i is the true label for the i-th sample and $\hat{y}_i$ is the output given by hypothesis $h \in \mathcal{H}$, that is the hypotheses class. Moreover Adam [15] is the chosen stochastic gradient descent optimization algorithm.

After the classification process is then applied a Non Maximum Suppression (NMS) of boxes which is a technique whose goal is to select the box with the highest score and then remove all the overlapping boxes. In order to do so, two threshold T_c and T_s can also be defined, such that only scores greater than T_s are selected and only boxes with a IoU value grater than a certain confidence T_c are removed. T_s is defined in a dynamic way such that it has a starting value that decreases until at least a proposed box is found or until T_s becomes lesser than 0.7. With a quite high starting value the detector is able to discard a lot of false positive boxes, if there are none, then the process is

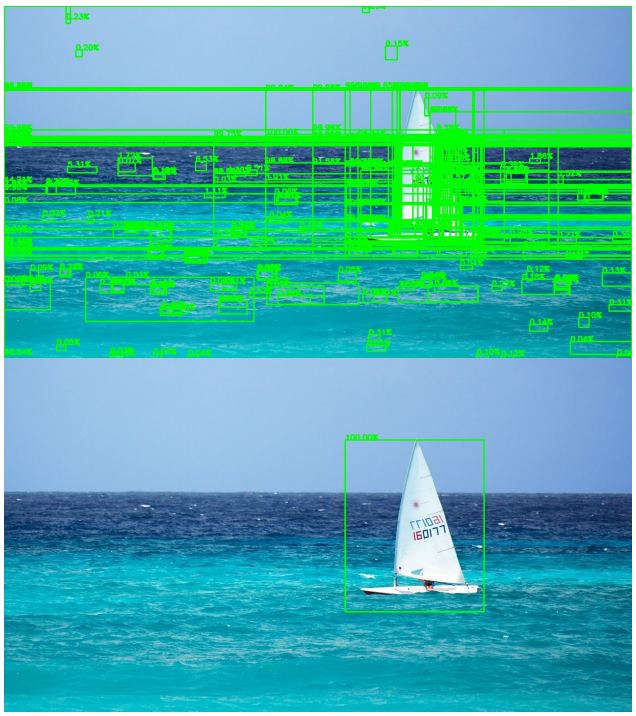


Figure 3: On the top there is an input image before NMS. After the application of NMS on the bottom there is the final result.

iterated at most until T_s is equal to 0.7, below this value the algorithm stops, so not to be stuck in a infinity loop and also for not considering scores too much low to be boats. Thus NMS is of great importance to reduce the number of false positive and in general decrease the number of bounding boxes for an object. An example is on Figure 3.

V. Experiments

All the experiments are carried out following the method described in the previous section. In specific Selective Search [1] is used in "fast mode", whose parameters can be seen in the official paper, while for the CNN three different possibilities has been explored. To chosen the best model from the ones considered, they have been trained with dataset as described in section III and in detail the test set is 33% out of the whole set, then of the remained samples 20% is used as validation set and at the end there is the training set. Thus the number of samples are respectively 12552, 3138 and 7728 for training, validation and test set. Furthermore checking the values of the validation loss, early-stopping technique has been implemented with patience equals 3 and the weights of the best configuration are used. Then the performances show that the best CNN configuration is:

- 2D convolutional layer, 32 filters, kernel size 3x3, activation function = Relu, padding = 'same';
- 2D Max-Pooling layer, kernel size 2x2;
- 2D convolutional layer, 16 filters, kernel size 3x3, activation function = Relu, padding = 'same';
- 2D Max-Pooling layer, kernel size 2x2;
- 2D convolutional layer, 4 filters, kernel size 3x3, activation function = Relu, padding = 'same';
- 2D Max-Pooling layer, kernel size 2x2;

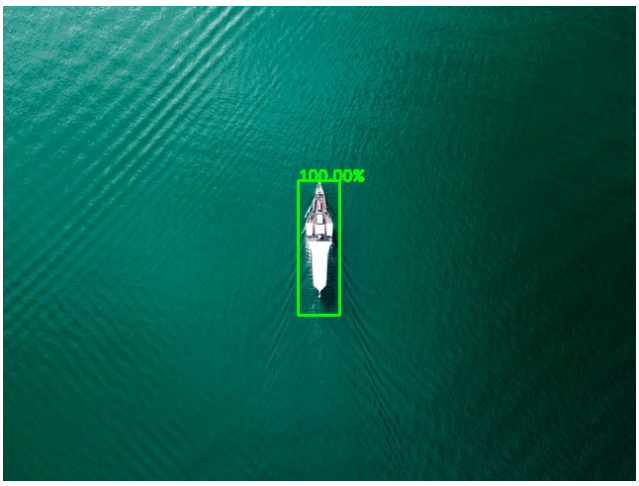


Figure 4: Boat detection of a single vessel in the middle of the sea. The boat is detected thanks a green bounding box with the score of class 1 (probability of the region to be a boat) on the top left corner.

- Flatten layer;
- Dense layer, activation function = 'Relu', 128 neurons;
- Dense layer, activation function = 'Relu', 64 neurons;
- Dense layer, activation function = 'Relu', 32 neurons;
- Dense layer, activation function = 'Sigmoid', 1 neuron;

Other settings used for all further evaluations are $T_r = 0.2$ and $T_c = 0.01$; T_s starts from the value 0.9975 and it decreases of 0.01 for each step until a least a box is found or $T_s < 0.7$. In addition for the filters applied before Selective Search, the median filter has 15x15 aperture and the bilateral filter has size equals 15, sigma color space $\sigma_c = 5000$ and sigma coordinate space $\sigma_s = 1500$.

For some images the detection results are quite good, especially if there is only one boat in the middle of the sea, like in Figure 4. Indeed in these cases ships have a well defined shape that can be clearly distinguished from regions of the waves or sky. In addition if the boat has a small or medium size compared with the whole image and not much other details such as small windows, writings and such, then the detection is easy. On the contrary if the boats are very big or if there are a lot details in the background, the detection task is really difficult. Several false positive (regions that are not boats are instead classified as ships) are identified and it becomes extremely difficult to distinguish from patches that are of bridges, windows and other architectural structures, from patches that are actually boats. Moreover it may also happen that for a ship, multiple boxes of different parts are identified, but not a single box with the whole vessel inside. Another case is when many boats are all located near each other and it becomes hard to distinguish the single boats, thus often one large box incorporates all of them.

Therefore according with everything just said, Kaggle dataset [8] is easier to compute because the background is

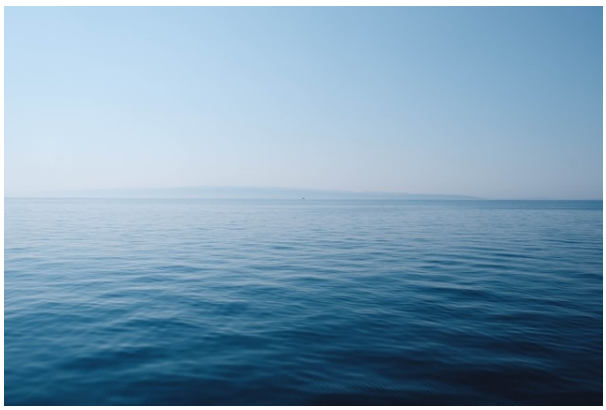


Figure 5: First test image with no boats.

Figure 6: Second test image with no boats. The possible boat is detected thanks a green bounding box with the score of class 1 (probability of the region to be a boat) on the top left corner.

mainly composed by sky and sea and often there is only one boat for image with only some exceptions of two/three ships for photo. On the other hand, MAR dataset [9] has photo of boats in Venice, where there are a lot of other elements than the ships, for example houses, windows, bridges, people, etc...; so here the detection is really difficult. As proof to these statements, in the following pages are showed the obtained results compared with the ground truth boxes. In detail the ground truth is showed with black boxes, instead the predicted boxes are showed in range of colors from green (the best performance) to yellow, orange and red (worst results) with written the corresponding percentage of IoU. Thus an almost perfect detection is for IoU $i >= 0.9$ and it will be green, for $i \in [0.7, 0.9)$ in yellow is a good detection, for $i \in [0.5, 0.7)$ in orange is a acceptable detection and at the end for $i < 0.5$ in red is a bad detection.
The results obtained can be seen in Table I and they prove what has been said about the MAR dataset to be more complex in background generating more false positive boxes.

Table I: It is shown the number of false positives which are the red boxes, and the number of detected boats which are orange, yellow and green boxes subdivided in ranges by the value of IoU.

	red	orange	yellow	green
Kaggle dataset [8]	23	**4**	**4**	**0**
MAR dataset [9]	107	**4**	**11**	**2**

It has also been tested the robustness of the proposed algorithm by giving in input images with no boats. For image with only sky and sea, such as in Figure 5, the approach correctly detects nothing. On the contrary when there are other elements such as rocks in Figure 6, then there are false detection.

VI. Conclusion

The boat detection task is very hard, due to the irregular sea surface that can generate trails and by the variety in boats appearance which can change from model to model and depending on the view point and/or illumination conditions. Also the boats are of different types, shapes and appear on many possible dimension w.r.t. the whole image size. Thus, even if some acceptable results are achieved, the proposed approach has surely a margin of improvement; first of all maybe using a more representative and bigger dataset for training the CNN classifier.

Note: the tools used are visual studio 2019 on a windows 10 PC and OpenCV 4.5.1 library. Almost all the code is in C++ with the exception of the CNN training which is done in Python, by a Jupyter Notebook and TensorFlow library.

References

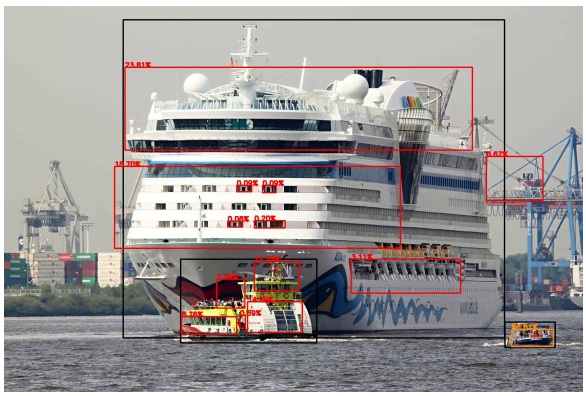
[1] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

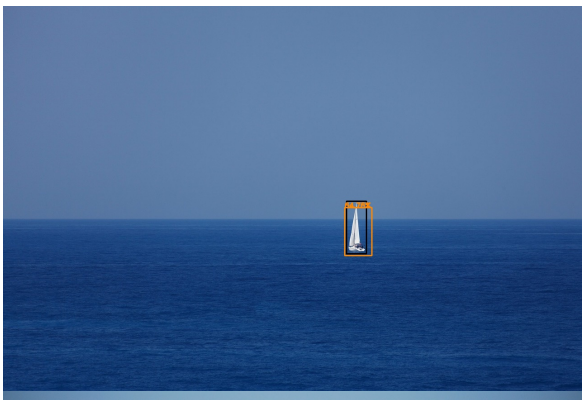
[2] Tatsuhiro Akiyama, Yosuke Kobayashi, Jay Kishigami, and Kenji Muto. Cnn-based boat detection model for alert system using surveillance video camera. In *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*, pages 669–670. IEEE, 2018.

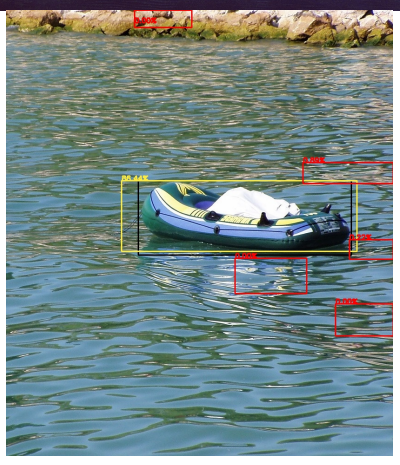
[3] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

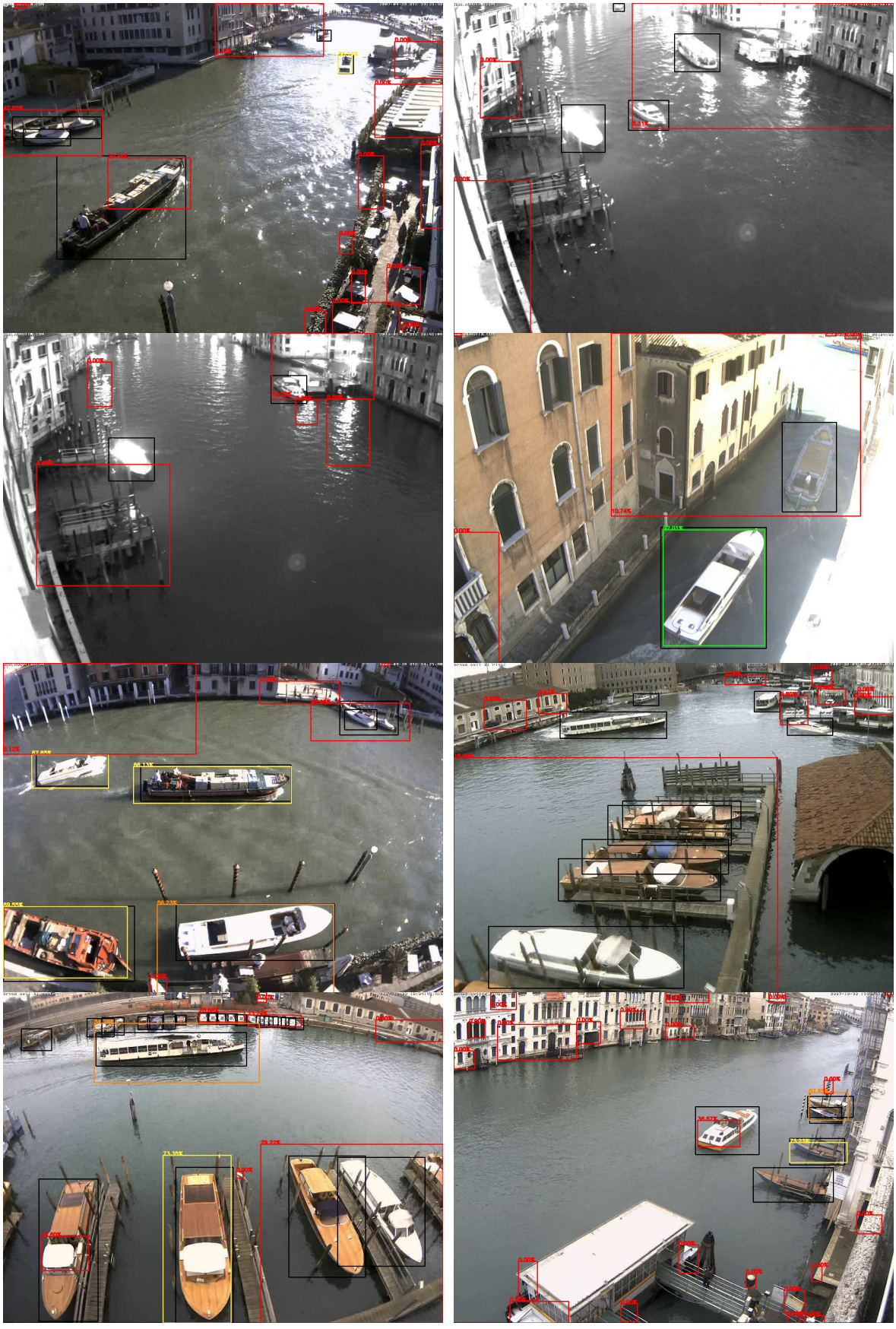
[4] Yang Liu, Miao-hui Zhang, Peng Xu, and Zheng-wei Guo. Sar ship detection using sea-land segmentation-based convolutional neural network. In *2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP)*, pages 1–4. IEEE, 2017.

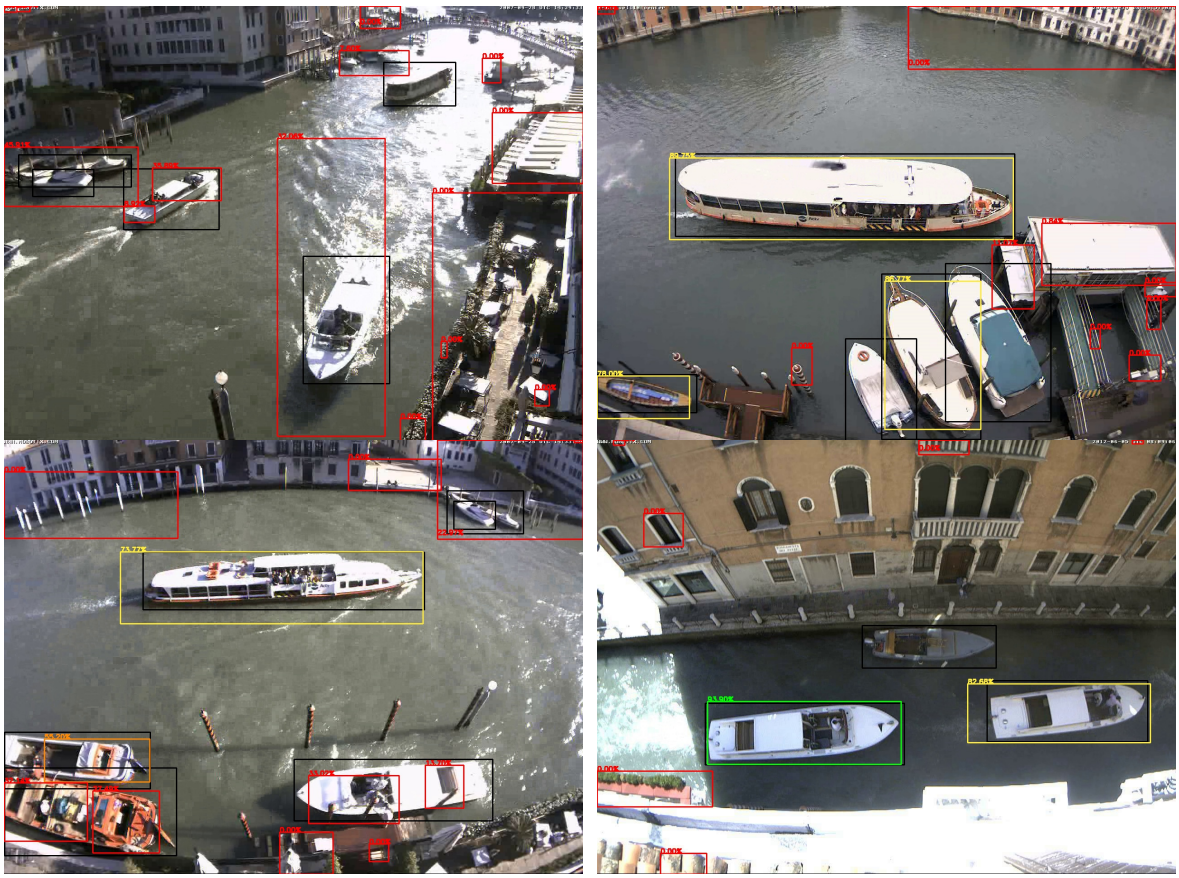
[5] S Iwin Thanakumar Joseph, J Sasikala, and D Sujitha Juliet. Detection of ship from satellite images using deep convolutional neural networks with improved median filter. In *Artificial Intelligence Techniques for Satellite Image Analysis*, pages 69–82. Springer, 2020.











[6] Domenico Bloisi, Luca Iocchi, Michele Fiorini, and Giovanni Graziano. Automatic maritime surveillance with visual target detection. In *Proc. of the International Defense and Homeland Security Simulation Workshop (DHSS)*, pages 141–145, 2011.

[7] *Machine Learning dataset for boat detection*, 2021. https://drive.google.com/file/d/1XkVfXNjq_KMANKUBSlbpPrlMNe9cMhKk/view?usp=sharing.

[8] *Kaggle dataset*, 2021. https://drive.google.com/file/d/1PToX_LH4JsjU2rSiD4vJQSV80cojxy5m/view?usp=sharing.

[9] *MAR dataset*, 2021. https://drive.google.com/file/d/1kCgOFIP7meuUDh49BYxGyTYTg-kJQwys/view?usp=sharing.

[10] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

[11] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014.

[12] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2189–2202, 2012.

[13] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3286–3293, 2014.

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.