COMPARING CNN-BASED OBJECT DETECTORS ON TWO NOVEL MARITIME DATASETS

Valentin Soloviev, Fahimeh Farahnakian, Luca Zelioli, Bogdan Iancu, Johan Lilius, Jukka Heikkonen

ABSTRACT

Vessel detection studies conducted on inshore and offshore maritime images are scarce, due to a limited availability of domain-specific datasets. We addressed this need collecting two datasets in the Finnish Archipelago. They consist of images of maritime vessels engaged in various operating scenarios, climatic conditions and lighting environments. Vessel instances were precisely annotated in both datasets. We evaluated the out-of-the-box performance of three state-of-the-art CNN-based object detection algorithms (Faster R-CNN [1], R-FCN [2] and SSD [3]) on these datasets and compared them in terms of accuracy and run-time. The algorithms were previously trained on the COCO dataset [4]. We explore their performance based on different feature extractors. Furthermore, we investigate the effect of the object size on the algorithm performance. For this purpose, we group all objects in each image into three categories (small, medium and large) according to the number of occupied pixels in the annotated bounding box. Experiments show that Faster R-CNN with ResNet101 as feature extractor outperforms the other algorithms.

Index Terms— Object detection, maritime vessel dataset, maritime vessel detection, convolutional neural network, deep learning, autonomous marine navigation

1. INTRODUCTION

Maritime vessel detection garnered a lot of attention as of late, given the latest developments in autonomous shipping and the interest that IMO (International Maritime Organization) showed towards it. Maritime traffic supervision and management are essential in regulating waterborne transportation safety, environmental impact, border crossing, sea and coastal security, etc. Maritime vessel detection becomes, in this context, a crucial concern for numerous applications. Due to a high variety of vessel types (sailboats, cargo ships, motorboats, etc), their detection can prove challenging. Moreover, environmental factors (glare, fog, clouds, high waves, lighting conditions etc) highly influence the accuracy of detection algorithms.

Convolutional neural networks (CNNs) made tremendous contributions to object detection overall in the past decade [2, 3]. CNNs combine features, represented at different levels of resolution, and classifiers in multilayers stacks. The depth of a stack (number of layers) promotes the selectivity and invariance of the feature in every layer. CNNs avail of the hierarchical representation of features in images. They employ convolutional layers to local features and progressively evolve an image representation from simple features to entire objects across several levels of resolution. Each layer in the hierarchy transforms the representation of the previous one, making it gradually more specific with every layer [5].

While maritime vessel detection from satellite imagery disposes of a fair number of datasets [6], inshore or offshore datasets are still largely unavailable [7]. For this reason, we investigate the performance of several prevalent object detectors on two waterborne marine datasets. The object detectors were previously trained on the COCO dataset [4]. In addition, we evaluate the detection accuracy of the detectors based on different object sizes and respective feature extractors.

The paper is organized as follows. Section 2 describes the most significant results in vessel detection. Section 3 describes two newly annotated datasets, collected in diverse environmental conditions and dynamic ranges. Section 4 illustrates the most prevalent deep neural networks that we employed in our analysis. Experimental results are presented in Section 5, where we investigate the effect of the feature extractor and object size on the detector performance. Conclusions are described in Section 6.

2. RELATED WORK

Generic object detection: While extensively studied for decades, object detection came to spotlight in recent years along with the advancements in deep learning. The groundbreaking results of Krizhevsky et al. [5] revealed a significant increase in the image classification accuracy on the ImageNet dataset, using exclusively supervised learning [5, 8]. Girshick et al. pushed the limits of this classification results investigating their applicability to object detection, which requires localization of objects in images. They address this challenge introducing proposal regions as an alternative to the conventional multi-scale scanning [9] and classifying them with linear SVMs [8].

Early promising results of CNNs for object detection compelled significant efforts towards improving their perfor-

mance. Consequently, two aspects were extensively investigated: CNNs' architecture and detection algorithm performance. To this end, several very deep networks emerged, i.e. VGGNet [10], ResNet [11], Inception [12], producing more accurate results than before. To enhance the performance of detection algorithms, a variety of region-based object detectors were proposed. They fall in two categories: two-stage detectors (R-CNN [8], Fast/Faster R-CNN [13, 14], R-FCN [2]) and one-stage detectors (SSD [3], YOLO [15]).

Maritime vessel detection: Beyond maritime vessel detection from spaceborne imagery [6], a few studies employed detection algorithms from waterborne images. Some of them focused on separating objects from the background [16], others employed the Histogram of Oriented Gradients (HOG) approach using sliding-windows [17].

CNNs were seldom exploited for seaborne vessel detection. Autonomous maritime navigation, however, can prompt the development of new datasets and applications. For instance, the *Singapore Maritime Dataset* is used in [18] for ship detection under a new proposed model, YOLO [15]. Another novel dataset *SeaShips* consisting of a collection of inshore and offshore ship images is introduced in [7]. Three object detectors (Faster R-CNN [1], SSD [3] and YOLO [15]) are utilized to identify maritime vessels and their performance is compared. In [19] maritime vessel images from a Vessel Tracking System (VST) are collected into a dataset, which represents authentic situations from traffic management operators. An SSD detector is used to identify vessels, taking into account intra-class variance.

3. DATASETS

Evaluating how general-purpose object detection algorithms perform on domain-specific data is most often hindered by data availability. Hence, we designed our own sensor platforms for data acquisition. We collected two real marine datasets, which provide us with the ability to evaluate the object detectors under a variety of factors: different light conditions, partial visibility due to image boundary truncation, object occlusion, etc. Our datasets consist of images from maritime scenarios under different illumination conditions with various marine vessels. This section describes the collected datasets.

3.1. Dataset1

To evaluate the object detectors, we selected 135 videos from a sightseeing watercraft operating between the cities of Turku and Ruissalo in South-West Finland, along the Aura river and into the Finnish Archipelago for a duration of 13 days, from June 26, 2018 to July 8, 2018, in the interval 10:15 - 16:45. Data showed typical scenarios the watercraft encountered on the specified route: the weather varied between sunny and cloudy, with a reduced number of instances of fog and rain. The environment includes urban landscapes, comprising urban constructions, vehicles, etc. The lighting environment is diverse given the timespan of the watercraft during one day. Moreover, we considered images with various degrees of occlusion and vessel proportion visibility among them. For data acquisition, a video camera with 65° field of view was mounted on the sightseeing watercraft. The data was stored in FullHD (1920x720) at 15 fps in MPEG format. To evaluate the models, 4800 photos were extracted from the videos. In order to avoid redundancy, 400 photos were chosen and precisely annotated. The total number of annotated vessels in this dataset is 850.

3.2. Dataset2

For the evaluation of the CNN-based detectors, we collected a second real maritime dataset in the Finnish Archipelago, which consists largely of maritime vessels in opensea landscapes. The dataset was recorded by two sensors continuously, providing data from various environmental and geographical scenarios. This sensor system included RGB (visible spectrum) and IR (thermal) camera arrays, providing output which was synchronized and stitched to form panoramic images. The individual visible cameras had FullHD resolution, while the thermal cameras had VGA resolution. Both camera types had a horizontal field of view of approximately 35 degrees. Images were sampled one frame per second in and stored in MPEG format. We manually annotated all vehicles (passenger vessel, motorboat, sailboat or docked vessel) within each IR/RGB sequence with a bounding box. The bounding box consisted of all pixels belonging to that object and at the same time be as tight as possible. This dataset consists of 1,750 images captured using visible cameras. The original size of all images is 3240×944 pixels for both scenarios. To reduce the computation time, we re-sized the original images into 1200×400 pixels. The number of vessel objects in the dataset is 9,137.

4. DEEP CONVOLUTIONAL NEURAL NETWORKS

R-CNN (Regions with CNN features) integrates region proposals within the CNNs. The framework comprises three modules: first, producing the region proposals, second, a deep convolutional neural network performing feature extraction on every proposed region, and third, a set of linear SVMs for region classification [8].

Yet highly acclaimed at the moment of their publication, the hindrance in the performance of R-CNNs originated from their intrinsic complexity, given their inability to share computation. Their immediate successors, SPPnets [20] brought about a performance increase promoting computation sharing. However, they bear a major weakness, i.e. reducing the accuracy in deep networks due to their design of convolutional layers (fixed) [13]. Detriments of these networks

Table 1: Average Precision (AP) (in %) of the proposed CNN-based detectors for *Dataset1* and *Dataset2* with different feature extractors and object sizes.

Method	Feature Extractor	Dataset1				Dataset2			
		AP_S	AP_M	AP_L	AP	AP_S	AP_M	AP_L	AP
SSD	MobileNet-v1	33.57	45.43	50.40	44.94	16.43	16.98	11.96	15.78
	MobileNet-v2	27.85	54.78	54.00	50.11	11.50	24.66	28.48	17.68
	Inception-v2	23.57	48.91	49.60	44.94	10.37	11.44	42.06	17.82
Faster R-CNN	NasNet	34.28	53.69	64.80	53.76	7.08	5.30	68.72	19.38
	Inception-v2	30.71	54.78	70.80	55.52	1.71	5.25	74.96	19.33
	Inception-resnet-v2	34.28	53.26	60.80	52.35	1.60	5.04	74.75	19.15
	ResNet101	31.42	55.65	74.00	**57.05**	1.91	5.69	75.39	**19.60**
	ResNet50	25.71	51.95	70.00	52.94	6.83	2.74	74.80	19.17
R-FCN	ResNet101	28.57	55.65	70.00	55.41	1.06	1.30	86.46	19.13

were overcome by a new architecture, Fast R-CNN, which generates a convolutational feature map straight from the input image through a series of convolutional and max pooling layers. Subsequently, a region of interest (*RoI*) pooling layer processes the feature map, resizing it to a fixed scale. The newly rescaled RoI is then reshaped into a fixed-size feature vector, which is subjected to a series of fully connected layers. This series of layers culminates in two output layers: a softmax classification layer, which determines the class of the object, and another that delivers the coordinates for each of the object classes through a bounding box regressor [13].

4.1. Faster R-CNN

The region-based detectors depicted above use *selective search* [21] to determine region proposals, which consumes precious computational time. The selective search section of the network spends roughly the same amount of time as object detection does. The newly proposed algorithm, Faster R-CNN, counters this inconvenience essentially omitting the selective search section, and passing the convolutional feature map to another network (e.g. Fast R-CNN) to produce the region proposals. Moreover, these region proposals are processed by two supplementary layers: one that constructs a feature vector used to classify the image within the bounds of the proposed region and a second one which predicts the bounding boxes through a regressor [14].

4.2. Region-based Fully Convolutional Networks (R-FCN)

Deep networks from the R-CNN family (Fast/Faster R-CNN) perform in two-stages, which essentially correspond to two separate subnetworks. First, an RPN (Region Proposal Network) generates region proposals aided by a RoI pooling layer. A second fully convolutional autonomous subnetwork classifies the images of the proposed regions and bounding boxes are determined through a bounding box regressor. R-FCN however employs a fully convolutional architecture, promoting computation sharing throughout it [2].

4.3. Single Shot multibox Detector (SSD)

Alike the RPN in the Faster R-CNN algorithm, SSD utilizes anchors (default boxes) for prediction. While the former employs anchors in feature pooling and subsequent image classification, SSD assigns scores for each anchor. Based on a feed-forward CNN, the SSD detector generates a limited number of anchors and their corresponding scores for each object class. The underlying MultiBox method applied on multi-scale feature maps enhances the likelihood of objects being detected with certain accuracy [3].

5. EXPERIMENTS

In this paper, we focus on prevalent one-stage (SSD) and two-stage (Faster R-CNN and R-FCN) object detectors. We explore the performance of these detectors on two real marine datasets. The proposed CNN-based detectors are previously trained on the Microsoft COCO object detection dataset on 91 object categories [4]. The training dataset includes 3,146 images of marine vessels. In this section, we discuss the effect of various feature extractors and object sizes on the detection accuracy and run-time of these detectors.

The effect of adjusting the feature extractor: We investigate the efficiency of different feature extractors in each detector. For this reason, we collected vessel detection results based on SSD with different feature extractors (MobileNet-v1 [22], MobileNet-v2 [23] and Inception-v2 [24]). In addition, we evaluated other detectors using three different feature extractors: NasNet [25], ResNet50 [26], ResNet101 and Inception-resnet-v2 [27]. Combining these feature extractors with the three proposed object detectors, a total of nine methods were investigated. The reader can find all information regarding the configuration of these methods in [28]. Table 1 shows the detection accuracy results in terms of Average Precision (AP) for every method in each dataset. The best results are highlighted in bold for each dataset.

In *Dataset1*, we obtained the highest AP (50.11%) using MobileNet-v2 between different SSD configurations. However, R-FCN achieved higher AP (55.41%) than SSD. Gener-

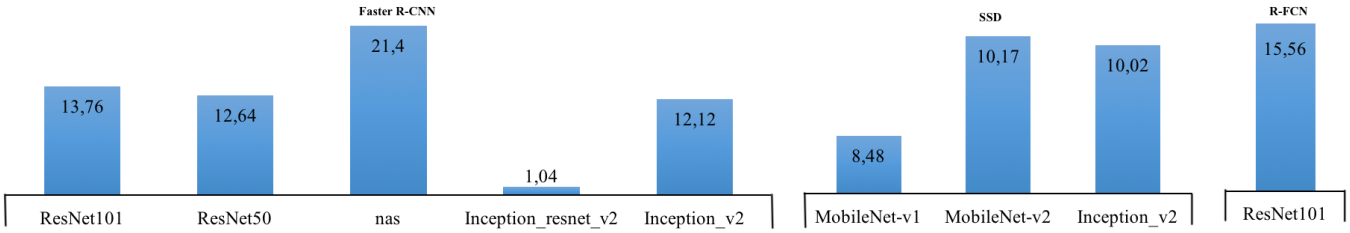

Fig. 1: The runtime (seconds) of the proposed CNN-based detectors for an example image

ally, Faster R-CNN with ResNet101 has the highest total AP for all objects (57.05%) compared to other methods.

In *Dataset2*, SSD with Inception-v2 attained the highest accuracy (17.82%) among the other configurations. R-FCN (19.13%) achieved higher AP than SSD in this dataset as well. Among all methods, Faster R-CNN with ResNet101 has the highest AP (19.60%).

The effect of object size: It has been observed by the authors that object size can impact the detection accuracy. To verify this, we divided all annotated objects in our dataset into three categories: small, medium and large. Then we explored the effectiveness of object size on detection accuracy. Our datasets comprise more small and medium objects than large objects. Specifically, 16% of object in *Dataset1* are small (area $< 32^2$), 54% are medium ($32^2 <$ area $< 100^2$), and 30% are large (area $> 100^2$). Area is measured as the number of pixels in each bounding box. In *Dataset2*, from the total objects, 22%, 57% and 21% are small, medium and large objects, respectively. AP_S, AP_M, AP_L and AP represent the AP for small, medium, large and all objects. From our experiments, we observe that the detection accuracy is remarkably decreased by reducing the object size. In general, we obtained higher AP from all methods in *Dataset1* as it consists of a higher count of bigger objects than *Dataset2*. The AP of the best-performing method (Faster R-CNN with ResNet101) is decreased almost 25% and 17% from large to small objects on *Dataset1* and *Dataset2*.

Faster R-CNN with ResNet101 can have high accuracy for medium (55.65%), large (74.00%) and all objects (57.05%) compared to other methods. Among different SSD configurations, we obtained the highest accuracy from MobileNet-v2 for detecting medium and large objects in *Dataset1* and *Dataset2*. However, SSD with MobileNet-v1 can get higher accuracy (33.57%) for small objects. In Faster R-CNN, ResNet101 can have high accuracy in *Dataset1* and *Dataset2*. Finally, R-FCN scored 55.41% and 19.13% accuracy in *Dataset1* and *Dataset2*, respectively. Among all configurations, *Faster R-CNN* with Resnet101 generates the best result for both datasets for all objects.

Run-time: We provide a fair comparison on the runtime of the proposed methods using the same hardware specification. All proposed detectors were tested on an example image from our dataset with 1200×400 pixels. The tested system specification is the Taito cluster, provided by CSC (The Finnish IT Center for Science). The cluster has the following configuration: 2x Xeon E5-2680 v4 CPUs with 14 cores each running at 2.4GHz with 16 G RAM. The GPU configuration is 4x NVIDIA Tesla P100 GPUs connected in pairs to each CPU. Fig. 1 shows that Faster R-CNN with Inception-resnet-v2 has the minimum running time (1.04 second). Faster R-CNN with Inception-resnet-v2 at 52.35% is 4.7% less accurate than Faster R-CNN with ResNet101 (best detector). However, it runs 13 times faster.

Qualitative results: Fig. 2 and Fig. 3 illustrate an example of detection results from three methods that scored the highest AP in each dataset. As we can see in Fig. 2 and Fig. 3 (c), there are more false positive samples in the results of SSD and R-FCN detection. For Faster R-CNN in Fig. 2 and Fig. 3(b), the results are satisfying, with the ll detected vessels regions registering high scores. We note that Faster R-CNN performed better at detection of vessel instances than others.

6. CONCLUSION

This paper evaluates the performance of three state-of-the-art object detection algorithms exploiting two domain-specific datasets collected in the Finnish Archipelago. We ensured the acquired datasets contain photos considering a series of factors: lighting conditions, object visibility due to boundary truncation, object occlusion etc. We performed an experimental comparison of three prevalent CNN-based object detectors (Faster R-CNN [1], R-FCN [2] and SSD [3]) on the aforementioned datasets. We assessed the performance (accuracy and run-time) of these objects detectors, with respect to the object size and feature extractor.

The experimental results show that Faster R-CNN with ResNet101 as feature extractor has the highest detection accuracy (74.0%) for large objects compared to other eight methods. However, its accuracy was significantly reduced when we considered smaller objects (less than 32×32 pixels) in our data.

For future work, an improved detection network structure will be investigated to address very small objects in our data. We plan to also employ transfer learning and evaluate the performance of these detectors further on domain-specific data.

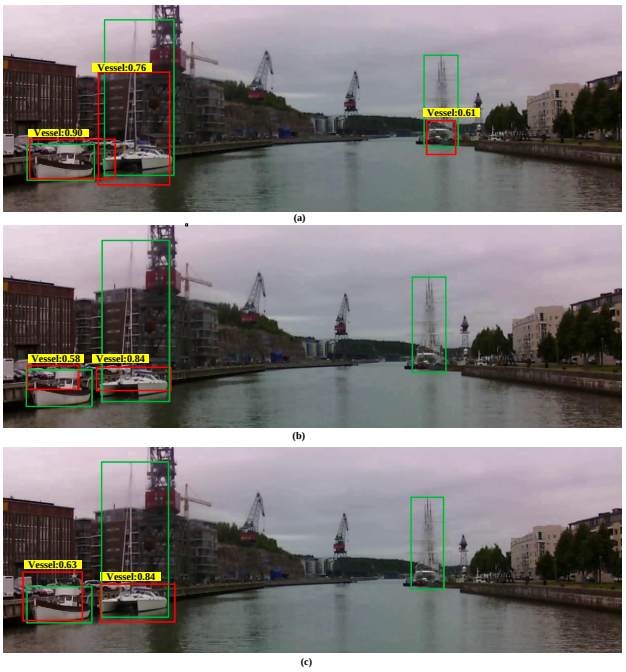

Fig. 2: Qualitative detection results for *Dataset1* on (a) SSD, (b) Faster R-CNN and (c) R-FCN. The ground truth bounding boxes are shown as green rectangles. Predicted boxes by these methods are depicted as red bounding boxes. Each output box is associated with a category label and a score value in [0, 1].

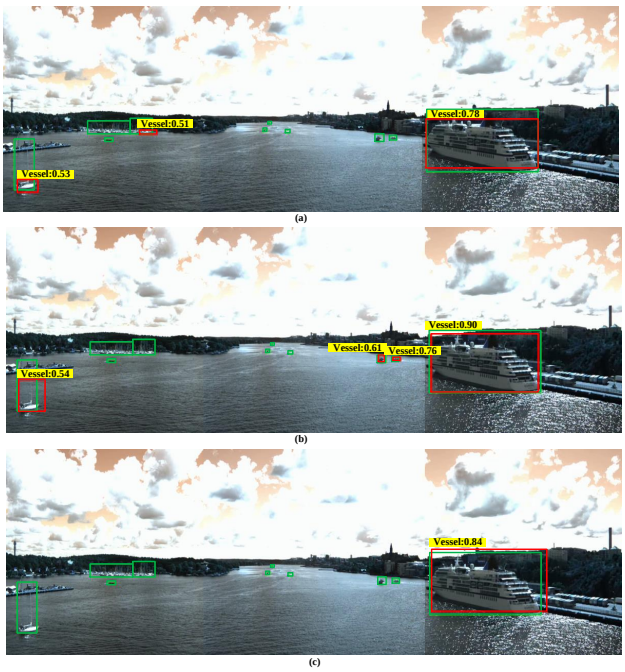

Fig. 3: Qualitative detection results for *Dataset2* on (a) SSD, (b) Faster R-CNN and (c) R-FCN. The ground truth bounding boxes are shown as green rectangles. Predicted boxes by these methods are depicted as red bounding boxes. Each output box is associated with a category label and a score value in [0, 1].

Moreover, we will extend our fusion framework by using data from lidar and radar besides RGB and IR cameras to improve the detection results.

We plan to improve the detectors to distinguish among different types of vessels encountered in our datasets. Moreover, we will employ vessel tracking algorithms on the original videos to study the impact on navigational safety.

7. ACKNOWLEDGMENTS

This work was partially supported by the Academy of Finland under project "Efficient Stream Computing by Fitting Computations to Cores" and by Business Finland under project "New 3D Analytics Methods for Autonomous Ships and Machines".

8. REFERENCES

[1] R. Girshick, J Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2014, CVPR '14, pp. 580–587, IEEE Computer Society.

[2] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A.C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Lawrence Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham, 2014, pp. 740–755, Springer International Publishing.

[5] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[6] U. Kanjir, H. Greidanus, and K. Oštir, "Vessel detection and classification from spaceborne optical images: A literature survey," *Remote sensing of environment*, vol. 207, pp. 1–26, 2018.

[7] Z. Shao, W. Wu, Z. Wang, W. Du, and C. Li, "Seaships: A large-scale precisely annotated dataset for ship detection," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2593–2604, 2018.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[9] C. Gu, J. J. Lim, P. Arbeláez, and J. Malik, "Recognition using regions," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1030–1037.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[13] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[15] Jo. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[16] N. Arshad, K. S. Moon, and J. N. Kim, "Multiple ship detection and tracking using background registration and morphological operations," in *Signal Processing and Multimedia*, pp. 121–126. Springer, 2010.

[17] R.G.J Wijnhoven, K. van Rens, E.G.T. Jaspers, and P. H. N. de With, "Online learning for ship detection in maritime surveillance," in *Proc. of 31th Symposium on Information Theory in the Benelux*, 2010, pp. 73–80.

[18] S.-J. Lee, M.-I. R., H.-W. Lee, J. S. Ha, I. G. Woo, et al., "Image-based ship detection and classification for unmanned surface vehicle using real-time object detection neural networks," in *The 28th International Ocean and Polar Engineering Conference*. International Society of Offshore and Polar Engineers, 2018.

[19] M.H. Zwemer, R.G.J. Wijnhoven, and P.H.N. de With, "Ship detection in harbour surveillance based on large-scale data and cnns.," in *VISIGRAPP (5: VISAPP)*, 2018, pp. 153–160.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[21] J.R.R. Uijlings, K.E.A. Van De Sande, T. Gevers, and A.W.M. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[22] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017.

[23] M. Sandler, A.G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2818–2826.

[25] B. Zoph, V. Vasudevan, J. Shlens, and Q.V. Le, "Learning transferable architectures for scalable image recognition," *CoRR*, vol. abs/1707.07012, 2017.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

[27] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. 2017, AAAI'17, pp. 4278–4284, AAAI Press.

[28] "TensorFlow Object Detection API," `https://github.com/tensorflow/models/tree/master/research/object_detection/samples/configs/`, 2017.