



HAL
open science

Marine Object Detection Based on Top-View Scenes Using Deep Learning on Edge Devices

J. Sharafaldeen, M. Rizk, D. Heller, Amer Baghdadi, J -Ph. Diguët

► **To cite this version:**

J. Sharafaldeen, M. Rizk, D. Heller, Amer Baghdadi, J -Ph. Diguët. Marine Object Detection Based on Top-View Scenes Using Deep Learning on Edge Devices. IC2SPM 2022: International Conference on Smart Systems and Power Management, Nov 2022, Beirut, Lebanon. pp.35-40, 10.1109/IC2SPM56638.2022.9988928 . hal-04198386

HAL Id: hal-04198386

<https://imt-atlantique.hal.science/hal-04198386>

Submitted on 7 Sep 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Marine Object Detection Based on Top-View Scenes Using Deep Learning on Edge Devices

J. Sharafaldeen[†], M. Rizk^{†‡§}, D. Heller^{*}, A. Baghdadi[‡] and J-Ph. Diguët[◇]

[†]Lebanese University, Faculty of Sciences, Lebanon

[§] Lebanese International University, CCE Department, Lebanon

[‡]IMT Atlantique, Lab-STICC UMR CNRS 6285, France

^{*} Université de Bretagne-Sud, Lab-STICC UMR CNRS 6285, Lorient, France

[◇] CNRS, IRL CROSSING, Adelaide, Australia

Abstract—Marine object detection and tracking is an important application for several disciplines such as sea surface monitoring, marine area management, ship collision avoidance, search and rescue missions, etc. Top-view scenes based on aerial or satellite imaging offer capturing objects from new angles of view or for locations that are not seen by capturing nodes fixed at the port side or mounted on moving boats. Moreover, artificial intelligence techniques based on deep learning provide robust solutions for classification and detection. Convolutional neural network (CNN) architectures are being used to detect multiple objects in images and videos. The achieved performance proves the relevance of CNNs in circumventing existing computer vision challenges. In this paper, we investigate the state-of-the-art CNN-based technique, so called You only look once (YOLO), to detect marine objects in images showing sea ships and humans from top-view. YOLO available models are trained using our collected dataset. The evaluation of the trained models illustrates the effectiveness of YOLO in detecting targeted classes (humans and sea ships) with high precision (90%). The deployment of the trained model on embedded edge devices achieves a high inference performance beyond 80 frames per second.

Index Terms—Marine, Object Detection, Neural Network, YOLO

I. INTRODUCTION

Nowadays, marine surveillance is a vital task. It is essential for many applications such as sea surface monitoring, marine area management, ship collision avoidance, search and rescue missions, counter illegal migration, fishing and smuggling, etc. Computer vision techniques have been adopted to assist marine surveillance. Several topics have been addressed in the field of maritime image processing. However, the detection and classification of marine objects remain open. Maritime environment arises additional challenges due to the dynamic nature of background, continuous tides and waves lead, presence of small floating objects and unpredictable movement of marine objects [1]. In addition, the illumination is related to weather conditions (haze, fog, rain, bright sunlight, twilight, etc.) and to the incident angle of solar rays on the water surface. Color ranges of seawater vary according to time (night, day, sunrise, sunset, etc.) and according to weather conditions [2].

The advances in artificial intelligence (AI) introduce robust solutions for detecting and classifying objects. Emergent techniques based on deep neural networks have tackled effectively computer vision tasks that were previously unsolvable. In

particular, convolutional neural networks (CNNs) have shown remarkable efficiency in detecting and classifying objects in images and videos. However, these networks impose additional requirements in terms of computational resources and memory needs. Real-life applications demand the requirement of real-time detection. This imposes additional challenges when implementing CNN-based techniques on embedded edge devices with limited computational resources and power budget.

Available methods to adapt CNNs on edge devices impact the performance in terms of detection speed, accuracy, and precision. Incorporating multiple views improves detection accuracy and precision. Providing diverse perspectives on the same scene can be utilized to increase detection performance, resulting in a reduction in the number of missed or incorrect detection. Using numerous views for detection, on the other hand, mitigates the effects of object occlusion, illumination, intra-class variation, the similarity between classes, background clutter, ambiguity, etc.

In this work, we aim to assist marine object surveillance using embedded artificial intelligence by exploiting the features of top-view scenes captured by satellites or drones. Fig. 1 depicts the overview of the proposed approach to assist detection using top-view scenes. Our work focuses on investigating and examining the recently introduced CNN-based object detection approach, so called You Only Look Once (YOLO), in detecting marine objects using images providing top-view scenes. YOLO has been initially introduced in [3] as an efficient unified model of all phases of a CNN for doing object



Fig. 1. Overview of the proposed approach

detection in real-time. Several versions of YOLO exist, with different network architectures and specifications. In this work we target YOLOv4 and YOLOv4 tiny models introduced in 2020 [4]. Compared to previous versions of YOLO, YOLOv4 model is optimized by using CSPDarknet-53, which adopts residual connection, as backbone architecture. YOLOv4 tiny model is a compressed version of YOLOv4 and has only two detection heads rather than 3 in YOLOv4 model. New versions of YOLO have been recently introduced in [5] and [6]. The evaluation of these versions is under the scope and the achieved results will be introduced in future publications. The main contributions of this paper are:

- collect a novel dataset showing marine objects from top-view, which has the widest number of annotations compared to available datasets,
- train and evaluate YOLOv4 and YOLOv4 tiny neural networks,
- deploy the trained networks on embedded edge devices and evaluate the detection performance and inference speed using real-life video sequences.

The rest of the paper is organized as follows. Section II presents a brief background and reviews related work. Section III describes the adopted methods for training and evaluation. Section IV describes the deployment of models on edge devices and presents the obtained results. Finally, Section V concludes the paper.

II. BACKGROUND AND RELATED WORK

A. Object Detection using Deep Learning

Currently, approaches based on deep learning are a key trend for computer vision applications. CNN-based methods provide robust solutions for object detection and classification that outperform traditional computer vision techniques [7], [8] and circumvent challenges. Several methods have been introduced in the literature. Two-stage methods such as region-based CNNs [9]–[11] have been considered pioneer approaches for object detection. However, these approaches can not ensure real-time inference when deployed on embedded devices as they compromise two basic steps: identification of regions from the input image and then performing forward propagation on each region to extract features. One-stage approaches such as single-shot detector (SSD) [12] and YOLO [3] reduce the complexity of object detection. A fully convolutional approach is adopted in which the network is able to find all objects within an image in one pass. Mainly a grid is used to divide the input image into regions instead of identifying them while execution.

YOLO approach has been introduced as a unified CNN model for object detection. YOLO networks have evolved since introduced in [3]. Several models with different architecture specifications have been proposed in [13], [14]. Recent proposed YOLO models can outperform other superior neural network models such as *EfficientDet* [15] and *RetinaNet* [16] provided by Google and Facebook respectively in terms of detection performance and inference speed as well [4]–[6].

B. Datasets for Top-View Marine Object Detection

Good efforts have been done to create datasets using satellite and aerial images. Seagull dataset [17] provides a set of video sequences captured by a fixed wing UAV flying above the sea in several regions. Different types of cameras are used to capture the videos covering the visible, near infrared, and infra-red ranges. The dataset includes annotations for several marine objects such as cargo ships, small boats, and life rafts. Maritime SATellite Imagery (MASATI) dataset [18] contains satellite images in the visible spectrum obtained from Microsoft Bing maps. The Airbus Ship Challenge reveals a large dataset of satellite images of sea ships [19]. Other datasets focus on detecting humans in open water for applications of detecting man overboard (MOB) cases and search and rescue missions (SAR) such as MOBDrone dataset [20], MaritimeSAR dataset [21] and SeaDroneSee dataset [22].

C. Top-View Marine Object Detection using Deep Learning

Most of the available works on marine object detection using deep learning address the detection of ships using images or videos from the side and front/back views [23] [2]. Only few works have targeted the detection of marine objects from top-view. In [24] provides a detection approach based on a feature pyramid network (FPN) using synthetic aperture radar images. The authors in [25] have adopted YOLOv3 model to distinguish between icebergs and ships in synthetic aperture radar images. In [26], the authors examine YOLOv3, YOLOv4, and YOLOv5 models to detect ships in satellite images using the dataset in [19]. In [20], different detection methods have been applied to examine the collected dataset. In [21], YOLOv4 models are adopted to detect humans in open water. Then, the authors optimize in [27] the trained models using pruning and quantization methods towards efficient deployment on edge devices.

III. METHOD

A. Collected Dataset

A huge dataset is collected of images showing marine objects from the top view. In this work, we focus on two classes: (1) sea ships and boats and (2) humans. The collected images show other marine objects such as jet skis and floating buoys. The collected images have been captured either from satellites or by drones. We make use of available published datasets [17], [18], [20]–[22], [28]–[30]. The collected images are all processed. Images without annotations are labeled manually. Available labels are converted to YOLO format and enhanced by adjusting the bounding boxes to meet the exact dimensions of the target marine objects. Fig. 2 shows some sample images from the final dataset.

The collected dataset includes images that are captured using imaging cameras with either RGB visible, infrared (IR), or near infrared (NIR) sensors. The images show marine objects from different perspectives and scales. The images differ in terms of altitudes, camera shooting angles, and illumination. The dataset includes negative samples that do not include labeled objects. These samples are used to ensure that



Fig. 2. Sample of images from the dataset

other objects in the images are not miss classified as the target classes (boats and humans). Note that the collected dataset is split into 90% for training and 10% for validation. Table I presents the specifications of the created dataset. It shows the number of images and the distribution of annotations per class.

TABLE I
SPECIFICATIONS OF THE CREATED DATASET

Dataset	Training	Validation	Total
# images	146,233	16,246	162,479
# negative samples	5,464	629	6,093
# Boats Annotations	110,410	10,808	121,218
# Person Annotations	165,584	13,105	165,584

B. Training and Validation

In our work, we targeted YOLOv4 and YOLOv4 tiny models. The main difference between both networks is that the number of convolutional layers in the CSP backbone is compressed [31]. Fig. 3 shows the block diagram of YOLOv4 network. Table II shows the specifications of the targeted models. Target models are trained using Darknet framework

TABLE II
SPECIFICATIONS OF THE TARGETED YOLOV4 MODELS

Model	Number of Layers	Activation function	Model Weights' Volume (MB)
YOLOv4	162	Mish	256.2
YOLOv4 Tiny	38	Mish	23.5

[32] on NVIDIA GeForce RTX 3080 graphics card with 8704 CUDA cores and 10 GB memory. Transfer learning is adopted to adapt pre-trained YOLO models towards the target classes (Boats and Humans) using the created dataset. Transfer learning is a machine learning technique where knowledge gained during training a set of problems can be used to solve other similar problems [33]. In our case, transfer learning is used to maintain the generalization gained by the pre-trained networks using COCO dataset [34]. We make use of the weights generated in previous training processes of networks with similar architecture specifications targeting COCO dataset. The weights of the feature extraction layers are reserved; whereas, the weights of the classification and detection layers are eliminated. During the new training, all network layers are unfrozen. Hence, all layers of the networks learn to turn the old feature into predictions on the new dataset. The structure of the YOLOv4 models is not modified. Only the number of filters in the last convolutional layer before the detector layers is adjusted to meet with the desired number of classes.

The batch size is set to 64 images. In Darknet framework, an iteration is defined as the number of batches to complete one epoch. In our case, the training process is conducted for 200,000 iterations, which is approximately equivalent to 79 epochs. The initial learning rate is set to 0.001 in the first 160,000 iterations where it is decreased to 0.0001 and then scaled down by 0.1 at iteration 180,000.

The input images size is adjusted to 416×416 . All images are down-scaled to meet the network size during training. The number of channels is set to 3 corresponding to the three RGB channels. Different data augmentation methods are applied while training. Saturation, exposure, and hue are selected to be randomly changed during training. Mosaic is also applied by merging 4 images into one. CutMix, a data augmentation strategy that removes remove pixels from an image and replace the removed regions with a patch from another image, is used for the classification only.

During the training process, the trained model is validated using the validation dataset. Starting from the 1000 iteration, the mean average precision (mAP) is calculated for every 4 epochs. Fig. 4 illustrates the training and validation performances of YOLOv4 model. Note that the blue curve corresponds to the training loss; whereas, the red curve corresponds to the computed mAP values. The mAP value is computed based on the metrics specified in the MS COCO competition which recommends the following expressions to calculate the values of Precision P , Recall R and Average Precision AP :

$$P = \frac{TP}{(TP + FP)} \quad R = \frac{TP}{(TP + FN)} \quad AP = \frac{1}{11} \sum_{R_i} P(R_i)$$

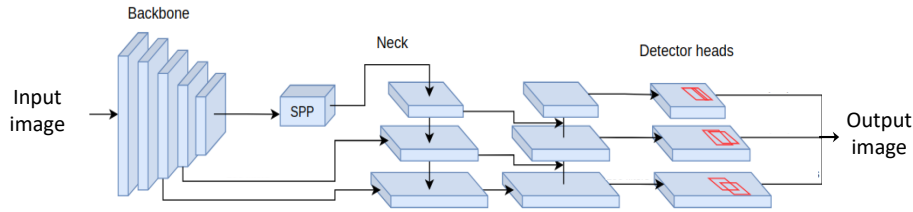


Fig. 3. Block diagram of YOLOv4 network

TABLE III
TRAINING DATA WITH RESPECT TO THE CLASSES

Model	Class	AP	Precision	Recall	F1-score	average IOU	mAP@0.5
YOLOv4	0: Boat	89.62%	0.9	0.87	0.89	72.28%	0.8911
	1: Human	88.61%					
YOLOv4 tiny	0: Boat	87.3%	0.84	0.87	0.86	66.84%	0.8742
	1: Human	87.27%					

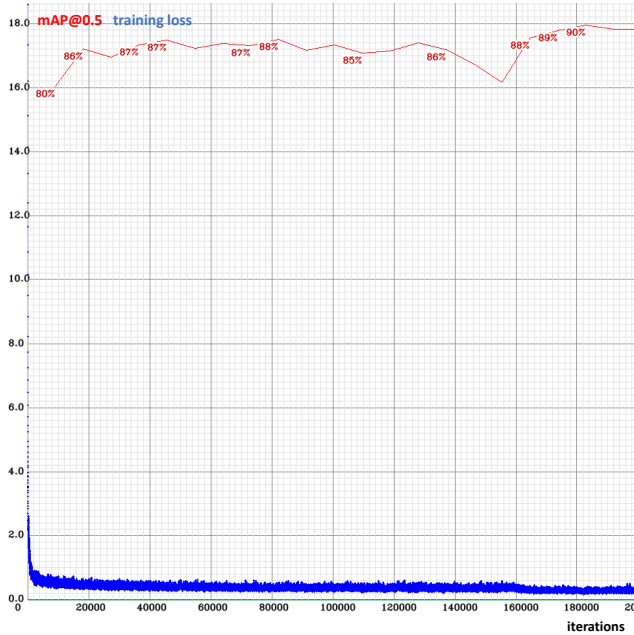


Fig. 4. Performance of the trained models when evaluated using the validation dataset

where TP , FP and FN stand for True Positive, False Positive and False Negative respectively. Note that the AP summarises the shape of the precision/recall curve, and is defined as the mean precision at a set of eleven equally spaced recall levels $\{0,0.1,0.2,\dots,0.9,1\}$. Table III presents the obtained performance metrics of the trained models when evaluated using the validation dataset. High mAP values of 89.11% and 87.42% are attained for YOLOv4 and YOLOv4 tiny models respectively. The obtained results demonstrate the effectiveness of trained YOLOv4 and YOLOv4 tiny models in detecting the boats and humans in aerial images.

C. Evaluation

The trained models are tested using a dataset that includes images that are not seen before. The test dataset includes 1114 images from [19] and [17] with 1672 annotations. Fig. 5

TABLE IV
SPECIFICATIONS OF TARGET EDGE EMBEDDED DEVICES

Target Device	Nvidia Jetson Nano	Nvidia Jetson Xavier NX
AI Performance	472 GFLOPs	21 TOPs
GPU	128-core NVIDIA Maxwell GPU	384-core NVIDIA Volta GPU with 48 Tensor cores
GPU Max Frequency	921 MHz	1100 MHz
CPU	Quad-core ARM Cortex A57 MPCore processor	6-core NVIDIA Carmel ARM v8.2 64-bit CPU
CPU Max Frequency	1.43 GHz	1.9 GHz
DL Accelerator	-	2× NVDLA Engines
Vision Accelerator	-	7-Way VLIW Vision Processor
Memory	4 GB 64-bits LPDDR4 25.6 GB/s	8 GB 128-bit LPDDR4x @ 51.2GB/s
Power consumption	5 -10 W	10 - 20 W
Mechanical	69.6 mm × 45 mm	69.6 mm × 45 mm

presents sample detection results. The figure shows that trained models are able to accurately detect and classify the presence of marine objects with a high confidence ratio in different maritime environments.

IV. DEPLOYMENT ON EMBEDDED EDGE DEVICES

The trained models are deployed on embedded edge devices to evaluate the inference speed. In particular, we use Jetson Nano and Jetson Xavier NX development kits. These kits are designed to ensure the powerful computation required to run neural networks for applications like image classification, object detection, segmentation, etc. In addition, these kits are characterized by their small size and low weight. This makes them ideal choices for airborne applications that require real-time inference on the edge. Table IV presents the technical specifications of both kits targeted in this work.

Table V presents the obtained inference speeds of the trained models in frames per second (FPS) when processing real captured videos on Jetson Nano and Jetson Xavier NX while operating on different power modes. The processed videos are

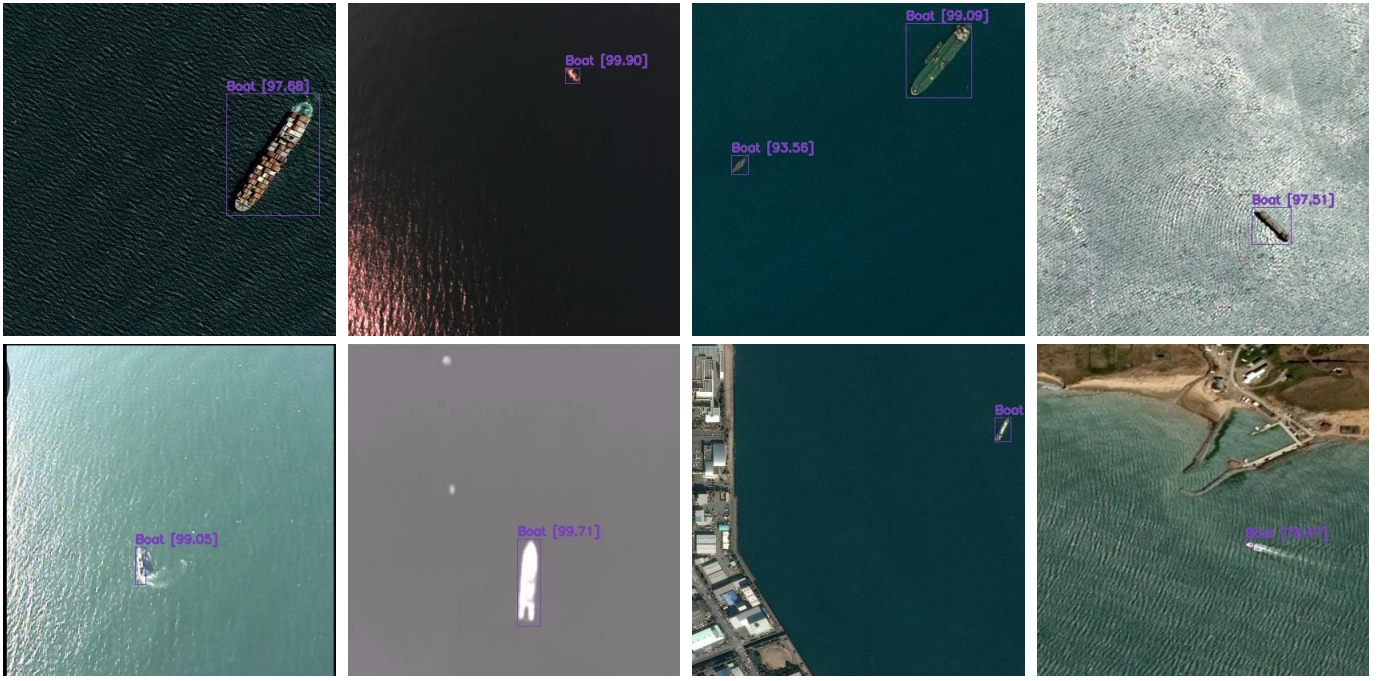


Fig. 5. Sample detection results in testing dataset images

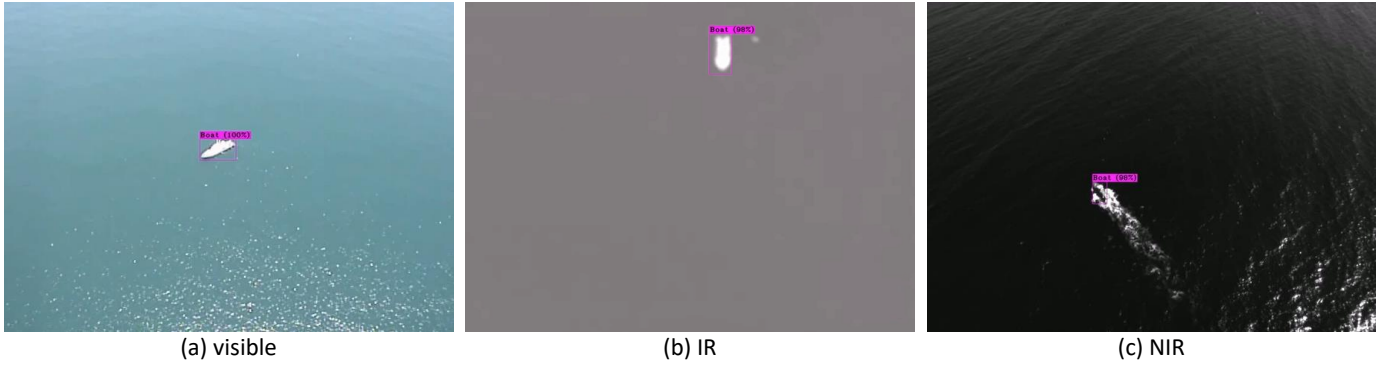


Fig. 6. Samples of the obtained detection results in video sequences

TABLE V
AVERAGE DETECTION PERFORMANCE IN FPS

Trained Model	Input Video Type	Nvidia RTX 3080	Nvidia Jetson Nano		Nvidia Jetson Xavier NX					
			Mode0	Mode1	Mode0	Mode1	Mode2	Mode3	Mode4	Mode5
YOLOv4	Visible	86.2	2	1.5	9.6	10.6	10.6	8.6	9.3	6.6
	IR	108.7	2	1.5	8.6	10.7	10.5	9.1	9.3	6.6
	NIR	101.5	2	1.5	9.8	10.7	10.7	8.6	9.3	6.6
YOLOv4 tiny	Visible	295.4	18.5	12.6	69.5	72.3	80.1	62.4	66.0	53.5
	IR	307.4	18.6	11.8	63.5	78.5	82.0	55.4	70.8	54.4
	NIR	307.3	18.6	12.5	73.2	71.5	81.3	63.3	66.4	53.5

captured during real drone flights using cameras with different types of image sensors (RGB, IR, and NIR) [17]. The table shows that YOLOv4 can achieve 2 FPS and 10.7 FPS when executed on Jetson Nano and Jetson Xavier NX respectively;

whereas, YOLOv4 tiny model can achieve 18.6 FPS and 82 FPS on Jetson Nano and Jetson Xavier NX respectively. Fig. 6 presents three samples of the obtained detection results in the processed video sequences.

V. CONCLUSION

This paper presents the usage of deep learning approach to detect marine objects using images providing top-view scenes. Available YOLOv4 networks are investigated in terms of detection performance and inference speed while deployed on edge devices. The paper introduces our novel collected dataset that includes the hugest number of images showing floating humans and boats from top-view. The adopted method to train, validate and evaluate the target models is presented. The obtained results show that YOLOv4 models can achieve a precision of 90% and a mAP value of 0.89. The trained models are deployed on small-size low-weight edge devices from Nvidia. The obtained inference speed demonstrates the ability to attain real-time detection. Future work will focus on applying optimization techniques such as pruning and quantization and investigate their impact on detection performance and inference speed.

REFERENCES

- [1] D. K. Prasad *et al.*, “Challenges in video based object detection in maritime scenario using computer vision,” *Inter. Journal of Computer and Information Engineering*, vol. 11, no. 1, pp. 31 – 36, 2017.
- [2] D. Heller, M. Rizk, R. Douguet, A. Baghdadi, and J.-P. Diguët, “Marine objects detection using deep learning on embedded edge devices,” in *Proc. of the IEEE International Workshop on Rapid System Prototyping (RSP)*, 2022, (in press).
- [3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, real-time object detection,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
- [4] A. Bochkovskiy, C. Wang, and H. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” *CoRR*, vol. abs/2004.10934, 2020. [Online]. Available: <https://arxiv.org/abs/2004.10934>
- [5] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie *et al.*, “YOLOv6: A single-stage object detection framework for industrial applications,” *arXiv preprint arXiv:2209.02976*, 2022.
- [6] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” *arXiv preprint arXiv:2207.02696*, 2022.
- [7] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.
- [8] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” *Proc. of the IEEE Computer society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893, 2005.
- [9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [10] R. Girshick, “Fast R-CNN,” in *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.
- [11] 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*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.
- [12] 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*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.
- [13] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [14] —, “YOLOv3: An incremental improvement,” 2018.
- [15] M. Tan, R. Pang, and Q. V. Le, “Efficientdet: Scalable and efficient object detection,” in *Proc. of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 781–10 790.
- [16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proc. of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [17] R. Ribeiro, G. Cruz, J. Matos, and A. Bernardino, “A dataset for airborne maritime surveillance environments,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 9, pp. 2720–2732, 2017.
- [18] A.-J. Gallego, A. Pertusa, and P. Gil, “Automatic ship classification from optical aerial images with convolutional neural networks,” *Remote Sensing*, vol. 10, no. 4, 2018.
- [19] “Dataset for Airbus ship detection challenge,” 2018. [Online]. Available: <https://www.kaggle.com/c/airbus-ship-detection/>
- [20] D. Cafarelli, L. Ciampi, L. Vadicamo, C. Gennaro, A. Berton, M. Paterni, C. Benvenuti, M. Passera, and F. Falchi, “MOBDrone: A drone video dataset for man overboard rescue,” in *Proc. of the International Conference on Image Analysis and Processing (ICIAP)*, S. Sclaroff, C. Distanto, M. Leo, G. M. Farinella, and F. Tombari, Eds. Cham: Springer International Publishing, 2022, pp. 633–644.
- [21] M. Rizk, F. Slim, A. Baghdadi, and J.-P. Diguët, “Towards real-time human detection in maritime environment using embedded deep learning,” in *Advances in System-Integrated Intelligence*, M. Valle, D. Lehmkus, C. Gianoglio, E. Ragusa, L. Seminara, S. Bosse, A. Ibrahim, and K.-D. Thoben, Eds. Cham: Springer International Publishing, 2023, pp. 583–593.
- [22] L. A. Varga, B. Kiefer, M. Messmer, and A. Zell, “Seadronesee: A maritime benchmark for detecting humans in open water,” in *Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2260–2270.
- [23] D. Qiao, G. Liu, T. Lv, W. Li, and J. Zhang, “Marine vision-based situational awareness using discriminative deep learning: A survey,” *Journal of Marine Science and Engineering*, vol. 9, no. 4, 2021.
- [24] P. Chen, Y. Li, H. Zhou, B. Liu, and P. Liu, “Detection of small ship objects using anchor boxes cluster and feature pyramid network model for SAR imagery,” *Journal of Marine Science and Engineering*, vol. 8, no. 2, p. 112, 2020.
- [25] F. S. Hass and J. Jokar Arsanjani, “Deep learning for detecting and classifying ocean objects: application of YOLOv3 for iceberg–ship discrimination,” *ISPRS International Journal of Geo-Information*, vol. 9, no. 12, p. 758, 2020.
- [26] K. Patel, C. Bhatt, and P. L. Mazzeo, “Deep learning-based automatic detection of ships: An experimental study using satellite images,” *Journal of Imaging*, vol. 8, no. 7, 2022.
- [27] M. Rizk, D. Heller, R. Douguet, A. Baghdadi, and J.-P. Diguët, “Optimization of deep-learning detection of humans in marine environment on edge devices,” in *Proc. of the IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Oct. 2022, (in press).
- [28] J. Gasienica-Jzkowy, M. Knapik, and B. Cyganek, “An ensemble deep learning method with optimized weights for drone-based water rescue and surveillance,” *Integrated Computer-Aided Engineering*, pp. 1–15, 01 2021.
- [29] J. Solawetz, “Aerial maritime drone object detection dataset,” July 2022. [Online]. Available: <https://public.roboflow.com/object-detection/aerial-maritime>
- [30] “Ships Dataset.” [Online]. Available: <https://www.kaggle.com/dsv/2906695>
- [31] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, “CSPNet: A new backbone that can enhance learning capability of CNN,” in *Proc. of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 390–391.
- [32] J. Redmon, “Darknet: Open source neural networks in C,” <http://pjreddie.com/darknet/>, accessed: 2022-04-14.
- [33] D. Sarkar, R. Bali, and T. Ghosh, *Hands-On Transfer Learning with Python: Implement Advanced Deep Learning and Neural Network Models Using TensorFlow and Keras*. Packt Publishing, 2018.
- [34] “COCO - common objects in context web site,” <https://cocodataset.org/>, accessed: 2022-06-20.