Real-Time Human Detection in Marine Environment Using Deep Learning on Edge Devices

Mostafa Rizk, Amer Baghdadi, J-Ph Diguet

Real-Time Human Detection in Marine Environment Using Deep Learning on Edge Devices

Mostafa Rizk[†§], A. Baghdadi[†], J-Ph. Diguet[‡]
† IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238 Brest, France
‡ CNRS, IRL CROSSING, Adelaide, Australia
§ Lebanese International University, CCE Department, Lebanon
mostafa.rizk@imt-atlantique.fr

***Abstract*—Artificial intelligence (AI) techniques based on deep learning provide robust solutions to detect and locate objects. The achieved performance prove the relevance of convolution neural networks (CNNs) in circumventing existing computer vision challenges. The goal of this work is to exploit the advantages of AI-based methods in detection of floating humans in open water to aid marine search and rescue missions. This has the potential to save lives while simultaneously saving efforts and expenses. In this work, we explore the use of You Only Look Once (YOLO) in detecting humans in maritime environment. A custom dataset is used to train the available YOLOv4 models. The trained models are assessed using recognized evaluation metrics. In addition, the inference speed is targeted towards embedded low-power hardware edge devices. The obtained results reveal that YOLOv4 can identify persons in a marine environment in real time with acceptable accuracy and precision.**

***Index Terms*—Deep learning, YOLO, Maritime, Human detection, Edge devices**

I. Introduction

Deep learning algorithms have recently revealed effective methods for detecting, classifying, and localizing multiple objects in pictures and videos. The development of neural networks has improved performance to the point that it is regarded on par with human performance. However, the improved detection performance comes at the cost of greater system resources and power consumption, particularly in real-life applications that demand real-time processing and high precision and accuracy.

You Only Look Once (YOLO) [1] was recently developed as an efficient CNN-based model for real-time object identification. YOLO outperforms two-stage models which classify objects based on pre-selected regions such as region-convolutional neural network (R-CNN) [2]–[4]. YOLO approach eliminates the required post-processing operations in order to reduce the complexity and enhance the detection speed. Several versions of YOLO have been introduced with different neural network architectures [5], [6]. The most recent architecture of YOLO, known as YOLOv4 [7], has been proven to identify objects in real time with great precision. It is shown that YOLOv4 exceeds in terms of speed and accuracy other real-time neural networks such as EfficientDet [8] and RetinaNet [9].

In this work, we aim to exploit the advancements in AI to rapid the direction of rescuers and medical teams to lost individuals during directing of rescuers and medical teams which greatly impacts in saving human lives and in lowering the missions' cost. The use of YOLOv4 model is explored in detecting individuals in maritime environments.

II. Method

We create a dataset of images showing humans in maritime environment. The dataset includes 6462 images with 16795 bounding boxes. The number of humans in the scene varies between the images. The images are split randomly by 70% as training dataset, 10% as validation dataset and 20% as testing dataset.

Several YOLOv4 models exist in the literature with different architecture specifications and detection performance in terms of accuracy and precision, detection speed and required energy budget. We examine three different YOLOv4 networks: YOLOv4 Large, YOLOv4 Tiny and YOLOv4 Tiny-3l. The original YOLOv4 network consists of 162 layers and uses mish activation functions. YOLOv4 Tiny is the compressed version of YOLOv4. It uses the simplified network structure of CSPDarknet53-tiny. It compromise 38 layers with LeakyRelu activation functions and only two detector heads. YOLOv4 Tiny-3l architecture is similar to YOLOv4 Tiny, but with three detector heads.

The training is conducted using the Darknet framework [10] using Quadro RTX 4000 from Nvidia. Transfer learning is adopted in order to maintain the generalization. We make use of the weights generated in previous training processes of networks with similar architecture specifications targeting COCO dataset. Note that the imported weights of the feature extraction layers are kept; whereas, the weights of the neck and the detector layers are eliminated. The networks' general architectures have not been altered. Only the depth size of the three convolution layers allocated before the YOLO detector layers are adjusted. The number of filters in these three convolution layers are modified considering our case where only one class (Person) is targeted.

The number of images per batch is set to 64. The total number of iterations is set to 2000. The initial learning rate for training is set to 0.001 and it scales down two times by 0.1 at iteration 1600 and 1800. The input images are down sampled into 416×416 or 608×608. While training the models, data augmentation is activated. Mosaic data augmentation type is

TABLE I
EVALUATION RESULTS OF THE TRAINED YOLOV4 MODELS

Target Model	Image Resolution	Training time (h)	Data Augmentation	mAP VOC07	mAP VOC12	Precession	Recall	F1 score	avg IOU	Average FPS on Jetson Xavier NX					
										Mode0	Mode1	Mode2	Mode3	Mode4	Mode5
YOLOv4 Large	416×416	03:22	-	60.46	58.78	61.61	70.80	65.88	62.14	10.1	10.7	10.8	8.8	9.4	6.5
			mosaic	64.27	65.66	62.57	75.48	68.42	63.8						
			mosaic+cutmix	65.63	69.04	61.95	78.03	69.07	64.83						
	608×608	06:00	-	55.16	59.15	66.15	69.98	68.01	63.13	5.6	5.7	5.7	5.0	4.9	3.7
			mosaic	64.64	64.91	66.74	73.45	69.93	64.92						
			mosaic+cutmix	65.82	69.37	63.96	78.28	70.4	65.39						
YOLOv4 Tiny	416×416	00:24	-	57.00	56.53	49.03	73.39	58.79	61.34	58.4	80.6	72.0	54.3	67.4	53.0
			mosaic	56.34	56.10	48.25	73.95	58.40	61.75						
			mosaic+cutmix	56.90	56.90	47.40	73.85	57.74	61.85						
	608×608	00:35	-	59.29	60.07	53.98	74.91	62.75	62.08	35.3	44.3	45.6	31.3	39.4	30.3
			mosaic	60.91	63.10	52.83	77.00	62.66	62.58						
			mosaic+cutmix	60.59	62.47	53.04	76.57	62.67	62.66						
YOLOv4 Tiny-3l	416×416	00:25	-	54.89	53.31	53.78	72.32	61.69	60.97	50.1	60.0	69.9	43.6	58.6	48.0
			mosaic	55.28	54.17	53.16	73.04	61.53	61.57						
			mosaic+cutmix	55.80	55.41	53.09	73.95	61.81	61.88						
	608×608	00:47	-	59.12	57.46	59.27	70.92	64.57	61.81	32.6	39.7	40.3	35.1	35.7	26.8
			mosaic	59.54	59.43	56.63	73.26	63.88	62.35						
			mosaic+cutmix	60.08	59.92	57.60	73.17	64.46	62.21						

used where 4 images are merged into one. When activated, Cutmix data augmentation type is applied for the classifier only. The saturation of input images and their exposure (brightness) are randomly changed as well as the rotation.

A. Results

The trained models are evaluated using the test dataset. Table I shows the obtained mean average precision considering VOC07 and VOC12 performance metrics [11]. In addition, the table shows the obtained values of precession, recall, F1-score and average intersection over union (IOU) considering 0.5 IOU threshold. Furthermore, the inference speed of the trained models is evaluated using several captured videos targeting embedded platforms. Table I shows the obtained speed of the trained models in frames per second (FPS) when applied to the captured videos on Jetson Xavier NX development kit while operating on different power modes. The obtained results show that applying DA enhances the detection performance (mAP, precision, recall, F1-score and average IOU). The use of cutmix DA increases the enhancement ratio in most of the cases. The use of higher image resolution enhances the mAP performance but at the cost of reduced inference speed and longer training time.

III. CONCLUSION

In this paper, the use of YOLOv4 in detection of humans in maritime environments is investigated. Available YOLOv4 architectures are trained on a custom dataset. The trained models are evaluated in terms of mAP, precession, recall and average IOU. Also, the performances of the models are examined on embedded platforms using our own videos showing humans in open water. The obtained results show that YOLOv4 can achieve real-time detection of humans in maritime environment with acceptable accuracy and precession. For example, YOLOv4 Tiny achieves an inference speed of 45.6 FPS with mAP of 63.10 when running on Jetson Xavier NX considering 608×608 resolution. Future work will include applying optimization techniques such as quantization and pruning to increase the inference speed and study their impact on the detection performance.

REFERENCES

[1] 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.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[3] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[4] 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.

[5] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[6] ——, "YOLOv3: An incremental improvement," 2018.

[7] 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

[8] 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.

[9] 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.

[10] J. Redmon, "Darknet: Open source neural networks in C," http://pjreddie.com/darknet/, accessed: 2022-04-14.

[11] R. Padilla *et al.*, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics*, vol. 10, no. 3, 2021.