IUST Iran University of Science and Technology

Iranian Journal of Electrical and Electronic Engineering

Journal Homepage: ijeee.iust.ac.ir

Research Paper

Real-Time YOLO Based Ship Detection Using Enriched Dataset

A. Ataee*, and J. Kazemitabar*(C.A.)

Abstract: We propose a real-time Yolov5 based deep convolutional neural network for detecting ships in video. We begin with two famous publicly available SeaShip datasets each having around 9,000 images. We then supplement that with our self-collected dataset containing another thirteen thousand images. These images were labeled in six different classes, including passenger ships, military ships, cargo ships, container ships, fishing boats, and crane ships. The results confirm that our Yolov5 based algorithm can classify the ship's position in real-time from 135 frames per second videos with 99% precision.

Keywords: Artificial intelligence, Convolutional neural network, Object detection, Real-time detection, Ship detection, Yolov5.

1 Introduction

SHIP detection helps prevent illegal transportation, goods smuggling, border intrusion detection, as well as preventing the depletion of pollutants at the water level and in general is helpful for the ship monitoring and safety. The complexity of the water surface and its extent has made monitoring a challenge [1]. In the past, people had to rely on local binary pattern [2] and gradient histogram [3] to identify the ship. In recent years, however, thanks to rapid development of deep learning, stronger tools capable of learning semantic features are introduced to resolve problems of the traditional architecture. These tools are normally based on deep convolutional neural networks. Subsequent studies worked on how to integrate most of the ships' features and how to detect them more accurately and faster. A principal component of such research is data collection in different classifications of ships.

Some of the valuable efforts on the data collection subject are SeaShips [4], ABOships [5] and McShips [6] which are from great and professional data sets in the field of ships detection that are successfully trained using R-CNN (Regions with Convolutional Neural Networks) [7], Fast R-CNN [8], Faster R-CNN [9], YOLOV2 [10], YOLOV3 [11] and EfficientDet (Efficient Detection) [12] architectures. Table 1 indicates the results associated with each of these models.

Table 1 Research on ships detection, based on deep learning technology.

Network model	Dataset	Map	FPS
Faster ResNet101 [4]	SeaShips	0.9240	7
Yolov2 [4]	SeaShips	0.7906	91
Yolov3 [6]	McShips	0.7769	44
Faster ResNet152 [6]	McShips	0.7913	5

The R-CNN network scans the image with a window at different scales using the selective search algorithm. It follows the adjacent pixels which share color and tissues and considers the brightness ratios and suggests the image type depending on the type of image. For example, some 2,000 different areas are designated in the image and these areas are then given to a conventional network for feature extraction to make a final prediction based on them.

It converts the regions to the same size, and then each of these independent regions enters the convolutional network as an input and the network decides on each of the independent regions. Fig.1

Iranian Journal of Electrical and Electronic Engineering, 2023.
Paper first received 28 Mar 2022, revised 21 Oct 2022, and accepted 08 Nov 2022.
*The authors are with the Department of Electrical Engineering, Babol Noshirvani University of Technology, Babol, Iran.
E-mails: atefeataei@gmail.com and j.kazemitabar@nit.ac.ir.
Corresponding Author: J. Kazemitabar.
https://doi.org/10.22068/IJEEE.19.1.2476

displays the overall schematic of the R-CNN network [7].

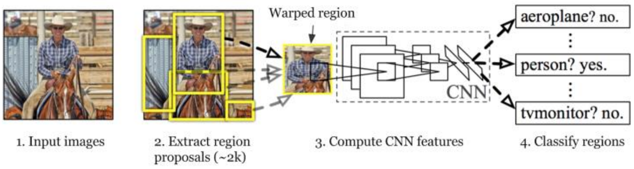


Fig. 1 Overview of R-CNN architecture [7].

In the improved version of this network, a better performance was found, but during studies related to ship detection based on deep learning, not only the precision of detecting the majority of models for each data collection was not high, the real-time performance of none of the models is comparable with the maximum reported precision. Choosing a single-stage model with processing power is necessary when we have more than 30 frames per second. In addition, we need to gather a general data collection of lighting variations, with diversity in ship's type from different brands of shipbuilding, shooting angle, image background, and scale [7].

In [13], a detection method titled feature balancing and refinement network (FBR-Net) was proposed. It eliminates the effect of anchors by adopting a general anchor-free strategy that directly learns the encoded bounding boxes. It then leverages the proposed attention-guided balanced pyramid to balance semantically the multiple features across different levels.

In [14], to resolve the problem of inconsistent imaging perspectives between ImageNet and earth observations, the authors propose an optical ship detector (OSD) pretraining technique to transfer the characteristics of ships in earth observations to SAR images from a large-scale aerial image dataset.

The authors in [15] propose a balance scene learning mechanism (BSLM) for offshore and inshore ship detection in SAR images. In [16], to solve the boundary discontinuity problem in OBB (oriented bounding box) regression, the authors propose to detect SAR ships by learning polar encodings. The encoding scheme uses a group of vectors pointing from the center of the ship target to the boundary points to represent an OBB.

Since 2016, five official versions of the YOLO Deep network have been published. Each version is an upgraded or integrated version with the previous versions of the algorithm. The YOLO network has become one of the most popular object identification algorithms today as a result of this technique. YOLO can recognize objects quicker than any other object detection method, including R-CNN, at a rate of at least 45 frames per second in real-time. Each picture in this method deals with both location and classification at the same time [7].

In this article, first, the data collection process for our custom-built dataset is been introduced. We then present our network. Applying two SeaShips data collection and self-collected dataset, the YOLO [15, 16] network is trained and its results are presented. The main contributions of our work can be summarized as follows:

- Building a custom ship dataset that is more versatile (has more classes) than the previous ones.
- The custom ship dataset has more images than the previously provided ones
- Implementing YoloV5 based algorithm that outperforms previously introduced algorithms for the famous SeaShip dataset.

2 Materials and Methods for Achieving Visual Data

SeaShips is a ship classification database, where 31,455 images are in 6 labeled classes, including four classes belonging to cargo ships and 2 for non-cargo ones. This collection covers container ships, fishing ships, dry-oil ships, general goods carrier ships, coal carrier ships, and passenger ships. Seaships collection is approximately 10,080 video sections in the real world obtained by surveillance cameras on the beach [4]. Despite its advantages, this collection is missing angle variations, backdrop alterations, and ship type variation. Moreover, the class of military ships is absent in the collection.

Another ship classification database is ABOships. The data set includes 9,880 images and 41,967 labels for nine ship classes. All of the ABOships pictures were captured during a multi-hour underwater filmmaking effort. This data was compiled from 135 videos shot with a camera having a 65-degree field of view and a Full HD resolution of (1920-720) at 15 frames per second in the MPEG format [5]. Although the backdrop in ABOships varies, the camera angle remains consistent, therefore it was preferable to film using Helishot rather than a fixed camera.

In this paper, we introduce our self-collected dataset. Six different types of ship categories were considered to train the model for the six-class problem. Passenger ships, military ships, cargo ships, container ships, fishing ships, and crane carrier ships are among the six types. Using Google images and YouTube videos, a collection comprising 13,129 images was created according to the Table 2. Fig. 2 indicates the steps to gather a self-collected dataset.

Table 2 Number of custom dataset images.

Classification	Number
Passenger Ship	2519
Military Ship	2034
Cargo Ship	1493
Container Ship	2553
Fishing Boat	2446
Crane Ship	2084
Total	13129

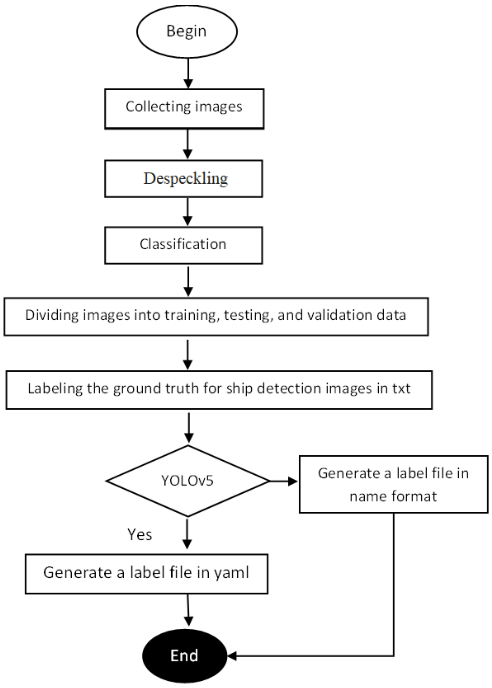


Fig. 2 The different stages of a self-collected dataset.

In the initial step, a large number of images was collected, and then the images are revised and removed using the average hash algorithm to discard repetitive images.

The hash method first transforms the input picture to a grayscale image before scaling it down. The size of the input picture will be reduced to 8×8 in order to generate a 64-bit hash. The mean of all 64 pixels is then computed, and the pixels are checked from left to right. If the gray value exceeds the mean, one is added to the hash; otherwise, it is zero [17]. The algorithm is shown in Fig. 3.

The video data is then transformed into an image at a rate of 25 frames per second, and the resulting images are then added to the self-collected dataset. As shown in Table 3, a total of 13,129 images are categorized as 640×640 pixels in jpg format. 10300 images are selected at random as training images, 2434 images as verification images, and 395 images as test images. Data collection from different ships was labeled in six different classes applying LabelImg software [18, 19], including crane ships, fishing boats, container ships, cargo ships, military ships and passenger ships. The coordinates of the center was also determined in the process. The width and length of the rectangle were stored in the tag txt file. Variety of viewpoint, scale, Intra-class, occlusion and illumination of pictures have all been taken into account in the image collection so that the network may operate better in the actual world. On a self-collected dataset, Fig.4 shows nine distinct pictures in the category of cargo ships.

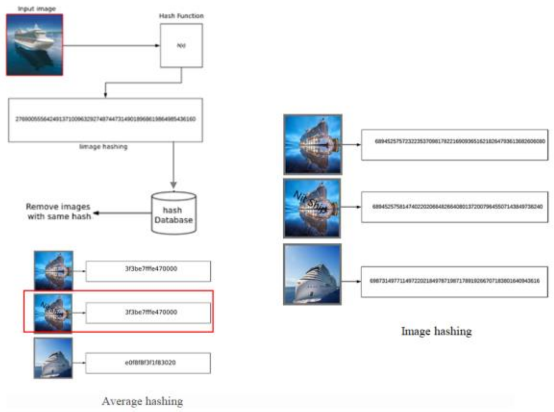


Fig. 3 Comparison of similarity of images with respect to the mean hash function. Hash value is represented in Hex.

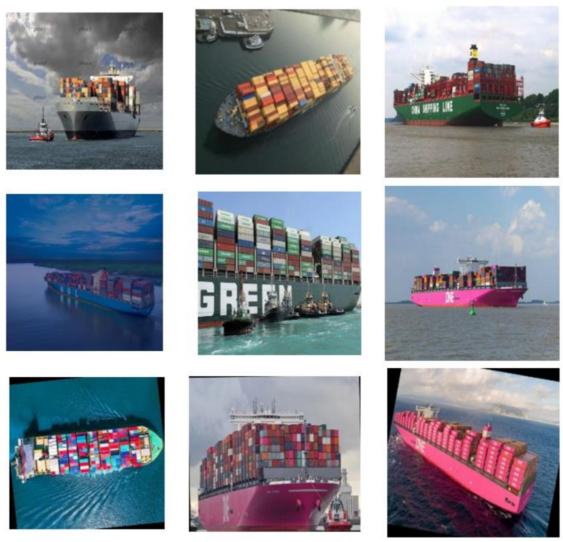


Fig. 4 Available ship types in the custom dataset.

Different ship datasets are compared in Table 3.

Table 3 Datasets for ship detection.

	Total Images	Number of Annotation Data	Number of Classes	Year of Publication
SeaShips	31455	40077	6	2018
Singapore	17450	192980	6	2018
McShips	14709	26529	13	2020
ABOships	9880	41967	9	2021
Our data	13129	14502	6	2023

3 You Only Look Once (YOLO)

YOLO is an object identification method that uses a neural network with several Convolutional layers to identify and determine the position of objects in real-time. It is a combination of a multi-stage process that implies "you only look once." It is ideal for this network's recognition function and can be built quicker than two independent neural networks to recognize and categorize items. YOLO network addresses the object detection issue as a regression problem that is independent of picture pixels and is more likely to be categorized.

In the YOLO algorithm, its IOU (Intersection over Union)[20] threshold value and the confidence are used for frames without object or frames with unclassified identified object or frames with overlapping or filtered identified objects. The criterion IoU ∈ [0:1] is used to filter frames with one unique object identified with it. According to Eq. (1), the confidence indicates the existence or absence of an object in the limiting box [21, 22].

$$\text{onfidence } = p(Object) * IOU_{pred}^{truth} \qquad 0 < p(Object) < 1 \tag{1}$$

$p(Object)$ is the probability of an object being within a cell and is between 0 and 1. Therefore, if there is nothing in that cell, the confidence is close to zero. First, the input image is divided into a raster grid of S×S. Then, a convolutional network is executed by Eq. (2) to calculate the output value. In Eq. (2), when p_c= 0 the network does not identify any object. In such a case, the corresponding predictions, $b_x, b_y, b_h, b_w, c_1, c_2, \ldots, c_p$ are ignored [14, 15].

$$y = [p_c(\text{confidence}), b_x, b_y, b_h, b_w, c_1, c_2, \ldots, c_p, \ldots]^T \in \mathrm{R}^{S+S+k+(5+p)} \tag{2}$$

3.1 Deep Yolov5 Network Configuration

The focus module is the first layer of the Yolov5 network, and it is intended to reduce model complexity and increase training speed. Figure 5 illustrates the architecture.

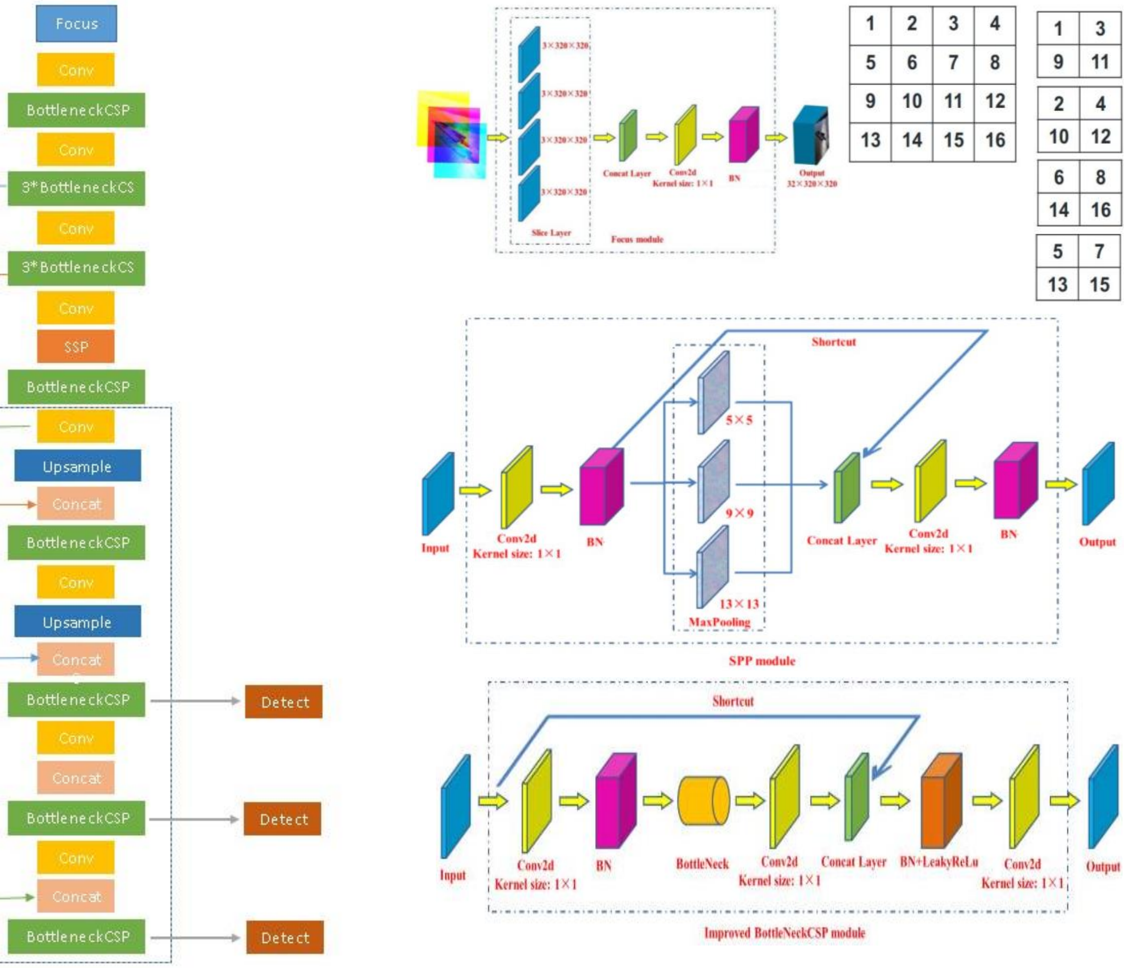


Fig. 5 Configuration of deep Yolov5s network [23].

The picture of the input color of three channels with the same size of 640× 640×3 is first to split into four slices of 320×320×3 in this structure. The cut process creates the second stage of the 320×320×12 matrix connection layer, after which the convolutional layer, which includes 32 convolutional layers, creates the 320×320×32 feature matrix. Finally, the results are transferred to the second module via the BN (Batch Normalization) layer and LeakyReLU activation functions. After crossing several convolutional layers and BottleneckCSP, the SPP (Spatial Pyramid Pooling) module, which is designed to improve the network and access multi-Resolute layer structure, comes the ninth layer of the network. This section improves the identification and comprehension of lower-level characteristics, as well as the detection of objects at various sizes. As a result, items of various sizes and scales may be recognized. The final detection layer is a vector with 33 properties: classes (6) +probability of the predicted class in the box (1)+coordinates of the box's surrounding location (4). The summation will then be multiplied by 3 to account for the anchor boxes [23, 24, and 25].

As a result of the architecture in Fig. 5, Yolov5 runs at more than 135 frames per second on a GPU of the Tesla P100 model.

3.2 Anchor Box Specifications

Anchor box feature in the default Yolov5s is 9 boxes with width and height of 10×13 ،16×30, 23×33, 30×61, 45×62, 59×119, 116×90, 156×198, 373×326 that is divided into three clusters. Next, different batch sizes are given to the network. However, regarding the apparent specifications by changing the Anchor box according to the Fig. 6 the network has been able to draw more specific bounding boxes to determine the ship's position.

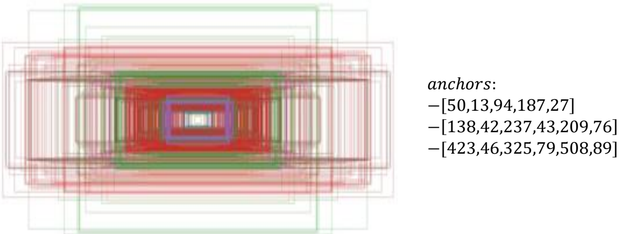


Fig. 6 The values of the proposed Anchor box.

4 Results

Figure 7 shows how the Yolov5 algorithm can identify and recognize each picture in 0.19 seconds. It also shows the results of ship detection using the SeaShips data set at various periods. Mineral ships are in the green boxes, passenger ships are in the blue boxes, oil ships are in the brown boxes, container ships are in the pink boxes, and finally fishing ships are in the purple boxes.

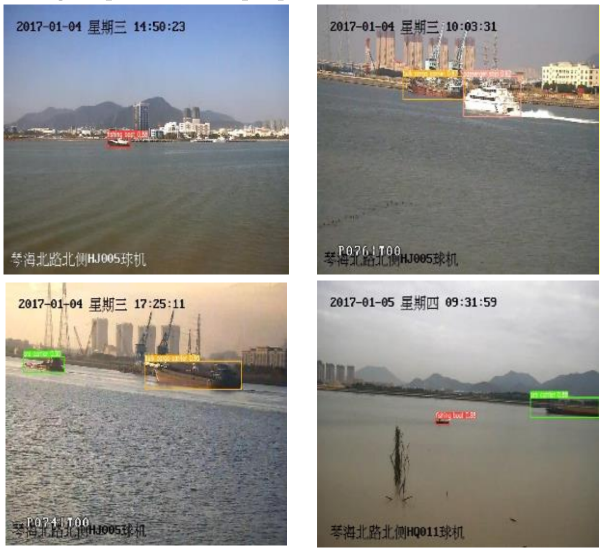


Fig. 7 SeaShips results dataset under Yolov5.

Figure 8 shows that when the number of iterations is increased, the error curve of the Yolov5 algorithm converges to zero. According to the Table 4, the error value is near zero after 500 epochs of the model (5).

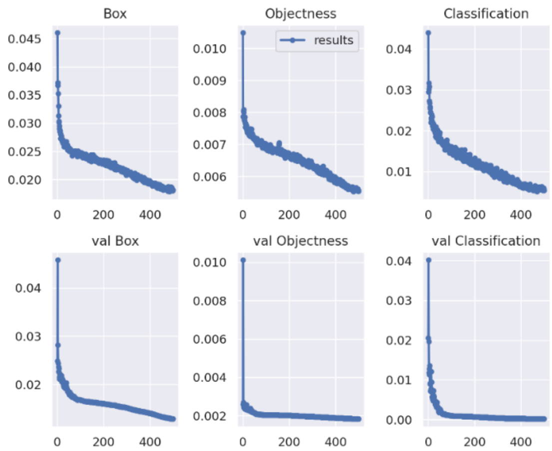


Fig. 8 Loss during training, related to both the predicted bounding box, classification and the loss related to the given cell containing an object, as well as their validation scores displayed as box, class and objectness.

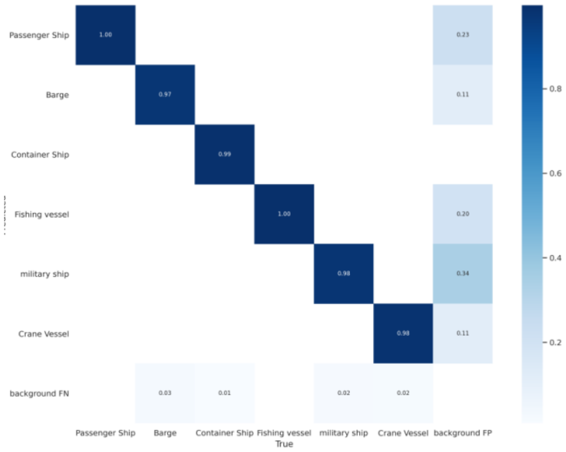
Table 4 Probability of loss according to the Fig. 8.

Objectness Loss (Validation)	0.01
The Loss of Box Validation	0.01
The Loss of Classification Validation	0.001
Objectness Loss	0.007
The Loss of Box	0.01
The Loss of Classification	0.001

4.1 Evaluating Yolov5 Model for Ship Detection of SeaShips Data

The crumble matrix is shown in Table 5. The True Positive sign in this matrix represents a positive member sample in the right category. False Negative is a positive class symbol that has gone undetected. True Negative is the sign of not misdiagnosing an item, whereas False Positive is the symbol of the incorrect label.

Table 5 Confusion Matrix.



Precision is the ratio of the number of positive results and the number of actual positive results. Regarding Fig. 9, model precision is increased via the training process and the model has reached the precision of 97.67%.

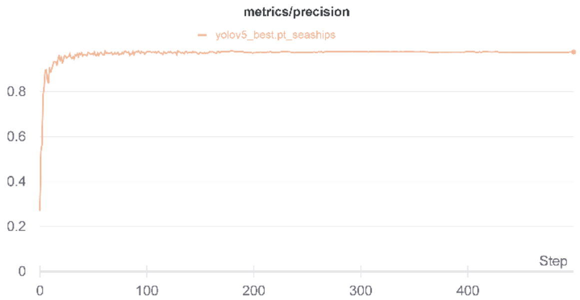


Fig. 9 Network precision criterion.

Figure 10 indicates the precision in terms of the confidence threshold for every 6 classes of ships and this forms at a confidence rate of 96.1%. It reaches its maximum value of 1.

The number of real ships (positive category) relative to the total recognized ship community is referred to as recall. In this research we intend to improve recall by identifying the model of all ships in the picture. Fig. 11 calculates the recall per epoch and shows that the model's accurate predictions improved with time, with the model reaching a peak recall of 97.28 %.

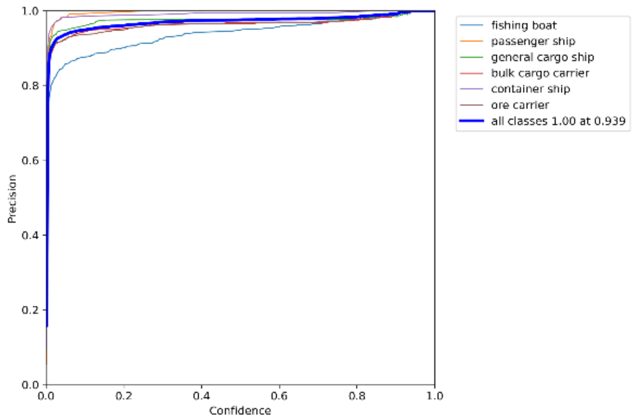


Fig. 10 Precision curve to confidence score.

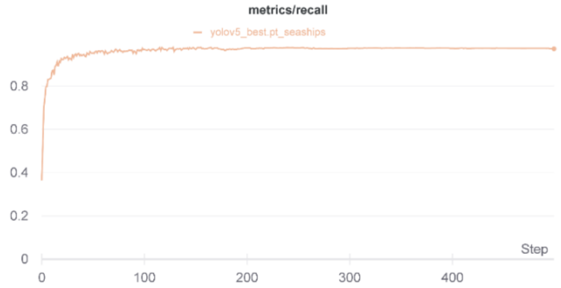


Fig. 11 Network recall criteria.

Figure 12 depicts the response of the confidence threshold to variations in the recall. Although the reduced threshold improves recall and precision in ship metrics, a significant number of ships are not examined. The high threshold exposes all stretches we will discover, but other objects are more likely to be misidentified. When the ship is proclaimed, and recall decreases, the diagram in Fig. 12 descends.

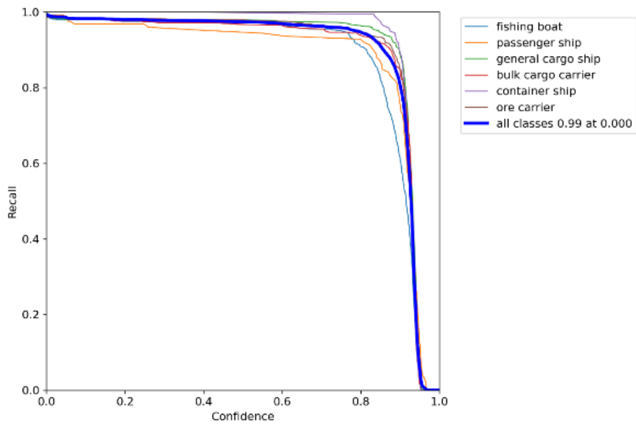


Fig. 12 Recall curve to confidence score.

The average precision of the diagram in Fig. 13 for all classes reached a stable point for the entire 500 epochs and was 98.51%. The precision-recall curve of the Fig. 14 is provided from the conformity of the diagrams in Fig. 12 and 10 and the area under its curve indicates the mean average precision (mAP) in every class.

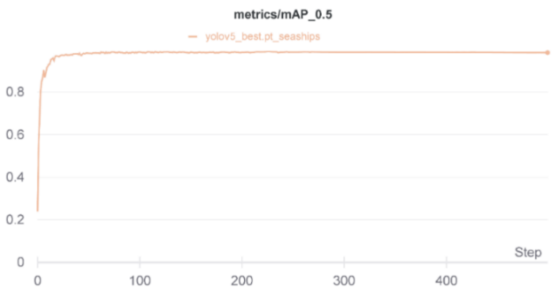


Fig. 13 Diagram of indicating the mean Average Precision(mAP) via the training process.

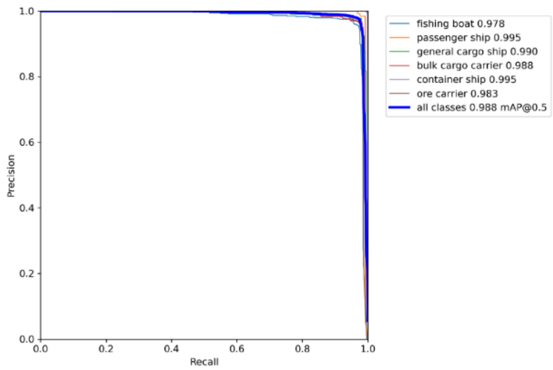


Fig. 14 Precision- Recall curve.

Figures 9-14 indicate that the model is gradually learning in every epoch, considering that the curves are stable up to the 500 epochs, so the number of network iterations is enough and the curves are stable at that point. Table 6 indicates the results of ship detection with SeaShips database.

Table 7 compares the results of average precision in the Yolov5 algorithm and Faster R-CNN (Resnet101) algorithm for every class.

Table 6 Performance of the Yolov5s model for SeaShips.

Type of Ships	mAP@ 0.5: 0.95	mAP@ 0/5	Recall	Precision
All	0.676	0.976	0.978	0.977
Fishing Boat	0.69	0.986	0.967	0.96
Passenger Ship	0.606	0.957	0.937	0.992
General Cargo Ship	0.682	0.984	0.986	0.976
Container Ship	0.677	0.976	1	0.998
Bulk Cargo Carrier	0.699	0.967	0.977	0.97
Ore Carrier	0.702	0.988	0.98	0.989

Table 7 Theredicted mAP of SeaShips vs. [4].

Type of Ships	Yolov5	Faster R-CNN(Resnet101)
All	0.976	0.924
Fishing Boat	0.986	0.8996
Passenger Ship	0.957	0.9178
General Cargo Ship	0.984	0.9384
Container Ship	0.976	0.9341
Bulk Cargo Carrier	0.967	0.9022
Ore Carrier	0.988	0.9368

In Fig.15, the result of the ship's detection is indicated at multiple times applying the personal dataset. In what follows, we review the error pattern and the precision of the network for this dataset.

Fig. 15 The result of the custom model test dataset for ship detection using the Yolov5 network architecture.

Fig.16 shows that the Yolov5 algorithm curve has progressively converged, and the error value has been lower as the number of iterations has steadily increased. According to Table 8, when the model is run for 500 epochs, the error value is very close to zero.

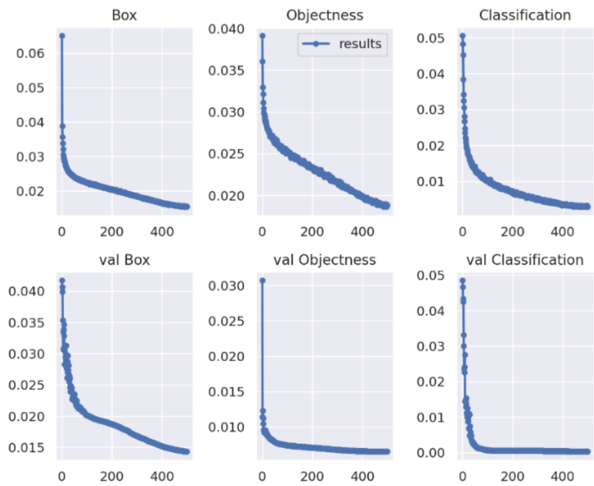


Fig. 16 Loss during training, related to both the predicted bounding box, classification and the loss related to the given cell containing an object, as well as their validation scores displayed as box, class and objectness.

Table 8 Probability of loss according to the Fig. 16.

Objectness Loss (Validation)	0.006
The Loss of Box Validation	0.01
The Loss of Classification Validation	0
Objectness Loss	0.01
The Loss of Box	0.01
The Loss of Classification	0.002

In Fig. 17, model precision increased via the training process and the model reached the precision of 98.19%. Fig. 18 evaluates the recall per repetition and indicates that the correct predictions of the model increases via the training process and the model reaches a recall of 98.18%.

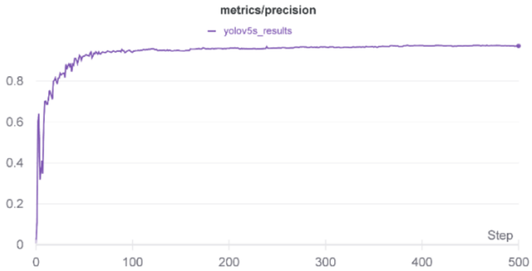


Fig. 17 Network precision criterion.

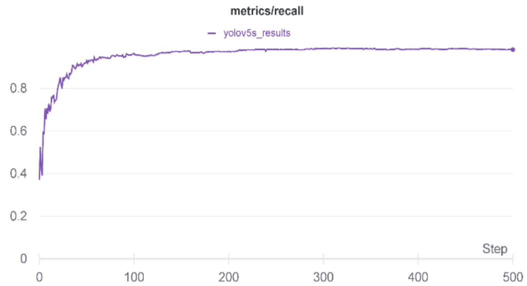


Fig. 18 Precision curve to confidence score.

From the conformity of Fig. 19 and 20, the precision-recall curve of Fig. 22 is provided and the area under the curve shows the mean average precision (mAP) in every class. The average precision of Fig. 21 for all classes per 500 epochs has reached stability and is equal to 99%.

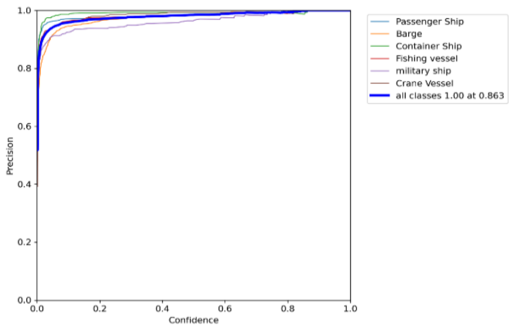


Fig. 19 Network recall criteria.

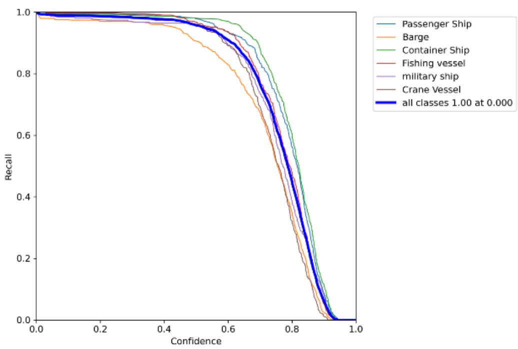


Fig. 20 Recall curve to confidence score.

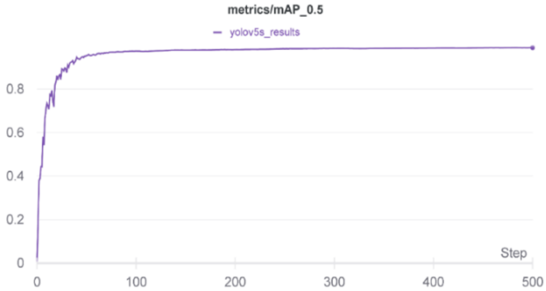


Fig. 21 Diagram of indicating the mean Average Precision(mAP) via the training process.

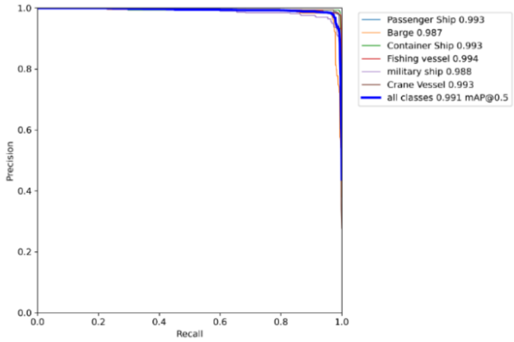


Fig. 22 Precision- Recall curve.

Table 9 indicates the results of ship detection with our custom collected dataset.

Table 9 Performance of the Yolov5s for custom datasets.

Type of Ships	mAP@ 0.5: 0.95	mAP@ 0/5	Recall	Precision
All	0.755	0.991	0.984	0.982
Passenger Ship	0.756	0.993	0.992	0.977
Barge	0.685	0.987	0.97	0.975
Container Ship	0.766	0.993	0.99	0.973
Fishing Boat	0.766	0.994	0.995	0.997
Military Ship	0.775	0.988	0.974	0.948
Crane Ship	0.753	0.993	0.982	0.987

5 Conclusion

In terms of the success of convolutional networks and more sophisticated networks like YOLO in classification and scenario identification, it has become clear in this paper that these networks extract high-level and appropriate features from huge picture databases. The average precision criteria were given particular attention in the backdrop of research in the area of object detection. As a result, choosing and building a network that can provide excellent speed while maintaining enough precision is impressive. Initially, the Yolov5 algorithm was used to identify and determine the ship in this study. Then, the processing speed of Yolov5 is compared to that of other methods. SeaShips data yielded the greatest mean network precision of 92.40 % when a quicker two-step method (ResNet101) was used. On this dataset, the YOLOv5 single-stage method is used to get high precision of 97.60 %. Furthermore, a personal dataset including 13,129 images was introduced in 6 labeled classes, resulting in an effective and re-identified model on video data with 99% precision.

Intellectual Property

The authors confirm that they have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing to publication, with respect to intellectual property.

Funding

The work of Javad Kazemitabar was supported by Babol Noshirvani University of Technology, under grant # 1400/BNUT/393049.

CRediT Authorship Contribution Statement

A. Ataee: Data Curation, Software and Simulation, Original Draft Preparation. **J. Kazemitabar:** Idea & Conceptualization, Software and Simulation, Revise & Editing.

Declaration of Competing Interest

The authors hereby confirm that the submitted manuscript is an original work and has not been published so far, is not under consideration for publication by any other journal and will not be submitted to any other journal until the decision will be made by this journal. All authors have approved the manuscript and agree with its submission to “Iranian Journal of Electrical and Electronic Engineering”.

References

[1] H. Luo, K. Wu, Z. Guo, L. Gu, and L. M. Ni. "Ship detection with wireless sensor networks." *IEEE Transactions on Parallel and Distributed Systems*, Vol. 23, No. 7, pp. 1336-1343, 2011.

[2] C. Zhu, R. Wang and Jun Guo, “A Novel Hierarchical Method of Ship Detection From Spaceborne Optical Image Based on Shape and Texture Features,” *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 48, No. 9, pp. 3446-3456, 2010.

[3] S. Qi, J. Ma, J. Lin, Y. Li, and J. Tian. "Unsupervised ship detection based on saliency and S-HOG descriptor from optical satellite images." *IEEE Geoscience and Remote Sensing Letters*, Vol. 12, No. 7, pp. 1451-1455, 2015.

[4] 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.

[5] B. Iancu, V. Soloviev, L. Zelioli, and J. Lilius. "ABOships-An inshore and offshore maritime vessel detection dataset with precise annotations." *Remote Sensing*, Vol. 13, No. 5, 2021.

[6] Y. Zheng, and S. Zhang. "McShips: A large-scale ship dataset for detection and fine-grained categorization in the wild." *In 2020 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1-6, 2020.

[7] G. Ross, 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*, pp. 580-587, 2014.

[8] X. Wang, A. Shrivastava, and A. Gupta. "A-fast-rcnn: Hard positive generation via adversary for object detection." *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2606-2615, 2017.

[9] B. Liu, W. Zhao, and Q. Sun. "Study of object detection based on Faster R-CNN." *In 2017 Chinese Automation Congress (CAC)*, pp. 6233-6236, 2017.

[10] J. Redmon, and A. Farhadi. "YOLO9000: better, faster, stronger." *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263-7271, 2017.

[11] J. Redmon, and A. Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv, 1804.02767, 2018.

[12] M. Tan, R. Pang, and Q. V. Le. "Efficientdet: Scalable and efficient object detection." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10781-10790, 2020.

[13] J. Fu, X. Sun, Z. Wang and K. Fu "An Anchor-Free Method Based on Feature Balancing and Refinement Network for Multiscale Ship Detection in SAR Images." *IEEE Trans. Remote Sensing*, Vol. 59, No. 9, pp. 1331-1344, 2021.

[14] W. Bao, M. Huang, Y. Zhang, Y. Xu, X. Liu and X. Xiang "Boosting Ship Detection in SAR Images With Complementary Pretraining Techniques," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 14, pp. 8941-8954, 2021.

[15] T. Zhang, X. Zhang, J. Shi, S. Wei, J. Wang, J. Li, H. Su and Y. Zhou "Balance Scene Learning Mechanism for Offshore and Inshore Ship Detection in SAR Images," *IEEE Geoscience and Remote Sensing Letters*, Vol. 19, pp. 1-15, 2020,

[16] Y. He, F. Gao, J. Wang, A. Hussain, E. Yang, and H. Zhou, “Learning Polar Encodings for Arbitrary-Oriented Ship Detection in SAR Images”, *IEEE Journal Of Selected Topics in Applied Earth Observations And Remote Sensing*, Vol. 14, pp. 3846 -3859, 2021.

[17] S. Ashwin, Y. Mao, and M. Wu. "Robust and secure image hashing." *IEEE Transactions on Information Forensics and Security*, Vol. 1, No. 2, pp. 215-230, 2006.

[18] Tzutalin, LabelImg. Available online: https://github.com/tzutalin/labelImg.

[19] T. Hastie, R. Tibshirani, and J. Friedman. "Overview of supervised learning." *The Elements of Statistical Learning*, Springer, New York, NY, pp. 9-41, 2009.

[20] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren. "Distance-IoU loss: Faster and better learning for bounding box regression," *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, No. 07, pp. 12993-13000, 2020.

[21] J. 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*, pp. 779-788, 2016.

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

[23] ultralytics. YOLOV5. AVAILABLE ONLINE: HTTPS://GITHUB.COM/ULTRALYTICS/YOLOV5.

[24] Y. Liu, B. Lu, J. Peng, and Z. Zhang. "Research on the use of Yolov5 object detection algorithm in mask wearing recognition." *World Scientific Research Journal*, Vol. 6, No. 11, pp. 276-284, 2020.

[25] B. Yan, P. Fan, X. Lei, Z. Liu, and F. Yang. "A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved Yolov5." *Remote Sensing* 13, no. 9, 2021.

A. Ataee received her B.S. degree in Electrical Engineering from Arak University of Technology in 2018 and the M.S. Electrical Engineering from the Babol Noshirvani University of technology in 2021. Since 2020, she has worked in several companies. Her research interests include machine vision, image processing, and deep learning.

J. Kazemitabar received his B.S. degree in Electrical Engineering from Sharif University of Technology in 2003 and the M.S. and Ph.D. degrees both in Electrical Engineering from the University of California Irvine in 2005 and 2008, respectively. From 2008 until 2015 he worked in several US companies. In 2015 he joined Babol Noshirvani University of Technology as an Assistant Professor. His research interests include, wireless communication, security, and fraud detection. In 1998, he won the silver medal in Iran national Mathematics Olympiad. He also ranked third in the nationwide entrance examination of Iranian universities in 1999. He is the author of the book “Coping with Interference in Wireless Networks”.