Convolutional Neural Networks for Detecting Ship Hatch Closing Moment: A Case Study Using YOLO Family

Francisco Carlos Silva Pimentel*, Roberto de Pádua Carvalho Reis*, Daniel Lima Gomes Jr.†, Siti Sarah Maidin‡, and Omar Andres Carmona Cortes†*
* Programa de Pós-Graduação em Engenharia da Computação e Sistemas (PECS)
Universidade Estadual do Maranhão (UEMA), São Luis – MA Brazil
Email:{fcalpimentel, adm.robertoreis}@hotmail.com
†Departamento de Computação (DComp)
Instituto Federal do Maranhão (IFMA), São Luis – MA Brazil
Email: {omar, daniellima}@ifma.edu.br
‡ INTI International University, Malaysia
Email: sitisarah.maidin@newinti.edu.my

***Abstract*—This paper presents a neural network-based computer vision approach for detecting ship cargo-hold hatch closure. The investigation is relevant since weather conditions, especially rainfall, cause damage to cargo such as sodium sulfate, sugar, corn, corn bran, and potassium chloride, among others. Registering when the cargo hold hatch is closed could prevent damage to the cargo, avoiding prejudice to transportation companies. Our proposal uses YOLO framework vision detection as an economical alternative to the current state-of-the-art for detecting ship hatch closing with expensive and complex solutions. This investigation presents an experiment in a tailored dataset, and results are applied to real-time video detection that validates a stable and accurate solution to the problem of ship hatch detection. Results have shown that even though regular YOLO v4 reaches better metrics, with an accuracy of 91.55%, Fast YOLO v4 is better for real-time detection but with a penalty of lower accuracy.**

***Index Terms*—Hatch cover closure, hatch recognition, neural networks, YOLO, computer vision**

I. Introduction

Innovations in the automation of mechanized port operations are significant, especially in loading and unloading products, considering that the number of studies published in this area is still small. However, many engineers and researchers have developed intelligent automated control systems for international containerized bulk operations. For instance, the Euromax terminal in Rotterdam, in the Netherlands, has achieved the world's highest degree of containerized automation. On the other hand, automation in bulk terminal operations has developed at a slow pace.

Moreover, building an automated bulk operations load system is based and premised on the recognition of ship hatches. In this context, detecting the hatch rapidly and accurately is still a problem that needs an urgent solution in the current port production scenario [1]. Weather conditions, such as rainfall, significantly impact bulk operation loading systems. In cargo such as sodium sulfate, sugar, corn, corn bran, wheat, potassium chloride, soybeans, and others, it affects logistics and losses for cargo and transport companies due to fines for late delivery. Brazilian ports report up to 110 days of rain annually, halting their operations. Not wasting time checking whether the hatch cover has been completely closed is essential. Capturing the exact moment of closure of the cargo hold is vital to validate an operational stop and make port operations more efficient.

Thus, this research aims to solve the following problem: how to record the exact closing moment of the cargo hold hatch cover. In its first approach, the main objective is to validate the recognition detection of the cargo hold hatch. The research is still relevant because the current state-of-the-art proposes solutions that require high-cost equipment. Therefore, the main goal of this paper is to use computer vision and image processing techniques as an economical and suitable alternative. As part of the solution, we propose using a Convolutional Neural Network (CNN), a technique categorized as deep learning with high performance in different types of classification and detection [2]. We compare two architectures You Only Look Once (YOLO) v4 [3] and Fast YOLO v4. We used this version because is better documented than the last version.

Although it seems simple to solve the problem by image recognition with a CNN, this issue is complex and offers some challenges to be completed. The first one, hatch cover changes according to the size and design of the ship [4], making available several types of mechanisms and systems. Figure 1 illustrates some examples. Secondly, considering the variety of ships and hatches, the algorithm must be able to detect what is hatch and what is not. Thirdly, the model must not be susceptible to the camera angle, which can change the object size and create occlusion in the image. Finally, the model has to be fast enough to perform detection in real-time videos.

Due to these challenges and the detection accuracy de-

Fig. 1. Types of hatches covering mechanisms

manded to recognize the hatch closure in real time, the CNN YOLO network has been chosen because of its accuracy and speed image recognition [3]. So, this paper is divided as follows: Section II presents some related works that propose much more expensive solutions for the problem; Section III introduces how the dataset was built and how data augmentation was used; Section IV shows the experiments in the dataset and real-time video; finally, Section V concludes this works summarizing results, exposing the weakness, and proposing future work.

II. Related Works

Ship hatch recognition is the first prerequisite for automating loading procedures [5]. The current studies have focused on expensive equipment and complex solutions based on point cloud data encoding algorithms.

Not long ago, studies have aimed using Mobile Laser Scanners (MLS) to obtain point cloud data along with the MLS trajectory. It became popular in the 3D reconstruction of walls, corridors, open doors, terrain, and surfaces [6]. To obtain relevant data, a laser scanner must perpendicularly faces the ship's surface with special cameras that capture the concentration of those 3D points. An algorithm converts the cloud points data into a 2D plane. Finally, other algorithms obtain the cargo hold's x and y coordinates and recreate the ship's deck and hatch edges [1], [7].

In the same sense of Laser Scanning or Laser Imaging Detection and Ranging (LiDAR), Ziang et al. [7], Miao et al. [8], and Mi et al. [9] use some form of point cloud building 3D models of the ships, which demands the use of expensive equipment and building costly infrastructure to hold and maintain the lasers cannons.

In this context, this research suggests employing a convolutional neural network known as You Only Look Once (YOLO) as a cost-effective and efficient alternative to replace laser-based systems. To the best of our knowledge, no other solutions are based on CNNs.

III. The Dataset

Even though there exists datasets for maritime application, most of them aim to detect the class of a ship, in which the recognition can result in providing critical information to maritime traffic, ship's crew, ship owners, and shipping companies, including offices of the Maritime Safety Administration (MSA) [10].

Additionally, an effort exists to consolidate datasets dedicated to the ship's images for classification and detection; however, the number of images is limited, and their resolution varies. Table I [11] summarizes those image datasets.

TABLE I
Ship image datasets

Dataset	Total images/videos
MARVEL	239,622
VMI	3,750
Singapore Marine	19, 600
MarDCT Classification	6,743
ARGOS Boat Classification Benchmark	2,339
VAIS	3,000
ISH	190,00
VesselFinder	190,000

Most of those datasets contain images divided into 25 ship classes or more. The image domain for this research will not use all classes of vessels, *i.e.*, we built a dataset using only three ship types: general cargo, bulk carriers, and other ship types. These categories have been chosen because those ships maintain cargo holds. Furthermore, a careful selection was made because not all images are suitable for our application; considering that those datasets aim to classify ship types, several images show the side hull, making it impossible to identify the cargo hold.

Therefore, to compose the ideal dataset, it was necessary to choose images from public datasets in angles showing all the cargo-hold with the hatch opened or closed, especially pictures in which the ship is docked. Preferably, we chose images during loading and unloading operations, as illustrated in Figure 2.

The final dataset was composed of the majority of public-domain images. Due to the lack of specific images, we initially gathered 417 images from different angles and bulk carriers' models. A few images registered situations where hatches are closed and opened in the same picture. To balance the dataset, half of the images expose the cargo hold with the hatch open and vice-versa.

Moreover, to submit the dataset to the YOLO training process, each image needs annotations, meaning that for every

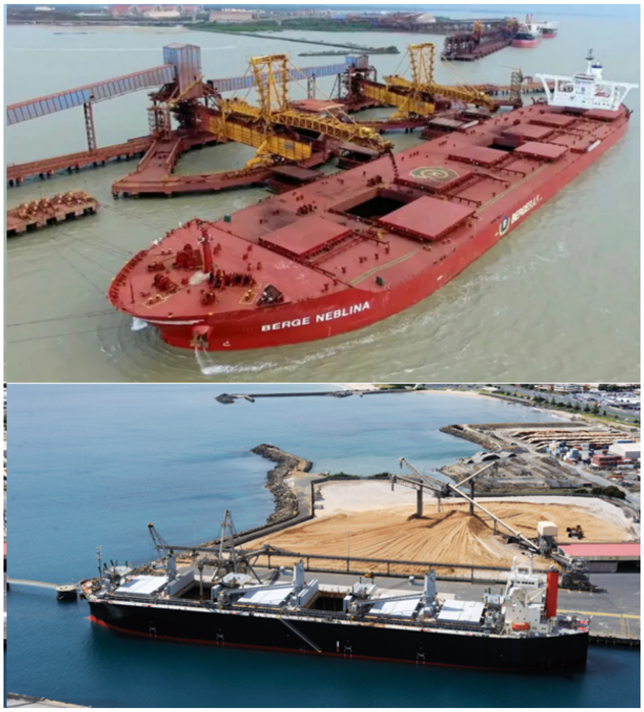

Fig. 2. Preferably images

single file, a text file with the same name as the image file, and .txt extension is required. Each text file contains annotations (numbers) for the corresponding image file, the class number of the objects and a box constraint with x and y box center coordinates followed by height and width. If the same image has two or more object classes, the same number of box coordinates need to be present in the annotation file.

Some public datasets offer images with annotations, but all the annotations needed to be created or recreated for the kind of images required for this research. The process of creating annotations was assisted by software, being a time-consuming part of the whole process that demands special attention; otherwise, results can be inferior.

A. Data Augmentation

In order to improve the number of images for training and testing the YOLO family architectures, we performed data augmentation in two steps. The first uses the most common augmentation methods, such as horizontal flip, vertical flip, rotation, and translation. After all techniques were completed, the data set increased from 417 to 4170, following parameters presented in Table II and updating the annotation files.

In the second stage, we performed operations that transform only the color and alpha channels in order to avoid labeling all images again. The following operations were performed: color light, color image saturation, color hue image, gaussian blur, distortion averaging blur, distortion median blur, distortion image erosion, distortion image dilatation, distortion image opening, distortion image closing, distortion image morphological gradient, distortion top hat, distortion black hat, gray

TABLE II
DATA AUGMENTATION - STAGE I

Method	Parameter
Horizontal flip	-
Vertical flip	-
Vert. and horiz. flip	-
Rotation	45^o,90^o,270^o
Translations	150, 150
Translations	-150, 150
Translations	150, -150
Translations	-150, -150

sharpen, gray emboss, gray edge, gray contrast, gray edge canny image, and grayscale. These operations increase the number of images to 27,213 in total.

IV. EXPERIMENTS

In this section, we detail the experiments that were carried out. The first one used the first version of the dataset, containing shape-based data augmentation with 4170 images, presenting an accuracy of only 62%. Thus, the remaining experiments in image results were based on the second-stage data augmentation, composed of 27,213 images, as previously mentioned. As stated earlier, this approach was chosen because we do not need to annotate each image again. Thus, the first set of experiments regards image recognition, while the second one uses the resulting models in real-time video detection.

A. Setup and Metrics

In the training stage, the dataset was split into the rate 90/10. The framework darknet was used for training YOLO v4 and compiled in a Windows 11 operating system. The architecture to train the CNNs was a notebook Acer Nitro Gamer i5, with 8G RAM and NVIDIA GFORCE GTX 1650. The code for the video experiments was implemented in Python 3.9 with OpenCV library.

Configurations for training YOLO and fast YOLO use the same hyperparameters such as number of classes 2, batch size 64, decay 0.0005, learning rate of 0.001, max batches of 6000, and steps [4800,5400].

The experiments were conducted using YOLO v4 and Fast YOLO v4, and to evaluate the detection of both models, we used the traditional metrics for machine learning algorithms: accuracy, precision, recall, F1 score, mAP, True Positives (TP), and False Positives (FP).

B. Image Results

Figure 3 presents the mAP metrics for YOLO V4 (left) and Fast YOLO (right), respectively. Because mAP is a valuable metric that takes into account precision and recall, we consider it the main result for this dataset. As we can see, YOLO V4 presents a significantly better mAP than Fast YOLO, 92% against 52%.

The good results presented in mAP for YOLO v4 are also reflected in Table III, in which we can see that YOLO v4 clearly presents the best results.

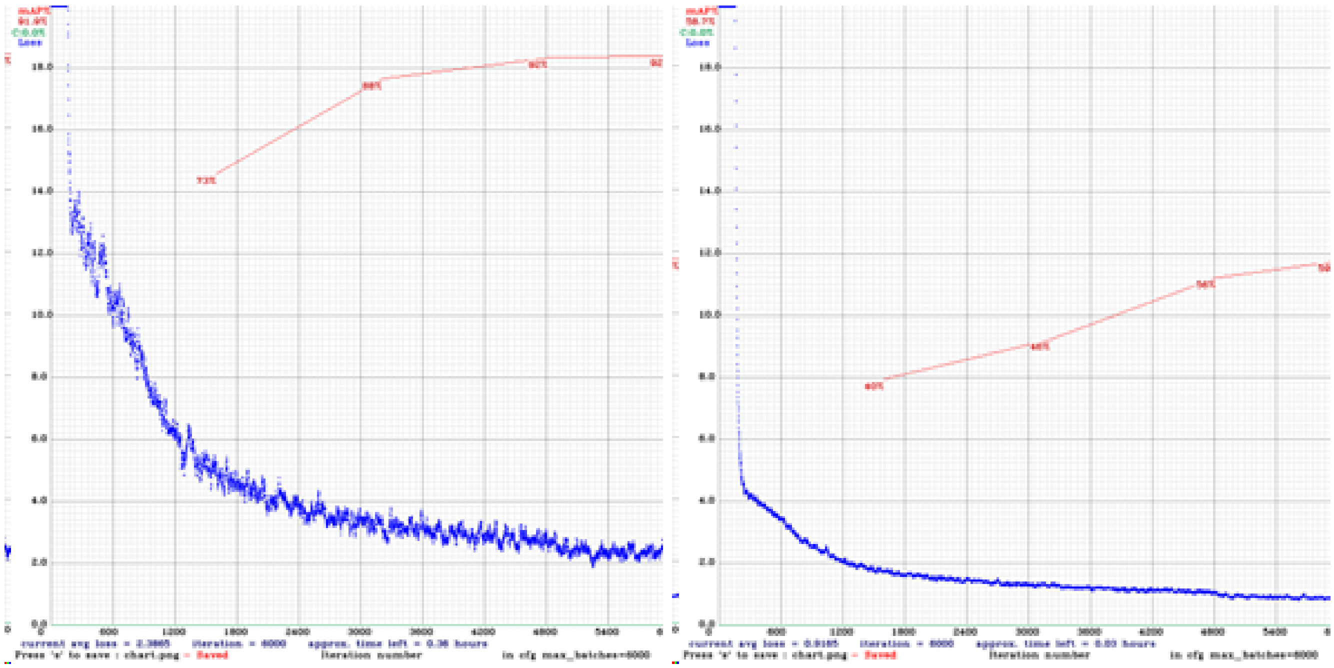

Fig. 3. mAP for YOLO v4 (left) and Fast YOLO (right)

TABLE III
TRADITIONAL METRICS

CNN	Accuracy	Precision	Recall	F1-score
YOLO	**0.92**	**0.81**	**0.90**	**0.80**
Fast YOLO	0.59	0.73	0.53	0.62

An experiment was conducted to verify the behavior of the algorithms through a k-fold method and check if the previous results using the ratio 90/10 are not effects of randomness. Tables IV and V show the outputs and the mean of each metric for YOLO and FastYOLO, respectively, using a k-fold experiment with $k = 5$. As we can see, YOLO consistently presents significantly better results than FastYOLO in all metrics, even when the randomly in the sample and the smaller number of training images lead to a performance loss.

TABLE IV
TRADITIONAL METRICS - K-FOLD (K=5) - YOLO

Fold	Accuracy	Precision	Recall	F1-score
k = 1	0.83	0.65	0.79	0.72
k = 2	0.96	0.88	0.93	0.91
k = 3	0.94	0.87	0.93	0.90
k = 4	0.95	0.89	0.93	0.91
k = 5	0.85	0.65	0.77	0.71
mean	**0.91**	**0.79**	**0.87**	**0.83**

C. Video Results

YOLO v4 took about 18 hours to train, and fast YOLO only 5 hours in the previously mentioned computer configuration. At the end of training, the YOLO accuracy was 92%, while the fast YOLO accuracy was 59%, a significant difference between the two CNN architectures.

TABLE V
TRADITIONAL METRICS - K-FOLD (K=5) - FASTYOLO

Fold	Accuracy	Precision	Recall	F1-score
k = 1	0.51	0.51	0.47	0.51
k = 2	0.67	0.77	0.61	0.68
k = 3	0.70	0.81	0.63	0.71
k = 4	0.69	0.78	0.65	0.71
k = 5	0.47	0.51	0.49	0.50
mean	0.59	0.68	0.57	0.62

Then, the trained YOLO architectures using the ratio 90/10 were tested in two different video conditions using OpenCV. In the first one, the video changes the filming angle, making the detection task for the CNNs challenging. Figure 4 shows the first and last frames, in which we can realize that the accuracy remains stable with some false positives and overlap detection in some ship hatches. Moreover, the small hatches tend to present false positives, especially the closed ones, producing a lower accuracy than the large ones, which is a YOLO's limitation, according to Redmond et al. [3]. Also, the movement caused double detection in some hatches, as seen in the bottom picture, but the case is easy to treat.

However, even though the YOLO V4 presents promising results, the prediction time becomes an issue because to predict all objects in the video, the CNN takes 0.35 seconds. Such a time reduces the frame rate to 2.86 frames per second, which is far from acceptable. The same experiment was conducted using the trained fast YOLO weights to solve the low frame rate issue. As a result, the fast YOLO increased the frame rate to 25 frames per second, making it a success in real-time video recognition. On the other hand, individual object detection accuracy decays considerably, as shown in Figure 5. Moreover,

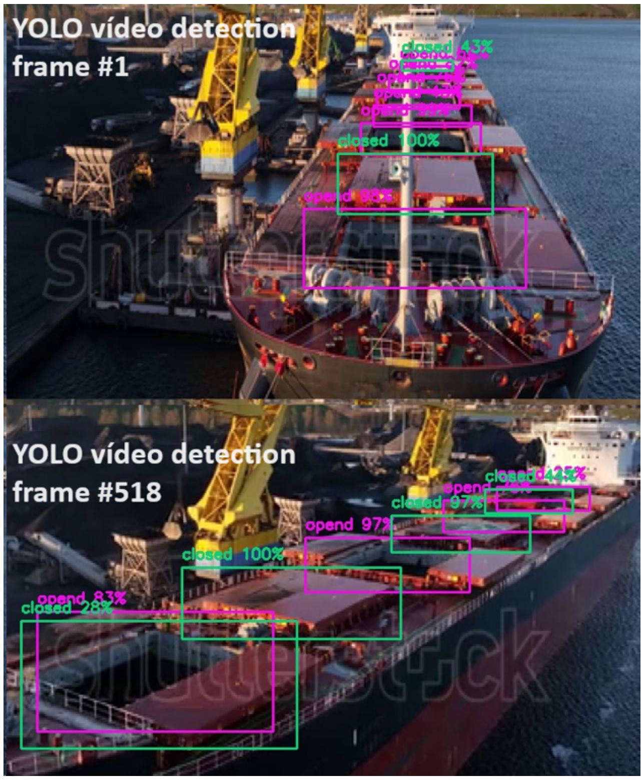

Fig. 4. YOLO vs Fast Yolo: the upper photo is frame number 1 and the bottom photo is last frame number 518

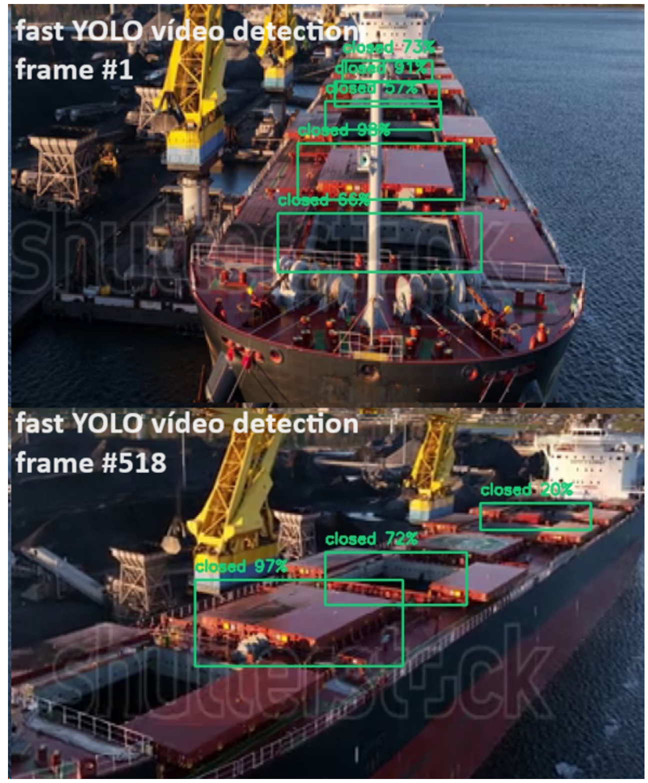

Fig. 5. Same video using Fast YOLO

in the last frame, the number of missed classifications is higher and presents double classification.

The second video is at night with bulk loading condition as seen in Figure 6, in which we can see that the best performance of YOLO at night in which all cargo holds were properly detected. Nonetheless, the individual accuracy is higher in spotlights. On the other hand, Fast YOLO just recognizes the cargo hold in good light conditions.

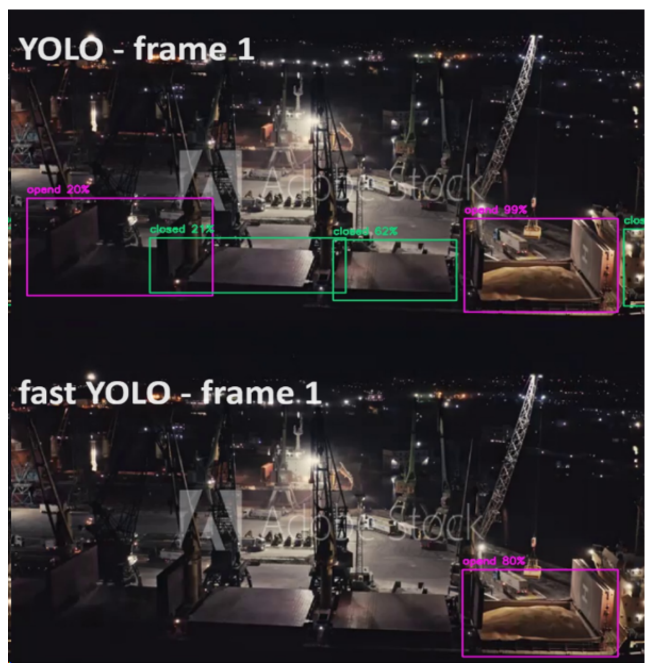

Fig. 6. YOLO fast YOLO detection at night and conditions of bulk loading

V. Conclusions

In this investigation, we present the performance of YOLO v4 and Fast YOLO in a pre-processed dataset, and the results were applied in video experiments. In the dataset, YOLO v4 reached much better metrics than Fast YOLO, in which the regular YOLO reached an mAP of 92%, an accuracy of 92%, precision of 81%, recall of 90%, and F1-Score of 80% outperforming Fast YOLO in 90/10 ratio experiment. A k-fold with $k = 5$ presents similar results in the sense of outperforming the FastYOLO with a mean of 91% in accuracy, 79% in precision, 87% in recall, and 83% in the F1-score.

Then, applying the obtained models in the 90/10 experiment, the regular YOLO presented the best accuracy in detecting the hatch in videos. On the other hand, fast YOLO in real-time video outperformed expectations with a prediction average time of 0.04 seconds versus 0.35 seconds in YOLO. Thus, Table VI summarizes the main findings of this investigation.

Future work includes: (i) Improving the small YOLO architecture with a large dataset to increase the accuracy, (ii) Performing the same experiments on YOLO's latest version, (iii) Increasing the number of images in different angles and

TABLE VI
FINDINGS IN YOLO AND FAST YOLO

Results	YOLO	Fast YOLO
Detection accuracy and stability	X	
Multiple cargo hold detection	X	X
Camera and angle rotation detection	X	
Night and bulk loading operations detection	X	
Static video image detection	X	X
Fast detection time		X
Best frame rate in real-time video		X

night light conditions, and, (iv) Developing a system to treat false positives and detect the transition of cargo hold status like opened to closed and vice-versa.

ACKNOWLEDGMENT

We thank the support from the Ministry of Science, Technology, and Innovation (MCTI) through the resources of Law No. 8,248, dated October 23, 1991, within the scope of PPI-Softex, coordinated by the Association for the Promotion of Brazilian Software Excellence (Softex), and published as TIC Residency 09 (process 01245.005714/2022-18), conducted by the Federal Institute of Maranhão (IFMA).

REFERENCES

[1] Li, Y.; Li, Z.; Yang, Y.; Zhao, L. and Yang, L., 2023. A fast recognition algorithm of ship hatch in bulk cargo terminal based on point cloud contour extraction. *Measurement and Control*, 56(1-2), pp.228-2.

[2] Alzubaidil, L.; J. Zhang; A. J. Humaidi; A. Al-Dujali; Y. Duan; O. Al-Shamma; J. Santamaria; M. A. Fadhel; M. Al-Amidie e L. Farhan (2021), Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, *Journal of Big Data* 8 (53), pp. 1–74.

[3] Redmon J.; Divvala,S.; Girshick, R and Farhadi, You Only Look Once: Unified, real-time object detection, IEEE CVPR, 2016, Retrieved from:https://arxiv.org/pdf/1506.02640.pdf.

[4] Sharma, M.; MT, M. A.; Ali, N.; Kumar, N.; Gopal, N. and Xavier, N., Single pull macgregor type hatch cover, Project Report, Bachelor of technology, Cochin University of Science and Technology, 2011.

[5] Hyla, P., The crane control systems: A survey, Proceedings of 17th International Conference on Methods and Models in Automation and Robotics (Miedzyzdroje, Poland), 2012, pp. 505-50936.

[6] Kaleci,B. and Turgut, K., Comparison of Deep Learning Techniques For Detection of Doors In Indoor Environments, Eskisehir Osmangazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi, vol.29, no.3, pp.396, 2021.

[7] Ziang, Z.; Liu, X; Ma, M.; Wu, G. and Farrel, J. A., LiDAR-Based Hatch Localization, *Remote Sensing*, 14 (20), 2022, 5069.

[8] Miao, Y.; Li, C.; Li, Z.; Yang, Y. and Yu, X., A novel algorithm of ship structure modeling and target identification based on point cloud for automation in bulk cargo terminals. Measurement and control, 54(3-4), 2021, pp.155–163.

[9] Mi, C.; Shen, Y.; Mi, W. and Huang, Y., Ship identification algorithm based on 3D point cloud for automated ship loaders. *Journal of Coastal Research*, (73), 2015, pp.28-34.

[10] Chen, X.; Yang, Y.; Wang, S.; Wu, H.; Tang, J.; Zhao, J. and Wang, Z., Ship type recognition via a coarse-to-fine cascaded convolution neural network. *The Journal of Navigation*, 73(4), 2020, pp.813-832.

[11] Zhao, R.; Wang, J.; Zheng, X.; Wen, J.; Rao, L. and Zhao, J., 2020. Maritime visible image classification based on double transfer method. IEEE Access, 8, pp.166335-166346.