

PAPER • OPEN ACCESS

## A YOLOv3-based non-helmet-use detection for seafarer safety aboard merchant ships

To cite this article: Mingliang Zhong and Fei Meng 2019 *J. Phys.: Conf. Ser.* **1325** 012096

View the [article online](#) for updates and enhancements.



**IOP | ebooks™**

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

# A YOLOv3-based non-helmet-use detection for seafarer safety aboard merchant ships

Mingliang Zhong<sup>1</sup>, Fei Meng<sup>1\*</sup>

<sup>1</sup> Merchant Marine College, Shanghai Maritime University, Shanghai, 201306, China

\*Corresponding author's e-mail: feimeng@shmtu.edu.cn

**Abstract.** International shipping industry undertaking more than 80% global trade volume, is indispensable to the world's economy and development. Seafarers, which are an important part of shipping, mainly engage in safe navigation and good maintenance of ships. However, they are prone to expose themselves to occupational accidents and injuries, especially fatal trauma on the head due to non-helmet-use behaviour. Wearing a safety helmet can effectively reduce the risks resulting in head injuries. Despite the vital role of safety helmet, many seafarers tend to take off their helmet because of discomfort caused by weight and high temperature in engine room or deck. To improve the ship's safe operation and prevent the loss of life at sea, this paper proposed a state-of-the-art method based on YOLOv3 model for automatic and real-time detection of non-helmet-use behaviour. The experimental results showed that the detection method can run real-time under ship's video surveillance and precisely detect the non-helmet-use behaviour with low miss rate.

## 1. Introduction

Maritime transport is the backbone of international trade and the global economy. Around 80 percent of global trade by volume and over 70 percent of global trade by value are carried by sea and are handled by ports worldwide[1]. And the profession of seafaring is a crucial component of the international shipping industry. Despite seafarers' vital role, seafaring is seen as one of the most dangerous occupations due to its high fatality in maritime perils and occupational accidents[2]. The fatal accident rate in shipping was 21 times that in the general workforce, 4.7 times that in the construction industry[3]. Among occupational accidents, slipping, stumbling, striking against or struck by objects, falling on board or fall from height are the main causes leading to injuries and deaths. In particular, the trauma to the head is an extremely fatal factor due to inadequate and belated medical care aboard. And the head as a crucial part of the body is very fragile and vulnerable to impact or collision. Wearing a safety helmet correctly can effectively reduce the incidence of work-related head trauma by 70% through protecting workers from penetration, hindering harm from direct hard object contact and mitigating electrical shock hazards[4]. Though flag state, shipowners or operators set mandatory regulation for wearing helmet, non-helmet-use (NHU) behaviour of seafarers is still ubiquitous due to ignorance and weak supervision. Thus, an effective and automatic way to detect NHU behaviour for ensuring seafarers' safety is of paramount importance. Considering the widespread use of onboard surveillance video as shown in figure 1 and significant progress in machine vision-based object detection techniques having been made, these make it possible to use existing assets to automatically monitor NHU behaviour. For NHU detection, a number of related works have contributed by researchers, but mainly focusing on construction filed. For example, Park et al. studied a vision-based method to automatically monitor whether worker wears helmet or not with the use of HOG features and SVM classifier in the video frames captured by on-site



construction cameras[5]. Similarly, Rubaiyat et al. proposed to use HOG and SVM, the combination of colour-based Circle Hough Transform (CHT) feature extraction techniques to detect NHU[6]. Though these traditional computer vision-based methods made small progress in NHU detection, it's difficult to apply in practical scenarios due to large variations in multiple targets, poses, occlusions, lighting conditions, noise, vibration and ever-changing backgrounds on deck or engine room. Thanks to the resurgence of the convolutional neural network, a powerful feature extractor, deep learning-based algorithms have shone light on object detection[7]. Thus, this paper proposes YOLOv3[8] algorithm to detect NHU behaviour of seafarer on real-time video surveillance.

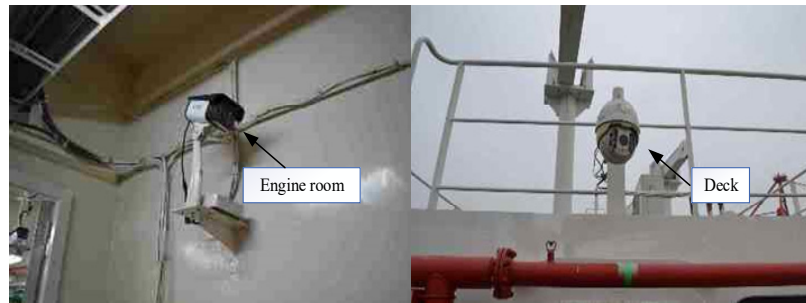


Figure 1. Engine room and deck video surveillance cameras of Shanghai Maritime University's training ship.

## 2. Image data pre-processing

An appropriate dataset was developed to train the YOLOv3 model to detect NHU behaviour since there is no off-the-shelf dataset available. Data pre-processing can be framed into three parts, as shown in figure 2. To get closer to real scenario, the image acquisition was conducted using a camera with high resolution and clipping from different video surveillance in the conditions at different time of the day, view angles, occlusion degree, locations, image taking distance and different weather situation; In order to enhance the diversity and richness of the experimental dataset, the collected images were expanded by augmentation techniques in terms of scaling, flipping, changing brightness, colour balance processing, adding noise and blurring. After image augmentation, the next step was to annotate the images using the graphical image annotation tool LabelImg[9]. Every image localization of NHU seafarers requires ground truth annotations. The annotations used here are XML files in PASCAL VOC[10] format with 4 coordinates representing the location of the bounding box surrounding an object and its label.



Figure 2. Image data pre-processing pipeline

## 3. Methodology

### 3.1. YOLOv3 model

Object detection is a domain that has benefited immensely from the recent developments in deep learning. The YOLOv3 algorithm is evolved from the YOLO and YOLOv2 proposed by the same author, respectively in 2016 and 2017. Compared with the Faster R-CNN network (i.e. region proposal-based ideas), the YOLO reframes object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. Using this system, you only look once (YOLO) at an image to predict what objects are present and where they are. This greatly increases the detection speed compared to Faster R-CNN. But YOLO lags behind state-of-the-art detection system in accuracy, especially for a small object. Then YOLOv2 pools a variety of ideas from past work with novel concepts to significantly improve YOLO's performance, such as batch normalization, high-resolution classifier, convolutional with anchor boxes, dimension clusters, fine-grained features, multi-scale training and a

custom deep architecture Darknet19. However, YOLOv2's architecture was still lacking some of the most important elements that are now staple in most of the newfangled algorithms. No residual blocks, no skip connections and no up-sampling. YOLOv3 incorporates all of these techniques and introduces Darknet53, a more powerful feature extractor as well as multi-scale prediction mechanism. All these techniques make YOLOv3 more effective for detecting small targets, meanwhile, it still runs in real time. The YOLOv3 network architecture is shown in figure 3.

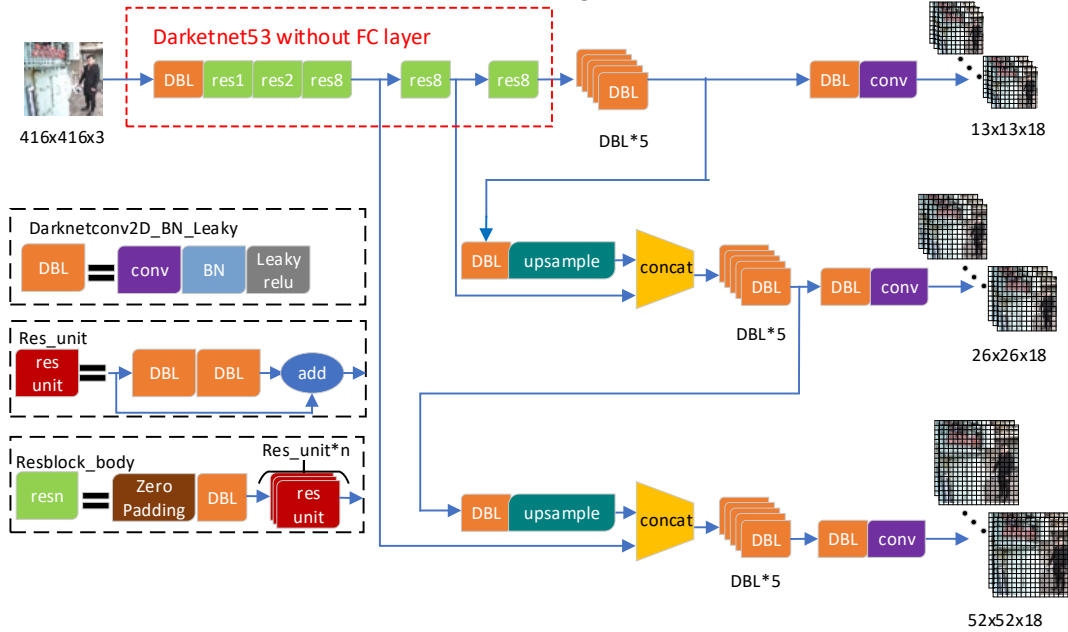


Figure 3. YOLOv3 network architecture diagram inspired by Levio (levio123@163.com)

### 3.2. Loss function

Deep learning discovers intricate structure in large datasets by using the optimization algorithm to optimize objective function and then indicate how a machine should change its internal parameters that are used to representation in each layer from the representation in the previous layer. Similarly, the object detection problem can be cast as the optimization of loss function requiring minimization with respect to parameters. So, the definition of the loss function is of significant core in YOLOv3. During training YOLOv3 optimize the following multi-part loss function:

$$\text{Loss} = \text{Error}_{coord} + \text{Error}_{obj} + \text{Error}_{cls} \quad (1)$$

**Localization loss:** The localization loss measures the errors  $\text{Error}_{coord}$  in the predicted boundary box locations and sizes. YOLOv3 only counts the box responsible for detecting the object.

$$\begin{aligned} \text{Error}_{coord} = & \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \end{aligned} \quad (2)$$

where  $\mathbb{I}_{ij}^{obj} = 1$  denotes that the  $j$ th bounding box predictor in cell  $i$  is responsible for that prediction, otherwise  $\mathbb{I}_{ij}^{obj} = 0$ .  $(x_i, y_i, w_i, h_i)$  are true values of center of the box relative to bounds of the grid cell, width and height relative to the whole image.  $(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i)$  are predicted values.  $\lambda_{coord}=5$ , increases the loss from bounding box coordinate predictions.

**Objectness loss:** The is  $Error_{obj}$  defined to measure the objectness loss of box containing an object or not in two parts. The first part penalizes the objectness score prediction for bounding boxes responsible for predicting objects (the scores for these should ideally be 1), the second part for bounding boxes having no objects (the scores should ideally be zero).

$$Error_{obj} = \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} \left[ \hat{C}_i \log C_i + (1 - \hat{C}_i) \log(1 - C_i) \right] + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} \left[ \hat{C}_i \log C_i + (1 - \hat{C}_i) \log(1 - C_i) \right] \quad (3)$$

where  $\hat{C}_i$  represents the confidence measuring objectness score of the box.  $\lambda_{noobj} = 0.5$  weighs down when detecting background for increasing the model's stability.

**Classification loss:** The  $Error_{cls}$  is defined to measure classification error of C conditional class probabilities. This loss penalizes the class prediction for the bounding box which predicts the objects.

$$Error_{cls} = \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} \left[ \hat{p}_i(c) \log p_i(c) + (1 - \hat{p}_i(c)) \log(1 - p_i(c)) \right] \quad (4)$$

where  $\hat{p}_i(c)$  denotes to the conditional class probability for class c in cell i.

#### 4. Experiment and results

After augmentation and annotation work, the onboard seafarer image dataset was established with 4,644 images. A total of 3,483 images from this dataset were randomly selected to comprise the training set, the remaining 1,161 images constituted the testing set. The NHU detection model was trained and tested on an NVIDIA Titan V GPU, as shown in figure 4.



Figure 4. NVIDIA Titan V GPU

##### 4.1. Training

In the training stage, the input images were adjusted to 416x416 pixels. Taking into account the memory constraints of the GPU, the batch size was set to 4 in this paper. The detection model was trained in 160k steps. The network initialization parameters are shown in table 1. Throughout the training, the Adam optimizer was used and the initial learning rate was set to  $1 \times 10^{-3}$ . Other parameters referred to the original parameters in the YOLOv3 model. In this paper, the learning rate reducing mechanism was used to overcome training plateau. Models often benefit from reducing the learning rate by a factor of 2-10 once learning stagnates. And early stopping techniques were adopted for avoiding model overfitting. The loss of YOLOv3 during training is shown in figure 5.

##### 4.2. Testing metrics and results

In the object detection domain, the related indicators for evaluating the effectiveness of the detection model are the following metrics.



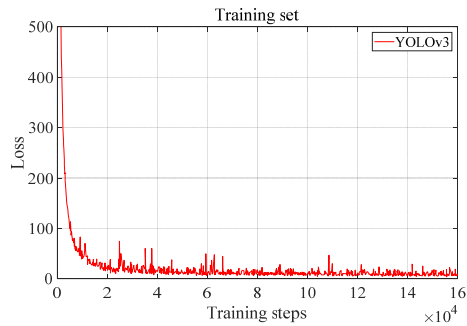


Figure 5. Loss curve of YOLOv3 model.

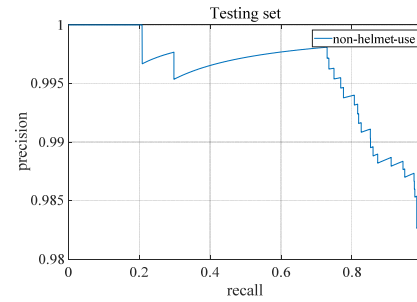


Figure 6. P-R curve of NHU detection.

**4.2.1. Precision, Recall and Miss rate.** Precision is defined as a ration of TP (true positive) to TP+FP (false positive). TP+FP is the numbers of seafarers detected as NHU based on the method. The Recall is the ratio of TP to TP+FN (false negative). TP+FN means the actual number of NHU seafarers of the testing set.

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$Miss\ rate = 1 - Recall = \frac{FN}{TP + FN} \quad (7)$$

**4.2.2. Precision/Recall curve.** The precision/recall curve from the detection method's output in order ranked by decreasing confidence. Precision and recall curve of NHU predicted by the detection model is shown in figure 6.

**4.2.3. Frame per second (FPS).** A real-time detection system should process streaming video in less than 25 milliseconds of latency (i.e. 24fps). Real-time NHU detection and alarming timely are very important because all risks happen in an instant.

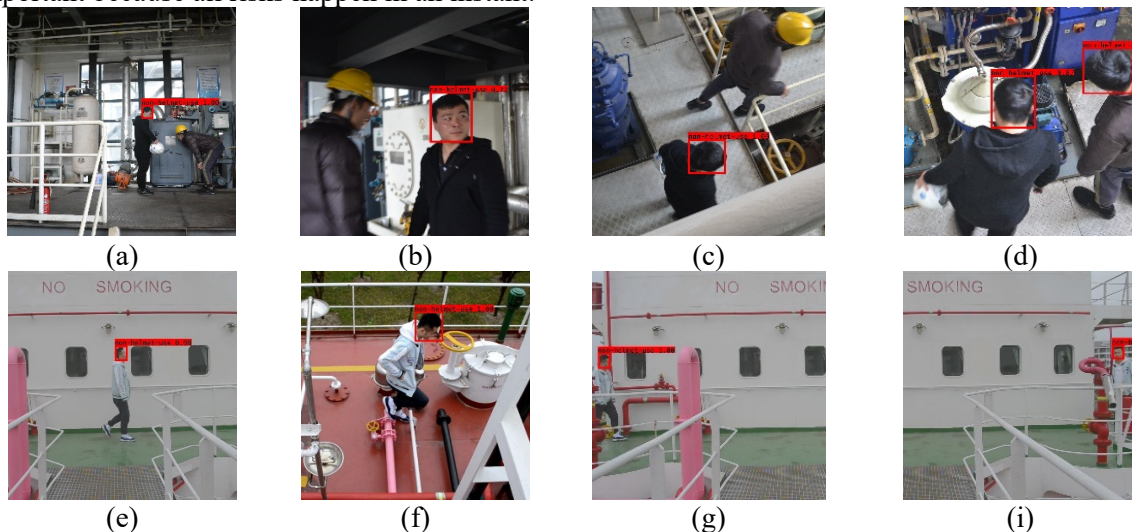


Figure 7. Images testing results: (a-d) testing results in engine room, (e-h) testing results on deck

**4.2.4. Results.** A wide range of collected images were tested in the YOLOv3 model. In the engine room and deck scenarios, detection results are shown in figure 7, respectively. The testing set statistics and the testing results in line with the aforementioned metrics are shown in table 1.

Table 1. Testing set statistics and results under different evaluation metrics

Size of input images	Number of test images	Number of NHU seafarers	TP	FP	FN	Precision (%)	Recall (%)	Miss rate (%)	FPS
416x416	1,161	1,439	1,418	26	21	98.1	98.5	1.5	34

## 5. Conclusion

Seafaring continues to be one of the most dangerous job sectors globally, and at the same is indispensable for global trade. To improve aboard safety, this paper proposes a new detection method based on YOLOv3 for non-helmet-use behaviour of seafarers preventing serious injuries or fatalities from head-related accidents. The experimental results show that the YOLOv3 model can effectively detect NHU behaviour in real time and has a low miss rate. Currently, the detection model runs on the expensive GPU. Therefore, it is recommended that future research focus on a lightweight device and the integration into Ship Monitoring and Alarming System for practical use.

## Acknowledgment

This work was supported by the Shanghai Association for Science and Technology under Grant 16391902702. Images dataset is supported Automatic Engine Room and Wu Song training ship of Shanghai Maritime University.

## References

- [1] United Nations Conference on Trade and Development. (2018) Review of Maritime Transport. <https://unctad.org/en/pages/PublicationWebflyer.aspx?publicationid=2245>
- [2] Zhang, P.F., Shan, D.S., Zhao, M.H., Nicola, P.R. (2019) Navigating seafarer's right to life across the shipping industry. *Marine Policy*, 99: 80-86.
- [3] Roberts, S.E., Nielsen, D., Kotlowski, A., Jaremin, B. (2014) Fatal accidents and injuries among merchant seafarers worldwide. *Occupational Medicine*, 64: 259-266.
- [4] Zhang, Z.W. (2009) Research in comfort design of safety helmet. Ph.D. thesis, College of Art, Graduate School of the Central South University, China.
- [5] Park, M.W., Elsafty, N., Zhu, Z.H. (2015) Hardhat-wearing detection for enhancing on-site safety of construction workers. *J. Constr. Eng. Manage.*, 141(9): 04015024.
- [6] Rubaiyat, A.H., Toma, T.T., Kalantari-Khandani, M., Rahman, S.A., Chen, L., Ye, Y., Pan, C.S. (2016) Automatic Detection of Helmet Uses for Construction Safety. In: 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops. Nebraska. pp. 135–142.
- [7] Yann, L., Bengio, Y., Hinton, G.E. (2015) Deep learning. *Nature*, 521(7553): 436-444.
- [8] Redmon, J., Farhadi, A. (2018) YOLOv3: An Incremental Improvement. <https://arxiv.org/abs/1804.02767>
- [9] Tzutalin. (2015) LabelImg. <https://github.com/tzutalin/labelImg>
- [10] Everingham, M., Gool, L.V., Christopher, K.L., Williams, J.W., Zisserman, A. (2010) The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88:303-308.