Ocean Systems Engineering, *Vol. 10, No. 4 (2020) 415-434*
DOI: https://doi.org/10.12989/ose.2020.10.4.415

Image-based ship detection using deep learning

Sung-Jun Lee[1a], Myung-Il Roh[*2] and Min-Jae Oh[3b]

[1]*Department of Naval Architecture and Ocean Engineering, Seoul National University, Republic of Korea*
[2]*Department of Naval Architecture and Ocean Engineering, and Research Institute of Marine Systems Engineering, Seoul National University, Seoul, Republic of Korea*
[3]*School of Naval Architecture and Ocean Engineering, University of Ulsan, Ulsan, Republic of Korea*

(Received May 14, 2020, Revised October 12, 2020, Accepted December 10, 2020)

Abstract. Detecting objects is important for the safe operation of ships, and enables collision avoidance, risk detection, and autonomous sailing. This study proposes a ship detection method from images and videos taken at sea using one of the state-of-the-art deep neural network-based object detection algorithms. A deep learning model is trained using a public maritime dataset, and results show it can detect all types of floating objects and classify them into ten specific classes that include a ship, speedboat, and buoy. The proposed deep learning model is compared to a universal trained model that detects and classifies objects into general classes, such as a person, dog, car, and boat, and results show that the proposed model outperforms the other in the detection of maritime objects. Different deep neural network structures are then compared to obtain the best detection performance. The proposed model also shows a real-time detection speed of approximately 30 frames per second. Hence, it is expected that the proposed model can be used to detect maritime objects and reduce risks while at sea.

Keywords: object detection; ship detection; deep neural network; deep learning; maritime dataset

1. Introduction

Both aerial and ground-based unmanned vehicles are being developed for use in military, maritime, research, and civil fields. One of the key technologies required in unmanned vehicle applications is visual recognition, as autonomous vehicles need to be able to detect adjacent objects clearly and rapidly to avoid obstacles, follow paths or targets, read signs, and interact with them. A typical modern unmanned vehicle uses not only video cameras (also known as EO: electro-optical sensors) but also many different sensors simultaneously, such as radar, sonar, LiDAR (light detection and ranging), and GPS. It also utilizes sensor fusion techniques that combine sensory data from the various source types to improve its ability to detect the presence of obstacles around it. At the final stage of recognition, however, it continues to have a high dependence on the use of video cameras to distinguish the precise type and shape of adjacent objects (Hermann *et al*. 2015). From a maritime perspective, although AIS (automatic identification system) can provide detailed information about

∗Corresponding author, Professor, E-mail: miroh@snu.ac.kr
[a] Ph.D. Student, E-mail: ship99@snu.ac.kr
[b] Assistant Professor, E-mail: minjaeoh@ulsan.ac.kr

http://www.techno-press.org/?journal=ose&subpage=7 ISSN: 2093-6702 (Print), 2093-677X (Online)

nearby ships, it is unable to detect objects without AIS transmitters, such as buoys, small boats, and kayaks. Thus, an image-based recognition that employs video cameras is extremely important, despite the use of other available sensors. In this respect, extensive research on image detection and classification are actively being conducted in line with the development of deep learning technology.

In this study, we introduce a state-of-the-art real-time object detection method that utilizes a convolutional neural network (CNN) to perform image-based ship detection. We firstly use one of the existing universally trained models as a "reference model" to examine its applicability in the maritime domain by testing how well it can detect ships from maritime images. We then present the proposed model, which has variations made to the network structures of the reference model, and train it based on a maritime domain-specific dataset. To evaluate the model, we compare how well the detection performance improves compared to the reference model in terms of recall and IOU (intersection over union) metrics. We also apply the proposed model to a video to confirm the applicability of real-time object detection at sea.

2. Related works

Visual recognition and object detection problems have been extensively studied by computer vision scientists for many decades, but recent CNN-based deep learning technologies are currently outperforming all existing algorithms (Huang *et al*. 2016). R-CNN (Regions with CNN) have applied CNN to the object detection problem and have achieved 30% better precision than previous best results (Girshick *et al.* 2014). The faster R-CNN not only further increases precision but also boosts the detection speed to a semi-real-time level of approximately five frames per second (Ren *et al.* 2015).

R-CNN series algorithms conduct the proposal and classification of regions separately, whereas single-stage detection algorithms can perform all object detection processes in one stage. The SSD: Single Shot MultiBox Detector (Liu *et al.* 2016) and YOLO: You Only Look Once (Redmon *et al.* 2016) are representative algorithms. They achieve a high speed of up to 45 frames per second (fps), although there is a slight trade-off with precision to some extent. However, the YOLO v2, which arrived in 2017, has an even faster speed, and it provides a higher precision than existing algorithms without any trade-off performances (Redmon and Farhadi 2017). Therefore, the YOLO v2 algorithm and its network model are chosen for use in ship detection in this study.

In the maritime domain, image-based ship detection has been the focus of many studies. For example, Lee *et al.* (2016) detected ships at sea using the Viola & Jones algorithm. Originally designed for face detection in 2011, this algorithm is fast in real-time, but is not as accurate as CNN-based models. Zhang *et al.* (2016) conducted ship detection within satellite images using a combination of image processing techniques and CNN classification, which provides a result similar to the R-CNN scheme. However, this method has a slower processing speed than the R-CNN, and although it is acceptable for use with satellite image analysis, it is not suitable for application to an unmanned surface vehicle that requires a real-time detection operation. Cuong *et al.* (2015) performed ship classification with a CNN model that takes single-ship-containing images as the input and classifies them into 35 ship classes. Although this image classification for a single ship can be performed as fast as the real-time processing by CNN, it cannot detect multiple objects within an image. Image classification (for a single object) and object detection (for multiple objects + bounding boxes) are treated as different problems in the computer vision field. In this study, we

propose an accurate and real-time fast ship detection model that uses a single-stage detection algorithm with CNN and the transfer learning method.

3. Image-based ship detection

3.1 Dataset for ship detection

The major difference between the reference model and the proposed model is the datasets used in training. The reference model is trained using one of the universal image datasets that are known as the PASCAL VOC (Pattern Analysis, Statistical modeling and Computational Learning Visual Object Classes) dataset (Everingham *et al*. 2015), as shown in Fig. 1. The dataset contains approximately 21,000 images, bounding-box information, and classifying labels for 20 general classes, such as a person, dog, cat, cow, table, car, truck, bus, and boat. The fully trained weight data of the CNN model based on this PASCAL VOC dataset is published online without any modifications.

In contrast, the proposed model uses a public dataset available for the maritime industry known as the Singapore Maritime Dataset (SMD) (Prasad *et al.* 2017). The dataset employs approximately 63 videos taken at sea during both the day and night and thus reflects the actual ocean environment. The dataset contains ground truth labels for every frame of each video that comprises object classes and bounding-box information for each object shown in every frame. Only floating objects that can be observed in the maritime environment are subject to classification, and they are divided into ten classes: ferry, buoy, vessel/ship, speedboat, boat, kayak, sailboat, swimming person, flying bird/plane, and other. Although a flying bird/plane is not a floating object, it is included because it can be seen in the maritime environment and also needs to be distinguished from floating objects. Fig. 2 shows sample images captured from a video in the dataset.

Fig. 1 Sample images from the PASCAL VOC dataset (Everingham *et al*. 2015)

Fig. 2 Sample snapshot images from the Singapore Maritime Dataset video (Prasad *et al*. 2017)

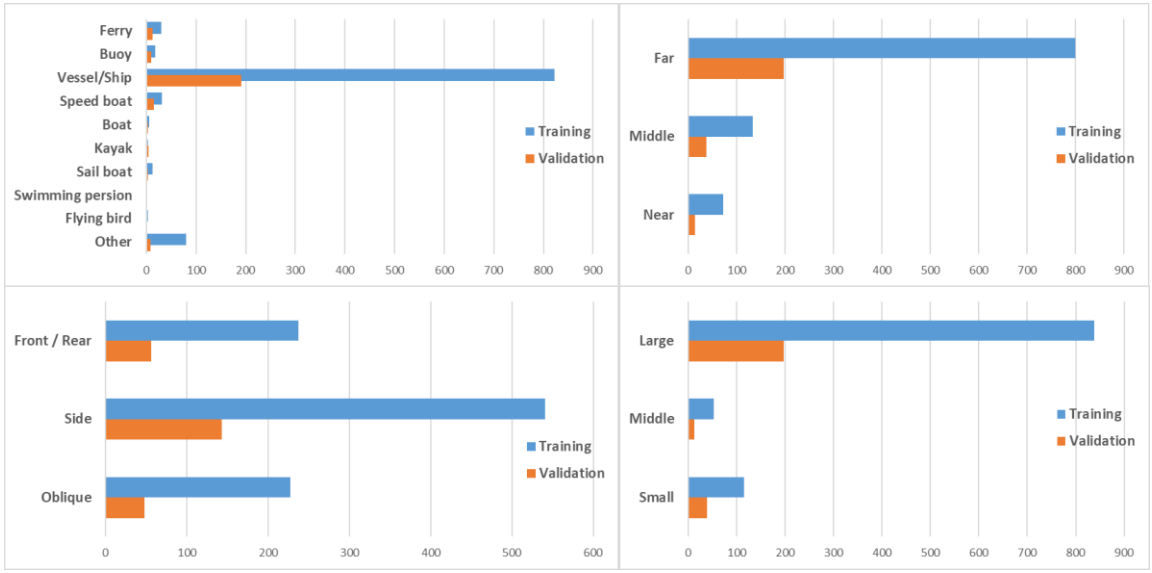

Fig. 3 Data distribution of the labeled objects in training (blue) and validation (orange) images with respect to class (top-left), distance (top-right), orientation (bottom-left), and size (bottom-right)

To train the CNN model for ship detection, 189 images were extracted from the SMD videos; 159 images were then used as training data, and the remaining 30 images were used as validation data. Although it was possible to obtain more images from the videos, all videos contained similar objects. To avoid an overfitting problem in learning, we selected a smaller number of images. There is an average number of seven objects in each image. Training and validation images contain 1,252 labeled objects in total. Data distribution of the objects in training and validation images with respect to class, distance, orientation, and physical size is shown in Table 1 and Fig. 3.

3.2 Neural network models and structure

The YOLO v2 algorithm (Redmon and Farhadi 2017) was selected to provide image-based ship detection and classification. The core algorithm of YOLO v2 is based on the first version of YOLO (Redmon *et al*. 2016), but it adopts several additional techniques that improve its detection speed

Table 1 Data distribution of the labeled objects in training and validation images

Item	Training	Validation	Item	Training	Validation
Class			**Distance**		
Ferry	30	12	Far (d > 500 m)	800	197
Buoy	18	9	Middle (250 < d ≤ 500 m)	133	37
Vessel/Ship	822	191	Near (d ≤ 250 m)	72	13
Speed boat	31	15	**Orientation**		
Boat	6	3	Front/Rear	237	56
Kayak	3	4	Side	541	143
Sailboat	12	3	Oblique	227	48
Swimming person	0	0	**Size**		
Flying bird	3	2	Large (L > 40 m)	838	197
Other	80	8	Middle (10 < L ≤ 40 m)	52	12
Total	1005	247	Small (L ≤ 10 m)	115	38

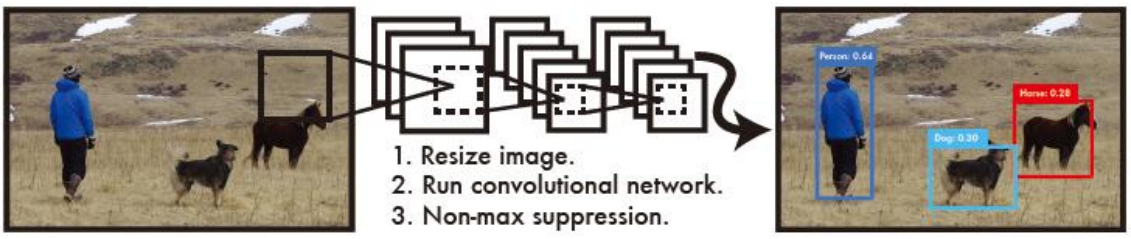

Fig. 4 YOLO detection system (Redmon *et al.* 2016)

and precision. Fig. 4 shows the keystream involved in the YOLO object detection process, where the system (1) resizes the input image to the pre-specified size, (2) runs a single convolutional network on the image, and (3) eliminates duplicate bounding boxes for the same object using the boxes' confidence scores. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities of objects in the boxes.

The CNN models used in this study are presented in Figs. 5 and 6. The basic structure, including the number of convolutions and pooling layers, is based on the CNN model proposed by Redmon and Farhadi (2017). We gave the reference model and the proposed model (Model #3) the same network structure to enable a comparison between the two models, but the different datasets were used for training. We also developed two network models for use in comparing the detection performance of the proposed model. The first model (shown in Fig. 5) is a simple CNN model that consists of 22 convolutional layers and 5 pooling layers, and was used for model #1. The neural network model in Fig. 6 was used for the reference model, model #2, and #3, and the networks consist of 23 convolutional layers and 5 pooling layers. Convolution layers with 1 x 1 convolution

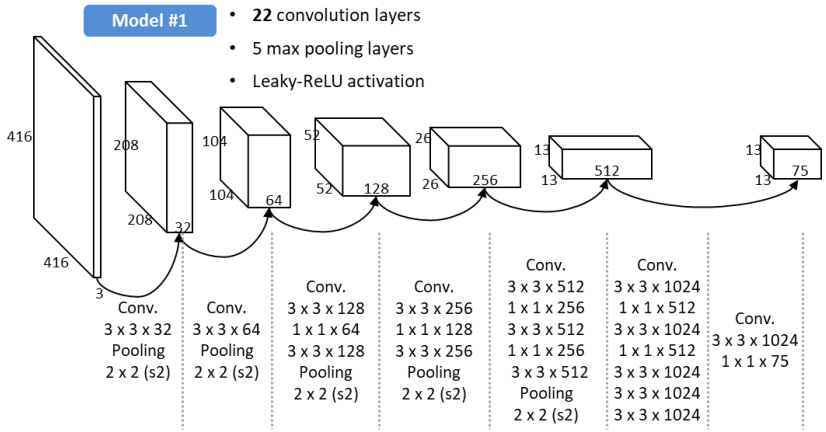

Fig. 5 Convolutional neural network (CNN) model structure tailored for ship detection

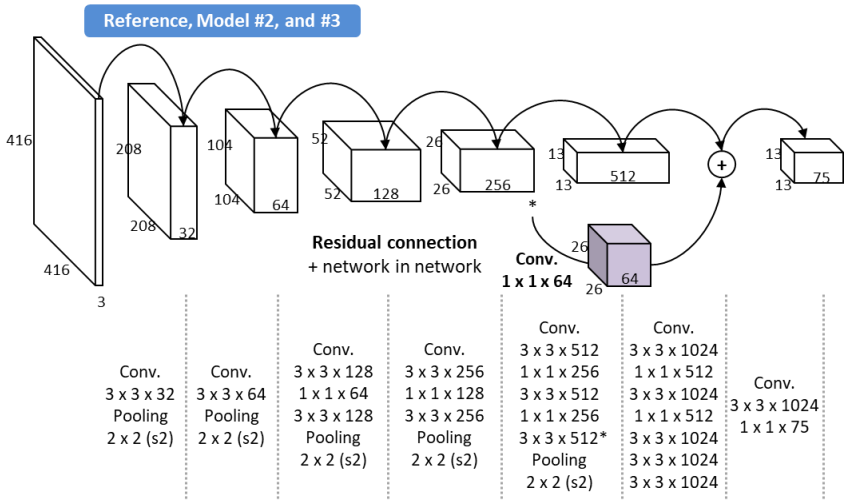

Fig. 6 Convolutional neural network (CNN) with passthrough layer tailored for ship detection

filters, which are the concept of 'network in network' (Lin *et al*. 2013), were added between some of the 3 x 3 convolution layers to provide better feature extraction. A passthrough layer was added to the 13th convolutional layer to enhance performance for small scale objects. The leaky ReLU activation function (Mass *et al*. 2013) was used in all neural network models.

To classify the detected objects into ten classes, the number of filters in the proposed model's last layer was changed from 125 to 75, which equals 5x(4+1+10). The first five correspond to the number of bounding boxes for each cell; 4 and 1 correspond to each bounding box's coordinate information (x, y, w, and h) and confidence score, respectively; and the last 10 correspond to the number of classes. The only difference between the reference and proposed model relates to the model structure:

the proposed model takes 416 x 416 resized input images and makes 13 x 13 x 75 tensor all the way through the network; it then finally performs the classification and bounding box regression (i.e., the object detection).

Table 2 Configuration of models used in training

Item	Reference	Model #1	Model #2	Model #3
Dataset for training	General images (PASCAL VOC: 21,493 images)	Maritime images (SMD training data: 159 images)	Maritime images (SMD training data: 159 images)	Maritime images (SMD training data: 159 images)
Model structure	23 convolutional layers, including passthrough layer	22 convolutional layers (no passthrough layer)	23 convolutional layers, including passthrough layer	23 convolutional layers, including passthrough layer
Transfer learning	Yes (ImageNet ➡ VOC)	No	No	Yes (ImageNet ➡ SMD)

Our detection algorithm and neural network models were implemented using Google's TensorFlow library in Python. It was also inspired by many open-source object detection projects, including the original YOLO source code (in C/C++) published by Redmon *et al.* (2017).

3.3 Transfer learning

In this study, as only 159 images were used to train the neural network, the transfer learning algorithm was adopted (Pan and Yang 2010), as the number of images was not sufficient for training the network (and insufficient data can cause overfitting problems).

Fig. 7 shows the general difference between the use of the transfer learning method and training from scratch. Transfer learning provides a shorter training time and uses a smaller dataset; however, the accuracy can be lower because of the size of the dataset. Prior to training the neural network using the main dataset, pre-training was thus conducted using the ImageNet classification dataset (ImageNet, 2019), which contains more than one million images and can be obtained publicly. The neural network was then initialized using ImageNet by employing the weight values pre-trained, and the SMD was finally used to train the weight values. To evaluate the efficiency of the transfer learning method, models #1 and #2 were trained only with SMD, and the reference model and model #3 were pre-trained firstly with ImageNet and then trained with PASCAL VOC and SMD, respectively. The neural network models used in this study are summarized in Table 2.

3.4 Loss function

The loss function is defined as Eq. (1) and is based on YOLO v2 (Redmon and Farhadi 2017) with slight modifications

$$
\begin{aligned}
&Loss = Loss^{bb} + Loss^{c} + Loss^{p},\\
&Loss^{bb} = \frac{\lambda_{coord}}{N_{obj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{i,j}^{obj} \left[\begin{array}{l} \left(x_{i,j} - \hat{x}_{i,j}\right)^2 + \left(y_{i,j} - \hat{y}_{i,j}\right)^2 \\ + \left(\sqrt{w_{i,j}} - \sqrt{\hat{w}_{i,j}}\right)^2 + \left(\sqrt{h_{i,j}} - \sqrt{\hat{h}_{i,j}}\right)^2 \end{array} \right],\\
&Loss^{c} = \frac{\lambda_{obj}}{N_{conf}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{i,j}^{obj} \left(IOU_{prediction_{i,j}}^{ground\ truth_{i,j}} - \hat{C}_{i,j} \right)^2\\
&+ \frac{\lambda_{noobj}}{N_{conf}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{i,j}^{noobj} \left(0 - \hat{C}_{i,j}\right)^2,\\
&Loss^{p} = \frac{\lambda_{class}}{N_{obj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{i,j}^{noobj} \sum_{c \in class} -p_{i,j}^{c} \log(\hat{p}_{i,j}^{c})
\end{aligned}
\tag{1}
$$

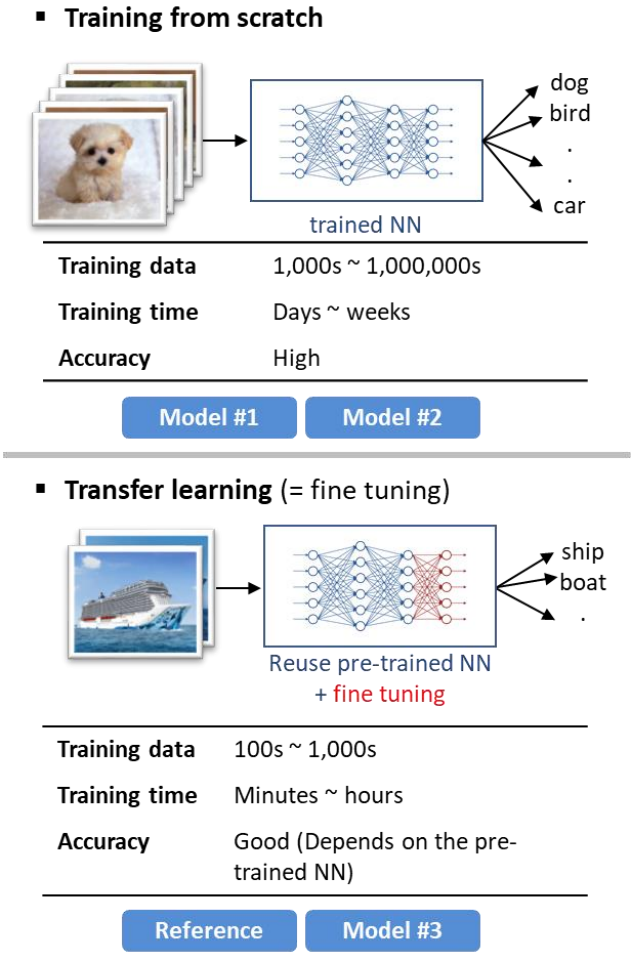

Fig. 7 Comparison between scratch and transfer learning training

where

$$N_{obj} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{i,j}^{obj},$$

$$N_{conf} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \left[\mathbf{1}_{i,j}^{obj} + \mathbf{1}_{i,j}^{noobj} \left(1 - \mathbf{1}_{i,j}^{obj} \right) \right],$$

$$ground\ truth_{i,j} = \left(x_{i,j},\ y_{i,j},\ w_{i,j},\ h_{i,j} \right),$$

$$prediction_{i,j} = \left(\hat{x}_{i,j},\ \hat{y}_{i,j},\ \hat{w}_{i,j},\ \hat{h}_{i,j} \right),$$

$$\mathbf{1}_{i,j}^{obj} = \begin{cases} 1 & \text{if } C_{i,j} = 1 \\ 0 & \text{else} \end{cases},$$

$$\mathbf{1}_{i,j}^{noobj} = \begin{cases} 1 & \text{if } \max_{i',j'} IOU_{prediction_{i,j}}^{ground\ truth_{i',j'}} < 0.6 \text{ and } C_{i,j} = 0 \\ 0 & \text{else} \end{cases}.$$

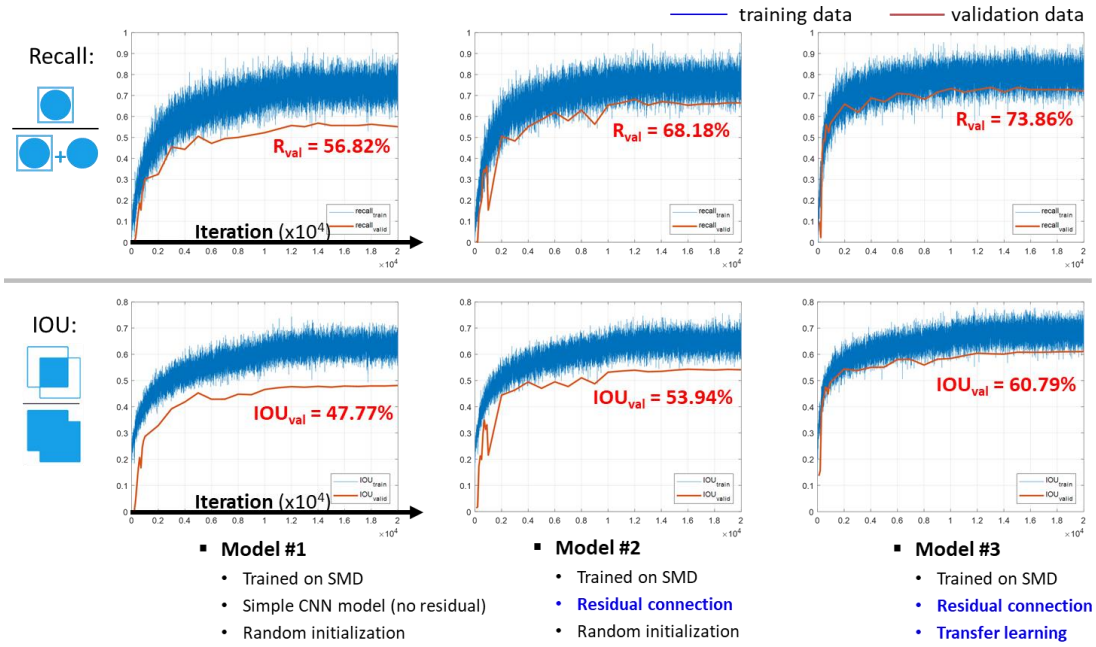

Fig. 8 Performance metrics of proposed neural network models

In this study, we selected $\lambda_{coord} = 1.0$, $\lambda_{obj} = 5.0$, $\lambda_{noobj} = 1.0$, and $\lambda_{class} = 1.0$. λ_{coord}, λ_{obj}, λ_{noobj}, and λ_{class} are the weight factors for the position and size of the detected bounding box, existence of an object, non-existence of an object, and classification of the object, respectively. $Loss^{bb}$ is the error value related to the accuracy of the bounding box of the detected objects; $Loss^{c}$ is the confidence level relating to whether an object is located in the bounding box or not; $Loss^{p}$ is the probability that shows whether the detected object is classified correctly; $Loss^{bb}$ and $Loss^{c}$ are mean square errors; and $Loss^{p}$ is the cross-entropy error. The loss function was applied to all proposed models.

3.5 Training network model with a dataset

The training of the proposed network model was conducted using the maritime dataset with 20,000 iterations of 32 mini-batch images, making a total of 640,000 image operations over a period of approximately 12 h using the GPU of NVIDIA GeForce GTX 1080. Two performance metrics of the network model are shown along the mini-batch iteration axis in Fig. 8. The recall scores represent the number of objects detected out of all true objects (upper), and the IOU (intersection over union) scores represent the level of accuracy with respect to bounding box predictions (lower). In each plot, the blue and red lines represent scores for the training and validation data, respectively. The model reached a recall of 73.86% and an IOU of 60.79% for the validation data at the converging point. All the performance metrics in Fig. 8 show that the proposed model converges around 15,000 iterations of training, which is approximately 3,000 epochs.

Table 3 Performance comparison between the reference and proposed models

Metric	Reference	Model #1	Model #2	Model #3
Recall	33.05 %	56.82 %	68.18%	73.86 %
IOU	29.65 %	47.77%	53.94%	60.79 %
Processing time	14 ms	12 ms	14 ms	14 ms

4. Applications

4.1 Performance comparison between models

The performances of the detection models were evaluated by measuring recall and IOU scores on the validation data described in Table 1. Recall and IOU metrics are defined by Eq. (2), where TP_c is the number of true positive predictions for class c; FN_c is the number of false negative objects in class c; GT_c is the number of ground truth objects in class c; $Intersection_{bb}$ and $Union_{bb}$ are the area of intersection and union between ground truth and predicted bounding boxes respectively. To present the recall score throughout the classes as a single value, we define an overall recall score as Eq. (3). All the recall scores mentioned in this paper, if not specified, mean the overall recall score.

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} = \frac{\text{TP}_c}{\text{GT}_c}, \qquad \text{IOU}_{bb} = \frac{\text{Intersection}_{bb}}{\text{Union}_{bb}} \tag{2}$$

$$\text{Recall}_{\text{Overall}} = \frac{\sum_{c \in class} \text{TP}_c}{\sum_{c \in class} \text{GT}_c} \tag{3}$$

A comparison between the reference and proposed models is summarized in Table 3. Results show that the proposed model can detect ships and floating objects with double the accuracy of the reference model, as evidenced by both the IOU and recall scores.

4.2 Object detection performance of the reference model

The reference model trained by the PASCAL VOC dataset was tested, and a sample test image is shown in Fig. 9. It was not always possible for this model to detect small ships or ships in the distance, which accordingly decreased its recall score. However, in tests using videos, the model was better able to detect ships approaching it when they occupied a larger proportion of the frame, thus delivering a higher confidence score. For this reason, although the overall performance scores of the reference model were relatively lower than those of the proposed one, the reference model is deemed acceptable for use in detecting nearby ships. Nevertheless, this universal (general-purpose) reference model has two major drawbacks in addition to its performance measurements, and these are as follows: first, there is only one class representing floating vessels in the PASCAL VOC dataset, and therefore, all types of ships, yachts, boats, and kayaks are classified simply as ‘boat’; and second, other types of floating objects, such as buoys, cannot be detected because there is no class that represents them.

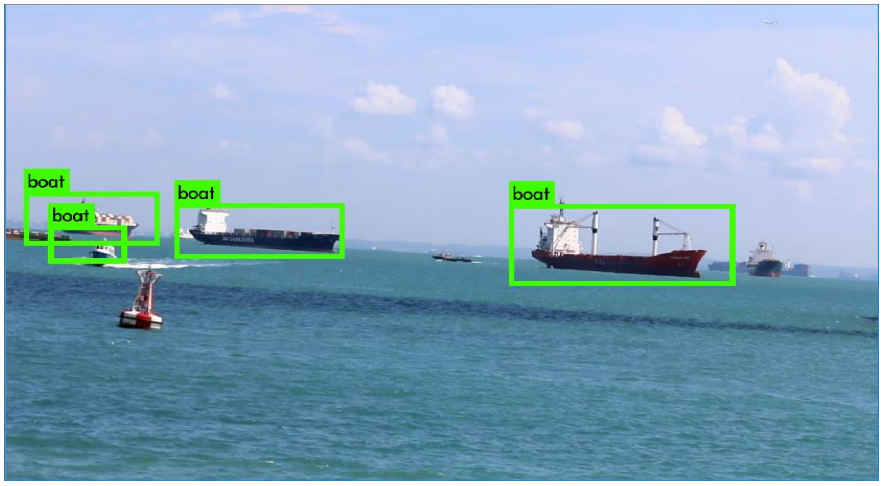

Fig. 9 (Reference model) Ship detection example by the reference model trained using PASCAL VOC dataset

4.3 Object detection performance of the proposed model

The proposed model can overcome these problems because it was trained using the maritime domain-specific dataset and classifies floating objects into ten classes, including ferry, vessel/ship, speedboat, and boat. It can also detect buoys and other types of floating objects that the reference model cannot. Fig. 10 shows the same sample image as Fig. 9, but with the detection test results of the proposed model. The proposed model shows that it is better able to detect small ships and ships in the distance than the reference model, which contributes to an increase in the number of ships detected and, consequently, a higher performance score. Furthermore, the bounding box prediction is more precise, which also contributes to the high IOU score.

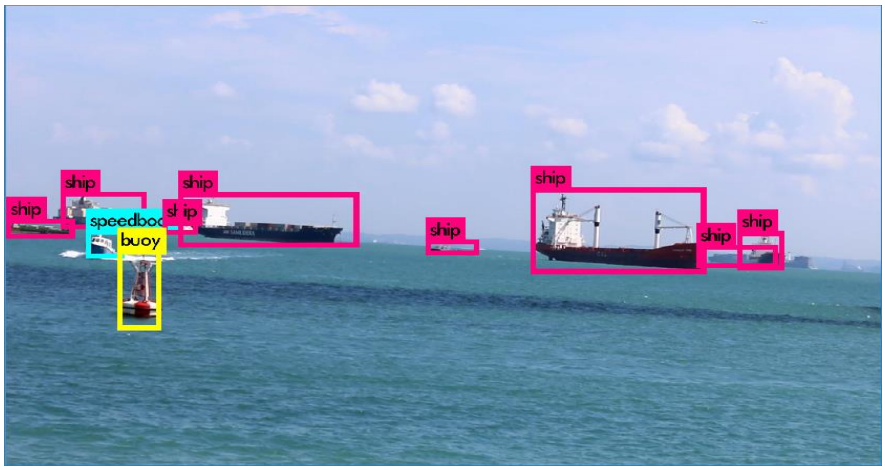

Fig. 10 (Model #3) Ship detection example by the proposed model trained using SMD

Table 4 (Model #3) Performance evaluation for each class on validation images

Class	# of Objects	Recall	Precision	AP @IOU ≥ 0.5
Ferry	**12**	**0.58**	**0.58**	**0.5411**
Buoy	9	0.33	0.17	0.2500
Vessel/Ship	**191**	**0.81**	**0.71**	**0.7722**
Speed boat	**15**	**0.33**	**0.71**	**0.3222**
Boat	3	1.00	1.00	1.0
Kayak	4	0.00	0.00	0.0
Sailboat	3	1.00	1.00	1.0
Swimming person	0	-	-	-
Flying bird	2	0.00	0.00	0.0
Other	8	0.75	0.30	0.6643
Weighted Mean		**0.73**	**0.66**	**0.6979**

The performance evaluation result of the final proposed model (model #3) was presented as a set of recall, precision, and average precision score per class in Table 4. The recall score for each class was evaluated according to the Eq. (2); and the precision score for each class was calculated by the Eq. (4), where TP_c and FP_c are the number of true positive and false positive predictions for class c, respectively.

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \tag{4}$$

Average precision or AP for each class is equivalent to the area under the precision-recall curve of that class in Fig. 11. Precision-recall curves for some classes are shown in Fig. 12. A predicted detection box was considered true positive or TP when its classification was correct, and the IOU score with a ground truth box was greater than the IOU threshold, which wais set to 0.5 in this paper.

AP scores for 'boat', 'kayak', 'sailboat' and 'flying bird' classes are obtained by testing over very limited numbers of samples (less than 5), and it is difficult to believe that their AP scores are properly representing the model's performance over those classes. For swimming person class, no AP can be evaluated at all, as there are no labeled objects in both the training and validation data.

Excluding those outliers, the AP for the 'vessel/ship' class is the highest at 0.77, followed by 'other' with 0.66, 'ferry' with 0.54, 'speed boat' with 0.32, and 'buoy' with 0.25. It can be assumed that those AP scores which are significantly lower than that of 'vessel/ship' are partially due to the severe class imbalance in the training data (see Table 1). The training images have only a few numbers (mostly less than 40) of objects for every class but 'vessel/ship', which is less than 5% of 'vessel/ship' objects.

Nevertheless, although the numbers of 'ferry' and 'speed boat' objects are almost the same in the training images, the detection performances on the two classes show a huge difference. Recall and AP scores of 'ferry' are approximately 70% higher than those of 'speed boat.

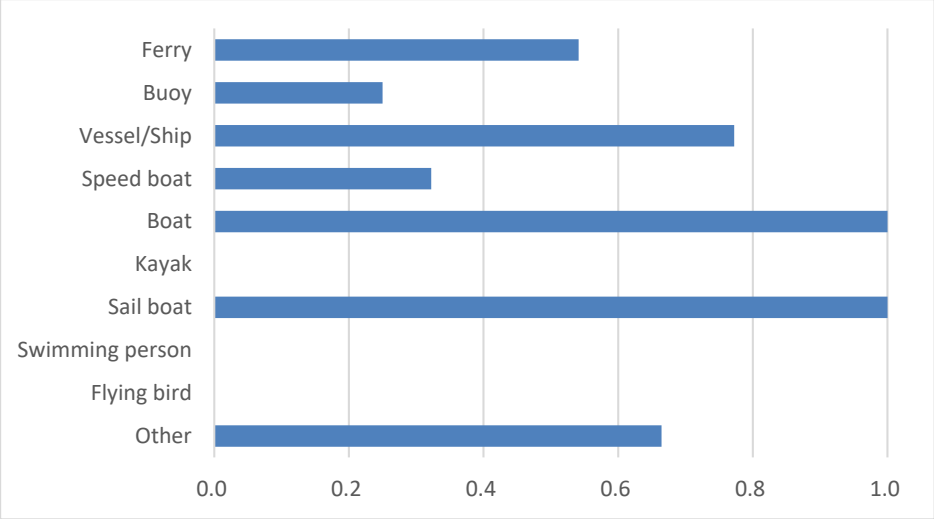

Fig. 11 (Model #3) Average precision for each class on validation images

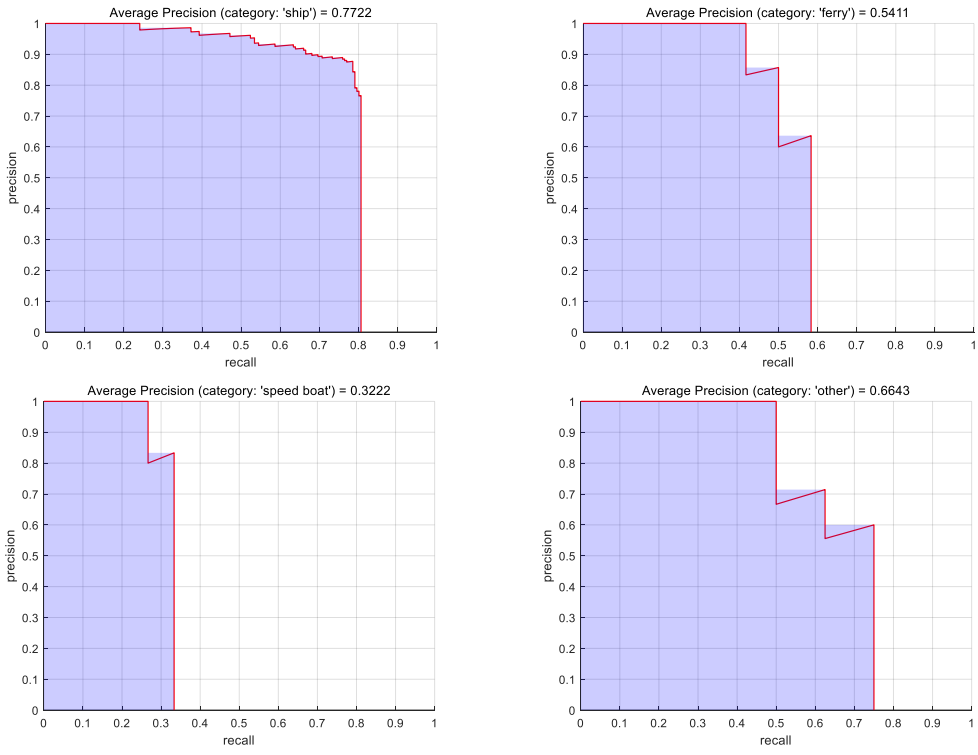

Fig. 12 (Model #3) Precision-recall curve for meaningful classes

Only the class imbalance or a small number of training samples cannot explain these significantly different performances between those classes. Another possible assumption is the scale of objects in the image. Objects belong to the 'speed boat' class tend to have smaller scales than the 'ferry' class. It is mainly because the speed boats in the dataset usually are smaller than ferries in their physical size.

We present additional evaluation results regarding other aspects of the objects, including their size. Table 5 shows the recall, precision, and average precision score sets with respect to distance, orientation, and physical size categories. The results show that the detection performance for 'small' objects is much lower than 'middle' or 'large' objects. Because most of the 'near' objects in the dataset are very small, they are small-scale objects in the images despite their short distances (See Fig. 17). Thus the detection performance for 'near' objects is low as well. Fig. 13 shows the average precision with respect to distance, orientation, and size of the objects on validation images. And Figs. 14-16 show the precision-recall curve for each category of the distance, orientation, and size of the objects, respectively.

Some examples of 'buoy' provide evidence of the small-scale difficulty issue. As shown in Fig. 17, detection boxes, especially for small-scale objects, are relatively hard to achieve the IOU score above the threshold; thus, they are easy to be misdetected. Although the proposed model can detect buoys while the reference model cannot (See Figs. 9 and 10), small-size buoys and other small-scale objects still seem to be a pain point of the proposed model. Our detection model is based on YOLO v2, one of the single-stage detection algorithms known to have lower performance for small-scale objects than two-stage or multi-stage detection algorithms. This issue should be overcome for the practical use of the detection model as it is crucial to detect those small and near objects like buoys that do not have an AIS transmitter but have a high risk of collision.

Table 5 (Model #3) Performance evaluation with respect to distance, orientation, and size of the objects on validation images

Item	Objects	Recall	Precision	AP @IOU ≥ 0.5
< Distance >				
Far (d > 500 m)	197	0.76	0.68	0.7295
Middle (250 < d ≤ 500 m)	37	0.68	0.52	0.6489
Near (d ≤ 250 m)	13	0.46	0.25	0.3333
< Orientation >				
Front/Rear	56	0.68	0.48	0.6242
Side	143	0.70	0.61	0.6604
Oblique	48	0.90	0.88	0.8922
< Size >				
Large (L > 40 m)	197	0.78	0.70	0.7563
Middle (10 < L ≤ 40 m)	12	0.92	0.65	0.8281
Small (L ≤ 10 m)	38	0.42	0.29	0.3387

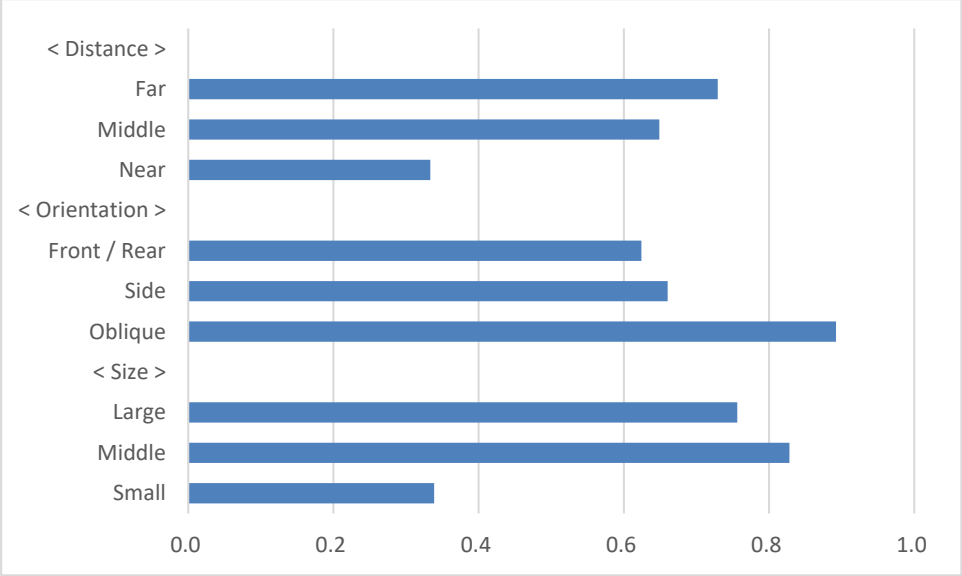

Fig. 13 (Model #3) Average precision with respect to distance, orientation, and size of the objects on validation images

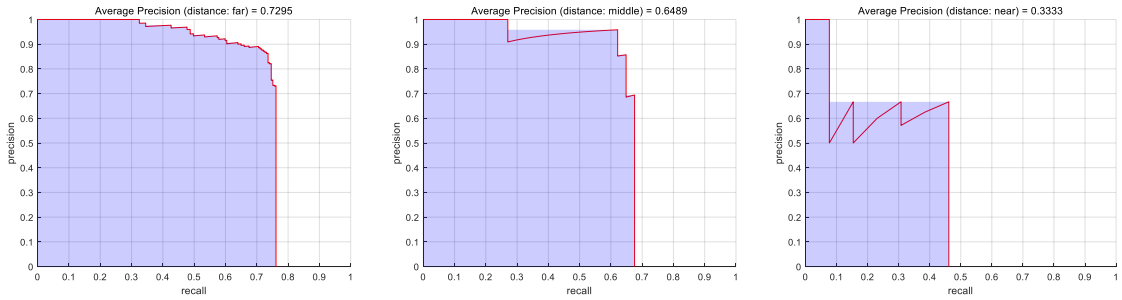

Fig. 14 (Model #3) Precision-recall curve for each distance category

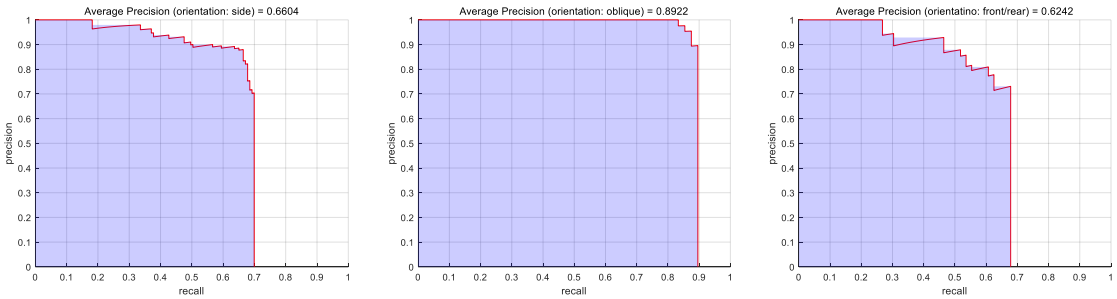

Fig. 15 (Model #3) Precision-recall curve for each orientation category

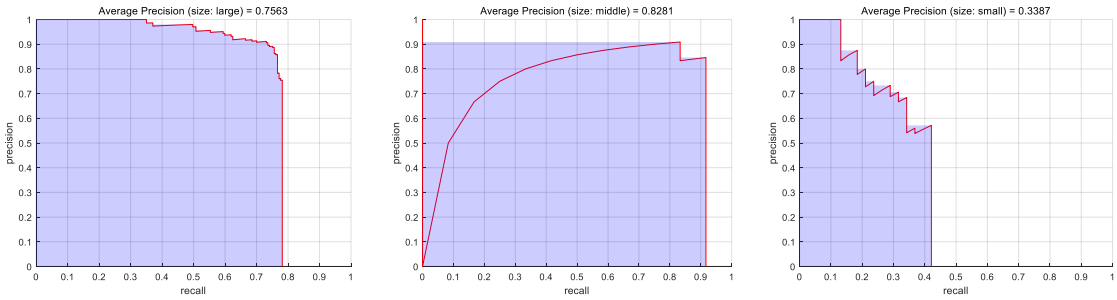

Fig. 16 (Model #3) Precision-recall curve for each size category

Fig. 17 Misdetection examples for the small-scale buoy object. A predicted detection box (red) is not overlapped with a GT box (green) enough (IOU < 0.5), the prediction is considered a false positive

Possible solutions to this issue can be utilizing advanced detection algorithms which can show higher performance for small-scale objects without jeopardizing real-time detection speed, and/or collecting more data of small objects for training.

4.4 Processing time for the object detection

Both the reference and proposed models showed similar processing times for object detection because they share a similar network structure. The detection process for each model took about 0.03 s per image using a GPU (NVIDIA GeForce GTX 1080), which is equivalent to approximately 30 fps. As this is equal to, or greater than, the frame rate of a regular video camera, the proposed model can operate real-time. The proposed model was tested using video to verify its detection speed; if the detection time is too slow, accidents can occur from the ship's inability to change direction in time. Therefore, real-time detection is an important factor, and from the video tests conducted, as shown in Fig. 18, the detection speed is confirmed to be over 30 fps.

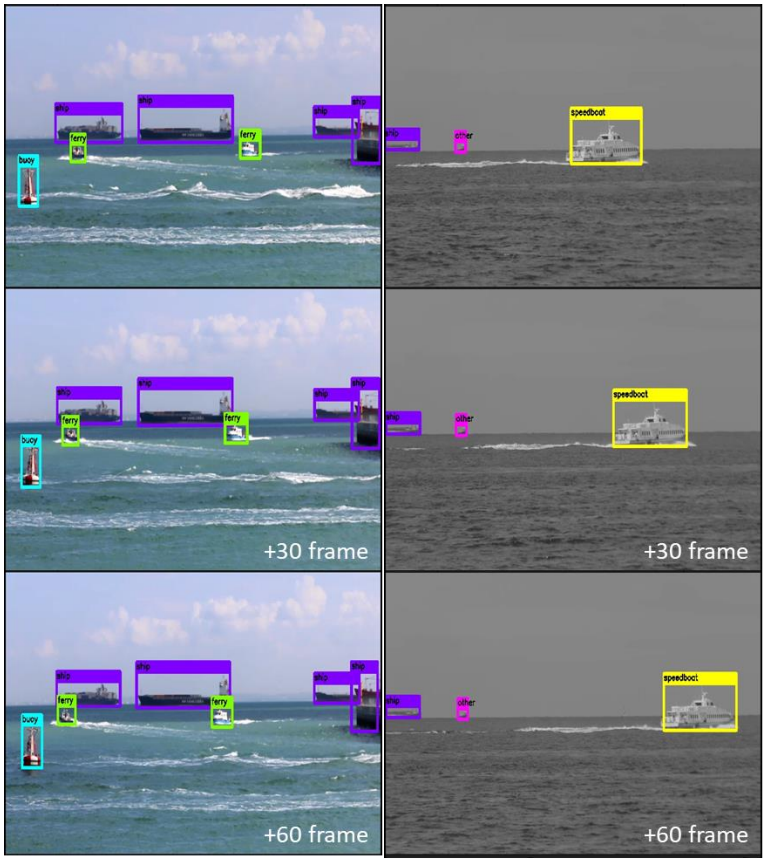

Fig. 18 (Model #3) Ship detection using video samples with 30 fps

4.5 Object detection under low visibility

The proposed model was tested to detect marine images during low visibility under severe weather or night-time conditions. Sample images were randomly prepared by web crawling, but object labeling was not performed for those images. This task is not intended to be an accurate assessment, but to visually demonstrate an approximate level of robustness for images that are much less similarity to the training images from the Singapore Maritime Dataset. Fig. 19 shows detection results for the sample images under the rain, fog, and darkness. All the detections were performed without preprocessing of images. The proposed model can recognize objects from unclear or obscure images.

5. Conclusions

We applied YOLO v2, one of the state-of-the-art CNN-based object detection algorithms, to ship detection and classification problems. The reference model was trained using the universal PASCAL VOC dataset, and it provided a moderately acceptable performance when detecting ships in the

Fig. 19 (Model #3) Ship detection test under low visibility: rain (top), fog (middle), and night (bottom) conditions

maritime field. The proposed models in this study were newly trained using the maritime domain-specific Singapore Maritime Dataset. The recall and IOU scores of Model #3 were almost double those of the reference model when detecting ships from validation images of SMD. Models #1 and #2 showed lower recall and IOU scores than model #3, but higher scores than the reference. This analysis indicates that the CNN structure with a passthrough layer provides better performance and less overfitting pattern. Transfer learning can help prevent overfitting and provide a higher detection rate, particularly when there is an inadequate number of trained images. Although the number of training images used was considerably less than the case of the reference model, it was evident that the domain-specific model can be successfully trained and utilized using the dataset.

We conducted a more detailed performance evaluation on Model #3. The recall, precision, and average precision scores were obtained with respect to class, distance, orientation, and size of the object. We found that the main drawback of the proposed model is its relatively low performance for the small-scale objects. Since they tend to be objects with a high risk of collision, this weakness must be overcome to use the model in practice.

All the detection processes operated on a GPU were faster than a real-time phase. Unlike the previous work by Lee *et al*. (2016) and Zhang *et al*. (2016), the proposed algorithm does not need a separate image preprocessing stage. Furthermore, the proposed model shows the ability to detect

objects under low visibility conditions such as rain, fog, and at night. Not only can the CNN-based object detection method achieve higher performance and faster speed, but the model is more robust and scalable than any other existing machine learning algorithms. As a result, the object detection mechanism based on the proposed CNN model is considered to be practical and feasible for the safe operation of the ship, provided that the performance for the small-scale objects is improved.

This study did not consider video frame information in detection. Therefore, the object detection method could be further studied using frame information to enhance detection accuracy. The method can also be extended to object tracking, which is important when calculating the risk of collision and determining a safe route.

Acknowledgments

This work is an expansion of our previous studies (Lee *et al.* 2018, 2019) and was partially supported by (a) Ministry of Trade, Industry and Energy, Republic of Korea, under "Development of LNG Carrier Performance and Safty Monitoring System Using Satellite Communication and 3D Visualization Technology" (20002720), and (b) Research Institute of Marine Systems Engineering of Seoul National University, Republic of Korea.

References

Cuong, D.D., Hua, X. and Morère, O. (2015), "Maritime vessel images classification using deep convolutional neural networks", *Proceedings of the 6th International Symposium on Information and Communication Technology*, 276-281.

Everingham, M., Eslami, S.M.A., Van-Gool, L., Williams, C.K.I., Winn, J. and Zisserman, A. (2015), "The PASCAL visual object classes challenge: a retrospective", *Int. J. Comput. Vision*, **111**(1), 98-136.

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014), "Rich feature hierarchies for accurate object detection and semantic segmentation", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580-587.

Hermann, D., Galeazzi, R., Andersen, J.C. and Blanke, M. (2015), "Smart sensor based obstacle detection for high-speed unmanned surface vehicle", *IFAC Workshop Series*, **48**(16), 190-197.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S. and Murphy, K. (2016), "Speed/accuracy trade-offs for modern convolutional object detectors", *arXiv preprint arXiv*, 1611.10012.

ImageNet (2019), URL http://www.image-net.org//

Lee, J.M., Lee, K.H., Nam, B. and Wu, Y. (2016), "Study on image-based ship detection for AR navigation", *Proceedings of the 6th International Conference on IT Convergence and Security*, ICITCS, 1-4.

Lee, S.J., Roh, M.I., Lee, H.W., Ha, J.S. and Woo, I.G. (2018), "Image-based ship detection and classification for unmanned surface vehicle using real-time object detection neural networks", *Proceedings of the International Society of Offshore and Polar Engineers 2018*, Sapporo, Japan.

Lee, S.J., Roh, M.I., Oh, M.J., Seok, Y.S., Lee, W.J., Lee, J.B. and Kim, H.S. (2019), "Image-based object detection and tracking method for ship navigation," *Proceedings of International Conference on Computer Applications in Shipbuilding 2019*, Rotterdam, Netherlands.

Lin, M., Chen, Q. and Yan, S. (2013), "Network in network," arXiv:1312.4400.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C. (2016), "SSD: single shot multibox detector", *Proceedings of European Conference on Computer Vision*, Springer, Cham.

Mass, A.L., Hannun, A.Y. and Ng, A.Y. (2013), "Rectifier nonlinearities improve neural network acoustic

models", *Proceedings of the 30th International Conference on Machine Learning*, JMLR: W&CP volume 28.

Pan, S.J. and Yang, Q. (2010), "A survey on transfer learning", *IEEE Transactions on Knowledge and Data Engineering*, **22**(10), 1345-1359.

Prasad, D.K., Rajan, D., Rachmawati, L., Rajabaly, E. and Quek, C. (2017), "Video processing from electro-optical sensors for object detection and tracking in maritime environment: a survey", *IEEE T. Intel. Transport. Syst.*, **18**(8), 1993-2016.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016), "You only look once: unified, real-time object detection", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Redmon, J. and Farhadi, A. (2017), "YOLO9000: better, faster, stronger", *Computer Vision and Pattern Recognition*, arXiv:1612.08242.

Ren, S., He, K., Girshick, R. and Sun, J. (2015), "Faster R-CNN: towards real-time object detection with region proposal networks", *Adv. Neural Inform. Process. Syst.*, 91-99.

Zhang, R., Yao, J., Zhang, K., Feng, C. and Zhang, J. (2016), "S-CNN-based ship detection from high-resolution remote sensing images", *International Archives of the Photogrammetry*, Remote Sensing & Spatial Information Sciences, 41, https://ui.adsabs.harvard.edu/link_gateway/2016ISPAr41B7..423Z/doi:10.5194/isprs-archives-XLI-B7-423-2016.

MK