Multidimensional Systems and Signal Processing
https://doi.org/10.1007/s11045-018-0602-4

A novel approach for detecting the horizon using a convolutional neural network and multi-scale edge detection

Chiyoon Jeong[1,2] · Hyun S. Yang[1] · KyeongDeok Moon[2]

Received: 1 November 2017 / Revised: 10 April 2018 / Accepted: 19 June 2018

Abstract

This paper proposes a novel method for horizon detection that combines a multi-scale approach and a convolutional neural network (CNN). The ability to detect the horizon is the first step toward situational awareness of autonomous ships, which have recently attracted interest, and greatly affects the performance of subsequent steps and that of the overall system. Since typical approaches for horizon detection mainly use edge information, two challenging issues need to be overcome: non-stability of edge detection and complex maritime scenes. The proposed method first detects line features by combining edge information from the various scales to reduce the computational time while mitigating the non-stability of edge detection. Subsequently, CNN is used to verify the edge pixels belonging to the horizon to process complex maritime scenes that contain line features similar to the horizon and changes in the sea status. Finally, linear curve fitting along with median filtering are iteratively used to estimate the horizon line accurately. We compared the performance of the proposed method with state-of-the-art methods using the largest database publicly available. The experimental results showed that the accuracy with which the proposed method can identify the horizon is superior to that of state-of-the-art methods. Our method has a median positional error of less than 1.7 pixels from the center of the horizon and a median angular error of approximately 0.1°. Further, our results showed that our method is the only one capable of detecting the horizon at high speed with high accuracy, which is attractive for practical applications.

Keywords Horizon detection · Convolutional neural networks · Multi-scale edge detection · Maritime images · Autonomous ships

✉ Chiyoon Jeong
dasungi@kaist.ac.kr; iamready@etri.re.kr

Hyun S. Yang
hsyang@kaist.ac.kr

KyeongDeok Moon
moon@etri.re.kr

[1] Department of Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea

[2] SW. Contents Research Laboratory, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea

Published online: 23 June 2018

1 Introduction

In recent years, there has been an increasing interest in remote and autonomous ships in the marine industry. These ships are envisioned to lead to direct cost-reducing benefits and other indirect benefits in the form of a reduction in the number of crewmembers. However, at the same time it is necessary to develop a system to replace the role of the crew to allow the vessel to navigate safely and avoid collisions. Radar systems are a primary sensor system to continuously collect and analyze information about the surrounding situation and to understand the context at sea. Yet, radar is insensitive to small, non-metallic objects and has poor classification power. Thus, additional situational awareness systems are necessary to supplement the radar system.

Electro-optical (EO) sensors are a representative complementary system of radar systems because the images they generate can be intuitively interpreted by human operators and can be automatically analyzed by using computer vision algorithms. Video analytics using EO sensors in the maritime scenario consist of three main steps that include horizon detection, object detection, and object tracking; thus, the detection of the horizon is the first step in the algorithm (Prasad et al. 2017). The result of the detection of the horizon can help to reduce the processing time by restricting the search regions for object detection (Broek et al. 2008; Voles et al. 2000) and it can be used for predicting the distance between a vessel and the EO sensor (Broek et al. 2008). Furthermore, based on information about the horizon, incorrectly detected objects can be removed by filtering, whereby the false alarm rate of detection systems can be reduced by 50% while maintaining the object detection rate (Bloisi et al. 2011). In addition, the horizon detection results can decrease the registration errors of consecutive frames which can increase the performance of the object tracking system (Fefilatyev et al. 2010; Cao et al. 2008). Likewise, horizon detection is the initial step of video analytics and the result of the horizon detection procedure can greatly affect the performance of subsequent steps as well as that of the overall system. Therefore, an accurate method for horizon detection is critical when using video analytics in maritime situations.

Typical approaches for horizon detection assume that the horizon is a straight line and include projection-based methods (Bloisi et al. 2011; Tang et al. 2013; Zhang et al. 2011; Wei et al. 2009; Alpatov et al. 2015), intensity analysis along with a vertical direction (Gershikov et al. 2013; Bouma et al. 2008), or statistical analysis of the sea and sky regions (Ettinger et al. 2002; Gershikov et al. 2013). Recently, hybrid methods combining different approaches were proposed to increase the accuracy of horizon detection (Fefilatyev et al. 2012; Lipschutz et al. 2013; Prasad et al. 2016a, b). However, these methods mainly use edge information, in relation to which there are two challenging issues: non-stability of edge detection and complex maritime scenes. To solve the non-stability of edge detection, several methods adopting a multi-scale approach were proposed (Prasad et al. 2016a, b), but these methods have the disadvantage of being computationally costly. Complex maritime scenes are a new challenging issue that has arisen because of the emergence of a new dataset containing line features similar to that of the horizon (such as through ships and vegetation) and changes in the sea status due to hazes, waves, glints, etc. Fig. 1 shows a sample image of new challenging datasets (Prasad et al. 2017) and a sample image of previously used datasets (Bloisi et al. 2015). Addressing these complex maritime scenes requires machine-learning approaches to be considered (Ahmad et al. 2013; Hung et al. 2013) that can distinguish the edges belonging to the horizon line used in other research areas.

In the field of navigation of unmanned aerial vehicles (UAVs) and micro-air vehicles (MAVs), the horizon is represented as a contour dividing the sky and the ground. Machine-

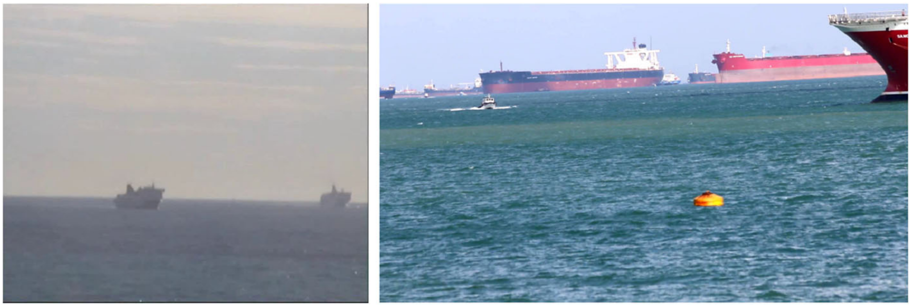

Fig. 1 Example images of maritime datasets. The images on the left and right show simple and complex scenarios, respectively

learning techniques using handcrafted features, such as local features (Ahmad et al. 2013) and color information (Hung et al. 2013), are applied to classify the pixels belonging to the horizon. However, the handcrafted features limit the performance of machine-learning algorithms and the performance of previous methods can be degraded when processing complex maritime scenes that include many objects that can be confused with the edges of which the horizon consists. Recently, a deep convolutional neural network (CNN) yielded satisfactory results in a large-scale visual recognition challenge (Krizhevsky et al. 2012), and Sermanet et al. (2013) demonstrated that the features extracted from a CNN trained for a classification task have improved discriminative power compared to the handcrafted features. This prompted us to explore a way to efficiently use the superior power of CNN for the horizon detection task.

Therefore, in this paper, we propose a novel approach for detecting the horizon line in maritime scenarios. Our approach is based on four steps. First, multi-scale edge detection performs edge extraction at each scale and combines the results to produce a single edge map, which suppresses small noisy edges but retains the prominent edges. Then, we classify each edge pixel as being either horizon or non-horizon using the CNN. Finally, the parameters of the horizon line are estimated using linear curve fitting along with median filtering.

In summary, this paper makes the following contributions. First, in a maritime scenario, we explore the effectiveness of CNN in a horizon detection task and demonstrate that CNN has superior discriminative power than the methods based on handcrafted features. Second, we propose a fusion method to improve the processing speed by effectively combining the edge information of various scales, and using it together with CNN, we developed a high-accuracy, high-speed method for detecting the horizon. Third, we demonstrate that the proposed method achieves the best performance on the Singapore Marine Dataset (SMD) (Prasad et al. 2017), which is the largest, most recent database that is publicly available in maritime scenarios.

The remainder of this paper is organized as follows: Sect. 2 briefly reviews related work on horizon detection in maritime scenarios. Section 3 outlines the proposed horizon detection method. Section 4 discusses the various experiments conducted and the results obtained. Finally, Sect. 5 concludes this paper.

2 Related work

A number of approaches have been developed and reported in the literature for detecting the horizon using EO sensors mounted on vessels, UAVs, and MAVs. The horizon boundary in the previous work can be represented either as a contour dividing the sky and ground or as a straight line. In a maritime scenario, as a horizontal line is generally represented as a straight line, we focused on methods that assume the horizon can be represented as a line. Typical approaches to horizon detection include projection-based methods (Bloisi et al. 2011; Tang et al. 2013; Zhang et al. 2011; Wei et al. 2009; Alpatov et al. 2015), intensity analysis along with a vertical direction (Gershikov et al. 2013; Bouma et al. 2008), or statistical analysis of the sea and sky regions (Ettinger et al. 2002; Gershikov et al. 2013).

In the projection-based methods, an edge detector is first applied to generate an edge map and then the horizon line is determined by projecting the edges to another parametric space where the dominant line can be easily detected. Intensity analysis methods first calculate the gradient map along with the vertical direction, after which they identify the maximal edges in each column of the image. Various optimization techniques such as the least-squares method and dynamic programming are used to calculate the optimal horizon line using maximal edges. Methods based on statistical analysis identify the horizon line by segmenting an image into regions containing the sea and sky. Recently, several methods combining different approaches were proposed to increase the accuracy of horizon detection (Fefilatyev et al. 2012; Lipschutz et al. 2013; Prasad et al. 2016a, b).

Fefilatyev et al. (2012) proposed a hybrid approach combining a Hough transform and statistical analysis. They first selected a limited number of candidate lines using the Hough transform and calculated the separation criteria to find the optimal line that maximizes the difference between the regions containing the water and sky. The Gaussian model in RGB color space was used to represent the color distribution of each region, and the Bhattacharyya distance (Bhattacharyya 1946) was used to measure the similarity of color distributions. Lipschutz et al. (2013) also proposed a method similar to that of Fefilatyev et al. (2012). They used a morphological filter as the pre-processing stage to partially eliminate false detection and used the histograms of both the sky and sea regions to represent the color distribution of each region. Torre and Poggio (1986) stated that, correctly detecting all meaningful edges appearing in an image would require derivatives of different types, and possibly different scales. However, because these methods perform edge detection at a single scale, they frequently fail to detect the edges belonging to the horizon and are thus sensitive to the pre-processing step.

Recently, horizon detection methods using multi-scale edge detection were proposed. A new method named Multi-Scale Cross Modal Linear Feature (MSCM-LiFe) was proposed by Prasad et al. (2016a) and they used multi-scale median filtering to use the persistence of the horizon in a multi-scale approach. MSCM-LiFe estimates the candidate line solutions for each scale using two different modalities, i.e., Hough transforms and intensity gradients. The final solution is selected by calculating the goodness and geometric proximity for each pair of two different methods. Prasad et al. (2016b) also proposed a novel method named multi-scale consistence of weighted edge Radon transform (MuSCoWERT). MuSCoWERT extracts several edge maps using multi-scale median filtering and the weighted edge map is computed by using edge weighting based on the length of the edge. For each scale, multiple line candidate solutions are selected using the Radon transform on a weighted edge map. All candidate solutions are voted as belonging to the solution spaces of a two-dimensional histogram and the bin with the highest frequency is selected as the line parameter. Although these methods can increase the accuracy of horizon detection by adopting the multi-scale approach, performance degradation occurs when the horizon is occluded by ships, waves, etc. Solving this problem necessitates the use of a machine-learning approach that can distinguish the edges belonging to the horizon line from other edges.

Ahmad et al. (2013) proposed a method that uses local features and machine learning to detect the horizon. They first applied a Canny edge detector with various parameters to an image and selected the edges surviving over a broad range of thresholds. The Scale Invariant Feature Transform (SIFT) descriptors (Lowe 2004) were extracted for each selected edge and then used as input to a Support Vector Machine (SVM) classifier to classify the pixels as being either horizon or non-horizon pixels. Dynamic programming was used to identify the horizon line using the edges classified as horizon pixels. Hung et al. (2013) also proposed a similar method that eliminates the non-horizon edges by using a machine-learning approach. They used the color information of the neighboring pixels and the variances of pixels in the upper half and lower half image as the features. They also adopted an SVM classifier to classify the non-horizon pixels using the proposed features. However, the handcrafted features prevent machine-learning algorithms from improving the performance. Recently, CNNs have been shown to excel at various computer vision tasks such as image classification, scene recognition, and action recognition and Sermanet et al. (2013) demonstrated that features extracted from CNN trained for the classification task have enhanced discriminative power compared to handcrafted features.

3 Proposed method

In this section, we elaborate a novel approach for detecting the horizon line in maritime scenarios. A flowchart of the proposed method is shown in Fig. 2. First, multi-scale edge detection is performed on an image and edge maps are combined to produce a single edge map which suppresses noisy edges such as water wakes and waves, but retains the horizon-

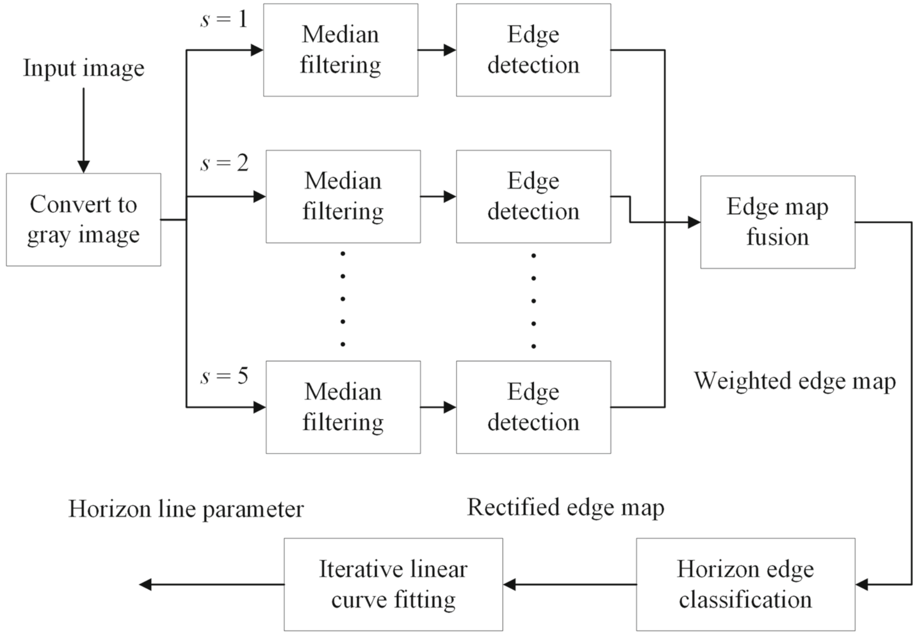

Fig. 2 Flowchart of the proposed method

related edges. In Fig. 2, s is the scale of the median filter applied before edge detection and the actual size of the median filter is determined according to the multiplied constants. Then, CNN is applied to classify each edge pixel as either horizon or non-horizon, resulting in a rectified edge map. Then, the rectified edge map is subjected to linear curve fitting to estimate the initial line candidate and median filtering is applied to reduce the outliers. Subsequently, linear curve fitting and median filtering are iteratively used to find the optimal line by only using the inlier edges.

3.1 Multi-scale edge detection

Methods using multi-scale edge detection were recently proposed because edge detection using a single scale is not sufficient and the use of multiple scales can reduce the ambiguity inherent to single-scale methods (Witkin 1983). Principally, multi-scale edge detection methods are characterized by the way they manage the information obtained at different scales (Lopez-Molina et al. 2013). In this regard, previous work (Prasad et al. 2016a, b) concerned with maritime images used an approach to perform edge detection and horizon estimation on each scale and later to find the final solution. However, because these methods process the image on each scale independently, they are computationally expensive. Especially, MuSCoWERT detects edges and computes the edge weighting based on the length of the edge at each scale, thereby requiring processing time of the order of tens of seconds (Prasad et al. 2016b).

Therefore, in our work, an approach that fuses edge information from each scale is used to reduce the computational time while taking advantage of multi-scale edge detection. Since the median filter effectively removes the noise but preserves the prominent edges, multi-scale images are generated using the median filter as follows:

$$I_s(x, y) = f_s \otimes I(x, y) = median\{I(x+i, y+j) \mid i, j \in [-5s, 5s]\}, \tag{1}$$

where $\otimes$ is the convolution operator, and s is the scale of the median filter. The weighted edge map is generated by using the edge maps obtained by applying the Canny edge detector (Canny 1986) to the smoothed image and the weights according to the scale of the median filter as follows:

$$W(x, y) = \sum_{s=1}^{N} w_s \cdot E_s(x, y), \tag{2}$$

where N is the number of median filters, ω_s is the weight of the scale s, and E_s is the edge maps of the scale s.

Thresholding is applied to the weighted edge map to retain edges that are consistent over multiple scales and to suppress noisy edges, as follows:

$$W_T(x, y) = \begin{cases} 255, & \text{if } W(x, y) \geq t \\ 0, & \text{if } W(x, y) < t \end{cases} \tag{3}$$

The weighted edge map and the edge maps generated using the median filter with various scales are shown in Fig. 3. In Fig. 3, a dilation filter and image inversion are applied to the edge images to improve their readability. Figure 3 shows that the proposed approach fusing edge information from each scale can effectively suppress the noise edges and preserving edges related to the horizon. Unlike previous work, the proposed method can effectively reduce the processing time by processing a single edge map that combines information of various scales.

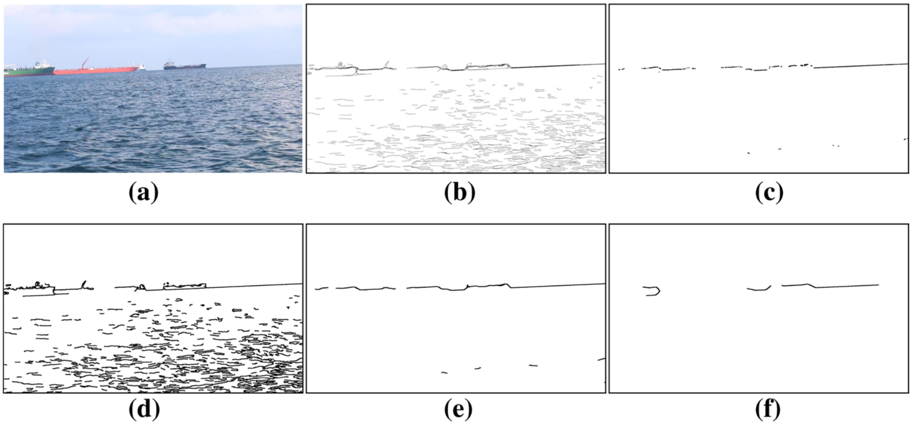

Fig. 3 Multi-scale edge detection **a** input image, **b** weighted edge map, **c** weighted edge map with thresholding and **d**–**f** edge images with a different-sized median filter ($s = 1, s = 3, s = 5$)

Table 1 Network parameters

Layer	CNN-I	CNN-II	CNN-III	CNN-IV
1	Input (32×32)	Input (32×32)	Input (32×32)	Input (32×32)
2	Conv. $32@(f \times f)$	Conv. $32@(f \times f)$	Conv. $16@(f \times f)$	Conv. $32@(f \times f)$
3	Conv. $32@(f \times f)$	Conv. $32@(f \times f)$	Pooling (2×2)	Conv. $32@(f \times f)$
4	Pooling (2×2)	Pooling (2×2)	Conv. $32@(f \times f)$	Pooling (2×2)
5	Conv. $64@(f \times f)$	Conv. $64@(f \times f)$	Pooling (2×2)	Conv. $64@(f \times f)$
6	Pooling (2×2)	Pooling (2×2)	Conv. $64@(f \times f)$	Conv. $64@(f \times f)$
7	Dense (512)	Dense (256)	Pooling (2×2)	Pooling (2×2)
8	Output (1)	Dense (256)	Dense (512)	Dense (512)
9		Output (1)	Output (1)	Dense (512)
10				Output (1)

3.2 Horizon edge classification

The methods that assume that the horizon is represented as a line do not attempt to accurately detect the edges belonging to the horizon because they use projection methods such as the Hough transform and Radon transform, both of which are robust to the noise edges. However, complex maritime scenes contain line features similar to those of the horizon and contain a partially occluded horizon, suggesting that a method that can distinguish the edges belonging to the horizon line from the other edges is necessary. Although methods based on handcrafted features were recently proposed to classify the horizon pixels, the ability of neural networks to learn feature representations from data that are superior to prior handcrafted ones have led to significant progress in the field of computer vision. This led us to employ a CNN to accurately detect the edges belonging to the horizon.

Given the weighted edge map and the input image, the proposed CNN classifies by distinguishing the edges belonging to the horizon from the other edges using the information of neighboring pixels. The feature of chrominance constitutes important information for identifying the edges belonging to the horizon because the horizon is a line that separates the sky and the sea region, which have different color distributions. Therefore, the proposed method extracted the image patch corresponding to the position in the weighted edge map from the input image, and used it as the input of the CNN classification framework. In this paper, we consider four CNN models with various shapes to find the optimal network structure for horizon edge classification. Each of the four models has a different number of convolutional layers and a different number of fully connected layers. A rectified linear unit (ReLU) activation function is applied after every convolutional layer and the fully connected layers. Max-pooling with a 2 × 2 kernel size are used as pooling layers. Details of the CNN models are summarized in Table 1. In Table 1, f is the size of the convolutional kernels. This size remains the same throughout the entire net, similar to VGG network architecture for image classification (Simonyan and Zisserman 2014). CNN-IV, an example architecture of CNN for horizon edge classification, contains the four convolutional layers and three fully connected layers and is illustrated in Fig. 4. For each edge pixel, a patch sized 32 × 32 centered at the edge pixel is extracted from the input image and then used as input for the CNN model.

Figure 5 shows examples of the result of horizon edge classification using the CNN-IV model, resulting in a rectified edge map. By using the high-quality discriminative power of deep neural networks, the proposed method can correctly classify the edges belonging to the horizon line from the other edges in complex maritime scenes. The rectified edge map can help to predict the horizon accurately by removing outliers.

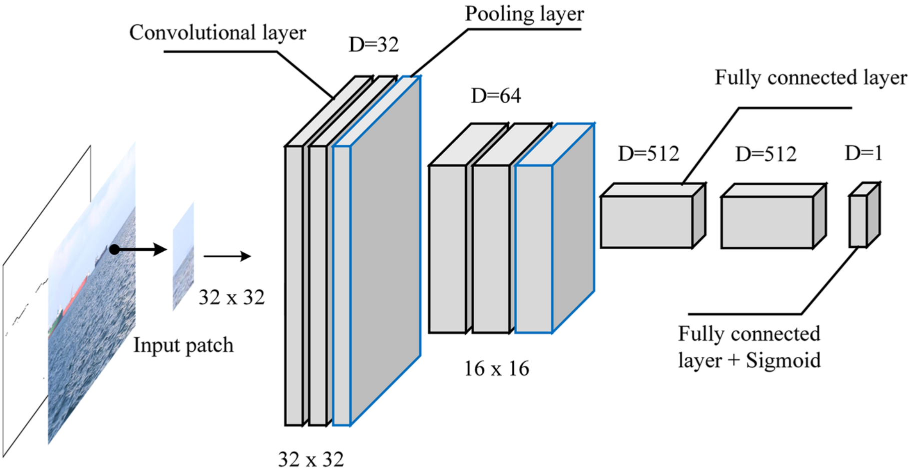

Fig. 4 An example architecture of the convolutional neural networks for horizon edge classification (CNN-IV)

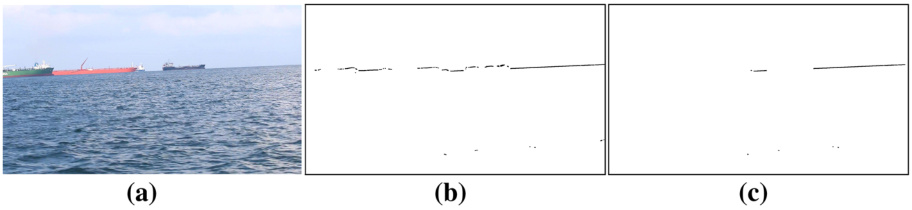

Fig. 5 Effect of horizon edge classification **a** input image, **b** weighted edge map with thresholding and **c** rectified edge map by horizon edge classification

3.3 Horizon line estimation

The Hough transform and linear curve-fitting method are popular methods for horizon line estimation. The Hough transform (Duda and Hart 1972) is a robust method for detecting lines in an image containing noisy and missing data, but the transform is sensitive to parameter quantization that affects the accuracy of line detection. The linear curve-fitting method accurately estimates the horizon line, but is highly sensitive to outliers. A recent report (Xu and Shin 2014) pointed out that the accuracy of the linear curve-fitting method in terms of line detection is higher than that of methods based on the Hough transform.

Linear curve fitting is applied to the rectified edge map to estimate the initial line candidate, which is subsequently used by the proposed method to calculate the distance between the estimated line and the edges, after which the edges with a larger distance than the median distance are removed. This process is repeated two more times to find the optimal parameter of the horizon line. The use of high-precision linear curve fitting along with median filtering enables the proposed method to perform horizon line detection more accurately.

4 Experimental results

4.1 Dataset

In previous work, researchers usually used their own dataset for evaluating the performance, but the number and variety of test images was limited. Therefore, we verified the performance of the proposed method on SMD (Prasad et al. 2017), which is the largest database publicly available and contains many complex maritime scenes. All the videos in SMD were acquired at high-definition resolution and are grouped as two sub-groups: onboard videos and on-shore videos. Onboard videos were captured by a camera installed on moving ships and on-shore videos were captured by a fixed camera installed on-shore. Onboard videos are adversely affected by distant vegetation close to the horizon, wakes from ships, and waves. The major challenges in on-shore videos are overlaps between the horizon and vessels, wakes from ships, and variations in the color of the sea.

Details of the dataset are summarized in Table 2, with example images from each shown in Fig. 6. In Fig. 6a, the ground truth annotation for the horizon line is shown and Y, α denote the vertical position and the gradient of the horizon line, respectively. The errors in computing Y and α are used as a performance metric.

Table 2 Details of the datasets

	Onboard	Onshore
No. of videos	11	28
No. of frames	2772	12604
Min (Y-mean(Y)) (pixels)	− 436.30	− 13.54
Max (Y-mean(Y)) (pixels)	467.86	9.95
Standard deviation of Y (pixels)	145.10	1.52
Min (α -mean(α)) (°)	− 26.34	− 9.99
Max (α -mean(α)) (°)	12.99	0.51
Standard deviation of α (°)	1.11	0.04

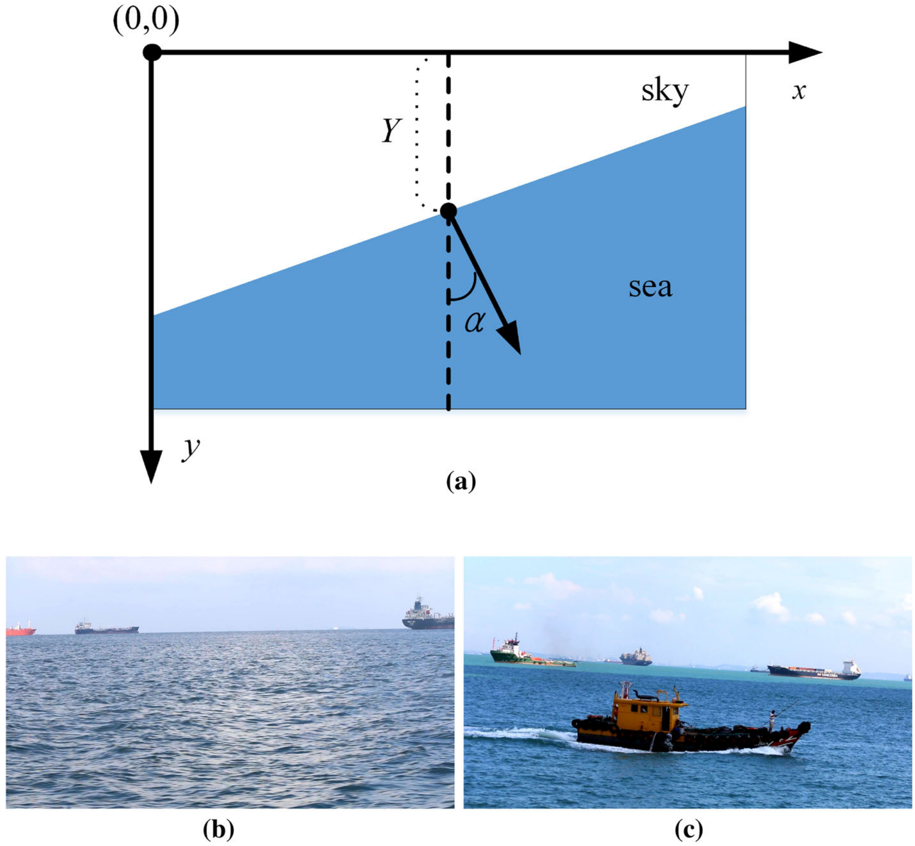

Fig. 6 **a** Ground truth annotation, **b** a sample image of onboard videos and **c** a sample image of onshore videos

4.2 Performance of horizon edge classification

In this section, we demonstrate the extent to which the proposed CNN model accurately classifies the edges associated with the horizon line. The SIFT descriptors and SVM classifier, which were used in previous work (Ahmad et al. 2013), were used to compare the classification accuracy. In addition, the handcrafted features proposed by Hung et al. (2013) were used. Although they used RGB color information, the variance of illumination change, and location information as the features, we excluded the location information for fair comparison. Therefore, the features consisted of 194 dimensions. (Originally, the number of feature dimensions was 210.) Additionally, the AdaBoost classifier with handcrafted features, such as SIFT descriptors and RGB color information, was used to compare the performance of horizon edge classification.

We trained and tested the edge classifiers by generating positive and negative samples from the SMD. Positive samples were randomly extracted from the edge pixels on the horizon lines and negative samples were randomly extracted from the edge pixels at a certain distance from the horizontal line. An example image for generating samples is shown in Fig. 7, where the red and blue rectangles signify positive and negative samples, respectively, and the green

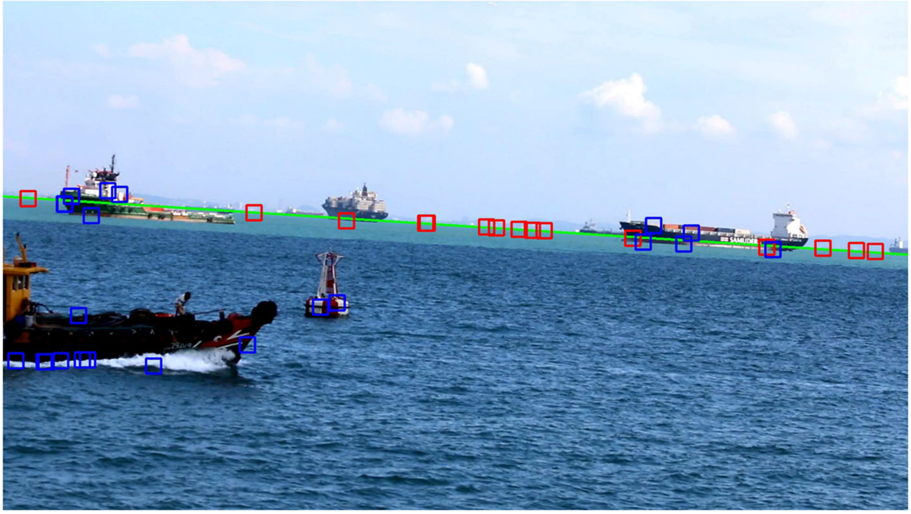

Fig. 7 Sample generation for CNN. The red and blue rectangles denote positive and negative samples, respectively. The green line represents the ground truth of the horizon (Color figure online)

line represents the ground truth of the horizon. The training set contained 7981 positive and 8477 negative samples and the test set comprised 2087 positive and 2234 negative samples.

All CNN models were implemented using the Tensorflow-GPU library (Abadi et al. 2015). Stochastic gradient descent (SGD) (Bottou 2012) was used to find the best models and a constant learning rate of 0.001 was used. The batch size was set to 100 and all fully connected layers used a dropout with a probability of 0.5. Data augmentation using a horizontal flip was applied to the training procedure and the number of epochs was adjusted using the early stopping method to prevent overfitting.

Both the SIFT and color features were extracted using the OpenCV Library (Itseez 2015). SIFT descriptors were extracted from each sample with various scales and the RGB color information and the variance of illumination change were extracted from the resized 8 × 8 image. LIBSVM (Chang and Lin 2011) was used to train the positive and negative samples and a radial basis function was used as kernel function of the SVM. The optimal parameters of the SVM classifier, the penalty parameter (C) and the kernel parameter (γ), were determined through grid search using cross-validation. The AdaBoost classifier was implemented using the scikit-learn machine-learning library (Pedregosa et al. 2011). We used the decision tree as a weak classifier for the AdaBoost classifier and the optimal parameter of the classifier, the number of estimators (n), and the maximum depth of the decision tree (d), were determined through grid search.

Figure 8 shows the performance of the horizon edge classification according to the type of CNN models and the size of the convolutional kernels, with the mean accuracy of 10 repeated experiments indicated by red rectangular marks. The CNN-IV model with the size of the convolutional kernels of 2 showed the best performance among the various CNN models. Thus, we adopted the CNN-IV model to classify the edges related to the horizon.

In addition, we compared the performance of the proposed CNN model with that of the SVM classifiers and the AdaBoost classifier, both of which use handcrafted features. The experimental results that were obtained are presented in Table 3. It is clear that the proposed CNN model has higher discriminative power than the handcraft-based method. By effectively

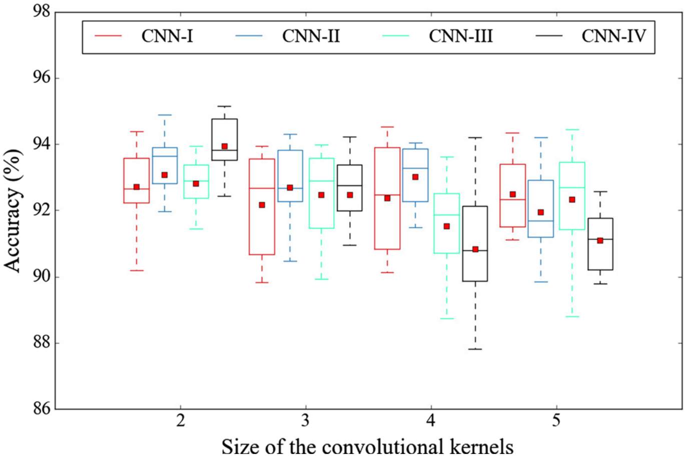

Fig. 8 Performance of the CNN models by increasing the size of convolutional kernels in the entire networks

Table 3 Results of the horizon edge classification

	Accuracy (%)
Proposed CNN model	95.06
SIFT (scale = 12), SVM ($C = 8.0$, $\gamma = 0.5$)	82.60
SIFT (scale = 12), AdaBoost ($d = 10$, $n = 250$)	85.26
SIFT (scale = 16), SVM ($C = 8.0$, $\gamma = 0.5$)	81.64
SIFT (scale = 16), AdaBoost ($d = 10$, $n = 300$)	83.96
SIFT (scale = 24), SVM ($C = 8.0$, $\gamma = 2.0$)	79.22
SIFT (scale = 24), AdaBoost ($d = 9$, $n =$200)	84.69
Color information (8 × 8 blocks), SVM ($C = 32.0$, $\gamma = 0.125$)	86.92
Color information (8 × 8 blocks), AdaBoost ($d = 6$, $n = 250$)	88.87

reducing the number of edge pixels that do not belong to the horizon, the proposed method for detecting the horizon can be expected to accurately identify the horizon.

4.3 Horizon detection performance

We compared the performance of the proposed method with that of state-of-the-art methods. All the known comparison methods, except MuSCoWERT, were implemented by the authors of this paper. Further, we compared the performance of the proposed method with and without edge classification: baseline, the proposed method without edge classification; AdaBoost, the proposed method with edge classification using the AdaBoost classifier and color information; CNN, the proposed method with edge classification using the best CNN model. For each of the methods, the horizon parameters of the detected line, Y and α, were computed and the difference between these parameters and those of the ground truth were calculated to measure the accuracy.

Hough transform based (Hough) (Gershikov et al. 2013) First, this method was used to preprocess an input image using the median filter and then, the Canny edge detector was applied to the preprocessed image to compute the edge map. Then, a Hough transform was applied to the edge map and the line parameters with the highest Hough score were determined as the final solution.

Intensity variation based (IntV) (Gershikov et al. 2013) A median filter was used to smooth an input image and then, to identify the points with maximum intensity variation in each column of the image. Then, the least-squares method was used to find the optimal line using all these points.

Method of Fefilatyev et al. (FGSL) (Fefilatyev et al. 2012) This method first performed the Hough transform on the edge map of the image, and then selected the candidate lines from the N largest distinct peaks in the Hough accumulator space. For each candidate, the color distributions of the two regions divided by the candidate line were modeled as Gaussian probabilistic density functions. The following measure, based on the Bhattacharyya distance (Bhattacharyya 1946), was used to calculate the difference between the color distributions of the sea and sky regions:

$$f(Y,\alpha) = (\mu_1 - \mu_2)^T(\Sigma_1 - \Sigma_2)^{-1}(\mu_1 - \mu_2), \tag{4}$$

where μ and Σ are the mean vectors and the covariance matrices of the color distribution of the region. We selected 10 candidates for improved computational efficiency and the median filter was used to reduce the noise as a pre-processing step.

Method of Lipschutzet al. (LHSL) (Lipschutz et al. 2013) This method is similar to FGSL but uses a different criterion to measure the similarity as follows:

$$f(Y,\alpha) = \sum_i H_1(i)H_2(i), \tag{5}$$

where H is the color histograms of each region and i is the candidate line. We used the same number of candidates as the FGSL method and employed morphological erosion as the pre-processing step.

MSCM-LiFe (Prasad et al. 2016a) This method was used to first generate 10 multi-scale images using the different sizes of the vertical median filter and then, to select 100 candidates by applying the Hough transform to the edge maps of multi-scale images. In addition, the intensity variation approach was used to determine the line candidate by using the mean multi-scale images computed for each scale. For each pair of Hough and intensity variation candidates, the goodness score of each candidate in the pair and the geometric similarity of candidate lines in the pair were measured to find the optimal solution.

MuSCoWERT (Prasad et al. 2016b) This method generated multi-scale images by applying a median filter with different sizes. For each multi-scale image, the edge map was generated and the weighted edge map was computed by using the length of the edge. The candidates were selected using the Radon transform on the weighted edge map and a voting scheme was used to select the final solution from the candidates of all scales.

Statistical details of the errors in parameters Y and α are listed in Tables 4 and 5. For both the parameters and both scenarios, we see that the proposed method outperformed all the other methods considering the small values of its errors. The proposed method has a median position error of less than 1.7 pixels from the center of the horizon and a median angular error of approximately 0.1°. The 95th percentile error of the proposed method was very small relative to the other methods and this implies that the proposed method is highly robust and consistent over datasets with great diversity.

Table 4 Results for the onboard videos in the Singapore marine dataset (results of the MuSCoWERT method were taken from Prasad et al. (2016b))

	$\mid Y - Y_{GT} \mid$ (pixels)			$\mid \alpha - \alpha_{GT} \mid$ (°)		
	25th percentile	50th percentile	95th percentile	25th percentile	50th percentile	95th percentile
Ours (CNN)	0.53	1.21	3.79	0.04	0.10	0.38
Ours (AdaBoost)	0.54	1.20	4.11	0.04	0.11	0.44
Ours (baseline)	0.71	1.60	14.02	0.05	0.13	1.46
MuSCoWERT	0.54	1.49	8.17	0.06	0.25	0.88
MSCM-LiFe	1.16	2.84	505.78	0.17	0.38	5.50
LHSL	13.78	25.65	507.92	0.88	1.37	6.52
FGSL	5.28	10.85	581.44	0.67	1.00	3.88
IntV	13.36	24.89	498.17	0.87	1.35	6.12
Hough	2.27	221.67	520.34	0.25	1.00	4.57

Table 5 Results for the onshore videos in the Singapore marine dataset (results of the MuSCoWERT method were taken from Prasad et al. (2016b))

	$\mid Y - Y_{GT} \mid$ (pixels)			$\mid \alpha - \alpha_{GT} \mid$ (°)		
	25th percentile	50th percentile	95th percentile	25th percentile	50th percentile	95th percentile
Ours (CNN)	0.85	1.71	10.89	0.03	0.12	0.82
Ours (AdaBoost)	0.90	1.90	13.32	0.04	0.12	0.76
Ours (baseline)	1.36	3.10	52.43	0.09	0.28	2.18
MuSCoWERT	1.14	2.63	11.41	0.14	0.21	1.07
MSCM-LiFe	1.63	3.88	81.59	0.11	0.18	1.14
LHSL	14.96	27.92	109.00	0.75	1.03	3.86
FGSL	5.88	11.53	64.70	0.75	1.00	2.87
IntV	2.08	5.82	39.89	0.14	0.52	5.37
Hough	3.12	165.02	460.24	0.14	0.36	3.80

The experimental results also showed that methods adopting the multi-scale approach such as the proposed method, MuSCoWERT, and MSCM-LiFE outperformed methods adopting the single-scale approach. Since the latter methods are sensitive to edge detection and parameter choices, they can neither accommodate varied situations nor complex maritime scenes. Therefore, the multi-scale approach offers a good solution to overcome this challenge.

In order to underscore the improvement affected by the horizon edge classification, we tested the performance of the proposed method with and without this classification. The results in Tables 4 and 5 indicate that the horizon edge classification helps to improve the accuracy of horizon detection. When the horizon detection methods using CNN are compared with the method using the AdaBoost classifier, the experimental results show that the accuracy with which the method using CNN can identify the horizon is superior to that of the method using AdaBoost. Especially, the method using CNN showed a lower 95th percentile error than the method using the AdaBoost classifier in the onshore videos containing many complex maritime scenes and thus provides reliable results in the worst case.

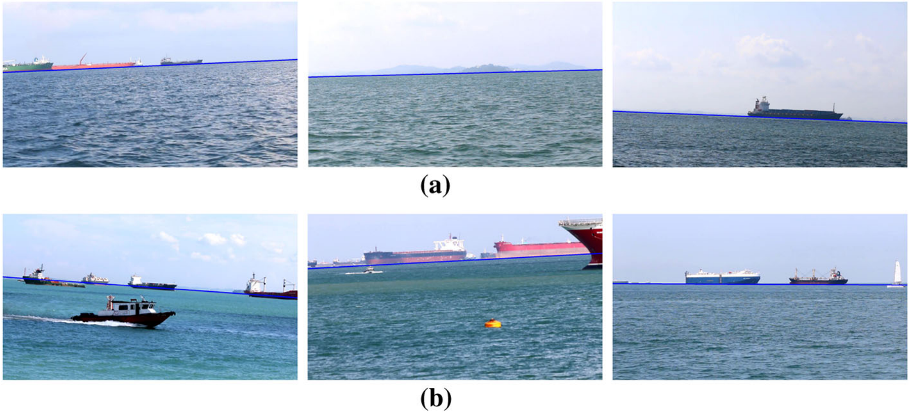

Fig. 9 Example images of horizon detection results. The blue line shows the results of the proposed method for resultant images of **a** onboard videos and **b** onshore videos (Color figure online)

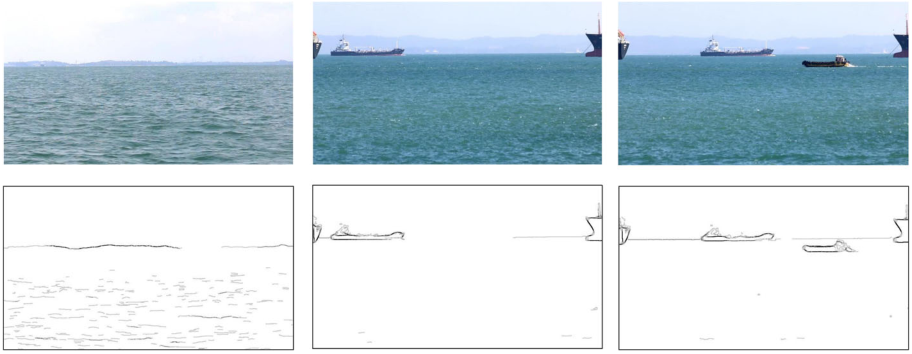

Fig. 10 Example images showing the degraded performance of the proposed method. The top and bottom rows show the input images and the weighted edge maps for the input images, respectively

Sample frames from each type of scenario in the SMD are shown in Fig. 9, which shows that the proposed method accurately identified the horizon in complex maritime scenes. Figure 10 shows examples in which the performance of the proposed method was degraded. When the horizon has relatively weak line features or line features are entirely absent from the horizon due to a smooth boundary between the sky and sea regions, the proposed method incorrectly detects the horizon because it cannot detect the edges representing the horizon.

The average computational times required per image for each method are provided in Table 6. We implemented all the methods except MuSCoWERT by using Python and executed them on an Intel E5-1680 CPU. MuSCoWERT was implemented using MATLAB 2015b, and the results obtained on an Intel i7-3770 CPU were taken from Prasad et al. (2016b). Although previous methods using the multi-scale approach had slow processing times of the order of tens of seconds, the proposed method processed images within seconds by processing a single edge map that combines information of various scales.

Figure 11 visualizes the results of the proposed method and the other methods as plotted on the computational time—accuracy axis. As can be seen in Fig. 11, our method is the

Table 6 Average processing time per frames in second (results of the MuSCoWERT method were taken from Prasad et al. (2016b))

	Onboard	Onshore
Our	0.63	0.61
MuSCoWERT	9.2	9.5
MSCM-LiFe	6.73	6.83
LHSL	13.75	13.76
FGSL	36.58	36.63
IntV	0.30	0.30
Hough	0.11	0.10

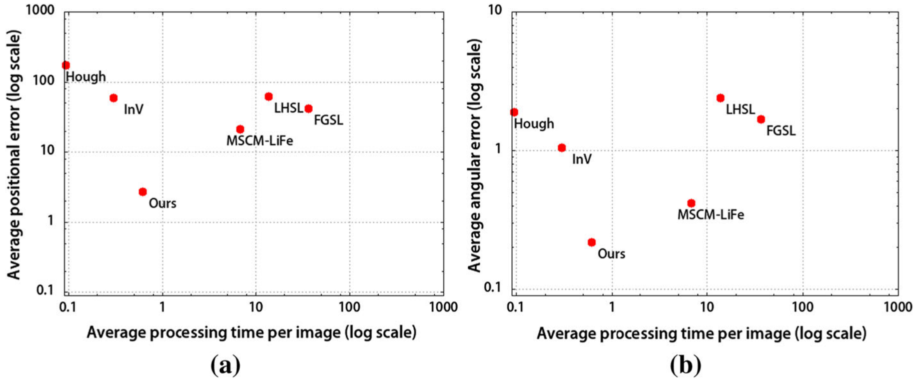

Fig. 11 Performance of horizon detection methods on the computational time versus accuracy axis

only one able to detect the horizon at high speed with high accuracy, which is attractive for practical applications.

5 Conclusion

In this paper, we proposed a novel approach that enables a seagoing ship to autonomously detect the horizon by combining a multi-scale approach with CNN. An approach whereby edge information from various scales is fused was used to reduce the computational time while taking advantage of multi-scale edge detection. We explored the effectiveness of CNN for the horizon detection task and confirmed that horizon edge classification using CNN is effective in improving the accuracy of horizon detection. We compared the performance of the proposed method with that of state-of-the-art methods on the SMD, which is the largest publicly available database that contains many complex maritime scenes, and the experimental results showed that the proposed method can identify the horizon more accurately than comparable state-of-the-art methods. The median position error of the proposed method is less than 1.7 pixels from the center of the horizon and this is approximately 0.16% of the height of the input image. Further, the median angular error of the proposed method is approximately 0.1°. The experimental results confirmed that our method was the only one capable of detecting the horizon at high speed with high accuracy, which is attractive for practical applications.

Although the proposed method showed excellent performance, it may fail in scenarios characterized by an absence of line features on the horizon. Thus, in the future, we plan to combine the proposed method with intensity analysis to overcome the absence of these line features.

Acknowledgements Chiyoon Jeong and KyeongDeok Moon were supported by the Industrial Strategic Technology Development Program (10077551, Development of original technology for artificial intelligence systems for autonomous navigating ships) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea). Hyun S. Yang was supported by the ICT R&D program of MSIP/IITP [2017-0-00162, Development of Human-care Robot Technology for Aging Society] and the research project of the Ministry of Oceans and Fisheries of Korea (A fundamental research on maritime accident prevention–phase 2).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/. Accessed 10 July 2017.

Ahmad, T., Bebis, G., Regentova, E. E., & Nefian, A. (2013). *A machine learning approach to horizon line detection using local features* (pp. 181–193). Berlin: Springer. https://doi.org/10.1007/978-3-642-41914-0_19.

Alpatov, B. A., Babayan, P. V., & Shubin, N. Y. (2015). Weighted Radon transform for line detection in noisy images. *Journal of Electronic Imaging*, *24*, 023023. https://doi.org/10.1117/1.JEI.24.2.023023.

Bhattacharyya, A. (1946). On a measure of divergence between two multinomial populations. *Sankhy: The Indian Journal of Statistics (1933–1960)*, *7*(4), 401–406.

Bloisi, D., Iocchi, L., Fiorini, M., & Graziano, G. (2011). Automatic maritime surveillance with visual target detection. In *Proceedings of the international defense and homeland security simulation workshop (DHSS)* (pp. 141–145), Rome, Italy. http://www.dis.uniroma1.it/~bloisi/papers/bloisi-maritime-surveillance.pdf.

Bloisi, D. D., Iocchi, L., Pennisi, A., & Tombolini, L. (2015). ARGOS-Venice boat classification. In *12th IEEE international conference on advanced video and signal based surveillance (AVSS), 2015* (pp. 1–6). https://doi.org/10.1109/AVSS.2015.7301727.

Bottou, L. (2012). *Stochastic gradient descent tricks* (pp. 421–436). Berlin: Springer. https://doi.org/10.1007/978-3-642-35289-8_25.

Bouma, H., de Lange, D. J. J., van den Broek, S. P., Kemp, R. A. W., & Schwering, P. B. W. (2008). Automatic detection of small surface targets with electro-optical sensors in a harbor environment. In *Proc. SPIE, Electro-Optical Remote Sensing, Photonic Technologies, and Applications II*, Cardiff, Wales (Vol. 7114). https://doi.org/10.1117/12.799813.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI*, *8*(6), 679–698. https://doi.org/10.1109/TPAMI.1986.4767851.

Cao, X., Rasheed, Z., Liu, H., & Haering, N. (2008). Automatic geo-registration of maritime video feeds. In *2008 19th international conference on pattern recognition* (pp. 1–4). https://doi.org/10.1109/ICPR.2008.4761567.

Chang, C. C., & Lin, C. J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 27:1–27:27. https://doi.org/10.1145/1961189.1961199.

Duda, R. O., & Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, *15*(1), 11–15. https://doi.org/10.1145/361237.361242.

Ettinger, S. M., Nechyba, M. C., Ifju, P. G., & Waszak, M. (2002). Vision-guided flight stability and control for micro air vehicles. In *IEEE/RSJ international conference on intelligent robots and systems* (Vol. 3, pp. 2134–2140). https://doi.org/10.1109/IRDS.2002.1041582.

Fefilatyev, S., Goldgof, D., & Lembke, C. (2010). Tracking ships from fast moving camera through image registration. In *2010 20th international conference on pattern recognition* (pp. 3500–3503). https://doi.org/10.1109/ICPR.2010.854.

Fefilatyev, S., Goldgof, D., Shreve, M., & Lembke, C. (2012). Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system. *Ocean Engineering*, *54*(Supplement C), 1–12. https://doi.org/10.1016/j.oceaneng.2012.06.028.

Gershikov, E., Libe, T., & Kosolapov, S. (2013). Horizon line detection in marine images: Which method to choose? *International Journal on Advances in Intelligent Systems*, *6*(1 and 2), 79–88.

Hung, Y. L., Su, C. W., Chang, Y. H., Chang, J. C., & Tyan, H. R. (2013). Skyline localization for mountain images. In *2013 IEEE international conference on multimedia and expo (ICME)* (pp. 1–6). https://doi.org/10.1109/ICME.2013.6607424.

Itseez. (2015). Open source computer vision library. https://github.com/itseez/opencv. Accessed 12 June 2017.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th international conference on neural information processing systems* (pp. 1097–1105). Curran Associates Inc., USA, NIPS'12. http://dl.acm.org/citation.cfm?id=2999134.2999257.

Lipschutz, I., Gershikov, E., & Milgrom, B. (2013). New methods for horizon line detection in infrared and visible sea images. *Ocean Engineering*, *3*(8), 226–233.

Lopez-Molina, C., Baets, B. D., Bustince, H., Sanz, J., & Barrenechea, E. (2013). Multiscale edge detection based on gaussian smoothing and edge tracking. *Knowledge-Based Systems*, *44*(Supplement C), 101–111. https://doi.org/10.1016/j.knosys.2013.01.026.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Prasad, D. K., Rajan, D., Prasath, C. K., Rachmawati, L., Rajabally, E., & Quek, C. (2016a). Mscm-life: multi-scale cross modal linear feature for horizon detection in maritime images. In *2016 IEEE region 10 conference (TENCON)* (pp. 1366–1370). https://doi.org/10.1109/TENCON.2016.7848237.

Prasad, D. K., Rajan, D., Rachmawati, L., Rajabally, E., & Quek, C. (2016b). Muscowert: Multi-scale consistence of weighted edge radon transform for horizon detection in maritime images. *Journal of the Optical Society of America A*, *33*, 2491.

Prasad, D. K., Rajan, D., Rachmawati, L., Rajabally, E., & Quek, C. (2017). Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey. *IEEE Transactions on Intelligent Transportation Systems*, *18*(8), 1993–2016. https://doi.org/10.1109/TITS.2016.2634580.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. CoRR abs/1312.6229, http://arxiv.org/abs/1312.6229.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556, http://arxiv.org/abs/1409.1556.

Tang, D., Sun, G., Wang, D. H., Niu, Z. D., & Chen, Z. P. (2013). Research on infrared ship detection method in sea-sky background. *Proc. SPIE, International Symposium on Photoelectronic Detection and Imaging 2013: Infrared Imaging and Applications* (Vol. 8907). https://doi.org/10.1117/12.2033039.

Torre, V., & Poggio, T. A. (1986). On edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI*, *8*(2), 147–163. https://doi.org/10.1109/TPAMI.1986.4767769.

van den Broek, S. P., Bouma, H., & Degache, M. A. C. (2008). Discriminating small extended targets at sea from clutter and other classes of boats in infrared and visual light imagery. *ProcSPIE*, *6969*, 10–34. https://doi.org/10.1117/12.777542.

Voles, P., Smith, A. A. W., & Teal, M. K. (2000). *Nautical scene segmentation using variable size image windows and feature space reclustering* (pp. 324–335). Berlin: Springer. https://doi.org/10.1007/3-540-45053-X_21.

Wei, H., Nguyen, H., Ramu, P., Raju, C., Liu, X., & Yadegar, J. (2009). Automated intelligent video surveillance system for ships. In *Proc. SPIE, Optics and Photonics in Global Homeland Security V and Biometric Technology for Human Identification VI* (Vol. 7306). https://doi.org/10.1117/12.819051.

Witkin, A. P. (1983). Scale-space filtering. In *Proceedings of the eighth international joint conference on artificial intelligence* (Vol. 2, pp. 1019–1022). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI'83. http://dl.acm.org/citation.cfm?id=1623516.1623607.

Xu, Z., & Shin, B. S. (2014). *A statistical method for peak localization in Hough space by analysing butterflies* (pp. 111–123). Berlin: Springer. https://doi.org/10.1007/978-3-642-53842-1_10.

Zhang, H., Yin, P., Zhang, X., & Shen, X. (2011). A robust adaptive horizon recognizing algorithm based on projection. *Transactions of the Institute of Measurement and Control*, *33*(6), 734–751. https://doi.org/10.1177/0142331209342201.