Object Detection based on Multi-sensor Proposal Fusion in Maritime Environment

Fahimeh Farahnakian[1], Mohammad-Hashem Haghbayan[1], Jonne Poikonen[1],
Markus Laurinen[2], Paavo Nevalainen[1] and Jukka Heikkonen[1]
[1]Department of Future Technologies, University of Turku, Turku, Finland
[2]Rolls-Royce Oy Ab, Rauma Finland
Email: {fahfar, mohhag, jukapo, ptneva, jukhei}@utu.fi, {markus.laurinen}@rolls-royce.com

***Abstract*—In this paper, we propose an effective object detection framework based on proposal fusion of multiple sensors such as infrared camera, RGB cameras, radar and LiDAR. Our framework first applies the Selective Search (SS) method on RGB image data to extract possible candidate proposals which likely contain the objects of interest. Then it uses the information from other sensors in order to reduce the number of generated proposals by SS and find more dense proposals. Finally, the class of objects within the final proposals are identified by Convolutional Neural Network (CNN). Experimental results on real dataset demonstrate that our framework can precisely detect meaningful object regions using a smaller number of proposals than other object proposals methods. Further, our framework can achieve reliable object detection and classification results in maritime environments.**

***Keywords*—autonomous vehicles, object detection, proposal generation, deep neural networks, maritime environment.**

I. Introduction

Most of state-of-the-art object detectors employ object proposals methods for guiding the search for object instances across images [1], [2], [3]. These methods can improve detection accuracy by extracting reliable proposals that contain objects of interest. Moreover, they can considerably reduce computation compared with a dense detection approach such as sliding window by avoiding exhaustive sliding window search across images. Region-based Convolutional Neural Network (R-CNN) [1] is one of the most popular proposal methods that has been extended to a variety of new tasks and datasets. Although it is computationally expensive because of passing all the proposals by Selective Search (SS) [4] (usually several thousand) separately to CNN. Fast R-CNN [2] and Faster R-CNN [3] achieves lower computational time and cost with a deep convolutional neural network.

In this paper, we present a robust object detection framework based on proposal fusion of multi-sensor. Fig. 1 shows an overview of the proposed framework. Firstly, our framework generate initial proposals (i.e. regions of interest that are likely to contain objects) using SS. SS is a famous object proposals methods in generating well-localized proposals in the last few years. Then, the framework finds more dense and reliable proposals from the initial proposals based on the information from various sensors. For this purpose, the framework fuses the object proposals extracted from other sensors such as IR cameras, radar and LiDAR using a bounding box matching metric. The reduced number of proposals compared to SS enables the use of stronger models for object identification. Therefore, the final proposals are feed to CNN as a classifier. CNN computes features for a proposal by identifying the object within the proposal. As the performance of CNN strongly depends on the network topology, we investigate the effect of both number of layers and neurons on CNN performance. The obtained results show that CNN can achieve better detection accuracy when it has six convolutional layers, three max-pooling layers, two fully connected layers, and a softmax layer.

To the best of our knowledge, currently there are no existing works on using real data from four sensors to detect and classify the objects in maritime environment. Moreover, the existing object proposal generation methods extract proposals based on grouping pixels or window scoring. Our framework generate more reliable and dense proposals based on fusing the detection results of multiple sensor. We demonstrate the efficiency of our framework on a real dataset which was collected in the Finnish archipelago by a ferry equipped with four kinds of sensors. The data was collected for the Advanced Autonomous Waterborne Applications Initiative (AAWA) project [5]. This project tested sensor arrays in a range of operating and climatic conditions in Finland and has created a simulated autonomous ship control system which allows the behaviour of the complete communication system to be explored after surrounding object detections. We focus on three main objects in maritime environment: boat, seamark and land. Moreover, our framework is evaluated on the real dataset for three main tasks of proposals generation, detection and classification. Experiment results show that our framework significantly outperforms recent well-known proposal methods selective search [4] and EdgeBoxes [6]. In addition, our framework achieves around 76.5% and 97.5% total accuracy in tasks of object detection and classification with few proposals, respectively. It also outperforms all other methods based on individual sensor when we fused the proposals of the detection results of four sensors. The remainder of the paper is organized as follows. Section II discusses some of the most important related works. The proposed framework is presented in Section III. Section IV describes the implementation issue of our framework and pre-processing tasks on the dataset. The experimental results are shown in Section V. Finally, we present our conclusions in Section VI.

II. Related Work

Recent advances in object detection have been driven by the success of object proposals methods. These methods generate relatively set of candidate proposals that likely contains the

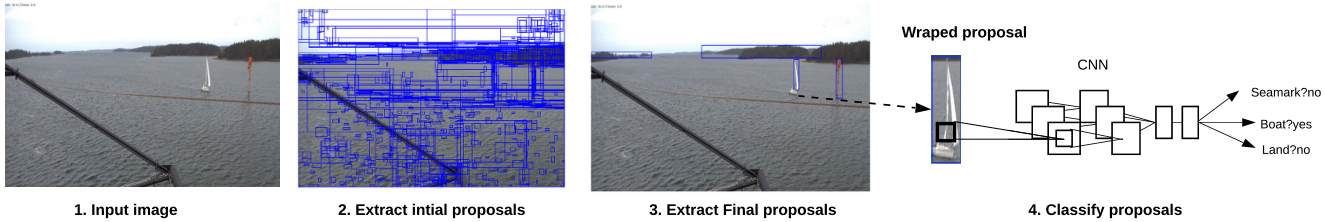

Fig. 1. Overview of the proposed framework. Initial proposals with 933 candidates are first generated by SS and are then filtered using proposal fusion of multiple sensors. After that, the final proposals are classified using CNN.

objects of interest. Sliding window [7] has been employed in a wide range of object detectors as it captures all possible locations by using windows with varied scales and ration to scan through the image. However, it has high computation and time complexity by searching whole image regions. To considerably reduce computation, object proposals methods have been proposed to avoid exhaustive sliding window search across images. These methods can generally be divided into two main categories: grouping methods and window scoring methods [8]. Grouping methods aim to produce multiple segments that are likely to corresponds to objects. The window scoring methods indicate score for each candidate window according to how likely is contained an object. Among these methods, Selective Search (SS), EdgeBoxes and Region Proposal Network (RPN) have made impressive performance. SS [4] is a well-known grouping proposal method that captures all possible object locations by combining exhaustive search and segmentation. It can efficiently reduce the number of object proposals with high detection accuracy. For instance, SS approximately generates only $\sim$2k proposals in 640$\times$ 480 image data while sliding window generates $\sim$100k. However, SS extract a smaller number of proposals than the sliding window but $\sim$2k proposals is still is a large number. EdgeBoxes [6] has been broadly used as one of the window scoring proposal methods. It generates region proposals form edges as they can sparse informative representation of an image. Moreover, EdgeBoxes can provide the best trade-off between running time and proposal quality. However, it consumes much running time in the region proposal step than the detection network. RPN utilizes supervision information to obtain more dense proposals. These extracted proposals can represent different and complementary information on images. Therefore, fusion of these proposals can provide more information in order to avoid the risk of object missing.

An efficient object proposals method able to provide possibility for using an expensive classifier such as deep models for each region by pruning away false positive before classification. Convolutional Neural Networks (CNNs) [9] have been recently used in the development of object detection and classification as a popular deep learning model. Inspired by the success of applying CNN in many number of challenging image classification problems [1], [3], our framework employed CNN for this purpose. In particular, the series of methods based on R-CNN [1] push forward the progress of object detection significantly. R-CNN [1] first identify region proposals by SS method and then classify $\sim$2k proposals into object categories or background using a CNN. One disadvantage of R-CNN is that it computes the CNN independently on each region proposal, leading to time-consuming and energy-inefficient computation. In order to reduce running time of R-CNN, Faster R-CNN [3] ignores the time spent on region proposals by using CNN for region proposals instead of running a separate SS. Faster R-CNN proposes a region proposal network (RPN) that mainly employs the supervised information to generate proposals.

III. Proposed Framework

A. Object Proposals Generation

To find all possible object locations, this module generate high-quality region proposals by combining data from multiple sensors. The proposed object proposals pipeline consists of two steps: generating initial proposals and filtering. To get initial proper proposals, this module uses Selective Search (SS) [4], which found well-studied for such purpose. SS greedily combines superpixels based on engineered low-level features to generate initial proposals in RGB image data. However, some of these proposals that are most likely to contain objects of interest. The final goal of this module is to yield a set of possible object locations for use in a practical object detection framework. To achieve this, our framework requires to find more dense and reliable proposals from the initial proposals. For this purpose, it uses the data from other three common sensors such for filtering the initial proposals based on detection results of each individual sensor as follows:

IR camera: we applied a feature segmentation on gray-scale images from IR camera. The feature segmentation is based on both gradient and intensity-based feature extraction. Image areas with significant and uniform horizontal gradients, which are not typical for the water surface are extracted with gray-scale convolution and threshold operations. Moreover, high-intensity features (hot objects) are extracted with a threshold operation. The results of the gradient and intensity evaluation are combined into a single binary feature image. After IR camera images have been segmented, they are stitched into a single binary image and a Connected-Component-Labeling (CCL) operation is applied to extract rectangle bounding boxes for each binary object. The bounding boxes are then given to a standard Kalman filter to remove temporal noise, such as blinking or very short-lived features.

Radar: first the marine radar data frames is mapped from polar to 2D cartesian coordinates. Radar data contains level of echo signal strength for each determined angle and radius. Then an intensity threshold filtering is applied to remove weak echos and extract the objects from radar data that is a bunch

of points wherein the signal strength has been high enough. The intensity threshold is determined through empirical experiments. After that the extracted objects are mapped into its corresponding RGB image via Perspective Mapping (PM) method. A morphological dilation technique is applied on the mapped data points to cluster the detected objects into more coherent groups. Finally, the boundary of the set of points for each obtained group is extracted that is called bounding boxes for objects.

LiDAR: the same process is applied to extract the bounding boxes from LiDAR data. After applying a low-pass/median filter on LiDAR data, the height component of LiDAR data is discarded and the x/y-coordinates of the LiDAR point cloud features are similarly mapped to RGB image via PM method. The mapped points are clustered and boundary of the set of points is extracted.

The generated bounding boxes proposals by all sensors are mapped on RGB input image for filtering out initial proposals via SS. Then, the final set of proposals is extracted from the RGB image and other three sensors is organized by considering the overlap between each two data modalities. Each initial proposal p_i by SS is assumed as a final proposal if it is overlapped by at least one of the neighboring sensor proposal p_j according to the following function:

$$f(p_i, p_j) = \begin{cases} 0, & \text{if } IoU < \alpha \\ 1, & \text{if } IoU >= \alpha \end{cases} \quad (1)$$

where α is Intersection of Unit (IoU) threshold between two proposals (bounding boxes) and is determined experimentally. IoU is intersection of two proposals divided by their union.

$$IoU(p_i, p_j) = \frac{S_{p_i} \bigcap S_{p_j}}{S_{p_i} \bigcup S_{p_j}} \quad (2)$$

where S_p represents the area of proposal p.

The pseudocode in Algorithm 1 creates a final set of object proposals $finalPros$ from $initPros$ based on information from four sensors. First, the SS method is employed to create a set of initial proposals $initPros$ (Lines 1). Then, the proposals are extracted by each sensor based on individual object detection method and then are mapped on the RGB input image (Lines 2-4). After that, the algorithm iterates over $initPros$ to find neighbouring sensor proposals of the initial proposal p_i (Lines 6-9). Finally, the algorithm added only the proposal p_i to the final set of proposals $finalPros$ if it is nearby at least one proposal generated by a sensor (Line 10-25). We implement "nearness" by assigning p_i to one of sensor proposal if the IoU overlap is greater than a threshold α (which we set to 0.6 using a validation set). All unassigned proposals are discarded. The algorithm outputs the final reliable proposals $finalPros$.

Algorithm 1 Object Proposals Algorithm

Input: RGB image
Output: Final set of obejct proposals $finalPros$

1: Obtain initial proposals $initPros = \{p_1, ..., p_n\}$ using SS [4]
2: Obtain IR camera proposals $irPros$
3: Obtain radar proposals $radarPros$
4: Obtain LiDAR proposals $lidarPros$
5: $finalPros = \emptyset$
6: **for** $p_i \in initPros$ **do**
7: nearIrPros ←neighbouring proposal pair $(p_i, irPros)$
8: nearRadarPros ←neighbouring proposal pair $(p_i, radarPros)$
9: nearLidarPros ←neighbouring proposal pair$(p_i, lidarPros)$
10: **for** $p_j \in nearIrPros$ **do**
11: **if** $IoU(p_i, p_j) >= \alpha$
12: $finalPros = finalPros \cup \{p_i\}$ **then**
13: **end if**
14: **end for**
15: **for** $p_j \in nearRadarPros$ **do**
16: **if** $IoU(p_i, p_j) >= \alpha$
17: $finalPros = finalPros \cup \{p_i\}$ **then**
18: **end if**
19: **end for**
20: **for** $p_j \in nearLidarPros$ **do**
21: **if** $IoU(p_i, p_j) >= \alpha$
22: $finalPros = finalPros \cup \{p_i\}$ **then**
23: **end if**
24: **end for**
25: **end for**

B. Object Proposals Classification

This module is dedicated to the development of a classification method which can classify the objects within the extracted final proposals by the object proposal generation module. It takes a set of object proposals as input and outputs are an objectness score and the class corresponding to the proposals. For this purpose, a Convolutional Neural Network (CNN) extract a fixed-length feature vector from each proposal. Based on the preliminary experiments, we develop a CNN consists of an input layer, six convolutional layers, three pooling layers, two fully-connected layers, and an output layer (Fig. 2). The input to the CNN is the warped regions of extracted final proposals. Each region is passed through a set of convolutional layer, each of which activates certain features (features maps) from the images. Finally, the feature map is obtained by operation of nonlinear activation function such as Rectified Linear Unit (ReLU). Each convolutional layer is followed by a pooling (down-sampling) layer to reduce the dimensionality of feature maps and computation in the network based on a fixed rule. Our CNN uses one of the most common rule that is called max-pooling. At the end of CNN, there are two fully-connected layers. These layers take the feature maps of images from the last max-pooling layer and generate an n dimensional vector where n is the number of class. This vector contains the probabilities for each class of any image being classified. In order to reduce overfitting in the fully-connected layers, we employed a popular regularization method called dropout that proved to be very effective [10]. To update network weights, we use Adam as a popular optimization algorithm instead of classical stochastic gradient descent. The main benefit of Adam is a little memory requirement as it only require first-order gradient. The main goal of training CNN is to minimize the mean square error. After scoring each final proposal by CNN, we predict a new bounding box coordinate for the object in each region proposal based on a simple linear regression. In fact, adjusting a tighter bounding boxes can improve object detection performance. We used the similar way by RCNN that regress the computed features by the CNN [1]. A set of class-specific bounding box regresses is learned to predict the bounding box for each class.

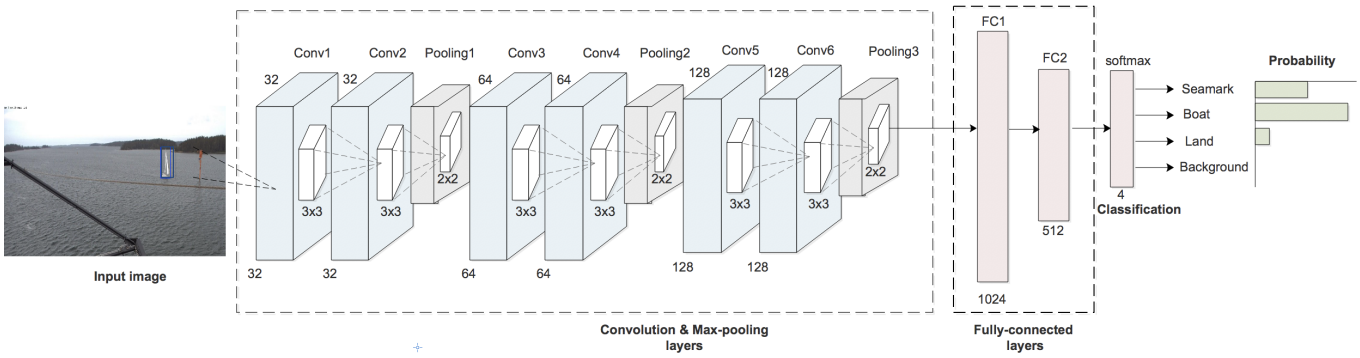

Fig. 2. Proposed CNN architecture for object proposals classification, see text for details.

IV. Experimental Setup

A. Dataset Description

To evaluate our proposed framework, we collected a real dataset with a ferry operating in the Finnish archipelago [5]. The deployment locations included the open sea. This dataset represent various weather conditions from 4^{th} October 2016 to 25^{th} July 2017. The dataset represents the measurements of four sensors: RGB camera, IR camera, LiDAR and radar. Two RGB cameras and IR cameras installed at the front right and left of the ferry. These HD visual range RGB cameras capture videos via 5MP image sensor with 92° lens angle. Image resolution was full-HD (1920x1080). Moreover, we recorded videos by two low resolution thermal IR cameras. The images resolution that we can get from the IR camera is 512×640 working between -50°C to 70°C temperature. Frame rate for RGB and IR cameras is 2 and 4 frames/sec, respectively. The radar range is upto 1.2KM with angular sampling interval of 0.4°. To collect 3D point cloud data, we use OPAL 3D LiDAR scanner from Neptec technologies.

B. Pre-processing

To train the proposed CNN for object classification, we use images of three interest objects from RGB cameras. These images are automatically generated by creating minimal bounding boxes around an object that is detected based on RGB camera. To detect objects based on RGB camera, we extract local (horizontal) gradients clusters differing from the typical water surface. Large high- intensity features (discarding image saturation) are also extracted with a threshold operation and combined logically with the gradient data. As the intensity gradients approach cannot efficiently detect and track some small objects such as seamark that hardly is distinguishable from the water, a red/green feature segmentation approach is employed. In addition, the image-based evaluation and processing tasks are applied on RGB camera data in order to take into account environmental issues such as day or night conditions and sun glare induced sensor saturation. Finally, the object detection is performed by extracting the binary features from RGB cameras. In order to use the obtained images by the CNN, we first convert the image data into a form that is compatible with the CNN (its architecture requires inputs of a fixed 32 by 32 pixel size). According to the input size of the CNN, we wrap all pixels in the tight bounding box around it to the required size (32). On the other hand, we anisotropically scales each object proposal to the CNN input size. Finally, the following pre-processing steps were performed on the images:

1) Feature Normalization: the numeric features must be normalized for removing the effect of original feature value scales. The pixel values are in the range of 0 to 255 for each of the red, green and blue channels. The pixel values were normalized into the range 0 to 1.

2) Class encoding: the non-numerical class types are converted into the numeric categorizes. We used one hot encoding to convert three categorical classes into three binary classes, with only one active.

3) Data augmentation: we create more images from the original images via a number of random transformations. Random transformations were applied on the original training images including rotation, cropping, swirl, vertical flip and horizontal flip. The number of images for each class after data augmentation is 4572, 3759 and 4757 for seamark, boat and land, respectively.

C. Choice of Hyperparameters

The performance of CNN strongly depends on the value of hyperparameter. For this reason, we tune different hyperparameter to select the best value of them. We utilize 10-fold cross-validation approach subjected to the dataset of 13,088 images. After the best value of hyperparameters are selected, the final model is trained with all 13,088 images. The performance of CNNs highly depends on the network topology. For this reason, we tried to find the network topology that is optimal to our object detection problem. The number of convolutional layers is varied from 2 to 6 in steps of 2. The layers' structure of proposed CNNs is described in Table I. On the other hand, Table I shows that which layers of CNN3 are utilized in CNN1 and CNN2. Experimental results demonstrate that we can get 91.6%, 94.5% and 96.2% test accuracy for $CNN1$, $CNN2$ and $CNN3$, respectively. Therefore, CNN with six convolutional layers is the optimal model (CNN3). In order to avoid overfitting, we use dropout in each fully-connected layers of CNN. The value of the dropout ranges from 0.0 to 0.9. We see that as dropout is 0.5, the model can get better accuracy. Moreover, the value of batch size and epochs are 25 and 10, respectively. In addition, the best optimizer for our neural network model in order to learn properly and tune the internal parameter is the Adam based on our experiments. Moreover, we tune the learning rate parameter that is used

TABLE I. PROPOSED CNNS FOR IMAGE CLASSIFICATION

Name	Conv1	Conv2	Pooling1	Conv3	Conv4	Pooling2	Conv5	Conv6	Pooling3	FC1	FC2
CNN1	✓	✓	✓								✓
CNN2	✓	✓	✓	✓	✓	✓				✓	✓
CNN3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

in the Adam with the grid search. Learning rate controls the speed of wight updating at the end We tried a suite small standard learning rate from 0.001 to 0.3 in steps of 0.1. The best performance of the model is achieved when the learning rate is 0.001.

V. EXPERIMENTAL RESULTS

Our framework is evaluated on a real test dataset which is collected by a ferry operating in the Finnish archipelago [5]. The number of three main interest objects in the test dataset for seamark, boat and land are 266, 103 and 1800, respectively. For a given RGB test image, our proposed framework firstly performed SS or EdgeBox to obtain initial object proposals. Then, it finds more dense and reliable proposals based on the information from other three sensors. After that, each proposal is warped and passed into the pre-trained CNN in order to classify the object within the proposals. In addition, the RGB images of test data set was manually tagged in order to provide a ground truth reference.

Several experiments have been done to evaluate our framework based on the real test dataset. In the first experiment,we investigate the effect of different region proposal methods in our framework performance in terms of number of proposals and running time. The benchmark proposal methods include fast selective search, quality selective search, single best strategy, EdgeBoxes, our framework based on SS and EdgeBoxes proposals and RGB-based detection. Table II shows the result of benchmark methods and our framework on 1000 RGB images with size 1080×1860. The results show that our framework extracts 9,717 proposals by filtering initial proposals by SS based on detection results of multiple sensors. It means it can reduce the number of proposals 386, 1046, 97 and 242 times less than fast selective search, quality selective search, single best strategy and EdgeBoxes. Our framework can reduce the number of proposals 1.5 times if it uses SS (single best strategy) instead of EdgeBoxes. The result shows that RGB-based detection is computationally efficient than SS. However, it has less detection accuracy (see Table III) in comparison with our framework.

TABLE II. COMPARISON BETWEEN PROPOSAL GENERATION METHODS AND OUR FRAMEWORK

Method	**Sensor**	**# Proposals**
Selective Search "Fast"	RGB	3,756,388
Selective Search "Quality"	RGB	10,170,585
Single Strategy	RGB	2,305
EdgeBoxes	RGB	1,447
Ours (based on SS)	R+L+IR+RGB	9,717
Ours (based on EdgeBoxes)	R+L+IR+RGB	13,850
RGB-based detecion	RGB	634

In the second experiment, we demonstrate the impact of proposal fusion of multiple sensor on the detection accuracy. The goal is to ensure that those proposals that accurately cover the desired objects. Table III shows that the detection results based on generated proposals from each sensor, fusion of SS proposals, fusion of EdgeBoxes proposals, R-CNN, Fast R-CNN and Faster R-CNN. The correct detection determines how many of each object is detected correctly. The false detection represents how many of all objects are not detected. The first row of each method shows that how many of objects are detected in each class. For clarity sake, the number of detections is also represented by percentages at the second row. The results show that the detection rate of three objects is improved by our framework based on proposal fusion of multi-sensor in comparison with other methods. We achieved 94.7% , 69.9% and 74.2% for three classes Seamark, Boat and Land, respectively. The false detection rates (23.4%) is due mainly to the noisy radar target detection and the reflection in raw LiDAR data which creates ghost objects. Moreover, our framework can achieve 3.9% higher detection accuracy if it employs SS instead of EdgeBoxes. In the third experiment, we collected the result after testing our trained CNN with on-line test data based on 10-fold cross-validation approach (Table IV). Correct classifications represent well classified objects of three classes when the region proposals obtained from each sensor and our framework based on proposal fusion. False classifications show the number of percentage of object that are miss-classified for each class. When the CNN is applied on the regions obtained by our framework, the classification accuracy is 88.8%, 100% and 99% for seamark, boat and land, respectively. Moreover, we can achieve a high classification accuracy from CNN on region proposals obtained by each sensor. Therefor, the classification rate of all objects by the proposed CNN are nearly perfect (86-100%).

TABLE IV. CLASSIFICATION RESULTS IN REAL TEST DATASET

Method	**Correct**			**False**	**Total accuracy**
	Seamark	Boat	Land	All	All
Radar based detection	81	n/a	1107	23	1188
	83.5%	n/a	99.3%	1.9%	98.1%
Lidar based detection	17	n/a	141	2	158
	100%	n/a	98.6%	0.0%	98.7 %
IR based detection	140	37	515	158	692
	77.7%	100%	81.3%	18.5%	81.4%
RGB based detection	158	53	343	80	554
	87.7%	100%	85.5%	12.6%	87.3%
Ours	224	72	1324	41	1620
	88.8%	100%	99.0 %	2.4%	97.5%

Fig. 3 shows an example of detection using SS, EdgeBoxes and our framework on the input image illustrated in Fig. 1. The total number of generated proposals by SS, EdgeBoxes and our framework is 933, 1989 and 4 in 1080×1860 image, respectively. Beside reducing the number of proposals, we can get more accurate region proposals (bounding boxes) of the image corresponding to objects by comparing between SS and our framework (see right figures in each row). Meanwhile, the running time for generating proposals by SS, EdgeBoxes and our framework is 2.49, 1.72 and 0.13 seconds for the image illustrated in Fig. 3.

TABLE III. DETECTION RESULTS IN REAL TEST DATASET

Method	Sensor	Correct			False
		Seamark	Boat	Land	All
Radar based detection	R	97	0	1114	958
		36.4%	0.0%	61.8 %	44.1%
Lidar based detection	L	17	0	143	2009
		6.3%	0.0%	7.9%	92.6%
IR based detection	IR	180	37	633	1319
		67.6%	35.9%	35.1%	70.7 %
RGB based detection	RGB	180	53	401	1535
		67.6%	51.4%	22.2%	60.8 %
Ours (based on SS)	R+L+IR+RGB	252	72	1337	508
		94.7%	69.9%	74.2%	23.4%
Ours (based on EdgeBoxes)	R+L+IR+RGB	228	46	1302	593
		85.7%	44.6%	72.3%	27.3%

VI. CONCLUSION

In this paper, we proposed an efficient object detection framework that can successfully identify the location and type of interest objects surrounding autonomous vehicles in maritime environment. The framework first employs SS to generate object proposals in RGB images. Then, the proposals are filtered based on the multiple sensor data in order to improve the robustness and accuracy of object localization. Moreover, we can greatly reduced computational cost by reducing the number of proposals. Finally, the framework uses a convolutional neural network to classify the objects within the final obtained proposals. The performance of our proposed framework is evaluated by conducting experiments with real data obtained by testing sensor arrays in a range of operating and climatic conditions in Finland. The obtained results show that our framework can precisely localize and identify interesting objects using a smaller number of proposals than other methods.

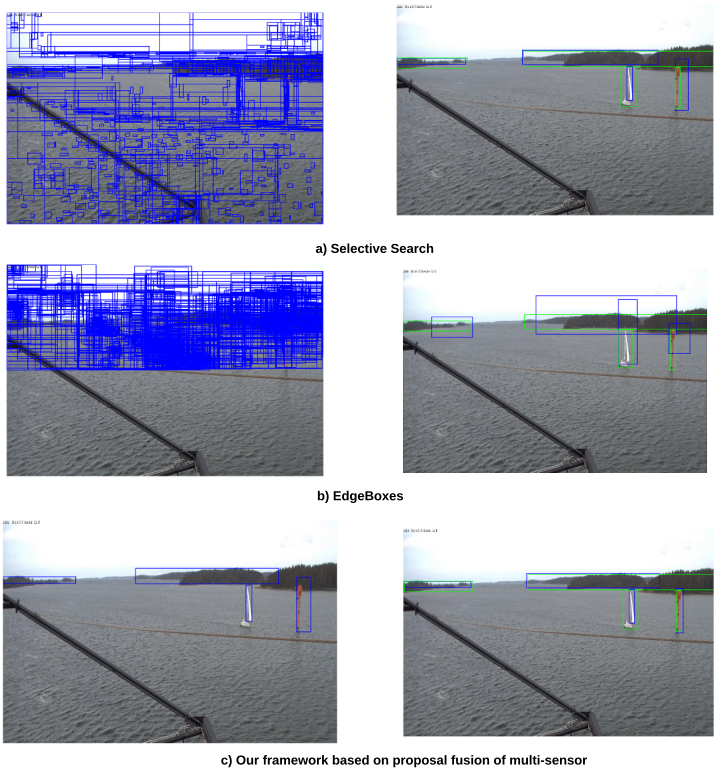

Fig. 3. Qualitative example of Selective search, EdgeBoxes and our framework result on the test set. Left figure in each row shows all generated proposals by each method. Right figure in each row, ground truth bounding boxes are shown in green, respectively. Blue bounding boxes are the closest produced object proposals to each ground truth bounding box.

ACKNOWLEDGMENT

This work is part of the Advanced Autonomous Waterborne Applications Initiative (AAWA) and the New 3D Analytics Methods for Intelligent Ships and Machines projects funded by the Tekes (Finnish Funding Agency for Technology and Innovation).

REFERENCES

[1] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(1):142–158, January 2016.

[2] Ross Girshick. Fast r-cnn. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1440–1448, Washington, DC, USA, 2015. IEEE Computer Society.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, June 2017.

[4] J. R. Uijlings, K. E. Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *Int. J. Comput. Vision*, 104(2):154–171, September 2013.

[5] S. Jokioinen, J. Poikonen, M. Hyvnen, A. Kolu, T. Jokela, J. Tissari, A. Paasio, H. Ringbom, F. Collin, M. Viljanen, R. Jalonen, R. Tuominen, M. Wahlstrm, J. Saarni, S. Nordberg-Davies, and H. Makkonen. Remote and autonomous ships - the next steps. *white paper*.

[6] C. Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.

[7] Woonhyun Nam, Piotr Dollar, and Joon Hee Han. Local decorrelation for improved pedestrian detection. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 424–432. Curran Associates, Inc., 2014.

[8] Jan Hosang, Rodrigo Benenson, Piotr Dollar, and Bernt Schiele. What makes for effective detection proposals? *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(4):814–830, April 2016.

[9] Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, January 2009.

[10] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.