

# High-Speed Maritime Object Detection Scheme for the Protection of the Aid to Navigation

Hyochan Lee<sup>1</sup>, Hyunhak Song<sup>1</sup>, Sungyoon Cho<sup>1</sup>, Kiwon Kwon<sup>1</sup>, Sunghyun Park<sup>2</sup>, Taeho Im<sup>2\*</sup>

<sup>1</sup>Smart Network Research Center, Korea Electronics Technology Institute  
Seoul, South Korea

[e-mail: lhc\_104@keti.re.kr]

<sup>2</sup>Dept. of Information and Communication, Hoseo University  
Asan, South Korea

[e-mail: taehoim@hoseo.edu]

\* Corresponding author: Taeho Im

*Received August 19, 2021; revised September 30, 2021; revised November 19, 2021;  
revised January 11, 2022; accepted February 5, 2022; published February 28, 2022*

---

## Abstract

Buoys used for Aid to Navigation systems are widely used to guide the sea paths and are powered by batteries, requiring continuous battery replacement. However, since human labor is required to replace the batteries, humans can be exposed to dangerous situation, including even collision with shipping vessels. In addition, Maritime sensors are installed on the route signs, so that these are often damaged by collisions with small and medium-sized ships, resulting in significant financial loss. In order to prevent these accidents, maritime object detection technology is essential to alert ships approaching buoys. Existing studies apply a number of filters to eliminate noise and to detect objects within the sea image. For this process, most studies directly access the pixels and process the images. However, this approach typically takes a long time to process because of its complexity and the requirements of significant amounts of computational power. In an emergent situation, it is important to alarm the vessel's rapid approach to buoys in real time to avoid collisions between vessels and route signs, therefore minimizing computation and speeding up processes are critical operations. Therefore, we propose Fast Connected Component Labeling (FCCL) which can reduce computation to minimize the processing time of filter applications, while maintaining the detection performance of existing methods. The results show that the detection performance of the FCCL is close to 30 FPS - approximately 2-5 times faster, when compared to the existing methods - while the average throughput is the same as existing methods.

---

**Keywords:** Aid to Navigation, Maritime, Computer Vision, Object Detection, High-Speed Processing.

---

A preliminary version of this paper was presented at APIC-IST 2021, and was selected as an outstanding paper. This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2021-2018-0-01417) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation) and a part of the project titled 'The advancement of smart aids to navigation facilities(20210636)', funded by the Ministry of Oceans and Fisheries, Korea.

## 1. Introduction

**A**toN (Aid to Navigation) is a direction sign that guides and determines the sea route for a vessel to navigate [1]. Typically, lighthouses, buoys, beacons, etc. exist on the surface of the sea and have textual meanings such as color and number, and the navigator decides a route based on these signs. Therefore, they are called sea traffic lights and play a very important role in ship safety, particularly for major routes.

The buoys consist of batteries, sensors, and communication devices, but not many are equipped with systems such as cameras and RADAR(Radio Detection And Ranging), LiDAR(Light Detection And Ranging) which monitor the surroundings. In the case of NOAA(National Oceanic and Atmospheric Administration) in the U.S., however, cameras are installed on buoys, but they are only used for recording purposes. The main purpose of the surrounding monitoring systems is the safety of the site. Frequent accidents with ships damage buoys and these damaged buoys can collide with another ship in the sea and cause further accidents. Also, buoy field work (such as replacing batteries or sensors) can be a dangerous task for workers if there is no nearby alarm through monitoring. Therefore, sea monitoring is an essential function of safety to prevent damage to buoys, loss of lives, etc.

Sea object recognition - part of the function of maritime monitoring - is an intelligent assist system for autonomous ships [2-4]. MASS(Maritime Autonomous Surface Ship) is a method of automatically recognizing and identifying floating collision risks through a computer rather than relying on human interaction [5-6]. The recent development of computer vision and artificial intelligence technology enable the identification and distinguishing the nature of the object in the image taken with the camera [7-8].

Despite these technological advances, however, few maritime monitoring systems exist in Korea because of the high cost and required high computational consumption. In the case of buoys, they are maintained through their own battery and the battery is replaced periodically by human intervention. If algorithms that require high operations are used in buoys, parallel operations such as GPU(Graphics Processing Units) need to be utilized, which requires a lot of power consumption and frequent battery replacement, making it difficult to maintain. Therefore, the surrounding detection algorithm of a maritime monitoring system should be properly maintained without consuming much computational resources.

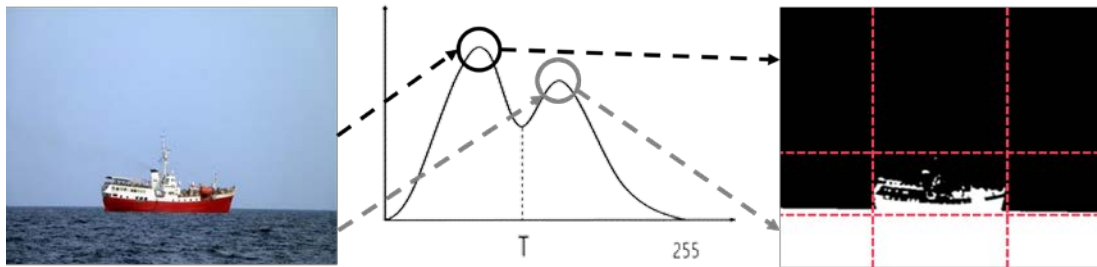
In this paper, we propose a floating object detection method algorithm suitable for monitoring the marine environment. This method assumes a camera is installed on the buoy to detect the objects. In addition, we propose a method to simplify computation by utilizing maritime features to reduce the amount of computation of embedded computers for buoys. The goal is to achieve the same level of detection performance of existing algorithms which are commonly used, although computation is reduced.

## 2. Related work

### 2.1 Binarization

Horizontal line detection at sea is an effective way of detecting items floating in the sea by separating the sea and background areas present in the image [9-11]. In general, horizontal line detection is achievable by utilizing either the Edge method or the Binary method. The Edge methods can clearly separate boundaries but are expensive and suitable for complex environments. On the other hand, the Binary method is not very accurate but is suitable for simple environments. In the case of a sea environment, we apply the Binary method because

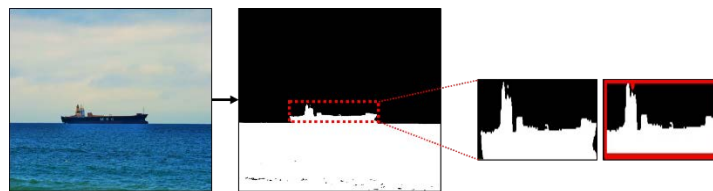
there are relatively few image changes when compared to land, as shown in [Fig. 1](#). Binary separation includes fixed and flexible threshold methods. A flexible threshold method requires more computation than do fixed methods but are advantageous in distinguishing between the ocean and sky. Therefore, we utilize Otsu's method which is a typical flexible threshold algorithm of the binary method to find areas of interest in the marine environment [\[12\]](#). The way of finding the region of interest after Otsu's method is as follows: first, the background is saved as black (0), and the sea and the ship as white (1) in a memory buffer. The region of interest are sequentially found by searching pixel values corresponding to the sea and the ship based on the image Y-axis. Based on the Y-axis, the first pixel value to be found is fixed at the top, and the value to be found later is fixed at the bottom to set the region of interest by drawing a straight line. By searching for pixels in the same way as the Y-axis on the X-axis, the algorithm finds the side lines of both sides of the ship, and set the line, the area of interest of the ship is finally found.



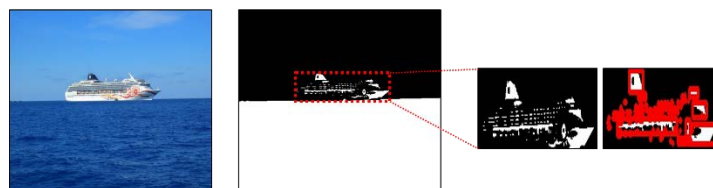
**Fig. 1.** Otsu's algorithm in the marine environment

## 2.2 Dilation

First, we need to separate the sea from the sky through the binary method and then find the objects on the sea. If the object is binarized properly, it will be simple to detect as shown in [Fig. 2](#). However, there are many variables in offshore objects, which may be crushed by the weather, the color of the ship, and the light reflection, as shown in [Fig. 3](#). In such cases, floating sea objects cannot be detected properly because they are recognized as multiple objects rather than one object.



**Fig. 2.** An example that the object is properly detected and distinguished



**Fig. 3.** An example that the object is not properly detected and distinguished

This problem can be solved by applying an dilation algorithm to the corresponding image region [13]. The dilation algorithm performs matrix operations through cross-like kernels in the pixel region and as a result 1 pixel is added up to the existing object as shown in Fig. 4-5 shows that the application of an dilation algorithm to the results in Fig. 3: from the first figure on the leftmost, the algorithm is applied once, twice and three times. The more expansion, the more regions are grouped into one object. Thus, the dilation algorithm is efficient in the marine environment and helps to locate objects better.

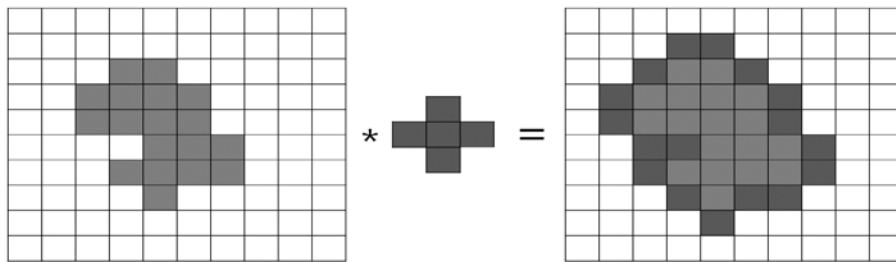


Fig. 4. The result of the dilation algorithm



Fig. 5. The result of applying the dilation algorithm to the marine object

### 2.3 Labeling

Labeling is utilized to locate marine objects after dilation filters. Labeling identifies those pixels are connected and locates them [14]. As shown in the leftmost in Fig. 5, if the pixel is separated, it is recognized as multiple areas, not one area, and it is not accurately located. On the other hand, the rightmost in Fig. 5 recognizes that it is one area and finds the object location which a high degree of accuracy. Labeling is performed as follows:

First, the following filters are used to verify the adjacency of pixels, as shown in Fig. 6 From the central pixel  $P(x,y)$ , 4-Connected checks each direction of  $P(x-1)$ ,  $P(y-1)$ , 8-Connected also checks each direction of  $P(x-1)$ ,  $P(y-1)$ ,  $P(x-1,y-1)$ , and  $P(x+1,y-1)$  to determine whether the pixels are connected or not.

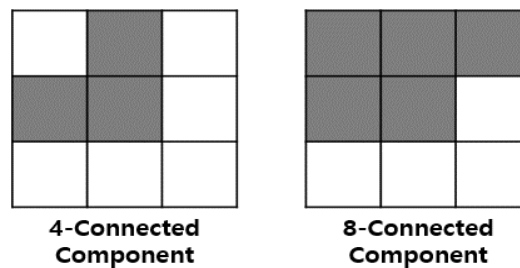
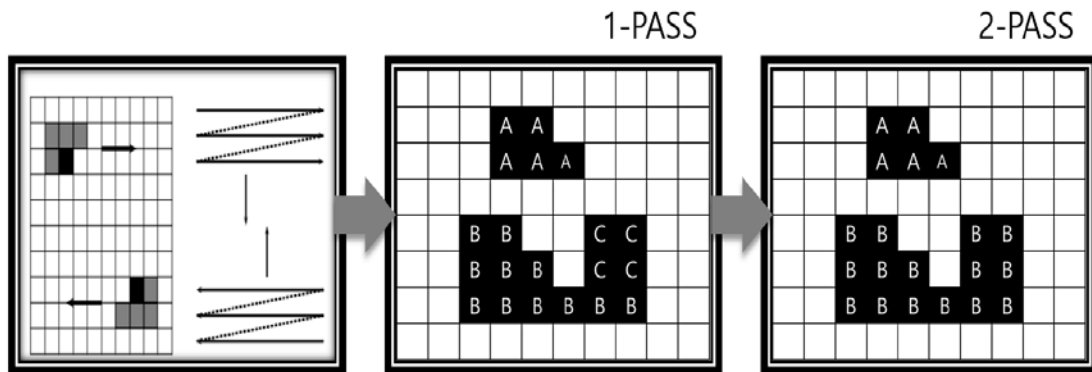


Fig. 6. Neighborhood filter for Labeling

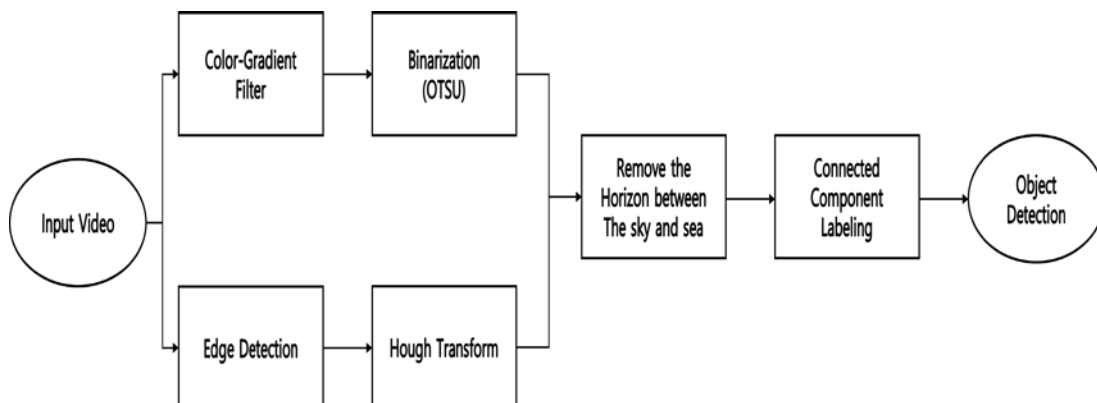
Typically, the Labeling process is performed twice in total. The first figure of **Fig. 7** shows the process of 1-PASS performed through an 8-Connected filter and the second figure shows the result of it. In the result, there are two objects, but the computer recognizes it as three objects. This is because the filter could not check the connectivity because there is a hole in the middle. Therefore, at the hole, these additional operations are required: the fact that B and C are adjacent using the equivalent table. At the end of 1-PASS for the entire image, 2-PASS can replace C with B using the data in the equivalent table and allocate label numbers to objects. Finally, we can find the location of marine objects by the Labeling process.



**Fig. 7.** Labeling process of binary connection factors

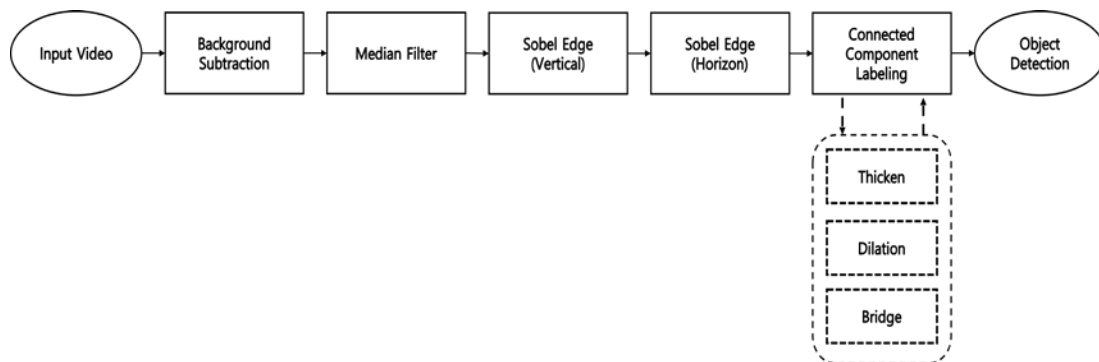
## 2.4 Literature review

In Paper 1 [Detection and tracking of ships with open sea with a Rapidly moving buoy-mounted camera system], it was proposed to detect horizontal lines using two methods: Binary and Edge, as shown in **Fig. 8** [15]. It separates the horizon from the vessel by detecting the horizon and removes the horizon area from the binarized image. Finally, the object is detected through Labeling. This method finds the region of interest well and utilizes the Labeling CCL (Connected Component Labeling), but as mentioned in 2.2 the object is not properly detected in sensitive color environments such as weather effects, light intensity, and ship colors.



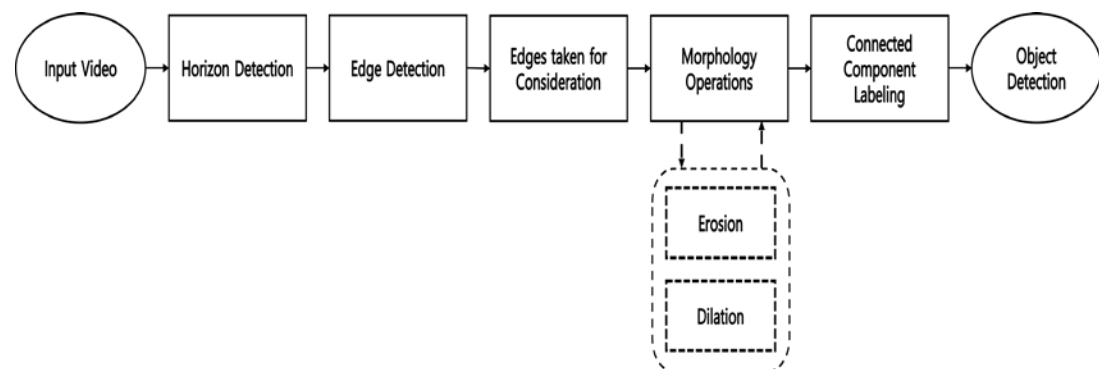
**Fig. 8.** The flow chart of the ship tracking

In Paper 2 [Multiple Ship Detection and Tracking Using Background Registration and Morphological Operations], the dilation algorithm mentioned in Section 2.2 is used to find sea objects by combining with CCL algorithms, as shown in Fig. 9 [16]. In other words, the method detects objects by restoring back to one original area which was separated from the marine environment. However, this method applies dilation kernels throughout the whole image and has the disadvantage of increasing computation resources because it requires at least 2-3 operations depending on the marine environment.



**Fig. 9.** Ship detection by the dilation algorithm and Labeling the connected factors

Paper 3 [Detection of marine vehicles in images and video of open sea] utilizes dilation and morphology techniques of erosion algorithms, as shown in Fig. 10 [17]. The erosion region shrinks the pixels, and the dilation expands the pixel region. Erosion and dilation techniques use different filters and conduct convolution with the entire image. At this time, erosion techniques are applied first to eliminate small noises after binarization. Then dilation techniques expand the separated regions of one object and presents it into one connected form. The object is then detected by applying the Connected Component Labeling. This minimizes noise to reduce the unnecessary effects of object ship detection.



**Fig. 10.** Marine vehicle detection by closing and Labeling the connected factors

While this method can prevent noise that can occur in existing dilation methods, it is considered inefficient because it requires twice the computational amount of the conventional dilation methods and it is more suitable for removing noise in complex environments than in simple environments such as marine environments.

Paper 4 [Design of Video Pre-processing Algorithm for High-Speed Processing of Maritime Object Detection System and Deep Learning based Integrated System] proposes multi-connect element Labeling to increase the performance of the previously proposed papers and to reduce computation, as shown in Fig. 11 [18]. As shown in Fig. 8, multi-connect element Labeling is a two-time iterative process of Connected Component Labeling. Through the first process, an object region in the input image is detected, the region is expanded by a specific margin using information such as pixel coordinates and size of the region, and pixels are filled by the object region (i.e., white). The second step detects one object connected through dilation. This does not perform multiple dilation algorithms and does not process unnecessary regions, which can simultaneously extend the region while eliminating noise within the detection region. However, this method also performs filters of linking element Labeling twice, and there are overlapping operations, resulting in unnecessary computations. The large amount of computation required, determines the performance and power consumption of embedded boards.

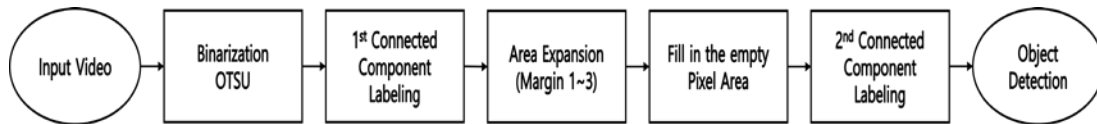


Fig. 11. Multi-Labeling the connected factors

Therefore, the aim of this paper is to reduce unnecessary computational volumes and procedures and to process them at a faster rate, while maintaining the object detection performance of existing algorithms. In this paper, the existing algorithms are categorized into 4 groups: 1) CCL (Connected Component Labeling), 2) DA (Dilation Algorithm), 3) CA (Closing Algorithm), and 4) ML (Multi Labeling). For the 'Aid to Navigation' safety, the detection performance needs to be maintained and the embedded board must be operated in order to be installed on buoys. The fast-processing software must be operated with low power and computation for practical operation and commercialization.

## 2.5 Application

The high-speed object detection proposed in this paper is also very useful for marine object recognition. The CNN (Convolution Neural Network) takes the same role as do human eyes in the recent deep learning-based image recognition technologies and it is simply available to recognize various marine floating objects (i.e. ships and buoys) [19] as shown in Fig. 12. CNN creates weights through data learning of desired objects and then can recognize them by use of its neural network.



Fig. 12. Application of CNN-based marine object recognition



[20] demonstrated that image pre-processing before the input to the deep learning neural network can efficiently detect objects. As a similar study, [18] conducts deep learning for only object candidates which are detected by image pre-processing as shown in Fig. 13. It is important to find object candidates rapidly and accurately and input them to the deep learning system to improve the system performance.

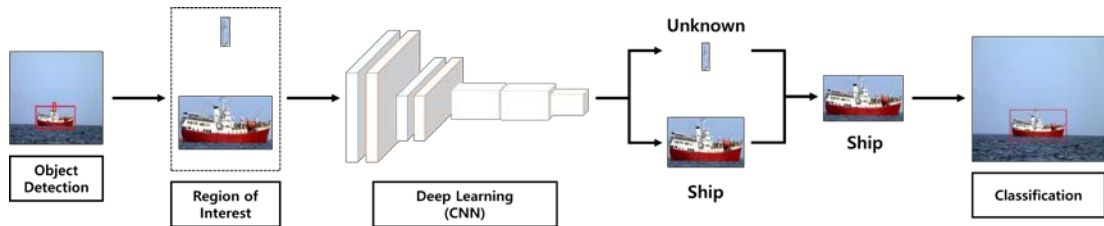


Fig. 13. Object detection before the deep learning-based object recognition

Therefore, chapter 3 in this paper proposes Fast Connected Component Labeling as a pre-processing method which can find the candidates more rapidly and accurately in order to improve deep learning-based marine object recognition.

### 3. Proposed Fast Connected Component Labeling

The problem of the existing methods is that CCL is not properly detected due to the external maritime variables. Hence, it is more efficient to perform after the preprocessing through the DA method. However, as mentioned in section 2-2, it is much more reliable to perform several times rather than only once. Therefore, the Fig. 14 is the ideal marine object detection method.

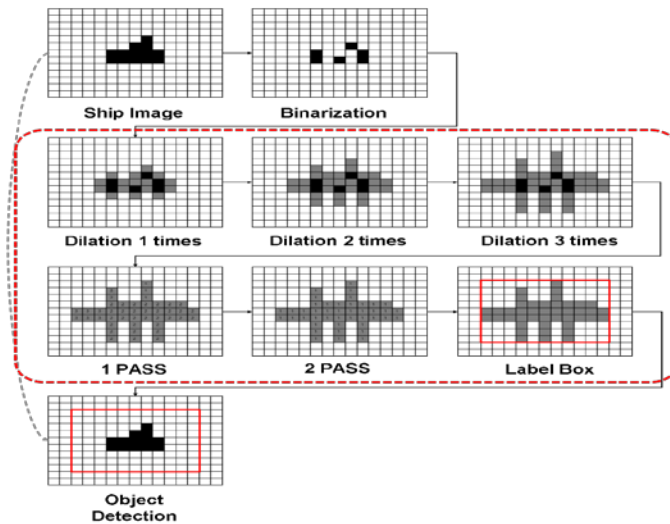


Fig. 14. Overall process of marine object detection

However, since this method requires that DA and labeling must be performed several times, there are many redundancy procedures and unnecessary computational elements. CA and ML is inefficient in terms of computer computation because they perform several times. In detail, When an image is entered, binarization is performed through the Otsu's algorithm.



Approximately  $N$  times of DA is performed on the binary images. Based on the result of DA which we performed on the marine environment image in section 2.2, three times of DA is not affected by the environment, so we assume three times of DA in this paper. There is also a way to significantly expand the DA kernel and process up to three at a time, but since the kernel size becomes larger, the number of computations on the image is identical. After this DA, the 2-PASS process of Labeling is used to locate and detect objects. The reason for using the same features multiple times is to increase the performance of object detection, which will result in higher power consumption as the computation is performed on the entire image, reducing throughput, and increasing computation. Therefore, in this paper, we propose ways to reduce unnecessary procedures and minimize computations. Operations must be performed on all image pixels for each single function, where all pixels must be accessed and processed at least five times. Therefore, processing one time with minimal pixel access can reduce unnecessary operations and we propose a high-speed method which excludes redundant operations. The proposed algorithm is described in four stages. This is named as the FCCL(Fast Connected Component Labeling) in this paper.

First, the first process is shown in Fig. 15. After receiving images, the process of Otsu's binarization is applied in the same way as in the previous methods. However, unlike the existing method which performs DA and then CL later (15-A), the DA is not performed immediately. Instead, the 1-PASS process of the label is performed (15-B). The 1-PASS process uses the 8-Connected Filter which is described in the previous chapters. After 1-PASS is performed, the separated objects are given each label number. The existing algorithm (CCL) performs 2-PASS at this stage, but in this study, instead, each isolated object records each coordinate point in a memory. The data recorded via the LUT stores the corner part of each pixel coordinate. The corner section consists of group 1 (minimum value on the X-axis, minimum value on the Y-axis) and group 2 (maximum value on the X-axis, maximum value on the Y-axis). In the first process, this study performs the process of storing information of groups 1 and 2 in the LUT first without performing 2-PASS immediately.

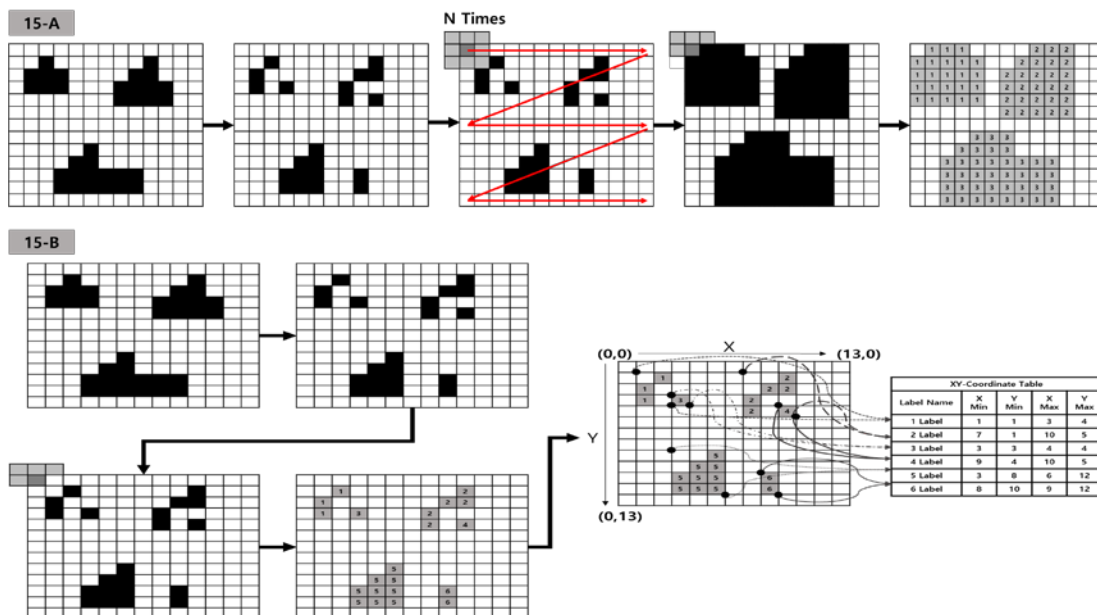


Fig. 15. The process of the FCCL (1)

The second stage processes the values stored in the LUT as shown in Fig. 16. For the coordinates of the six objects detected, it expands the stored values of group 1 and group 2. The figure only increases 1 Pixel to provide the simple example. In practical implementations, 2-3 Pixels are increased. (This is referred as 'Margin'). The DA process moves Group 1 from the existing coordinates to the upper left by the Margin, and Group 2 to the lower right by Margin. The second process ends by updating the expanded data values through the LUT. This method performs an identical role as the DA method described in Fig. 14. The DA operation only takes place in the LUT and only the arithmetic operation takes place in the memory. Therefore, it is a stored LUT rather than existing processing methods directly in image pixels – i.e. a processing method within memory, enabling high-speed processing while applying the DA.

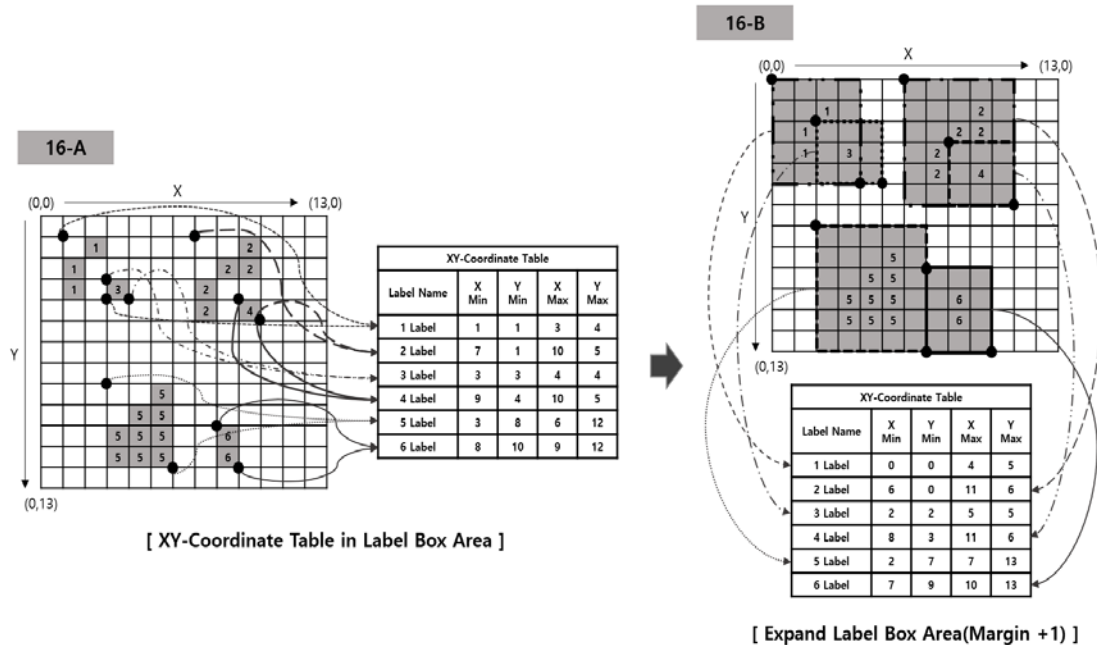


Fig. 16. The process of the FCCL (2)

After the DA process using the LUT, post-processing is required as shown in Fig. 17. Post-processing refers to the process of merging regions of detected boxes, which correspond to the 2-PASS process. (Here, light grey is referred to the criterion area and dark grey is referred to the comparison area.) For the expanded pixels, there are three types of case detection. 1) parts within a criterion, 2) inclusion of a criteria part, and 3) point of contact of the criterion part. For these three conditions, the coordinates of the vertex in the lower right corner is changed according to the three cases, as shown in 17-B and 17-C. Thus, the third process ends by updating the changed points to the LUT. The reason why this overlapping extended regions occurs is that adjacent pixels have caused pixels to be separated by binaries and this is highly likely to be an identical object. Therefore, we propose a process for integrating into one object through merging in those three conditions. This process is almost like the role of 2-PASS in Fig. 14 and this method is handled within the LUT (i.e., within the memory).

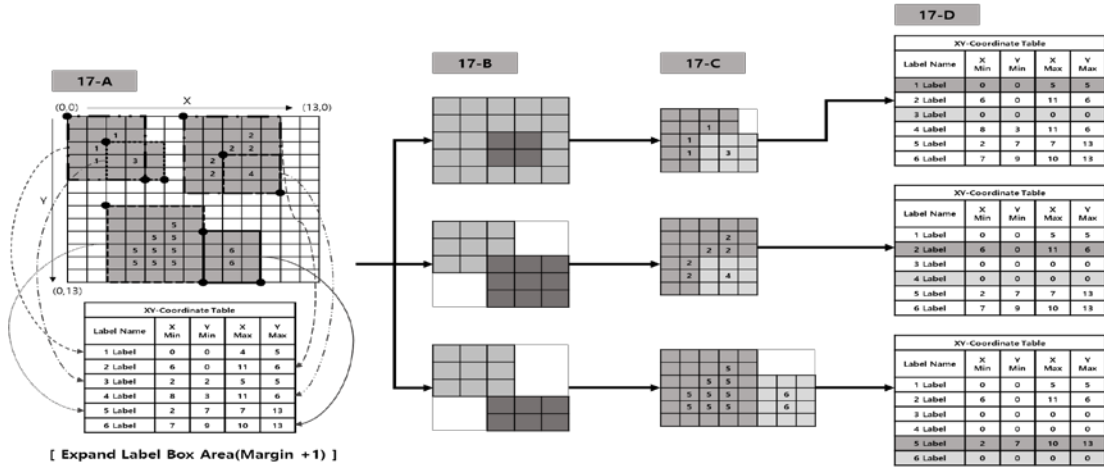


Fig. 17. The process of the FCCL (3)

At the end of post-processing, the object location is detected through the integration of the LUT as shown in Fig. 18. As shown in 18-A, 1 Label is set as the criterion area and 2 to 6 Labels are set as the comparison area, it figures out if it can be merged or not. Merge conditions are handled if they fall within the three conditions, and the coordinates are updated and stored in the final table which is 18-D, according to the merge conditions. For example, 1 Label and 3 Label of 18-A are overlapped and corresponds to condition1, therefore it is combined into the criterion area Table. The criterion area of 18-B - 2 Label - is combined with 4 Label, which falls under the second condition. 5 Label of 18-C is combined with 6 Label, which falls under the third condition. If there is no merge condition within the LUT, it is recognized as one object and then moved on to the next label. Since the merging process is integrated into the criterion area, the LUT in the comparison area indicates that the object has been merged into one through '0' processing. Lastly, the final merged coordinates can be identified by the 18-D LUT, and the image can be boxed and output to the monitoring system, as shown in 18-E.

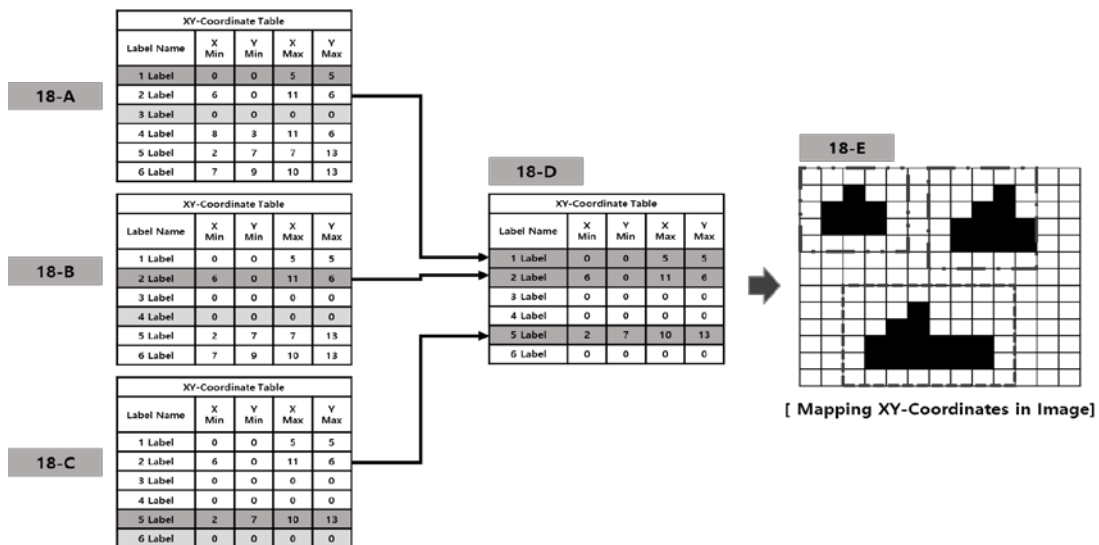


Fig. 18. The process of the FCCL (4)

In summary, the FCCL operates as a flowchart as shown in Fig. 19. The image is entered and binarized in the same way as the existing algorithm(Otsu's method). The existing methods(which conducts DA and then CL later) perform the DA and then processes the label, but the FCCL performs the label first. Instead, it stores the coordinates of the separated objects into the LUT. Then, the LUT is updated by expanding inside the memory through stored coordinates. In addition, post-processing is performed to perform 2-PASS, which is also performed in the LUT. Three situations occur for DA and merge is performed according to those conditions. In order to perform the conditional merge, the LUT is updated and integrated on its own. The final coordinates of the merged LUTs, the position of the objects in the image is stored and then the marine objects can be detected by displaying them in the image. Existing algorithms(CA and ML) show the high volume of computation since they access and process multiple times throughout the whole image, however, FCCL reduces the computation and processes at high speed by pre-identifying isolated object locations, storing them in LUTs, and updating only a portion of coordinate values through internal memory.

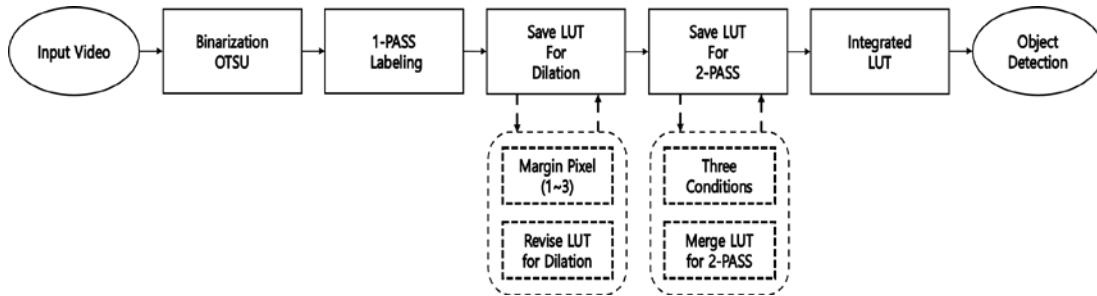


Fig. 19. The flow chart of the FCCL

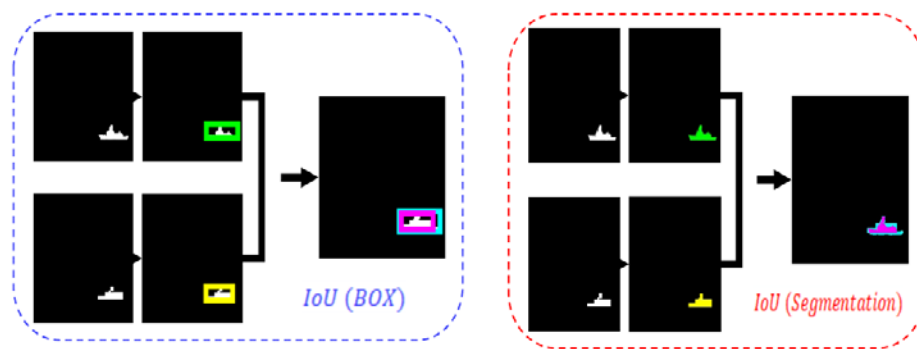
#### 4. Experiment Results

The experiment analyses the FCCL with the performance and speed of detecting nearby vessels under the assumption that buoys are monitoring the marine environment. Earlier in Section 2.4, we experiment and compare the FCCL with the existing methods in the performance of 1) CCL, 2) DA, 3) CA, and 4) ML. The on-site experiment where cameras are installed on buoys is difficult, so a part of Singapore Maritime Dataset are used for experiments [21]. The dataset is videos or images and some representative images of them are selected and experimented through about 30 lists. The criteria of image selection are: 1) weather clear/cloudy 2) horizontal height, 3) horizontal gradient, 4) ship's various colors, and 5) sunset. 30 candidates are selected, which are representative in terms of weather conditions. The embedded board Raspberry Pi 3B+ is used for the experiment. Although recently high-speed processing embedded boards such as NVIDIA's Xavier board are available, this experiment aims to achieve nearly 30 FPS(Frame Per Second) using low-cost embeds so that this study selects a single process with no acceleration capabilities. Raspberry Pi is a low-power processor with quad-core ARM(Advanced RISC Machines) processors of more than 1Ghz and supports multiple operating systems, making it more cost-effective for developing embedded software.

First, the performance is measured using the IoU(Intersection over Union), an indicator of the accuracy of the ship detection [22]. The IoU measurement divides the overlapping area between ['Ground-Truth' which is the actual answer area (A)] and [the detected area by the computer algorithm (B)], by the area of the sum (Equation 1). For example, based on BoX, 1

point of overlap area as shown in **Fig. 20** means the regions overlap completely. 0.5 means approximately 2/3 overlap. Here 0.5 becomes the threshold value and determines proper detection. The Box refers to the vertex based on the pixel of the detected object, which is shown as a rectangle. Segmentation is a method to check whether the location of pixels is accurately matched between the actual pixels and the position of the pixels in the detected area. This paper conducts comparative experiments by measuring performance through those two methods.

$$IoU = \frac{A \cap B}{A \cup B - A \cap B} \quad (1)$$



**Fig. 20.** Example of IoU Box and Segmentation

**Table 1** shows the results of applying existing algorithms (CCL, DA, CA and ML) and FCCL for 30 marine images selected. The IoU performance for each image is measured and evaluated by each of the four existing algorithms (CCL, DA, CA and ML) and FCCL. The overall mean value of the experimental results explains that the performance of DA twice, ML, and FCCL exceed the threshold of 0.5 or higher. It is shown that the FCCL has similar or better performance of Box IoU compared to other algorithms.

**Table 1.** Comparison of Box IoU performance of each detection algorithm

Image Number	IoU measurements for each image (Box)								
	[1] CCL	[2-1] DA (1 time)	[2-2] DA (2 times)	[2-3] DA (3 times)	[3-1] CA (1 time)	[3-2] CA (2 times)	[3-3] CA (3 times)	[4] ML	[5] FCCL
1	0.356	0.415	0.528	0.655	0.386	0.386	0.386	0.528	0.516
2	0.267	0.377	0.415	0.338	0.311	0.311	0.311	0.415	0.407
3	0.723	0.796	0.817	0.728	0.774	0.774	0.774	0.817	0.807
4	0.602	0.611	0.618	0.498	0.602	0.602	0.602	0.618	0.615
5	0.115	0.409	0.512	0.513	0.212	0.212	0.212	0.512	0.471
6	0.853	0.909	0.924	0.695	0.853	0.853	0.853	0.924	0.903
7	0	0	0	0	0	0	0	0	0
8	0.802	0.775	0.598	0.333	0.771	0.771	0.771	0.598	0.634
9	0.027	0.078	0.221	0.305	0.048	0.048	0.048	0.221	0.195
10	0.525	0.514	0.457	0.227	0.525	0.525	0.525	0.457	0.534
11	0.266	0.391	0.635	0.784	0.266	0.266	0.266	0.635	0.601

12	0.742	0.689	0.597	0	0.742	0.742	0.742	0.597	0.662
13	0.074	0.123	0.532	0.565	0.074	0.074	0.074	0.532	0.539
14	0.092	0.489	0.757	0.710	0.394	0.394	0.394	0.757	0.742
15	0.662	0.853	0.857	0.572	0.662	0.662	0.662	0.857	0.818
16	0.924	0.986	0.904	0.771	0.924	0.924	0.924	0.904	0.944
17	0.744	0.885	0.921	0.741	0.846	0.846	0.846	0.921	0.903
18	0.044	0.179	0.264	0.343	0.044	0.044	0.044	0.264	0.317
19	0.701	0.791	0.777	0.602	0.701	0.701	0.701	0.777	0.744
20	0.68	0.75	0.766	0.615	0.68	0.68	0.68	0.766	0.756
21	0.712	0.809	0.809	0.811	0.757	0.757	0.757	0.809	0.793
22	0.406	0.826	0.855	0.884	0.796	0.796	0.796	0.855	0.834
23	0.837	0.852	0.871	0.881	0.837	0.837	0.837	0.871	0.86
24	0.683	0.740	0.746	0.748	0.77	0.77	0.77	0.746	0.735
25	0.449	0.598	0.692	0.702	0.588	0.588	0.588	0.817	0.82
26	0.514	0.544	0.820	0.838	0.514	0.514	0.514	0.82	0.773
27	0.745	0.856	0.909	0.914	0.84	0.84	0.84	0.909	0.906
28	0.768	0.822	0.887	0.91	0.779	0.779	0.779	0.887	0.876
29	0.158	0.494	0.535	0.678	0.173	0.173	0.173	0.535	0.514
30	0.634	0.737	0.836	0.837	0.679	0.679	0.679	0.839	0.838
AVG	0.504	0.61	<b>0.669</b>	0.607	0.552	0.552	0.552	0.673	<b>0.669</b>

**Table 2** shows the results of applying the IoU Segmentation method to the same images. The results shows that the best performance is achieved by applying it twice. On the other hand, the FCCL does not show the good performance achieved in Segmentation. This is because the FCCL reduces performance-wise accuracy to increase processing speed.

**Table 2.** Comparison of Segmentation IoU performance of each detection algorithm

Image Number	IoU measurements for each image (Segmentation)								
	[1] CCL	[2-1] DA (1 time)	[2-2] DA (2 times)	[2-3] DA (3 times)	[3-1] CA (1 time)	[3-2] CA (2 times)	[3-3] CA (3 times)	[4] ML	[5] FCCL
1	0.39	0.586	0.687	0.703	0.46	0.46	0.46	0.674	0.678
2	0.297	0.476	0.563	0.349	0.37	0.37	0.37	0.622	0.569
3	0.478	0.523	0.509	0.495	0.486	0.486	0.486	0.271	0.338
4	0.359	0.342	0.328	0.32	0.351	0.351	0.351	0.254	0.281
5	0.106	0.417	0.606	0.68	0.195	0.195	0.195	0.293	0.531
6	0.715	0.817	0.797	0.772	0.759	0.759	0.759	0.564	0.617
7	0	0	0	0	0	0	0	0	0
8	0.749	0.744	0.618	0.531	0.825	0.825	0.825	0.525	0.571
9	0.031	0.094	0.179	0.206	0.055	0.055	0.055	0.179	0.149
10	0.678	0.653	0.437	0.315	0.682	0.682	0.682	0.395	0.48
11	0.327	0.489	0.586	0.598	0.345	0.345	0.345	0.511	0.522
12	0.742	0.49	0.352	0.276	0.742	0.742	0.742	0.289	0.321

13	0.066	0.132	0.499	0.559	0.074	0.074	0.074	0.503	0.439
14	0.053	0.314	0.63	0.719	0.182	0.182	0.182	0.641	0.587
15	0.535	0.672	0.659	0.615	0.565	0.565	0.565	0.437	0.435
16	0.78	0.858	0.815	0.713	0.785	0.785	0.785	0.443	0.469
17	0.77	0.84	0.838	0.826	0.78	0.78	0.78	0.716	0.697
18	0.056	0.258	0.401	0.43	0.056	0.056	0.056	0.401	0.432
19	0.579	0.667	0.66	0.633	0.582	0.582	0.582	0.579	0.565
20	0.596	0.656	0.629	0.6	0.602	0.602	0.602	0.535	0.528
21	0.612	0.784	0.772	0.764	0.713	0.713	0.713	0.664	0.781
22	0.523	0.72	0.711	0.665	0.661	0.661	0.661	0.581	0.654
23	0.795	0.889	0.85	0.8	0.898	0.898	0.898	0.825	0.851
24	0.736	0.807	0.763	0.704	0.811	0.811	0.811	0.642	0.793
25	0.574	0.729	0.761	0.756	0.67	0.67	0.67	0.677	0.739
26	0.584	0.678	0.779	0.772	0.618	0.618	0.618	0.714	0.722
27	0.708	0.790	0.764	0.713	0.771	0.771	0.771	0.625	0.742
28	0.691	0.811	0.843	0.836	0.746	0.746	0.746	0.711	0.842
29	0.189	0.612	0.777	0.851	0.214	0.214	0.214	0.78	0.848
30	0.825	0.662	0.515	0.449	0.889	0.889	0.889	0.725	0.77
AVG	0.474	0.584	<b>0.611</b>	0.588	0.53	0.53	0.53	0.526	0.565

**Table 3** shows FPS by averaging the time to process one frame for the input image. This study compares the existing algorithms (CCL,DA,CA and ML) with the FCCL, and the experimental results shows that the FCCL can provide several times faster effects than other algorithms.

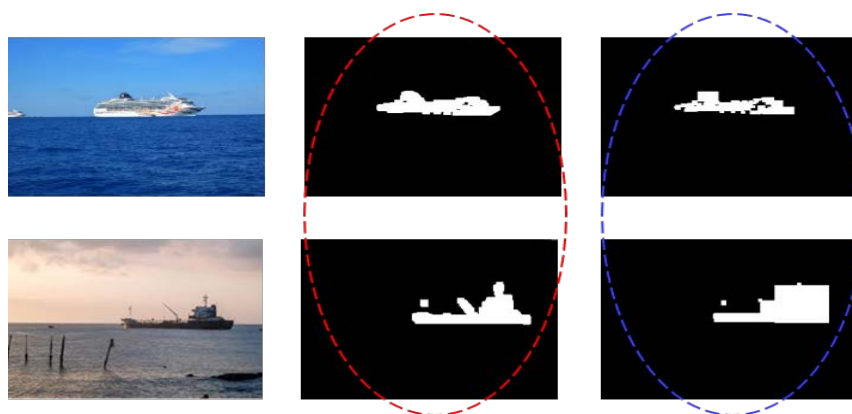
**Table 3.** The speed of image processing of each detection algorithm

Image Number	Number of 20 images processing time(ms)								
	[1] CCL	[2-1] DA (1 time)	[2-2] DA (2 times)	[2-3] DA (3 times)	[3-1] CA (1 time)	[3-2] CA (2 times)	[3-3] CA (3 times)	[4] ML	[5] FCCL
1	37.3	65.3	74.2	86.6	80.6	122.3	164.4	80.7	31.7
2	36.3	64.7	78.9	87.6	81.3	126.9	169.4	69.9	27.2
3	39.1	65.6	78.9	89.4	83.9	129.2	173	76.6	30
4	38.8	65.2	77	92.2	84.2	129.1	170.6	76	29.8
5	30.5	53.9	64.4	79.2	75.3	117.9	159.8	66.4	17.6
6	56.9	88.9	103.6	119.8	100	143.9	186.5	83.2	64.6
7	28.7	50.4	59.7	70.7	73.8	115.6	157.2	64.5	14.9
8	32.7	55.1	65.3	78	77.2	119.4	161.8	66.7	20
9	29.4	51.7	61.7	73.4	76.8	164.2	181	68.1	16.6
10	30.2	52.4	63.3	76	72.9	116.7	158	65.3	16.7
11	31.1	53.6	66.5	76.2	74	117.6	159.4	66.2	19.1
12	95.6	126.6	144.1	202.4	181.3	186.3	247.7	117.7	139.6
13	29.9	51.7	61	71.4	73.3	192.2	159.5	66	15.6



14	32.4	60.6	72.7	84.7	76.6	119.4	161.8	70.2	35
15	32.9	55.7	66.2	78.6	77.6	119	162.6	68.6	20.5
16	35.7	59.6	70.1	86.2	84.2	126.7	170.2	74.1	27
17	44.2	68.8	83	94.2	91.1	134.8	175.6	74.8	36.5
18	29	51.7	63.5	74.1	76.7	119.7	163.5	69	15.1
19	31.9	54.3	64.8	76.8	77.3	122.4	166.1	70.7	17.9
20	31.2	54.4	64.4	77.3	78.2	121.4	165	68.7	18.2
21	73.4	91.5	112.4	168.1	118.4	189.4	224.1	86.4	59.8
22	37.7	66.5	71.5	97.4	84.1	129.9	177.1	72.1	23.4
23	84	113.1	135.4	186.6	161.1	167.8	223.2	97.4	66
24	71.2	103.4	121.8	174.3	142.1	148.6	214.2	86.2	41.6
25	78.8	114.5	128.8	161.3	153.8	137.7	201.6	71	43.2
26	69.2	85.9	98.8	101	103.2	141.2	196.4	96.9	42.3
27	94.7	121.9	149.9	201.7	184.4	226.9	267.9	127.8	61
28	100.5	145.6	168.9	222.1	203.4	246.9	297.9	134.7	66.8
29	38.7	69.9	76.7	88.7	82.4	125.4	175.4	88.6	41
30	88.6	117.1	130.4	191.9	161.4	209.8	248.7	116.7	50.6
AVG	49.69ms	75.99ms	89.26ms	112.3ms	103ms	145.61ms	187.99ms	81.37ms	36.98ms
FPS	20.13 fps	13.16 fps	11.2 fps	8.91 fps	9.71 fps	6.87 fps	5.32 fps	12.29 fps	27.04 fps

**Fig. 21** shows the DA which is one part of the segmentation experiment and some of the detected shapes by the FCCL. While two inflations have a nearly similar shape to the original image, the image on which the FCCL is performed is constructed in a cascaded shape and thus is detected in an unusual shape. Instead, the performance of the Box side is almost similar, and is faster in terms of processing speed by the FCCL. This can be explained as a principle similar to quantization.



**Fig. 21.** Part of the Segmentation experiment (Red: DA, Blue: FCCL)

It shows a processing speed of approximately 2 times faster or more compared to the most similar DA, which is DA (twice or three times). Also, it is able to process approximately 4 to 5 times faster than the CA. As mentioned earlier, we could see that the DA and CA improve

performance by performing multiple times, but gradually decreasing the processing speed. In addition, the processing speed is not much different from that of Method 1, but the speed does not imply significance because it does not exceed the threshold in detection of performance. In addition, in the case of method 4, the processing speed is relatively slow due to the disadvantage of finding and expanding DA points in advance, like the FCCL, but having to perform Labeling twice. The experiments demonstrate that the FCCL processes quickly using LUT method close to about 30 FPS on the Raspberry Pi 3 board, which is faster than the existing algorithms (CCL, DA, CA and ML). **Table 4** summaries the mean values for **Table 1-3**. Overall, the FCCL shows best performance in terms of BoX Detection but lower performance in Segmentation.

**Table 4.** The overall results from table 1 to 3.

Evaluation Type	Categories of Algorithms								
	[1] CCL	[2-1] DA (1 time)	[2-2] DA (2 times)	[2-3] DA (3 times)	[3-1] CA (1 time)	[3-2] CA (2 times)	[3-3] CA (3 times)	[4] ML	[5] FCCL
IoU (Box)	0.504	0.61	<b>0.669</b>	0.607	0.552	0.552	0.552	0.673	<b>0.669</b>
IoU (Segmentation)	0.474	0.584	<b>0.611</b>	0.588	0.53	0.53	0.53	0.526	0.565
Processing Time (ms)	49.69	75.99	89.26	112.3	103	145.61	187.99	81.37	<b>36.98</b>
FPS	20.13	13.16	11.2	8.91	9.71	6.87	5.32	12.29	<b>27.04</b>

However, due to the nature of the maritime monitoring system, the FCCL is designed to improve the performance of Box Detection because it is more important to detect it roughly and quickly than to detect it in great detail. The processing speed is at least twice as fast as the other algorithms. Considering the speed is controlled only by software, increased the processing speed is significant. In addition, experiments demonstrated that it is practical and applicable by processing at a speed close to 30 FPS on a small embedded-board.

**Fig. 22** shows the results from the implementation of the FCCL. The original image and the FCCL are sequentially listed horizontally and the images selected from 1 to 30 vertically are listed in the Figure. There are some exceptions, because detection of horizontal lines on the sea, such as image 7, is not detected by the binary Otsu's algorithm. It is not easy to distinguish in binary when colors of sky and sea are almost the same. If no object is found after binarization, it is judged that it can be solved by applying a post-processing method such as finding the most changed high-frequency component in the entire image to find the ship. In addition, there are exception cases where binarization is not good due to reflection by the sun, such as image 12. Several experiments showed that the detection box area accounts for a significant portion in the entire image when incorrectly detected due to the weather conditions. Therefore, it may be possible to forcibly remove an abnormally large box area (e.g., about 1/2 or more) and exclude it from the detection target. Alternatively, although more computation is needed, but it is judged that it can be a safer way to determine and remove the object in the box using recognition technology or deep learning. Therefore, future work needs to focus on handling those exceptions on sites.



Fig. 22. The results of object detection from the marine safety monitoring system

## 5. Conclusion

In this paper, we optimize algorithms for detecting nearby ship objects through digital image analysis to prevent collisions of suspended solids at sea and enhance maritime safety. In addition, this study designs and proposes the FCCL which is able to process approximately 30 fps solely by software without hardware on the small embedded-board. Through experiments, performance analysis is conducted and compared to existing algorithms (CCL, DA, CA and ML) and the performance achieves similar performance to that of existing algorithms (CCL, DA, CA and ML), however the speed is improved by about 2-5 times speed. In this study, only the detection of objects in the marine environment is applied and described, however this can be implemented as an alarm system to measure approximate distance through image analysis for vessels approaching buoys. Also, if deep learning is applied to this algorithm, it will be possible to identify marine floats. If the identification by deep learning becomes the pre-process before the FCCL, better process speed and recognition rate is predicted. In this paper, experiments have been conducted using image datasets, however, the FCCL is planned to be designed as a real-time alarm system through CCTV (Closed Circuit Television) images installed on actual buoys on sites.

## References

- [1] S. Andres, F. Piniella, "Aids to Navigation systems on inland waterways as an element of competitiveness in ULCV traffic," *International Journal for Traffic and Transport Engineering*, vol.7, no.1, pp.1-18, 2017. [Article \(CrossRef Link\)](#).
- [2] S. J. Lee, M. I. Roh, H. Lee, M. J. Oh, "Image-based ship detection using deep learning," *Ocean Systems Engineering*, vol.10, pp.415-434, 2020. [Article \(CrossRef Link\)](#).
- [3] M. Pan, Y. Liu, J. Cao, Y. Li, C. Li, and C. H. Chen, "Visual recognition based on deep learning for navigation mark classification," *IEEE Access*, vol.8, pp.32767-32775, 2020. [Article \(CrossRef Link\)](#).
- [4] Z. Chen, D. Chen, Y. Zhang, X. Cheng, M. Zhang, and C. Wu, "Deep learning for autonomous ship-oriented small ship detection," *Saf. Sci.*, vol.130, no.104812, 2020. [Article \(CrossRef Link\)](#).
- [5] M. Abilio Ramos, I.B. Utne, A. Mosleh, "Collision avoidance on maritime autonomous surface ships: operators' tasks and human failure events," *Saf Sci*, vol.116, pp.33-44, 2019. [Article \(CrossRef Link\)](#).
- [6] S. Li and K. S. Fung, "Maritime autonomous surface ships (MASS): implementation and legal issues," *Maritime Business Review*, Vol. 4 No. 4, pp.330-339, 2019. [Article \(CrossRef Link\)](#).
- [7] S. Moosbauer, D. Konig, J. Jakel, and M. Teutsch, "A benchmark for deep learning based object detection in maritime environments," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019. [Article \(CrossRef Link\)](#).
- [8] M. Leclerc, R. Tharmarasa, M. C. Florea, A.-C. Boury-Brisset, T. Kirubarajan, and N. Duclos-Hindie, "Ship classification using deep learning techniques for maritime target tracking," in *Proc. 21st Int. Conf. Inf. Fusion (FUSION)*, pp.737-744, 2018. [Article \(CrossRef Link\)](#).
- [9] C. Y. Jeong, H. S. Yang and K. D. Moon, "Fast horizon detection in maritime images using region-of-interests," *Int. J. Distrib. Sens. N.*, vol.14, no.7, pp.1-11, 2018. [Article \(CrossRef Link\)](#).
- [10] C. Y. Jeong, H. S. Yang, and K. D. Moon, "Horizon detection in maritime images using scene parsing network," *IET Electron. Lett.*, vol.54 no.12, pp.760-762, 2018. [Article \(CrossRef Link\)](#).
- [11] T. Praczyk, "A quick algorithm for horizon line detection in marine images," *Journal of Marine Science and Technology*, vol.23, no.1, pp.164-177, 2018. [Article \(CrossRef Link\)](#).
- [12] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. Syst. Man and Cybern.*, vol.9, pp.62-66, 1979. [Article \(CrossRef Link\)](#).
- [13] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Trans. Pattern Anal. Machine Intell.*, vol.9, pp.532-550, 1987. [Article \(CrossRef Link\)](#).

- [14] L. Di Stefano, A. Bulgarelli, "A simple and efficient connected components labeling algorithm," in *Proc. of 10th International Conference on Image Analysis and Processing*, pp.322-327, 1999. [Article \(CrossRef Link\)](#).
- [15] S. Fefilatyev, D. Goldgof, M. Shreve, and C. Lembke, "Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system," *Ocean Engineering*, vol.54, pp.1-12, 2012. [Article \(CrossRef Link\)](#).
- [16] N. Arshad, K. Moon, and J. Kim, "Multiple ship detection and tracking using background registration and morphological operations," *Signal Processing and Multimedia*, vol.123, pp.121-126, 2010. [Article \(CrossRef Link\)](#).
- [17] S. Fefilatyev, D. Goldgof, "Detection and tracking of marine vehicles in video," in *Proc. of 19th International Conference on Pattern Recognition*, 2008. [Article \(CrossRef Link\)](#).
- [18] H. H. Song, H. C. Lee, S. J. Lee, H. S. Jeon and T. H. Im, "Design of Video Pre-processing Algorithm for High-speed Processing of Maritime Object Detection System and Deep Learning based Integrated System," *Journal of Internet Computing and Services (JICS)*, vol.21, no.4, pp. 117-126, 2020. [Article \(CrossRef Link\)](#).
- [19] T. Liu, B. Pang, L. Zhang, W. Yang, and X. Sun, "Sea Surface Object Detection Algorithm Based on YOLO v4 Fused with Reverse Depthwise Separable Convolution (RDSC) for USV," *Journal of Marine Science and Engineering*, vol.9, no.7, pp.753, 2021. [Article \(CrossRef Link\)](#).
- [20] S. J. Lee, H. C. Lee, H. H. Song, H. S. Jeon, and T. H. Im, "Comparative Analysis of CNN Deep Learning Model Performance Based on Quantification Application for High-Speed Marine Object Classification," *Journal of Internet Computing and Services (JICS)*, vol.22, no.2, pp.59-68, 2021. [Article \(CrossRef Link\)](#).
- [21] D. K. Prasad, C. K. Prasath, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Object detection in maritime environment: Performance evaluation of background subtraction methods," *IEEE Transactions on Intelligent Transportation Systems*, vol.20, no.5, pp.1787-1802, 2019. [Article \(CrossRef Link\)](#).
- [22] H. RezaTofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid and S. Savarese, "Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression," in *Proc. of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [Article \(CrossRef Link\)](#).



**Hyochan Lee** received the B.S. and M.S. degrees in Information & Communication Engineering from Hoseo University, Asan, Korea, in 2014 and 2016, respectively, and Ph.D. degrees in the school of Information & Communication Engineering from Hoseo University in 2019. He has been an researcher in Korea Electronics Technology Institute (KETI), Korea, Since 2021. His research interests are embedded system, parallel computing and computer vision.



**Hyunhak Song** received the B.S. and M.S. degrees in Information & Communication Engineering from Hoseo University, Asan, Korea, in 2019 and 2021. He has been an researcher in Korea Electronics Technology Institute (KETI), Korea, Since 2021. His research interests are computer vision, deep learning and artificial intelligence.



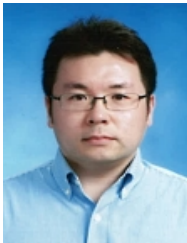
**Sungyoon Cho** received the B.S., M. S. and Ph.D. degrees in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2006, 2008, and 2013, respectively. From 2013 to 2020, he was with Samsung Electronics, Korea, as a Staff Engineer to research cellular communication systems and develop 4G and 5G modem chipset. Since 2020, he has been with Korea Electronics Technology Institute (KETI) as a Principal Researcher to develop the advanced technologies for embedded network. His research interests are in the fundamental aspects of wireless communication and signal processing, and learning algorithms for practical application.



**Kiwon Kwon** received B.S. and M.S. degrees in computer engineering from Kwangwoon University, Korea, in 1997 and 1999. He also received the Ph.D. degree in the School of Electrical & Electronics Engineering from Chung-Ang University, Korea, in 2011. In 1999, he joined in KETI, Korea, where he is currently a Group Leader with Oceans and Fisheries ICT Group. His research interests are in the area of advanced broadcasting/communication system, digital twin, oceans and fisheries ICT.



**Sunghyun Park** received the B.S. and M.S. degrees in electronic engineering from Kwangwoon University, Seoul, in 2005 and 2007, respectively. She also received the M.B.A. degree from Aston University, Birmingham, U.K., and Ph.D degree from the University of York, York, U.K., in 2014 and 2021 respectively. In 2016, she was awarded a research scholarship from the Department of Electronic Engineering, University of York, to undertake the Ph.D. degree. From 2008 to 2013, she became a Senior Researcher with Innoresearch, a manufacturer of mobile technology test solutions, South Korea. From 2014 to 2016, she went on to work at Qucell, London, U.K., an LTE small cell provider. She was a Technical Support Manager and collaborated with many major British mobile service providers. She is now an assistant professor in Hoseo University, South Korea. She is interested in the design and testing of novel intelligent Medium Access Control protocols.



**Taeho Im** received the B.S., M.S., and Ph.D. degrees in electrical and electronics engineering from Chung-Ang University, Seoul, South Korea, in 2006, 2008, and 2012, respectively. From 2012 to 2014, he worked for Samsung Electronics as a Senior Research Engineer, in charge of R&D on baseband systems for 5G communication systems. He is currently an Associate Professor with the Department of Information and Communication Engineering, Hoseo University, Asan-Si, South Korea. His research interests include underwater communication systems, 5G communication systems, and deep learning.