

스마트 항로표지 시스템에서 해상 객체 감지 가속화를 위한 방법에 관한 연구[☆]

A Study on Methods for Accelerating Sea Object Detection in Smart Aids to Navigation System

전 호 석¹ 송 현 학² 권 기 원² 김 영 진³ 임 태 호^{1*}
Ho-Seok Jeon Hyun-hak Song Ki-Won Kwon Young-Jin Kim Tae-Ho Im

요 약

최근 몇 년 사이 바다 신호등 역할을 하는 항로표지 시설이 디지털화되면서 단순한 표지판 기능을 넘어 해양 정보 수집, 감독, 관제 등 다양한 기능을 제공하는 방향으로 발전하고 있다. 국내에서도 울산항과 부산항이 부표 일부에 카메라를 설치하고 영상정보를 수집하여 충돌을 감독하는 등 선도 기술 적용을 주도하고 있다. 그러나, 이 첨단기술은 장기간 배터리 운용이 필요하고 관리·유지 등이 해양 특성에 지장을 받기 때문에 주요 기능을 수행하는 것이 어렵다. 이러한 문제들은 육상시설과 해양시설의 특성 차이를 극복하기 위한, 해양 분야의 풍부한 연구의 필요성을 제기한다. 본 연구에서는 표지 주변을 지나가는 해양 객체를 영상정보로 분석해 자동으로 알림을 줄 수 있는 시스템을 제안한다. 기존 시스템들은 표지와 해양 객체가 충돌하면 센서를 기반으로 객체를 포착해 서버로 전송하지만, 이 방식은 사고가 발생한 뒤에야 안전사고 대처가 가능해 개선의 여지가 필요하다. 따라서, 제안하는 시스템은 이러한 한계점을 극복하기 위해, 해양 특성을 기반으로 해상 객체 감지 시스템을 설계하였다. 이는 기존의 해양 영상처리 감지 시스템과 유사한 성능을 보였으며, 보다 효율적인 모니터링을 위해 약 5배 빠른 처리 속도를 기록한 해상 안전 시스템을 제안한다.

☞ 주제어 : 해상 객체 검출, 모션 검출, 영상 처리, 항로표지, 안전사고 보호

ABSTRACT

In recent years, navigation aids, which plays as sea traffic lights, have been digitized, and are developing beyond simple sign purpose to provide various functions such as marine information collection, supervision, control, etc. For example, Busan Port which is located in South Korea is leading the application of the advanced technologies by installing cameras on buoys and recording video images to supervise maritime accidents. However, there are difficulties to perform their major functions since the advanced technologies require long-term battery operation and also management and maintenance of them are hampered by marine characteristics. This study proposes a system that can automatically notify maritime objects passing around buoys by analyzing image information. In the existing sensor-based accident prevention systems, the alarms are generated by a collision detection sensor. The system can identify the cause of the accident whilst even though it is difficult not possible to fundamentally prevent the accidents. Therefore, in order to overcome these limitations, the proposed a maritime object detection system is based on marine characteristics. The experiments demonstrate that the proposed system shows about 5 times faster processing speed than other existing algorithms.

☞ keyword : Maritime Object detection, Motion detection, Image processing, Aids to Navigation, Protection of Safety Accidents

1. 서 론

최근 안전의식 강화 및 국가 기반 방재 요구가 사회 전 반적으로 증가함에 따라 지능형 CCTV(Closed-circuit Television) 기술이 사회 안전 분야의 핵심 기술로 부상하면서 스마트 홈, 시티 등 다양한 영역에 활용되어 재난 대

학ICT연구센터지원사업의 연구결과로 수행되었음
(IITP-2022-2018-0-01417)

☆ 이 논문은 2022년 해양수산부 재원으로 해양수산과학기술진흥원의 지원을 받아 수행된 연구임(AI 기반 스마트 어업관리 시스템 개발, 202104992)

¹ Oceanic IT convergence Technology Research Center, Hoseo University, Asan-si, 31499, Korea.

² Smart Network Research Center, Korea Electronics Technology Institute, Seoul, 03924, Korea.

³ Aids to Navigation Division, Ministry of Oceans and Fisheries, Sejong, 30110, Korea.

* Corresponding author (taehoim@hoseo.edu)

Received 08 August 2022, Reviewed 17 August 2022, Accepted 21 August 2022

☆ 이 논문은 2022년 해양수산부 재원으로 해양수산과학기술진흥원의 지원을 받아 수행된 연구임(스마트항로표지 현장시설 고도화, 20210636)

☆ 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대

응 역량 강화에 기여하고 있다. 기존의 CCTV 방식이 아날로그에서 디지털 방식으로 전환되면서 네트워크를 통한 원격 모니터링이 가능해졌고, 다양한 분야에서 실시간 측면의 지능화된 CCTV 영상분석 적용을 시도하고 있다. 이러한 지능형 CCTV 영상분석 기술은 공장, 항만, 철도 등 대규모 시설 내에서도 활용되고 있다[1][2].

해양 분야도 ICT(Information&Communication Technology)가 해운, 항만, 수산, 조선 등의 정보통신 기술들과 접목되면서 스마트 해운, 항만물류시스템, 스마트 양식, 자율운항 선박 등 빅데이터 IoT(Internet of Things) 기술의 발전을 통해 고도화되고 있다. 또한, 해상 교통의 길잡이인 항로표지도 선박의 안전한 운항을 도와주는 표지판 기능을 넘어서, 해양 정보수집 및 안전 강화 모니터링을 위한 디지털 기술이 접목되고 있다. 해양의 종합적인 기상 현상을 다양한 기상장비로 관측하거나, 안전 항해를 유도하기 위한 선제적 해양 정보를 제공함으로써 해양 관련 종사자의 생업에 중요하게 활용되고 있다. 하지만, 선박과 항로표지의 충돌 사고 및 위치이탈 사고가 빈번히 발생하고 있으며, 최근 6년간 총 38건의 선박과 항로표지의 충돌 사고가 발생했다. 파손된 항로표지는 장시간 수리 기간 소요 및 피해복구에 큰 비용이 발생하며, 해당 기간에 해양 기상 정보 수집 활용이 어려워 불편함을 초래한다. 이러한 문제를 해결하기 위해, 국내에서는 등부표 충돌 방지 영상정보 시스템을 개발하고 실용화하여, 항로표지 시설 상태를 감시, 수집하고 있다[3].

그러나, 육상에서의 지속적인 전자 장치 운용에 비해 해양 환경의 특성상 전력 공급이 제한적이다. 항로표지에 적용되는 영상 시스템은 배터리와 같은 한정된 에너지에서 장기간 유지하기 위한 저전력 시스템으로 구성되어야 한다. 따라서, 본 논문은 최소한의 영상분석을 통해, 해양 주변의 객체 존재 여부를 판단하는 영상처리 알고리즘을 제안한다.

일반적으로 해상 객체는 수평선 위에서 객체 판별이 이루어지며 알고리즘 진행의 기준이 된다[4]. 보편적인 수평선 검출은 캐니 에지 검출(Canny edge detection)을 이용하여 이미지의 윤곽을 검출하고, 잡음 제거를 위한 필터 등을 적용하여 허프 변환(Hough transform)을 통해 수평선의 직선 성분을 추출하는 방식으로 검출한다. 이 방식은 검출률이 낮고 수평선의 끝 좌표만 획득하기 때문에, 상세한 구간의 수평선 좌표를 알 수 없어서 활용하기 어려우며, 객체 탐지에 대한 알고리즘이 별도로 필요하다.

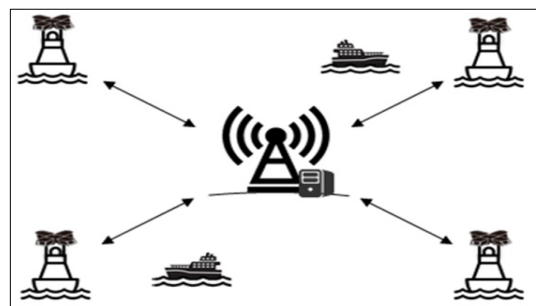
제안하는 알고리즘은 기존 수평선 검출의 복잡한 전처리 연산을 영역별 색상 비교 분석 방식으로 간소화시키

며, 이를 해상 객체 감지까지 연계한다. 간단하게는 입력 이미지를 그리드 영역으로 분할하여 영역별 좌표와 평균 픽셀을 획득하고, 하늘과 바다의 색이 구분되는 해양 특성을 이용하여, 그리드 영역 간 색상 변화량 분석을 통해 수평선 검출과 객체 판별 정보를 획득한다. 실험은 구성된 실험 환경을 기준으로 기존 수평선 검출 알고리즘보다 검출률이 높고, 연산속도가 약 5배 빠른 결과를 보여준다. 속도 개선은 동일 시간 대비 소모 전력량을 낮출 수 있으며, 이는 사용하는 프로세서 성능에 따라 성능이 달라져 저가형 시스템일수록 효율이 극대화된다. 본 알고리즘은 수평선 검출의 단순한 성능 개선뿐만 아니라 객체 탐지까지 최소한의 연산으로 연계할 수 있으며, 저전력 환경이 필요한 해양 IT 임베디드 시스템에 적합하다.

2. 기존 관련 연구

2.1 센서 기반 해양 객체 감지

기존 센서 기반 항로표지 충돌 방지 시스템은 충격 감지 센서에 의해 촬영 이벤트가 발생하는 방식이다. 사람이 직접 주기적으로 영상 상태를 확인하며, 사고 규명 원인에 대해서는 파악할 수 있지만, 근본적인 충돌 방지를 예방하기는 어렵다. 효율적으로 사고에 대처하기 위해서는 표지 장비 자체가 능동적으로 사고 알람을 관리자에게 줄 수 있어야 한다. 이에 대해, 본 연구는 그림 1과 같이 해상에서 주변의 상황을 감시하기 위하여, 다수의 카메라를 사용한 360도 전경을 촬영한다. 따라서, 이를 처리할 디지털 프로세서가 필요하며, 항로표지는 배터리를 기반으로 운용되므로, 고성능 영상분석의 기술을 도입하기에는 어려움이 있다. 즉, 영상처리 소프트웨어의 저전력화 기술이 필요한 상황이다.



(그림 1) 해상 안전 모니터링 시스템
(Figure 1) Maritime Safety Monitoring System

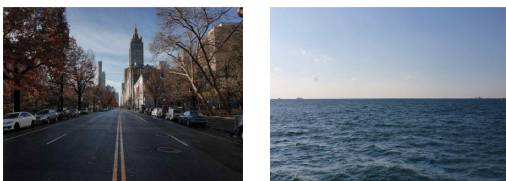
2.2 비전 기반 해양 객체 감지

최근 인공지능 기술과 비전 기술이 발달함에 따라 카메라, 레이더, 라이더(LiDAR) 등 이미지 센서를 통해 획득한 정보를 컴퓨터로 분석하고 판단하여 객체의 인식을 돕는다. 레이더는 장거리 측정이 가능하고 기후에 영향을 적게 받지만, 객체 식별이 매우 어렵고, 라이더는 카메라와 레이더에 비해 인식 정밀도가 높고 매우 민감하여 성능이 좋지만, 비용이 많이 든다. 카메라는 날씨 영향은 많이 받으나 영상처리 소프트웨어 개선에 따라 극복 여지가 있으며, 가격은 라이더보다는 훨씬 저렴한 편이기에 비용적인 측면에서는 가장 효율적이다[5].

카메라는 기존 충격 감지 센서보다는 전력 효율이 떨어지지만 사고 발생 이전에 미리 탐지하여 방지할 수 있다[6]. 하지만 부표는 배터리 방식이기 때문에 최소한의 비전 처리 방법으로 평상시에는 저전력 모드로만 동작하게 하는 것이 중요하다. 이러한 방법으로는 특정 픽셀 그룹을 추적하여, 해당 관심 영역의 변화 상태를 감지하는 것, 존재 여부를 판단하는 것이 가장 효과적으로 전력 소모를 줄일 방법이다. 지상처럼 다수 불특정 대상으로 객체의 존재 여부를 판단하기 위해서는 많은 픽셀 처리가 있어 전력 소모가 높아지지만, 해상은 수평선을 기준으로 객체의 존재 여부를 판별한다. 해당 영역만 관측하여 존재 여부 상태를 판단하는 기준점이 존재하기 때문에 검출 방법이 더욱 유리하다.

2.3 이미지 기반 해양 객체 감지 방법

입력 이미지로부터 어떠한 목적을 달성하기 위하여, 영상 전처리를 이용한 객체를 검출하고 특정 동작을 수행하기 위해서는 객체의 관심 영역을 찾는 것이 중요하다. 즉, 이미지에서 일정하게 유지되는 환경 패턴을 찾아 규칙화하여 객체의 관심 영역을 찾는 영상처리 알고리즘 설계가 필요하다.

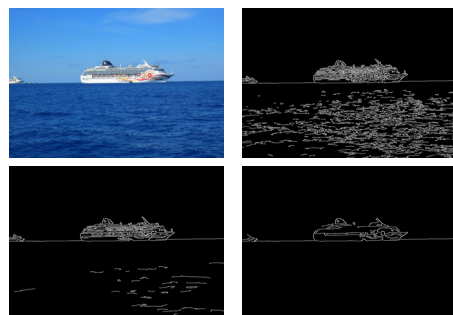


(그림 2) 육상과 비교되는 해상 이미지의 특성
(Figure 2) Characteristics of Maritime Image Compared with Land

이는 그림 2의 도식처럼 건축물과 지형지물이 복잡하게 나열된 환경에서의 규칙적인 이미지 패턴으로 차선을 예시를 들 수 있고, 이처럼 해양의 이미지 패턴은 하늘과 바다가 하늘색, 파란색으로 색상 경계가 구분되는 수평선을 예시를 들 수 있다. 해양은 도식처럼 건축물이나 지형지물 등의 방해 요소가 상대적으로 적고, 이를 이용한, 해상 객체들을 검출하기 위한 전처리 연구들이 시도되고 있으며, 이에 관련된 기법들을 2.4절에서 소개한다.

2.4 해양 객체 감지 알고리즘

해상 객체의 존재 여부 판별은 주로 하늘과 바다 영역의 경계인 수평선을 기준으로 활용한다. 해상 객체의 경우, 수평선 위에서부터 천천히 접근하는 객체를 파악하는 게, 수평선 아래에서 가깝고 빠르게 접근하는 객체를 탐색하는 것보다, 데이터 처리 시간 및 잡음에 유리하다. 일반적인 기존의 수평선 검출은 에지 검출을 통해 화소가 많이 변화되는 경계선을 찾아 수평선을 탐색한다. 해양 환경의 경우 가장 큰 픽셀 군집으로는 하늘과 바다 영역이기 때문에, 수평선 구간에서 가장 큰 변화가 나타나는 특징이 있다. 주로 캐니 에지 검출과 같은 2차 미분 방법을 적용한 후, 평균값(Mean), 가우시안(Gaussian), 미디언(Median) 등 1차 미분 필터와 컨볼루션(Convolution) 연산을 통해, 그림 3과 같이 잡음을 제거하여 에지를 검출하게 된다[7]. 기존의 캐니 에지 기반의 수평선 검출 과정은 잡음에 크게 민감하여, 이상적인 수평선 검출을 위해 에지 검출 강도와 필터 크기의 임계치 조절이 수동적으로 필요하다. 이는 환경이 유동적으로 빠르게 변하는 해양 환경에서 수평선 검출을 위한 에지 검출의 범용적인 임계치를 설정하기 쉽지 않다. 이에 대한 임계치를 이진 이미지에서 픽셀 히스토그램을 이용하여 적절한 중간값을 찾아주는 알고리즘이 활용되기도 한다[8].



(그림 3) 필터링된 에지 검출 알고리즘
(Figure 3) Filtered Edge Detection Algorithm

만일 이상적인 예지가 검출되었다면, 수평선만을 검출하기 위해, 주로 허프 변환 알고리즘을 직선 검출에 사용한다[9]. 그러나, 허프 변환은 수평선으로 추정되는 예지가 존재하여도, 직선 성분을 미검출하거나 오검출의 경우가 빈번히 발생하며 연산량도 많다. 또한, 수평선 검출 이후에 개별적으로 객체를 찾아내는 알고리즘이 필요하며, 연산량이 많으면 임베디드 환경에서 프로세서 처리에 부담이 존재한다.

검출된 수평선의 상단을 관심 영역으로 지정하였을 경우, 객체를 탐지하는 방법 중 지연선으로 활용되는 알고리즘은 이진 영상 차 영상을 이용한 방법을 통해 객체를 감지할 수 있다. 시간이 흘러가는 이진 입력 영상에서 현재와 과거 두 프레임 간의 차이를 식 (1)과 같이 계산한다.

$$B_t(x,y) = \begin{cases} 1 & \text{if } |f_t(x,y) - f_{t-1}(x,y)| \geq T \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

식 (1)은 이진 차 연산 방법을 이용한 모션 객체 검출에 관한 식이다. $B_t(x,y)$ 은 변화가 없는 부분을 제외하고 움직인 요소를 추출하는 한 프레임의 이진 이미지를 의미하고, (x,y) 는 프레임의 픽셀 좌표를 의미한다. 이미 지 픽셀값은 현재 프레임 $f_t(x,y)$ 에서 바로 전 과거 프레임 $f_{t-1}(x,y)$ 의 픽셀값 차이를 계산한다. 픽셀의 차이가 임계치보다 높으면 움직인 객체인 이진 픽셀로 간주하며, T 는 미리 정의된 임계값을 각각 의미한다. 이 방법은 간단한 연산으로 움직이는 객체를 추적하는 것이 쉽다 [10][11]. 그러나, 이와 같은 방식으로 해양 객체에 대해 정확히 추적하기 위한 성능에는 한계가 있다. 해양 환경은 지상과 같이 고정적인 프레임이 아니라 흔들림이 존재하여 카메라의 초점이 변하는 유동성 프레임으로 객체만이 아닌 화면 전체가 흔들린다. 결국, 해양 객체 존재 여부를 감지하기 위해서는 이러한 유동적인 문제를 해결할 수평선 검출 방법과 해양 환경에 맞는 객체 검출 방법이 고안되어야 한다. 이에 대하여, 다른 접근 방식으로 객체를 탐지하는 시스템을 3장에서 제안한다.

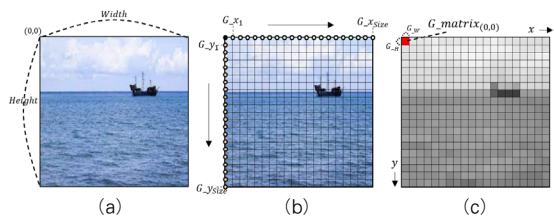
3. 제안하는 시스템

제안하는 알고리즘도 마찬가지로 하늘 또는 바다의 영역에서 비슷한 색상 간의 차이는 크지 않다는 점을 이용하였다. 밀집된 색상 영역을 벗어났을 때, 그 경계에서 픽셀 변화량이 크게 변동하였다. 본 연구는 이미지를 임의

의 영역으로 분할하여, 영역 간의 픽셀 차이를 비교하는 방식으로 수평선 검출과 객체 판별하였다. 분할의 목적은 색상 영역의 비교뿐만 아니라 영역별 좌표 정보도 함께 얻기 위함이며, 본 논문은 $N*N$ 이미지를 격자 분할하는 것에 대하여 그리드 분할이라 정의한다. 그리드 영역의 색상 비교를 위해, 각 영역의 내부 픽셀이 회색조(GrayScale)된 평균값으로 통합되어 단일 고유 픽셀값을 갖는다. 이후 알고리즘은 영역의 각 고유 픽셀값 비교를 통하여, 최대 변화량 구간에서 수평선과 객체 존재를 검출한다.

3.1 그리드 분할

해양 이미지는 그림 4의 (a)처럼 하늘과 바다 그리고 객체는 유사한 색상끼리 구성되어 있다. 이를 정확한 영역과 좌표로 구분하기 위해, 이미지를 (b)와 같이 그리드 분할하였다. 픽셀 하나에 대한 미세한 움직임보다는 멀리 있는 선박의 크기 정도로 이미지를 분할하여, 움직임 변화를 관찰하는 것이 검출률과 잡음 처리 등에 유리하다. 분할된 영역 간의 픽셀 비교를 위하여, (c)와 같이 분할된 그리드 영역별로 모든 내부 픽셀값에 대한 회색조를 적용한 후 평균값을 취한다. 이로 인한, 영역은 3채널 컬러에서 1채널 회색조로 바뀌고, 영역 내부의 픽셀값이 평균값으로 통일되면서, 전체 이미지에 대한 연산이 그리드 영역 크기(Grid_Size)로 간소화된다. 영역 간 픽셀 변화를 분석하는 알고리즘 과정에서 연산량을 가장 크게 차지하는 부분은 이미지로부터 픽셀의 값을 읽어오는 부분이다. 픽셀 비교와 연산속도를 효율적으로 하기 위해, 픽셀을 직접 접근 방식으로 전체 이미지를 한 번만 읽고, 픽셀의 데이터만 배열로 활용하여 연산을 최소화하였다.



(그림 4) 그리드 알고리즘 과정
(Figure 4) Grid Algorithm Process

그림 5는 그림 4의 과정처럼 입력 이미지를 그리드화 시키는 알고리즘의 의사코드를 제시한다. 단순하게는 $N*N$ 이미지와 그리드 적용할 분할 크기(Grid_size)를 입력 받고, 분할된 그리드 영역의 폭(Gw)과 높이(Gh)를 계산하

```

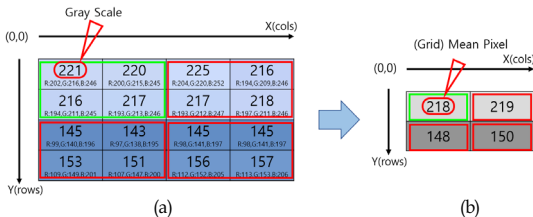
Void function GRID_MATRIX

get image_matrix[rows][cols]
get grid_size
get grid_matrix[grid_size][grid_size]
grid_width = image_matrix.cols / grid_size
grid_height = image_matrix.rows / grid_size
grid_pixel = 0
For (y = 0; y < grid_size; y++) {
  For (x = 0; x < grid_size; x++) {
    For (j = (y * grid_height); j < ((y + 1) * grid_height; j++) {
      For (i = (x * grid_width); i < ((x + 1) * grid_width; i++) {
        grid_pixel = grid_pixel + image_matrix[j][i](R + G + B color)/3
      }
    }
    grid_matrix[y][x] = grid_pixel / (grid_width * grid_height)
    grid_pixel = 0
  }
}
print grid_matrix
    
```

(그림 5) 그리드 알고리즘의 의사코드

(Figure 5) Pseudo Code of Grid Algorithm

여, 분할된 각 그리드 영역의 내부 픽셀을 회색조 평균으로 연산한다. 영역의 내부 픽셀은 모두 같은 값을 가지기 때문에, 그리드 영역 분할 크기만큼의 2차원 배열에 저장하여 간소화된 픽셀 데이터를 활용한다. 그림 6은 회색조 평균 픽셀 산출 예시이다. (a) (4x4) 컬러 이미지를 (b) (2x2) 그리드 분할한 연산 결과이다.



(그림 6) 그리드 영역에 대한 평균 픽셀 계산 예시

(Figure 6) Example of calculating average pixels for a grid area

$$GrayScale = (R + G + B) / 3 \quad (2)$$

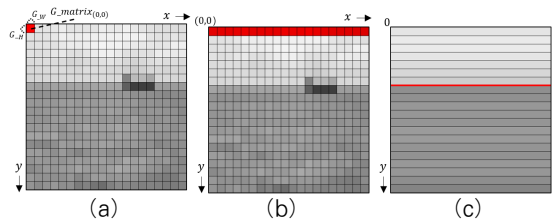
$$(Green\ Area)MeanPixel = (221 + 220 + 216 + 217) / 4 \quad (3)$$

식 (2)는 그림 6의 (a) 이미지에서 3채널 RGB 픽셀값이 1채널 회색조 값으로 변환된 식이며, 식 (3)은 간소화되는 과정에서 초록 영역의 연산 과정을 의미한다. (a) 초록 영역 속 픽셀값을 모두 더한 후, 영역의 픽셀 개수만큼 나누면 (b) 한 칸의 초록 그리드 영역에 대한 평균 픽셀값이 된다. 각 그리드 영역의 평균 픽셀값은 색상 변화량을 비교할 수 있는 색상 정보가 된다. 이러한 간소화는 이미지 전체에 대한 비교 대신, 그리드 영역 간의 비교 연산이 되고, 픽셀 접근 횟수가 줄어들어 연산량이 감소할 뿐만 아

니라, 잔 파도의 음영처럼 미세한 잡음에도 검출이 덜 민감하도록 검출률을 높일 수 있는 전처리 방법이 된다.

3.2 수평선 라인 검출

해양 객체를 감지하기 위해서는 수평선 검출은 필수적이다. 여기서는 보편적인 수평선 검출 Canny Edge, Hough Transform을 사용하는 것이 아닌 그림 7과 같이 3.1절에서 그리드 분할로 저장된 픽셀값을 이용하여 수평선의 대략적인 위치를 찾는다. 그리드 영역 사이에서 수평선을 검출하기 때문에, 실제 수평선의 모양 측면에서는 보편적인 검출 방법이 조금 더 정교하지만, 매우 정교한 수평선을 찾는 것이 크게 중요한 요소는 아니다. 또한, 그리드 분할의 크기를 조절하여 정확도를 높일 수 있으며, 본 목적은 해양 객체의 존재 판별이 가능한 관심 영역 지정이므로, 수평선 검출률과 처리 속도 측면에서 제안하는 수평선 검출이 더 우수한 성능을 보인다. 이에 대한, 그리드 수평선 검출은 그림 7과 같은 그리드 분할된 이미지 (a)에서 X축의 그리드 영역을 (b)의 빨간 표기된 영역처럼 평균 병합한다. 이후 (c)와 같이 X축 영역이 병합된 이미지에서, 각 영역의 평균값 차이를 비교함으로써 차이 값이 최대가 되는 구간을 수평선으로 검출하는 알고리즘이다.



(그림 7) 그리드 기반 수평선 검출

(Figure 7) Grid-based Horizon Detection

```

Void function GRID_HORIZON

get grid_size
get grid_matrix[grid_size][grid_size]
get horizon_matrix[grid_size]
width_total = 0
For (y = 0; y < grid_size; y++) {
  For (x = 0; x < grid_size; x++) {
    width_total = width_total + grid_matrix[y][x]
  }
  width_total = width_total / grid_size
  grid_horizon[y] = width_total
  width_total = 0
}
print horizon_matrix
    
```

(그림 8) 그리드 수평선 알고리즘 의사코드

(Figure 8) Pseudo Code of Grid Horizon Algorithm


```

Void function HORIZON_DETECTION

get grid_size
get horizon_matrix[grid_size]
pixel_diff = 0
max_diff = 0
horizon_index = 0
For (y = 0; y < grid_size; y++) {
    pixel_diff = absolute value(grid_horizon[y + 1] - grid_horizon[y])
    if (max_diff ≤ pixel_diff) {
        max_diff = pixel_diff
        horizon_index = y
    }
print horizon_index
    
```

(그림 9) 그리드 수평선 검출 의사코드
(Figure 9) Pseudo Code of Grid Horizon Detection

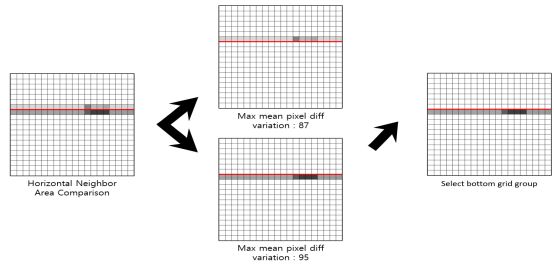
수평선 검출은 하늘과 바다의 경계를 픽셀 변화량이 가장 높은 구간으로 가정하여 수평선을 검출한다[12]. 그림 8은 그리드 기반으로 수평선을 검출할 수 있는 그리드 구조로 만드는 의사코드이다. 그림 7의 (a)처럼 이미지가 분할된 그리드 영역에서 X축 영역의 픽셀값을 Width_total에 모두 더한 뒤 평균을 내면 (c)와 같은 그리드 수평선 구조가 된다. 그림 9는 수평선을 검출하는 의사코드이고, 평균화된 X축 영역의 픽셀들을 Y축 상단에서부터 비교하며, 다음과 같이 픽셀 차이를 계산한다.

$$Y \text{ mean pixel diff}_i = | y_{i+1} - y_i | \quad (4)$$

식 (4)에서 픽셀 차이가 가장 큰 구간의 인덱스를 수평선으로 가정하기 때문에, 수평선 검출의 의사코드는 픽셀값을 비교하며 최대 픽셀 차이와 해당 구간의 인덱스 번호를 탐지한다. y_i 는 X축이 병합된 그리드 영역의 픽셀값이며, i 는 상단에서 인덱스 번호가 0부터 시작한다. 다음 인덱스 y_{i+1} 간의 평균 픽셀 차이를 구하고, 픽셀값 차이가 가장 높은 구간에서 인덱스를 반환하여 수평선을 검출한다. 그림 7의 (c)에서 붉은색 라인으로 표기된 해당 구간의 위치에서 수평선이 검출되었다.

3.3 객체 존재 여부 검출

객체의 경우 수평선의 상단에서부터 접근하며, 매우 근접하게 가까워질수록 수평선 하단으로 접근한다. 본 알고리즘은 매우 근접한 곳이 아닌 위험 알림 또는 전처리 연계 동작이 가능한 거리에서 탐색하고자 한다. 따라서, 객체를 탐지하는 관심 영역은 그림 10과 같이 객체가 수평선에 걸쳐진 경우를 고려하여, 수평선에 이웃한 그리드 영역이다.

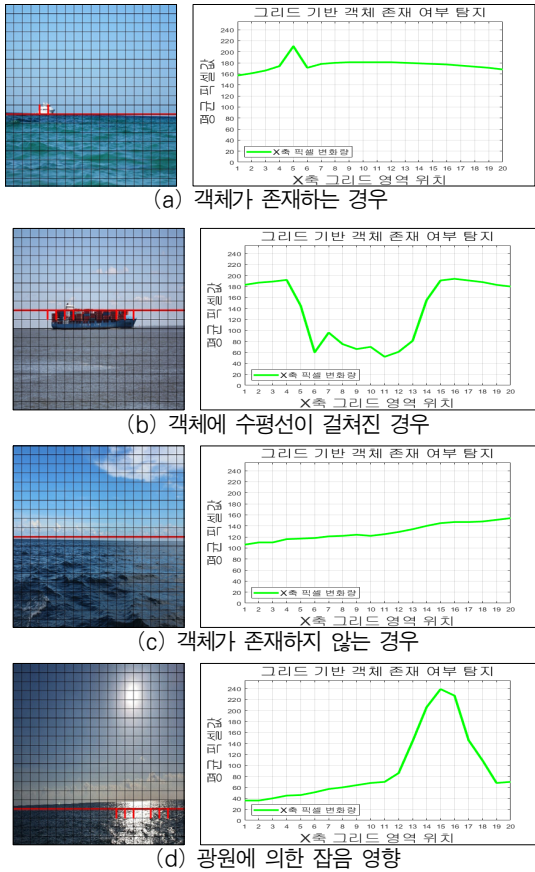


(그림 10) 객체 탐지 행의 선택 방법
(Figure 10) Selection Method for Object Detection Rows

수평선 검출의 경우에는 Y축을 기준으로 픽셀 차이를 비교한 것처럼, 객체 탐지는 그림 10과 같이 수평선에 이웃한 두 행의 각 X축 그리드 영역을 좌측부터 픽셀 변화량을 탐지하는 것이다. 객체 판별 방법은 수평선 검출과 똑같이 진행되며, 탐지하는 축에 대한 방향만 바뀐 것이고, 다음과 같이 픽셀 차이를 계산한다.

$$X \text{ mean pixel diff}_i = | x_{i+1} - x_i | \quad (5)$$

식 (5)는 x_i 는 그리드 영역의 픽셀값을 의미하며, i 는 인덱스 번호이다. 식 (4)와 마찬가지로 다음 인덱스 x_{i+1} 의 픽셀 변화량을 계산한다. 객체 존재 판별에서 객체가 존재하지 않는 경우는 하늘과 바다 영역에서 주로 균일한 색상을 띄어 픽셀 변화량이 적다. 픽셀 변화량 값이 클수록 객체 또는 잡음 등 다른 요소의 개입으로 예측할 수 있다. 객체 탐지는 수평선에 이웃한 상단과 하단행 중에 식 (5)의 최대 픽셀 변화량이 높은 쪽에서 객체 판별을 진행한다. 객체 판별의 기준은 픽셀 변화량이 급변하는 구간으로 가정하였고, 픽셀값이 급변하는 크기는 상대적으로 유동적이기 때문에, 이에 대한 구간을 찾는 알고리즘 개선이 필요하다. 임계치는 강한 잡음 요소를 제외한 실험 데이터 세트의 결과를 참고하여 20픽셀 차이로 설정하였다. 객체 존재 여부는 임계치를 기준으로 복수 판별되며, 직접적인 햇빛과 같이 일반적으로 알려진 강한 잡음이 아니라면 긍정적인 객체 판별 결과를 나타냈다. 또한, 객체 판별의 연산은 이미 계산된 그리드 영역의 픽셀값을 활용하였기 때문에, 최소한의 연산 과정으로 진행되었다.



(그림 11) 해양 객체의 존재 여부 판별
(Figure 11) Determining the Presence of Marine Objects

그림 11은 해양 객체의 존재 여부에 따라 판별된 유형의 경우를 나타낸다. 이에 대한 X축 그리드의 영역 픽셀 변화량 차이는 Matlab 이용하여 그래프로 나타내었다. 그래프의 X축은 그리드 영역의 인덱스 번호이고, Y축은 해당 인덱스의 픽셀값이다. 이미지의 인덱스 구간에 표기된 작은 세로 막대 바는 임계치 20픽셀을 초과한 구간을 의미한다.

(a) 수평선에 객체 존재하는 경우이다. 객체가 존재하지 않는 구간은 균일한 작은 변화량을 보이며, 객체가 존재할 경우, 급변하는 구간을 나타낸다. 급변하는 값이 임계치를 넘으므로 객체 검출로 판별되었다.

(b) 수평선이 객체의 일부에 형성된 경우이다. 주로 이미지상의 객체 부피 또는 색상 대비가 큰 객체에 발생한다. 다른 보편적인 알고리즘에서도 발견되며, 수평선 검출

자체는 오검출로 판단한다. 하지만, 객체가 존재하지 않는 수평선 검출과 비교하여, 객체 존재 판별과 연계되는 그리드 기반 수평선 검출에서는 하늘과 객체 사이의 픽셀 변화도 정상 판별이 가능하다는 가정으로 별도 분류한다. 오검출과는 다르게 객체 수평선이라 정의하며, 수평선 검출로 취급한다.

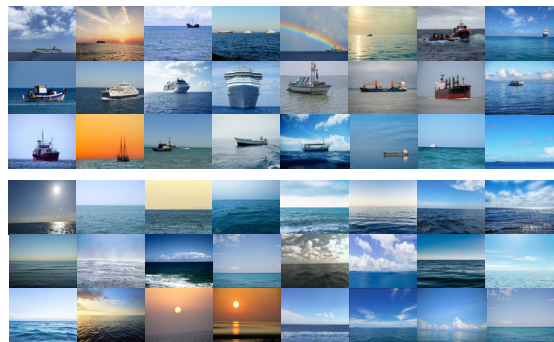
(c) 객체가 존재하지 않는 경우이다. 픽셀 변화량이 임계치 이하로 적은 변화량을 보이며, 객체가 존재하지 않는 경우로 판별된다.

(d) 잡음의 영향으로 객체가 오검출된 경우이다. 해당 경우는 햇빛이 직접 바다에 비치는 윤슬이 객체로 오검출되었다. 햇빛에 의한 잡음은 광량이 가장 강한 부분에서 피크 점을 보이며, 근원지와 멀어질수록 점점 감소하는 규칙성을 확인할 수 있었다. 추후, 이에 대한 픽셀 패턴분석을 통한 개선의 여지가 있다.

4. 실험 및 성능평가

4.1 실험 환경 구성

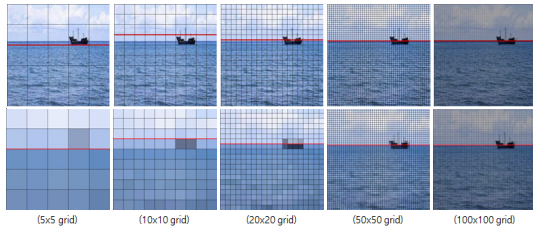
제안하는 알고리즘의 시스템 구성을 설명하고, 비교 및 실험 결과를 제시한다. 실험환경은 Raspberry-Pi4 임베디드 보드의 Raspbian Buster OS 환경에서 OPENCV 4.1 라이브러리를 기반으로 알고리즘을 구성하였다. 본 연구에 적합한 해양 객체 데이터를 직접 수집하기 어려움이 존재하여, 웹에서 이미지를 수집하였다. 제안하는 시스템의 성능 분석을 위한 테스트 데이터는 그림 12를 예시로 수평선 근처에 단일 객체가 존재하는 이미지 65장과 객체가 존재하지 않는 이미지 35장으로 총 100장을 테스트하였다.



(그림 12) 해상 데이터 세트 예시
(Figure 12) Maritime Datasets Example

4.2 그리드 분할 성능 비교

제안하는 알고리즘의 그리드 분할량에 따른 성능 차이를 확인하기 위하여, 데이터 세트에 그리드 분할 개수별로 수평선 검출 및 객체 존재 판별 알고리즘을 적용하여 비교하였다. 이는 알고리즘의 성능 분석을 진행하기 위하여, 그림 13처럼 그리드 분할량을 점차 증가시키며 결과를 분석하는 방식으로 진행하였으며, 성능평가 방식은 표 1과 같은 기준으로 분류하였다.



(그림 13) 그리드 분할 성능 비교

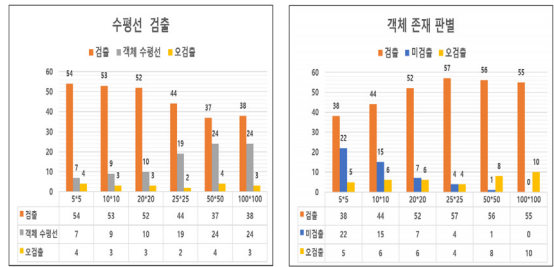
(Figure 13) Performance comparison for grid segmentation

(표 1) 그리드 분할 알고리즘 성능 평가 기준
(Table 1) Performance Evaluation Criteria for Grid Segmentation Algorithm

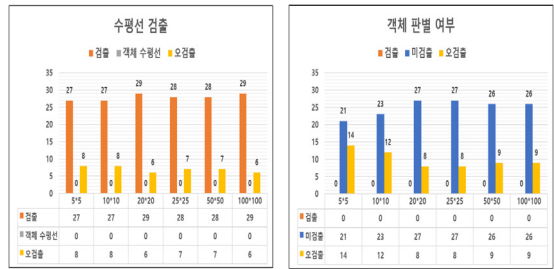
| | 분류 | 평가기준 |
|----------|--------|--|
| 수평선 검출 | 검출 | 그리드 기반으로 검출된 수평선 영역에 실제 수평선이 존재 |
| | 객체 수평선 | 객체의 일부에 수평선 검출 경계가 형성된 경우 |
| | 오검출 | 실제 수평선과 해상 객체를 제외한 영역에 검출된 경우 |
| 객체 존재 판별 | 검출 | 객체 일부에서 픽셀 변화량 임계치(20)가 이상인 경우 |
| | 미검출 | 탐색 영역에서 픽셀 변화량 임계치(20)가 미만인 경우 |
| | 오검출 | 해상 객체가 아닌 영역에서 검출된 경우 (수평선 오검출 시 객체 판별은 모두 오검출 처리) |

그리드 분할량에 따른 성능 비교 실험 결과는 그림 14와 같다. 객체 존재 이미지 65장과 존재 이미지 35장에 대한 알고리즘 적용을 상단과 하단에 각각 나타내었다. 그리드 분할량이 적을수록 크게 형성된 그리드 영역이 수평선 검출을 모호하게 표현하여 혼합된 색상으로 객체 검출률이 낮았다. 반면에, 분할량이 많을수록 정밀한 탐지를 하여 작은 객체 검출에 유리하였으나 잡음으로 인한 오검출도 증가하였다. 따라서, 적절한 값은 중간값인 그리드

분할 20x20과 25x25가 가장 좋은 성능을 보여주었다. 본 알고리즘은 색상 기반 알고리즘이기 때문에, 객체에 표현되는 수평선이 많았으나, 이는 보편적인 수평선 검출에서도 발생하며, 연계되는 객체 존재의 판별 과정에서 큰 영향을 끼치지 않으므로 탐지 목적성 검출로 취급하였다.



(해상 객체가 존재하는 이미지 - 65장 분류)



(해상 객체가 존재하지 않는 이미지 - 35장 분류)

(그림 14) 그리드 분할 성능 분석 결과

(Figure 14) Grid Segmentation Performance Analysis Results

객체 존재 판별 성능은 객체가 존재하는 이미지에서 검출률이 높아야 하며, 존재하지 않는 이미지에서는 미검출률이 높아야 한다. 분석 결과에서 객체가 존재하는 이미지에서 미검출은 멀리 있는 작은 객체에서 자주 나타나지만, 이는 추돌 감시를 목적으로 가까워질 때, 적당한 크기에서 검출될 것으로 예상된다. 필요시 그리드 분할량의 조절로 대응할 수 있다. 객체가 존재하지 않는 이미지의 오검출 요소는 빛에 의한 요소로 운슬, 노을 등이 있으며, 수평선 검출을 포함하여 영향이 있다. 빛은 해양 환경의 영상처리에서 주로 언급되는 잡음 요소이며, 픽셀의 변화량이 직접적인 광원의 위치까지 꾸준히 상승하여 대칭되어 패턴이 존재하는데, 이를 객체로 오검출하는 부분에서 이 자체를 객체로 탐지하고 추후 패턴분석을 통한 개선을 시도해볼 수 있다. 그 밖의 밀집된 파도 거품, 구름 등은 악천후를 제외하고 일시적인 시간 후에 완화될 수 있다.

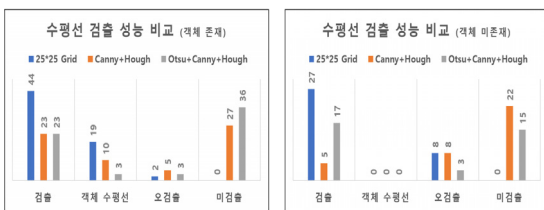
제안하는 검출 알고리즘의 수평선 처리 속도는 표 2와 같다. 처리 속도 산출은 해상도 500*500의 같은 조건의 해양 환경 영상에서 본 알고리즘을 적용한 후, 평균치를 구하여, 한 프레임에 대한 처리 속도로 나타내었다. 그리드 분할이 많아질수록 처리 속도가 작은 쪽으로 변동되지만, 이는 운용하는 임베디드 프로세서에 따라 더 유의미한 값을 가질 수 있다.

(표 2) 그리드 분할 처리 속도 비교
(Table 2) Processing Speed Comparison by Grid Segmentation

| | 수평선 검출 처리 속도 (1 frame) |
|--------------|------------------------|
| 5x5 grid | 8.5ms |
| 10x10 grid | 9.0ms |
| 20x20 grid | 9.3ms |
| 50x50 grid | 9.5ms |
| 100x100 grid | 11ms |

4.3 수평선 검출 알고리즘 성능 비교

수평선 검출은 객체의 대략적인 관심 영역을 찾아낼 수 있는 중요한 요소이다. 제안하는 알고리즘은 수평선 검출에서 연계하여 객체를 탐지하기 때문에, 개선된 효과를 보편적인 수평선 검출 알고리즘과 비교하여 증명한다. 본 알고리즘은 (25*25) 분할 크기의 그리드 수평선 검출을 적용하여, 캐니 에지 허프 검출과 오즈 알고리즘을 통한 캐니 에지 허프 검출을 비교한다. 캐니 에지의 임계치는 (1,255)으로 주었고, 비교 결과는 그림 15와 같다.



(그림 15) 수평선 검출 알고리즘 성능 비교
(Figure 15) Performance Comparison of Horizontal Detection Algorithms

비교 데이터는 실험 데이터 셋으로 이루어졌으며, 마찬가지로 객체가 존재하는 데이터 65장, 객체가 존재하지 않는 데이터 35장으로 구성되었다. 제안하는 알고리즘은

오검출 2개를 제외하고 모두 검출되었으며, 객체 수평선 부분을 제외한다고 하더라도 약 2배 이상의 검출률을 보인다. 검출 정확성 부분에서도 기존 알고리즘은 수평선 라인에 이중 중복 검출이 잦았고, 이를 검출로 반영하였지만, 제안하는 알고리즘은 수평선만 정확히 단일적으로 검출하였으며, 성능 또한 더 우수하였다. 또한, 기존 알고리즘은 전처리 연계에서 누락되어 수평선 자체에 대한 미검출률이 높는데, 이는 캐니 에지 알고리즘의 수평선에서 직선 성분의 윤곽선이 검출되지 않으면 허프 변환 직선 검출에서 수평선을 미검출하기 때문에, 이상적인 임계치 조절이 필요하다. 따라서, 기존의 수평선 검출 알고리즘으로 다양한 해양 환경에서 이를 범용적으로 적용하는 것은 어려움이 있다. 제안하는 알고리즘은 색상 변화량의 최대 차이를 수평선으로 검출하기 때문에, 미검출이 발생하지 않으며, 알맞지 않은 구간에 오검출이 발생할 수 있지만, 결과적으로 기존 알고리즘보다 전체적으로 더 좋은 성능을 보이고 있다. 또한, 제안하는 알고리즘과 기존 알고리즘의 연산 처리 속도 비교를 표 3에서 나타내었다.

(표 3) 제안된 수평선 검출 알고리즘과 처리 속도 비교
(Table 3) Comparison of processing speed with proposed horizontal line detection algorithm

| Raspberry Pi4 | 수평선 검출 처리 속도 (1 frame) | | |
|----------------|------------------------|------------------|--------------|
| Algorithm type | Canny+Hough | Otsu+Canny+Hough | Our Proposed |
| Speed | 51ms | 47ms | 8.5 ~ 11ms |

기존의 허프 변환을 통한 수평선 알고리즘은 보편적인 수평선 검출 알고리즘이지만, 처리 속도가 느린 단점이 있었다. 이와 임계치 수동 설정을 극복하기 위한 (Otsu) 이미지 이진화를 통해 처리하는 방법도 존재하였다. 각 알고리즘의 처리 속도 측면에서 제안하는 알고리즘과 비교하여 기존 보편적인 수평선 알고리즘 대비 5배로 빠르게 처리할 수 있음을 실험을 통해 확인하였다. 해당 실험 결과는 사용하는 프로세서에 따라 차이가 달라지며, 저가형일 경우 처리 속도의 차이가 더 큰 속도 차이를 보일 수 있다. 이는 배터리 환경의 저전력 시스템을 운용하는 상황에서 더 효율적으로 활용할 수 있다.

4.4 제안하는 알고리즘의 딥러닝 연계

제안하는 알고리즘을 다루는 실험은 4.3절까지이며, 본 알고리즘을 전처리로 활용하여 딥러닝을 연계하였을 때, 그에 대한 운용 방법과 시뮬레이션 구성을 소개한다. 만일, 감지 알고리즘에 의해 항로표지가 객체를 사전에 감지하였을 때, 그 과정에서 감지 여부가 실제 위험 요소나 오작동 여부에 대한 판별은 사람에게 의해 모니터링이 필요하다. 결국 알고리즘의 정확성 유무에 상관없이 인적요소를 통해 확인해서 대처해야 한다. 딥러닝은 사람이 확인 해주어야 할 객체를 대신 확인하여, 연계되는 위험 대처 시스템에 보조할 수 있다. 또한, 영상처리로 어려움이 있는 잡음 요소에도 2차적 판별하여, 오검출에 대한 사항을 안정적 관리가 가능하다.

하지만, 딥러닝 시스템은 고성능 사양을 요구하기 때문에 비용적 문제가 존재하며, 전력 소모가 크게 발생하여 해상 환경에서 임베디드 시스템으로 항시 운용하기에 안정성이 부족하다. 따라서, 제안하는 알고리즘을 1차적 전처리로 연계하여 저전력 시스템으로 객체의 존재 여부만을 상시 판별한 뒤, 딥러닝을 조건부 알고리즘으로 특정 상황에서만 간헐적으로 활용함으로써, 연산량 및 전력 제어를 효율적으로 운용할 수 있다. Raspberry-Pi4 임베디드 실험 환경에 YOLOv4 모델을 구성하여, 실험 데이터 세트에 대한 딥러닝 검출 성능을 확인하였으며, 그에 대한 결과는 표 4와 같다.

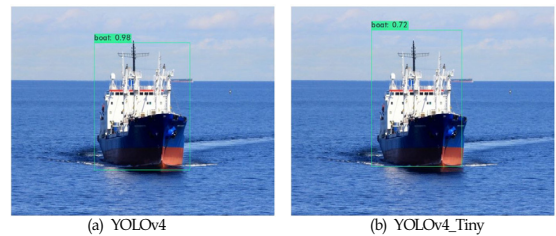
(표 4) Raspberry-Pi4 환경의 Yolov4 성능
(Table 4) YOLOv4 Performance in Raspberry Pi4 Environment

| Pi4 | YOLOv4 | | YOLOv4_Tiny | |
|--------|-----------|------------|-------------|------------|
| | 객체 존재(65) | 객체 미존재(35) | 객체 존재(65) | 객체 미존재(35) |
| 데이터 검출 | 64장 검출 | 35장 미검출 | 54장 검출 | 35장 미검출 |
| 처리 속도 | 64s | | 7.2s | |

Yolov4의 학습모델은 COCO 데이터 세트를 이용하였으며, 입력 이미지 크기는 416x416로 검출하였다. 표 4의 검출 결과는 실험 데이터 세트 100장 중 해상 객체가 존재하는 이미지 65장에 대하여, Yolov4 모델은 1장을 제외하고 전부 검출하였고, Tiny 모델은 11장을 제외한 54장을 검출하였다. Tiny 모델은 주로 가깝고 큰 객체에 대해 미검출하였다. 객체가 존재하지 않는 이미지 35장에 대해서는 두 모델 모두 미검출하였다. 검출률은 인식률과 별

개로 딥러닝 모델이 객체 자체를 분류한 것을 의미한다.

그림 16은 YOLOv4 모델의 해상 객체 검출의 예시이다. 수평선 인근의 객체 감지를 고려하였을 경우, Tiny가 본래 모델보다 성능이 조금 부족하더라도 처리 속도가 7초이기 때문에, 먼 거리에서부터 다가오는 객체를 신속하게 판별하여 2차 검출할 수 있다. 따라서, 제안하는 알고리즘을 1차 전처리로 객체 존재 여부 판별 후, 인적 확인 요소를 대신하여 2차 판별 용도로 조건부 응용이 가능함으로 판단된다.



(그림 16) 딥러닝 YOLOv4 해양 객체 검출 예시
(Figure 16) Example of maritime object detection in YOLOv4 model

5. 결 론

본 논문에서는 해상 안전을 위해 카메라로 영상을 입력받고, 이를 통해 위험 요소를 판단할 수 있는 해상용 객체 감지 시스템을 설계하였다. 일반적으로 항로표지의 부표는 배터리에 의해 의존되기 때문에 전력이 매우 중요하며, 현재 항로표지 환경을 감시하기 위해서는 서버에서 관리자가 직접 영상을 열람하거나 충돌 센서에 의해 알림을 받는 형태이다. 제안하는 시스템은 일정한 주기를 통해 캡처하여, 빠른 이미지 분석을 통해 관리자에게 알림 유무를 제공한다. 이는 해양의 특성에 알맞게 변형하였고 저 연산을 시스템을 위해, 단순하고 효율적인 객체 감지 알고리즘 시스템을 제안하였다. 기존 보편적으로 사용되는 수평선 검출보다 연산량을 줄이고 검출률을 높였으며, 실험을 통해 성능을 기존보다 높게 유지하면서 처리 속도를 5배 이상 빠르게 처리하는 것을 증명하였다. 따라서, 관리자는 본 시스템을 이용하여 시설에 관한 사고 관리를 후처리가 아닌 예방 조치의 용도로 활용할 수 있다. 또한, 딥러닝 영상인식과 제안하는 시스템과 연계하여 더 진보적인 운용을 할 수 있는 방향을 제시하였다. 제안하는 알고리즘을 통해 딥러닝의 소모 전력을 낮출 수 있으며, 딥러닝은 제안하는 알고리즘이 색상 기반으로 감지할 수 없

는 객체나 객체 판별 후 사람의 2차 판별이 필요한 역할을 대체할 수 있는 상호보완적 요소가 존재한다. 현재는 시뮬레이션 레벨로 이미지를 선정하여 소프트웨어 처리 하였지만, 기상 변화, 배터리 효율 등을 고려한 개선된 알고리즘과 하드웨어 및 통신 프로토콜이 정밀하게 설계된다면 상용화 수준까지 가능할 것이라 사료된다.

참고문헌(Reference)

- [1] K. S. Pachpor, Deshmukh, Chandrashekar, Pachpor Kapil. "Home Automation and Surveillance: A Review", International Journal of Science and Engineering Applications, Vol. 8, No. 4, pp. 111-114, 2019.
<http://dx.doi.org/10.7753/IJSEA0804.1002>
- [2] S. Jain, A. Jatain and S. Bhaskar, "Smart City Management System using IoT with Deep Learning", 2019 International Conference on Communication and Electronics Systems(ICCES), pp. 1214-1222, 2019.
<http://dx.doi.org/10.1109/ICCES45898.2019.9002414>
- [3] D. H. Lee, S. N. Kim, W. S. Baek, I. G. Jang, S. J. Kim, "항로표지 관리운영시스템을 통한 효율적 운영방안", Korean Institute of Navigation and Port Research, Nov. 02, pp. 154-155, 2017.
<https://www.koreascience.or.kr/article/CFKO201716269584184.page>
- [4] M. A. Hashmani, M. Umair, S. S. H. Rizvi, A. R. Gilal, "A Survey on Edge Detection based recent Marine Horizon Line Detection Methods and their Applications", 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), pp. 1-5, 2020.
<http://dx.doi.org/10.1109/iCoMET48670.2020.9073895>
- [5] Ponnaganti V., Moh M., Moh TS. "Deep Learning for LiDAR-Based Autonomous Vehicles in Smart Cities", In: Augusto J.C. (eds) Handbook of Smart Cities, pp. 957-980, Springer, Cham., 2021.
http://dx.doi.org/10.1007/978-3-030-69698-6_65
- [6] H. G. Hwang, B. S. Kim, H. W. Kim, Y. S. Gang, D. H. Kim, "A Development of Active Monitoring and Approach Alarm System for Marine Buoy Protection and Ship Accident Prevention based on Trail Cameras and AIS" Journal of the Korea Institute of Information and Communication Engineering, Vol. 22, No. 7, pp 1021-1029, Jul. 2018.
<http://dx.doi.org/10.6109/jkiice.2018.22.7.1021>
- [7] J. Canny, "A Computational Approach to Edge Detection", in IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 6, pp. 679-698, Nov. 1986.
<http://dx.doi.org/10.1109/TPAMI.1986.4767851>
- [8] N. Otsu, "A threshold selection method from gray-level histograms", IEEE Trans. Systems Man Cybernet, Vol. 9, No. 1, pp. 62-66, 1979.
<http://dx.doi.org/10.1109/TSMC.1979.4310076>
- [9] P. V. C. Hough, "Method and Means of Recognizing Complex Patterns", Dec. 1962.
<https://patents.google.com/patent/US3069654A/en>
- [10] X. Fang, W. Xiong, B. Hu, L. Wang, "A Moving Object Detection Algorithm Based on Color Information", International Symposium on Instrumentation Science and Technology, Vol. 48, pp. 384-387, 2006.
<http://dx.doi.org/10.1088/1742-6596/48/1/072>
- [11] O. Boufares, M. Boussif and N. Aloui, "Moving Object Detection System Based on the Modified Temporal Difference and OTSU algorithm", 2021 18th International Multi-Conference on Systems, Signals & Devices (SSD), pp. 1378-1382, 2021.
<http://dx.doi.org/10.1109/SSD52085.2021.9429516>
- [12] Praczyk, Tomasz, "A quick algorithm for horizon line detection in marine images", Journal of Marine Science and Technology, Vol. 23, No. 1, pp. 164-177, 2018.
<http://dx.doi.org/10.1007/s00773-017-0464-8>

◎ 저 자 소 개 ◎



전 호 석(Ho-Seok Jeon)

2020년 호서대학교 정보통신공학과(공학사)
2022년 호서대학교 대학원 정보통신공학과(공학석사)
관심분야 : 컴퓨터 비전, 딥러닝, 임베디드 펌웨어
E-mail : wjsghtjr33@naver.com



송 현 학(Hyun-Hak Song)

2019년 호서대학교 정보통신공학과(공학사)
2021년 호서대학교 정보통신공학과(공학석사)
2021년~현재 한국전자기술연구원 연구원
관심분야 : 컴퓨터 비전, 인공지능
E-mail : rainy930@keti.re.kr



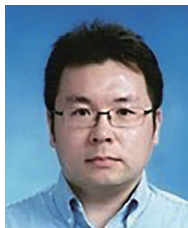
권 기 원(Ki-Won Kwon)

1997년 광운대학교 컴퓨터공학과(공학사)
1999년 광운대학교 대학원 컴퓨터공학과(공학석사)
2011년 중앙대학교 대학원 전자전기공학과(공학박사)
1999년~현재 한국전자기술연구원 해양수산ICT사업단 단장
2022년~현재 한국전자기술연구원 스마트네트워크연구센터 센터장
관심분야 : 디지털트윈, 유무선디지털통신·네트워크시스템, 해양수산ICT융합
E-mail : kwonkw@keti.re.kr



김 영 진(Young-Jin Kim)

2000년~2018년 해양수산부, 인천·울산·평택지방해양수산청
2019년 부산지방해양수산청 제주해양수산관리단 항로표지팀장
2020년~2021년 울산지방해양수산청 항로표지과 과장
2021년 한국해양대학교 대학원 해양경찰학과(공학석사, 해양안전환경전공)
2022년~현재 해양수산부 항로표지과 해양수산사무관
관심분야 : 토목, 항로표지, 스마트항로표지
E-mail : ds3kgu@korea.kr



임 태 호(Tae-Ho Im)

2012년 중앙대학교 대학원 전자전기공학과(공학박사)
2012년~2015년 삼성전자 DMC연구소 책임연구원
2015년~현재 호서대학교 해양IT융합기술연구소 조교수
2019년~2020년 호서대학교 정보통신공학과 조교수
2021년~현재 호서대학교 정보통신공학과 부교수
관심분야 : 5G 이동통신, LPWAN, 수중통신, 딥러닝
E-mail : taehoim@hoseo.edu