

항로표지 충돌 방지를 위한 영상 전처리 알고리즘과 딥러닝을 활용한 해상 객체 고속 검출[☆]

High-Speed Maritime Object Detection Using Image Preprocessing Algorithms and Deep Learning for Collision Avoidance with Aids to Navigation

김 영 민¹ 권 기 원² 임 태 호^{1*}
Young-Min Kim Ki-Won Kwon Tae-Ho Im

요 약

해양에서 사용되는 부이와 같은 항로표지는 항해 중인 선박들에게 정확한 정보를 제공하여 선박이 자신의 위치를 명확히 파악하고, 주변의 위험 지역을 표시하여 안전한 항로를 유지하는데 필수적인 역할을 한다. 하지만, 선박과의 충돌 사고로 인해 부이의 파손 및 복구에 막대한 비용이 소요된다. 이러한 사고를 예방하기 위해 현재 고가형 장비를 사용하기도 하지만 비용 부담 문제로 도입하기 쉽지 않다. 해당 문제를 해결하기 위해 본 논문에서는 임베디드 시스템을 활용한 해상 객체 검출 알고리즘 연구를 진행한다. 기존 연구에서는 수평선 검출을 위해 허프 변환을 사용하였지만, 연산량이 많아 실시간 처리가 어렵다는 한계가 존재한다. 이를 개선하기 위해 본 논문에서는 이미지 분할을 진행한 후, Otsu 알고리즘을 최적화하여 수평선을 검출한다. 또한, 검출된 수평선을 기준으로 선박 충돌 위험성이 높은 구역을 위험 영역(Region of Interest, ROI)으로 설정한다. 위험 영역을 설정한 이후, 수평선 아래에 위치한 영역에서 해상 객체를 검출하고 해당 영역에 CNN(Convolutional Neural Network)모델을 활용하여 해당 객체의 선박 여부를 판별한다. 만약 선박으로 판별된 객체가 위험 영역에 존재하는 경우, 이를 위험 선박으로 간주한다. 제안하는 알고리즘은 항로표지 충돌 사고를 사전에 방지할 수 있다.

☞ 주제어 : 수평선 검출, 선박 검출, 이진화, 항로표지, 이미지 분할

ABSTRACT

Aids to navigation, such as buoys used in maritime environments, play a crucial role in providing accurate information to navigating vessels, enabling them to precisely determine their position and maintain safe routes by marking surrounding hazardous areas. However, collisions between ships and these aids result in substantial costs for buoy damage and repair. While high-end equipment is currently used to prevent such accidents, its widespread adoption is hindered by cost concerns. This paper presents research on a maritime object detection algorithm utilizing embedded systems to address this issue. Previous studies employed the Hough transform for horizon detection, but its high computational demands posed challenges for real-time processing. To overcome this limitation, our approach first performs image segmentation, followed by an optimized Otsu algorithm for horizon detection. Subsequently, we establish a Region of Interest (ROI) based on the detected horizon, focusing on areas with a high risk of ship collision. Within this ROI, particularly below the horizon line, maritime objects are detected. A Convolutional Neural Network (CNN) model is then applied to determine whether the detected objects are ships. Objects classified as ships within the ROI are considered potential collision risks.

☞ keyword : Horizontal line detection, vessel detection, binarization, route sign, image segmentation

¹ Information and communication, Hoseo University, Asan, 31499, Korea

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

* Corresponding author (tahoim@hoseo.edu)

[Received 23 September 2024, Reviewed 25 September 2024, Accepted 2 October 2024]

☆ 이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-지역지능화혁신인재양성사업의 지원(IITP-2024-RS-2024-00436765,50%)과 2024년 해양수산부 재원으로 해양수산과학기술진흥원의 지원을 받아 수행된 연구임(스마트항로표지현장시설 고도화, RS-2021-KS211516, 50%).

1. 서 론

해양 환경에서 사용되는 부이와 같은 항로표지는 항해 중인 선박들에게 정확한 정보를 제공하여, 안전한 항로를 유지하는 데 중요한 역할을 한다. 항로표지는 선박이 자신의 위치를 명확히 파악하고, 주변의 위험 요소나 장애물을 피할 수 있도록 지형, 수로의 변화 및 위험 지역을 표시함으로써 충돌이나 좌초 사고를 예방하고, 해상 교통의 원활한 흐름을 유지하는 데 기여한다. 그러나 매년 선박과의 충돌 사고로 인해 부이가 파손되며, 바다에 떠 있는 특성상 복구 또는 교체하는 데 상당한 비용이 소요된다.

해양수산부 통계에 따르면, 2023년에는 총 3,092건의 해양 사고가 발생하였으며, 이는 전년 대비 229건(8.0%) 증가한 수치이다. 그 중 주요 사고 중 하나인 선박 간의 충돌 및 항로표지와의 충돌이 39.4%를 차지하고 있다. 이러한 사고를 예방하기 위해 레이더나 소나 같은 고가형 장비를 사용하지만, 높은 비용으로 인해 도입이 어려운 실정이다. 이에 비해 임베디드 시스템을 활용한 해상 객체 검출 알고리즘은 비교적 저렴한 비용으로 선박과 같은 객체를 효과적으로 인식하고, 조기에 위험을 감지하여 사고를 사전에 예방할 수 있는 가능성을 제공한다.

해상 객체 검출 알고리즘의 정확도와 속도를 높이기 위해서는 수평선 검출의 신속성과 정확도가 매우 중요하다. 본 연구에서는 수평선 검출을 통해 선박 충돌 위험 구역을 설정하고, 해당 수평선을 기준으로 객체 검출 영역을 설정하여 기존 연구보다 더 신속하고 정확한 검출 방법을 제안한다.

본 논문에서는 첫째, 임베디드 시스템 기반 해상 객체 검출 알고리즘을 제안하였다. 이를 통해 고가형 장비 없이도 저렴한 비용으로 선박과 같은 해상 객체를 실시간으로 검출할 수 있는 방법을 제시하였다. 둘째, 기존 허프 변환 대신 Otsu 알고리즘을 최적화하여 수평선을 더욱 빠르고 정확하게 검출하는 방법을 제안하였다. 셋째, 수평선을 기준으로 위험 영역(ROI)을 설정하고, 해당 영역 내에서 선박과 같은 해상 객체를 효과적으로 검출하는 방안을 제시하였다. 넷째, 경량화된 CNN 모델인 EfficientNet-Lite를 적용하여 임베디드 시스템 환경에서도 실시간 객체 검출을 수행할 수 있도록 하였다. 마지막으로, 실험을 통해 기존 방법 대비 처리 속도를 약 5배 이상 향상과 동시에 검출 정확도를 유지하는 성능을 입증하였다.

2. 관련 연구

2.1 해상 객체 검출 알고리즘

현재 카메라를 이용한 비전 알고리즘이 원격 제어, 자율 주행 차량 많은 곳에서 사용되고 있다[1]. 하지만 주로 육상 환경을 다루고 있어, 해상 환경에서 발생하는 문제를 분석하고 해결하는 연구의 증진이 필요하다. 본 논문에서는 특히 항로표지의 충돌 사고를 방지하기 위해 항로표지에 근접한 해상 객체 검출에 대해 다루며, 먼저 이와 관련한 연구를 소개한다.

충돌로 인한 선박과 항로표지의 손상은 많은 수리 비용을 요구할 뿐 아니라 해상 교통 환경에도 많은 영향을 미치므로, 해상 객체 검출 기술 알고리즘들이 개발되고 있다[2]. Zhang[3]은 해상 환경에서 수평선 검출을 진행하여 바다 영역과 하늘 영역을 구분하고 수평선 아래의 해수면 영역을 추출하여 해당 영역에 DCT(Discrete Cosine Transform)를 기반으로 한 가우시안 혼합 모델을 사용하여 파도와 물결과 같은 동적 특성을 모델링 한 뒤, 해수면 영역에서 객체를 검출 알고리즘을 제안했다. 또 “Marine Vision-Based Situational Awareness Using Discriminative Deep Learning: A Survey” 논문에서 따르면 해상 환경을 배경으로 하여 수평선 검출 또는 해상 객체를 검출하는 다양한 연구가 존재한다[4]. Steccanella[5]는 CNN을 사용해 이미지에서 물과 물이 아닌 영역을 픽셀 단위로 분할한 후, RANSAC(Random Sample Consensus)을 적용해 수평선을 추정하고, 수평선 아래의 물체들을 검출하는 방안을 제안했다. Bovcon[6]는 자율 운항 선박의 해양 장애물 검출을 위해 시멘틱 세그멘테이션과 IMU(관성 측정 장치)를 결합하여 수평선을 정확하게 분할하고, 수평선 이하 영역의 장애물을 검출하는 방안을 제안했다.

2.2 수평선 검출 알고리즘

본 연구에서 수평선은 위험 영역을 설정하는 기준이 되기에 기존 연구보다 정확한 수평선 검출이 필요하다. 기존 연구에서는 수평선을 검출하기 위하여 주로 케니엣지 기반의 허프 변환(Hough Transform) 알고리즘을 사용한다. [7] 허프 변환은 컴퓨터 비전 분야에서 이미지의 선과 원 같은 단순한 형태를 찾는 방식이다. 이진화된 이미지에서 직선을 찾을 때 상대적으로 빠른 검출을 보여주며 자율 주행 자동차에서 차선을 검출하거나 해상 환



(그림 1) 해상 객체 검출 순서도
(Figure 1) maritime object detection flow chart

경에서 수평선을 검출하는 데 활용된다. 허프 변환은 이미지의 각 엣지 픽셀을 파라미터 공간으로 변환하고 다양한 각도와 거리를 계산하여 직선을 검출한다. 이를 통해 노이즈가 많은 이미지에서도 직선을 효과적으로 검출할 수 있다. 하지만 이미지의 각 엣지 픽셀에 대해 가능한 각도를 모두 계산해야 하기에 연산량이 많아 임베디드 시스템에 적용하는 것은 어렵다[8]. 이러한 문제를 해결하기 위해 Chi[9]는 영역을 제한하여 수평선을 검출하는 방안을 제안하였다. 해당 논문은 수평 모양으로 이미지를 분할한 상태에서 바다와 배경의 색상 특성을 이용하여 수평선이 존재할 수 있는 ROI 영역을 제한한다. 이를 통해 처리 시간을 단축 시킬 수 있게 된다. 이후, 검출된 ROI 영역을 다중 스케일 에지 검출을 활용하여 단일 결합 에지 맵을 생성하고 최종적으로 수평선 최적의 파라미터를 찾기 위해 최소제곱법을 사용한다. 해당 논문에서 제안한 방법은 기존의 수평선 검출과 비교하였을 때, 정확도를 저하시키지 않으면서도 처리 시간을 단축시켰다. 하지만 하늘과 바다의 경계가 명확하지 않은 경우 수평선 오검출이 발생한다는 한계가 존재한다.

반면에 이진화 알고리즘은 구현이 간단하며, 연산 속도가 빠르다는 장점이 있지만, 이진화 알고리즘은 이미지 전체에 동일한 임계값(Threshold)으로 이진화를 진행하므로 적절한 임계값을 설정해야 한다는 문제점이 있다. 적절한 임계값을 설정해준다면 바다와 배경의 경계가 확실해져 정확한 수평선 검출에 상당히 적합하다. 이진화 알고리즘을 활용하여 수평선을 검출하고 해상 객체를 검출하는 과정을 3장에서 설명하도록 하겠다.

3. 제안하는 해상 객체 고속 검출 방법

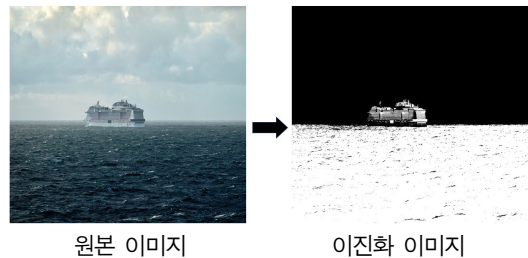
본 논문에서 제안하는 해상 객체 검출을 위한 알고리즘의 순서도는 그림 1과 같다. 제안하는 전체 작업 프로세스는 수평선 검출, 위험 영역 선정, 객체 검출, 선박 검출,

위험 선박 판별로 구성된다. 먼저, 수평선 검출 단계에서는 이진화 기법을 이용해 배경과 바다를 분리하여 수평선을 검출한다. 다음으로, 위험 영역 선정 단계에서는 검출된 수평선을 기준으로 선박 충돌 위험성이 높은 구역을 위험 영역(Region of Interest, ROI)으로 설정한다. 이어지는 객체 검출 단계에서는 바다 영역을 제거하고 검출한 객체들을 바운드 박스로 표시한 뒤 선박 검출 단계에서 CNN 모델을 사용해 이 객체들이 선박인지 판별하고 분류한다. 마지막으로, 위험 선박 판별 단계에서는 선박으로 판별된 객체가 설정된 위험 영역 내에 있을 경우 해당 선박을 위험 선박으로 간주한다.

3.1 제안하는 수평선 검출 알고리즘

3.1.1 이진화를 통한 바다와 배경 분리

본 연구에서는 해상 객체 검출의 처리 시간을 단축하기 위해 수평선 검출 과정에서 이진화 알고리즘을 활용하였다. 일반적인 이진화 알고리즘은 이미지 전체에 동일한 임계값을 적용하므로, 적절한 임계값을 설정하는 것이 중요하다. 그러나 가변 임계값을 제공하는 Otsu 알고리즘을 사용하면 이러한 문제를 효과적으로 해결할 수 있다.



(그림 2) 해상 이미지에 Otsu 알고리즘 적용
(Figure 2) Apply the Otsu to the Sea Image

Otsu 알고리즘은 이미지 이진화에서 널리 사용되는 방법으로, 전체 이미지의 히스토그램을 분석하여 최적의 임계값을 자동으로 선택해준다. 이 알고리즘은 두 클래스 간의 분산을 최대화하는 임계값을 찾아내는 것을 목표로 하므로, 환경 변화가 적은 해상 환경에서 바다와 배경을 분리하는 데 적합하다[10].

그림 2와 같이, 해상 이미지에서 바다와 배경 영역이 분리된 후, 배경과 바다 사이의 수평선이 검출된다. 본 연구에서는 이진화된 이미지의 양쪽 끝에서 y축을 따라 첫 번째로 검출되는 1값(흰색) 픽셀을 탐지하고, 이를 연결하여 수평선을 추출하는 방법을 사용하였다. 바다와 배경이 명확하게 분리된 경우, 이 방법을 통해 신속하게 수평선을 검출할 수 있다.



수평선 검출 성공 이미지 수평선 검출 실패 이미지

(그림 3) Otsu 알고리즘 적용 후 수평선 검출
(Figure 3) Horizontal line detection after Otsu

3.1.2 이미지 분할을 통한 수평선 검출

그림 3은 Otsu 이진화 알고리즘을 적용한 후 수평선 검출 과정에서 발생하는 오검출 문제를 나타낸다. 해상 환경에서는 선박과 같은 객체로 인해 수평선 검출 시 수평선의 기울기가 급격히 변하는 현상이 종종 발생한다. 이러한 문제를 해결하고 수평선을 보다 효과적으로 검출하기 위해, 본 연구에서는 이미지를 7개의 수직 세그먼트로 나누는 방법을 제안한다.

다양한 세그먼트 수를 적용한 실험 결과, 세그먼트 수가 적을 경우 각 세그먼트의 영역이 넓어져 세밀한 분석이 어려워졌으며, 반대로 세그먼트 수가 많을 경우 연산량이 증가하여 처리 시간이 길어지는 현상이 나타났다. 최적의 성능은 7개의 세그먼트로 이미지를 분할했을 때, 수평선 검출의 정확도와 처리 속도 측면에서 얻을 수 있었다.

제안하는 수평선 검출 과정은 다음과 같다. 먼저 이미지를 7개의 수직 세그먼트로 나눈다. 각 세그먼트의 너비는 이미지 전체 너비를 7로 나눈 값으로 설정한다. 이후



(그림 4) 이미지 분할을 통한 수평선 검출
(Figure 4) Horizontal line detection with image segmentation

각 세그먼트 내에서 독립적으로 Otsu 알고리즘을 사용하여 이진화를 수행한다. 이진화된 각 세그먼트에서 가장자리(좌측과 우측)에서 첫 번째로 발견되는 1값(흰색) 픽셀의 y좌표를 탐지하고, 이 좌표들을 연결하여 수평선을 검출한다. 이후 검출된 수평선 중 특정 기울기 범위 내에 있는 것들만을 모아 표준 편차와 평균값을 계산한 후, 최적의 수평선을 최종적으로 선택한다.

이미지 데이터셋 분석 결과, 대부분의 수평선은 기울기가 0에 매우 가까운 것으로 확인되었다. 그러나 해상 환경에서는 파도, 바람, 그리고 항로표지나 부표와 같은 해양 구조물의 기울어짐과 같은 외부 요인들로 인해 수평선이 미세하게 기울어질 수 있다. 이러한 외부 요인들로 인해 발생하는 기울기를 수용하면서도 수평선의 정확한 검출을 보장하기 위해, 본 연구에서는 기울기 범위를 -0.1에서 0.1로 설정하였다.

그림 4는 이미지를 세그먼트로 나누어 수평선을 검출하는 과정을 보여주며, 그림 5는 오검출이 발생하던 이미지에 제안한 방법을 적용하여 수평선을 성공적으로 검출한 결과를 나타낸다.

3.2 위험 선박 검출 프로세스

3.2.1 관심 영역 선정

수평선을 검출한 후, 해당 수평선을 기준으로 항로표지에 설치된 임베디드 카메라의 관점에서 해상 환경을 관찰할 때, 수평선 근처에 위치한 선박은 항로표지와 일



(그림 5) 이미지 분할 전, 후 수평선 검출 결과 비교
(Figure 5) Compare horizontal line detection before and after image segmentation

정한 물리적 거리를 유지하고 있어 충돌 위험이 비교적 낮다. 반면, 수평선 아래쪽 영역으로 가까워질수록 선박과 항로표지 간의 충돌 위험은 점차 증가한다. 따라서 본 연구에서는 수평선을 기준으로 원본 이미지의 상단 30% 영역을 제외하고 나머지 하단 영역을 충돌 위험 구역으로 설정하여, 해당 구역 안에 존재하는 선박을 충돌 위험 선박으로 간주한다.

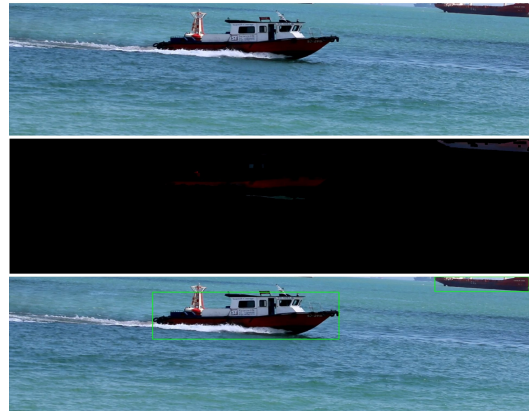
해당 접근 방식은 이미지 상단 영역에서 발생하는 불필요한 분석을 줄이고, 보다 중요한 하단 영역에 분석 자원을 집중할 수 있다. 만약 수평선을 검출한 결과, 수평선 아래부분이 원본 이미지의 30%에도 미치지 않는 경우, 항로표지가 파도와 같은 환경적 요인으로 인해 기울어졌을 가능성이 높다고 판단하여 해당 이미지를 분석 대상에서 제외한다. 이러한 방식을 그림 6에서 구현하였다.



위험 영역 선정
(그림 6) ROI 선정
(Figure 6) Select ROI

3.2.2 해상 객체 검출

검출한 수평선을 이용해 위험 영역을 선정하였다면, 검출한 수평선 아래의 영역에서 객체 검출을 진행한다.



(그림 7) 바다 영역 제거를 통한 객체 검출
(Figure 7) Object detection by removing the sea area

데이터셋에서 바다의 RGB 값을 분석한 결과, 파란색 채널(B)의 값이 평균적으로 90 이상이었으며, 초록색 채널(G)의 값은 B값과 유사한 수치를 보이는 것을 확인할 수 있었다. 이러한 바다의 색상 특성을 활용하여, 본 연구에서는 관심 영역 내에서 파란색 채널(B)과 초록색 채널(G)의 값이 모두 90 이상인 픽셀을 바다로 간주하고, 해당 바다 영역을 제거하는 방식을 채택하였다.

바다 영역을 제거한 후에는 모폴로지 연산을 통해 남아 있는 노이즈를 제거하고, 객체의 경계를 더욱 강화한다. 이후 컨투어 검출 기법을 사용하여 객체를 찾아내고, 각 객체에 바운딩 박스를 표시한다. 바운딩 박스를 표시하는 과정에서 중복된 바운딩 박스를 제거하기 위해 IoU(Intersection over Union) 검증 방법을 적용하여 객체 검출의 정확도를 향상시킨다. 그림 7과 같이 객체를 검출을 진행한 뒤 CNN 모델을 이용하여 해당 객체의 선박 여부를 확인한다.

3.3 EfficientNet-Lite

컨볼루션 신경망(CNN)은 이미지 인식 및 분류 작업에서 널리 사용되는 심층 학습 모델이다. CNN은 입력 이미지로부터 특징을 자동으로 추출하여 분류 작업을 수행하는 과정에서 탁월한 성능을 보인다. CNN을 이용하여 이미지나 영상 속 객체가 존재하는 위치를 검출할 수 있으며 해당 모델로 R-CNN, Fast R-CNN, SSD, YOLO 등이 등장하였다. 최근에는 MobileNet, ResNet, EfficientNet 등과 같이 연산 자원이 제한된 환경에서도 실시간 성능을 요



(그림 8) 제안 기법 수평선 검출 결과
(Figure 8) Horizon Detection Results for suggested algorithms

구하는 임베디드 시스템에서도 사용 가능한 경량화 모델들이 등장하고 있다. MobileNet, ResNet, EfficientNet 중에서 본 연구에서는 EfficientNet을 선택했다. 그 이유는 MobileNet은 경량화된 구조 덕분에 모바일 환경에서 빠른 속도를 제공하지만, 정확도 면에서 한계가 있다. ResNet은 깊은 네트워크로 복잡한 패턴을 잘 학습할 수 있으나, 높은 연산량과 메모리 요구량 때문에 실시간 처리나 저전력 장치에서는 비효율적이다. 반면, EfficientNet은 복합 스케일링(compound scaling) 기법을 통해 적은 연산 자원으로도 높은 정확도를 달성할 수 있어, 정확도와 자원 효율성 사이에서 최적의 균형을 제공한다. 그러나 본 연구에서는 경량화와 저전력 소비가 핵심 과제인 만큼, 더 가볍고 효율적인 EfficientNet-Lite 모델을 사용하여 이를 해결하고자 한다[11-12].

EfficientNet-Lite 모델은 Squeeze-and-Excitation(SE) 블록을 제거하여 모델의 복잡도를 줄이고 경량화하여 모바일 및 임베디드 시스템에서 실시간 성능을 제공하면서 높은 정확도를 유지할 수 있게 해준다. EfficientNet-Lite는 연산량의 크기에 따라 Lite0부터 Lite4까지 총 5가지 변형 모델을 제공한다. 본 연구에서 사용하는 임베디드 시스템은 라즈베리파이 CM4 모델로, 쿼드 코어 Cortex-A72(Arm v8) 64비트 SoC를 탑재하고 있다. 또한, 이 임베디드 시스템이 설치된 항로표지가 육지로부터 먼 해상에 위치해 있다는 점을 고려하여, 연산량이 가장 적은 저전력의 Lite0 버전을 선택하였다.

4. 실험 및 성능 평가

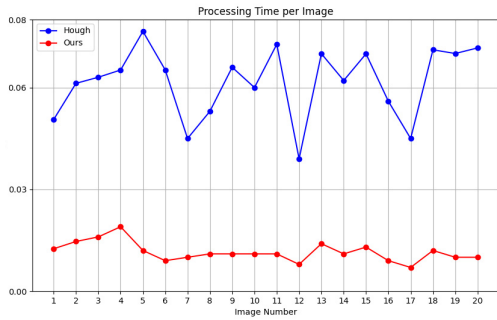
본 장에서는 실험 환경을 설명한 뒤, 실제 해상 데이터셋을 활용하여 제안된 기법의 수평선 검출 및 해상 객

체 검출한 실험 결과를 제시한다.

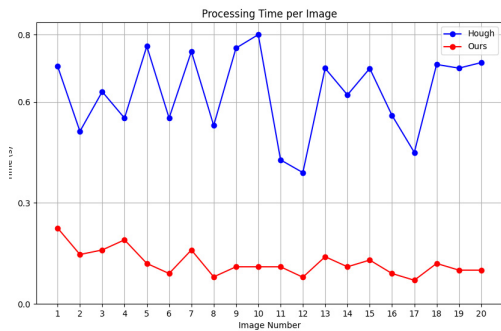
4.1 실험 환경 및 성능 검증

본 논문의 OS환경에서는 CPU 12th Intel(R) Core(TM) i7-12700@2.1Ghz이고, RAM은 32G, 운영체제는 Windows 10.0-64bit, GPU인 PC를 이용했고, 임베디드 시스템 환경에서는 코어 Cortex-A72(Armv8) 64비트 SoC를 탑재한 라즈베리파이 CM4 모델을 사용했다. 데이터셋은 싱가포르 해양 데이터셋(Singapore Maritime Dataset, SMD)의 일부 데이터와 온라인 검색 엔진을 활용하여 수집한 다양한 해상 환경 이미지를 결합하여 SBD(Sea Boat Dataset)을 구축하였으며, 이를 바탕으로 실험을 진행했다. 본 논문에서는 SBD를 기반으로 기존의 수평선 검출 방법과 제안하는 수평선 검출 방법을 비교하였다. 또한 라즈베리파이 환경에서 제안된 기법을 적용하여 해상 객체 검출의 성능을 평가하였다.

본 논문에서 제안하는 기법을 통한 수평선 검출 결과를 그림 8에서 확인할 수 있으며, 그림 9에서는 그림 8에서의 20장의 이미지를 대상으로 기존 방법과 제안된 방법의 처리 속도를 비교하였다. 그림 9의 x축은 SBD 데이터셋 이미지 20장을 차례대로 나열하였으며, y축은 이미지 1장당 처리 속도 시간을 의미 한다. 파란색 그래프는 허프 변환을 이용한 수평선 검출, 빨간색 그래프는 세그먼트와 이진화를 이용한 수평선 검출 처리 속도를 보여 준다. OS 환경에서 기존 방법은 이미지 당 평균 62.3ms의 수행 시간이 소요되었고, 제안하는 방법은 평균 11.2ms 정도로 약 5배 이상의 속도 차이가 발생했다. 또한, 라즈베리파이 환경에서는 기존 방법은 이미지 당 평균 627ms의 수행 시간이 소요되었고, 제안하는 방법은 평균 127ms로 수행되어 라즈베리파이 환경에서도 약 5배 이상의 처리 속도를 보였다.



OS 환경에서 수평선 검출 처리 속도



라즈베리파이 환경에서 수평선 검출 처리속도

(그림 9) 수평선 검출 처리속도

(Figure 9) Horizontal line detection processing speed

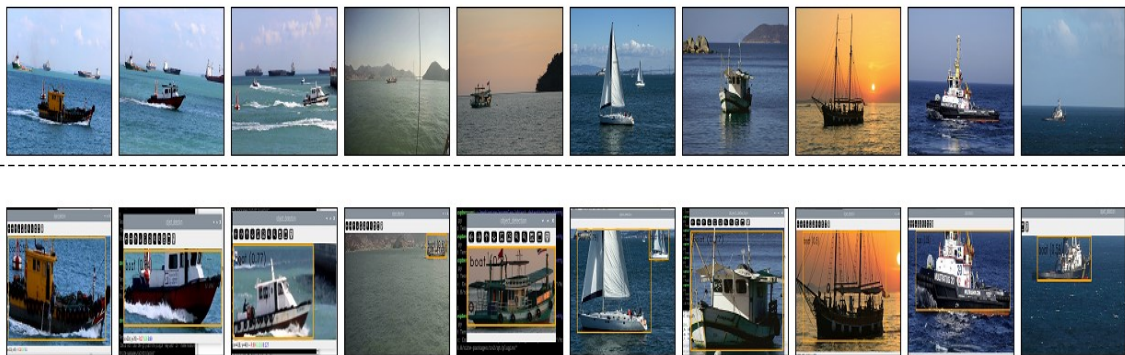
4.2 해상 객체 검출 성능

그림 10은 라즈베리파이 환경에서 본 논문의 제안 기법을 적용하여 선박 여부를 판별한 결과를 보여준다. 먼저, 제안된 전처리 과정을 통해 객체를 검출한 뒤, 이 객체들을 EfficientNet-Lite 모델을 사용하여 각 객체가 선박인지 아닌지를 판별한다. 만약 선박이 아닌 다른 객체로 판별될 경우 해당 객체는 자동으로 제외되며, 최종적으로, 선박으로 판별된 객체들만 이미지에 표시된다.

4.3 충돌 위험 선박

그림 11과 12는 CNN 모델을 통해 선박만을 검출한 후 선박으로 판별된 객체가 위험 영역 내부에 존재한다면 충돌 위험이 있는 위험 선박으로 판별되는 결과를 보여준다. 만약 위험 영역 내부에 존재하는 선박이라면 충돌 위험성이 높은 선박으로 빨간색의 바운드 박스와 함께 Danger이라는 문구로 위험 선박을 검출한다.

반면, 위험 영역 외부에 존재해 충돌 위험이 적은 선박이라면 초록색 바운드 박스로 표시한다. 그림 11의 1번 이미지는 선박으로 판별된 객체가 위험 영역 내부에 존재하여 빨간색 바운드 박스와 함께 Danger이라는 글씨를 통해 위험 선박이라는 것을 알려주는 모습이다. 반면에 2번 이미지는 위험 영역 외부에 있는 선박을 위험하지 않은 선박으로 초록색 바운드 박스로 표시하고, 위험 영역 내부에 존재하는 선박을 충돌 위험 선박으로 검출한 모습이다.



(그림 10) 라즈베리파이 환경에서 boat 검출

(Figure 10) boat detection in raspberry pie environment



위험 영역 내부와 외부 충돌 위험 선박 검출

(그림 11) 충돌 위험 선박 검출

(Figure 11) ship detection at danger of collision



위험 영역 내부에 존재하는 충돌 위험 선박

(그림 12) 위험 영역 내부에 존재하는 충돌 위험 선박

(Figure 12) ship detection at danger of collision inside ROI

5. 결 론

본 논문은 해상 환경에서 항로표지와와의 충돌 사고를 예방하기 위해 임베디드 시스템을 활용한 고속 해상 객체 검출 알고리즘을 제안한다. 현재는 레이더나 소나와 같은 고가의 장비를 사용하여 충돌 사고를 방지하고 있으나, 비용 문제로 인해 소형 선박에는 적합하지 않다. 또한, 기존의 객체 검출 방식은 주로 PC 환경에서 실행되며, 연산량과 처리 속도를 충분히 고려하지 않은 알고리즘이 사용되지만, 실제 항로표지에 설치하려면 임베디드 시스템 환경에서의 효율적 구현이 필수적이다.

임베디드 시스템에서 처리 속도를 향상시키기 위해, 본 논문은 이미지 분할과 이진화 알고리즘을 활용한 수평선 검출 방법을 제안하며, 바다의 특성을 반영해 채널 값을 이용한 바다 영역 제거를 통해 객체 검출 성능을 개선하는 방안을 제안하였다.

본 논문에서 제안하는 방안을 적용한 결과 속도 측면에서는 기존보다 약 5배 이상의 속도를 개선하였고, 성능 측면에서도 기존 방법의 성능 이상을 유지하면서 객체를 효과적으로 찾아내 충돌 위험 선박을 찾아내는 효과를 실험을 통해 증명하였다.

참고문헌(Reference)

- [1] Buch, N., Velastin, S. A., & Orwell, J., "A review of computer vision techniques for the analysis of urban traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 920-939, 2011.
<https://doi.org/10.1109/tits.2011.2119372>
- [2] Er, M. J., Zhang, Y., Chen, J., & Gao, W., "Ship detection with deep learning: a survey," *Artificial Intelligence Review*, vol. 56, no. 10, pp. 11825-11865, 2023.
<https://doi.org/10.1007/s10462-023-10455-x>
- [3] Zhang, Y., Li, Q. Z., & Zang, F. N., "Ship detection for visual maritime surveillance from non-stationary platforms," *Ocean Engineering*, vol. 141, pp. 53-63, 2017.
<https://doi.org/10.1016/j.oceaneng.2017.06.022>
- [4] Qiao, D., Liu, G., Lv, T., Li, W., & Zhang, J., "Marine vision-based situational awareness using

- discriminative deep learning: A survey,” *Journal of Marine Science and Engineering*, vol. 9, no. 4, pp. 397, 2021.
<https://doi.org/10.3390/jmse9040397>
- [5] Steccanella, L., Bloisi, D. D., Castellini, A., & Farinelli, A., “Waterline and obstacle detection in images from low-cost autonomous boats for environmental monitoring,” *Robotics and Autonomous Systems*, vol. 124, 2020.
<https://doi.org/10.1016/j.robot.2019.103346>
- [6] Bovcon, B., & Kristan, M., “A water-obstacle separation and refinement network for unmanned surface vehicles,” *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020.
<https://doi.org/10.1109/ICRA40945.2020.9197194>
- [7] Praczyk, T., “A quick algorithm for horizon line detection in marine images,” *Journal of Marine Science and Technology*, vol. 23, no. 1, pp. 164 - 177, 2017.
<https://doi.org/10.1007/s00773-017-0464-8>
- [8] Lee, J.-R., Bae, K.-R., & Moon, B., “A hardware architecture of Hough transform using an improved voting scheme,” *The Journal of Korea Information and Communications Society*, vol. 38A, no. 9, pp. 773-781, 2013.
<https://doi.org/10.7840/kics.2013.38A.9.773>
- [9] Jeong, C. Y., Yang, H. S., & Moon, K., “Fast horizon detection in maritime images using region-of-interest,” *International Journal of Distributed Sensor Networks*, vol. 14, no. 7, 2018.
<https://doi.org/10.1007/s00773-017-0464-8>
- [10] Otsu, N., “A threshold selection method from gray-level histograms,” *Automatica*, vol. 11, pp. 285-296, 1975.
<https://doi.org/10.1109/TSMC.1979.4310076>
- [11] Lee, S. W., Lee, G. D., Ko, J. G., Lee, S. J., & Yoo, W. Y., “Recent trends of object and scene recognition technologies for mobile/embedded devices,” *Electronics and Telecommunications Trends*, vol. 34, no. 6, pp. 133-144, 2019.
<https://doi.org/10.22648/ETRI.2019.J.340612>
- [12] Tan, M., “Efficientnet: Rethinking model scaling for convolutional neural networks,” *arXiv preprint arXiv:1905.11946*, 2019.

◎ 저 자 소 개 ◎



김 영 민(Young-Min Kim)

2023년 호서대학교 정보통신공학과(공학사)
2023년 호서대학교 대학원 정보통신공학과(공학석사)
관심분야 : 영상 처리, 딥러닝
E-mail : zeromin1998@gmail.com



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



임 태 호 (Tae-Ho Im)

2012년 중앙대학교 대학원 전자전기공학과(공학박사)
2012년~2015년 삼성전자 DMC연구소 책임연구원
2015년~2018년 호서대학교 해양IT융합기술연구소 조교수
2019년~2020년 호서대학교 정보통신공학과 조교수
2021년~현재 호서대학교 정보통신공학과 부교수
관심분야 : 5G 이동통신, LPWAN, 수중통신, 딥러닝
E-mail : taehoim@hoseo.edu