



Received: 2026/08/28
Revised: 2026/09/08
Accepted: 2026/09/23
Published: 2026/09/30

Corresponding Author:*Tae-Jun Kim**

1161 Gyebaek-ro, Seo-gu, Daejeon, Republic of Korea

Tel: +82-2-2088-4050

E-mail: taejun.kim@advl.ai

시뮬레이션 기반 합성 데이터셋을 활용한 선박 탐지 모델의 성능 개선

Improving Ship Detection Performance Using Simulation-Based Synthetic Datasets

김태준^{1*}, 한민성², 이화섭³¹어드바이저로렌 AI연구실 연구원²어드바이저로렌 대표이사³해군소령/해군 작전사령부 해양작전본부 전투발전참모처 무인체계작전혁신과**Tae-Jun Kim^{1*}, Min-Sung Han², Hwa-Seob Lee³**¹Research engineer, AI Research & Data Div., AdvisorLauren²CEO, AdvisorLauren³LCDR, ROK Navy/Combat Development Department, Naval Operation Command, ROK Navy**Abstract**

군사·해상 감시 분야는 보안, 운용 제약으로 실선박 데이터 확보가 어렵고, 확보된 데이터도 기상, 조도 변동성을 충분히 담지 못해 탐지 성능 저하로 이어진다. 본 연구는 시뮬레이션에서 domain randomization으로 생성한 합성 선박 이미지로 YOLOv8n과 Faster R-CNN을 사전학습해 더 나은 초기 가중치를 확보하고, 이를 선박 탐지 데이터셋에서 파인튜닝해 성능 개선 효과를 검증하였다. 두 모델 모두 mAP50:95에서 일관된 개선을 보였으며, 특히 Faster R-CNN은 고IoU 구간 위치 정확도 향상과 오탐 감소가 두드러져, 합성데이터 사전학습이 탐지 정밀도 개선에 기여함을 확인하였다.

In military and maritime surveillance applications, security and operational constraints limit the acquisition of sufficient real-world ship imagery, while existing datasets often fail to adequately represent diverse weather and lighting conditions that can degrade detection performance. In this study, YOLOv8n and Faster R-CNN are pretrained on synthetic ship images generated via domain randomization in simulation to obtain improved initial weights, followed by fine-tuning on a ship detection dataset to evaluate the resulting performance gains. Both models demonstrate consistent improvements in mAP50:95, with Faster R-CNN showing particularly notable gains in localization accuracy at high IoU thresholds and a reduction in false positives. These results indicate that synthetic-data pretraining can contribute to improved ship detection precision.

Keywords

합성 데이터(Synthetic Data),
도메인 랜덤화(Domain Randomization),
객체 탐지(Object Detection),
사전학습(Pretraining),
해상 표적 인식(Marine Target Recognition)

Acknowledgement

본 연구는 창업진흥원과 대구창조경제혁신센터의 “2026년 「민관협력 오픈이노베이션 지원」 방산 스타트업 챌린지” 지원을 받아 수행된 연구결과이다(과제번호: 20406831, 과제명: Non-GPS 대응 합성 자율복귀·자율 이착함 AI 드론 운용체계). 본 논문의 투고 및 심사에 소요되는 비용은 2026년 해군사관학교 학술연구과제(과제명: 소형표적의 효과적 대응을 위한 양자센싱 기반 탐지체계 개념연구)의 지원을 받았다.

1. 서론

해군 함정 및 무인체계(무인기, 무인수상정)에 부착된 항해용 카메라 및 EO/IR 센서 영상 기반의 선박 자동탐지 기술은 항해안 전, 기지경계, 수상 표적 탐지 및 식별에서 매우 중요한 역할을 수행한다. 이러한 선박 자동탐지기술 개발을 위해서는 딥러닝 기술 기반의 객체탐지 모델 개발이 필요하며, 이들 모델의 성능은 딥러닝에 사용되는 데이터의 양과 다양성에 크게 좌우된다. 하지만, 실제 해상에서 촬영된 선박 이미지 데이터셋을 대량으로 구축하는 데는 많은 비용과 시간이 소요된다. 특히, 군함에 탑재된 전투체계 또는 군 장비에서 작전임무 수행 중 확보한 자료는 보안상 목적으로 딥러닝 학습을 위해 사용되는 것이 제한된다. 따라서, 현재 해군에서 개발중인 선박 객체 자동탐지 기술은 공개된 선박 이미지를 기준으로 학습 중이나, 이러한 이미지는 특정 촬영시점, 해역, 기상 조건에만 편중되어 있는 경우가 대부분이다. 그렇기 때문에 이렇게 학습된 모델을 실제 작전에 적용할 경우, 탐지 성능 저하로 이어지는 문제가 발생할 수 있다. 이 문제에 대한 대안으로 최근 시뮬레이션 환경에서 생성한 합성 데이터를 활용하는 접근이 주목받고 있다. 특히 NVIDIA Isaac Sim과 같은 물리 기반 시뮬레이터는 3D

에셋과 조명, 카메라, 재질 등을 임의로 변화시키는 domain randomization을 통해 적은 비용으로 대량의 다양한 학습 이미지를 자동 생성할 수 있다는 장점이 있다.

본 연구는 이러한 domain randomization 기반 합성 선박 이미지를 생성하여 서로 다른 탐지 패러다임을 대표하는 1-stage 탐지기와 2-stage 탐지기를 사전 학습한 뒤, 이를 통해 얻은 초기 가중치가 실제 선박 탐지 벤치마크(ABOships)에서의 성능을 얼마나, 그리고 어떤 양상으로 개선하는지 정량적으로 검증하는 것을 목표로 한다.

본 연구의 기여는 다음과 같다. 첫째, NVIDIA Isaac Sim 환경에서 해상 배경(바다, 하늘)과 다종 선박 3D 에셋을 결합한 domain randomization 기반 합성 선박 이미지 데이터셋 생성 파이프라인을 구축하였다. 둘째, 서로 다른 탐지 패러다임을 대표하는 1-stage 탐지기(YOLOv8n)와 2-stage 탐지기(Faster R-CNN)를 대상으로 ABOships 벤치마크에서 baseline 성능을 측정하였다. 셋째, 합성데이터 사전학습 적용 전/후의 성능을 두 모델에서 비교하여, 사전학습이 탐지 성능에 미치는 효과와 그 효과가 탐지 아키텍처에 따라 달라지는 양상을 정량적으로 제시하였다.

2. 관련 연구

2.1 Domain Randomization과 시뮬레이션 기반 합성 데이터

Domain randomization은 시뮬레이션과 실제 환경 사이의 시각적 격차를 극복하기 위한 대표적인 sim-to-real 전략으로, 장면 구성 요소(재질, 색상, 배경, 조명 등)를 충분히 다양하게 무작위화하면 학습된 모델이 실제 데이터를 단지 또 다른 변수로 인식하게 된다는 가설에 기반한다[1]. 이 개념은 로봇틱스 분야에서 시작되어 이후 다양한 산업 분야의 객체 탐지 문제로 확장되었다[2,3]. 제조업 분야를 대상으로 한 최근 연구는 domain randomization의 세부 요소(재질 속성, 렌더링 방식, 후처리, distractor 객체 등)가 합성-실제 데이터 간 성능 전이에 미치는 영향을 체계적으로 분석하였으며[2], 순수 합성데이터와 domain randomization만으로 학습한 YOLOv11 모델이 한 객체 탐지 경진대회에서 mAP₅₀ 0.910의 높

은 정확도를 달성한 사례도 보고되었다[4].

NVIDIA Isaac Sim[5,6]은 omniverse replicator[5,6]를 통해 물리 기반 렌더링과 프로그래머블 domain randomization, 자동 어노테이션(바운딩박스, 세그멘테이션 마스크 등)을 결합한 합성데이터 생성 파이프라인을 제공한다. Isaac Sim 기반 합성데이터를 이용한 sim-to-real 전이는 과수 인식[7], 수중 로봇 인식[8] 등 다양한 응용에서 검증되고 있으나, 해상·함정 인식 분야에서 Isaac Sim 기반 합성데이터의 기여도를 정량적으로 검증한 연구는 아직 드물다.

2.2 선박 탐지 연구 및 벤치마크

선박 탐지·분류에 딥러닝을 적용한 연구들이 다수 발표되었다. Han과 Hwang은 가변 해상 환경에서의 실시간 객체 탐지를 위해 YOLOv11 기반 동적 가지치기 및 경량화 기법을 제안하였고[9], Bak은 세분 선박 원격탐사 이미지 분류를 위해 엣지 정보를 강화한 CNN 학습 프레임워크를 제안하였다[10]. Im 등은 SAR 영상에서 편파 조합에 따른 YOLO 기반 선박 탐지 모델의 정확도 특성을 분석하였고[11], Kim과 Moon은 수정된 YOLOv4 기반의 해상 객체 탐지·분류 모형을 제안하였다[12]. 국외에서는 Li 등이 복잡한 해상 조건에서의 선박 탐지를 위해 동적 컨볼루션과 적응적 융합 네트워크를 결합한 모델을 제안하였고[13], Gong 등은 원거리 소형 선박 탐지를 위해 슬라이싱 기법을 결합한 2단계 탐지 방법을 제안하였다[14]. 이들 연구는 실제 선박 탐지·분류 성능 개선에 초점을 맞추었으나, 학습 데이터 자체를 시뮬레이션 기반 합성데이터로 확충하는 접근은 다루지 않았다는 점에서 본 연구와 차별화된다.

해상 선박 탐지를 위한 공개 데이터셋은 촬영 시점에 따라 크게 일반 각도(지상·해상 카메라 촬영)와 항공뷰(위성/원격탐사)로 구분된다. 일반 각도 계열로는 본 연구에서 사용한 ABOships[15] 외에도 WUTDet[16], SeaShips[17], MARVEL[18], 그리고 국내 21개 해역에서 촬영된 KOLOMVERSE[19] 등이 있으며, 항공뷰 계열로는 HRSC2016[20], ShipRSImageNet[21] 등이 있다. 본 연구에서 사용한 ABOships는 핀란드 군도 해역의 inshore-offshore 환경을 모두 포괄하며, 배경·기상·조도·가려짐·스케일 변화가 다양하게 반영되도록 촬영

된 정밀 어노테이션 벤치마크이다.

3. 제안방법: 합성 선박 데이터셋 생성

3.1 개요 및 씬 구성

본 연구의 합성데이터 생성 파이프라인은 NVIDIA Isaac Sim에서 해상 배경(하늘+바다)에 다중 선박 3D 에셋을 배치하고, 카메라·선박 배치·환경 조건을 randomization하여 다양한 학습 이미지와 바운딩박스 라벨을 자동 생성하는 구조로 설계되었다. 합성 이미지는 실제 카메라 대비 지나치게 선명하므로, 화질 저하 후처리(3.3절)를 추가로 거쳐 사전학습에 사용하였다.

바다는 Isaac Sim에 내장된 warp 기반 해양 시뮬레이션(TMA 스펙트럼 기반 파형 합성)으로 구성하였으며, 하늘은 태양 방위각, 고도, 하늘색, 구름, 안개를 파라미터로 조절하는 절차적 셰이더로 구성하였다. 선박 3D 모델은 Sketchfab[22]에서 creative commons attribution(CC-BY) 라이선스로 배포된 모델만 필터링하여 수집하였다. 화물선, 크루즈선, 페리, 군함, 여객선, 범선, 소형 보트 등 7개 세부 유형에 걸쳐 총 28종을 확보하여 사용하였다.

3.2 Domain Randomization 설계 및 데이터 생성

(1) 카메라 거리·고도각: 절대 거리가 아닌 “카메라-선박 거리/선박 전장” 비율로 샘플링하고, 목표 화면 내 세로 위치(bbox 중심의 y좌표)를 ABOships 실측 분포에 맞춰 먼저 정한 뒤 이를 만족하는 카메라 고도각을 역산하는 방식으로, bbox 크기·화면 내 위치 분포가 실제 데이터셋에 통계적으로 근접하도록 캘리브레이션하였다.

(2) 프레임당 선박 인스턴스 수 및 배치: ABOships의 실측 “프레임당 인스턴스 수” 분포에 로그정규분포를 적합하여 매 이미지의 선박 수를 샘플링하고, 카메라 시야각 내를 인스턴스 수만큼 균등 분할한 슬롯에 배치하였다.

(3) 선박 선수 방향: 매 이미지마다 독립적으로 무작위 배정하되, 실제 촬영 이미지에서 나타나는 종횡비 분포에 맞춰 선수/선미 방향으로 편향되도록 조정하였다.

(4) 환경 조건: 태양 방위각·고도, 하늘색조, 구름, 안개, 해수색 5개 축을 독립적으로 randomization하였다.

(5) 배경: 실사진 기반 HDRI(high dynamic range image) 7종(자연경관, 도심 등) 중 무작위로 선택하여 돛 라이트로 적용하였다.

(6) 가려짐: 계산적 투영 대신 instance segmentation 마스크에서 바운딩박스를 산출하여, 파도, 다른 선박에 의한 가려짐이 라벨에 자동으로 반영되도록 하였다.

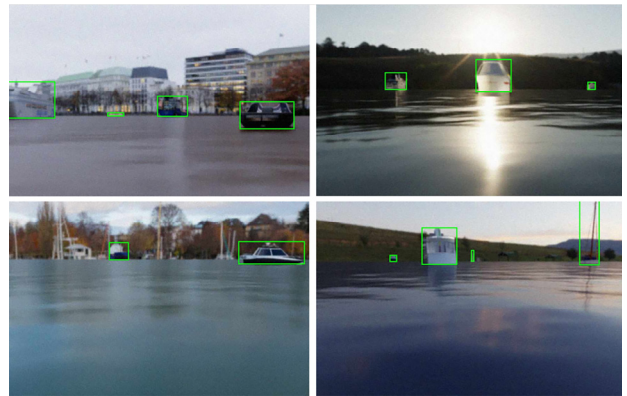


Fig. 1. Examples of generated synthetic ship images under different domain randomization conditions (background, weather, HDRI)

이 설계에 따라 렌더링된 RGB 이미지와 라벨을 저장하여 총 4,894장의 합성 이미지-라벨 쌍을 생성하였다. Fig. 1은 이러한 domain randomization 설계에 따라 생성된 합성 이미지 예시를 보여준다.

3.3 화질 저하 후처리

Isaac Sim으로 렌더링한 합성 이미지는 실제 카메라로 촬영한 이미지에 비해 노이즈·블러·압축 열화가 없어 지나치게 선명하다. 이 sim-to-real 시각적 격차를 줄이기 위해 생성된 전체 합성 이미지에 다운스케일 후 재확대(배율: 0.55)에 의한 디테일 손실, 가우시안 블러(반경: 1.0), 가산 가우시안 노이즈(표준편차: 10), JPEG 재압축(품질: 45)을 순서대로 적용하였다. 이 화질 저하 정도는 강도가 다른 5개 후보를 육안으로 비교하여 선정하였다. Fig. 2는 씬 구성부터 화질 저하까지 전체 생성 파이프라인의 단계별 중간 산출물을 보여준다.

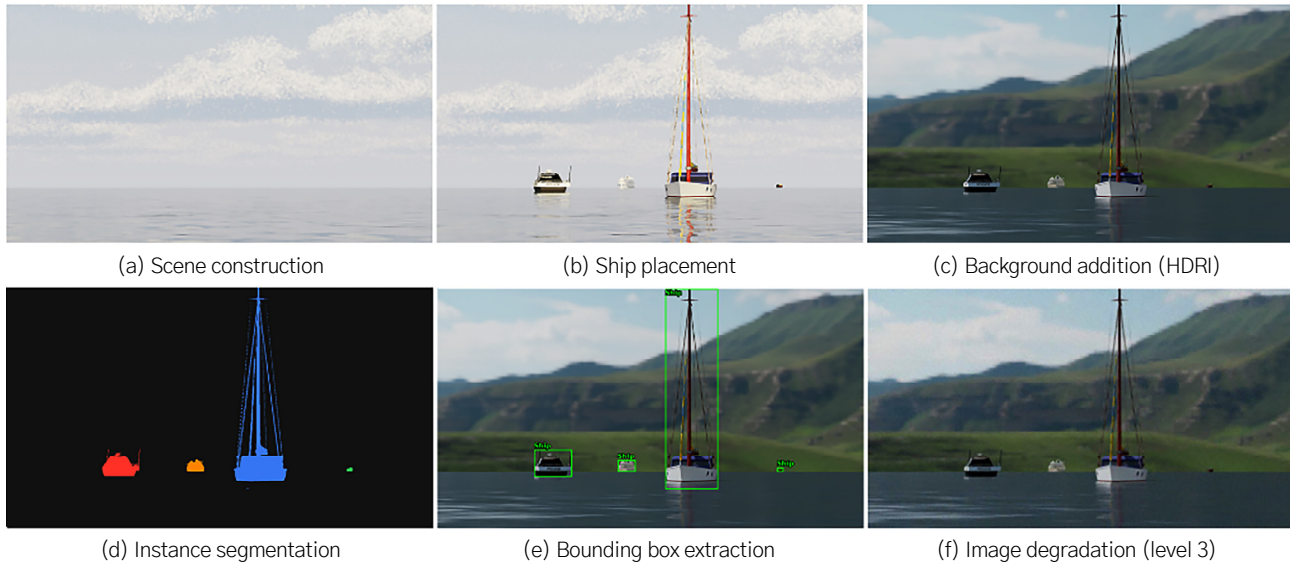


Fig. 2. Synthetic ship image generation pipeline

3.4 사전학습과 파인튜닝 적용 방식

각 baseline 모델(4장)에 대해, 3.2~3.3절에서 생성한 화질 저하 합성데이터로 먼저 사전학습을 수행한 뒤, baseline과 동일한 학습 설정(optimizer, learning rate)으로 실데이터에 파인튜닝한 모델을 별도로 학습하였다. Baseline(실데이터만 학습)과 동일한 실데이터 test set에서 두 모델의 성능을 비교하여, 합성데이터 사전학습이 실제 환경 성능에 기여하는 순효과를 측정하였다.

4. 실험 설계

4.1 데이터셋 및 전처리

본 연구는 공개 선박 탐지 벤치마크인 ABOships를 사용하였다. ABOships는 핀란드 군도 해역에서 이동 수상버스에 탑재된 카메라로 촬영된 9,880장/41,967 어노테이션 규모의 정밀 어노테이션 벤치마크로, 11개 세부 선박·항로표지 클래스를 포함한다 [15]. Fig. 3는 ABOships의 실촬영 이미지 예시를 정답(ground-truth) 바운딩박스과 함께 보여준다.

초기 검증 단계에서의 실험 복잡도를 낮추기 위해 탐지 태스크를 단일 클래스("ship" 존재 여부 탐지)로 단순화하였다. 11개 클래스 중 실제 선박이 아닌 항로표지 클래스(seamark)와 내용이 불명확한 기타 클래스(miscellaneous)를 제외하고, 나머지 9개 선

박 클래스(boat, cargoship, cruiseship, ferry, militaryship, miscboat, motorboat, passengership, sailboat)를 단일 클래스로 병합하였다.

ABOships는 원본 프레임이 연속 촬영된 영상에서 추출되었다는 특성이 있어, 무작위 프레임 단위 분할 시 인접 프레임이 서로 다른 split(train/val/test)에 나뉘어 배정되는 data leakage 위험이 있다. 이를 방



Fig. 3. ABOships real-image examples with ground-truth ship bounding boxes (green)

지하기 위해 촬영 날짜(13개 폴더)를 분할 단위로 사용하여, 날짜 폴더 전체를 train/val/test 중 하나에 통째로 배정하였다. 다만 가장 희귀한 클래스(cargo-ship, 13개 폴더 중 3개에는 부재)가 val-test에도 반드시 포함되도록, 목표 비율(80/10/10%)에 가장 근접하면서도 cargoship이 포함되는 폴더 조합을 탐색하여 확정하였다. 최종 분할 결과는 train 7,952장, val 956장, test 972장이다.

Table 1. YOLOv8n: AP, Recall, FP/TP comparison by IoU threshold (baseline vs. baseline + synthetic data)

IoU	Baseline			Ours		
	AP	Recall	FP/TP	AP	Recall	FP/TP
0.50	0.722	0.751	0.563	0.725	0.756	0.559
0.55	0.687	0.725	0.618	0.683	0.727	0.621
0.60	0.623	0.680	0.725	0.624	0.680	0.733
0.65	0.544	0.618	0.897	0.549	0.625	0.885
0.70	0.455	0.551	1.128	0.457	0.555	1.123
0.75	0.350	0.460	1.552	0.351	0.465	1.531
0.80	0.239	0.358	2.281	0.230	0.356	2.307
0.85	0.120	0.228	4.150	0.124	0.242	3.865
0.90	0.041	0.112	9.477	0.046	0.123	8.591
mAP_{50}		0.726			0.735	
$mAP_{50:95}$		0.382			0.384	

Table 2. Faster R-CNN: AP, Recall, FP/TP comparison by IoU threshold (baseline vs. baseline + synthetic data)

IoU	Baseline			Ours		
	AP	Recall	FP/TP	AP	Recall	FP/TP
0.50	0.743	0.865	1.509	0.744	0.856	1.308
0.55	0.696	0.826	1.628	0.699	0.821	1.407
0.60	0.624	0.771	1.815	0.628	0.765	1.584
0.65	0.526	0.686	2.163	0.532	0.684	1.888
0.70	0.409	0.585	2.710	0.428	0.593	2.335
0.75	0.300	0.481	3.514	0.299	0.477	3.146
0.80	0.191	0.350	5.206	0.188	0.347	4.701
0.85	0.083	0.211	9.306	0.090	0.211	8.351
0.90	0.018	0.082	25.377	0.029	0.093	20.305
mAP_{50}		0.739			0.739	
$mAP_{50:95}$		0.357			0.362	

4.2 Baseline 모델 및 학습 설정

특정 탐지 아키텍처에 국한되지 않는 개선 효과를

검증하기 위해, 서로 다른 탐지 패러다임을 대표하는 두 모델을 baseline으로 선정하였다. YOLOv8n[23]은 1-stage anchor-free CNN 계열로, 실시간 해상 객체 탐지 분야의 표준적 접근이며 다수의 선행 연구에서도 채택되었다[9]. Faster R-CNN[24]은 2-stage anchor 기반 CNN 계열로, 원격탐사·선박 탐지 문헌에서 표준적으로 사용되는 고전적이면서도 강력한 baseline이다.

두 모델 모두 COCO로 사전학습된 가중치에서 시작한다. YOLOv8n은 Ultralytics 프레임워크를 사용하여, 입력 해상도 1280×1280, 조기종료 patience=20 (검증 mAP50-95 기준 20 epoch 연속 개선이 없으면 종료)으로 학습하였다. Faster R-CNN은 SGD 옵티마이저(learning rate=0.005, momentum=0.9)로 고정 30 epoch 동안 학습하였다.

4.3 평가지표 및 비교 설계

탐지 성능은 IoU 0.5 기준 평균 정밀도(mAP50)와 IoU 0.5~0.95 구간 평균의 mAP(mAP50-95)를 기본 지표로 사용하였다. 이와 함께 IoU 임계값별 세부 거동을 분석하기 위해 IoU 0.50~0.90(0.05 간격) 구간별 AP, recall, FP/TP(오탐 대비 정탐 비율)를 추가로 산출하였다(세부 산출 방식은 5.1절 참고). 모든 지표는 홀드아웃 test set에서 측정하였다.

각 모델에 대해 다음 두 조건을 비교한다. Baseline: 실제 데이터(train)만으로 파인튜닝한 모델. Baseline + 합성데이터: 실제 데이터에 합성데이터를 추가하여 파인튜닝한 모델. 두 조건 모두 동일한 실제 데이터의 test set에서 평가하여, 합성데이터 추가로 인한 순수한 성능 변화를 측정한다. 결과는 5장에서 제시한다.

5. 실험 결과

두 모델 모두 동일한 ABOships 홀드아웃 test set(972장, ship 인스턴스 2,867개)에서 평가하였다. mAP50과 mAP50:95는 각 모델의 학습 프레임워크(YOLOv8n은 Ultralytics, Faster R-CNN은 torchmetrics ‘MeanAveragePrecision’)가 제공하는 표준 평가 파이프라인의 결과를 그대로 사용하였다. 이와 별도로, IoU 0.50~0.90 구간별 세부 거동을 분석하기 위해 각 IoU 임계값에서의 AP와, con-

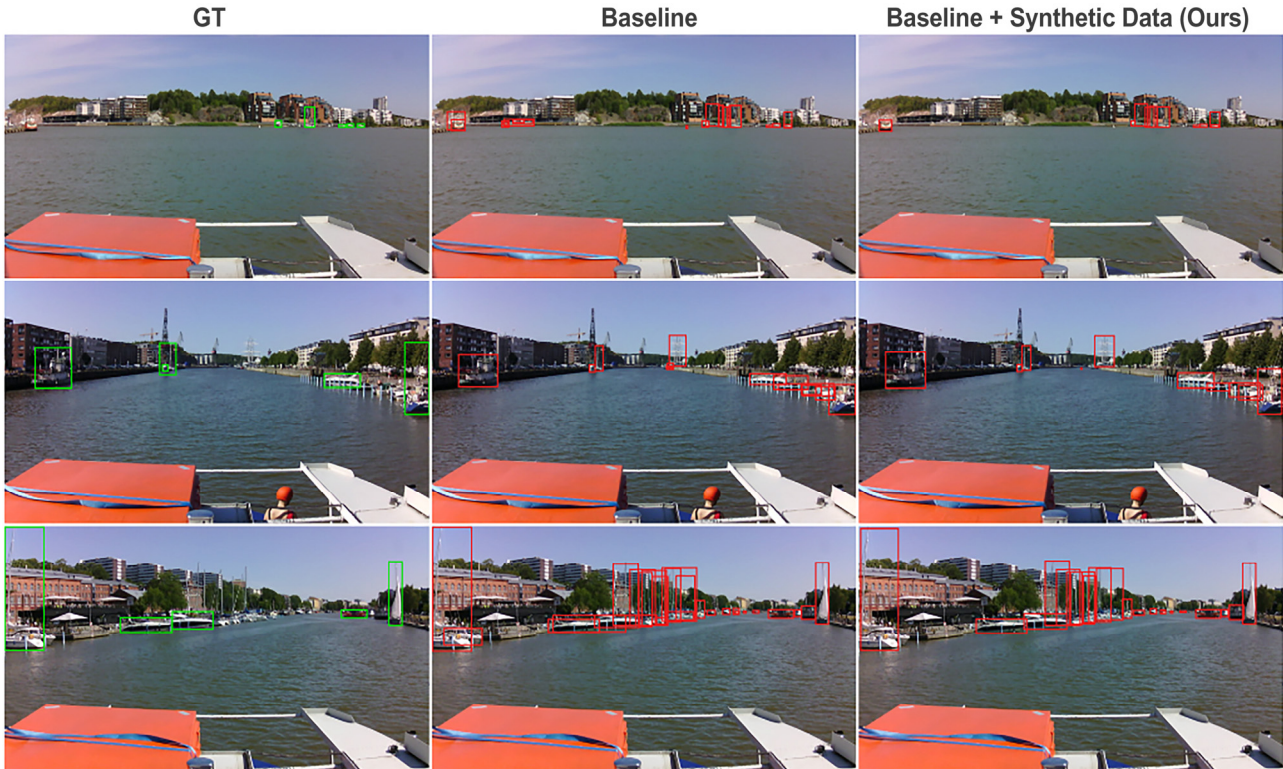


Fig. 4. Qualitative comparison between baseline and baseline + synthetic data (Faster R-CNN) on three test images (rows), each showing ground truth (left), Baseline predictions (center), and ours predictions (right) at confidence ≥ 0.25 (green = ground truth, red = prediction)

fidence 0.25로 고정한 operating point에서의 recall 및 FP/TP를 별도로 산출하였다. FP/TP 비율은 값이 작을수록 동일한 정탐 수 대비 오탐이 적음을 의미한다.

Table 1에서 보듯이, YOLOv8n은 합성데이터 파인튜닝 후 mAP50이 0.726에서 0.735로, mAP50:95가 0.382에서 0.384로 상승하였다. IoU별 추이를 보면 IoU 0.50~0.60 구간에서는 두 모델이 거의 동일이며, IoU 0.65 이상 고정밀 구간에서는 IoU 0.80을 제외한 전 구간에서 일관된 성능 개선을 보였다.

Table 2에서 보듯이, Faster R-CNN은 mAP50 자체는 0.739→0.739로 사실상 변화가 없었으나, mAP50:95는 0.357에서 0.361로 YOLOv8n보다 뚜렷한 개선을 보였다. 이는 IoU 임계값이 높아질수록 격차가 커지는 IoU별 추이로 설명된다: IoU 0.70에서 AP가 0.409에서 0.428로, IoU 0.90에서는 0.018에서 0.029로 고정밀 구간일수록 개선폭이 크게 확대된다. 또한 FP/TP 비율이 모든 IoU 구간에서 예외 없이 감소하였다. 반면 recall은 저IoU 구간(0.50~0.65)에서 baseline이 오히려 근소하게 높고, 고IoU 구간에

서는 합성데이터 적용 모델이 소폭 앞선다. Fig. 4는 Faster R-CNN의 오탐 감소 양상을 보여주는 정성적 비교 예시 3건을 제시한다.

종합하면, YOLOv8n과 Faster R-CNN 모두 합성데이터 파인튜닝을 통해 일관된 성능 개선을 보였으며, 특히 Faster R-CNN에서 더 큰 폭의 성능 개선이 나타났다. Faster R-CNN에서는 합성데이터 파인튜닝이 탐지 개수 자체(recall)보다는 바운딩박스 위치 정확도와 오탐 억제 측면에서 두드러진 효과를 보였다. 저IoU 구간에서의 소폭의 recall 손실을 고IoU 구간의 정밀도 향상과 전 구간에 걸친 FP 감소가 상쇄하고도 남아, mAP50:95 기준으로 가장 뚜렷한 순개선으로 이어진 것으로 해석된다.

6. 결론

본 연구는 시뮬레이션 domain randomization으로 생성한 합성 선박 이미지가 실제 선박 탐지 성능 개선에 기여하는지, 그 효과가 탐지 아키텍처에 따라 달라지는지를 ABOships 벤치마크에서 검증하였다.

YOLOv8n과 Faster R-CNN 모두 합성데이터 사전학습 후 mAP50:95 기준 일관된 개선을 보였으나 양상은 아키텍처마다 달랐다: YOLOv8n은 전 IoU 구간에서 고른 소폭 개선을 보인 반면, Faster R-CNN은 IoU가 높아질수록 개선폭이 급격히 커지고 전 구간에서 오탐 비율이 일관되게 감소하는 등 위치 정확도와 오탐 억제 측면에서 더 구조적인 개선을 보였다. 이는 2-stage 구조가 합성데이터의 다양한 스케일·시점·조명 변화를 더 효과적으로 반영했을 가능성을 시사하며, 합성데이터 활용 전략을 아키텍처 특성에 맞추어 설계할 필요가 있음을 보여준다.

다만 본 연구는 검증 아키텍처, 데이터셋 범위가 제한적이다. 향후 연구에서는 더 다양한 아키텍처, 벤치마크로 검증을 확장하고, domain randomization 요소별 ablation과 합성/실데이터 혼합 비율 탐색을 통해 시뮬레이션 기반 합성데이터를 합정, 해상 표적 인식 체계에 적용하기 위한 구체적 전략을 제시하고자 한다.

참고문헌

- [1] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, & Pieter Abbeel, 'Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World,' in proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, Canada, 2017, pp. 23–30.
- [2] Xiaomeng Zhu, Jacob Henningson, Duruo Li, Pär Mårtensson, Lars Hanson, Mårten Björkman, & Atsuto Maki, 'Domain Randomization for Object Detection in Manufacturing Applications using Synthetic Data: A Comprehensive Study,' arXiv:2506.07539, 2025.
- [3] Jie Fan, Yongqiang Xie, & Zhongbo Li, 'Domain Randomization for Object Detection: A Survey,' in proceedings of the 8th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Nanjing, China, 2025, pp. 201–214.
- [4] Luisa Torquato Niño & Hamza A. A. Gardi, 'Synthetic-to-Real Object Detection Using YOLOv11 and Domain Randomization Strategies,' arXiv:2509.15045, 2025.
- [5] NVIDIA, Isaac Sim – Robotics Simulation and Synthetic Data Generation, NVIDIA Developer. <https://developer.nvidia.com/isaac/sim> (Accessed 2026.08.25.)
- [6] Sicong Gao, Maurice Pagnucco, Tomasz Bednarsz, & Yang Song, 'NVIDIA Isaac Sim: Enabling Scalable, GPU-Accelerated Simulation for Robotics,' arXiv:2606.03551, 2026.
- [7] Martina Hutter-Mironovova, 'Sim-to-Real Fruit Detection Using Synthetic Data: Quantitative Evaluation and Embedded Deployment with Isaac Sim,' arXiv:2603.28670, 2026.
- [8] Jingyu Song, Haoyu Ma, Onur Bagoren, Advait V. Sethuraman, Yiting Zhang, & Katherine A. Skinner, 'OceanSim: A GPU-Accelerated Underwater Robot Perception Simulation Framework,' arXiv:2503.01074, 2025.
- [9] Min-Seok Han & Jaeryong Hwang, 'Dynamic Pruning and Integrated Lightweight Optimization of YOLOv11 for Real-Time Object Detection in Variable Maritime Environments,' Journal of the KNST, VOL. 8, NO. 4, 2025, pp. 916–921.
- [10] Changju Bak, 'Edge-Enhanced CNN Training for Fine-Grained Ship Classification in Remote Sensing,' Journal of the KNST, VOL. 8, NO. 4, 2025, pp. 710–719.
- [11] Yun-Gyo Im, Youjeong Youn, Jonggu Kang, Seoyeon Kim, Ye-min Jeong, Soyeon Choi, Youngmin Seo, & Yang-Won Lee, 'Ship Detection from SAR Images Using YOLO: Model Constructions and Accuracy Characteristics According to Polarization,' Korean Journal of Remote Sensing, VOL. 39, NO. 5, 2023, pp. 997–1008.
- [12] Junghwan Kim & Hoseok Moon, 'A Model Development for Object Detection and Classification at Sea Using Modified YOLO v4,' Journal of the Military Operations Research Society of Korea, VOL. 47, NO. 2, 2021, pp. 13–25.
- [13] Zhisheng Li, Zhihui Deng, Kun Hao, Xiaofang Zhao & Zhigang Jin, 'A Ship Detection Model Based on Dynamic Convolution and an Adaptive Fusion Network for Complex Maritime Conditions,' Sensors, VOL. 24, NO. 3, 2024, Article 859.
- [14] Yanfeng Gong, Zihao Chen, Jiawan Tan, Chaozhong Yin, & Wen Deng, 'Two-Stage Ship Detection at Long Distances Based on Deep Learning and Slicing Technique,' PLoS ONE, VOL. 19, NO. 11, 2024, Article e0313145.
- [15] Bogdan Iancu, Valentin Soloviev, Luca Zelioli & Johan Lilius, 'ABOShips — An Inshore and Offshore Maritime Vessel Detection Dataset with Precise Annotations,' Remote Sensing, VOL. 13, NO. 5, 2021, Article 988.
- [16] Junxiong Liang, Mengwei Bao, Tianxiang Wang, Xinggang Wang, An-An Liu, & Ryan Wen Liu, 'WUTDet: A 100K-Scale Ship Detection Dataset and Benchmarks with Dense Small Objects,' arXiv:2604.07759, 2026.
- [17] Zhenfeng Shao, Wenjing Wu, Zhongyuan Wang, Wan Du, & Chengyuan Li, 'SeaShips: A Large-Scale Precisely Annotated Dataset for Ship Detection,' IEEE Transactions on Multimedia, VOL. 20, NO. 10, 2018, pp. 2593–2604.
- [18] Erhan Gundogdu, Berkan Solmaz, Veysel Yücesoy, & Aykut Koç, 'MARVEL: A Large-Scale Image Dataset for Maritime Vessels,' in proceedings of the 13th Asian Conference on Computer Vision (ACCV), Taipei, Taiwan, 2016, pp. 165–180.
- [19] Abhilasha Nanda, Sung Won Cho, Hyeopwoo Lee, & Jin

Hyoung Park, 'KOLOMVERSE: Korea Open Large-Scale Image Dataset for Object Detection in the Maritime Universe,' IEEE Transactions on Intelligent Transportation Systems, VOL. 25, NO. 12, 2024, pp. 20832–20840.

[20] Zikun Liu, Liu Yuan, Lubin Weng, & Yiping Yang, 'A High Resolution Optical Satellite Image Dataset for Ship Recognition and Some New Baselines,' in proceedings of the 6th International Conference Pattern Recognition Applications and Methods (ICPRAM), Porto, Portugal, 2017, pp. 324–331.

[21] Zhengning Zhang, Lin Zhang, Yue Wang, Pengming Feng, & Ran He, 'ShipRSImageNet: A Large-Scale Fine-Grained Dataset for Ship Detection in High-Resolution Optical Remote Sensing Images,' IEEE Journal of Selected

Topics in Applied Earth Observations and Remote Sensing, VOL. 14, 2021, pp. 8458–8472.

[22] Sketchfab, The Leading Platform for 3D & AR on the Web. <https://sketchfab.com> (accessed 2026.08.25.)

[23] Glenn Jocher, Ayush Chaurasia, & Jing Qiu, 'Ultralytics YOLOv8,' version 8.0.0, 2023.

<https://github.com/ultralytics/ultralytics> (accessed 2026.08.25.)

[24] Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun, 'Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,' Advances in Neural Information Processing Systems (NIPS), 2015, pp. 91–99.