

Shipping Container Load State and Accident Risk Detection Techniques Based Deep Learning

Yeon Jeong Hum[†] · Seo Yong Uk^{††} · Kim Sang Woo^{†††} · Oh Se Yeong^{††††} · Jeong Jun Ho^{†††††} · Park Jin Hyo^{†††††} · Kim Sung-Hee^{††††††} · Youn Joosang^{†††††††}

ABSTRACT

Incorrectly loaded containers can easily knock down by strong winds. Container collapse accidents can lead to material damage and paralysis of the port system. In this paper, We propose a deep learning-based container loading state and accident risk detection technique. Using Darknet-based YOLO, the container load status identifies in real-time through corner casting on the top and bottom of the container, and the risk of accidents notifies the manager. We present criteria for classifying container alignment states and select efficient learning algorithms based on inference speed, classification accuracy, detection accuracy, and FPS in real embedded devices in the same environment. The study found that YOLOv4 had a weaker inference speed and performance of FPS than YOLOv3, but showed strong performance in classification accuracy and detection accuracy.

Keywords : Shipping Container, YOLOv4, Deep Learning, Object Detection

딥러닝 기반 컨테이너 적재 정렬 상태 및 사고 위험도 검출 기법

연 정 흥[†] · 서 용 욱^{††} · 김 상 우^{†††} · 오 세 영^{††††} · 정 준 호^{†††††} · 박 진 효^{†††††} · 김 성 희^{††††††} · 윤 주 상^{†††††††}

요 약

최근 항만에서는 부정확한 컨테이너 적재로 인해 컨테이너가 강풍에 쉽게 쓰러지는 컨테이너 붕괴 사고가 빈번히 발생하고 있으며 이는 물적 피해와 항만 시스템 마비로 이어지고 있다. 본 논문에서는 이런 사고를 미연에 방지하기 위해 딥러닝 기반 컨테이너 적재 상태 및 사고 위험도 검출 시스템을 제안한다. 제안된 시스템은 darknet 기반 YOLO 모델을 활용하여 컨테이너 상하의 코너캐스팅을 통해 컨테이너 정렬 상태를 실시간으로 파악하고 관리자에게 사고 위험도를 알리는 시스템이다. 제안된 시스템은 추론 속도, 분류 정확도, 검출 정확도 등을 성능 지표와 실제 구현 환경에서 최적의 성능을 보인 YOLOv4 모델을 객체 인식 알고리즘 모델로 선택하였다. 제안된 알고리즘인 YOLOv4가 YOLOv3보다 추론속도와 FPS의 성능 측면에서 낮은 성능을 보이기는 했지만, 분류 정확도와 검출 정확도에서 강력한 성능을 보임을 증명하였다.

키워드 : 컨테이너, YOLOv4, 딥러닝, 객체인식

1. 서 론

현재 노르웨이, 독일과 같은 여러 나라에서 스마트 항만을 추진하기 위한 활발한 연구가 진행된다. 스마트 항만이란 공

유 네트워크를 기반으로 의사결정이 이루어지고 실행하는 항만이다[1]. 스마트 항만은 물류, 에너지, 인프라의 효율적인 활용을 중요하게 평가한다. 이 평가항목에는 구체적인 계획이 있고 많은 연구도 진행 중이다[2-5]. 하지만 컨테이너 적재 안전성과 관련된 연구는 많이 부족하다. 항만은 강풍이 많이 부는 지역적 특성 때문에 부정확한 컨테이너 적재로 인한 컨테이너 붕괴 사고의 확률이 높아진다는 문제점이 있다. 컨테이너 붕괴 사고는 물적 피해와 붕괴한 컨테이너로 인한 항만 내 차량흐름에 방해가 된다. 특히 부산항은 대한민국의 전체 컨테이너 화물 처리량 중 75.5%(2021년 기준, 2,270만 TEU (Twenty-foot Equivalent Unit)[6])로 많은 컨테이너가 있어 국가적인 항만 시스템 마비로 이어진다.

본 논문은 컨테이너 위아래의 코너캐스팅 오차에 따른 컨테이너 적재 상태 분류 기준을 정의하고, Darknet 기반의

※ 이 논문은 연구개발특구진흥재단의 지역연구개발혁신지원 지능형 무인 자동화 스마트물류 시스템 구축 과제(2020-DD-UP-0281-03-210)와 2022년도 BB21+ 사업으로 지원되었음.

† 비 회 원 : 부산항만공사 항만 R&D실 실장
†† 비 회 원 : ㈜서안에스앤씨 대표
††† 비 회 원 : ㈜서안에스앤씨 부장
†††† 비 회 원 : 동의대학교 IT융합학과 석사과정
††††† 준 회 원 : 동의대학교 IT융합학과 석사과정
†††††† 정 회 원 : 동의대학교 산업융합시스템공학부 조교수
††††††† 종신회원 : 동의대학교 산업ICT기술공학과 교수

Manuscript Received : September 2, 2022

Accepted : September 26, 2022

* Corresponding Author : Youn Joosang(jsyou@deu.ac.kr)

YOLO(You Only Look Once)를 활용하여 컨테이너 위험도를 실시간으로 파악하기 위한 컨테이너 사고 위험도 검출 기법을 제안한다[7]. 연구를 진행하기 위해 YOLOv3와 YOLOv4를 사용하였으며 같은 환경에서 추론 속도, 분류 정확도, 검출 정확도, 실제 임베디드 장치에서의 FPS(Frame Per Second) 등을 기준으로 효율적인 학습 알고리즘을 선정한다.

2. 관련 연구

2.1 딥러닝 기반 객체 탐지 알고리즘

객체 검출 작업은 찾고자 하는 객체의 특징을 사전에 학습하여 주어진 영상 내에서 해당 특징을 검출해 주어진 카테고리 객체를 식별과 위치를 찾는 작업이다. 전통적인 객체 검출 작업은 한정된 애플리케이션, 작은 카테고리 수, 검출 결과 및 일반화 능력이 부족하다는 단점이 있다. 하지만 딥러닝의 활발한 연구로 객체 검출 작업은 다양한 애플리케이션과 높은 성능을 가지게 되었다.

딥러닝 기반 객체 검출알고리즘은 YOLO와 같은 single-stage method와 Faster R-CNN(Region-based Convolutional Neural Network)과 지역 추천 기반의 two-stage method로 구성되어 있다[8]. YOLO는 이미지 전체에서 객체의 카테고리 위치를 예측하는 객체 검출알고리즘에 새로운 접근법을 제공했다. 입력 이미지를 $S \times S$ 그리드로 나누어 각 그리드의 신뢰도를 계산해 객체를 찾는 알고리즘이다[9]. Faster R-CNN은 기존 Fast R-CNN의 구조를 계승하면서 selective search를 제거하고 RPN(Region Proposal Network)와 ROI(Region of Interest) Pooling을 적용해 카테고리 분류와 객체를 찾는 알고리즘이다[9-11]. Fig. 1은 Faster R-CNN의 아키텍처를 보여준다.

YOLO 알고리즘은 Faster R-CNN보다 높은 연산속도를 가지고 있어 실시간으로 객체를 검출하는데 적합한 알고리즘이다[12]. 본 논문의 코너캐스팅과 같이 작은 객체를 검출하기 어렵다는 단점이 있지만, 버전을 거듭할수록 성능을 개선했다[13].

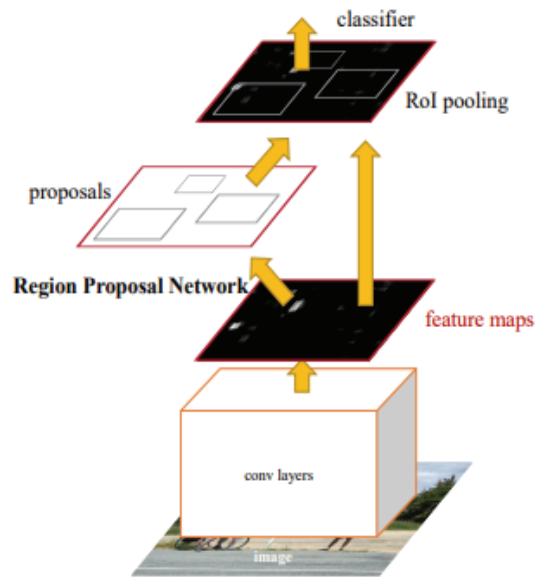


Fig. 1. Faster R-CNN Architecture[10]

2.2 YOLO

1) YOLOv1

YOLOv1[14]은 GoogLeNet을 기반으로 24개의 Convolution layer와 2개의 Fully Connected layer로 구성됐다. 이미지를 여러 개의 그리드로 나누어 각각 그리드에서 객체의 카테고리를 예측한다는 의미가 있는 YOLO의 초기 알고리즘이다. 이미지가 $448 \times 448 \times 3$ 으로 고정된 한계가 있다. 또한, 작은 객체를 감지하지 못하는 단점 때문에 근처에 있는 객체를 각각 감지하지 못하는 단점이 있다. Fig. 2는 YOLOv1에 대한 구조를 설명하고 있다.

2) YOLOv2

YOLOv2[15]는 Fully Connected layer를 제거하고 anchor box 개념을 도입해 박스에 대한 사전 정보를 기반으로 박스의 크기나 위치를 정확하게 예측한다. Multi-Scale Training 개념을 추가하여 기존의 모델에서 작은 객체에 대

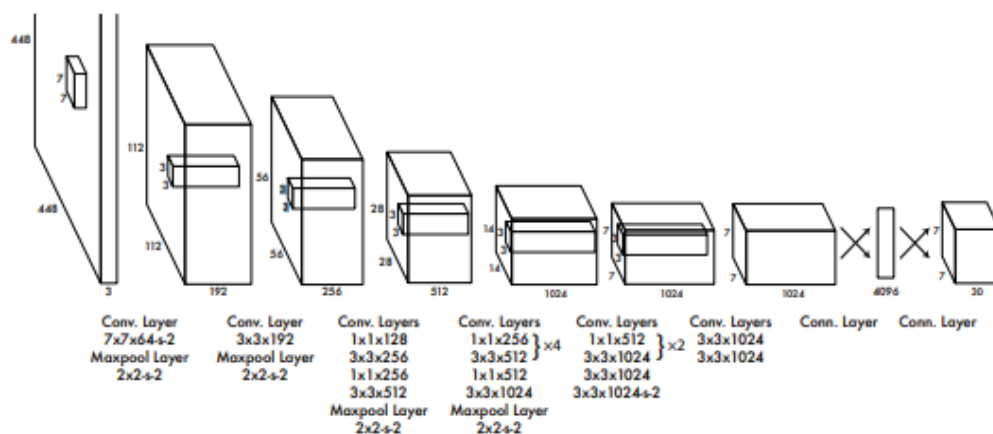


Fig. 2. Yolov1 Architecture[14]

한 정보가 사라지는 문제를 상위 레이어 특징맵을 하위 레이어 특징맵에 합쳐주는 pass-through layer를 추가했다. 또한, 배치 정규화와 같은 추론 속도와 관련된 엔지니어링 개념들을 도입해 아키텍처의 변화 없이 기존의 YOLOv1 알고리즘보다 정확도가 높은 모델을 만들었다. 그 결과 추론 속도와 정확도를 개선했고, 실시간으로 9,000개의 서로 다른 카테고리들 감지하면서 강한 실시간 성능을 보였다. Table 1은 YOLOv2와 다른 기법 간 성능을 비교한 결과를 보여준다.

3) YOLOv3

YOLOv3[16]는 백본네트워크를 Darknet-53으로 변경했다. ResNet에서 제시한 skip connection 개념으로 많은 레이어를 쌓는 구조를 보여준다. 또한, GPU를 더 활용시켜 효율적이고 빠르게 수행하게 만들어 정확도는 유지하면서 FPS를 높게 개선했다[16,17]. 또한 목적에 따라 모델의 크기를 바꿈으로 속도와 정확도를 조절할 수 있다. Table 2은 YOLOv3와 이전 알고리즘들의 성능 비교이다. 정확도는 조금 상승하거나 비슷한 성능을 보이지만 추론 시간은 좋은 성능을 보인다.

Table 1. Compare with YOLOv2[15]

Detection Frameworks	Train	mAP	FPS
Fast R-CNN	PASCAL VOC 2007+2012	70.0	0.5
Faster R-CNN VGG-16		73.2	7
Faster R-CNN ResNet		76.4	5
YOLO		63.4	45
SSD300		74.3	46
SSD500		76.8	19
YOLOv2 288 × 288		69.0	91
YOLOv2 352 × 352		73.7	81
YOLOv2 416 × 416		76.8	67
YOLOv2 480 × 480		77.8	59
YOLOv2 544 × 544	78.6	40	

Table 2. Compare with YOLOv3 [16]

Method	mAP	mAP-50	time
SSD321	28.0	45.4	61
DSSD321	28.0	46.1	85
R-FCN	29.9	51.9	85
SSD513	31.2	50.4	125
DSSD513	33.2	53.3	156
FPN FRCN	36.2	59.1	172
RetinaNet-50-500	32.5	50.9	73
RetinaNet-101-500	34.4	53.1	90
RetinaNet-101-800	37.8	57.5	198
YOLOv3-320	28.2	51.5	22
YOLOv3-416	31.0	55.3	29
YOLOv3-608	33.0	57.9	51

MS COCO Object Detection

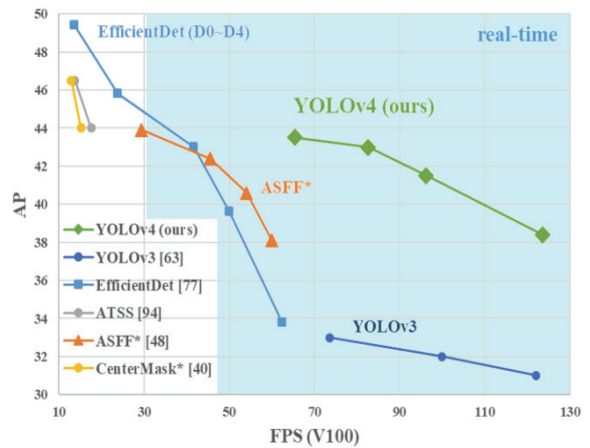


Fig. 3. Comparison Graph for YOLOv4[20]

4) YOLOv4

YOLOv4[18]는 YOLOv3와 같은 헤더를 사용해 유사한 구조를 가졌지만 속도와 정확도가 개선되었고 실시간 성능도 강력해졌다. 최근 GPU 자원의 활용에 대한 새로운 연구로 구조는 유사하지만 YOLOv3보다 뛰어난 성능을 보였다[10]. YOLOv4는 CSPDarknet53(Cross Stage Partial Darknet 53)을 채택했다. neck 부분에서는 SPP(Spatial Pyramid Pooling)모듈을 사용하여 CNN과 특징맵을 동일한 D × D 블록으로 나눈 뒤, bag-of-words를 사용해 특징을 추출하는 SPM(Spatial Pyramid Matching)과 결합했다. 또한, 정보흐름을 축소하여 low-level 정보를 최대한 보존하기 위해 FPN (Feature Pyramid Network) 대신 PAN(Path Aggregation Network)을 활용해 작은 객체에 대한 감지 능력이 상승했다[19]. Fig. 3은 YOLOv4와 다른 기법 간의 비교 그래프를 보여준다[20]. YOLOv3에 비하여 AP는 10%, FPS는 12% 향상된 것을 볼 수 있다.

본 논문에서는 컨테이너에서 작은 영역을 갖는 코너캐스팅 검출하기 위해서 YOLOv4를 주 학습 알고리즘으로 사용한다.

3. 컨테이너 적재 정렬 상태 및 사고 위험도 검출 기법

본 연구의 최종목표는 코너캐스팅의 정렬 상태를 판별하여 외부 요인으로 인한 컨테이너 안전사고를 사전에 예방하기 위함이다. 이를 위해 컨테이너 영상 데이터에서 컨테이너 위아래의 코너캐스팅을 기준으로 컨테이너의 정렬을 판별하였다[3].

3.1 컨테이너 정렬 상태 분류 기준

영상 데이터에서 컨테이너의 정렬 상태 분류 기준은 다음과 같이 정의한다. 컨테이너 위아래의 코너캐스팅 홀을 기준으로 면적이 50% 이상(0mm~25.5mm)이 일치한다면 Safe

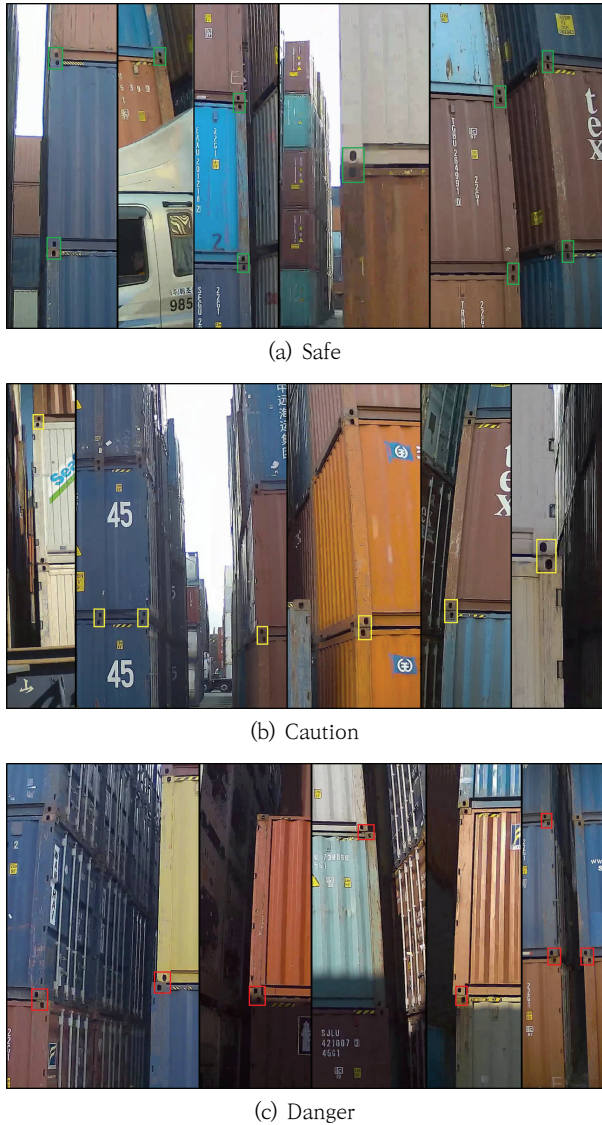


Fig. 4. Classification by Object

(안전), 0%를 초과하고 50% 미만의 영역(25.5mm~51.0mm)이 일치한다면 Caution(주의), 일치하는 영역이 없다(51mm 이상)면 Danger(위험)로 3가지의 클래스로 구분하여 위험도를 분류하였다. Fig. 4는 클래스별 예시 이미지이다. Fig. 4(a)는 Safe, Fig. 4(b)는 Caution, Fig. 4(c)는 Danger이다.

3.2 딥러닝 기반 컨테이너 적재 상태 검출 과정

딥러닝을 이용한 컨테이너 적재 정렬 상태 검출 과정은 다음과 같이 진행된다. 입력된 컨테이너 정렬 상태 이미지를 모델에 따라 416x416 또는 640x640 크기의 이미지로 변환한다. 변환한 이미지를 7x7 그리드로 나누어 각 그리드에서 경계 박스(Bounding Box)를 생성한다. 각 경계 박스에 대해서 신뢰도(Confidence score)를 구해 낮은 신뢰도를 갖는 경계 박스를 제거한다. 남은 경계 박스에 NMS 알고리즘을 적용해 겹쳐진 영역을 갖는 경계 박스 중 코너캐스팅을 잘 표현하는 경계 박스만 남긴다. 남은 경계 박스의 class conditional probabilities 구해 코너캐스팅이 어떤 정렬 상태를 가지는지 결정한다. Fig. 5는 딥러닝 기반 컨테이너 적재 상태 검출 과정의 흐름도이다.

4. 모델 구현 및 검증

4.1 컨테이너 정렬 상태 데이터셋

컨테이너 적재 상태 분류 모델에 학습하기 위한 컨테이너 정렬 데이터가 필요해 부산항에서 직접 취득한 영상 데이터를 활용했다. 취득한 영상 데이터가 모두 코너캐스팅만 있는 것은 아니므로 코너캐스팅이 있는 부분을 10프레임 간격으로 추출해 총 19,112장의 이미지를 생성했다. 생성한 이미지를 활용하기 위해 데이터에 Yolo_Label 프로그램을 활용해 라벨링했다. Table 3과 같은 기준으로 클래스별 Object_id를 부여하여 이미지 데이터에서 객체가 어느 위치에 어떤 객체가 있는지 라벨링 작업을 수행했다.

Table 3. Object_id for Class

Object_id	0	1	2
Class	Danger	Caution	Safe

4.2 실험환경 구성

분류 기준을 3가지로 구분한 코너캐스팅 데이터는 한 이미지에 2개 이상이 있는 경우를 포함해 총 79,423개로 구성했다. 자세한 오브젝트 수는 Table 4와 같다.

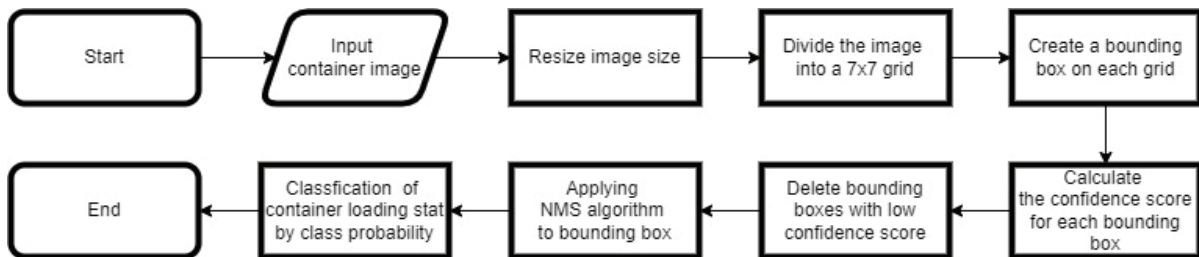


Fig. 5. Flowchart for Detecting Container Load State

Table 4. Number of Objects

Class	Danger	Caution	Safe	Total
Count	16,878	21,506	41,039	79,423

데이터 구성은 컨테이너 정렬 데이터셋에서 전체 데이터의 70%(55,597개)를 훈련 데이터로, 15%(11,913개)를 검증 데이터로, 15%(11,913개)를 시험 데이터로 구분하여 실험을 진행했다. 가벼우면서도 비교적 정확도가 높고 실시간 오브젝트 탐지에 적합한 YOLO 모델을 사용했다. 산업에서 일반적으로 많이 쓰이는 모델인 YOLOv3과 YOLOv4 모델에 각각 이미지 크기를 416픽셀(px)과 640픽셀(px)로 적용했으며, 각 모델은 총 200 에폭(Epoch) 동안 학습시켰다. 일반적으로 객체 감지 모델은 IoU(Intersection of Union)를 통한 mAP(mean Average Precision)를 성능 지표로 사용하지만, 해당 모델에서는 검출된 박스의 크기보다 검출 여부와 분류의 정확도가 중요하기에 속도, 검출 정확도, 분류 정확도, 실제 임베디드 장치에서의 속도 총 4가지 항목을 각각 측정했다.

4.3 성능평가

본 논문에서는 검출알고리즘과 이미지 해상도에 따른 추론 속도, 검출 정확도, 분류 정확도, 임베디드 장치에서의 FPS를 기준으로 모델의 성능을 비교한다. 추론 속도는 한 장의 이미지를 추론하는 속도, 검출 정확도는 TP(True Positive)의 비율, 분류 정확도는 Micro-F1과 Cohen's Kappa Score, 임베디드 장치에서의 FPS는 실제 임베디드 장치에서 영상을 추론할 때, 평균 FPS를 성능 지표로 사용한다. 일반적인 지표는 NVIDIA RTX3080 GPU를 통해 측정하였고 임베디드 디바이스에서의 성능평가는 NVIDIA Jetson AGX Xavier 개발자 키트를 사용해 진행하였다. Fig. 6은 각 모델의 컨테이너 적재 상태 검출 결과 이미지이다.

감지 모델별 추론 결과는 입력이미지 해상도별 차이보다 학습 알고리즘별 차이가 명확하게 보였다. YOLOv3 모델은 약 2개의 코너캐스팅을 감지하지 못 했고, 잘못된 감지 결과를 보였다. 하지만, YOLOv4 모델은 모든 코너캐스팅을 감지하고 정확한 감지 결과를 보였다.

1) Speed

Table 5는 YOLOv3과 YOLOv4 모델이 이미지 한 장을 추론하는 시간을 보여준다. YOLOv4(8.7ms)가 YOLOv3(8.1ms)보다 조금 느린 추론 속도를 보였지만 이미지 해상도 측면에서 검출알고리즘보다 큰 차이를 보였다. 416px(8.1ms)이 640px(14.6ms)보다 약 2배의 빠른 추론 속도를 보였다.

2) Detection

Table 6은 컨테이너의 코너캐스팅을 얼마나 잘 찾아내는 지 확인할 수 있는 지표에 대한 결과를 나타낸다. 입력 이미

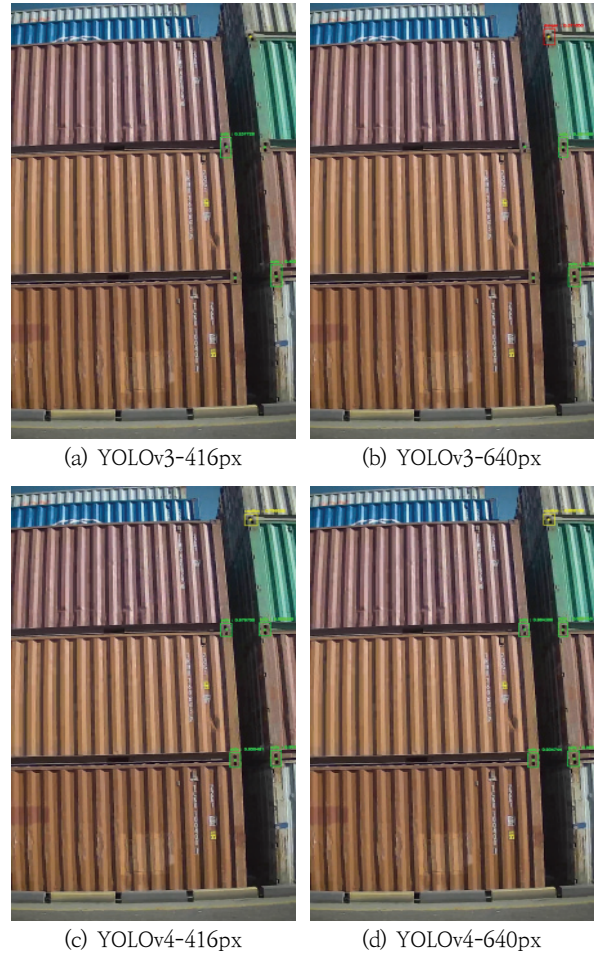


Fig. 6. Inference Results by Model

Table 5. Speed of Inference per Image

	YOLOv3		YOLOv4	
	416px	640px	416px	640px
Speed	8.1ms	14.6ms	8.7ms	16.0ms

Table 6. Result for Detection Model

		YOLOv3		YOLOv4	
		416px	640px	416px	640px
Speed		8.1ms	14.6ms	8.7ms	16.0ms
Danger	Detected	1486	1522	2461	2591
	Missed	1105	1069	130	101
	Total	2591	2591	2591	2591
Caution	Detected	1879	1854	3041	3194
	Missed	1315	1340	153	104
	Total	3194	3194	3194	3194
Safe	Detected	3703	3665	5866	6138
	Missed	2435	2473	272	185
	Total	6138	6138	6138	6138
Accuracy		0.593	0.591	0.953	0.967

Table 7. Classification Result

	YOLOv3		YOLOv4	
Micro-F1	0.796	0.820	0.902	0.925
Cohen's Kappa	0.653	0.696	0.840	0.879

Table 8. FPS on Embedded Device

		YOLOv3		YOLOv4	
		416px	640px	416px	640px
FPS	Min	11.3	8.4	10.9	5.2
	Average	12.8	8.7	11.3	5.9
	Max	15.0	9.0	13.6	7.5

지의 크기와는 상관없이 YOLOv4 모델이 YOLOv3 모델보다 더 좋은 결과를 보여줬으며, 각 모델에서 입력 이미지의 크기는 큰 차이를 보이지 않았다.

3) Classification

검출된 코너캐스팅의 위험도를 얼마나 정확하게 분류하는지에 대한 지표를 확인한다. 클래스별 오브젝트의 수가 균일하지 않은 불균형한 데이터셋이므로 성능 측정 지표로 Micro-F1과 Cohen's Kappa Score를 사용하였다. 측정 결과는 Table 7과 같다.

4) 임베디드 디바이스

본 모델을 성능이 제한되는 임베디드 장치에서 어떤 모델이 사용하기에 적합한지 확인하기 위해 비교해 보았다. Table 8은 두 모델(YOLOv3, YOLOv4)을 임베디드 장치에서 성능평가를 한 결과이다.

FPS는 YOLOv4(11.4ms)로 YOLOv3(11.6ms)으로 큰 차이를 보이지 않았지만, 이미지 해상도 측면에서는 416px이 640px보다 우수한 성능을 보였다.

5. 결 론

본 논문은 컨테이너 붕괴 사고를 예방하기 딥러닝 기반의 컨테이너 사고 위험도를 검출하는 기법을 제안한다. 직접 취득한 컨테이너 영상 데이터를 활용해 컨테이너 이미지 데이터셋을 구축했다. YOLOv3와 YOLOv4의 학습 알고리즘을 이용해 컨테이너 사고 위험도 검출 모델을 구현하였다. YOLOv4 모델은 YOLOv3 모델보다 처리 속도는 낮지만 사고 감지와 위험도를 측면에서 정확도가 매우 우수함을 확인하였다. 따라서 제안된 시스템에서는 YOLOv4 기반 딥러닝 컨테이너 사고 위험도 검출 모델을 제안하였다. 제안된 시스템은 현재 NVIDIA Jetson AGX Xavier 개발자카드 환경에서 구현 중에 있으며 추후 NVIDIA Jetson AGX Xavier 보다 저사양의 임베디드 디바이스에서 동일한 성능으로 활용할 수 있는 방안을 연구할 예정이다.

References

- [1] T. H. Lee, "Smart port policy trend of europe and singapore and its political implications," *Journal of Korea Port Economic Association*, Vol.36, No.1, pp.77-90, 2020.
- [2] C. A. Durán, C. Fernández-Campusano, R. Carrasco, M. Vargas, and A. Navarrete, "Boosting the decision-making in smart ports by using blockchain," *IEEE Access*, Vol.9, pp.128055-128068, 2021.
- [3] E. Nishimura, "Yard arrangement problem with the external truck arrival," *2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp.88-92, 2021.
- [4] G. Parise, L. Parise, L. Martirano, P. B. Chavdarian, C. -L. Su, and A. Ferrante, "Wise port and business energy management: Port facilities, electrical power distribution," *IEEE Transactions on Industry Applications*, Vol.52, No.1, pp.18-24, 2016.
- [5] K. -L. A. Yau, S. Peng, J. Qadir, Y. -C. Low, and M. H. Ling, "Towards smart port infrastructures: enhancing port activities using information and communications technology," *IEEE Access*, Vol.8, pp.83387-83404, 2020.
- [6] Container Cargo Processing Performance (2022) [Internet], https://www.index.go.kr/potal/main/EachDtlPageDetail.do?idx_cd=1267#quick_02.
- [7] S. Y. Oh, J. H. Jeong, B. Choi, J. H. Yeon, Y. U. Seo, S. W. Kim, and J. S. Youn, "A study on deep learning model for container load status monitoring," *Proceedings of the Annual Spring Conference of Korea Information Processing Society Conference (KIPS) 2022*, Vol.29, pp.320-321, 2022.
- [8] J. Zhao, S. Hao, C. Dai, H. Zhang, L. Zhao, Z. Ji, and I. Ganchev, "Improved vision-based vehicle detection and classification by optimized YOLOv4," *IEEE Access*, Vol.10, pp.8590-8603, 2022.
- [9] J. Fan, J. Lee, I. Jung, and Y. Lee, "Improvement of object detection based on faster R-CNN and YOLO," *2021 36th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pp.1-4, 2021.
- [10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.39, No.6, pp.1137-1149, 2017.
- [11] Q. Guo, L. Liu, W. Xu, Y. Gong, X. Zhang, and W. Jing, "An improved faster R-CNN for high-speed railway dropper detection," *IEEE Access*, Vol.8, pp.105622-105633, 2020.

- [12] Y. H. Lee and Y. S. Kim, "Comparison of CNN and YOLO for Object Detection," *Journal of the Semiconductor & Display Technology*, Vol.19, No.1, pp.85-92, 2020.
- [13] Y. Cai et al., "YOLOv4-5D: An Effective and Efficient Object Detector for Autonomous Driving," *IEEE Transactions on Instrumentation and Measurement*, Vol.70, pp.1-13, 2021.
- [14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.779-788, 2016.
- [15] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.7263-7271, 2017.
- [16] J. Redmon and A. Farhadi, Yolov3: An incremental improvement, *arXiv preprint arXiv:1804.02767*, 2018.
- [17] Z. Cong and X. Li, "Track obstacle detection algorithm based on YOLOv3," *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp.12-17, 2020.
- [18] H. Wang, H. Pei, and J. Zhang, "Detection of locomotive signal lights and pedestrians on railway tracks using improved YOLOv4," *IEEE Access*, Vol.10, pp.15495-15505, 2022.
- [19] T. Yang and C. Tong, "Small Traffic Sign Detector in Real-time Based on Improved YOLO-v4," *2021 IEEE 23rd Int Conf on High Performance Computing & Communications: 7th Int Conf on Data Science & Systems: 19th Int Conf on Smart City: 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, pp.1318-1324, 2021.
- [20] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, Yolov4: Optimal speed and accuracy of object detection, *arXiv preprint arXiv:2004.10934*, 2020.



연정흠

<https://orcid.org/0000-0002-5872-0727>
 e-mail : jhyeon@busanpa.com
 2001년 한국해양대학교 물류시스템공학 (학사)
 2003년 한국해양대학교 물류시스템공학 (석사)

2006년 한국해양대학교 물류시스템공학(박사)
 2006년 ~ 현 재 부산항만공사 항만 R&D실 실장
 관심분야: 스마트 물류, 항만 자동화, 디지털 전환



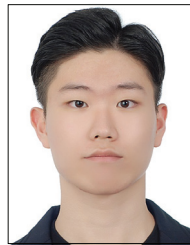
서용욱

<https://orcid.org/0000-0003-2129-5760>
 e-mail : craser@naver.com
 2003년 부산외국어대학교 경영학과(학사)
 2022년 동의대학교 IT융합학과(석사)
 2022년 ~ 현 재 (주)서안에스앤씨 대표
 관심분야: 인공지능, 영상인식, 사물인터넷



김상우

<https://orcid.org/0000-0001-5544-6008>
 e-mail : woogigi@hanmail.net
 1999년 동서대학교 컴퓨터공학과(학사)
 2021년 부산외국어대학교 ICT창의융합학과 (석사)
 2021년 ~ 현 재 동의대학교 인공지능학과 박사과정
 2021년 ~ 현 재 (주)서안에스앤씨 부장
 관심분야: 머신러닝, 딥러닝, 빅데이터, 데이터 관리



오세영

<https://orcid.org/0000-0002-8771-5839>
 e-mail : osy0784@gmail.com
 2022년 동의대학교 응용소프트웨어공학 (학사)
 2022년 ~ 현 재 동의대학교 IT융합학과 석사과정

관심분야: 객체인식, 딥러닝, 영상처리



정준호

<https://orcid.org/0000-0002-4603-5917>
 e-mail : jeong@junho.dev
 2021년 동의대학교 응용소프트웨어공학 (학사)
 2021년 ~ 현 재 동의대학교 IT융합학과 석사과정

관심분야: 인공지능, 머신러닝, 강화학습



박진효

<https://orcid.org/0000-0003-1646-6035>
 e-mail : jhpark3679@gmail.com
 2021년 동의대학교 응용소프트웨어공학 (학사)
 2021년 ~ 현 재 동의대학교 IT융합학과 석사과정

관심분야: 딥러닝, 빅데이터, 데이터 관리



김 성 희

<https://orcid.org/0000-0002-9716-8349>

e-mail : sh.kim@deu.ac.kr

2006년 이화여자대학교

컴퓨터정보통신공학과(학사)

2008년 이화여자대학교

컴퓨터정보통신공학과(석사)

2014년 Purdue University 산업공학(박사)

2017년 ~ 현 재 동의대학교 산업융합시스템공학부 조교수

관심분야 : 데이터시각화, 기계학습, 인간-컴퓨터 상호작용



윤 주 상

<https://orcid.org/0000-0001-9952-9649>

e-mail : jsyoun@deu.ac.kr

2001년 고려대학교 전기전자전파공학과
(학사)

2003년 고려대학교 전자공학과(석사)

2008년 고려대학교 전자컴퓨터공과(박사)

2008년 ~ 현 재 동의대학교 산업ICT기술공학과 교수

관심분야 : 사물인터넷, 엣지컴퓨팅, 클라우드, 강화학습