

딥러닝을 활용한 건설 현장 사고 예방

Prevention of Construction site Accidents using Deep Learning

김지수* 박세연**

Kim, Ji-Su, Park, Se-Yeon

키워드 : 딥러닝, 건설 안전, 센서, 건설 장비, 이미지 처리

Keywords : Deep Learning, Construction Safety, Sensor, Construction Equipment, Image Processing

1. 연구 배경 및 목적

1.1 건설 산업 재해 현황

건설 산업 현장에서의 사고는 피할 수 없는 문제이다. 국토교통부에서 조사한 업종별 산업 재해 사고 사망 통계에 따르면(그림 1- [1]) 건설업이 52%로 절반을 넘는 비율을 차지하고 있다. 이는 다른 업종보다 사고가 발생할 확률, 즉 위험성이 높다는 것으로 해석된다.

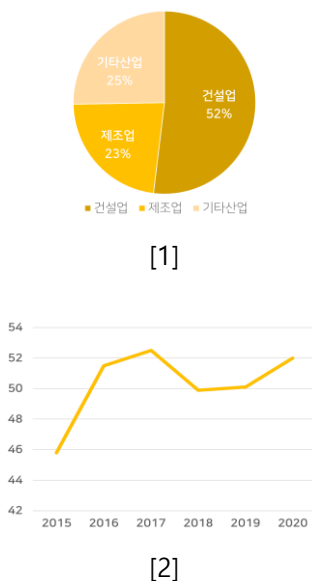


그림 1. [1] 2020 업종별 산업재해 사고 사망 비율

[2] 산업재해 사망자 중 건설업 비율

연도별로 조사한 건설업 산업재해 사망자 비율(그림 1- [2])을 봤을 때 그 비율은 매 해 증가하고 있다. 현장 점검 확대, 입찰 및 발주 방식 개선, 건설안전특별법 개선 등 정부에서는 건설현장 안전문제를 인식하고 이에 대한 해결책을 제시하고 있다. 하지만 현장에서 사망사고가 끊이지 않는 것으로 보아 새로운 안전관리 방식의 도입이 필요해 보인다.

1.2 연구 목적

4 차 산업혁명 시대를 맞아 다양한 기술들이 현대 시대를 이끌어 가고 있다. 그 중 하나가 알고리즘을 시작으로 대중들에게 알려지기 시작한 ‘인공지능’이다. 이는 비교적 적은 데이터로 인공지능을 학습하여 원하는 결과를 도출해낼 수 있다는 점에서 장점을 갖는다. 목적에 따라 데이터를 달리한다면 그 결과 또한 달라지게 활용성은 무궁무진하다. 이는 건축 분야에서도 마찬가지이다. 활용방식에 따라 유의미한 목적을 이뤄낼 수 있는데, 본 연구에서는 안전관리를 목적으로 한다.

앞의 내용과 같이 연간 산업 재해 사망자 중 건설업은 매년 절반 이상을 차지할 정도로 사고 사망율이 높으며 매해 비율이 증가하고 있다. 국가에서도 심각성을 인지하고 제도 마련에 관심을 갖고 있으나 개선 효과가 미미하다. 따라서 새로운 안전관리 방식을 도입할 필요가 있다. 본 연구는 사람의 노력으로 사고를 예방하는데 한계가 있다고 판단하여 딥러닝을 활용해 건설현장 관리를 자동화하고자 한다. 이를 통해 제도적인 예방에서 나아가 안전이 보장되는 건설현장을 만드는 것이 본 연구의 최종 목적이다.

2. 연구 설계 방향

건설 현장에서 사용하는 장비는 대부분이 중장비이기 때문에 사고발생 시 그 심각성과 위험성이 높다. 장비 운행 시, 운전자가 주의를 기울인다고 해도 전진 운행과 후진 운행이 많기에 비가시적인 상황에 놓일 수 있다. 해당 이유로 건설현장에서는 잠재적인 사고 발생율이 높다. 안전보건공단에서 2021 년 발행한 건설기계 사망사고 발생경보에 따르면 4 월 내에만 차량계 건설기계 사망사고 5 건이 연이어 발생하였다. 재해 발생 원인을 분석한 결과, 차량계 건설 기계 접촉 방지 미실시, 차량계 건설기계 작업 계획서 미 작성 등 차량계 건설기계를 원인으로 재해가 발생한 것을 확인할 수 있다. 따라서 본 연구는 작업자와 장비, 장비와 장비 간의 상황을 인식하도록 하여 안전관리를 자동화하고자 했다.

* 김지수, 성균관대학교 건축공학과, jisu7935@g.skku.edu

** 박세연, 성균관대학교 건축공학과, seyeon9704@skku.edu



그림 2. 안전보건공단 건설기계사망사고 발생경보 2021

3. 연구 이론적 배경

본 연구에서는 건설 현장 자동화의 방법으로 영상에서 객체를 인식하는 과정을 포함한다. 영상에서 객체를 인식하기 위해 실시간으로 보장된 segmentation 성능을 보이는 딥러닝 모델 YOLACT(you only look at coefficients)를 사용한다.

3.1 객체 인식 선행 연구

YOLACT 이전의 객체 탐지 모델은 두 단계에 걸쳐 localization 을 먼저 수행하고 classification 과 segmentation 을 수행한다. Mask-RCNN 모델을 참고하면 RPN 에서 관심영역 RoI 를 추출하는 localization 을 먼저 수행하는 것을 확인할 수 있다.

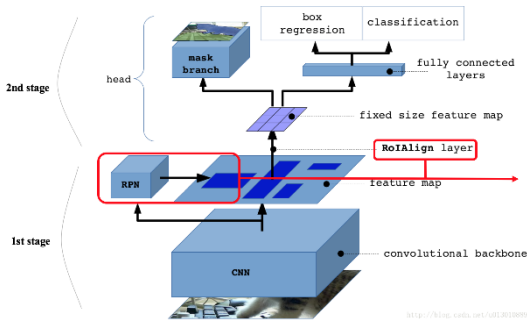


그림 3. Mask-RCNN architecture

3.2 YOLACT

본 연구에서는 건설현장 자동화를 위해 딥러닝 모델 YOLACT(you only look at coefficients)를 사용한다. YOLACT 모델을 통해 클래스를 분류해 Bounding Box 를 그려내는 object detection 과 pixel 단위로 객체의 영역을 탐지하는 instance segmentation 모두 실현할 수 있다.

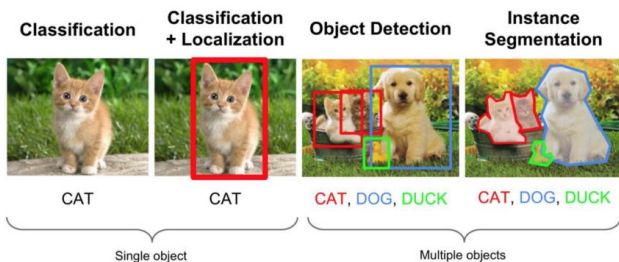


그림 4. Object Detection & Instance Segmentation

YOLACT 는 기존 객체 인식 방식에서 속도를 개선한 모델이다. 속도를 개선하기 위하여 기존에 두 단계로 진행했던 localization 대신에 두 개의 substack 을 병렬로 실행한다. Prediction head 와 Protonet 두 개의 substack 을 병렬로 실행함으로써 localization 을 수행할 수 있다.

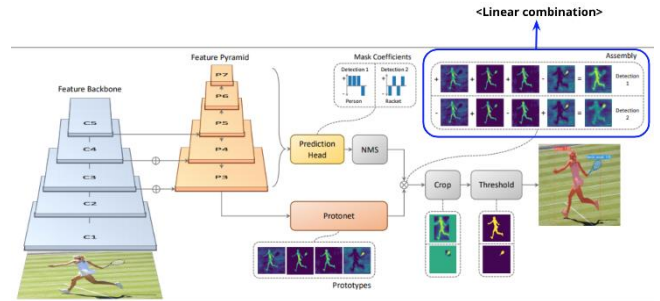


그림 5. YOLACT architecture

Protonet 에서는 prototype mask 를 생성한다. 생성된 prototype mask 는 이미지 개수(k) 에 따라 채널 k 개를 생성하는데, 채널마다 이미지의 다른 영역을 activate 한다. 그림 6 의 prototype mask 로 예를 들면 이미지는 a~f 6 개가 존재하며 그에 따라 채널도 6 개가 생성된다. 채널 1~3 은 점선 곡선 안쪽 객체를, 채널 4 는 이미지 좌하단을, 채널 5 는 배경과 객체 간의 edge 를, 채널 6 은 ground 를 각각 activate 한다.

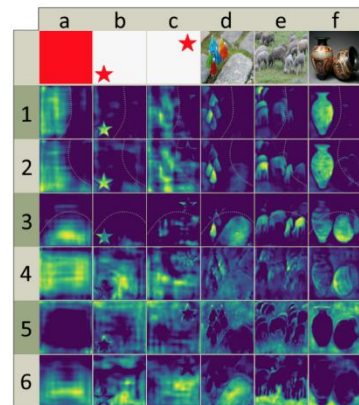


그림 6. Prototype mask

activate 하는 방식은 padding 을 이용한다. 그림 7 과 같이 이미지에 zero-padding 을 추가하고 왼쪽만 1 로 활성화된 필터와 convolution 연산을 거치면 오른쪽만 활성화된 output 이 결과로 도출된다. 이와 같은 방식으로 원하는 이미지 영역을 activate 한다.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 2 \\ 0 & 3 & 3 \\ 0 & 2 & 2 \end{bmatrix}$$

그림 7. Padding 을 통한 activate

Protonet 에 의해 생성된 prototype mask 는 특정 영역만 activate 하고 있는데, instance segmentation 을 위해서는 하나의 instance 만 activate 해야 한다.

따라서 prediction head 를 통해 mask coefficient 를 생성 및 적용하여 하나의 instance 만 activate 할 수 있다. 그림 8 에서 확인할 수 있듯, 그림 6 의 이미지 d[1]에 대해 빨간색(R), 초록색(G) 이 강조된 채널 2[2]와 빨간색(R)이 강조된 채널 3[3]이 있다. [2]에서 [3]을 뺀다면 초록색(G)만 localization 된 결과가 남는다. 이와 같이 채널끼리 감하여 하나의 instance 만 activate, 즉 localization 할 수 있다.



그림 8. Mask Coefficient 적용 방법

YOLOACT 에서 속도 개선을 위한 또 하나의 차이점은 Fast NMS 이다. 앞의 과정들을 거치면 각 픽셀마다 앵커박스를 생성하기에 여러 앵커박스가 하나의 인스턴스를 가리켜 겹치는 모양이 된다. 기존에는 Non-Maximum Suppression(NMS)를 적용하여 신뢰도가 가장 높은 앵커박스 하나만 남겨 Object Detection 을 수행했다.



그림 9. [1] NMS 적용 전 [2] NMS 적용 후

YOLOACT 에서는 NMS 에 행렬 연산을 도입하여 GPU 로 연산을 수행하고 for 문을 한 번으로 축소하여 기존의 NMS 보다 빠른 속도를 자랑한다. 결과적으로 기존에 사용하던 객체 인지 알고리즘과 비교했을 때 그림 10 과 같이 FPS 와 TIME 측면에서 우수한 성능을 보인다. 본 연구는 건설 현장 장비에 카메라를 설치해 실시간으로 객체를 인식하는 것을 가정한다. 따라서 실시간으로 보장된 segmentation 성능을 보이는 YOLOACT 모델을 본 연구의 딥러닝 모델로 채택하였다.

Method	Backbone	FPS	Time	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
PA-Net [29]	R-50-FPN	4.7	212.8	36.6	58.0	39.3	16.3	38.1	53.1
RetinaMask [14]	R-101-FPN	6.0	166.7	34.7	55.4	36.9	14.3	36.7	50.5
FCIS [24]	R-101-C5	6.6	151.5	29.5	51.5	30.2	8.0	31.0	49.7
Mask R-CNN [18]	R-101-FPN	8.6	116.3	35.7	58.0	37.8	15.5	38.1	52.4
MS R-CNN [20]	R-101-FPN	8.6	116.3	38.3	58.8	41.5	17.8	40.4	54.4
YOLOACT-550	R-101-FPN	33.5	29.8	29.8	48.5	31.2	9.9	31.3	47.7
YOLOACT-400	R-101-FPN	45.3	22.1	24.9	42.0	25.4	5.0	25.3	45.0
YOLOACT-550	R-50-FPN	45.0	22.2	28.2	46.6	29.2	9.2	29.3	44.8
YOLOACT-550	D-53-FPN	40.7	24.6	28.7	46.8	30.0	9.5	29.6	45.5
YOLOACT-700	R-101-FPN	23.4	42.7	31.2	50.6	32.8	12.1	33.3	47.1

그림 10. 객체 인지 알고리즘 성능 비교

4. 연구 내용 및 과정

4.1 연구 진행 절차

YOLOACT 를 활용하여 건설 장비와 사람을 실시간으로 인식하여 위험 상황을 판단하는 연구를 위해 먼저 인식할 데이터 셋을 정의하여 YOLOACT 모델을 학습했다. 학습한 모델에서 위험 지역을 설정한 후 해당 위험 지역에 객체의 중심점 위치에 따라 위험도를 표시하였다. 그리고 현장에서 직접 촬영한 영상을 이용해 최종 모델을 테스트하는 과정으로 연구를 진행하였다.

4.2 데이터 생성



그림 11. 건설기계 산업재해 사망자 수

그림 11 의 한국건설안전학회의 통계 자료를 통해 보았을 때 건설 현장에서 발생하는 사망 사고 중, 굴삭기, 트랙터에 의한 사망자 수가 높은 비율을 차지하였다. 이를 토대로 인식할 건설 장비 데이터로 굴삭기, 트랙터, 그 외에 불도저까지 3 개의 장비를 선정하였다.

해당 연구에서는 YOLOACT 에서 사용하기 위한 데이터셋으로 COCO Dataset 을 생성하였다. COCO Dataset 은 Object Detection, segmentation, Keypoint Detection 을 위한 데이터셋으로 데이터 이미지에 대한 Bounding Box 와 segmentation 정보를 포함한다.

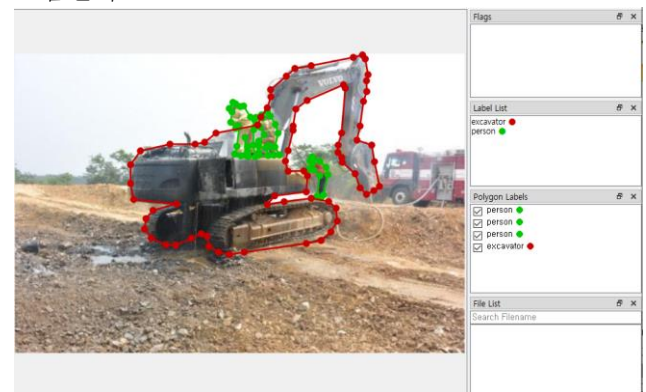


그림 12. labelme

그림 12 과 같이 labelme 프로그램을 사용하여 건설 장비(excavator, truck, bulldozer)와 사람(person) 클래스로 데이터를 라벨링 하였다. labelme 는 이미지 annotation tool 로 annotation 을 수행할 이미지를 선택하고 마우스 클릭을 통해 객체를 지정하여 클래스를 설정해줄 수 있는 프로그램이다. 이렇게 라벨링한 데이터는 json 파일로 생성이 되고, 해당 json 파일을 COCO Dataset 으로 변환하는 작업을 수행하였다. 이 작업은 labelme2coco.py 프로그램을 사용하였다. 이렇게 굴삭기(excavator), 불도저(bulldozer), 덤프트럭(truck), 사람(person)이 포함된 약 200 개의 이미지를 라벨링하고 COCO Dataset 을 생성하였다.

4.3 YOLACT 모델 학습

다음으로 생성한 데이터셋을 활용하여 YOLACT 모델을 학습시켰다. YOLACT 를 사용하여 카메라를 이용해 얻는 실시간 영상에서 건설 장비들과 사람을 인식하고, 설정한 위험지역에 인식한 객체들이 위치하는 경우 위험을 알리고자 한다. YOLACT 학습 시 사용하는 설정 파일(config.py)에 생성한 COCO Dataset 을 저장한 경로를 지정해주고, 데이터에 맞게 class name 을 지정하여 데이터셋을 정의하였다.

Calculating mAP...

	all	.50	.55	.60	.65	.70	.75	.80	.85	.90	.95
box	59.47	69.62	69.62	69.62	69.62	69.62	69.62	69.62	62.44	33.66	11.29
mask	45.76	69.62	69.62	61.53	51.53	50.25	50.00	50.00	39.73	25.33	0.00

그림 13. mAP

수정한 설정 파일을 사용하여 모델 학습을 수행하였고, 학습을 종료하여 얻은 모델의 정밀도 값은 그림 13 과 같다. 현재 사용한 이미지는 200 개이고, 학습 환경은 구글 Colab 에서 수행하였다. 학습한 환경이 GPU 를 활용한 효율적인 학습이 불가능한 환경이라는 점에서 모델의 정밀도가 그림 13 에서 얻은 값 이상으로 올라가지 않는다고 판단하여 해당 값에서 학습을 종료하였다.

4.4 위험 지역 설정

다음으로는 인식한 객체를 활용하여 위험 상황을 판단하기 위해 위험 지역을 설정했다. 해당 연구에서는 센서를 사용하여 거리를 측정하는 등의 방법을 사용하지 않았기 때문에 인식한 객체 간의 실제 거리를 구할 수 있는 방법이 없다. 따라서 위험 상황을 판단하기 위해 카메라를 부착한 장비에서 인식한 영상 내에 위험 지역을 설정하였다. 영상 아래 쪽에 다른 장비의 중심점이 위치할 경우, 해당 장비와 거리가 가깝다고 판단하였다.

```
m1 = int((x1 + x2) / 2) # 가로 중심
m2 = int((y1 + y2) / 2) # 세로 중심
```

그림 14. 중심점 좌표 코드

먼저 인식하는 객체의 중심점을 표시하기 위해 YOLACT 에서 eval.py 파일을 수정하였다. eval.py

코드에서 인식한 객체의 Bounding Box 에서 각 꼭지점의 좌표 값을 얻을 수 있다. 그림 14 와 같이 얻은 꼭지점 좌표 값을 사용하면 객체의 가로 중심, 세로 중심을 구할 수 있다.



그림 15. 중심점 인식 결과 이미지

객체의 중심 좌표를 인식한 이미지(실제로는 영상)에 표시하면 그림 15 과 같다. 사진으로 인식하는 것과 같이 영상에서도 인식하는 객체의 중심 좌표의 값을 얻을 수 있다.



그림 16. 위험 지역 영역 예시

다음으로는 위험 지역을 구분한 방법이다. 건설 장비는 보통 차체가 높아 장비의 운전석에 위치한 운전자는 장비의 뒤쪽, 그리고 아래 쪽에 사각이 생긴다. 따라서 영상을 인식할 카메라를 장비 후미에 위치시킨다고 가정한다. 해당 위치에 카메라를 설치한다고 가정하였을 때, 그림 16 과 같이 초록색 부분은 장비에서 먼 거리, 노란색 부분은 장비와 중간 거리, 빨간색 부분은 장비와 가까운 거리라고 판단한다. 따라서 객체의 중심점이 어느 범위에 위치하는지에 따라 위험도를 다르게 표시한다. 이번 연구에서는 위험 표시를 다르게 하였고, 후속 연구에서는 경고음을 다르게 발생시키고자 한다.

4.5 결과 분석



그림 17. 이미지 인식 결과 1

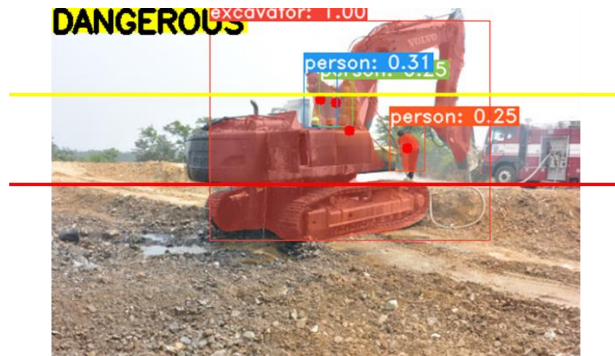


그림 18. 이미지 인식 결과 2

그림 17, 18 에서 보는 것과 같이 인식한 객체의 중심점이 위치하는 부분에 따라 상단 왼쪽의 DANGEROUS 의 배경이 다른 색으로 표시된다. 만약 객체의 중심점이 다른 부분에는 위치하지 않고 초록색 범위 부분에만 위치한다면 상단에는 SAFE 가 표시된다. 이를 먼 곳에서부터 객체에 가까워지는 방식으로 직접 촬영한 영상에 적용했다. 그 결과 인식한 객체의 중심점에 따라 상단의 위험도 표시가 실시간으로 바뀌는 것을 확인할 수 있었다.

5. 결론

본 논문에서는 산업 재해 중 지속적으로 높은 비율을 차지하는 건설 현장 사고를 해결하고자 했다. 그 방법으로 4 차 산업의 핵심 기술인 딥러닝의 영상 인식 알고리즘을 사용하였다. 이를 통해 건설 현장 안전 관리를 자동화하여 사고 발생률을 줄이는 방안에 대해 연구하였다.

인식할 건설 장비 데이터를 직접 수집 및 라벨링 하였다. 해당 데이터로 학습한 딥러닝 모델을 영상에 적용하여 위험 상황을 인식했다. 객체의 중심좌표가 영상 위험 지역에 위치하는 지를 판단하고, 위험도를 알려주는 방식으로 구현하였다.

본 연구의 한계점으로는 실제 공사 현장의 지형에 대한 고려가 부족하다는 점이 있다. 실제 공사 현장은 단순히 평지가 아니라 지형 상 높낮이가 다르기 때문에 단순히 영상에서의 좌표 분할로 위험지역을 설정하는 방법은 정확도가 떨어질 수 있다. 후속 연구에서

는 공사 현장의 지형을 고려하여 위험지역을 설정하는 과정이 필요하다. 또한 해당 연구에서는 운전자의 화면에 위험도를 표시하는 방식으로 위험을 알린다. 후속 연구에서는 위험 인식률을 높이기 위해, 운전자와 작업자에게 소리로 위험을 알리도록 구현하고자 한다.

References

- [1] 심승보, 최상일, “YOLO-v3 을 활용한 건설 장비 주변 위험 상황 인지 알고리즘 개발 (2019)”
- [2] Daniel Bolya, Chong Zhou, Fanyi Xiao, Yong Jae Lee, “YOLACT: Real-time Instance Segmentation (2019)”
- [3] YOLACT github, “Installation YOLACT” , <https://github.com/dbolya/YOLACT>
- [4] Immersive Limit, “Train YOLACT with a Custom COCO Dataset” , <https://www.immersivelimit.com/tutorials/train-YOLACT-with-a-custom-coco-dataset>
- [5] 한즈온머신러닝, “Make COCO dataset” , <https://hansonminlearning.tistory.com/52>
- [6] 프라이데이, “YOLACT 설명 및 정리” , <https://ganghee-lee.tistory.com/42>