YOLOv5 기반 밀집도 추정(Density Estimation based on YOLOv5)

CUAI 5기 CV T2
김태윤(소프트웨어학부), 오용희(소프트웨어학부), 이강민(산업보안학과), 정다연(산업보안학과)

[요약] 본 연구의 목적은 딥러닝 모델을 활용하여 인파로 인한 사고가 일어날 가능성을 사전에 파악해주는 시스템을 구현하는 것이다. 본 연구에서는 YOLOv5 모델을 사용해 사람이 존재하는 이미지를 입력하여 그 수를 파악하였고, 면적 당 인원을 기준으로 한 밀집도 경보 단계를 기반으로 이미지 내 상황이 안전사고가 발생할 가능성이 있는지를 진단하였다. 본 연구는 작년 이태원 압사사고와 같은 비극이 되풀이되지 않도록 AI 기술을 활용한 위기경보 시스템의 구축을 시도했다는 점과 도심 내 집회 및 행사 안전관리에 해당 연구가 유용하게 쓰일 수 있다는 점에서 의의가 있다.

1. 서 론

2022년 10월 29일, 대한민국 서울 이태원에서 일어나서는 안될 끔찍한 압사사고가 발생했다. 현재까지도 사고의 원인에 관하여 여러 의견들이 오가고 있지만, 인파가 몰리기 시작한 이른 저녁 시간부터 주변 경찰 인력이 사전에 적절하게 통행량을 조절하는 등의 조치를 취했더라면 이렇게 많은 희생자가 나오지는 않았을 것이라는 의견에 많은 사람들이 공감하고 있다. 사고 현장 주위에는 CCTV가 여러 대 설치되어 있었고 당시 사람들이 많이 몰린 모습이 고스란히 찍혀 있었지만, 이렇게 인파가 몰리고 있다는 것을 기기가 사람에게 자동으로 알려주는 시스템이 없다 보니 현장 상황을 정확하게 파악한 뒤 신속하게 대응하지 못했다.

현재 Computer Vision 분야 객체 탐지(object detection) 모델들의 정확도는 날이 갈수록 향상되고 있고, 이러한 모델들을 일상의 문제 해결에 적용하는 연구가 활발하게 진행되고 있다. 이번 이태원 사고 이후 행정안전부에서도 AI 기술을 활용하여 인파 밀집 위험을 감지한 뒤 휴대전화로 경보를 보내는 체계가 구축될 수 있도록 2024년까지 연구개발을 진행할 것이라고 발표했다.[1] 따라서 본 연구에서는 현재 공개된 객체 탐지 모델로는 어느 수준까지 인파 밀집을 파악할 수 있는지를 알아보기 위해 YOLOv5 모델을 토대로 카메라가 촬영하고 있는 공간의 인파 밀집도를 추정해보고자 한다.

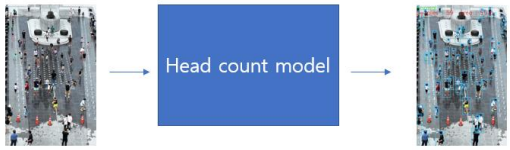

그림 1. 객체 탐지 흐름 요약도

2. 본 론

2.1. 선행 연구

1) YOLO

YOLO[2]는 2016년 Joseph Redmon이 최초로 제안한 객체 탐지 신경망으로, 객체 탐지 알고리즘 중 속도가 빠른 편에 속하기 때문에 현재까지도 실시간 영상 또는 카메라 입력 처리 데모용으로 자주 사용되고 있는 모델이다. 이 모델은 당시 일반적인 객체 탐지 모델에 존재했던 영역 제안 단계가 아예 없는 대신 입력을 S x S개의 격자 형태로 분할하고, 분할된 영역을 대상으로 직접 경계 박스와 객체 분류를 수행한 뒤, NMS(Non-Maximum Suppression) 기법을 활용하여 예측한 바운딩 박스 중 정확한 바운딩 박스를 선택하도록 최종 예측 결과의 범위를 좁히는 방식으로 작동한다.

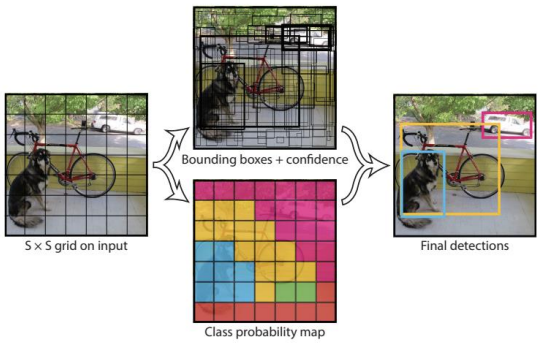

그림 2. YOLO 모델 작동 방식

본 연구에 사용하고자 하는 YOLOv5[3]는 2020년 6월 공개되었는데, 가장 큰 변화는 이전 버전까지의 YOLO 모델은 C언어로 구현된 Darknet을 사용한 반면, YOLOv5부터는 이를 PyTorch로 구현했다는 점이다. 따라서 본 연구에 맞게 코드를 변형하기에 적합하다고 판단하여 YOLOv5를 선택하였다.

2) 밀집도 경보 단계

현재 우리나라에서는 면적 몇 m^2당 몇 명 이상이면 위험 수준으로 판단하여 위기경보를 보낸다는 명확한 기준이 없는 상황이다. 따라서 해외 연구 중 군중의 밀집도(crowd density)에 따른 유동률(crowd flow rate)을 분석한 연구[4]를 토대로 대략적인 기준을 잡아보았다.

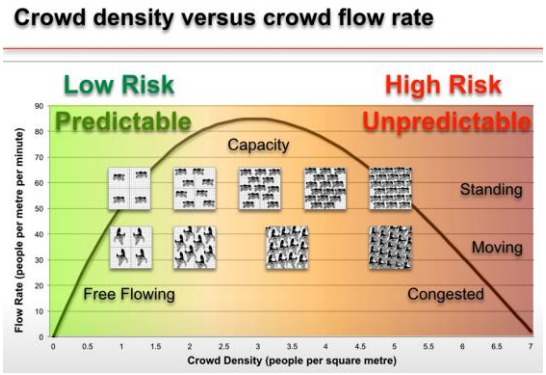

그림 3. 군중 밀집도와 유동률 비교

연구에 따르면 1m^2당 1~4명까지는 안전에 크게 지장이 없지만 5명부터는 군중 간 신체 접촉이 많아지고 6명에 이르면 위험해지기 시작한다고 한다. 따라서 본 연구에서는 1m^2당 사람 수가 5명 이하인 경우 '정상', 5명 초과 6명 이하인 경우 '경고', 그리고 6명 초과부터는 '위험'으로 분류하여 군중의 밀집도를 판단하기로 하였다.

2.2. 데이터셋

학습 데이터셋으로는 coco128, crowdhuman 데이터셋을 사용했다. 학습 데이터의 증강에는 Mosaic, Copy and Paste, Random affine 등의 방법을 적용했다.

테스트 데이터셋으로는 공간의 면적을 알 수 있으면서 충분한 수의 사람이 존재하는 이미지들을 직접 선정하였고, 위 조건을 만족하는 광화문 광장의 명량 분수에 사람들이 모인 사진, 이태원 사고 당시 사진, 횡단보도를 건너가고 있는 사람들 사진, 그리고 교실에 학생들이 모인 사진을 테스트 이미지로 사용했다.

2.3. 밀집도 추정 모델 구현

1) 모델

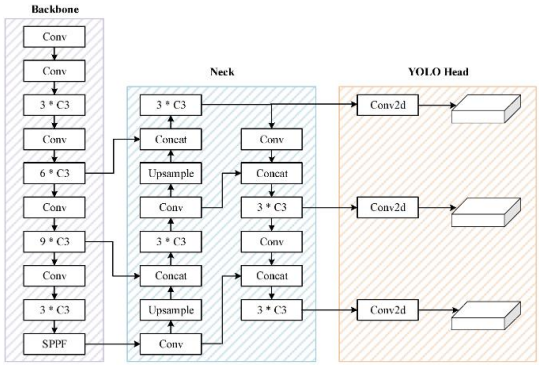

그림 4. YOLOv5 아키텍처

YOLOv5는 기본적으로 Backbone-Neck-Head로 구성된다. Backbone은 이미지로부터 feature를 추출하는 부분으로 Conv 층과 C3 층, SPPF 층으로 이루어져 있는데, 모델을 더 깊게 쌓기 위해 이전 버전에 존재했던 BottleneckCSP 층이 C3 층으로 대체되고 기존 SPP 층과 수학적으로 동일한 결과를 내지만 속도가 더 빠른 SPPF 층을 사용하고 있다. Neck에는 Conv 층과 C3 층뿐만 아니라 이미지를 다시 늘려주는 Upsample 층도 존재한다. Head는 추출된 feature map을 바탕으로 물체의 위치를 찾는 역할을 하며 이전 YOLOv3 및 v4의 Head와 동일하다.

정리하면 전체 모델의 깊이는 초기 YOLO 및 이전 버전에 비해 깊어졌지만 연산이 빠른 층을 적절히 섞어주어 속도와 정확도 측면에서 보다 향상된 모델이 되었다.

2) 학습

YOLOv5에서는 학습을 진행하기 전 학습하고자 하는 데이터셋에 맞는 anchor box를 K-means 알고리즘 등으로 새로 정의하여 사용한다. 그러면 모델의 신경망은 이 anchor box의 비율을 조정하면서 학습을 진행하게 되고, 입력으로 넣은 이미지의 각 격자 셀에 대한 small, medium, large 세 개의 bounding box 좌표 값, 객체 존재 여부, 클래스에 속할 확률을 출력한다.

YOLOv5의 손실함수는 다음과 같다.

$$Loss = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{loc}$$

수식 1. YOLOv5 손실함수

전체 loss를 이루고 있는 요소를 하나씩 살펴보면 L_{loc}은 bounding box에 대한 loss로 GIoU 값에 해당한다. L_{cls}는 객체 탐지가 잘 되었는가에 대한 loss로 MSE와 유사하게 계산되며, L_{obj}는 격자 안의 객체 탐지에 대한 loss를 의미한다.

모든 격자에 대해 위 3가지 loss를 합산하여 전체 loss를 구한다.

2.4. 결과

본 연구에서는 일반적인 YOLOv5에 crowdhuman 데이터셋을 추가로 학습시킨 모델을 사용하였다. 이 모델에 넣은 테스트 이미지들은 하단의 그림 5와 같은 형태로 출력된다.

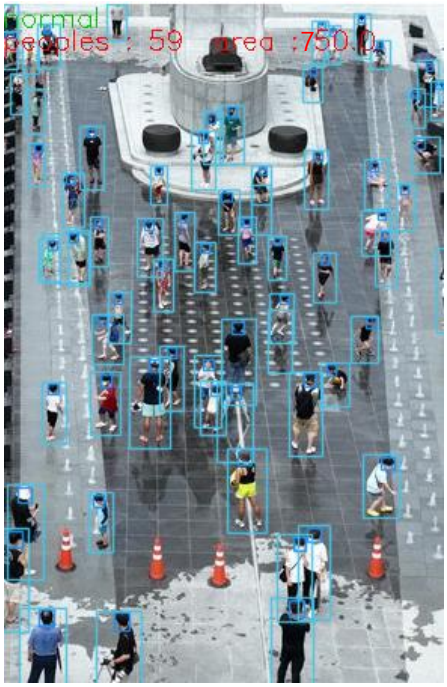

그림 5. "광화문" 이미지 출력 결과

모델의 성능을 비교해보기 위해 테스트 이미지들을 coco128 데이터셋으로 학습한 일반 YOLOv5에도 돌려보았다. 평가 지표로는 실제 밀집도(명/m^2) 값에 대한 각 모델에서 측정된 밀집도 값의 비율을 사용하였고 결과를 정리하면 다음과 같다.

image / model	광화문	이태원	건널목	교실
YOLOv5-crowdhuman	0.922	0.410	1.000	1.000
YOLOv5-coco128	0.344	0.000	0.706	0.694

표 1. 모델 탐지 정확도(accuracy) 비교

표 1의 수치를 보면 crowd human 데이터셋을 추가로 학습시킨 모델이 훨씬 더 좋은 성능을 보이고 있음을 알 수 있다.

3. 결 론

본 연구에서 사용한 모델로 객체 탐지를 수행했을 때 coco128로만 학습한 YOLOv5에 비해 더 나은 성능을 보여줬다. 또한 밀집도를 계산함으로써 현재 얼마나 밀집되어 있는지 수치화 하여 알 수 있고, 그에 따른 안전사고를 미연에 방지할 수 있을 것으로 기대할 수 있다.

4. 후속 연구

현재까지는 면적을 이미 알고 있는 지역에 대해 밀집도를 계산했다. 추후에는 면적을 모르는 곳에서도 밀집도를 계산해야 한다. 2D 이미지 한 장으로 실제 땅의 면적을 추정하기 위해 semantic segmentation 방법을 생각해 볼 수 있다[6]. 하지만 이 방법은 문제점이 있다. 첫 번째로 원근감 문제점, 두 번째로 사물 가림 문제점이다. 거리에 따라 각 pixel 마다의 이미지 측정 단위가 달라지므로 원근감 문제를 가진다. 또한 충분히 사람을 수용 가능한 면적이 있는데도 불구하고 2D 이미지 상에서 뒤 공간을 가리는 큰 물체가 있다면 해당 부분은 수용 가능한 면적으로 분류되지 않는다.

따라서 후속 연구 방향으로 BEV(Bird Eye View) 데이터의 사용을 제안한다. 이때 사용할 수 있는 데이터는 카메라의 매개변수 초점거리(focal length)와 화각(field of view)를 알고 있는 완벽한 BEV 이미지이다. 이것을 이용해 이미지의 실제 크기 구할 수 있고, segmentation 하여 땅의 면적을 구한다. 또는 Point cloud를 BEV 이미지로 전처리 후 사용한다. Point cloud 데이터는 2대 이상 또는 회전하는 카메라, fish eye lens 등을 이용해서 얻는다.

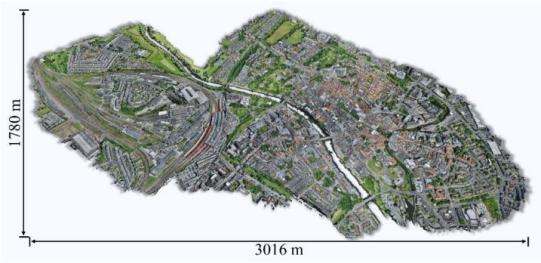

그림 6. 도시 경관에 대한 Point Cloud

그림 6과 같이 일정한 부피의 도시 경관에 대한 point cloud와 segmentation label 값이 있는 SensatUrban[7] 데이터가 있다.

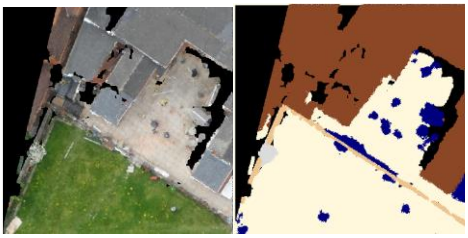

그림 7. Point Cloud 전처리 이미지

그림 7과 같이 3D point cloud를 전처리해서 25m x 25m의 완벽한 BEV 이미지를 구한다. 이렇게 전처리된 이미지를 학습 데이터로 사용한다[8].

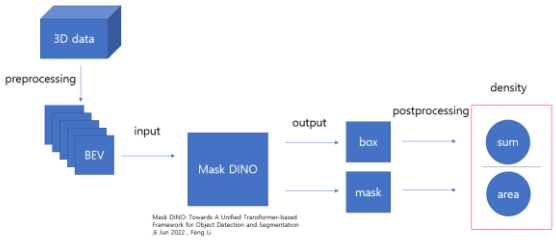

그림 8. 후속 연구 예상 구조도

그림 8에 보이는 과정에 따라 모델은 MaskDINO[9]를 사용한다. MaskDINO은 이미지를 입력으로 받고 출력층은 segmentation된 mask와 object detection이 된 ROI (Region Of Image)가 있다. 이것을 후처리 하여 면적과 인구수를 바탕으로 밀집도를 계산할 수 있다.

위와 같은 후속 연구를 통해 향후에는 면적에 대한 정보 없이 밀집도를 구할 수 있을 것이라고 예상된다.

참고 문헌

[1] 전국매일신문, "인파 밀집 AI로 위험감지...위기경보 안내", 2022.
[2] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detecion", 2016.
[3] YOLOv5 Github Link: https://github.com/ultralytics/yolov5
[4] G. Keith Still, "Crowd Safety and Crowd Risk Analysis", 2018.
[5] 이미지 출처: 동아일보, "탁 트인 광화문광장, 분수도 인기...녹지-그늘 부족은 아쉬워", 2022.
[6] Wang, Panqu, et al. "Understanding convolution for semantic segmentation." 2018 IEEE winter conference on applications of computer vision (WACV). Ieee, 2018.
[7] Hu, Qingyong, et al. "Towards semantic segmentation of urban-scale 3D point clouds: A

dataset, benchmarks and challenges." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.
[8] Zou, Zhenhong, and Yizhe Li. "Efficient Urban-scale Point Clouds Segmentation with BEV Projection.", 2021.
[9] Li, Feng, et al. "Mask dino: Towards a unified transformer-based framework for object detection and segmentation.", 2022.