

yolo모델을 활용한 기차 내 혼잡도 분석 시스템

A Congestion Measurement System Using YOLO Object Detection Model

김진성, 이민형, 홍진하, 유동영, 홍익대학교 소프트웨어융합학과
정영훈 기계정보공학과

Dept. of Computer & Information Communications Engineering,
Hongik University

Dept. of Software & Communications Engineering, Hongik University

요약

본 논문은 사람과 짐의 물체 감지를 위한 YOLO 모델을 활용하여 특정 공간 내 혼잡도를 측정하는 시스템을 제안한다. YOLO를 학습시켜 기차 내에서 사람 및 짐과 같은 객체를 탐지하는 모델을 만든다. 그리고 이 모델을 이용하여 기차 내에서 객체를 탐지하고, 객체의 위치와 개수 정보를 얻는다. 이렇게 얻은 정보를 기반으로, 혼잡도를 측정하기 위해 다양한 지표를 활용한다. 이를 인터페이스에 시각적으로 보여준다. 결과적으로, 제안된 시스템은 승객들의 안전과 편의를 보장하며, 특정 공간의 혼잡도 파악에 유용한 도구로 사용될 수 있다.

I. 서론

[5]철도통계연보에 따르면, 2020년에는 COVID-19 팬데믹으로 인해 여행 인원 감소와 비대면 수업 등의 영향으로 정기권 사용 인원이 소폭 감소하였지만, 2021년에는 규제가 완화되면서 정기권 사용 인원이 다시 증가하는 모습을 보이고 있다.

정기권을 이용하는 사람들은 혼잡한 상황에서 이동해야 하는 경우가 많다. 이러한 상황에서 정기권 승객들은 역내 기차 혼잡도를 파악하지 못하여 자유석을 찾아 헤매는 경우가 있다. 또한, 기차 내부에 자유석 정보가 없어 특정 열차 칸에만 사람들이 몰리는 현상이 발생한다. 이러한 문제점을 해결하기 위해, 본 논문에서는 YOLO 모델을 활용하여 특정 공간 내 혼잡도를 측정하는 시스템을 제안한다.

II. 관련 연구

객체를 탐지하는 방식에는 대표적으로 R-CNN 계열 방식과 YOLO 계열 방식이 있다. R-CNN 계열 방식은 이미지 내 객체를 탐지하기 위해 먼저 후보 영역(proposal)을 추출하고, 추출된 후보 영역에 대해 딥러닝 모델을 적용하여 객체를 탐지한다. 이 방식은 높은 정확도를

보이지만, 처리 속도가 느리며 실시간 적용이 어렵다.

[6]YOLO 계열 방식은 이미지 전체를 한 번에 처리하여 객체를 탐지하는 방식이다. 이미지를 그리드로 나눈 후, 각 그리드 셀에 대해 딥러닝 모델을 적용하여 객체의 존재 여부와 위치, 크기 등의 정보를 추출한다. 이 방식은 실시간으로 적용이 가능하며, 처리 속도가 빠르다는 장점이 있다. YOLO 모델은 R-CNN 계열 방식과 달리 이미지에서 후보군을 추출할 필요가 없기 때문에 정확도가 조금 낮을 수 있지만, 적절한 학습모델을 제공한다면 R-CNN 계열보다 높은 정확도를 나타낼 수 있다.

[7]따라서 기차 내부와 같이 실시간으로 혼잡도를 파악해야 하는 상황에서는 YOLO 계열 방식이 더욱 적합할 것으로 판단된다. YOLO를 기반으로 한 물체 탐지 모델을 이용하여 기차 내부의 혼잡도를 파악하는 시스템을 구축하면, 높은 정확도와 빠른 처리 속도로 승객들의 안전과 편의를 보장할 수 있을 것이다.

III. 탐지 모델 설계

3.1 YOLO 모델 설정

[1]기차 내 혼잡도를 예측하기 위해 YOLOv5 모델을 이용하는 것이 좋다. 전체 이미지를 한 번에 처리하는 YOLOv5는 R-CNN보다 높은 속도와 정확도를 보이며,

입력 데이터를 전체적으로 이해하는 능력이 높아 Background Error가 낮아지는 이점이 있다. 표1을 보면 [2][3]GPU RTX 2060인 환경에서 EfficientDet 와 YOLOv5의 세부 모델들을 비교한 표이다.

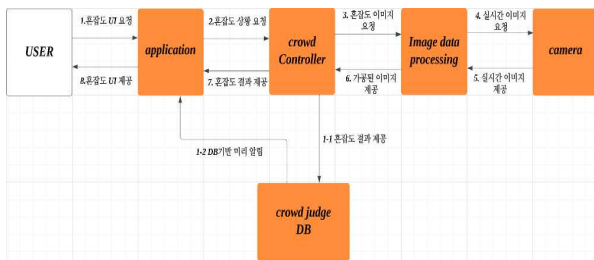
<표 1> EfficientDet VS YOLOv5

Model	mAP	FPS
EfficientDet-D7x	53.7	1.5
YOLOv5-YOLOv5x	49.2	17.0
EfficientDet-D4	48.8	4.5
YOLOv5-YOLOv5l	47.7	20.0
YOLOv5-YOLOv5m	44.3	27.0
YOLOv5-YOLOv5s	37.0	30.0

EfficientDet-D7x가 FPS가 가장 낮지만 가장 높은 mAP를 보인다. YOLOv5 모델은 초당 프레임 수가 성능에 큰 영향을 주지 않는다는 연구 결과가 나와, YOLOv5-YOLOv5x에서 가장 우수한 성능을 보여주었다. 이를 토대로 YOLOv5 모델이 최적화된 딥러닝 모델임을 확인할 수 있다. 따라서 YOLOv5 모델을 이용하여 영상을 실시간으로 처리하여 기차 내 혼잡도를 파악하는 작업을 수행할 것이다.

[4]YOLO 모델은 학습에 사용되는 데이터의 다양성과 양이 모델의 성능에 큰 영향을 미친다는 것을 밝혀냈다. 특히, 기차 내부와 같은 특정한 공간에서는 사람과 짐과 같은 객체를 정확하게 탐지하는 것이 중요하다. 따라서, 사람과 짐에 대한 이미지 학습을 중점으로 하여 보다 정확한 모델구현이 가능하다.

3.2 모델 설계도



<그림 1> 모델 설계도

User가 Application에 혼잡도를 요청하면, Application에서 혼잡도 상황을 Crowd Controller에 요청한다. 혼잡도 분석을 위해 Camera에서 보낸 실시간 이미지를 Image data processing을 통해 가공한 뒤 Crowd Controller에서 혼잡도를 분석한다. 분석한 결과를 Crowd Judge DB에 제공하고 DB기반 알림을 Application에 전달하여 최종적으로 User에게 혼잡도 분석 결과를 시각적으로 제공한다.

IV. 결론

본 논문에서는 열차 내부의 객체 감지를 통해 혼잡도를 파악하고 시각적으로 보여주는 시스템을 개발하였다. 해당 시스템은 객체 탐지에 특화되어 높은 정확도로 혼잡도를 파악할 수 있어, 해당 열차에 탑승하는 사람들의 효율적인 분배로 열차 탑승 시간을 감소시킬 수 있다.

그러나, 가변적인 공간에서는 객체 탐지의 정확도가 떨어질 수 있기 때문에 보다 넓은 공간과 다양한 객체를 정확하게 탐지할 수 있도록 모델을 학습시키는 연구가 필요하다는 문제점이 있다. 따라서, 향후 연구에서는 모델을 다양한 공간에 적용하여 높은 정확성을 가진 탐지 모델을 탐구하는 것이 필요하다. 이러한 연구를 통해 객체 탐지와 혼잡도 파악이 필요한 다양한 분야에서 해당 모델을 활용할 수 있을 것이다.

참고문헌

[1] Joseph Redmon, Ali Farhadi, "YOLO9000:Better, Faster, Stronger", University of Washington, Allen Institute for AI, arXiv:1612.08242v1 [cs.CV] , 25 Dec 2016

[2] Table. 1 Comparson of efficientdet vs yolov5 장성진, 장종욱, 차량탑승인원 탐지를 위한 딥러닝 영상처리 기술 연구, 한국정보통신학회논문지, Vol. 25 No. 8, p1026~1031, 2021

[3] D. Dubey and G. S. Tomar, "Echelon Based Pose Generalization of Facial Images Approaches," Asia-pacific Journal of Convergent Research Interchange, vol. 3, no. 1, pp. 63-75, Mar. 2017.

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", University of Washington, Allen Institute for AI, Facebook AI Research, arXiv:1506.02640v5 [cs.CV] 9 May 2016

[5] 2021_철도통계연보(PDF), 국토교통부, 한국철도공사, p32-33, 10 2022

[6] 이용환, 김영섭, "객체 감지를 위한 CNN과 YOLO의 비교", 반도체디스플레이기술학회지, 19권 1호, p 85 - 92, 2020

[7] 장성진, 장종욱, 차량탑승인원 탐지를 위한 딥러닝 영상처리 기술 연구, 한국정보통신학회논문지, Vol. 25 No. 8, p1026~1031, 2021