YOLO 경량 모델 성능 비교 연구

정예은, 정민영
안산대학교

okid04@naver.com, myjung@ansan.ac.kr

Performance Comparison of Lightweight

YeEun Jung , MinYoung Jung
Ansan Univ.

요 약

본 연구는 자율주행 환경에서 요구되는 실시간 객체 인식 성능을 충족하기 위해 경량 YOLO 모델의 구조적 효율성과 CPU 기반 추론 성능을 비교·분석하였다. 기존 연구는 주로 GPU 환경 또는 전체 처리 시간에 의존하여 모델을 평가함으로써, 엣지 디바이스 특유의 제한된 연산 자원과 폴백 상황을 충분히 반영하지 못하는 한계가 있었다. 이에 본 연구는 GPU 가속이 불가능한 환경을 가정하고, YOLOv5n과 YOLOv8n을 동일 조건에서 실행하여 순수 추론 시간만을 측정하였다. 이러한 분석을 바탕으로 본 연구는 자율주행용 엣지 컴퓨팅 환경에서 모델 선택 시 정확도와 속도 간의 균형을 판단할 수 있는 정량적 기준을 제시하고, 안전성 중심의 경량 모델 도입에 실질적인 근거를 제공하고자 한다.

Ⅰ. 서 론

최근 자율주행 기술은 인공지능 기반의 인식·판단·제어 기술의 발전과 함께 고도화되고 있다. 자율주행 차량은 보행자, 차량, 신호등 등을 실시간으로 탐지해야 하며, 이러한 객체 인식 성능은 탑승자의 안전과 직결된다. 특히 자율주행 차량에 탑재되는 엣지 컴퓨팅(Edge Computing) 유닛은 공간적·전력적 제약으로 인해 고성능 GPU를 상시 가동하기 어려울 수 있으며, 시스템 오작동 시 안전 상태(Safe State)로 전환하기 위해 최소한의 연산 자원(CPU 등)만으로 구동되는 안전비상 시스템(Fallback System)이 필수적으로 요구된다. 자율주행 시스템에서 센서 데이터의 처리 지연(Latency)은 차량의 제동 거리(Stopping Distance)와 직결된다. 특히 GPU 장애 상황을 대비하여 CPU만으로 구동되는 환경에서도 최소한의 안전을 보장할 수 있는 프레임 레이트(FPS) 확보가 필수적이다.[1-2]

Ⅱ. 본론

2.1 실험 환경 및 데이터셋

본 연구는 자율주행 엣지 디바이스의 연산 자원 제약 상황을 시뮬레이션하기 위해 Google Colab의 CPU-only 환경에서 실험을 진행하였다. 이는 GPU 가속기가 고장 나거나 가용하지 않은 최악의 상황(Worst-case Scenario)을 가정한 것이다.

- 하드웨어: Intel Xeon CPU (2.20GHz), RAM 12GB
- 소프트웨어: Ubuntu 22.04, Python 3.10, PyTorch 2.0.1 (CPU version)
- 모델 버전: YOLOv5 (v7.0), YOLOv8 (Ultralytics 8.x)

2.2 자율주행 최적화를 위한 실험 설정

자율주행 시스템의 실시간성을 평가하기 위해 전처리 및 후처리 시간을 포함한 전체 파이프라인 시간이 아닌, 모델 자체의 연산 부하를 나타내는 순수 추론 시간(Inference Time)을 중점적으로 측정하였다. 입력 이미지 크기는 자율주행 객체 인식에서 널리 사용되는 640x640 해상도로 설정하였다.

2.3 YOLO모델 성능평가 방법

두 모델 모두 동일한 하이퍼파라미터(epochs 20, batch 8, img size 640)를 적용하여 학습하였다. 추론 성능의 신뢰성을 확보하기 위해 Warm-up 10회 수행 후, 테스트 이미지에 대해 100회 반복 추론하여 평균 지연 시간(Latency)과 표준편차(Standard Deviation)를 측정하였다.

Ⅲ. 결론

데이터 분석 결과, YOLOv8n은 YOLOv5n 대비 추론 속도가 약 24.8% 저하되었으며, 모델 크기(가중치 용량) 또한 60.3% 증가하였다. 특히 주목할 점은 표준편차(Standard Deviation)이다. YOLOv8n의 표준편차(71.13ms)가 YOLOv5n(36.88ms)보다 2배 가까이 큰 것은, C2f 모듈의 복잡한 연산 그래프가 CPU 스케줄링에 불규칙한 부하를 주고 있음을 시사한다. 이는 실시간성이 엄격하게 보장되어야 하는 자율주행 시스템(Hard Real-time System)에서 지터(Jitter) 문제를 야기하여 제어 안정성을 해칠 수 있는 위험 요소이다. 실험결과는 아래 표와 같다.

Table 1. Performance comparison of YOLOv5n and YOLOv8n on CPU environment

Model	Release Year	Inference Time (ms)	Std Dev (ms)	Model Size (MB)	FBS
YOLOv5n	2021	187.34	36.88	3.88	5.34
YOLOv8n	2023	233.84	71.13	6.22	4.28

참 고 문 헌

[1]Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, “Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing”, Proceedings of the IEEE, Vol. 107, No. 8, pp. 1738-1762, Aug. 2019.DOI: https://doi.org/10.1109/JPROC.2019.2918951

[2]J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 779-788, June 2016.