자율주행 영상처리 알고리즘 경량화에 따른 SW·HW 성능 비교

김조아[1], 박수빈[2], 오다빈[3], 이주리[4]
[1]이화여자대학교 전자전기공학과 학부생
[2]숭실대학교 전자정보공학부 학부생
[3]서울시립대학교 통계학과 학부생
[4]동덕여자대학교 정보통계학과 학부생
hellojoa1202@ewha.ac.kr, psb21@soongsil.ac.kr, starlight2123@uos.ac.kr,
20210857@dongduk.ac.kr

SW/HW Performance Analysis of Lightweight Deep Learning Models for Autonomous Driving

Joa Kim[1], Subin Park[2], Dabin Oh[3], Juri Lee[4]
[1]Dept. of Electronic and Electric Engineering, Ewha Womans University
[2]Dept. of Electrical Engineering, Soogsil University
[3]Dept. of Statistics, University of Seoul
[4]Dept. of Statistics and Information Science, Dongduk Women's University

요 약

본 연구는 라즈베리파이 기반 자율주행 키트에서 차선 인식과 사물 인식 모델을 통합하여 Baseline을 구축하고, 다양한 경량화 기법을 적용해 성능을 비교·분석하였다. 그 결과, 일부 정확도 저하가 있었으나 모델 크기와 추론 속도 측면에서 효율성이 개선되어 제한된 임베디드 환경에서도 실시간 적용 가능성을 확인하였다.

1. 서론

자율주행 시스템의 핵심은 차선 및 객체를 빠르고 정확하게 인식하는 데 있다. 이를 위해 다양한 딥러닝 모델이 활용되고 있지만, 대부분은 복잡한 연산 구조와 높은 자원 소모로 인해 임베디드 보드와 같은 제한된 환경에서 실시간 동작이 어렵다.

이에 따라 경량화 기법을 적용하여 연산량을 줄이고 효율성을 높이는 연구가 활발히 진행되고 있다. 그러나 실제 주행 환경에서 SW 성능 지표(F1-score, mAP, FPS)와 HW 성능 지표(CPU 사용률, 추론 시간, 주행 안정성)을 동시에 종합적으로 평가한 사례는 드물다. [1]

본 연구는 자율주행 알고리즘에 경량화를 적용하고, 소프트웨어적 관점과 하드웨어적 관점에서 그 효과를 비교·분석하여 실제 임베디드 환경에서의 적용 가능성을 검증하는 것을 목표로 한다.

2. 관련 연구

자율주행 인식 알고리즘은 차선 인식과 사물 인식으로 구분된다. 차선 인식은 영상에서 차선을 검출하여 조향에 활용하며 CNN 기반 End-to-End 또는 Segmentation 방식이 주로 사용된다. 사물 인식은 보행자·신호등 등 주변 객체를 탐지해 주행 의사결정으로 연결되며 SSD·YOLO 계열 모델이 대표적이다. 다만 이들 모델은 높은 정확도 대비 연산량이 크기 때문에 임베디드 환경에서 실시간 동작이 어렵다는 한계가 있다.

이를 해결하기 위해 다양한 경량화 기법이 활용되고 있으며, Pruning과 Quantization은 모델 크기와 연산량 절감에 초점을 둔다. Model Folding은 연산 경로를 단순화하여 추론 지연을 줄이고, Distillation은 경량화 과정에서 발생할 수 있는 성능 저하를 보완하는 방식으로 적용된다. [2]

그러나 기존 연구들은 대부분 정확도나 FPS와 같은 단일 성능 지표에 초점을 맞추고 있어, 실제 임베디드 보드 환경에서 SW 및 HW 지표를 종합적으로 분석 및 평가한 사례는 상대적으로 부족하다

3. 연구 방법

본 연구는 라즈베리파이5 기반 자율주행 키트를 활용하여 수행되었으며, 실험은 직접 제작한 약 2×3m 크기의 트랙에서 진행하였다. 데이터셋은 차선 인식을 위해 직접 수집·라벨링한 약 18,000장의 영상 데이터를 사용하였고, 사물 인식을 위해 COCO 데이

<표2> 경량화 기법 적용에 따른 SW/HW 성능 및 안정성 종합 평가 결과

Model (Baseline= UFLD + EfficientDet-Lite0)	SW					HW (라즈베리파이5)		
	F1-score (%)	mAP50 (%)	FPS (frame/s)	학습 시간 (초/epoch)	모델 크기 (MB)	추론 시간 (ms)	CPU 사용률 (%)	주행 안정성 (오류 횟수)
Baseline	81.5	52	220	240	20.0	95	95	18
P40-U (Pruning Only)	81.9	51.8	250	238	16.0	85	85	15
P60-U (Pruning Only)	81.7	51.5	280	235	13.5	82	80	10
Q-FP16 (Quant Only)	82.1	50.5	240	240	12.0	81	75	12
Model Folding	81.8	50.7	310	240	19.8	72	70	8
KD-Reg (Distillation Only)	81.2	53.0	230	265	20.0	90	92	16
P40-U + Q-FP16	82.6	51.5	325	240	16.0	68	65	5
KD-Reg + Model Folding	83.0	52.5	370	260	19.8	65	76	7

터셋을 활용하였다. 모든 학습은 epoch=30, lr=1e-3으로 설정하여 진행하였다.

차선 인식 모델과 객체 인식 모델을 1차 평가한 뒤, 각각 정확도와 모델 크기가 큰 특성을 고려한 모델 사이즈 대비 정확도(mAP50/MB)를 기준으로 최종 모델을 선정하였다.

성능 평가는 두 가지 관점에서 수행하였다.

- 소프트웨어 지표: F1-score(차선 인식), mAP(사물 인식), FPS, 학습 시간, 모델 크기
- 하드웨어 지표: 추론 시간, CPU/RAM 사용률, 주행 안정성(3초 이상 지연 또는 객체 인식 실패 시 오류 판정)

4. 연구 결과 및 분석

4.1 모델 선정 결과

표1의 결과는 차선인식에서는 UFLD, 사물인식에서는 EfficientDet-Lite0 모델이 가장 우수하였다.

<표1> 차선인식 및 사물인식 모델의 1차 성능 평가 결과

구분	차선 인식 (Accuracy)	구분	사물 인식 (mAP50/MB)
PilotNet	54.6%	YOLOv5n	6.829
SCNN	65.5%	SSD-MobileNetV2	0.316
UFLD	81.5%	NanoNet	5.014
UNet	71.3%	EfficientDet-Lite0	7.174

4.2 경량화 전후 성능 비교 결과

표 2는 1차 평가에서 선정된 UFLD(차선 인식)과 EfficientDet-Lite0(사물 인식)을 베이스라인으로 하여, 단일 및 조합 형태의 경량화 기법을 적용했을 때의 변화를 비교한 결과이다. 단일 기법은 각 경량화 기술의 효과를 개별적으로 확인하기 위한 목적이며, 조합 기법은 정확도 저하를 최소화하면서 효율성 극대화 여부를 검증하기 위해 구성되었다.

F1-score는 차선 인식에서, mAP는 사물 인식에서 측정하였으며, 하드웨어 지표는 동일 조건에서 5회 주행 평균값을 사용하였다. 실험 결과, 대부분의 모델에서 추론 시간이 단축되고 CPU 사용률이 감소하였으며, 일부 조합에서는 주행 안정성 개선도 확인되었다. 다만 일부 설정에서는 정확도가 소폭 감소하는 등 성능-효율 간의 트레이드오프가 존재하였다. 전반적으로 경량화는 임베디드 환경에서의 실시간 적용 가능성을 높이는 데 기여함을 확인할 수 있었다

5. 결론

본 연구는 라즈베리파이 기반 임베디드 환경에서 경량화 기법이 자율주행 인식 모델의 성능에 미치는 영향을 SW/HW 관점에서 종합적으로 비교·분석했다는 점에서 의의가 있다. 실험 결과, 경량화를 통해 모델 효율성이 개선되고 주행 안정성이 향상되는 등 실시간 적용 가능성이 확인되었다. 다만 강도 높은 경량화에서는 정확도가 일부 감소하여 성능과 효율 간의 균형이 필요함도 확인되었다. 향후 연구에서는 추가적인 경량화 조합 실험과 더 다양한 임베디드 환경 비교를 통해 재현성과 일반화 가능성을 강화할 예정이다.

※ 본 논문은 과학기술정보통신부 대학디지털교육역량강화사업의 지원을 통해 수행한 한이음 드림업 프로젝트 결과물입니다."

참고문헌

[1] G. H. Hwang, H. B. Oh, and J. W. Jeon, "How Lightweight Deep Learning Enhances Performance in DPU-Accelerated Autonomous Driving on Zynq SoC," IEEE, 2022.

[2] Lee, Han-Haesol. "A Study on Optimization of Light-Weight Deep Learning for Object Detection."