

임베디드 환경에서의 안드로이드 OS기반 실시간 영상 AI 객체인식 및 실 주행 적용/평가

안재홍, 손권욱*, 정광현, 김상유, 주재영

한국광기술원

jhng834@kopti.re.kr, *kwson.emc@yu.ac.kr, randy105@kopti.re.kr, sangyoo.kim@kopti.re.kr, jyjoo@kopti.re.kr

Android OS-based real-time image AI object detection and real driving application/evaluation in embedded environment

An Je hong, Son Kwon wook*, Jung Kwang hyun, Kim Sang yoo, Joo Je young
Korea Photonics Technology Institute(KOPTI), *Yeunnam University

요약

본 논문은 임베디드 보드환경에서의 안드로이드 운영체제 기반 실시간 카메라 영상 객체인식과 실제 주행 차량에 적용하여 인식 정도를 평가하는 논문이다. 임베디드 보드는 차량용 셋탑박스 및 운전자 통합 인터페이스 전문기업인 모토모사의 안드로이드 OS기반 모토로이드 보드를 활용하였으며, 안드로이드 OS에서의 딥러닝 기반 객체인식을 위해 YOLOv4, YOLOv4-tiny 기술을 응용하였다. 또한 딥러닝 모델 학습을 위한 이미지 데이터 세트 중, cocodataset에서 자율주행 시 필요한 데이터만을 추출하여 사용하였다. 실 주행 적용 및 평가를 위해 차량 내의 장착된 카메라 및 디스플레이를 활용하였으며, 평가 환경은 소속기관 주변 도로로 선정하였다. 적용/평가결과 YOLOv4는 속도는 느리지만 인식성능이 높았으며, YOLOv4-tiny는 속도는 빠르지만 인식성능이 YOLOv4에 비해 낮은 것을 확인할 수 있었다.

I. 서론

딥러닝 기반의 객체인식의 경우 현재까지 많은 결과를 보이고 있으나, 그에 비해 실제 주행 환경에서의 적용과 기술에 따른 비교 관련 연구는 크게 두각을 나타내고 있지 않은 실정이다. 그러므로 본 논문에서는 제한적인 임베디드 보드 환경에서의 안드로이드 OS기반으로 실시간 영상을 통한 딥러닝 객체인식과 이를 실 주행에 적용하여, 인식정도와 기술 비교 평가를 구현한다.

II. 본론

본론은 임베디드 보드, 데이터 수집 및 추출, 안드로이드 OS기반 AI 객체인식, 실 주행 적용/평가 총 4개로 분류하였다.

2.1 임베디드 보드

임베디드 보드는 차량용 셋탑박스 전문기업인 모토모 제품을 활용하였으며, 본 제품은 안드로이드 운영체제 기반으로 통합 운전자 인터페이스 제품이다.



그림 1. 사용된 임베디드 보드 예

2.2 데이터 수집 및 추출

객체인식 시, 필요한 모델을 학습하기 위한 데이터로 오픈된 데이터세트인 COCOdataset[2]를 활용하였다. COCOdataset[2]는 총 80개의 클래스와 약 33만장의 이미지로 이루어져 있으며, 현재 다중 객체 인식 시, 가장 많이 쓰이는 이미지 데이터 세트 중 하나이다. 이 중, 자율주행 및 ADAS 환경(차량, 보행자, 자전거, 바이크 등)에서의 필요한 데이터 클래스 선정 후, 클래스에 속한 이미지와 이미지 정보를 텍스트 한 파일만을 추출하여 활용하였다.

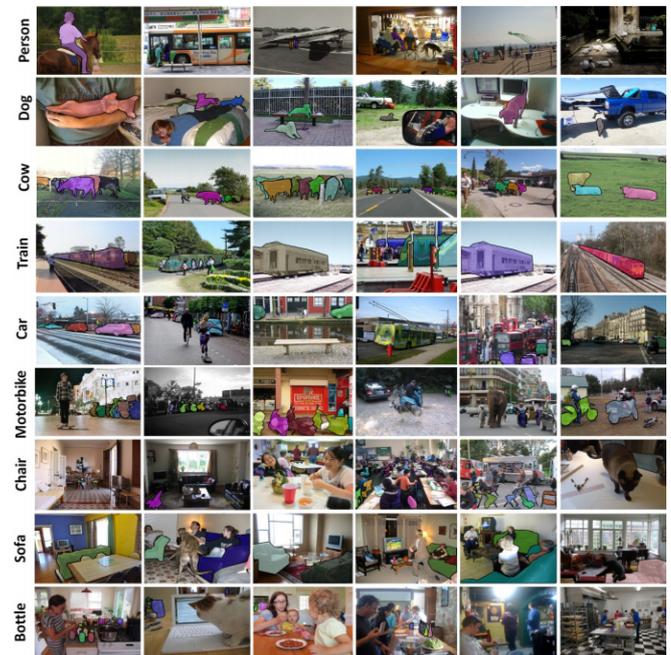


그림 2. COCOdataset 예

2.3 안드로이드 OS기반 AI 객체인식

임베디드 보드 운영체제 환경에 따라 안드로이드 운영체제에서 구현하였으며, AI 객체인식 기법 중, 유명기법인 YOLO[1]를 활용하였다. YOLO[1]는 바운딩 박스 좌표와 분류를 동일한 신경망 구조와 동시에 실행하는 통합인식으로 처리 속도와 인식률을 높임으로써, 실시간 구현환경에 적합하다.

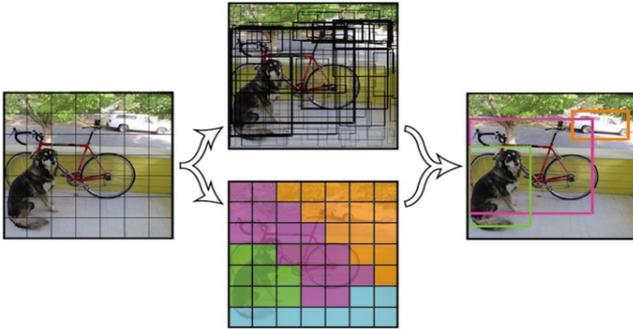


그림 3. YOLO 인식과정

본 연구에서는 다양한 YOLO 시리즈 중, YOLOv4[1]와 YOLOv4-tiny를 적용하였으며, YOLOv4[1]의 경우 기존의 YOLO 기술보다 동일한 환경에서의 빠른 학습으로 시간 대비 효율성이 좋다. YOLOv4[1]와 YOLOv4-tiny의 차이점은 아키텍처 모델의 크기로 YOLOv4-tiny가 YOLOv4[1]에 비해 크기가 작다.

아래 그림 4는 GPU서버(1080ti) 기준으로 세로축은 COCOdataset 베이스로 한 인식 성능과 가로축은 초당 프레임이며, 모델 크기 차이에 의한 인식 성능 및 연산 속도 차이점을 보여준다.

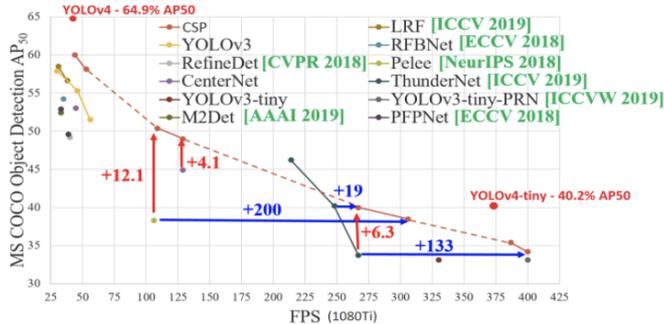


그림 4. YOLOv4, YOLOv4-tiny 비교 그래프

2.4 실 주행 적용/평가

실 주행 환경은 소속기관인 한국광기술원 주변으로 선정하였으며, 평가는 YOLOv4[1]와 YOLOv4-tiny 총 2가지의 기술을 구현한 후, 속도 및 인식성능 정도를 확인하였다. 실 주행 적용결과는 다음과 같다.



그림 5. YOLOv4 실 주행 적용 결과



그림 6. YOLOv4-tiny 실 주행 적용 결과

III. 결론

안드로이드 운영체제 임베디드 환경에서의 AI 객체인식 알고리즘 (YOLOv4[1], YOLOv4-tiny)을 실 주행 적용결과 YOLOv4[1]는 그림 4와 같이 멀리보이는 객체에 대한 인식과 인식 정도가 높으나, 임베디드 환경에서 적용 시, 프레임 수가 낮음을 확인할 수 있었다. YOLOv4-tiny는 멀리보이는 객체와 인식 정도가 낮으나, YOLOv4[1]에 비해 프레임 수가 높은 것을 확인할 수 있었다.

ACKNOWLEDGMENT

본 논문은 2022년 중소기업기술개발지원사업으로 지원된 연구결과입니다(S3182829, 객체인식 카메라를 활용한 안전 운전 시스템을 탑재한 통합형 안드로이드 OS 셋톱박스)

참고 문헌

- [1] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection", arXiv:2004.10934v1 [cs.CV] 23 Apr 2020
- [2] Tsung-Yi Lin, Michael Maire, Serge Belongie, etc "Microsoft COCO: Common Objects in Context" arXiv:1405.0312v3 [cs.CV] 21 Feb 2015