

YOLOv3-Tiny 를 이용한 자율주행 차량의 객체 탐지 시스템과 SNNs 변환

허정윤, 임재한*
광운대학교, *광운대학교

jylemon1128@naver.com, *ljhar@kw.ac.kr

Object detection system and SNNs conversion for autonomous vehicles using YOLOv3-Tiny

Jeong-Yun Heo, Jae-Han Lim*
Kwangwoon Univ., * Kwangwoon Univ.

요 약

자율주행에서 중요한 기술 중 하나인 객체 탐지에 대한 연구가 활발히 이루어지고 있다. 그러나 광범위한 클래스에 대한 객체 탐지나 우주 공간 같이 특수한 상황에서의 객체 탐지에 대한 연구는 진행되어 왔으나, 자율주행 차량만을 위한 연구는 부족하다. 그렇기에 본 논문에서는 자율주행 차량을 위한 Vehicle-YOLOv3-Tiny 를 제시한다. 제시한 모델은 여러 성능 지표에서 좋은 성능을 보였다. 또한 YOLO 와 같이 깊은 신경망 모델의 단점인 높은 전력 소모를 극복하기 위해 SNNs 로 변환하는 실험을 진행하였다. 그 결과 변환한 모델이 낮은 성능을 보이는 이유로 2 가지의 정보 손실 원인을 분석하여 제시하였다.

I. 서론

자율주행 기술이 탑재된 차량의 상용화가 이루어짐에 따라서 관련 연구가 활발하게 진행되고 있다. 그 중에서 주행 상황 중 객체를 탐지하고 추출하는 것은 가장 기본적이고 중요한 사항이다[1]. 객체 탐지 및 추출을 위해 많은 연구에서 심층 신경망(deep neural networks, DNNs)과 합성곱 신경망(convolution neural networks, CNNs)을 사용하고 있다[2][3]. 그러나 DNNs 과 CNNs 을 기반으로 하는 신경망 구조는 매우 높은 연산량과 전력 소모량을 보인다. 이를 해결하기 위해서 3 세대 신경망인 스파이킹 신경망(spiking neural networks, SNNs)이 높은 주목을 받고 있다. SNNs 는 event-driven 한 특징으로 저전력으로 작동한다는 장점을 갖는다. 그러나 SNNs 는 미분할 수 없는 스파이크로 작동하기 때문에 훈련하기가 어려워, 깊은 신경망을 필요로 하지 않는 간단한 작업에 치중된 경향이 있다[4]. 이를 해결하기 위해 제안되는 방법 중 하나는 DNNs 을 SNNs 로 변환하는 방법이다. DNNs to SNNs 는 DNNs 에서 가중치와 같은 매개변수를 학습하고 이를 적절히 변환하여 SNNs 에 적용하는 것이다. 이러한 방법을 사용하면 SNNs 로 깊은 신경망을 구축하여 복잡한 문제를 해결할 수 있다. 최근에는 객체 탐지 알고리즘 중 하나인 YOLO 에 DNNs to SNNs 방식을 적용한 방법이 제안되었다[4][5]. 그러나 제안된 방법들은 일반적인 객체 탐지와 우주 공간에서의 객체 탐지를 하기 때문에 자율 주행 차량에서 사용하기에는 어려움이 있다.

따라서 본 논문에서는 자율 주행 차량에서의 사용을 위한 Vehicle-YOLOv3-Tiny 를 제안하고 성능을 측정한다. 그리고 차후 연구를 위하여, 학습이 완료된 모델을 SNNs 로 변환시킨 후 성능을 측정하고 문제점 및 개선 사항에 대하여 제시한다.

II. 배경 지식

2.1 You Only Look Once V3

YOLO 는 객체 탐지를 위한 딥러닝 모델 중 하나로 bounding box 를 통해 객체의 위치를 특정하고 분류(classification)를 수행한다. YOLOv3 는 convolution 계층을 53 개 사용하여 YOLOv2 에 비해 깊은 네트워크 구조를 갖는다. 그럼에도 불구하고 높은 성능과 빠른 추론 시간을 보인다[6].

2.2 Spiking Neural Networks

SNNs 는 사람의 뇌의 작동방식을 모방한 3 세대 인공신경망으로 그림 1 과 같이 스파이킹 뉴런과 시냅스로 이루어져 있다. $P_{l,i}$ 는 l 번째 계층의 i 번째 시냅스 이전 뉴런이며, $U_{l+1,j}$ 는 $l+1$ 번째 계층의 j 번째 시냅스 이후 뉴런이다. 스파이킹 뉴런은 이전 뉴런으로부터 전달받은 스파이크에 비례하여 막전위를 상승시킨다. 이때 막전위가 임계값에 도달하면 막전위를 초기화하고 스파이크를 발화한다. 이런 event-driven 한 특징으로 인하여 SNNs 은 저전력으로 작동한다.

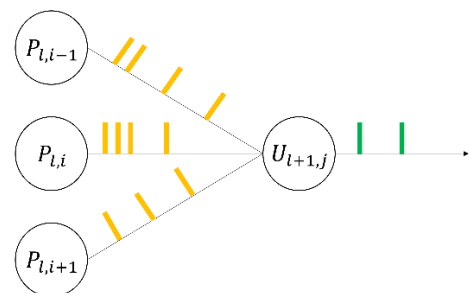


그림 1. SNN 의 구조

III. 실험 결과

3.1 Dataset

객체 탐지를 위한 데이터셋으로 Microsoft Common Objects in Context (MS COCO) 2014 를 사용한다. COCO 2014 는 약 82.8k 의 훈련 이미지와 약 40.5k 의 검증 이미지를 제공한다. 이때 각 이미지는 80 개의 클래스와 함께 bounding box 가 제공되며, 각 이미지에 포함된 bounding box 의 개수 및 클래스의 개수는 다르다.

3.2 Vehicle-YOLOv3-Tiny

전처리 과정으로 COCO 2014 에서 제공하는 80 개의 클래스 중에서 차량 주행과 관련이 먼 클래스들(horse, sheep, cow, elephant, 등)을 제거하여, 11 개의 클래스들(person, bicycle, car, traffic light, 등)이 포함된 새로운 데이터셋을 구성한다.

Vehicle-YOLOv3-Tiny 는 YOLOv3-Tiny 를 수정한 네트워크 구조이다. 기존의 Tiny 모델은 활성화 함수로 leaky-ReLU 를 사용한다. Leaky-ReLU 는 입력 값이 양수이면 양수 값을 출력하고 입력 값이 음수이면 음수 값을 출력한다. 그러나 SNNs 의 스파이크는 양수 값을 갖는다. 그렇기 때문에 leaky-ReLU 가 아닌 ReLU 를 활성화 함수로 사용한다. 추가로 YOLOv3-tiny 를 SNNs 으로 변환하기 위해 upsample 을 transposed convolutional 로 수정하며, maxpool(stride=1) 계층을 제거한다. Maxpool(stride=2) 계층은 convolution 계층의 stride 를 2 로 늘리는 것으로 제거 및 수정한다.

수정한 네트워크 모델을 학습시키기 위해 51,739 개의 이미지 데이터를 batch size 32 로 15 epoch 만큼 학습을 진행한다. 이후 test 에서는 25,080 개의 이미지를 batch size 32 로 진행한다.

3.3 결과

성능 측정 지표로 정밀도(Precision), 재현율(Recall), 평균 정밀도 (mAP), F1-score 를 사용한다. 정밀도는 모델의 예측을 얼마나 신뢰할 수 있는지에 대한 지표다. 재현율은 모델의 감지 성능에 대한 지표다. mA 는 모델의 여러 클래스에 대한 정밀도 지표다. F1-score 는 모델의 종합적 성능에 대한 지표다.

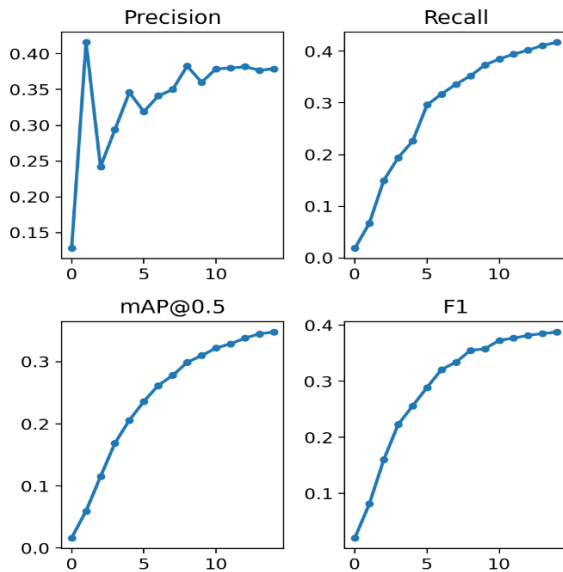


그림 2. Vehicle-YOLOv3-Tiny 의 성능

그림 2 는 설계한 전처리 과정 및 네트워크 구조를 사용하였을 때의 성능이다. 최종 결과 정밀도는 0.379, 재현율은 0.417, mAP 는 0.348, F1-score 는 0.388 을 보인다.

Class	Precision	Recall	mAP@0.5
all	0.273	0.00014	0.00677
person	1.000	1.12E-05	0.0164
car	1.000	6.63E-05	0.0164
stop sign	1.000	0.00146	0.0219

표 1. SNNs 으로 변환한 모델의 성능

표 1 은 앞서 제안한 Vehicle-YOLOv3-Tiny 를 SNNs 으로 변환한 후 측정된 값이다. 표 1 에는 전체 클래스에 대한 수치와 11 개의 클래스 중에서 차량 주행에 있어서 중요한 person, car, stop sign 에 대한 값이다. 위의 클래스는 높은 정밀도를 보이지만 재현율과 mAP 에서 낮은 수치를 보인다.

본 논문에서는 SNNs 로 변환 시, 낮은 성능 수치를 보이는 이유는 정보의 손실 때문으로 두 가지의 원인을 제시한다. 첫째, leaky-ReLU 를 ReLU 로 변환을 하면서 음에 값에 대한 정보의 손실이다. 둘째, 데이터를 표현하는 시간인 time step 의 부족으로 인한 정보의 손실이다. 이러한 원인들을 해결하기 위해 추가적인 실험과 모델에 대한 연구가 필요하다.

IV. 결론

본 논문에서는 자율 주행 차량을 위한 Vehicle-YOLOv3-Tiny 제안하였다. 또한 깊은 신경망 모델의 단점인 높은 전력 소모를 해결하기 위해 SNNs 변환에 대한 실험을 진행하였다. 그 결과 SNNs 변환이 이루어지면서 발생하는 성능 하락을 확인하였으며, 2 가지의 정보 손실 원인을 제시하였다.

ACKNOWLEDGMENT

본 연구는 한국연구재단 신소자원과학기술개발사업 (grant no. NRF-2021M3F3A2A01037962)의 지원을 받아 수행된 연구입니다.

참고 문헌

- [1] Jieun Kim, Chan-young Jung, Woong Hwang, Dong June Lim, & Hyeong-ju Noh, "YOLO based Object detection for Autonomous driving and Collision Warning", Korean HCI Society conference, 2023, pp. 1107-1110.
- [2] A. Wong, M. J. Shafiee, F. Li and B. Chwyl, "Tiny SSD: A Tiny Single-Shot Detection Deep Convolutional Neural Network for Real-Time Embedded Object Detection", 2018 15th Conference on Computer and Robot Vision (CRV), Toronto, ON, Canada, 2018, pp. 95-101, doi: 10.1109/CRV.2018.00023.
- [3] B.-Y. LIU, H.-X. CHEN, Y.-Z. YANG and C.-Y. WANG, "A Deep Neural Network Object Detection Method Using Multiscale Poisson Fusion", 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing, Chengdu, China, 2019, pp. 340-344, doi: 10.1109/ICCWAMTIP47768.2019.9067605.
- [4] Seijoon Kim and Seongsik Park and Byunggook Na and Sungroh Yoon, "Spiking-YOLO: Spiking Neural Network for Energy-Efficient Object Detection", The Thirty-Fourth AAAI Conference on Artificial Intelligence, 2019, doi:10.48550/arXiv.1903.06530
- [5] L. Shen, M. Zhong and Y. Chaojie, "Conversion of CNN to SNN for Aerospace Object Detection Tasks", 2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), Chengdu, China, 2022, pp. 919-923, doi: 10.1109/PRAI55851.2022.9904132.
- [6] Joseph Redmon and Ali Farhadi, "YOLOv3: An Incremental Improvement", 2018, doi: 10.48550/arXiv.1804.02767