SSD-Mobilenet-V2 모델을 사용한 Edge Device 에서의 객체검출 성능 비교 및 분석

최석윤*, 최준혁*, 임승호
*한국외국어대학교 정보통신공학과
한국외국어대학교 컴퓨터공학부

csysyjy@hufs.ac.kr, emin38@hufs.ac.kr, slim@hufs.ac.kr

Comparative Analysis of Object Detection Performance on Edge Devices using SSD-Mobilenet-V2 Model

Seok-Yoon Choi*, Joon-Hyuk Choi*, Seung-Ho Lim
* Division of Information Communications Engineering, HanKuk University of Foreign Studies
Division of Computer Engineering, HanKuk University of Foreign Studies

요 약

CPU 와 GPU 의 성능이 지속적으로 발전함에 따라 객체 인식 인공지능의 정확도와 추론 속도는 점차 향상되고 있으나 이러한 성능을 Edge Device 와 같은 제한된 환경에서 구현하기에 아직 여러 한계점이 존재한다. 본 논문에서는 여러가지 Edge Device 에서 객체 인식을 위한 경량화 된 모델 중 하나인 SSD-Mobilenet-V2 를 활용하여 결과값을 통해 각 Device 간 경향성을 분석하였다. 본 결과를 바탕으로 다양한 환경에서의 객체인식 인공지능 모델의 성능 개선을 위한 연구에 활용할 수 있다.

1. 서론

CPU 와 GPU 의 성능이 지속적으로 발전함에 따라 객체 인식 인공지능의 정확도와 추론 속도는 점차 향상되고 있다. 그러나 이러한 성능 향상에는 전력소비와 발열과 같은 제약조건이 따른다. 따라서 Edge Device 와 같이 자원이 제한된 환경에서 객체 인식 인공지능 모델을 실행할 때는 모델의 경량화가 필수적이다.

최근에는 모바일 기기, IoT, 자율주행 시스템, CCTV 와 같은 감시장비에서도 객체 인식 기술이 활용되고 있다. 이러한 환경에서는 높은 정확도는 물론이고, 실시간 추론이 가능할 수준의 짧은 수행시간이 매우 중요하다. 이러한 제약조건을 고려하여, 본 연구에서는 Edge Device 인 Raspberry Pi 4B, Coral Dev Board, Jetson Nano 에서 각 Device 에 맞게 경량화 및 최적화 된 SSD-Mobilenet-V2 모델을 실행하여 FPS, Accuracy, Recall 및 추론시간의 수치를 비교하였고, 그 경향성을 분석하였다.

2. 실험 환경 및 요구사항

SSD-Mobilenet 은 SSD 모델[1]과 Mobilenet 모델[2]의 아이디어를 결합한 것으로 기존의 딥러닝 모델에 비해 훨씬 적은 파라미터를 가지고 있고 연산량을 줄여 Edge Device 환경에 적합한 객체 감지 모델이다.

각 디바이스에서의 가장 효율적인 모델 선택을 위해 Coral Dev Board 에는 EdgeTPU 가속 지원을 위한 SSD_Mobilenet_v2_EdgeTPU.tflite 모델을, Raspberry Pi 에는 SSD_Mobilenet_v2_CPU.tflite 모델로 구성하였다. Tensorflow 의 32-bit floating point 모델에서 TFLite 의 8-bit fixed point 로의 양자화 단계를 거쳤으며 이는 기존 모델 비트를 1/4 로 줄여 추론시간을 감소시킨다.[3] Jetson Nano 에서는 GPU 가속 지원을 위해 TensorRT 로 최적화된 16-bit floating point 연산 모델로 시스템을 구축하였다.

각 디바이스 간 모델의 추론 이미지 크기는 모두 300x300 으로 동일하며 90 Objects 의 COCO Dataset 으

로 학습된 weight 를 사용하였다.

(그림 1)과 (그림 2)는 Tensorflow 와 TensorRT 를 기반으로 하는 모델의 동작 구조이다.

(그림 1) TFLite 모델의 동작구조

(그림 2) TensorRT 모델의 동작구조

3. 실험 결과

각 디바이스별 최적의 모델로 FPS(Frames Per Second), Accuracy, Recall 성능을 측정하였다.

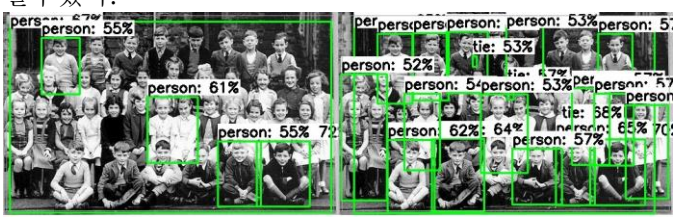
<표 1>은 각 디바이스에서 Webcam 을 이용하여 3 분 간 객체 인식 시 측정한 평균 FPS 값이다. 객체 인식 시작 시 모든 디바이스에서 모델을 로드하는 과정으로 인해 첫 프레임 값은 낮게 측정된다. 이에 따라 첫 프레임 값은 제외하고 계산하였다. Raspberry Pi 의 경우 Tensorflow2 로 작성된 모델이 Tensorflow1 로 작성된 모델보다 약 24%정도의 프레임 향상을 확인할 수 있다. 반대로 Coral Dev Board 에서는 Tensorflow1 모델이 Tensorflow2 모델보다 약 50%정도 높은 프레임을 보이는 것을 확인할 수 있다. 각 디바이스에서의 FPS 값을 이용하여 객체인식의 Inference Time 또한 얻을 수 있다.

Device	Model (TensorFlow version)	FPS
Raspberry Pi	SSD_Mobilenet_V2.tflite (TF1, TF2)	6.45 / 8.04
Coral Dev Board	SSD_Mobilenet_V2_EdgeTPU.tflite(TF1, TF2)	76.73 / 49.81
Jetson Nano	SSD_Mobilenet_V2 (TensorRT)	24.02

<표 1> 각 Device 별 Model 에 따른 FPS

(그림 3), (그림 4)는 5000 장의 COCO image data 를 이용하여 객체 인식 후 각 디바이스의 Accuracy 와 Rcall 성능을 잘 비교 할 수 있는 결과 이미지이다.

CoralBoard 와 Raspberry pi 두 기기 모두 Tensorflow1 모델이 Tensorflow2 모델에 비해 Accuracy 가 대체적으로 높았다. Recall 측면에서는 Tensorflow2 모델이 더 정확도 높은 모습을 보인다. Jetson Nano 의 경우에는 Tensorflow1 모델과 Tensorflow2 모델의 중간 수준을 보인다. 정확도와 인식률이 종합적으로 높음을 확인할수있다.



(그림 3) 좌측부터 Raspberry Pi TF1, TF2

(그림 4) 좌측부터 Coral Dev Board, Jetson Nano

4. 결론

본 논문에서는 SSD-Mobilenet-V2 모델을 이용하여 객체인식 추론능력을 다양한 Edge Device 간에 비교하는 실험을 수행하였다. 결과적으로, Raspberry Pi 는 CPU 를 사용하는 모델로 가장 낮은 성능을 보였다. Coral Dev Board 는 Edge TPU 가속을 지원하는 TFLite 모델을 사용하여 가장 빠른 추론시간을 보였으며, Jetson Nano 는 GPU 가속을 지원하는 TensorRT 모델을 사용하여 정확도와 추론시간 측면에서 준수한 성능을 보였다. 또한, 모델의 경량화는 FPS 와 Accuracy 측면에서 효과적이었다. 그러나 EdgeTPU 가속 지원모델에서 Tensorflow1 으로 작성된 SSD-Mobilenet-V2 모델이 Tensorflow2 로 작성된 모델보다 성능이 높게 나오는 점을 확인하였다.

각 Edge Device 의 특성과 제약조건을 고려하여 적절한 경량화와 최적화된 모델을 선택하는 것이 중요하다. 또한, 추가적인 연구를 통해 다양한 환경에서의 객체인식 인공지능 모델의 성능을 개선하는 방안을 모색할 필요가 있다. 추가적으로 EdgeTPU 모델에서 Tensorflow1 모델이 Tensorflow2 모델보다 추론시간이 느린 이유에 대해서 후속 연구 및 실험을 진행해 나갈 계획이다.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) (NRF-2021R1F1A1048026).

참고문헌

[1] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg – “SSD: Single Shot MultiBox Detector” ECCV 2016

[2] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam - “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications” (2017)

[3] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, Dmitry Kalenichenko – “Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference” (2017)