모바일 기기 환경에서의 최적 경량 영상인식 모델에 관한 연구

배은지, 이성진

동서울대학교 전자공학과

ejbae25@du.ac.kr, sungjinlee@du.ac.kr

Research of Optimal Lightweight Image Classification Models in environments of Mobile Devices

Bae Eunjee, Lee Sungjin

Dong Seoul University, Department of Electric Engineering

요 약

영상인식 기술에 대한 산업적 요구와 관심에도 불구하고 그 방대한 연산량으로 인해 모바일 기기에 탑재하여 실시간으로 쓰기에는 아직까지 성능적 어려움이 존재한다고 알려져 있다. 하지만 이런 성능적 어려움에 대해 다양한 영상인식 경량화 연구들이 진행되어 최근 인식 속도, 메모리, 정확도 모든 측면에서 기존 심층 네트워크에 필적할 만한 성능을 내고 있다. 본 논문에서는 이 기법들에 대해 여러 모바일 기기에 탑재하여 그 하드웨어들에 따른 성능들을 비교 분석하며 이들의 성능 차이들에 대한 이유들을 분석 고찰하였다. 마지막으로 해당 모델들의 성능 분석을 통해, 주어진 하드웨어 자원에 최적화된 영상인식 기법들의 조합에 대해 제시하였다.

Ⅰ. 서 론

본 논문에서는 최근의 딥러닝 기반 영상인식 기술에 대한 산업적 관심은 다양한 분야에서 일어나고 있다. 비단 검색이나 소셜 네트워크 서비스와 같은 소프트웨어 IT 산업 뿐 아니라 자율주행, 전기자동차와 같은 자동차 산업, 드론 및 항공기기 산업, 모바일 및 IoT 기기 산업, 보안기기 산업, 관련된 반도체 산업 등에서 그 성장 잠재력으로 인해 엄청난 자본과 기술력이 모이고 있다.

하지만, 이런 딥러닝 기술은 높은 연산량과 높은 메모리 용량 요구로 인해 고사양의 하드웨어 자원을 요구하고 처리 속도도 늦어서 일반적으로 모바일 기기에 탑재하기 힘든 것으로 알려져 있다.

그럼에도 이와 같은 산업적 요구에 MobileNet [1]을 필두로 하여 SqueezeNet [2], Squeeze and Excitation Network [3], ShuffleNet v1, v2 [4,5], ShiftNet [6], EfficientNet [7], MNasNet [8], MobileNet v2, v3 [9,10] 등이 개발되었다.

하지만 이런 기술들의 출현에도 해당 연구들이 제안하는 기법들에 대한 다양한 하드웨어 기기 환경에서 그 성능들을 비교한 연구는 지금까지 없었다.

본 연구에서는 기존 경량화 기술들을 동일한 데이터 셋에 대해서 훈련하여 다양한 모바일 기기 하드웨어 환경들에 탑재하고 각각의 환경에서 수행되는 추론 속도, 정확도를 비교하여 보았다. 그리하여 각 기기 및 서비스 별로 최적의 Convolution 추출기 네트워크의 분류를 제시하였다.

Ⅱ. 영상분류 모델

1. 영상분류 모델의 선정

딥러닝 기반 영상분류 모델의 성능 향상은 그 Convolution 기반 특성 추출의 탁월함으로 점철될 수 있다. 그러다 보니, 이런 Convolution 추출기 네트워크를 어떻게 설계하느냐가 전체 성능과 연산 효율성을 결정지을 수 있는 중요한 요소가 되었었다.

딥러닝 도입 초기의 Convolution 추출기 네트워크 설계는 주로 성능 향상에 초점을 맞추어 진행되다 보니, 계층 수는 점차 증가하고 구조는 더욱 복잡하게 설계 되었었다 [11]. 하지만, 매우 복잡한 구조와 연산량을 가지고 있는 Inception 과 성능은 유사하면서도 구조는 훨씬 간단하면서 연산량 또한 줄어든 VGGNet [12] 이 나오면서 convolution 추출기 네트워크의 연산 효율성에 대한 관심이 높아지기 시작하였다. 대표적인 네트워크로 MobileNet 이 있고, 이를 개선한 MobileNet v2, MobileNet v3 등이 나왔다. 또한 이런 MobileNet v2, 3가 출현하기 까지 영향을 미쳤던 ResNet [13], Squeeze and Excitation Network 이 등장하고 급기야 모델링 자체를 기계학습으로 자동적으로 수행하는 AutoML 기반의 MNasNet 까지 등장하였으며 이런 Convolution 연산에서 곱셈 연산을 Shift 연산으로 대체한 ShiftNet 라는 연구 까지 등장하며 Convolution 경량화는 많은 발전을 이루었다.

본 연구에서는 이런 경량화 네트워크들 중에 가장 보편적인 MobileNet, Quantized-MobileNet 와 EfficientNet, Quantized-EfficientNet 을 성능 평가 모델을 대상으로 하였다.

2. 동작 하드웨어 선정

실험을 위해 사용된 모바일 기기는 Android OS와 iOS를 기준으로 삼성 갤럭시 S9, S10, S20 와 iPhone XR, iPhone 11 Pro를 대상 기기로 선정하여 실험하였다. 실험결과에 결정적인 영향을 미칠 수 있는 각 기기의 Hardware 사양은 표 1과 같다.

	CPU	GPU	RAM	카메라
갤럭시S20	ARM Cortex-A77 Single-Core 2.84 GHz + Triple-Core 2.42 GHz CPU ARM Cortex-A55 Quad-Core 1.8 GHz CPU	퀄컴 Adreno 650 587 MHz GPU	12GB	12.0 MP + 12.0 MP + 64.0 MP
갤럭시S10	Samsung Exynos M4 Dual-Core 2.73 GHz CPU ARM Cortex-A75 Dual-Core 2.31 GHz CPU ARM Cortex-A55 Quad-Core 1.95 GHz CPU	ARM Mali-G76 12-Core 702 MHz GPU	8GB	12.0 MP + 16.0 MP + 12.0 MP
갤럭시S9	Samsung Exynos M3 Quad-Core 2.7 GHz CPU ARM Cortex-A55 Quad-Core 1.79 GHz CPU	ARM Mali-G72 18-Core 572 MHz GPU	4GB	12.0 MP
iPhone11 pro	Apple Lightning MP2 2.67 GHz CPU Apple Thunder MP4 1.73 GHz CPU	Apple 3rd Design GPU Architecture MP4 GPU	4GB	12.0 MP + 12.0 MP + 12.0 MP
iPhoneXR	Apple Vortex MP2 2.5 GHz CPU Apple Tempest MP4 1.53 GHz CPU	Apple G11P MP4 1.1 GHz GPU	3GB	12.0 MP

표 1. 실험 기기의 Hardware 사양

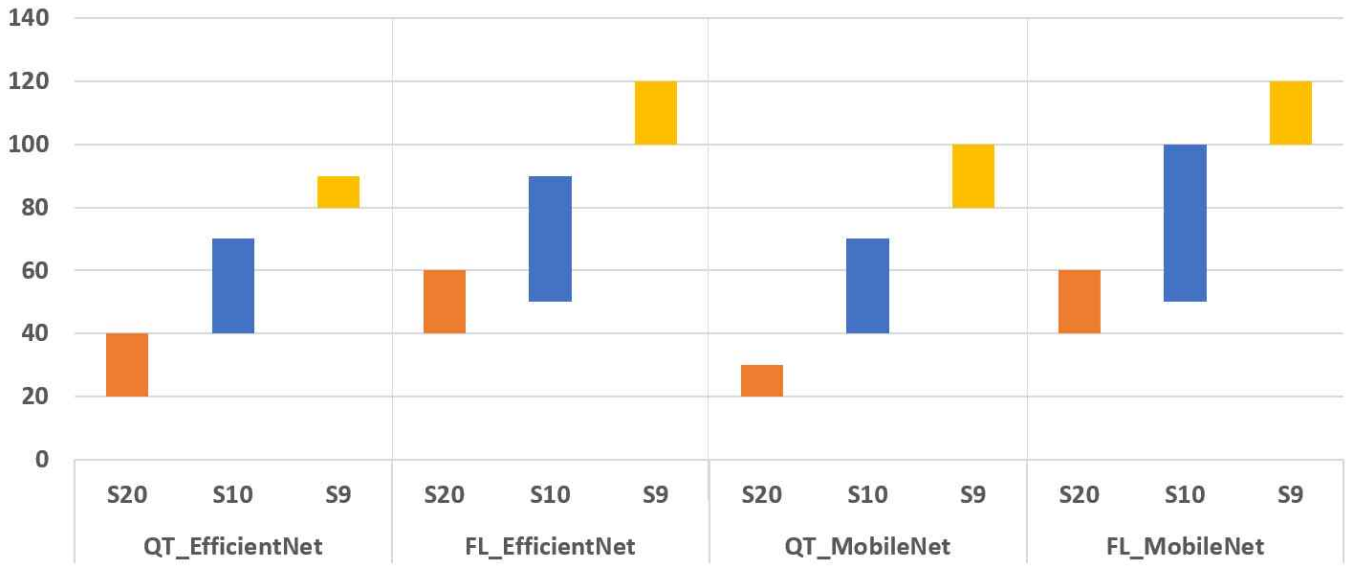

그림 1. 모델별 인식 지연 시간 비교

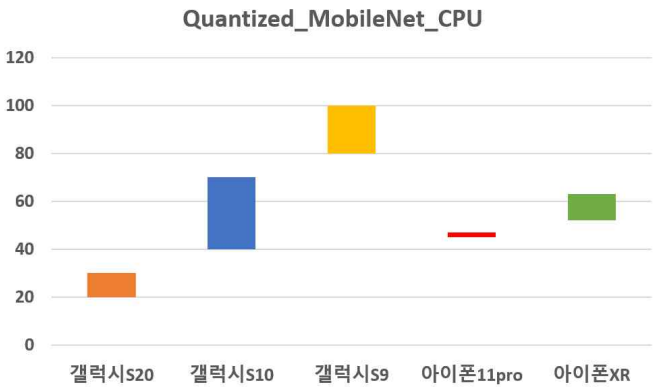

그림 2. 모바일 기기별 인식 지연 시간 비교

3. 데이터 셋 및 성능 측정 Metric 선정

훈련 用 데이터 셋은 ImageNet을 대상으로 하였다. 성능평가 用 데이터 셋은 ImageNet의 validation의 이미지들을 사용하였다. 평가 Metric은 동작속도로 Frame Per Second를 사용하였다.

4. 모델 변환

일반적인 딥러닝 개발 툴인 TensorFlow를 기반으로 해당 모델들을 각 모바일 기기 환경에 최적화된 포맷인 TF-Lite로 변환하여 탑재하였다. 해당 과정은 tensorflow 공식 홈페이지 [14]를 참고하여 수행하였다.

Ⅲ. 영상분류 모델

그림 1은 갤럭시 S9, 10, 20 기기에 MobileNet과 EfficientNet의 Float32 bit 버전과 Quantization 버전모델을 탑재하였을 때 의 인식 지연 시간을 비교한 그래프이고 그림 2는 아이폰 XR과 11 Pro 기기에 Quantized MobileNet 모델을 탑재하였을 때의 인식 지연 시간을 비교한 그래프 이다.

일단 첫 번째 발견으로 S9에서 S20으로 기기가 발전될수록 인식 지연 시간이 50% 정도씩 줄어든다는 것을 알 수 있다. 마찬가지로 아이폰 또한 XR에서 11 Pro 로 발전될수록 인식 지연 시간이 약 20% 정도 줄어든 것을 확인할 수 있다. 당연하지만 이를 통해 하드웨어의 발전이 인식속도에 긍정적인 영향을 미친다는 것을 확인 할 수 있으며 특히 세 기기 (S9, S10, S20) 모두 CPU의 Clock 조건은 거의 동일함에도 불구하고 인식속도가 많은 차이를 보이는 것으로 보아 RAM 사양이 인식 속도에 영향을 미칠 수 있다는 것을 알 수 있다.

두 번째 발견으로 Quantization 모델을 사용할 경우 인식 속도에 긍정적인 영향을 미칠 수 있다는 것을 알 수 있다.

Ⅲ. 결론

본 연구는 대표적인 영상분류 모델들과 그 연산량 감소를 위해 개선된 경량화 구조 모델들, 그리고 양자화, 프루닝 기법들의 다양한 기기 환경에서 그 인식 속도 성능을 비교 분석하여 보았다. 그 결과 인식 속도에 결정적으로 영향을 미치는 요인으로 CPU 성능, RAM 사양, 모델의 축소 여부(Quantization) 가 있으며 이는 해당 스마트폰 종류에 따라, S/W (Android, iOS) 에 따라 달라 질 수 있음을 알았다.

ACKNOWLEDGMENT

이 논문은 2020년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기본연구사업임(No. NRF-2019R1F1A1062878)

참 고 문 헌

[1] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. CoRR,abs/1704.04861, 2017.

[2] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. CoRR, abs/1602.07360, 2016.

[3] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-Excitation Networks. ArXiv e-prints, Sept. 2017.

[4] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. CoRR, abs/1707.01083, 2017.

[5] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. ECCV, 2018.

[6] Bichen Wu, Alvin Wan, Xiangyu Yue, Peter H. Jin, Sicheng Zhao, Noah Golmant, Amir Gholaminejad, Joseph Gonzalez, and Kurt Keutzer. Shift: A zero flop, zero parameter alternative to spatial convolutions. CoRR, abs/1711.08141, 2017.

[7] Mingxing Tan, and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning, PMLR 97:6105-6114, 2019.

[8] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, and Quoc V. Le. Mnasnet: Platform-aware neural architecture search for mobile. CoRR, abs/1807.11626, 2018.

[9] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks, detection and segmentation. CoRR, abs/1801.04381, 2018.

[10] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for MobileNetV3. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019

[11] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. CoRR, abs/1602.07261, 2016.

[12] Karen Simonyan, and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.

[14] https://www.tensorflow.org/lite/guide?hl=ko