CNN 기반의 모바일 디바이스 실시간 물체 탐지 어플리케이션

Real-Time Object Detection Using CNN on Mobile Device

손장민 조혁준

요 약

모바일 기기의 카메라의 모듈을 사용하여 영상의 이미지를 Tensorflow를 사용하여 CNN을 이용한 사물을 분석 및 인식하는 모바일 어플리케이션을 만들고자 한다.

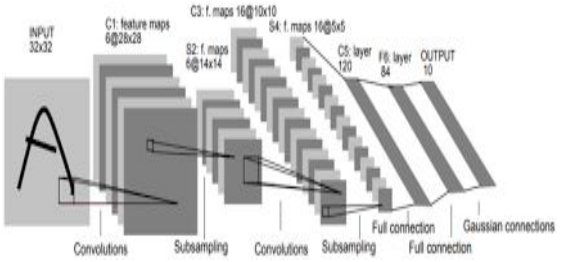
1. 서론

우리는 스마트폰이 가지고 있는 장점인 카메라와 머신러닝 분야에서 이미지를 처리 할 때 주로 사용되는 CNN(Convolution Neural Network) 기술을 접목시키면 이미지 처리를 위한 다양한 분야에서 사용 할 수 있다고 생각했다. 예를 들어 상품을 촬영하면 해당 제품에 대한 상세 정보를 출력 해 줄 수 있고, 농촌 지역의 야생동물을 감지하여 알람을 울려 농작물 피해를 막거나, 시각 장애인들을 위해 물체를 판별하고 음성으로 제공 할 수 있다. 이러한 무궁무진한 활용도에 기여하기 위해 편의점의 물체를 데이터셋으로 이용하여 무인 계산에 도움을 줄 수 있는 어플리케이션을 만들기로 기획했다.

2. 기존연구

2.1 CNN(Convolution Neural Network)

컨볼루션 신경망(CNN)은 심층 감독학습을 기반으로 하는 machine learning 모델이며, 적용력이 뛰어나고 국부적 특징 추출 및 분류에 강하다. 가중치 공유 구조 특징 때문에 컨볼루션 신경망 모델은 생물학적 신경망과 더욱 유사하게 설계되어으며 패턴인식 영역에서 탁월한 성과를 얻고있다. 컨볼루션 신경망은 convolution layer, pooling layer(or subsampling layer), fully connection layer 등으로 구성된다. 컨볼루션 신경망 모델에서 일반적으로 입력 층 input과 출력 층 output은 각각 1개로 고정되어 있고 convolution layer와 pooling layer를 여러 개로 구성할 수 있다. 전역 연결 층인 fully connection layer는 최종 출력층을 구성하기 위한 구조임을 볼 수 있다. Fully connection layer에서는 일반적으로 back propagation 알고리즘이나 back propagation 알고리즘의 단점을 보완한 gradient descent method나 wake-sleep 알고리즘을 적용한다. 컨볼루션 신경망은 특징을 보면 sparse weight를 통해 모델의 복잡도를 줄여주는 장점이 있고 parameter sharing을 통해 특정 가중치 그룹들은 가중치 값이 항상 같도록 변수를 공유하게 한다. 마지막으로 상위 sparse weight를 특정한 형태로 배치하였을 때, 주어진 입력 값의 변화에 대해 출력이 효율적으로 변화하는 방식이 동등하게 변화도록 한다. CNN의 기본 구조는 Figure 1과 같다.[1]



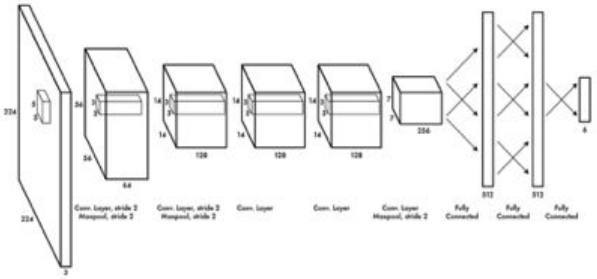
(**Figure 1**. CNN Structure)

2.2 CNN Model

When building our grasp detection system we want to start from a strong foundation.

We derive our model from a version of the widely adopted. convolutional network proposed by Krizhevsky et al. for object recognition tasks (AlexNet).

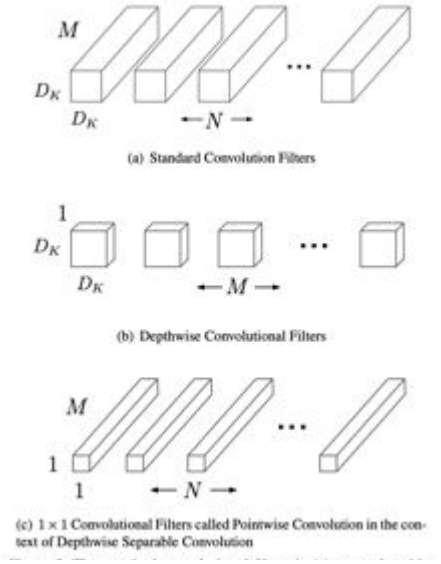
Our network has five convolutional layers followed by three fully connected layers. The convolutional layers are interspersed with nomalization and maxpooling layers at various stages. A full description of the architecture can be found in Fugure 2.



(**Figure 2**. YOLO GRASP DETECTION WITH NEURAL NETWORKS Architecture)

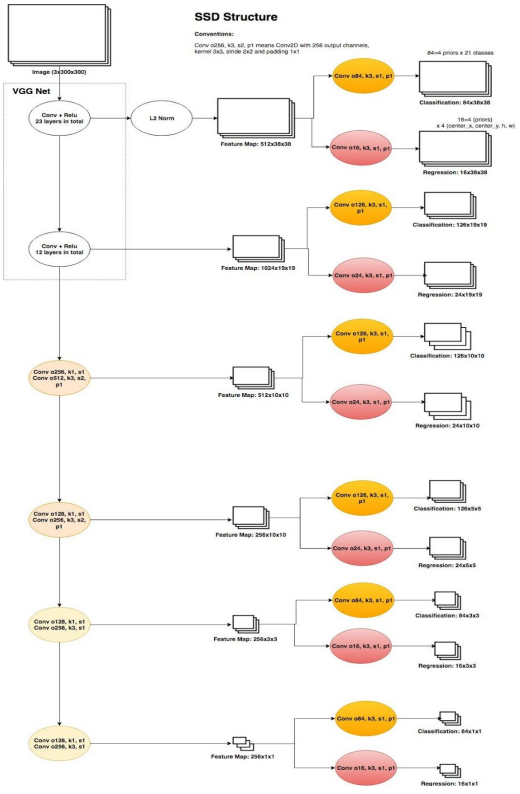
2.3 Mobilenet

The MobileNet model is based on depthwise separable convolutions which is a form of factorized convolutions which factorize a standard convolution into a depthwise convolution and a 1×1 convolution called a pointwise convolution. For MobileNets the depthwise convolution applies a single filter to each input channel. The pointwise convolution then applies a 1×1 convolution to combine the outputs the depthwise convolution. A standard convolution both filters and combines inputs into a new set of outputs in one step. The depthwise separable convolution splits this into two layers, a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size. Figure 3 shows how a standard convolution 2(a) is factorized into a depthwise convolution 2(b) and a 1 × 1 pointwise convolutionn 2(c).[2-1]



(**Figure 3**.Mobilenet Architecture)

2.4 MobileNet SSD



(**Figure 3**.VGG based SSD Architecture)

After going through a mobilenet network for fearture extraction, we obtain a feature layer of size *m*n(number of locations)*

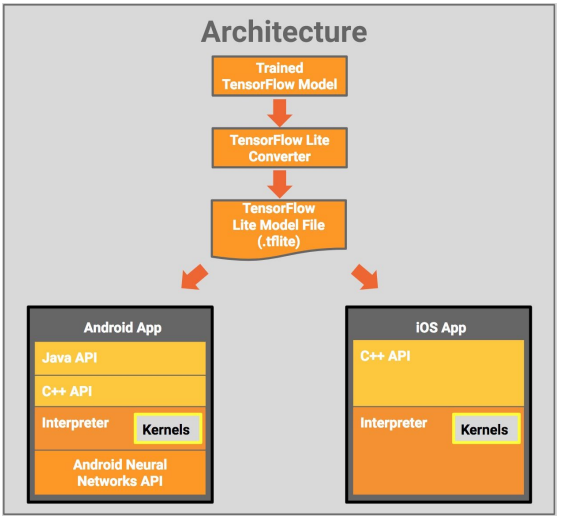
with *p* channels for each location, we got *k* bounding boxes, Thes *k* bounding boxes have different size and aspect ratios, The concept is, maybe a vertical rectangle is more fit for human, and a horizontal rectangle is more fit for car. For each of the bounding box, we will compute *c* class scores and 4 offsets relative to the original default bounding box shape.

3. 시스템 모델

3.1 기존 연구와 차이점 및 해결방안

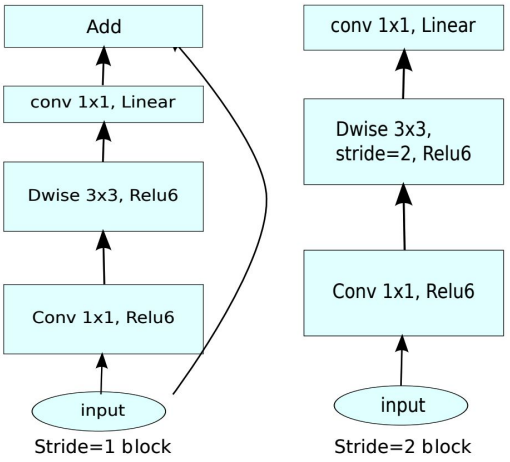
기존 연구들의 성능저하 이슈는 GPU의 메모리, 성능의 문제로부터 발생 한다는 점에서 착안하여 우리는 추론단계의 퍼포먼스를 최대화 하기 위하여 모델의 메모리 사용량을 최소화 한다. 이를 위하여 Tensorflow로 학습 된 모델을 TFLite 형식으로 변환하여 사용한다. 또한 모바일 환경에서 가능한 GPU의 성능을 고려하여 모델에서 사용되는 데이터의 형식을 가공하여 Latency Delay를 낮춘다. 때에 따라서 메모리의 사용량을 최소화 하면 정확도가 떨어 질 수 있는데 해당 부분을 잘 조율하기위해 모델의 구조를 일부 변경하거나 모델에 사용되는 변수의 데이터 형태를 조정하는 방법을 통해 가장 시간 대비 효율이 좋은 모델을 사용 할 것이다.

4. 프로젝트 내용



(**Figure 5**. tflite architecture)

이 프로젝트는 Tensorflow를 이용하여 생성 된 모델을 TF-Lite 형식으로 변환하여, 변환된 파일들을 안드로이드OS 운영체제의 모바일 환경에 이식하는것을 목표로 한다. 모델은 TF-Slim을 사용하여 학습하며, 이 때 이용하는 모델은 SSD Mobile Net V2 모델이다.

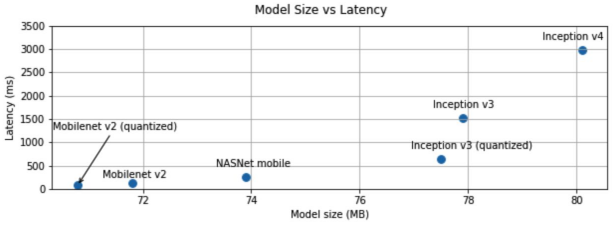


(**Figure 4**. Mobile Net V2)

Size	MobileNetV1	MobileNetV2	ShuffleNet (2x,g=3)
112x112	64/1600	16/400	32/800
56x56	128/800	32/200	48/300
28x28	256/400	64/100	400/600K
14x14	512/200	160/62	800/310
7x7	1024/199	320/32	1600/156
1x1	1024/2	1280/2	1600/3
max	1600K	**400K**	600K

(Figure 5. Memory Usage Table)

해당 모델은 이전 버전의 V1, ShuffleNet 보다 채널수, 메모리 부분에서 가장 적은 수를 차지하기 때문에 제한적 상황인 모바일 디바이스에서 적합하다고 판단하였고 또한 Real Time으로 들어오는 입력을 처리하기위해 *Fig.6*에서 Latency가 낮은 모델을 사용하는것이 합리적이라 생각하였다.



(Figure 6. Latency vs Model Size)

학습 된 모델을 *Fig. 4*와 같이 tflite 확장자로 변경 한 후 안드로이드 기기에 이식 한다.
사용 된 데이터 셋과 개수는 다음과 같다.

Object	Trained Img	Validation Img
코카콜라	400	40
팹시콜라	400	40
포카칩	400	40
도리토스	400	40
닭다리	400	40
말보로골드	400	40
고래밥	400	40
신라면	400	40

5. 결과

직접 데이터셋을 모아 라벨링을 거친 후 생성한 모델을 기반으로 물체를 인식하는 실험을 하기위해 유사한 텍스쳐와 질감을 가진 대조군을 가지고 비교하였다.

5-1. 캔(Can)

Coca Cola	Pepsi Cola
	

(Figure 7. Can object detection and classification)

Figure 7. 에서 확인할 수 있듯이 코카콜라와 펩시를 비교한 결과 인식하고자 하는 물체의 위치정보와 식별 정보를 정확히 추출한것을 확인할 수 있다.

5-2. 비닐(Plastic bag)

Pocachip	Doritos
	

(Figure 8. Plastic bag object detection and classification)

해당 대조군은 비닐이라는 특징을 지니고 있는 물체에 대한 물체 식별이다. Figure 8. 을 보면 포카칩과 도리토스 모두 바운딩박스의 위치 정보와 식별 정보를 확인 할 수 있다. 하지만 비닐이라는 물체의 특징 때문에 빛의 변화에 민감하게

반응하여 빛의 변화에 따라 적절한 인식 결과를 도출하지 못하는 경우가 발생한다. 따라서 다양한 빛의 변화를 적용한 데이터셋을 구성할 경우 더 높은 인식 성능을 확인할 수 있을것으로 보인다.

5-3. 박스(Box)

Whale Food	Chicken Leg
	

(Figure 9. Box object detection and classification)

박스(Box) 의 경우 대부분의 텍스쳐가 매트한 성질이 존재하여 빛에 대해 덜 민감한 반응을 보인다. 이에 따라 빛의 변화가 존재하더라도 높은 object detection 성능을 보는 것을 확인 할 수 있었다.

5-4. 유사한물체(Smiliarity)

신라면(Front View)	신라면(Top View)	닭다리	코카콜라
			

(Figure 10. Smiliarty object detection and classification)

유사한 물체에 대하여 물체의 성질과 각도, 촬영한 빛의 변화에 따른 object detection 과정을 거칠 경우에도 충분한 dataset을 기반으로 모델을 훈련 하였을 경우 Figure 10. 에서 확인 할 수 있듯이 인식을 성공한다.

6. 결론 및 향후 연구

모바일 디바이스의 카메라 모듈을 통해 실시간으로 사물을 판별함으로써 사람의 감독이 필요 한 부분이 기계로 모두 자동화가 될 수 있다. 농촌 지역의 야생동물로 인한 농작물 피해, CCTV를 이용한 사람 인식, 시각 장애인을 위한 안내 장치 등 사용 분야의 영역이 상당히 넓다. 하지만 아직 하드웨어적으로 머신러닝을 완벽히 지원 하지 못하기 때문에 소프트웨어적 설계가 중요하다. 따라서 우리는 CNN 모델을 적용하여 적합한 모바일 어플리케이션을 개발하는것이 목적이다. 향후 다양한 분야에서 적용해서 개발할 수 있을것으로 기대된다.

참 고 문 헌

[1] Ga-Ae Ryu, Kwan-Hee Yoo, “Application of Manufacturing Process Data Classification Using Image Data based CNN”Journal of Information Technology and Architecture Vol. 15. No. 3, September 2018, Pages 337-343

[2] Joseph Redmon, Anelia Angelova, “Real-Time Grasp Detection Using Convolutional Neural Networks”

[2-1] 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”:arXiv:1704.04861v1 [cs.CV] 17 Apr 2017

[3] Nur ÇÜRÜKOĞLU, Buse Melis ÖZYILDIRIM, “Deep Learning on Mobile Systems”

[4] Mark Sandler Andrew Howard Menglong Zhu Andrey Zhmoginov Liang-Chieh Chen Google Inc. “MobileNetV2: Inverted Residuals and Linear Bottlenecks”