딥 러닝을 이용한 첨단 운전자 보조 시스템 어플리케이션

Deep Learning for Advanced Driver Assistance System Applications

조혁준

요 약

모바일 기기의 카메라 모듈을 통해 입력되는 입력 영상을 CNN(Convolution Neural Network)을 이용하여 학습한 딥러닝 모델을 통해 차량 운전 시 필요한 앞 차량과의 거리 경보, 차선 변경 시 경보 기능 등을 구현하고자 한다.

1. 서론

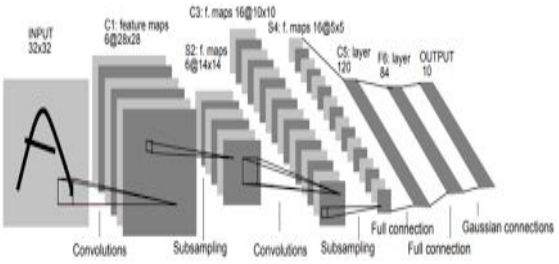
스마트폰이 가지고 있는 장점인 카메라와 딥러닝 분야에서 이미지를 처리할 때 주로 사용되는 CNN(Convolution Neural Network) 기술을 접목시키면 이미지 처리를 위한 다양한 분야에서 사용 할 수 있다고 생각했다. 예를 들어 상품을 촬영하면 해당 상품에 대한 상세 정보와 계산을 할 수 있고, 농촌 지역의 야생동물을 감지하여 알람을 울려 농작물 피해를 막거나, 시각 장애인들을 위해 물체를 판별하고 음성으로 정보를 제공할 수 있다. 이러한 딥러닝의 무궁무진한 활용도에 기여하기 위해 차량 이미지를 데이터셋으로 이용하여 운전자가 차량 내부의 블랙박스와 같이 주행 영상을 촬영하는 모바일 디바이스를 통해 입력되는 영상의 데이터를 처리하여 앞 차량과의 거리가 일정 범위 안에 근접할 시 운전자에게 경보를 주거나 차량이 차선을 변경하는 경우에도 사용자에게 경보를 줌으로써 운전자가 주의해야 될 상황일 경우를 환기시킨다. 이를 위해 모바일 어플리케이션에 적용할 수 있는 object detection 을 위한 딥러닝 모델을 구현하고 앞 차량과의 거리 측정 알고리즘을 적용하는 것이 목표이다.

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.

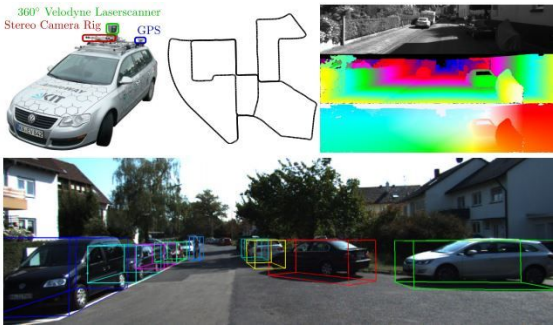
Fully connection layer에서는 일반적으로 back propagation 알고리즘이나 back propagation 알고리즘의 단점을 보완한 gradient descent method나 wake-sleep 알고리즘을 적용한다. 컨볼루션 신경망은 특징을 보면 sparse weight를 통해 모델의 복잡도를 줄여주는 장점이 있고 parameter sharing을 통해 특정 가중치 그룹들은 가중치 값이 항상 같도록 변수를 공유하게 한다. 마지막으로 상위 sparse weight를 특정한 형태로 배치하였을 때, 주어진 입력 값의 변화에 대해 출력이 효율적으로 변화하는 방식이 동등하게 변화도록 한다. CNN의 기본 구조는 Figure 1과 같다.[1]



(**Figure 1**. CNN Structure)

2.2 Autonomous driving system

Developing autonomous systems that are able to assist humans in everyday tasks is one of the grand challenges in modern computer science. One example are autonomous driving systems which can help decrease fatalities caused by traffic accidents. While a variety of novel sensors have been used in the past few years for tasks such as recognition, navigation and manipulation of objects, visual sensors are rarely exploited in robotics applications: Autonomous driving systems rely mostly on GPS, laser range finders, radar as well as very accurate maps of the environment.[2]

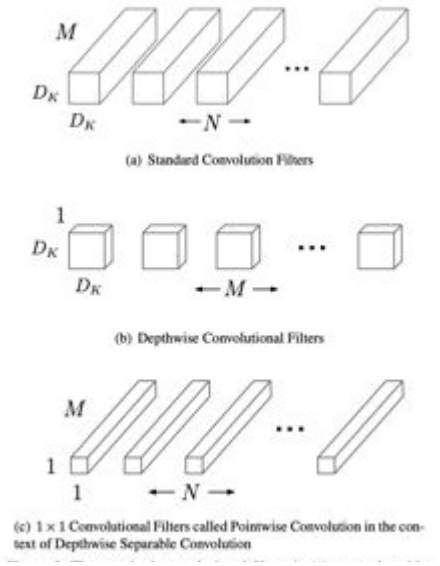


(**Figure 2**.1. Recording platform with sensors (top-left), trajectory form our visual odometry benchmark (top-center), disparity and optical flow map (top-right), and 3D object labels(bottom).)

2.3 Mobilenet

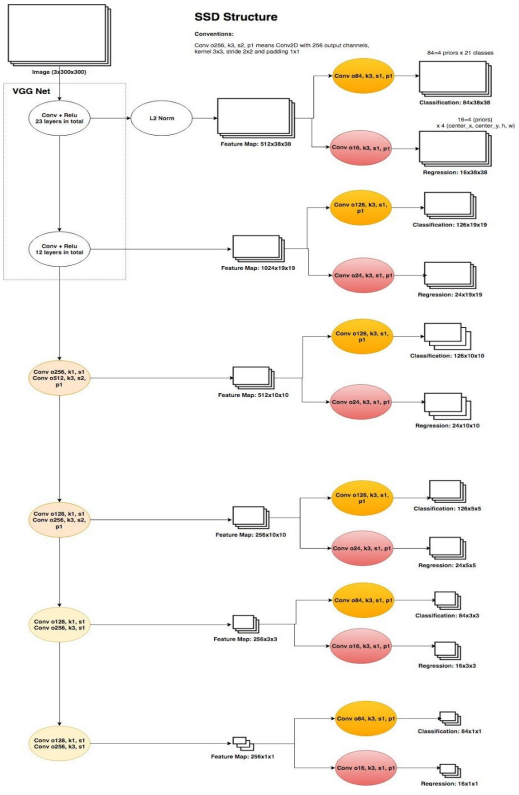
The MobileNet model is based on depthwise separable convolutions which is a form of factorized convolutions which factorize a standard convolution into a depthwise convolution and a 1×1 convolution called a pointwise convolution. For MobileNets the depthwise convolution applies a single filter to each 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 (3-a) is

factorized into a depthwise convolution (3-b) and a 1 × 1 pointwise convolution (3-c).[3]



(**Figure 3**.Mobilenet Architecture)

2.4 MobileNet SSD



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

After going through a mobilenet network for feature 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.

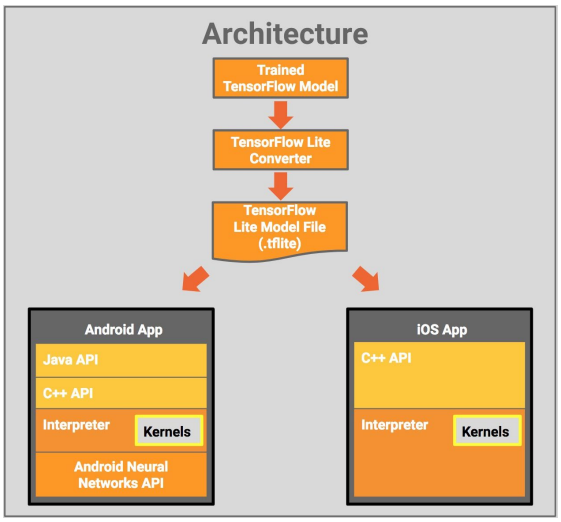
3. 시스템 모델

3.1 기존 연구와 차이점 및 해결방안

대부분의 연구에서 ADAS(Advanced Driving Assistance System)을 구현하기 위해 radar, laser range finders, stereo camera 등 과 같은 장비를 이용하여 구현한다. 또한 딥러닝 모델을 사용하기 위해 고성능의 컴퓨팅 파워를 요구한다. 이는 ADAS 기능을 구현하기 위해 높은 비용을 요구하며 기능이 탑재되지 않은 기존의 차량에서는 동작을 할 수 없게 만드는 요소로 작용한다. 따라서 기존 연구들의 고성능 장비와 대비하여 간단한 스마트폰과 같은 모바일 디바이스에서 ADAS 기능을 구현한다. 이를 통해 사용자는 기존의 차량에서 ADAS 기능을 사용할 수 있을것으로 기대된다. 모바일 디바이스에서 사용하기 위해 모델의 컴퓨터 리소스 사용량을 최적화한다. 모바일 디바이스 상에서 리소스 과부화로 성능이 저하되는 것을 해결하기 위해 mobilenet 과 같은 압축된 모델을

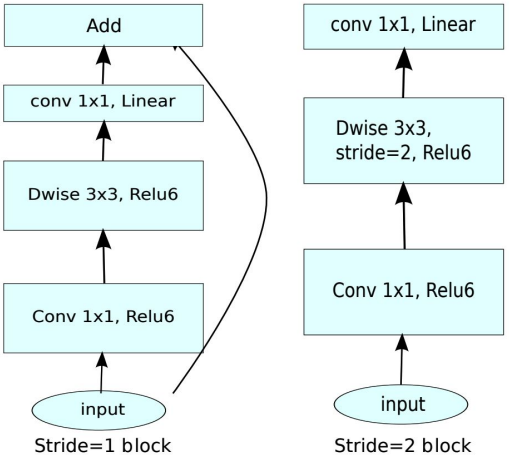
사용함으로써 해결할 수 있을것으로 기대된다.

4. 프로젝트 내용



(**Figure 4**. tflite architecture)

본 프로젝트는 Tensorflow를 이용하여 생성 된 모델을 TF-Lite 형식으로 변환하여, 변환된 파일들을 iOS 운영체제의 모바일 환경에 이식하는것을 목표로 한다. 모델은 gpu 서버 및 클라우드 서버에서 수행하며 이 때 이용하는 모델은 SSD Mobile Net V2 모델이다.

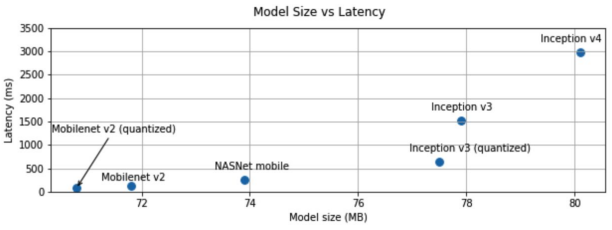


(**Figure 5**. Mobile Net V2)

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

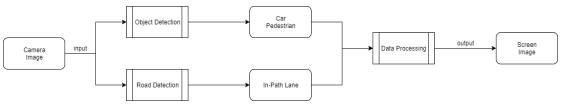
(Figure 6. Memory Usage Table)

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.7*에서 Latency가 낮은 모델을 사용하는것이 합리적이라 생각하였다.



(Figure 7. Latency vs Model Size)

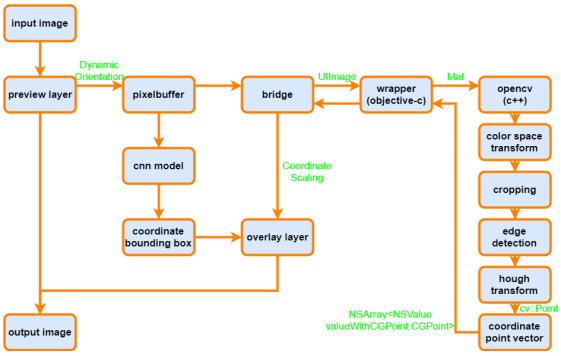
학습 된 모델을 *Fig. 4*와 같이 tflite 확장자로 변경 한 후 iOS 기기에 이식 한다.



(**Figure 8.** System Diagram)

Fig 8. 은 본 프로젝트의 어플리케이션의 작동방식에 대한 다이어그램이다. car, pedestrian detection 과 lane detection 을 통해 기능을 구현한다. 해당 기능을 구현하기 위해 모델의 성능과

모바일 디바이스의 리소스 관리를 만족시켜 구현한다.

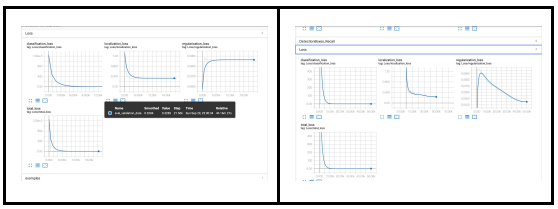


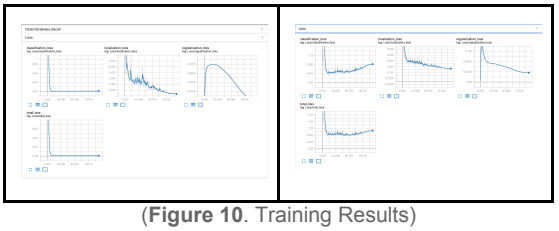
(**Figure 9**. System Architecture)

Figure 9은 iOS 어플리케이션 내부에서 입력받은 pixel-buffer를 처리하는 전체적인 구조를 보여준다. 크게 두 방향으로 pixel-buffer 를 처리하는데 먼저 lane detection 기능을 구현하기 위해 opencv framework를 통해 색영역 기반 처리를 수행하고 해당 부분을 cropping 하는 과정을 거친다 다음 차선의 edge detection을 구한 다음 hough 변환으로 차선에 일치하는 직선을 구한 좌표값을 구한다. 해당 좌표값을 overlay layer에 적합한 coordinate value 를 scaling 과정을 거쳐 전송한다. 다음 pixel-buffer는 학습한 cnn 모델을 통해 bounding-box 좌표값을 얻어 overlay layer에 전송하여 최종적으로 preview layer와 함께 출력 영상을 생성한다.

5. 결과

모델을 생성하는데 있어서 데이터셋과 learning rate를 수정하면서 최적의 모델 성능을 도출하도록 다양한 모델을 생성하였다. Figure 10은 생성한 모델의 loss value 의 결과이다.

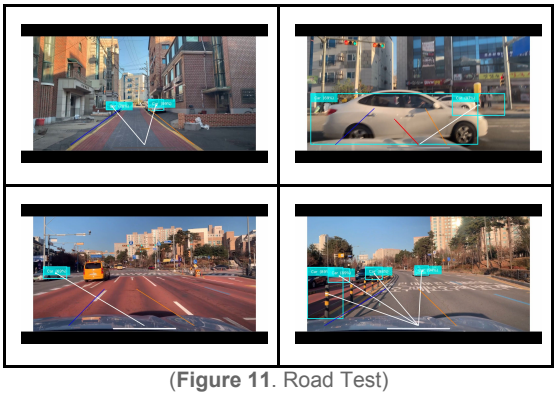




(**Figure 10**. Training Results)

먼저 kitti 데이터셋을 기반으로 훈련한 모델의 경우 실제 상황에서 다소 저조한 성능이 확인되어 data augmentation 및 custom data scaling 과정을 거쳐 모델의 성능을 개선시키는 방향으로 모델을 훈련하였다.

다음 Figure 11은 직접 차량을 타고 본 프로젝트의 어플리케이션을 테스트한 결과이다.



(**Figure 11**. Road Test)

실제 도로에서 테스트한 결과 차량을 인식하고 차량이 가까울 때 경고 알람을 울리는 기능이 정상적으로 작동하는 것을 확인할 수 있었다. 하지만 lane detection 기능이 실제 도로 상황에서 사거리와 같은 특수한 경우 적합하지 않은 결과를 다소 보였다. 또한 어린이집 차량, 트럭과 같은 특수 차량을 인식하는데 다소 저조한 성능을 보인다는것을 확인할 수 있다.

6. 결론 및 향후 연구

모바일 기기의 카메라 모듈을 통해 입력되는 입력 영상을 딥러닝 모델을 통해 차량 운전 시 필요한 앞 차량과의 거리 경보, 차선 변경 시 경보

기능 등을 구현하는 어플리케이션을 개발하고자 한다. 해당 어플리케이션을 통해 사용자는 차량의 연식 및 기능과 무관하게 ADAS 기능을 사용할 수 있을것이다. 이는 운전 사고를 감소시키고 사용자의 운전 집중력을 향상할 것이다. 이를 위해 딥러닝 모델을 최적화함으로써 어플리케이션의 신뢰성을 확보할 수 있을 것이다.

이를 위해 먼저 Convolution Neural Network Model 의 개선이 필요하다. 이는 다양한 차량과 상황의 데이터셋을 통해 개선시킬 수 있으며 object detection 이 아닌 object segmentation을 통해 입력 영상을 분석한다면 각 object의 상호관계를 파악하여 lane, car, pedestrian, traffic light 등 다양한 object 를 적용시켜 발전할 수 있다. 또한 반대로 기존의 프로젝트에서 부족하였던 lane detection 기능을 보완하기위해 영상처리 알고리즘을 개선시킬 필요가있으며 모바일 디바이스 gps 기능을 사용하여 차량간 상대속도를 계산하고 이를 통해 적절한 운전 보조 시스템을 구현할 수 있을 것이다.

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] Andreas Geiger and Philip Lenz Karlsruhe Institute of Technology, Raquel Urtasun Toyota Technological Institute at Chicago, “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite ”

[3] 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

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

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