



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

석사학위논문

Convolutional Neural Network을
이용한 차량 인식

Vehicle Recognition using
Convolutional Neural Network

윤 성 민

한양대학교 공학대학원

2016년 8월

석사학위논문

Convolutional Neural Network을
이용한 차량 인식

Vehicle Recognition using
Convolutional Neural Network

지도교수 조 인 휘

이 논문을 공학 석사학위논문으로 제출합니다.

2016 년 8 월

한 양 대 학 교 공 학 대 학 원

컴퓨터공학 전공

윤 성 민



이 논문을 윤성민의 석사학위 논문으로 인준함

2016년 8월

심사위원장 : 이 병 호



심사위원 : 조 인 휘



심사위원 : 박 용 수



한양대학교 공학대학원

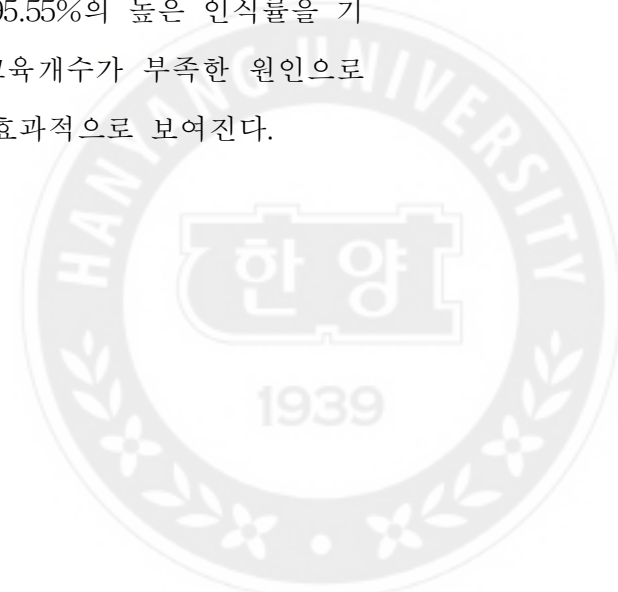


국문 요약

최근의 차량의 자율주행에 대한 사회적인 필요성이 가중되어지고 있다. 차량이 자율주행을 하기 위해 요구되는 이미지 내에서 차량의 인식과, 차선, 신호등, 교통 표지판 등의 인식이 필요하다. 실질적으로 이전에 사용하던 많은 방법들 또한 성능적으로 뛰어나지만 여러 가지 사물의 분류에 있어서 최근 가장 급부상하고 있는 머신러닝의 한부분인 딥러닝으로서의 Convolutional Neural Network을 이용하여 차량을 검출하는 시도를 해보고자 한다. 또한 CNN의 특성이 이미지의 분류에 적합한 형태이므로 차량과 일반사물을 함께 인식시켜서 그 분류성능을 강화시킨다. 이러한 기술의 발전에 따라 각 차종의 인식과 차량의 타입을 알아낼 수 있게되면 톨게이트의 차종에 따른 금액의 과금이 다른 업무를 전산화시의 업무적 효율성이 높아지게 되며 또한 공장 내의 생산품에 대해서도 수치가 아닌 시각으로 확인이 필요한 업무의 자동화가 가능하며 사람의 얼굴인식과 인식 후에 권한 부여 등으로 활용성이 높은 기술이라고 볼 수 있다.

본 연구에서는 그러한 업무의 초기적인 형태에서 차량의 전면과 후면을 신경망에게 학습시키고 가능성 검증하기 위하여 각각의 이미지 Dataset을 준비하고 그 이미지들의 학습방법과 횟수, 초기값 등을 측정하여 CNN의 사용이 유용한 방법임을 알아보고자 한다.

실험 결과로 전체 700개의 학습이미지에서 200개의 차량샘플 중 9개의 인식 오류가 발생하였고 이러한 결과는 차량에 대해서 95.55%의 높은 인식률을 기록하였다. 오류로 인식되는 이미지 또한 전체의 교육개수가 부족한 원인으로 파악되어지며 CNN을 이용한 차량의 인식은 매우 효과적으로 보여진다.



차 례

제 1 장 서론	1
제 2 장 관련연구	3
2.1. Scale Invariant Feature Transform(SIFT)를 이용한 차량인식	3
2.2. Haar-like를 이용한 차량검색	5
2.3. 차량검출(Vehicle Detection)	8
제 3 장 설계 및 구현	10
3.1. CNN의 기본 원리	10
3.2 구현	17
제 4 장 평가결과	25
제 5 장 결론 및 향후연구	32
Abstract	35
참고문서	37
감사의 글	40



그림 차례

그림 1. Difference of Gaussian, 이미지 마다 가우시안값을 다르게 만들어서 그 차이를 뺀다.[4]	3
그림 2. 이미지의 Gaussian를 다르게 주어서 차이를 보여주는 모습[5]	4
그림 3. SIFT방법으로 차량의 feature를 뽑아낸 모습[5]	5
그림 4. 좌측은 차량의 후보영역을 추출한것, 우측은 후보영역을 대상으로 차량의 light feature를 가지고 유효여부성을 판단후의 최종 차량인지한 이미지[17]	6
그림 5 적분의 특성을 이용한것으로 D의 Pixel의 합은 다른4개의 값을 이용해서 계산이 가능하다 [6][22]	7
그림 6 Haar-like Features들의 모습[22]	7
그림 7 대상이미지를 점점 작게 만드는 방법	8
그림 8 검출박스의 크기를 고정하고 대상이미지를 작게 만드는 방법 예시	9
그림 9 검출박스를 좌측에서 우측으로 이동하면서 이미지를 가져와서 비교하는 모습의 예시(offset 10pixel을 준 형태)	9
그림 10 차량의 이미지를 128×128의 크기로 만들어서 RGB색상으로 분리한후 ...	11
그림 11 필터를 적용시킨 모습	12
그림 12 차량이미지에 3×3 필터를 적용시킨 모습	12
그림 13 이미지에서 실제적으로 필터를 계산하는 모습[16]	13
그림 14 상단이미지 : 차량의 원본이미지의 RGB로 분리한 값,	13
그림 15 이미지RGB분리한 Byte배열에서 필터값을 계산해서 최종값을 얻는 계산의 형태(Zero Padding을 적용하며, 2 stride로 계산되어는 예시[9]	14
그림 16 ImageNet Classification with Deep Convolutional Neural Networks[10]에서 사용된 필터 11×11×3 사이즈	15
그림 17 Conv레이어를 쌓아가는 구조의 예시[9]	16
그림 18 본 연구에서 사용하는 구조의 예시[18]	16

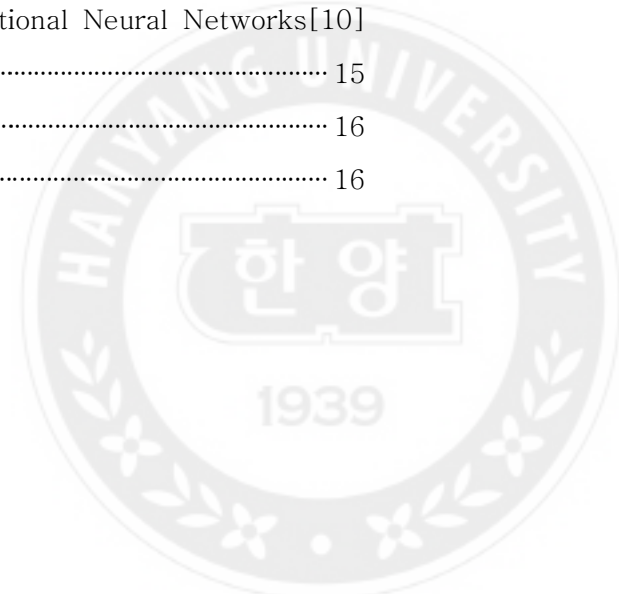


그림 19 얼굴의 Landmark를 판별할 때 사용한 레이어구조[19]	16
그림 20 신경망학습에 사용되어지는 120개의 차량정면이미지(부분)	18
그림 21 Dustbin이미지를 넣었을때의 교육과 테스트의 에러율이 줄어드는 형태[20]	19
그림 22 학습에 사용되는 차량의 뒷면 이미지	20
그림 23 학습에 사용되어지는 도로와 건물등의 차량근처의 이미지	21
그림 24 샘플이미지에게 제외한 경우들	22
그림 25 본 연구에서 사용되어진 레이어 구조 모습	22
그림 26 연구에 사용되어진 CNN 레이어 구조	23
그림 27 구현된 프로그램 모습	24
그림 28 학습의 반복회수에 따른 인식향상 그래프	29
그림 29 차량오류인식 개수	29
그림 30 인식오류개수가 축소되는 비교	29
그림 31 SGD Levenberg-Marquardt 95%인식에서 오류차량 9개	30
그림 32 SDG 93%인식에서 오류 차량 12개	30
그림 33 Multi-Channel형태의 CNN학습으로 Sobel필터를 이용하여 추출한 이미지를 대상으로 다시 CNN으로 학습후에 이 채널들을 합쳐서 인식처리하는 방식이 다. 이런 유사한 형태를 MCNN (Multichannel Convolutional Neural Network)이라고 할 수 있다.[14]	32
그림 34 Multi-Channel형태로 이미지를 각 파트형태로 나누어서 다시 Multi-Scale형태의 CNN구조를 만들어 학습시켜 처리하는 구조[15]	33



표 차례

표 1. 신경망 교육시의 초기값	25
표 2 SGD-Levenberg-Marquardt로 교육한 실험결과	25



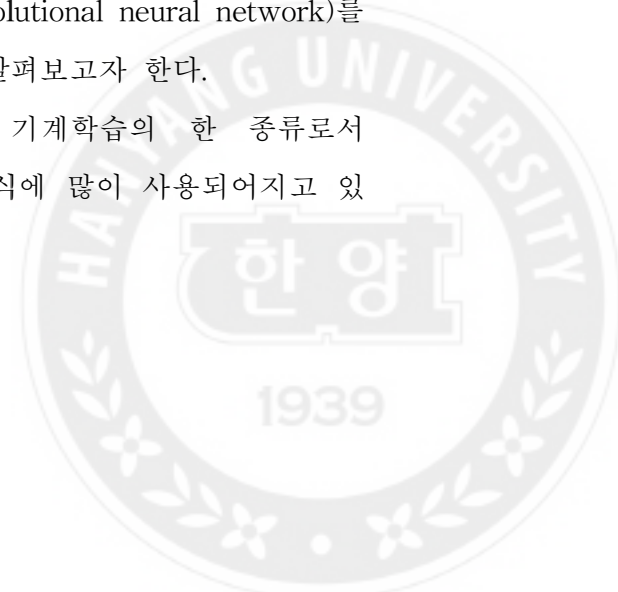
제 1 장 서론

컴퓨터가 이미지 내에 있는 객체를 인지할 수 있다면 실질적으로 많은 형태의 업무의 변화가 생길 것이다. 가장 기본적으로 현재에도 제공되어지고 있는 여러 포털사이트의 이미지 검색의 경우에 많은 이미지들 속에서도 키워드에 해당되는 이미지를 찾아서 보여주고 있다. 또한 스마트폰의 폭발적인 사용이 늘어나게 됨으로서 현대 사회는 방대한 이미지의 홍수 속에서 살아간다. 이미지 내에서 특정 물체를 인식하는 것은 매우 중요한 일이 되어지고 있으며 그러한 업무 중 하나인 이미지 내 차량인식을 시키는 것도 매우 활용성이 높은 업무 중 하나이다.

이미지 내에서 차량을 인식시키게 되면 차량의 종류, 차량의 번호판 인식등의 다음 업무가 진행이 될 수 있는 사항으로 차량의 인식은 매우 중요한 업무 중 하나이다. 기존의 차량번호판의 인식업무만 하더라도 차량을 검색하지 못하고 번호판만을 검색하기 위한 알고리즘을 사용하는 경우에 인식률이 많이 떨어지는 단점을 보완할 수도 있다.

차량이나 물체를 인식하기 위한 여러 가지 방법들이 존재하며 그 방법에 대해서 살펴보면 Scale Invariant Feature Transform(SIFT)을 이용한 방법[1], 색상을 이용한 방법[2], Haar-like를 이용한 방법[3], CNN (convolutional neural network)을 이용한 방법 등이 존재하며 각각의 방법들은 매우 훌륭하지만 상황에 따른 장단점을 가지게 된다. 본 연구는 객체분류에서 현재 가장 많이 사용되어지고 있는 딥러닝 기법인 CNN(convolutional neural network)를 사용해서 이미지 내에서의 차량을 인식에 대해서 살펴보고자 한다.

최근에 부상되어지고 있는 Deep learning은 기계학습의 한 종류로서 Supervised 적인 학습방법이다. 주로 이미지의 인식에 많이 사용되어지고 있



으며 현재 ImageNet Challenge의 상위권을 차지하고 있는 대부분의 알고리즘이 Deep learning을 이용한 알고리즘이며 그중 CNN의 방법이 가장 많이 이용되는 방법이다. 또한 그 활용방법이 이미지의 인식이외에도 음성인식과 여러 분야에 많이 사용되어지고 있다.

본 연구에서는 ImageNet에서 처럼 CIFAR-10 dataset으로 분류 클래스마다 6000개의 이미지가 존재하는 32×32 의 이미지 6만개의 이미지를 가지고 테스트를 할 수는 없지만 주변의 차량이미지를 직접 수집하여 Dataset을 만들어서 차량인식을 시도해 보고 그 결과를 비교해 보고 또한 차량의 전면부와 후면부의 이미지를 구분할수 있는지에 대한 테스트와 CNN의 신경망의 학습방법중에서도 SGD과 SGD Levenberg-Marquardt 방법을 통해서 학습을 해보고 그 결과를 비교해 보고자 한다.



제 2 장 관련연구

2.1. Scale Invariant Feature Transform(SIFT)를 이용한 차량 인식

SIFT를 이용한 방식은 실질적으로 매우 훌륭한 accuracy를 나타낸다. Scale의 변화나, 빛의 변화, 회전변형에 대해서 인지성이 강한 반응을 나타낸다. 단지 계산을 위한 CPU의 계산량이 많은 것이 단점으로 지적된다.

SIFT를 이용한 차량인식의 논문의 Mo, Guoliang, et al. "A Method of Vehicle Detection Based on SIFT Features and Boosting Classifier."의 내용을 일부분 이용해 보면

“By extracting SIFT features in the video frame and using the trained boosting classifier in vehicle detection, it does not require a fixed background and can be very good to deal with complex background, the brightness changes, scale changes, view changes and so on. Test on the data from UIUC database indicates that the vehicle detection accuracy rate can be up to 98%.”[1]

실질적으로 매우 훌륭한 형태의 인식을 보여주고 있다.

SIFT방식은 특징점을 추출하기 위해서 DOG(Difference of Gaussian)를 기반으로 다른 가우시안 필터를 적용시킨 상태에서 그 차이를 구해서 Keypoint를 구하게 된다.

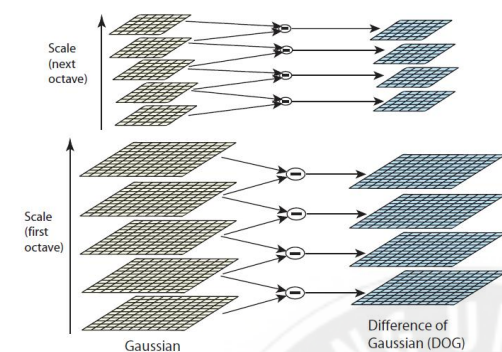


그림 2. Difference of Gaussian, 이미지마다 가우시안값을 다르게 만들어서 그 차이를 뺀다.[4]

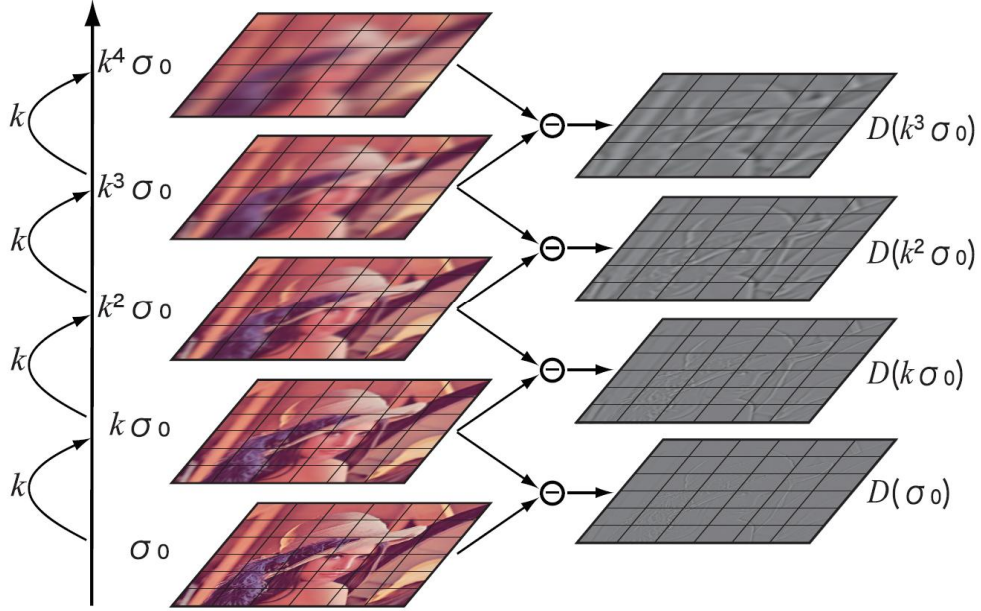


그림 3. 이미지의 Gaussian를 다르게 주어서 차이를 보여주는 모습[5]

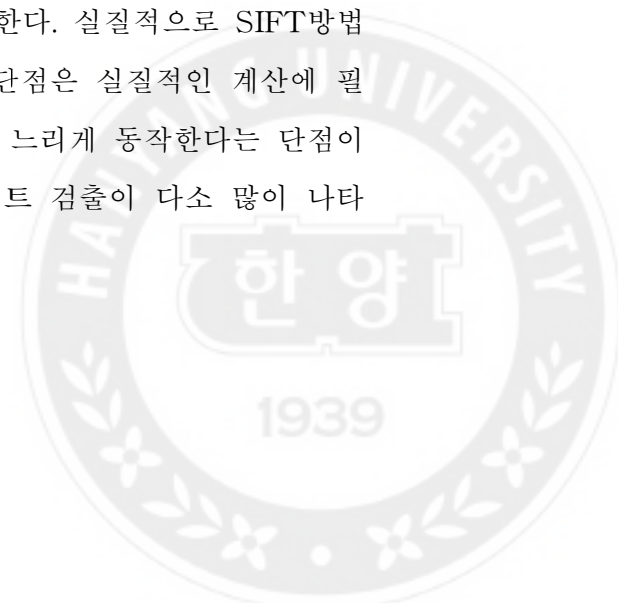
각 Keypoint를 구한 후 그 Keypoint의 방향성에 대한 정보를 구하는데 각 pixel간의 차이를 이용해서 그 방향성을 구한다.

Keypoint의 방향성을 구하는 수식[4] 은 아래와 같다.

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y)))$$

각 Pixel의 차이를 구해서 Keypoint의 값을 부여한다. 실질적으로 SIFT방법을 이용한 차량의 검색은 높은 정확성을 가진다. 단점은 실질적인 계산에 필요한 CPU파워가 많이 필요하게 되며 이로 인하여 느리게 동작한다는 단점이 존재한다. 또한 주변 환경에 따라 불필요한 키포인트 검출이 다소 많이 나타



나는 문제가 있다. 이러한 단점을 보완하기 위한 여러 가지 필터를 도입하거나 계산이 조금 더 빠른 SURF[13]의 (Speeded up robust features) 형태의 방식이 나와 있다.



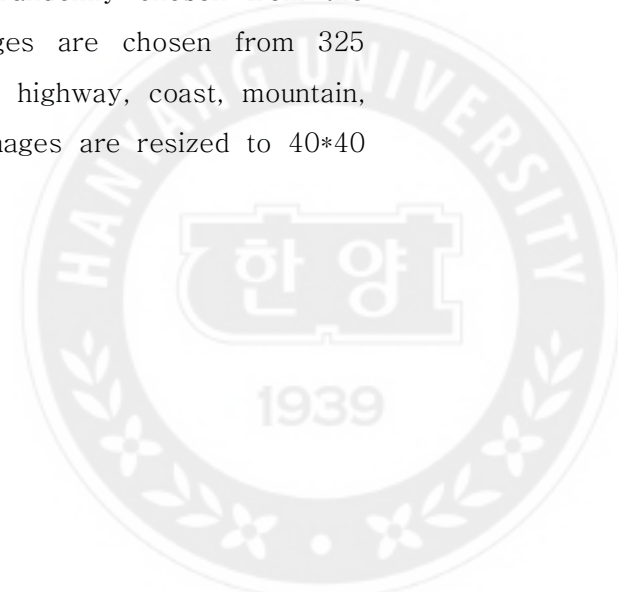
그림 3. SIFT방법으로 차량의 feature를 뽑아낸 모습[5]

2.2. Haar-like를 이용한 차량검색

Haar-like방식의 Adaboost 알고리즘의 경우 처음에 이미지 내에서의 얼굴탐색을 위해서 개발된 방법으로 현재로서도 얼굴 탐색시에 성능과 속도가 탁월하며 임베디드 형태의 기기에도 적용이 쉬운 방법으로 많이 사용되어지고 있다.[6]

기본적으로 학습에 필요한 Positive Sample이미지와 Negative Sample이미지를 가지고 교육을 시키며 이것을 여러번 학습시켜서 최적값을 추출해야 한다. Wang, Yufei. "Monocular Vehicle Detection and Tracking."의 논문을 인용해서 보면

"For cascade training, 2000 positive images are randomly chosen from the 1600-frame training dataset. 1300 negative images are chosen from 325 non-vehicle images, the scenes of which fall into highway, coast, mountain, open country, building, and street. The training images are resized to 40*40 patches."[17]



차량과 차량이 아닌 이미지들을 학습하는데 차량이 아닌 것은 도로, 산, 빌딩이나 거리 등의 이미지를 사용했다고 나온다.[17] 이러한 형태로 학습 후, 이것을 통해서 차량을 인식시키는데 실질적으로 대상영역의 오류인식률이 존재하므로 다시금 대상영역이 유효한 것의 여부를 확인하는 방법을 사용하고 있다.

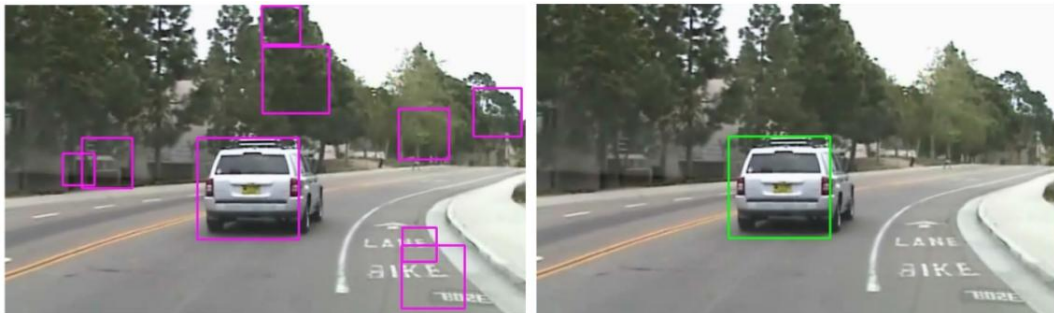


그림 4. 좌측은 차량의 후보영역을 추출한것, 우측은 후보영역을 대상으로 차량의 light feature를 가지고 유효여부성을 판단후의 최종 차량인지한 이미지[17]

“Haar-like의 특징은 임의의 영역간의 밝기 차이를 특징으로 한다. 영역간의 밝기차이를 구하기 위해 하나의 영역안의 밝기의 적분 값을 구하게 된다.”[7]

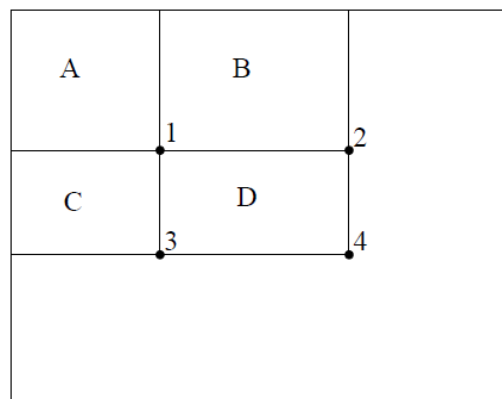


그림 5 적분의 특성을 이용한것으로 D의 Pixel의 합은 다른4개의 값을 이용해서 계산이 가능하다 [6][22]



위의 그림5에서 보여주고 있는 것은 1은 A영역의 합을 가지고 2는 A+B의 영역의 합을 가지고 3은 A+C의 영역의 합을 4는 A+B+C+D의 합을 가진다고 할때 여기서 사각형 D영역을 구할 수 있다는 것이다. 각각의 영역들의 $4+1-(2+3)$ 의 계산식으로 실제 계산을 처리하지 않고 D영역의 합산값을 구할 수 있으므로 좀 더 빠른 계산이 가능하다는 것이다.

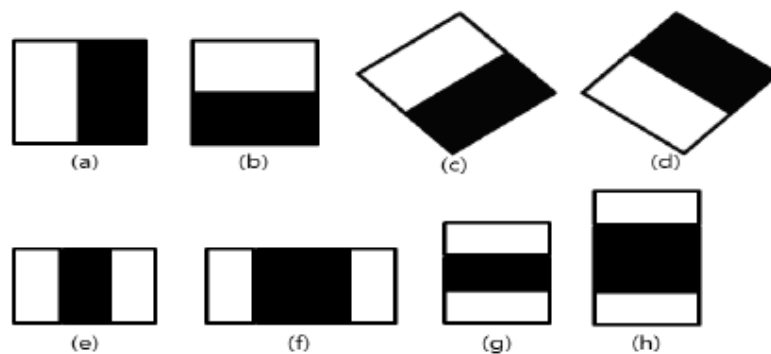
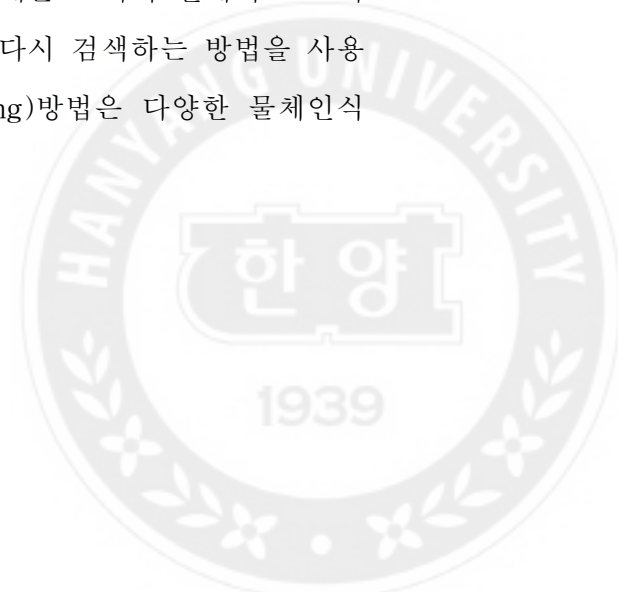


그림 6 Haar-like Features들의 모습[22]

이런 형태의 계산이 필요한 이유는 Haar-like가 그림 6처럼 특정한 영역간의 밝기차를 계산하여 이용하기 때문에 그 흰색영역과 검은색 영역의 pixel의 합을 구해서 계산해야 되기 때문이다.

Haar-like의 단점은 영역간의 밝기차를 이용하기 때문에 검출하고자 하는 물체가 영역간의 밝기차가 없거나, 회전변경이 많은 경우, 인식대상 이미지의 부분적인 그림자나 같은 부분적인 밝기 차이가 발생하는 경우 검출이 힘들어 질 수 있다. 하지만 뛰어난 검출속도를 지니고 있기 때문에 여러 가지 용도로 많이 사용되어지며 회전변형에 대해서는 취약한 형태를 보여서 실제적으로 회전변형을 검출하기 위해서 원본이미지를 회전하여 다시 검색하는 방법을 사용하기도 한다. haar-like Adaboost(Adaptive boosting)방법은 다양한 물체인식에는 한계를 지닌다.[21]



2.3. 차량검출(Vehicle Detection)

이미지 내에서 특정물체를 검출하기 위해서는 Sliding windows 방식으로 이미지를 처음부터 끝까지 살펴보는 방식이 있다. 이것도 두가지 형태가 존재하는데 인식할 수 있는 크기가 특정사이즈로 정해지는 경우 대상 이미지를 그대로 두고 검출 박스의 사이를 점점 키우면서 전체 이미지를 살펴보는 방식과 대상이미지를 점점 작게 하면서 검출박스의 크기는 동일하게 하는 방식이 있다. 전자의 경우 박스가 사이즈가 변경이 됨으로 인식사이즈로 변환하는 연산을 한번 더 해야 된다. 방법의 선택은 상황에 따라 필요한 형태를 사용한다.

이러한 대상이미지를 작게 만드는 방법을 Pyramid형태처럼 점점 작아지는 모습이라고 하여 Pyramid라고 호칭하며 대상이미지를 점점 작게 만들어서 검색하는 특징을 가진다.

Pyramid형태의 검출 방법은 대상이미지를 점점 작게 만들고 검출박스 사이즈는 고정크기로 검색을 계속하여 검출을 하는 방법으로 검출후에는 원본이미지의 위치값을 계산해야 되지만 검출박스의 사이즈를 변경하는것에 비해서 계산량이 작다.



그림 7 대상이미지를 점점 작게 만드는 방법





그림 8 검출박스의 크기를 고정하고 대상이미지를 작게 만드는 방법 예시

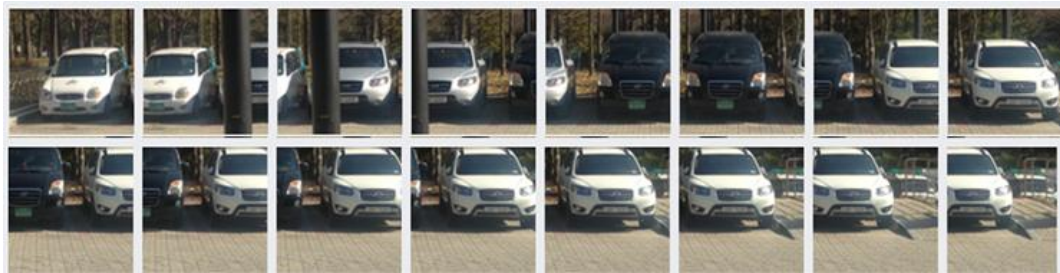


그림 9 검출박스를 좌측에서 우측으로 이동하면서 이미지를 가져와서 비교하는 모습의 예시(offset 10pixel을 준 형태)

CNN의 경우에는 이런 방법이 아닌 CNN의 특성을 이용해서 검색하는 방법이 존재한다[25].

박제강, et al. “딥러닝을 이용한 영상내 물체 인식 기법.”에 따르면 “일반적으로 CNN은 고정된 크기의 입력 데이터를 사용한다. CNN의 후반부에 위치한 완전연결 층이 고정된 크기의 입력을 받기 때문이다. 반면 컨볼루션 층은 입력 데이터의 크기에 따라 출력 데이터의 크기가 결정된다. 때문에 완전연결 층을 마스크 크기가 1인 컨볼루션 층으로 대체하면 입력 데이터의 크기 제한에서 자유로워진 CNN을 학습할 수 있다. 이러한 구조의 CNN은 학습에 사용한 영상보다 큰 영상을 입력하면 영상을 전역 탐색한 결과와 정확하게 동일한 결과를 출력한다.”[24]

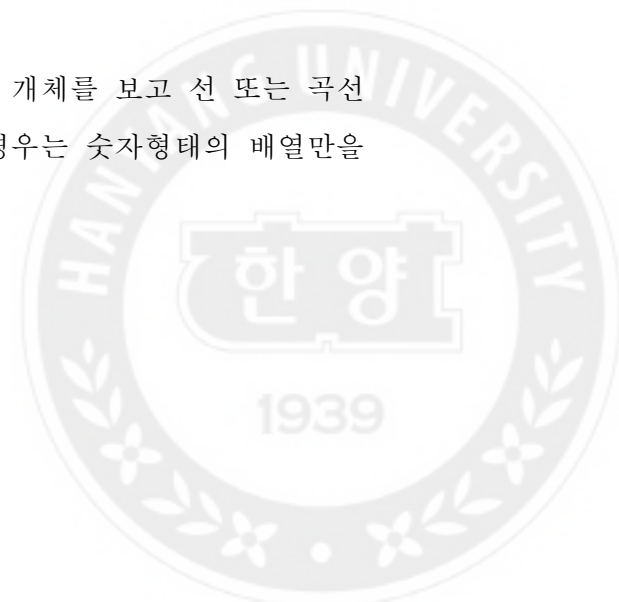


제 3 장 설계 및 구현

3.1. CNN의 기본 원리

현재의 Machine Learning의 여러 기법 중에서 CNN(convolutional neural network)은 Deep learning의 Supervised형태의 교육방식에 속한다. 인식시키고자 하는 이미지를 컴퓨터에게 학습을 시키는 것으로, 기존의 방식은 특징이 지를 벡터방식으로 인식시키거나 앞서 관련연구에 다룬 SIFT같이 Keypoint를 추출시켜 처리하는 것과는 다른 방식이다. 이미지의 형태 그대로의 pixel단위의 바이트배열을 그대로 학습시켜서 처리하는 것으로 학습에 따른 Layer구조를 설계하여 학습처리만 하게 되면 이미지를 인식처리할 수가 있다. SIFT방식의 경우에는 더 좋은 edge를 얻기 위하여 좀 더 명확한 Keypoint를 가지기 위해서 heuristic한 방법적으로 여러 가지를 시도하게 되지만 CNN의 경우는 Kernel을 제시하고 이러한 형태로 학습에 대한 방향성을 제시하는 형태이다. 또한 haar-like의 학습과 매우 유사하기도 하지만 그것과는 달리 CNN의 경우 최종출력단을 여러 개로 구별하도록 설계가 가능하므로 차량만을 인식하는 것이 아닌 다른 사물을 같이 인식 처리할 수 있다. 이것은 자율주행의 경우에 매우 장점이 되는 특성으로서 실질적으로 도로에서 다른 차량 뿐아니라 보행자, 자전거 등 여타 다른 사물을 인지하는 것이 중요한 문제이기 때문이다. 쉬운 예로 차량의 전면과 후면을 구분해야지만 도로상에서 반대편에서 오는 차량과 내가 가는 차선의 차량을 인지할 수 있기 때문이기도 하다.

인간이 물체를 인지하기 위해서는 물체의 각각의 개체를 보고 선 또는 곡선 등으로 형태를 인식할 수 있다. 하지만 컴퓨터의 경우는 숫자형태의 배열만을



접하게 된다(그림 5). 이러한 숫자형태의 배열값을 가지고 컴퓨터에게 개체를 인식시키기 위해서는 바이트 배열값을 특정의미를 가질 수 있도록 하는 것이 필요로 하다.

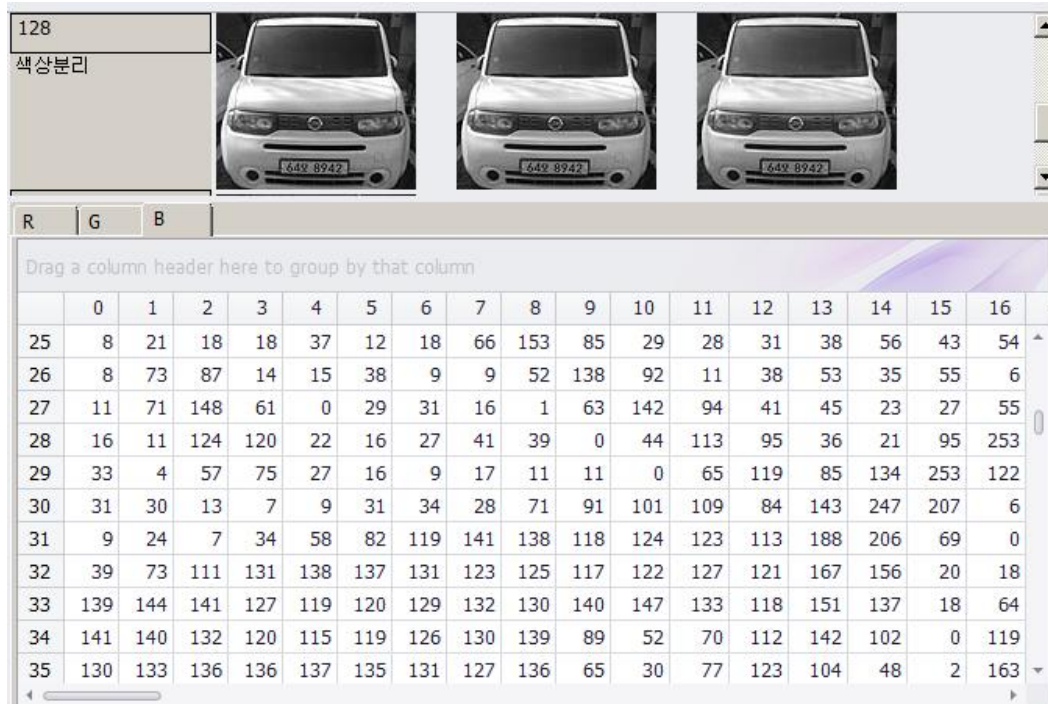


그림 10 차량의 이미지를 128×128의 크기로 만들어서 RGB색상으로 분리한후 Pixel의 Byte배열값을 표현한것

예를 들어서 특정한 이미지에 가로필터를 적용시킨다면 가로선만이 추출되게 될 것이다. 필터는 매우 다양하게 존재하며 수평마스크, 블록마스크, 에지 검출 필터 등 매우 다양한 것이 존재한다. 필터가 가지는 이러한 특징을 이용해서 CNN에서는 하나의 이미지를 여러 가지 필터를 적용해서 많은 이미지를 만들어낸 다음 그것을 신경망에다가 교육을 하여 이미지가 가지는 특징을 인식시키도록 하는 방법이다.



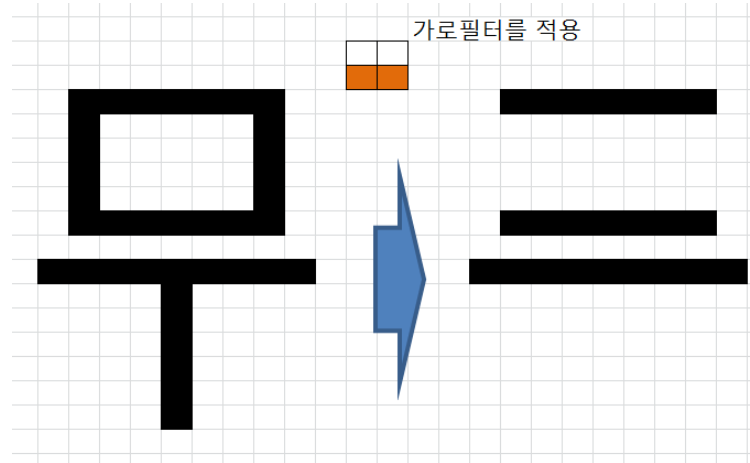


그림 11 필터를 적용시킨 모습

본 연구에서 사용되어지는 차량이미지에 필터를 적용시킨 것을 살펴보면 다음과 같다.

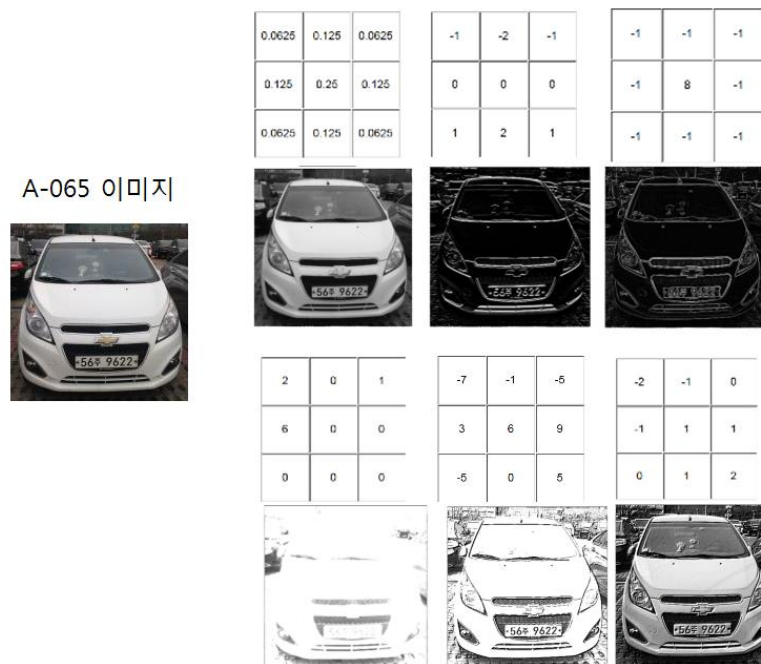


그림 12 차량이미지에 3×3 필터를 적용시킨 모습



위의 그림은 본 연구에서 사용되어지는 65번 차량이미지로서 이것에 특정한 형태의 필터를 적용시키게 되면 예시와 같은 형태의 이미지가 나오게 된다.

이미지에 필터를 적용시키는 형태를 CNN에서는 Image kernel이라는 용어를 사용하며 이렇게 연산하는 방법은 유사한 형태이다.

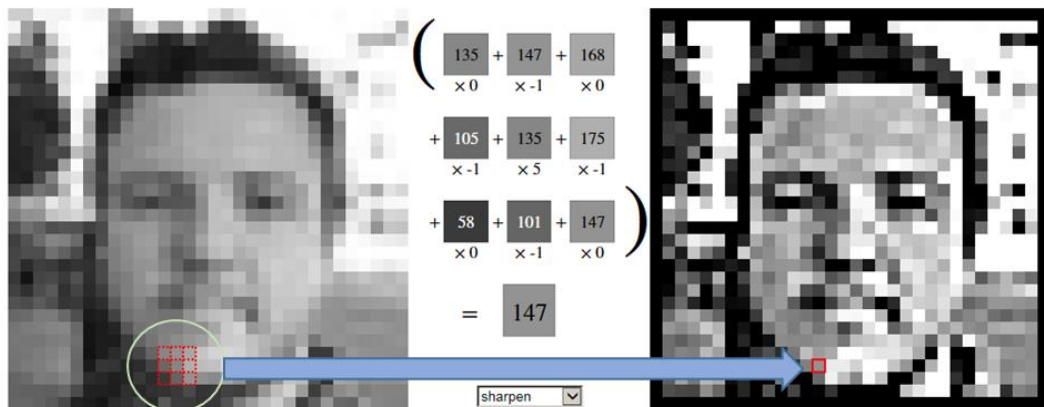


그림 13 이미지에서 실제적으로 필터를 계산하는 모습[16]

좀 더 상세한 실제의 Conv Layer의 형태를 표현해 보면 다음과 같다.

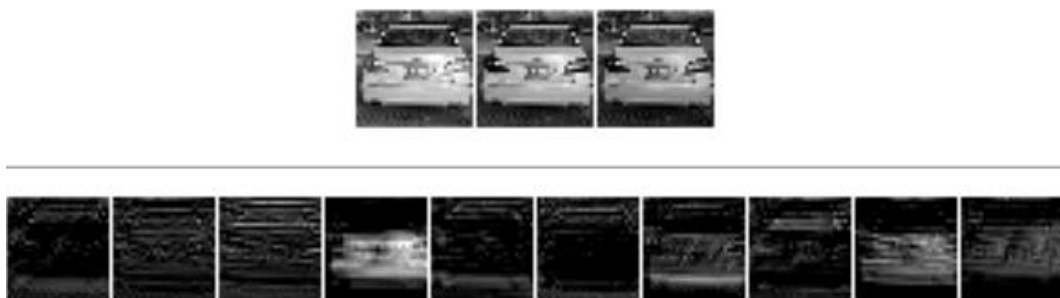


그림 14 상단이미지 : 차량의 원본이미지의 RGB로 분리한 값,
하단이미지 : 차량을 64개의 Conv Layer로 처리한 값의 일부분



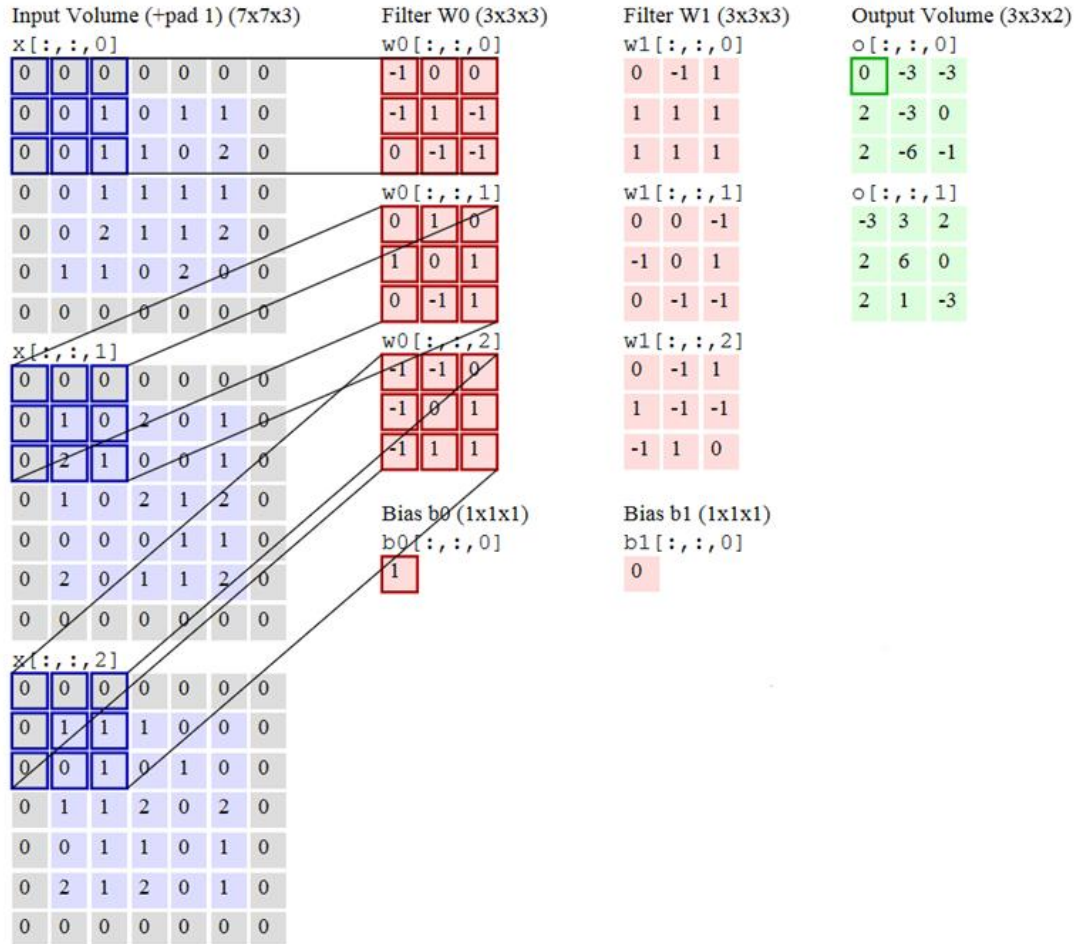


그림 15 이미지RGB분리한 Byte배열에서 필터값을 계산해서 최종값을 얻는 계산의 형태 (Zero Padding을 적용하며, 2 stride로 계산되어는 예시[9])

여기에서 이미지의 RGB로 분리한 바이트배열이 3개이며 동시에 필터를 적용하는 배열도 3개임을 알아야 한다. 이러한 방식으로 여러 개의 필터를 적용해서 여러 개의 이미지를(Activation maps) 얻을 수 있다. 이때에 각 맵을 만들 때의 가중치와 바이어스를 포함하는 형태로 계산되어야 한다. 일반적으로 이런 형태로 계산되어지는 것을 Conv Layer라고 말하며 CNN의 구조는 Full

Connection(FC), Convolutional Layer(Conv Layer), Pooling(Pooling layer) 등의 형태로 구성이 되어진다.

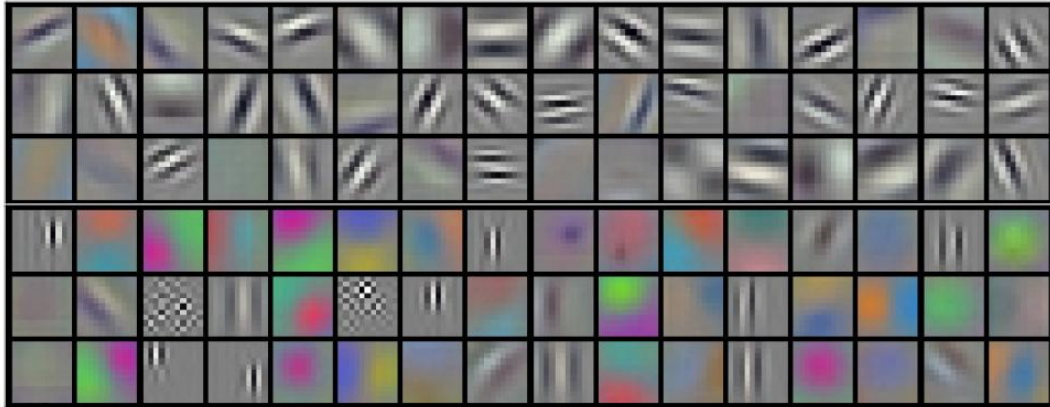


그림 16 ImageNet Classification with Deep Convolutional Neural Networks[10]에서 사용된 필터 11×11×3 사이즈

이러한 형태로 계산이 되어진 Conv레이어를 가지고 다시금 Conv레이어로 반복을 처리하여서 여러개의 Activation map을 만들 수 있다. 여기서 이 레이어들의 구조를 어떻게 쌓는 것에 따라서 CNN의 성능이 달라지게 된다. 이러한 구조들의 몇몇 유명한 것들을 살펴보면

- * LeNet[11] 으로서 코드나 숫자를 읽는 구조로 유명하다.

- * AlexNet[10]은 ImageNet ILSVRC challenge 2012에서 1등을 ~~한~~ 구조로 유명한 것으로 LeNet과 유사한 구조이지만 더 깊고 큰 형태를 가지고 있다. 227×227×3 의 형태로 CNN 구조에서는 매우 큰 사이즈로 신경망을 교육시켰다.

그 외에도 ZF net, GoogLeNet, VGGNet, ResNet등의 여러 가지 ConvNet들이 존재한다.



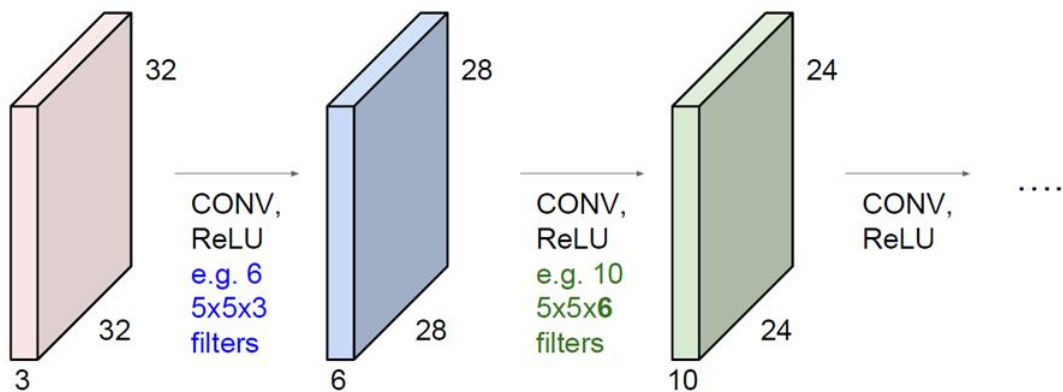


그림 17 Conv레이어를 쌓아가는 구조의 예시[9]

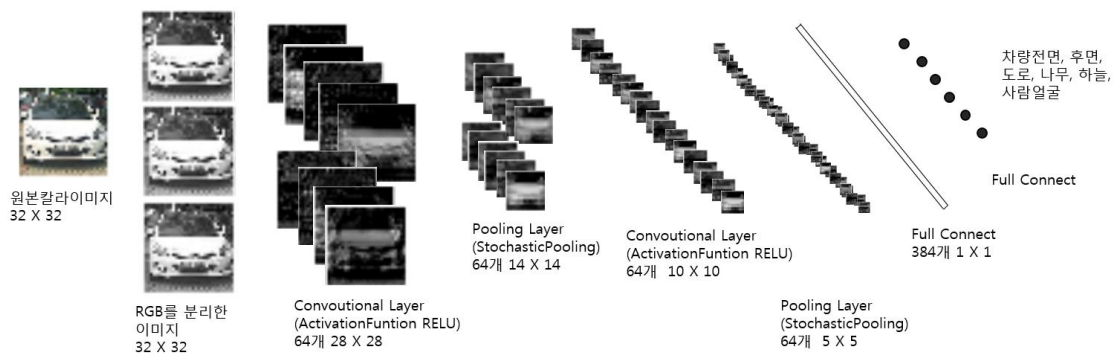


그림 18 본 연구에서 사용하는 구조의 예시[18]

CNN의 형태의 사용은 매우 범용적으로 늘어나고 있으며 글자인식이나 사람의 얼굴의 Landmark를 인식하여 현재웃는표정인지, 성별, 또는 안경을 쓰고 있는지도 판별이 가능한 수준으로 발전되고 있다.[19]

이러한 형태의 판별을 위해서 실질적으로 동일한 형태의 ConvNet과 Pooling Layer등을 교차 반복시켜서 접근이 가능하며 이러한 결과값을 통해서 특정한 형태의 이미지를 판별하는 것이 가능하다.



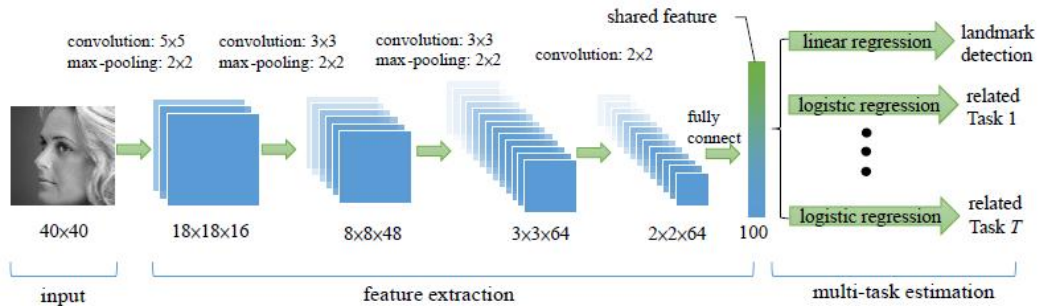


그림 19 얼굴의 Landmark를 판별할 때 사용한 레이어구조[19]

3.2 구현

구현에 따른 준비에 있어서 가장 기본적인 Dataset이 필요하다. 이 데이터셋의 구조에 따라 성능도 달라지게 되므로 신중해야 한다. 우선, 데이터셋을 구성하기 위해서 주차장에 있는 차량을 주로 대상으로 했으며 주변의 마트, 병원, 학교, 거리 등의 외부 주차장에 속한 차량을 대상으로 사진촬영을 했다. 가능한 낮시간 대의 밝은 상태에서 차량을 촬영하였다.

이미지의 경우 차량의 종류도 매우 다양하며 사이드미러를 접은 차량, 불필요한 안내판이 들어간 차량, 주차장에서 다른차량과 중복으로 이미지내에 보여지는 차량, 해상도가 다르게 촬영된 형태, 낮과 밤의 차이로 인한 노출의 차이, 차량에 그림자가 드리워져서 반만 어둡고 반은 밝은 형태, 차선이 사진내 포함된 경우, 차량에 표기된 문자이자가 많은 경우, 사진의 각도 등의 아주 다양한 형태의 문제점들이 내포되어 있다.

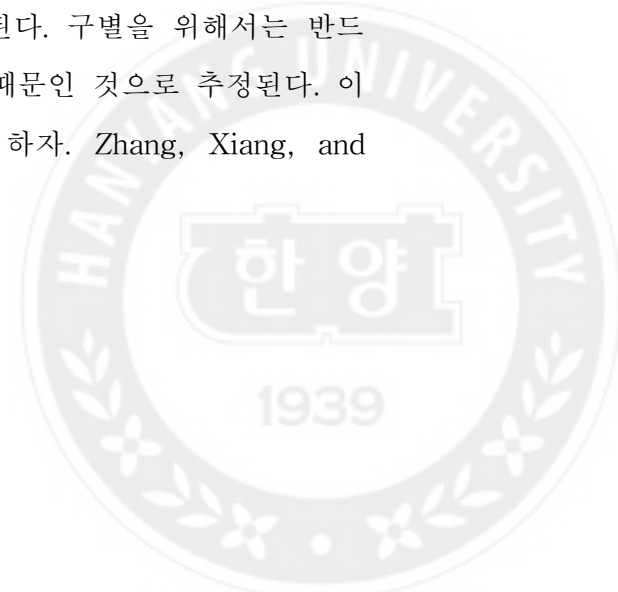
일단은 기본적인 눈높이 시점에서 촬영 데이터를 만들었으며 아주 번잡한 이미지가 내포된 경우는 제외시키도록 했다. 이미지의 사이즈는 32×32 로 변환해서 교육했으나 점차 여러 가지 조건별로 데이터의 구조를 변경해야 되는 관계로 기본적인 이미지를 1024×1024 사이즈로 만들어 놓은 상태에서 이것

을 학습시점에 썸네일 형태로 변환하여 다시금 필요한 사이즈인 32×32 또는 64×64 의 사이즈로 자동으로 소스에서 변환시키도록 하였다.



그림 20 신경망학습에 사용되어지는 120개의 차량정면이미지(부분)

각 이미지들에게는 순서적인 형태의 번호를 부여하였다. 초기에는 단지 차량의 앞면과 뒷면만을 준비했으나 실질적으로 CNN에게 차량만을 교육시키게 되면 다른 이미지와의 구별을 처리할 수가 없게 된다. 구별을 위해서는 반드시 그에 비교될 수 있는 이미지도 존재해야 되기 때문인 것으로 추정된다. 이것과 유사한 결론을 가지는 논문을 살펴보도록 하자. Zhang, Xiang, and



Yann LeCun. "Univsum Prescription: Regularization using Unlabeled Data." 의 논문 에 나타나는 그래프를 참조해 보면[20] 아래와 같다.

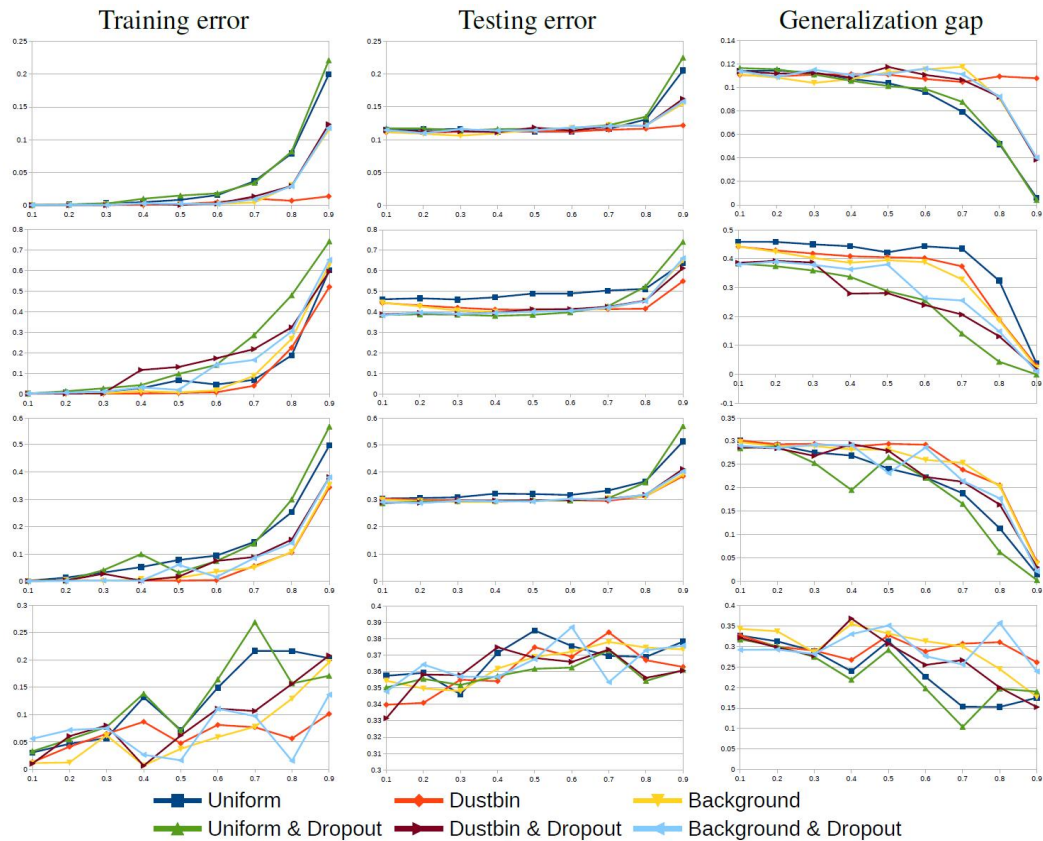
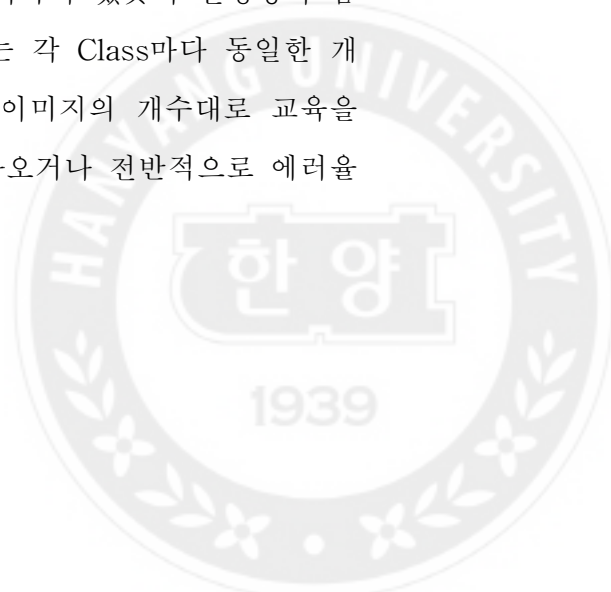


그림 21 Dustbin이미지를 넣었을때의 교육과 테스트의 에러율이 줄어드는 모습[20]

그러므로 추가적으로 차량이미지의 근처에 해당되는 도로, 나무, 하늘, 그리고 확연한 구별이 가도록 사람의 얼굴을 이미지셋이 더 넣어서 교육을 시켰다. 이러한 이미지를 더 넣어서 사용한 것은 앞서 이야기 했듯이 신경망이 좀 더 차량을 명확하게 구분하기 위해서 이며 교육시는 각 Class마다 동일한 개수를 교육하는 것이 중요하다. 초기에는 무턱대고 이미지의 개수대로 교육을 했으나 개수가 많은 이미지에 대해서 수치가 높게나오거나 전반적으로 에러율



이 상승하는 경험을 하였다. 그래서 각각의 이미지를 교육용을 100개, 테스트 용을 20개로 고정해서 실험을 진행하였다.

전체적으로 $120 \times 6 = 720$ 개의 이미지를 이용해서 신경망을 학습과 평가를 시켰으며 부분적으로 몇몇 이미지는 preprocessing관점에서 제거하고 교육했으며 실험결과를 관리하기 위해서 DB를 이용 하여 데이터와 Tag정보를 입력 하여 관리하고 그 Tag정보는 이미지의 전면, 후면, 도로, 하늘, 나무, 사람얼굴 인 구별정보와 테스트 그룹, 차량의 대형차, 트럭, 승용차의 구별 정보를 관리 했다.



그림 22 학습에 사용되는 차량의 뒷면 이미지



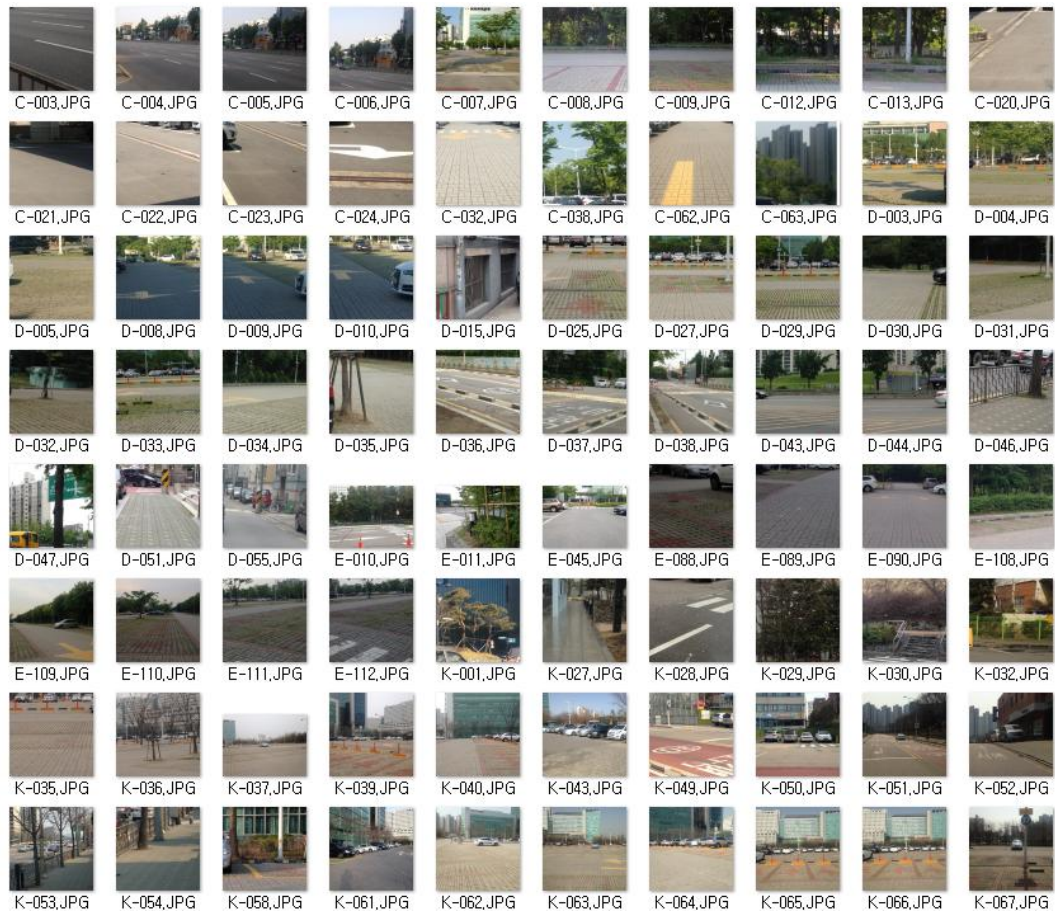


그림 23 학습에 사용되는 도로와 건물등의 차량근처의 이미지

교육에 사용되는 이미지들은 되도록 글자가 많거나 햇빛에 대한 번들임이 매우 심하거나, 차종이 너무 차이나는 형태, 사람이 같이 이미지에 존재하는 경우, 하나의 이미지에 차량이 복수형태로 들어가는 경우 등의 몇가지에 대해서는 제외했다.



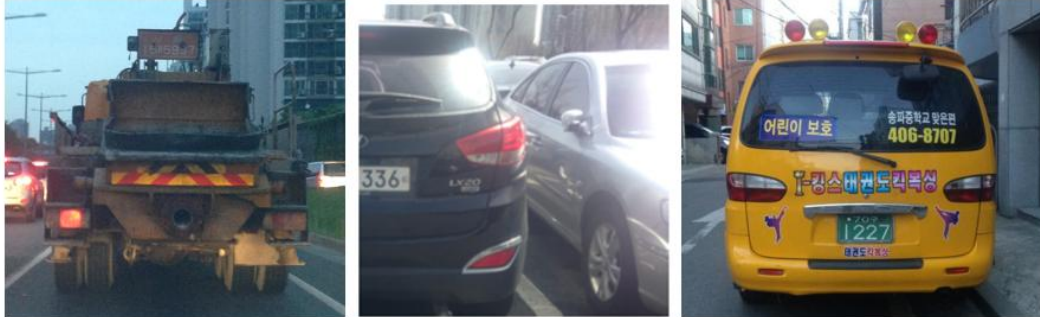


그림 24 샘플이미지에게 제외한 경우들

레이어의 기본구조도는 CNN의 공개된 소스[18]를 사용해서 테스트했으며 이 구조에서 Pooling Layer를 MaxPoling으로 변경하거나, 64×64 로 변경 또는 레이어의 구조의 변경등을 진행하면서 처리했다. 하지만 본 연구에서는 32×32 의 형태만 살펴보도록 하겠다.

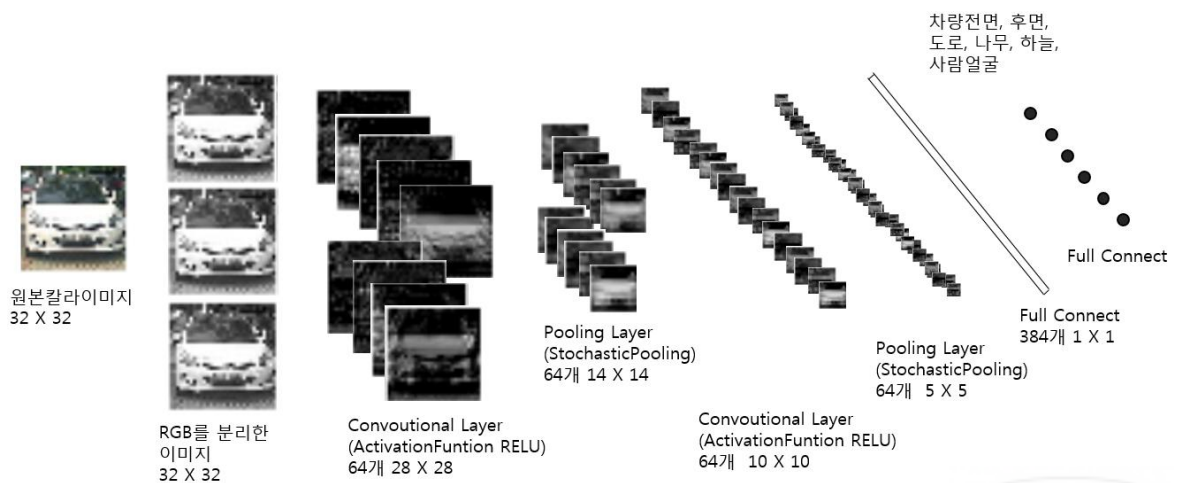
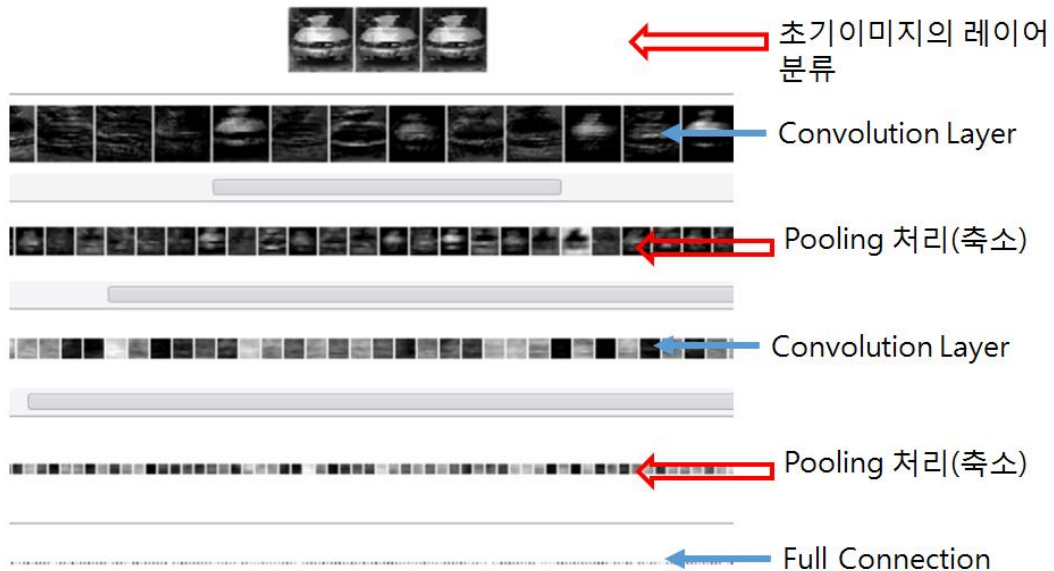


그림 25 본 연구에서 사용되어진 레이어 구조 모습



CNN Layer 구성

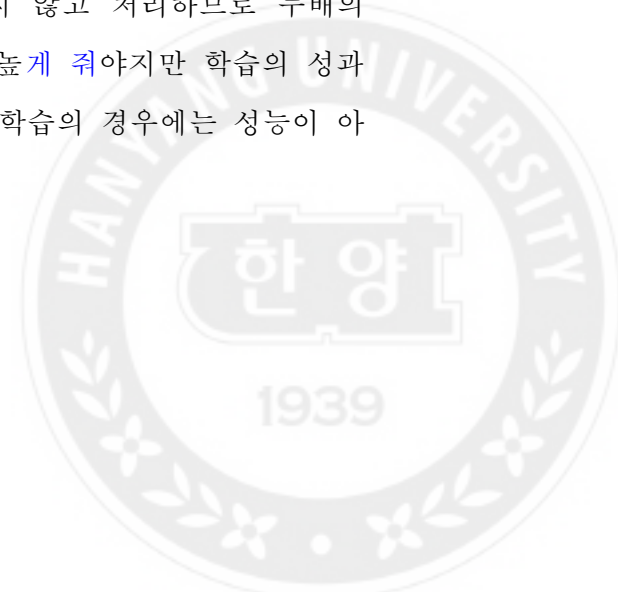


0 정면, 1 뒷면, 2 도로, 3 하늘, 4 나무, 5 사람얼굴

그림 26 연구에 사용되었던 CNN 레이어 구조

이미지를 RGB레이어의 Byte배열로 분리한 다음 최초의 ConvLayer 의 개수는 64개로 ZeroPadding이 없이 진행했다. 기계학습의 방법은 여러 가지가 존재하는데 SGD(Stochastic Gradient Descent)[23]의 Hessian이 없는 상태로 교육이 가능한 방법과, SGD Levenberg-Marquardt 방법으로 비선형 최소자승법 형태의 교육 방법으로 Hessian을 만들어야 되는 두가지로 진행해 보았다.

실질적으로 SGD[23]의 경우 Hessian을 생성하지 않고 처리하므로 두배의 속도로 학습이 가능하다. 하지만 초기 학습비율을 높게 줘야지만 학습의 성과가 나타나며 속도적인 면에서는 뛰어나나 현 차량 학습의 경우에는 성능이 아주 뛰어나지는 않는듯 보여지는 결과가 나왔다.



실질적인 구현처리에 대해서 서술하면 프로그램의 구현언어는 Microsoft의 C#을 이용했으며 Windows내에서 실행되는 프로그램으로 작성하였다. 기본적인 CNN엔진의 학습과, 처리는 GitHub에 공개된 라이브러리를 이용해서 교육하도록 하였다.[18]

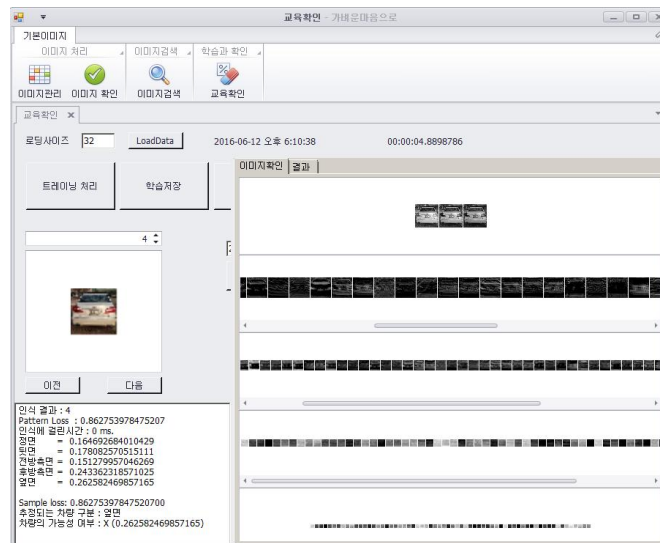


그림 27 구현된 프로그램 모습

학습은 일반적인 인텔 i7 CPU 수준에서 대략 기본적인 50~70회 교육시에 대략 4시간 가량이 소요되었다.



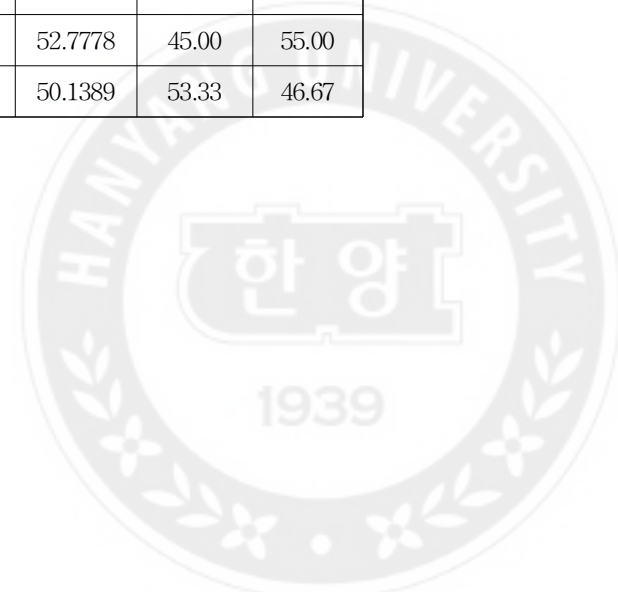
제 4 장 평가결과

	학습방법	학습비율 초기값	가중치 감소율	변형률
1	SGD Levenberg-Marquardt	0.0005	0.785	20%
2	SGD Levenberg-Marquardt	0.0005	0.915	7%~20%
3	SGD[23]	0.0050	0.950	7%~12%
4	SGD	0.0050	0.950	20%~25%

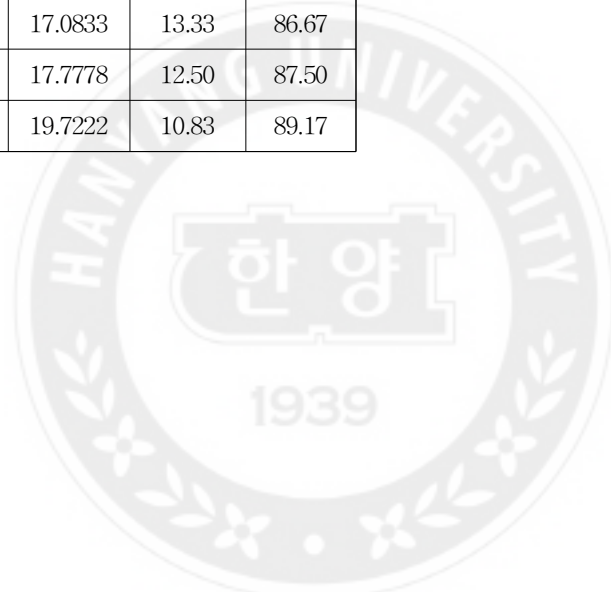
표 1. 신경망 학습시의 초기값

대략의 수치는 표1의 형태로 주었으며 이중에서 하나의 실험 결과를 살펴보도록 하자. SGD-Levenberg-Marquardt의 두 번째 실험결과의 경우.

	변형율	학습가중치	학습 Error Count	Test Error Count	평균 학습 Lose	평균 Test Loss	Train Error Rate	Test Error Rate	정확도
1	20	0.000500	598	100	0.9016	0.8800	83.0556	83.33	16.67
2	20	0.000457	564	100	0.8900	0.8614	78.3333	83.33	16.67
3	20	0.000417	529	67	0.8723	0.7702	73.4722	55.83	44.17
4	20	0.000381	470	62	0.8262	0.7115	65.2778	51.67	48.33
5	20	0.000347	437	65	0.7758	0.6827	60.6944	54.17	45.83
6	20	0.000317	401	65	0.7366	0.7374	55.6944	54.17	45.83
7	20	0.000290	380	54	0.7091	0.5903	52.7778	45.00	55.00
8	20	0.000264	361	64	0.6865	0.6199	50.1389	53.33	46.67

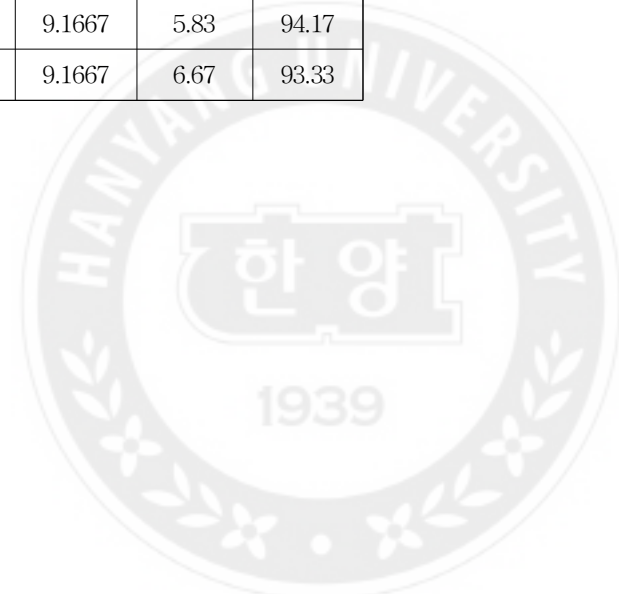


9	20	0.000241	349	58	0.6642	0.6049	48.4722	48.33	51.67
10	20	0.000220	328	52	0.6435	0.5418	45.5556	43.33	56.67
11	20	0.000201	313	49	0.6339	0.5249	43.4722	40.83	59.17
12	20	0.000184	300	48	0.6119	0.5131	41.6667	40.00	60.00
13	20	0.000168	299	49	0.6053	0.4534	41.5278	40.83	59.17
14	20	0.000153	289	48	0.5925	0.4696	40.1389	40.00	60.00
15	20	0.000140	291	45	0.5787	0.4481	40.4167	37.50	62.50
16	20	0.000128	265	37	0.5677	0.4293	36.8056	30.83	69.17
17	20	0.000117	283	43	0.5599	0.4039	39.3056	35.83	64.17
18	20	0.000106	253	39	0.5522	0.4066	35.1389	32.50	67.50
19	20	0.000097	261	36	0.5419	0.4071	36.2500	30.00	70.00
20	20	0.000089	247	33	0.5373	0.3868	34.3056	27.50	72.50
21	20	0.000500	272	44	0.5445	0.4214	37.7778	36.67	63.33
22	20	0.000457	252	23	0.5135	0.3322	35.0000	19.17	80.83
23	20	0.000417	229	33	0.4853	0.4081	31.8056	27.50	72.50
24	20	0.000381	206	28	0.4691	0.3504	28.6111	23.33	76.67
25	20	0.000347	195	19	0.4485	0.2675	27.0833	15.83	84.17
26	20	0.000317	171	25	0.4379	0.3155	23.7500	20.83	79.17
27	20	0.000290	176	26	0.4228	0.2925	24.4444	21.67	78.33
28	7	0.000500	167	32	0.4203	0.3448	23.1944	26.67	73.33
29	7	0.000457	169	18	0.4026	0.2574	23.4722	15.00	85.00
30	7	0.000417	149	18	0.3850	0.2262	20.6944	15.00	85.00
31	7	0.000381	138	16	0.3695	0.2550	19.1667	13.33	86.67
32	7	0.000347	142	27	0.3581	0.3508	19.7222	22.50	77.50
33	7	0.000317	123	16	0.3469	0.2341	17.0833	13.33	86.67
34	7	0.000290	128	15	0.3413	0.2372	17.7778	12.50	87.50
35	20	0.000500	142	13	0.3534	0.2272	19.7222	10.83	89.17



36	20	0.000457	130	21	0.3352	0.2698	18.0556	17.50	82.50
37	20	0.000417	128	21	0.3276	0.2630	17.7778	17.50	82.50
38	20	0.000381	126	13	0.3176	0.1906	17.5000	10.83	89.17
39	20	0.000347	106	18	0.2977	0.2815	14.7222	15.00	85.00
40	20	0.000317	110	12	0.2975	0.1701	15.2778	10.00	90.00
41	20	0.000290	106	12	0.2891	0.2061	14.7222	10.00	90.00
42	7	0.000500	117	13	0.2957	0.1736	16.2500	10.83	89.17
43	7	0.000457	105	10	0.2852	0.1742	14.5833	8.33	91.67
44	7	0.000417	94	12	0.2723	0.1930	13.0556	10.00	90.00
45	7	0.000381	101	10	0.2715	0.1877	14.0278	8.33	91.67
46	7	0.000347	88	17	0.2585	0.2666	12.2222	14.17	85.83
47	7	0.000317	95	14	0.2571	0.1670	13.1944	11.67	88.33
48	7	0.000290	86	11	0.2471	0.1665	11.9444	9.17	90.83
49	7	0.000264	78	11	0.2400	0.1545	10.8333	9.17	90.83
50	7	0.000241	86	11	0.2378	0.1848	11.9444	9.17	90.83
51	7	0.000220	80	9	0.2426	0.1382	11.1111	7.50	92.50
52	7	0.000220	84	8	0.2365	0.1363	11.6667	6.67	93.33
53	7	0.000201	82	6	0.2323	0.1332	11.3889	5.00	95.00
54	7	0.000184	83	11	0.2239	0.1616	11.5278	9.17	90.83
55	7	0.000168	79	6	0.2220	0.1424	10.9722	5.00	95.00
56	7	0.000153	76	11	0.2174	0.1664	10.5556	9.17	90.83
57	7	0.000140	75	10	0.2131	0.1524	10.4167	8.33	91.67
58	7	0.000128	75	9	0.2113	0.1367	10.4167	7.50	92.50
59	7	0.000117	65	13	0.2105	0.1994	9.0278	10.83	89.17
60	7	0.000106	66	7	0.2120	0.1294	9.1667	5.83	94.17
61	7	0.000097	66	8	0.2106	0.1335	9.1667	6.67	93.33

표 2 SGD-Levenberg-Marquardt로 교육한 실험결과



도표에서 실질적으로 트레이닝에서의 최저 66개의 이미지의 에러를 보여주고 있는데 차량은 9개의 이미지가 오류이며 나머지 57개는 교육을 위한 불필요한 이미지로서 이미지내의 정보의 일관성이 부족한 형태로 오류율이 높다고 보여진다. 학습가중치는 초기에 높게 주고 천천히 감소시키는 방식으로 학습을 진행하였다.

학습에 있어서는 학습의 가중치값과 그 가중치가 점점 감소해야 되는 비율을 지정하게 되는데 이 수치를 통해서 반복학습시의 학습의 정도를 제어하기 위함이다. 변형은 기본적인 이미지만을 교육하는 것이 아닌 원본이미지의 각도나 평형변형 등을 이용해서 이미지를 비틀어서 교육하는 방식이다. 대략적으로 7~20%정도의 변형이 무난했으며 70~100%변형의 경우는 좋은 결과값이 나타나지 않았다. 이러한 변형되어진 형태로 교육시키는게 실질적으로 학습후의 인식결과가 높아진다고 하는 연구들이 존재하며 실질적으로 차량의 사진을 찍을 때나 인식할 때의 이미지의 각도가 변형되어짐 존재하기 때문에 약간의 변형을 통한 교육이 인식률의 성능이 좋아진다고 고려된다. 이와 유사한 연구로서 신경망에 학습에 Dustbin을 넣어야 된다는 연구도 존재한다.[20]

간단한 비교를 위한 테스트의 결과에 따르면 CNN의 학습방법과 가중치, 감소율, 변형률, 교육이미지의 해상도등 많은 변수들에 의해서 학습의 형태가 달라진다. 하지만 인식률의 정확도가 특정수준까지 오르기 위해서는 대략 60~70회 수준의 반복이 필요했으며 이러한 형태는 가중치가 다름에도 유사하게 나타났다. 또한 일정한 수준의 이상에서는 계속 교육을 진행하더라도 과최적화의 방지형태로 인해서 인식률이 떨어졌다가 다시 특정수치로 회귀하는 모습을 보여준다.



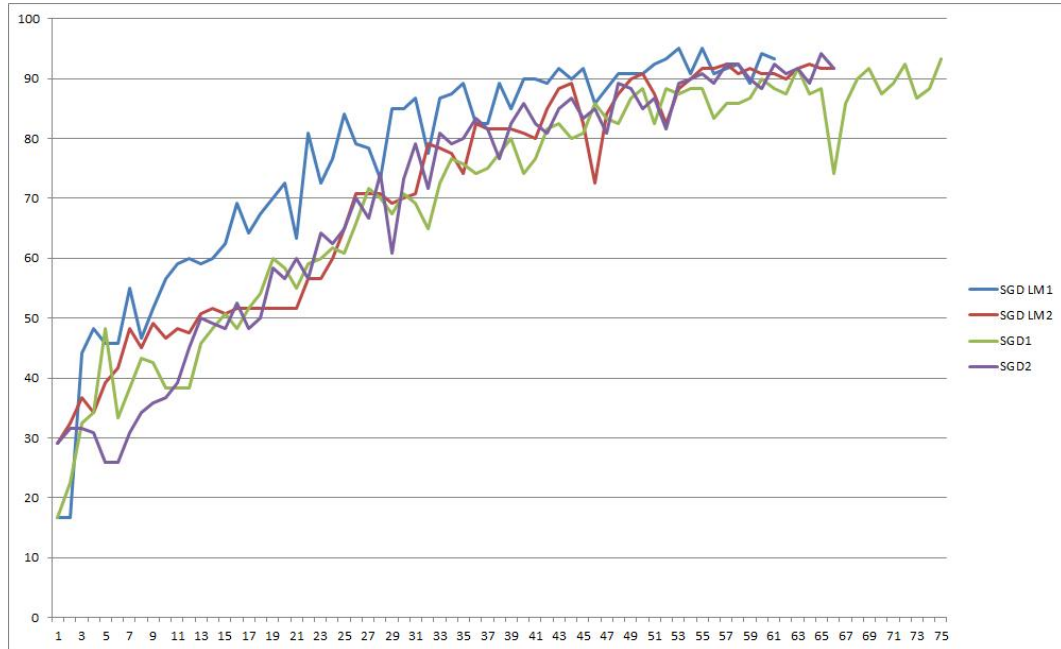


그림 28 학습의 반복회수에 따른 인식향상 그래프

실험의 경우의 일관성이 없는 불특정한 나무, 도로, 하늘, 사람 얼굴 등의 교육으로 인해서 전반적으로 오류률이 높게 나오지만 실질적으로 차량만을 가지고 비교할 때는 높은 형태의 인식률을 보여주고 있다.

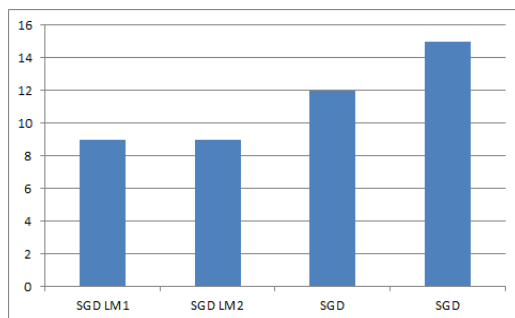


그림 29 차량오류인식 개수

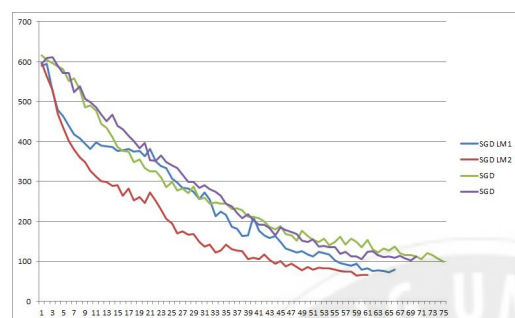


그림 30 인식오류개수가 축소되는 비교



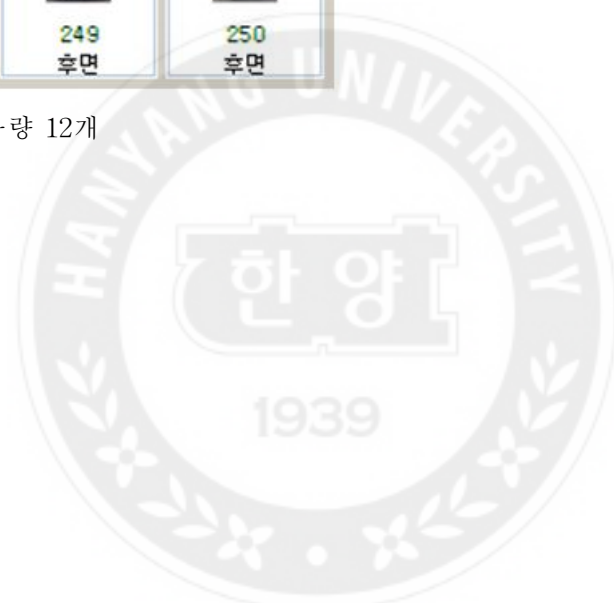
몇 번의 테스트를 진행해 본 결과 특히 인식이 잘 되지 않는 차량이 나타났는데 그 리스트는 아래와 같다. 이러한 오류가 나타나는 것은 근본적으로 학습에 필요한 최초 데이터셋의 부족으로 추정된다. 학습방법과 무관하게 오류 형태의 차량이 유사함을 보여주고 있는데 차량에 광량이 많이 들어가 반사가 심한경우와 차량이 다른 차량과 모형이 많이 다른경우 이러한 형태의 교육데이터가 많지 않아서 인식오류가 생긴다고 판단된다.



그림 31 SGD Levenberg-Marquardt 95%인식에서 오류차량 9개



그림 32 SDG 93%인식에서 오류 차량 12개



이것의 해소방법은 현 연구가 각 클래스마다 100개의 교육데이터와 20개의 테스트 데이터가 가지는 한계라고 판단이 되며 타 논문을 비교할 때 최소 각 클래스마다의 천여장이상의 이미지를 교육해야 될 것으로 예상된다. 다른 논문에서 표지판을 인식시키기 위하여 교육시킨 샘플의 개수를 보면 상당한 양의 이미지를 이용해서 교육을 하며 박제강, et al. "딥러닝을 이용한 영상내 물체 인식 기법."의 논문을 인용해 보면 "표지판 샘플이 약8만장, 배경 샘플이 약 80만장으로 충분한 수의 샘플로 학습"[24] 하였다는 형태가 보인다. 오류인식의 차량들의 경우 샘플의 이미지가 현저히 부족한 문제로 인해서 학습되어진 정보가 부족한 이유로 인식의 오류가 발생하는 것으로 추정된다.



제 5 장 결론 및 향후연구

차량을 인식하는데는 많은 기술적인 방법이 존재하지만 자율 주행과 같은 경우에 전면에 있는 모든 사물을 인식해야지만 그 활용성이 뛰어나다고 할 수 있다. 다른 방법의 경우는 인식률이 매우 높게 나타나며 또한 여러 가지 방법을 통해서 에러의 교정을 시도할 수 있고 속도적인면도 뛰어나지만 단점으로 단편적인 차량만을 인식하거나 차량의 전면, 측면, 후면 등의 구별이 용의하지 못하거나 또한 다른 사물과의 복합적인 인식이 불가하거나 인식을 위한 다른 방법을 시도하게 되면 성능적으로 매우 부적합하게 되는 경향이 발생한다.

CNN의 경우에는 각각의 Layer의 구성이나 신경망의 학습방법, 학습가중치, 검색방법, 그리고 최종적으로 교육시킬 많은 데이터셋과 여러 변수를 제어해야 되는 어려움, 교육에 따른 학습시간이 오래걸린다는 단점은 존재하나 그 결과는 실로 매력적인 형태를 나타내고 있다. 인식의 속도나, 사물의 검색속도도 상당히 훌륭한 것으로 향후의 차량과 사물의 인지에 가장 유용한 방법으로 고려된다.

CNN을 통한 차량인식의 테스트의 결과에서 완전한 인식의 결과를 도출하지는 못하였지만 실질적으로 이는 교육의 데이터셋이 부족한 형태로 보여지며 실질적인 차량인식 방법으로서는 신뢰도 높은 방법이라고 여겨진다. CNN의 경우에 차량뿐만 아닌 여러 가지로 객체를 인식하는데 있어서 탁월한 성능을 보여주고 있다. 현재의 연구들은 CNN의 레이어 인식을 Multi-Scale과 Multi-Channel[14][15]형태로 변경해서 신경망에 학습시키는 방식들이 연구되어지고 있다.



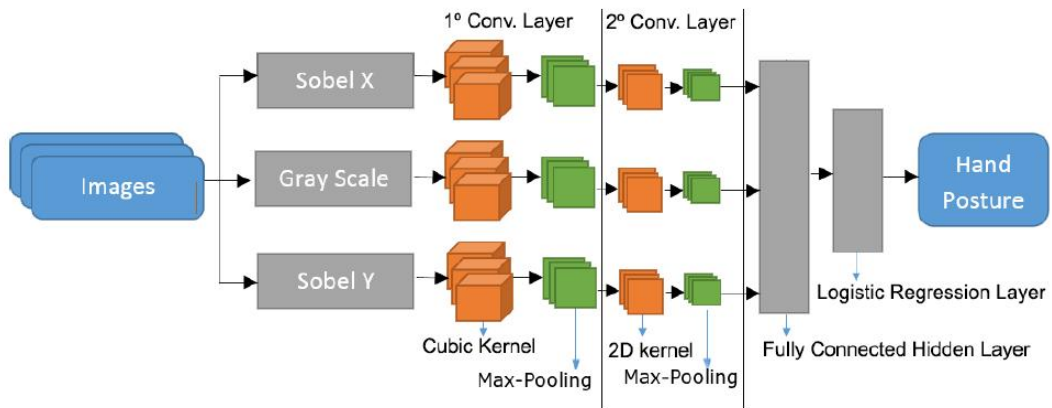


그림 33 Multi-Channel형태의 CNN학습으로 Sobel필터를 이용하여 추출한 이미지를 대상으로 다시 CNN으로 학습후에 이 채널들을 합쳐서 인식처리하는 방식이다. 이런 유사한 형태를 MCNN (Multichannel Convolutional Neural Network)이라고 할 수 있다.[14]

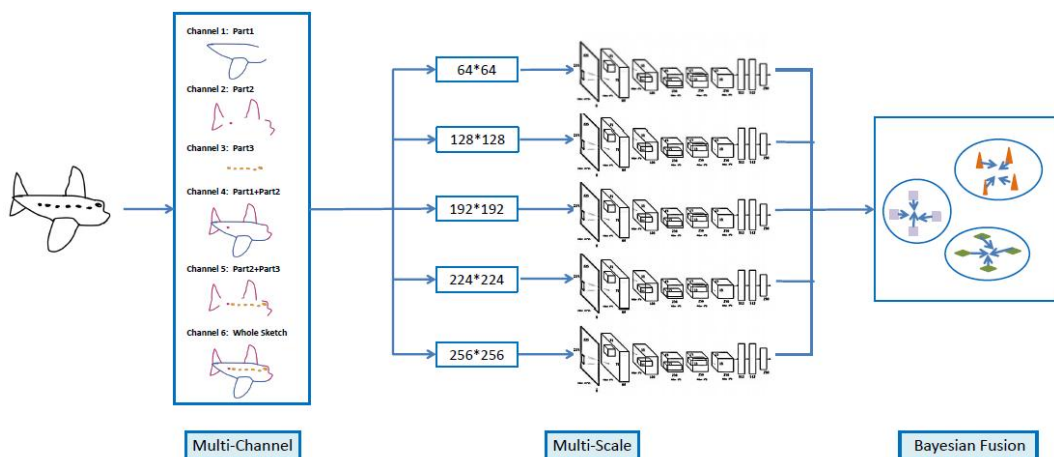


그림 34 Multi-Channel형태로 이미지를 각 파트형태로 나누어서 다시 Multi-Scale 형태의 CNN구조를 만들어 학습시켜 처리하는 구조[15]

본 연구의 학습Data의 형태를 사람의 얼굴이나 또는 다른 객체로 변경 시에도 유사한 결과를 얻을 것으로 예측된다. 향후의 연구는 CNN의 경우 개체인 식으로의 성능이 탁월한 알고리즘이므로 이것을 이용한 Multi Channel방식으



로의 학습을 통해서 좀 더 정밀한 차량의 정확한 차종을 판별하거나 또는 이
미지내에 있는 특정동물이 있을 때 동물인 것을 판별 하는 것보다 한발 나아
가서 그 동물의 품종을 알아낼 수 있는 연구를 해보고 싶다. 좀 더 정밀한 형
태의 판별이 가능하도록 컴퓨터에게 학습을 시키려면 어떻게 해야 하는지에
대해서 더 많은 연구가 필요하다. 또한 더 높은 해상도를 통해서 학습시켰을
경우와, polling layer를 없도록 학습시키는 경우등 아직은 CNN에서 변경해
봐야 되는 변수가 많이 존재하고 있다. 이러한 연구를 위해서는 가장 중요한
것이 교육에 필요한 데이터셋이 필요하며 다음 연구에서는 충분한 데이터셋을
가지고 이미지 인식에 대한 연구를 진행해 보고자 한다.



Abstract

Vehicle Recognition using Convolutional Neural Network

Yun, Sung Min

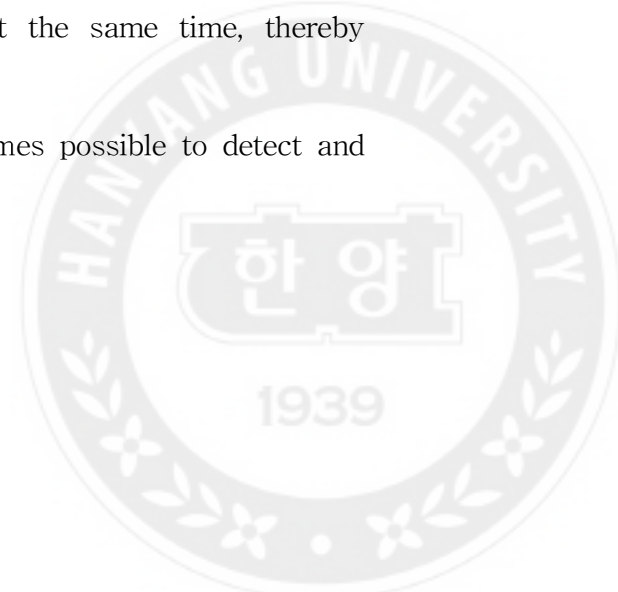
Major in Computer Science and
Engineering

The Graduate School of Engineering
Hanyang University

Recently there has been an increase in social demand for autonomously driving vehicles. Autonomous driving requires images from which to detect vehicles, road markings, traffic signals and traffic signs.

There have been many methods with similar performances. In this study, I try to find a way to detect vehicles using the Convolutional Neural Network(CNN), which is a deep running method, or one of the machine running methods. This method recently emerged quickly in terms of classifying objects. In addition, CNN has the proper means of classifying images, it detects vehicles and other objects at the same time, thereby strengthening its classification capability.

If this kind of technology is developed, it becomes possible to detect and



discover the type of vehicle. It would be very beneficial for tollgates to impose fees in an electronic way, thereby enhancing work efficiency. Also, if products produced inside a factory were recognized visually instead of by numerical identification, it would be possible for production to become automatic. As well, it may also grant appropriate authorization to a person since it recognizes the person's face. This technology can be seen as one with many possibilities and uses.

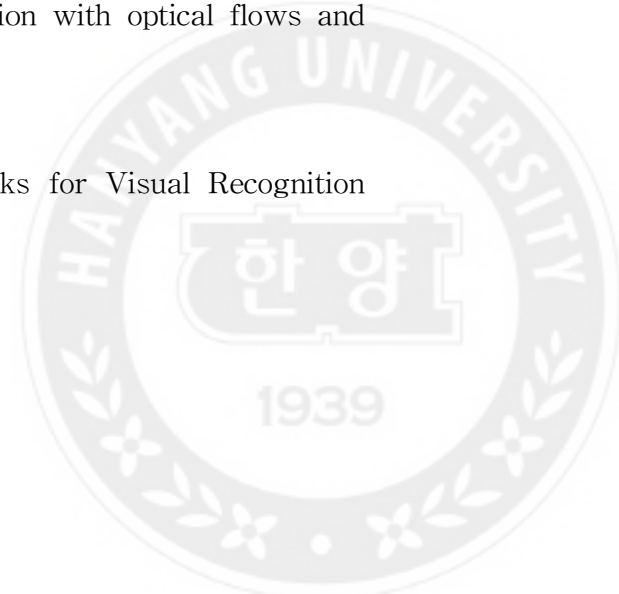
In this study, I first allowed the neural network to study the front and rear sides of the vehicles in their initial form. Then in order to test the possibility, every image dataset was prepared, and the methods of study, frequency, initial values was measured to find out whether CNN is a useful method.

Findings from the tests show that there were 9 errors from 200 samples which were taken from 700 educational images, therefore demonstrating a high recognition rate of 95.55%. Since error is attributed to the lack of education images, it appears that CNN is very effective in detecting vehicles.



참고문서

- [1] Mo, Guoliang, et al. "A Method of Vehicle Detection Based on SIFT Features and Boosting Classifier." *Journal of Convergence Information Technology* 7.12 (2012).
- [2] Gao, Lei, et al. "Vehicle detection based on color and edge information." *Image Analysis and Recognition* (2008): 142-150.
- [3] Han, Sungji, Youngjoon Han, and Hernsoo Hahn. "Vehicle detection method using Haar-like feature on real time system." *World Academy of Science, Engineering and Technology* 59 (2009): 455-459.
- [4] Lowe, David G. "Distinctive image features from scale-invariant keypoints." *International journal of computer vision* 60.2 (2004): 91-110.
- [5] Fujiyoshi, Hironobu. "Gradient-based feature extraction: SIFT and HOG." *Information Processing Society of Japan Research Paper CVIM160* (2007): 211-224.
- [6] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Vol. 1. IEEE, 2001.
- [7] 조경순, et al. "Haar-Like 특징을 이용한 고성능 보행자 및 차량 인식 회로 설계." *정보처리학회논문지 A* 19.4 (2012): 175-180.
- [8] Choi, Jaesik. "Realtime on-road vehicle detection with optical flows and Haar-like feature detectors." (2012).
- [9] 인터넷 자료
Stanford CS321n Convolutional Neural networks for Visual Recognition

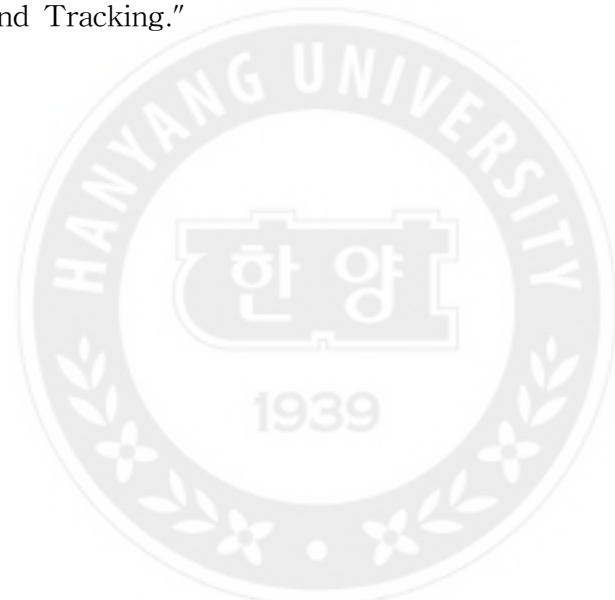


의 강의 내용

자료출처 : <https://cs231n.github.io/convolutional-networks/>

자료검색일 : 2016.05

- [10] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
- [11] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.
- [12] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
- [13] Bay, Herbert, et al. "Speeded-up robust features (SURF)." Computer vision and image understanding 110.3 (2008): 346-359.
- [14] Barros, Pablo, et al. "A multichannel convolutional neural network for hand posture recognition." Artificial Neural Networks and Machine Learning - ICANN 2014. Springer International Publishing, 2014. 403-410.
- [15] Yu, Qian, et al. "Sketch-a-net that beats humans." Proceedings of the British Machine Vision Conference (BMVC). 2015.
- [16] 인터넷 자료
자료출처 : <http://setosa.io/ev/image-kernels/>
자료검색일 : 2016.05
- [17] Wang, Yufei. "Monocular Vehicle Detection and Tracking."
- [18] 인터넷 자료
자료출처 : <https://github.com/supby/cnnwb>
자료검색일 : 2016.05



- [19] Zhang, Zhanpeng, et al. "Facial landmark detection by deep multi-task learning." Computer Vision - ECCV 2014. Springer International Publishing, 2014. 94-108.
- [20] Zhang, Xiang, and Yann LeCun. "Universum Prescription: Regularization using Unlabeled Data." arXiv preprint arXiv:1511.03719 (2015).
- [21] Kwak, Ju-Hyun, Il-Young Woen, and Chang-Hoon Lee. "Learning Algorithm for Multiple Distribution Data using Haar-like Feature and Decision Tree." KIPS Transactions on Software and Data Engineering 2.1 (2013): 43-48.
- [22] 안경준, and 이상용. "Haar-like 와 베지어 곡선을 이용한 얼굴 성분 검출." 디지털융복합연구 11.9 (2013): 311-318.
- [23] Kiros, Ryan. "Training neural networks with stochastic hessian-free optimization." arXiv preprint arXiv:1301.3641 (2013).
- [24] 박제강, et al. "딥러닝을 이용한 영상내 물체 인식 기법." 제어로봇시스템 학회지 21.4 (2015): 21-26.
- [25] Sermanet, Pierre, et al. "Overfeat: Integrated recognition, localization and detection using convolutional networks." arXiv preprint arXiv:1312.6229 (2013).



감사의 글

삶을 살아가는데 있어서 가족만큼 중요한 사람들이 없을 듯하다. 그 존재만으로도 나에게 힘이 되어주는 사람들 덕분에 현재의 위치에 지금의 모습으로 살아가는 것이 아닌가하고 감사한다. 언제나 나의 정신적인 기둥이 되어준 윤희철과 나보다 앞서 석사, 박사과정을 통과한 윤은주, 윤희성, 그리고 모든 자녀들이 석사과정 이상을 공부하도록 항상 뒤에서 지켜봐주시고 지원해주신 어머니님, 아버지께 진심으로 감사를 드린다.

부족하고 모자란 연구를 좀 더 가능성 있게 봐주시고 격려해주신 조인휘 교수님이 계셨기에 지금 이 자리까지 올 수 있었으며, 동기들과 함께 연구를 진행하면서 큰 힘을 얻었다. 아마도 혼자였다면 여기까지 오지 못했을 것이다. 혼자라면 빨리 가지만 함께 가면 멀리 간다는 말처럼, 모든 여정을 동기들과 함께 했기에 이루게 된 성취라고 생각한다.

업무와 학업을 병행해야 하는 쉽지 않은 과정을 거치면서, 인간으로서 좀 더 성장하고 성숙해진 본인을 되돌아보게 된다. 다시금 논문을 쓸 기회가 주어진다면 스스로에게 좀 더 발전된 모습을 보여주고 싶다. 여러모로 지원해준 Heather에게도 감사의 마음을 전한다.

감사의 마음을 담아

2016. 8.



연구 윤리 서약서

본인은 한양대학교 대학원생으로서 이 학위논문 작성 과정에서 다음과 같이 연구 윤리의 기본 원칙을 준수하였음을 서약합니다.

첫째, 지도교수의 지도를 받아 정직하고 엄정한 연구를 수행하여 학위논문을 작성한다.

둘째, 논문 작성시 위조, 변조, 표절 등 학문적 진실성을 훼손하는 어떤 연구 부정행위도 하지 않는다.

셋째, 논문 작성시 논문유사도 검증시스템 "카피킬러"등을 거쳐야 한다.

2016년06월14일

학위명: 석사

학과: 전기·전자·컴퓨터공학과

지도교수: 조인휘

성명: 윤성민



한 양 대 학 교 공 학 대 학 원 장 귀 하



Declaration of Ethical Conduct in Research

I, as a graduate student of Hanyang University, hereby declare that I have abided by the following Code of Research Ethics while writing this dissertation thesis, during my degree program.

"First, I have strived to be honest in my conduct, to produce valid and reliable research conforming with the guidance of my thesis supervisor, and I affirm that my thesis contains honest, fair and reasonable conclusions based on my own careful research under the guidance of my thesis supervisor.

Second, I have not committed any acts that may discredit or damage the credibility of my research. These include, but are not limited to : falsification, distortion of research findings or plagiarism.

Third, I need to go through with Coppykiller Program(Internet-based Plagiarism-prevention service) before submitting a thesis."

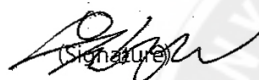
JUNE 14, 2016

Degree : Master

Department : DEPARTMENT OF ELECTRICAL, COMPUTER ENGINEERING

Thesis Supervisor : INWHEE JOE

Name : YUN SUNGMIN


(Signature)

