

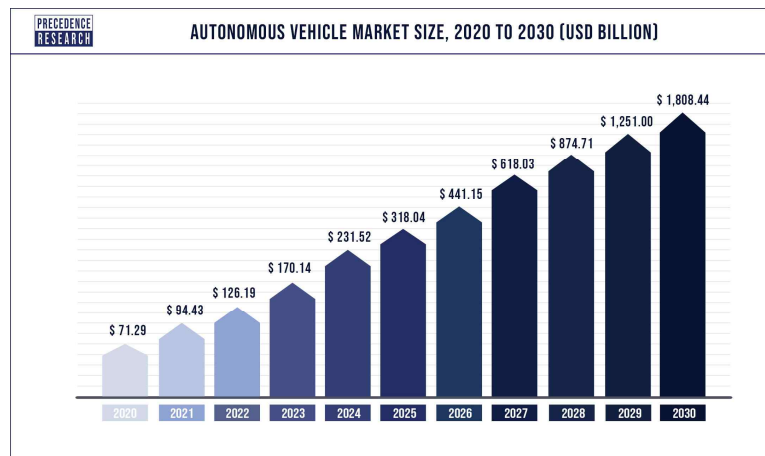
빅데이터 분석 사례
자율주행차량의 차선 인식

<목차>

1. 주제 선정 이유	3
2. 차선 인식 문제란?	4
3. 초창기의 차선 인식 접근 방법	5
3.1. 특징 (Feature)	6
3.2. 순수 만들어진 특징(Handcrafted feature)	6
3.3. 소벨(Sobel) 필터와 허프(Hough) 변환을 이용한 차선 인식	6
3.4. 초창기 차선 인식 접근 방법의 한계와 의의	7
4. 현재의 차선 인식 방법	7
4.1. 데이터 세트	7
4.1.1. TuSimple	8
4.1.2. CULane	9
4.1.3. BDD100K	10
4.2. LaneNet (with H-Net)	10
4.3. 인공지능망 (Artificial Neural network)	11
4.3.1. 심층 신경망 (Deep neural network)	13
4.3.2. 합성곱 신경망 (Convolution neural network, CNN)	15
4.3.3. RGB 이미지에서의 필터와 합성곱 연산	20
4.3.4. 이미지 분할 (Image segmentation)	23
4.3.5. 차선 인식 신경망의 구조	24
4.4. 차선 인식 기술의 전망	27
5. SWOT 분석	28
5.1. Strength	28
5.2. Weakness	28
5.2.1. 해결 방안	28
5.3. Opportunity	29
5.4. Threat	29
5.4.1. 해결 방안	29
6. 참고자료	30

1. 주제 선정 이유

최근 자동차 산업에서는 "자율주행"이라는 단어가 화제이다. 자율주행의 개념은 1960년 자동차 회사 벤츠(Benz)를 중심으로 제안된 오래된 개념이지만, 2010년부터 컴퓨터 성능, 센서 기술, 인공지능 기술 등의 발전에 힘입어 현재인 2022년에는 운전자 보조 주행 기능이나 유의미한 수준의 자율주행 기능이 탑재된 상용화된 제품이 시장에 출현하고 있다. 급속한 발전 속에서 세계 자율주행차량 시장의 크기는 2030년까지 연평균 38.8%씩 성장할 것으로 전망된다.



[그림 1] 2020년부터 2030년까지의 세계 자율주행차량 시장 전망 (단위 10억 달러)

자율주행차량의 자율주행 기능에서 가장 중요한 것은 시각 정보의 처리이다. 차량 외곽에 장착된 카메라에서 얻은 이미지로부터 주변 사물과 지형 등에 대한 정보를 파악하고 이를 차량 제어 신호와 연동하는 것은 자율주행 기능 구현에서 핵심적인 부분을 담당한다.

자율주행차량의 카메라에서 얻을 수 있는 정보에는 여러 가지가 있지만, 이 중 가장 비중 있게 다루어지는 것은 물체 정보와 차선 정보이다. 이는 사람이 운전할 때 교통상황(물체 정보)을 파악하고, 차선을 준수하여 운전하는 것과 일맥상통한다. [그림 2]에서 차량에 대한 물체 정보는 박스로 표시되고 인식된 차선은 선으로 강조 표시된 것을 확인할 수 있다.



[그림 2] 테슬라사 자율주행 기능의 전방 이미지 분석 화면

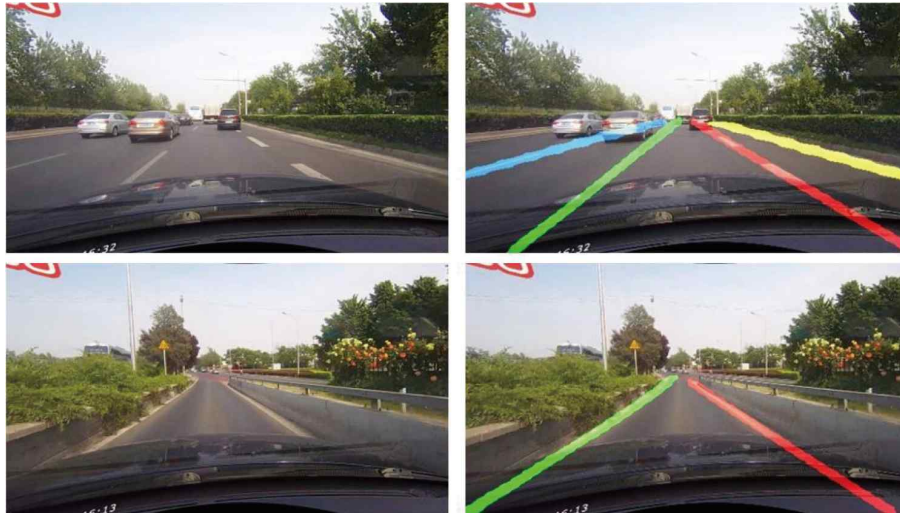
이 두 가지 정보처리를 적극적으로 이용한 완전한 자율주행 기능은 아직 고가의 고급 차량에만 탑재되고 있다. 반면에 차선 인식을 주요하게 이용한 차선 유지 보조 시스템(LKAS)이라고 불리는 저수준 자율주행 기능은 최근 자율주행차량 시장의 성장과 함께 기술개발이 치열해지면서 SUV, 중형 세단까지 탑재가 확대되었다.¹⁾

이런 배경에서 일상에 이미 녹아든 자율주행 기술이 어떤 방식으로 빅데이터를 활용했는지 알아보기 위해 “자율주행차량의 차선 인식”을 분석 사례로 선정하였다.

2. 차선 인식 문제란?

차선 인식 문제는 여러 상황과 관점에서 서로 다른 형태와 방법으로 기술될 수 있다. 이 문제를 해결하기 위해서, 본 보고서에서는 차선 인식 문제를 주행 차량의 전면 카메라에서 얻은 이미지를 통해 도로에 표시된 차선과 주행 도로의 경계면을 찾아내는 문제로 한정한다. [그림 3]의 차선 정보 사진에서는 도로의 경계면을 인식하는 모습을 확인할 수 있다.

1) 조재환, 「고급 세단에서 SUV까지...LKAS 대중화」, 『지디넷코리아』, 2017. 12. 04., <<https://zdnet.co.kr/view/?no=20171204161920>>



[그림 3] 전면 카메라 이미지(좌)와 차선 정보가 표시된 이미지(우)

차선 정보는 여러 가지 형태로 표현될 수 있다. 카메라 이미지를 좌표 공간에 올린다고 했을 때, 각 차선을 차선에서 일정 간격으로 추출된 점의 집합으로 표현할 수도 있고, 다항식으로 회귀²⁾하여 나타낼 수도 있다. 보통 점 집합으로 차선을 표현하는 방식을 더 많이 사용한다.



[그림 4] 점과 다항식으로 표시된 차선 정보

차선 인식 문제는 이처럼 주어진 이미지를 수치 또는 수학적으로 기술될 수 있는 데이터로 변환하는 문제를 말한다. [그림 4]에서 [그림 3]의 인식된 차선 정보를 각각 점 집합과 다항 함수로 표시한 예시를 확인할 수 있다.

3. 초창기의 차선 인식 접근 방법

차선 인식 기술의 역사는 길지 않다. 1992년 미쓰비시가 자사의 고급 세단에서 사용할 수 있도록 도로의 차선 표시를 추적할 수 있는 기본적인 카메라 작동 추적 시스템³⁾을

2) 다항 회귀(Polynomial Regression)란 점 집합을 가장 가까운 형태의 다항 함수로 변환하는 방법을 말한다.

만든 것이 차선 인식을 활용한 최초의 응용 사례이다. 이 당시부터 2010년 초반까지 차선 인식은 기초적인 컴퓨터 비전⁴⁾ 기술을 이용하여 이루어졌는데 이를 간략히 소개하고자 한다.

3.1. 특징 (Feature)

특징이란 어떤 집합이나 그룹이 가지고 있는 특이한 패턴이나 규칙을 말한다. 차선 인식 문제에서는 차선의 모양, 색깔, 형태 등이 차선 집합의 특징으로 정의될 수 있고 이렇게 정의한 특징은 차선을 검출하는 데 사용될 수 있다.

3.2. 손수 만들어진 특징(Handcrafted feature)

후술할 인공신경망 기반 차선 인식 방법은 사람의 시각으로 정의한 특징을 사용하지 않는다. 인공신경망은 내부의 특징 매개변수들을 역전파 과정을 통해서 갱신하면서 입력으로 들어온 개체들의 특징들을 학습한다.

반면에 초창기 차선 인식 접근 방법에서는 사람이 직접 차선의 특징을 “흰색의 10픽셀의 두께를 가지는 세로선(점) 집합”등으로 정의하고 이를 검출하기 위한 수학적 변환이나 검출 알고리즘을 직접 설계했다.

3.3. 소벨(Sobel) 필터와 허프(Hough) 변환을 이용한 차선 인식



[그림 5] OpenCV⁵⁾와 허프 변환을 이용한 차선 인식 예

이 차선 인식 방법은 3.2.절에서 다룬 바와 같이 사람이 직접 특징을 정의하고 검출하기 위한 수학적 변환 과정이 절차적으로 사용된다. 소벨 필터⁶⁾로 전처리 된 이미지에 서 직선을 검출하는 허프 변환을 통해 최종적으로 차선 정보를 얻는다. 과정별로 간략히 이를 설명하면 다음과 같다.

Input Frame: 차량 전면부에서 얻은 차선을 인식할 이미지

Grayscale: 각 픽셀당 RGB⁷⁾의 정보가 들어있는 이미지를 정보량을 줄여 연산량을 줄이고 변환 과정을 쉽게 처리하기 위해서 픽셀당 회색조 강도 값 하나를 갖는 이미지로 변환한다.

Gaussian Blur: 이미지의 잡신호를 없애고, 소벨 필터가 더 잘 모서리를 구분할 수 있도록 전처리한다.

3) 이 시스템은 실제로 미쓰비시의 고급 세단인 미쓰비시 데보네어에 장착하여 사용되었다.

4) 기계의 시각에 해당하는 부분을 연구하는 컴퓨터 과학의 연구 분야

5) OpenCV(Open Source Computer Vision)은 실시간 컴퓨터 비전을 목적으로 한 프로그래밍 라이브러리이다.

6) 이미지의 모서리를 검출하는 필터

7) 모든 색을 구성할 수 있는 삼원색 빨강(Red), 초록(Green), 파랑(Blue)

Sobel(소벨): 이미지의 모서리를 검출하는 필터로, 이 필터를 적용하면 이미지는 외곽선 부분만 남게 된다.

Segment: 사전 정의된 영역을 제외한 영역을 제거한다.

Hough(허프): 처리된 이미지에서 차선 직선을 검출하는 변환을 시행한다.

Output Frame: 차선 직선이 표시된 이미지

3.4. 초창기 차선 인식 접근 방법의 한계와 의의



a) Soil hid

b) Tree shadow

c) Street Light

[그림 6] 차선 인식 분야에서 인식하기 어려운 이미지들의 예

위와 같은 차선 인식 방법은 사람이 정의한 특징을 검출하는 원리를 기반으로 이루어지다 보니, 미리 사람에 의해 정의된 특징 검출식에서는 찾아낼 수 없는 형태의 지형인 경우, 차선 페인트가 벗겨지거나 가려져 있는 경우⁸⁾, 환경의 광량에 따라서 차선의 특징이 명확하게 보이지 않는 경우⁹⁾ 등의 다양한 환경의 예외적 사례에 대한 검출률이 현저하게 떨어졌다.

인공신경망을 이용한 방법이 주류가 된 지금에는 많이 사용하지 않게 되었지만, 이 접근 방법들은 이후 인공신경망을 활용한 방법에서 이미지를 전/후 처리하거나, 신경망 학습 간 차선 정보의 힌트를 제공하기 위해 일부¹⁰⁾가 차용되어 사용되고 있다.

4. 현재의 차선 인식 방법

4.1. 데이터 세트

손수 만들어진 특징에 대한 검출기를 만들어서 발견법적 방법으로 차선을 검출하는 초창기 차선 인식 방법과는 달리, 지금의 차선 인식 방법은 인공신경망을 이용한 방법을 사용한다. 이전의 방법은 손수 만들어진 특징 검출기가 찾아내지 못하는 예외를 수정하거나, 성능과 검출률이 더 나은 발견법적 알고리즘을 적용하는 방향으로 개발되었기 때문에 차선 인식에 사용되는 데이터 세트는 주로 알고리즘의 검증을 위해서만 소규모로 사용되었다.

하지만 인공신경망을 이용한 차선 인식 방법은 특징 자체를 학습된 데이터 세트로부터

8) [그림 6]의 a) Soil hid는 차선 페인트가 흙에 가려진 예시이다.

9) [그림 6]의 b) Tree shadow, c) Street Light는 나무나 거리 조명에 의해서 광량이 불규칙한 경우의 예시이다.

10) 3.1.2.에서 활용한 소벨 필터, 허프 변환 등을 통해 얻은 특징 정보(모서리 정보)를 신경망에 제공하여, 신경망이 차선의 특징을 효율적으로 검출할 수 있도록 하는 용도 등으로 활용된다.

습득하기 때문에 더 많은 차선에 대한 특징을 인공지능망에 학습시켜 검출률을 높이기 위해서 대규모 데이터 세트의 중요성이 증가했다. 이러한 상황 속에서 2010년 중후반부터 여러 차선 인식을 위한 데이터 세트의 제작이 이루어졌다. 이 장에서는 현재 차선 인식 분야에서 주요하게 사용되고 있는 데이터 세트의 종류와 구조에 관해서 설명한다.

	CULane	Caltech	NEXET	DIML	KITTI	TuSimple	UAH	BDD100K
Sequences	More the 55 hours of videos	4 clips	NA	470	22	7000	500 MINUTE VIDEO	100,000
Images	133,235	1225	50,000	-	14999	140000	NA	120,000,000
Multiple cities	NO	NO	Yes 77 cities	YES	NO	YES	YES	YES
Multiple Weathers	NO	YES	YES	YES	NO	NO	YES	YES
Multiple times of day	YES	YES	YES	YES	NO	NO	YES	YES
Multiple Scene Types	YES	YES	YES	YES	YES	YES	YES	YES
Multiple cameras	YES	NO	YES	YES	YES	NO	NO	YES
Multiple street type	NO	NO	YES	YES	YES	NO	YES	YES
Streets labeled	NO	YES	YES	YES	NO	YES	YES	YES

[그림 7] 차선 인식을 위한 데이터 세트 비교

모든 차선 인식 데이터 세트는 기본적으로 [그림 4]와 같은 원본 이미지에 대해 수치로 기술된 차선 정보가 1:1로 대응된 구조로 이루어진다. 원본 이미지의 다양성이나 차선 정보의 표현 형태는 데이터 세트의 제작 목적에 따라 달라질 수 있다.

데이터 세트에 따라 차선에 대한 수치 정보 뿐만 아니라 원본 이미지에 대한 다양한 표지를 제공하는 경우도 있다. [그림 7]에서 각 데이터 세트들의 이미지 특성¹¹⁾과 정보¹²⁾를 간략하게 표로 확인할 수 있다.

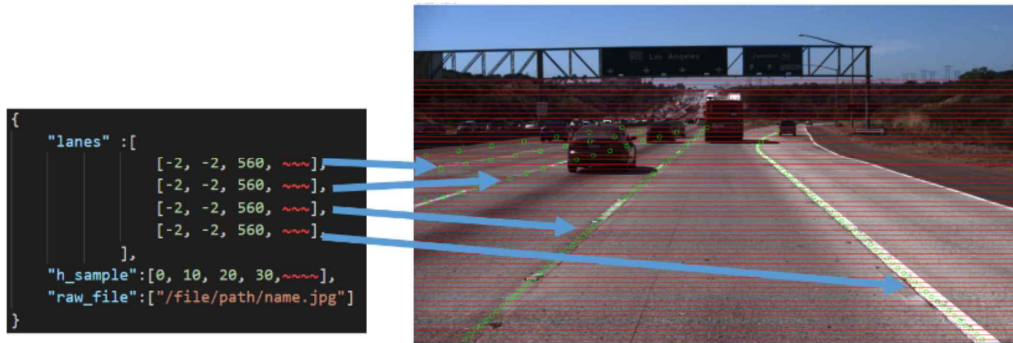
4.1.1. TuSimple



[그림 8] TuSimple 데이터 세트에 포함된 이미지

-
- 11) 이미지의 개수, 촬영된 도시의 단일 도시 여부, 날씨 및 시간대의 다양성 그리고 다수의 카메라를 사용했는지 등이 있다.
- 12) 이미지가 찍힌 거리에 대한 표지 정보의 제공 여부 등, 특히 BDD100K 데이터 세트에서는 GPS, 관성 센서 측정값, 이미지에 나타난 물체에 대한 영역과 종류 표지 등이 추가 정보로 제공된다.

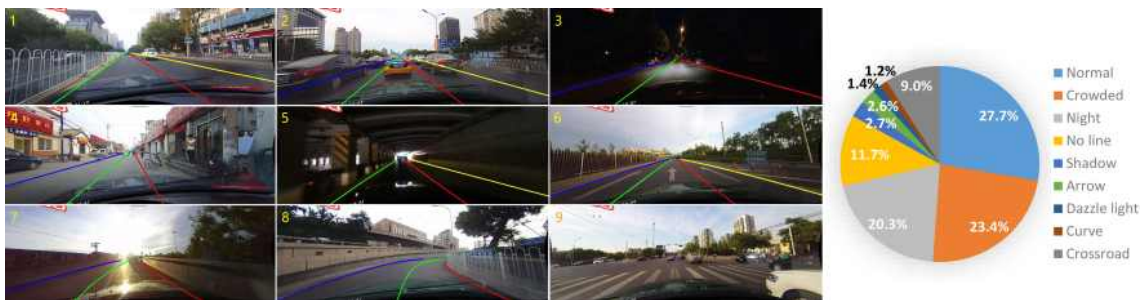
TuSimple 데이터 세트는 차선 인식 학습과 평가에 빈번하게 사용되는 데이터 세트 중 하나로 1280x720 해상도를 가진 6,408개의 미국 고속도로 이미지로 구성되어있다. 일반적인 도로 환경은 물론이고, 차선이 희미하거나 차선 없이 요철로 이루어진 구간의 데이터도 포함된다. 하지만 비가 오거나 안개가 끼는 등 다양한 기상 상황에 대한 데이터는 포함되어 있지 않고, 대체로 주간에 촬영한 사진들로만 구성되어있다.



[그림 9] TuSimple 데이터 세트의 차선 정보 표현 방식¹³⁾

사진에 대한 차선 정보는 JSON¹⁴⁾ 형태로 주어지고, 세로축 샘플링 지점인 h_sample 수치 배열의 각 원소에 위치에 대응하는 차선이 존재하면, lanes 내부의 각 차선의 수치 리스트에 가로축 좌표가 추가된다. 세로축 샘플링 지점에 대응하는 차선이 없는 경우 -2가 기록된다.

4.1.2. CULane



[그림 10] CULane 데이터 세트에 포함된 이미지

CULane은 차선 감지 분야의 학술 연구를 위한 도전적인 대규모 데이터 세트이다. 베이징에서 서로 다른 운전자가 운전하는 6대의 차량에 장착된 카메라로 수집된 1640x590 해상도 이미지로 구성되어있다. 차선 인식을 위한 단일 데이터 세트¹⁵⁾ 중 규모가 큰 편

13) 좌측 차선 정보에서 “~~~~~” 부분은 생략 표시로, 원래 데이터 세트에서는 수치가 들어가는 부분이다.

14) JSON(JavaScript Object Notation)은 Javascript 객체 문법으로 구조화된 데이터를 표현하기 위한 문자 기반의 표준 포맷이다

15) 인공지능망 학습 분야에서는 학습 데이터 세트로부터 검출하고자 하는 특징을 익히게 되므로 데이터 세트가 같은 분포에 있는 것이 매우 중요하다. 서로 다른 데이터 세트를 뒤섞어 쓰려면 별도

으로, 차선 인식 관련 논문에서 매우 빈번하게 사용된다. 해당 데이터 세트의 특징으로는 해상도가 통상적으로 사용하는 16:9가 아니고, 야간에 수집한 어두운 환경에서의 이미지가 포함되어 있다.

05130.lines.txt														
1	-30.082	480	8.74749	470	47.485	460	86.2224	450	124.96	440	163.697			
	430	203.403	420	242.141	410	280.878	400	319.616	390	358.353	380			
	397.238	370	435.975	360	474.713	350	513.45	340	553.156	330	591.893			
	320	630.631	310	669.368	300	708.106	290	746.843	280					
2	392.352	590	404.496	580	416.854	570	429.211	560	440.797	550	453.154			
	540	465.512	530	477.097	520	489.455	510	501.813	500	514.171	490			
	525.756	480	538.114	470	550.471	460	562.829	450	574.414	440	586.953			
	430	599.311	420	610.896	410	623.254	400	635.612	390	647.97	380			
	659.555	370	671.913	360	684.27	350	696.628	340	708.213	330	720.571			
	320	732.929	310	745.287	300	756.872	290	769.23	280					
3	1152.18	590	1141.73	580	1130.82	570	1120.64	560	1109.73	550	1099.55			
	540	1088.64	530	1077.73	520	1067.55	510	1056.65	500	1046.46	490			
	1035.56	480	1025.38	470	1014.47	460	1003.56	450	993.379	440	982.413			
	430	972.232	420	961.324	410	951.144	400	940.236	390	930.055	380			
	919.147	370	908.239	360	898.058	350	887.15	340	876.969	330	866.061			
	320	855.88	310	844.972	300	834.064	290	823.883	280					

[그림 11] CULane 데이터 세트의 차선 정보 표현 방식

차선 정보는 3차 다항식 보간법¹⁶⁾으로 얻은 함수로 제공되며 CULane 공식 배포처에서 함수 정보가 담긴 텍스트 파일을 픽셀 상의 점으로 변환해주는 코드와 방법을 제공한다.

4.1.3. BDD100K



[그림 12] CULane 데이터 세트의 포함된 이미지

BDD100K는 버클리 인공지능 연구 실험실(BAIR)에서 자율주행과 관련된 다양한 분야(물체 탐지, 차선 인식 등)의 인공지능 학습을 위해 만든 데이터 세트이다. 다양한 지리적, 환경적, 날씨 환경에서 촬영한 1억 2000만 개의 1280x720 해상도 이미지 데이터에 차선 인식 정보뿐만 아니라 도로 형태, 날씨, GPS, 물체 정보가 모두 포함되어 있다.

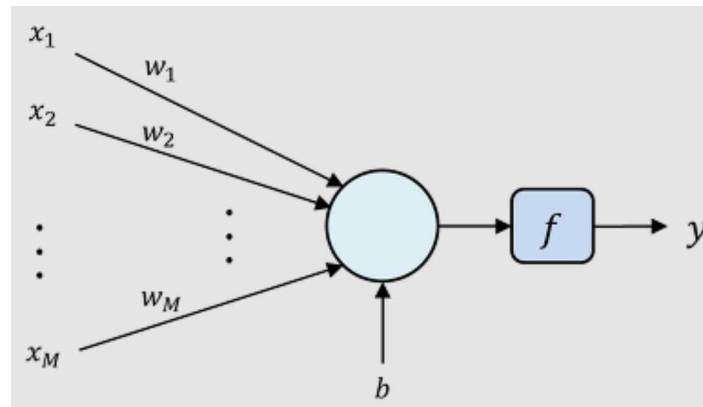
의 캘리브레이션 작업을 수행하거나 데이터 세트의 통계적인 분포가 일치하지 않는 위험을 감수하고 학습시켜야 하는데, 이 경우 인공지능망의 성능을 개선하는 과정에서 어떤 부분이 문제인지 파악하는 것이 어려워질 수 있다.

16) 두 개의 점을 연결하는 부분을 3차 다항식으로 표현하여 점 집합에서 매끄러운 함수를 구하는 방법

4.2. LaneNet (with H-Net)

LaneNet은 인공신경망을 사용한 차선 인식 분야에서 유의미한 수준의 정확도와 성능을 보여준 차선 인식 신경망으로 평가받는다. 이 장에서는 인공신경망(LaneNet) 기반 차선 인식을 설명하기 위해 사용되는 기본적인 원리와 개념에 대해서 알아본 후 “Neven, Davy, et al. "Towards end-to-end lane detection: an instance segmentation approach." 2018 IEEE intelligent vehicles symposium (IV). IEEE, 2018.” 논문에서 구현한 차선 인식 신경망의 원리에 대해서 알아보겠다.

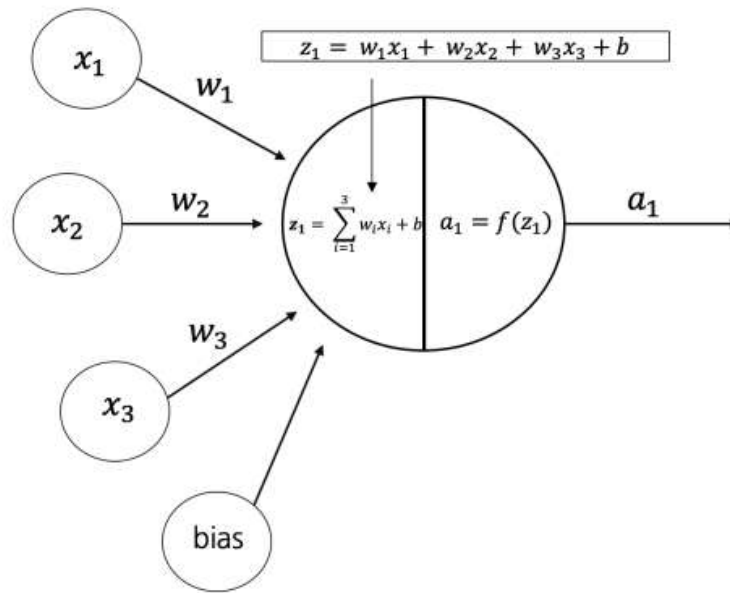
4.3. 인공신경망 (Artificial Neural network)



[그림 13] 단일 노드로 이루어진 신경망의 예

인공신경망은 생물학의 신경망¹⁷⁾에서 영감을 얻은 기계학습 알고리즘이다. 인공신경망은 기본적으로 입력 x 에 대한 출력 y 를 내는 구조로 설명된다. [그림 13]에서 입력에 대한 화살표가 도달하는 중간 지점의 원은 퍼셉트론 혹은 노드라고 불린다.

17) 뇌와 신경 세포 뉴런의 구조에서 영감을 받았다.



[그림 14] 인공신경망 노드의 계산 과정

이 노드는 신경망이 내부적으로 가지게 되는 가중치 w 와 입력 x 를 곱한 값에 편향 b 를 더한 뒤, 활성화 함수¹⁸⁾라는 것에 넣어 다음 노드(또는 출력)로 전달한다. 활성화 함수는 신경망의 목적에 따라 어떤 결과를 최종적으로 나타낼지에 따라 여러 가지 종류의 함수를 적용할 수 있다.

이렇게 인공신경망은 입력 x 를 주고 작동시켰을 때 단순히 지정된 연산에 따른 출력 값, y ¹⁹⁾를 반환한다. 이 인공신경망 구조의 놀라운 점은 가지고 있는 데이터가 가지고 있는 상관관계를 학습시켜 우리가 가지고 있지 않은 데이터에 대해서 예측하는데 사용될 수 있다는 점이다.

어떤 지역에 있는 주택의 가격 그리고 그 주택 반경 10km 이내의 대형 상가 수, 복지 시설 수, 의료 시설 수에 대한 정보를 우리가 가지고 있다고 가정하자. 대형 상가, 복지 시설, 의료 시설의 수는 분명히 주택 가격과 모종의 상관관계가 있을 것이다. 신경망의 입력 x_1, x_2, x_3 을 각각, 대형 상가의 수, 복지 시설의 수, 의료 시설의 수로 설정하고, 최종 출력 y 를 주택 가격으로 생각하자.

우리는 가중치와 편향 값에 의해서 입력에 대한 출력 y 가 형성되는 것을 알고 있다. 처음에 가중치와 편향이 아무렇게나 설정된²⁰⁾ 신경망은 주택 가격과는 먼 이상한 값을 출력할 것이다. 하지만 이런 가중치와 편향을 조절해서 주택 가격에 가깝게 출력이 나오게 설정한다면 어떨까? 우리가 가지고 있지 않은 x_1, x_2, x_3 조합에 대해서도 예상 주택 가격인 y 를 출력해낼 수 있을 것이다.

단순히 가중치와 편향을 조절한다고 알지 못하는 x_1, x_2, x_3 조합에 대한 y 출력이 실

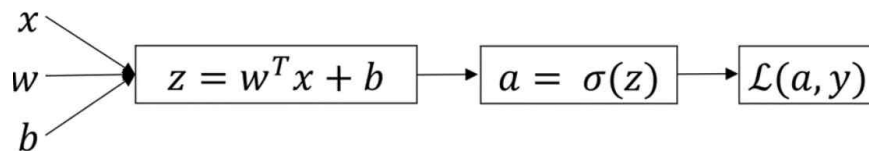
18) f [그림 13] 또는 $\sigma(z)$ [그림 14]로 표시된다.

19) 아직은 의미가 없는 출력 값이다.

20) 가중치와 편향은 여러 가지 값으로 초기화하는 기법이 있지만 여기서 가중치는 무작위 값으로, 편향은 모두 0으로 초기화되었다고 가정한다.

제로 올바른 값을 가리킬 것이라고는 보장할 수는 없지만, 세상의 많은 문제는 규칙성이 있고 신경망의 가중치와 편향을 조절하는 방법론은 우리가 가지고 있는 입력과 출력에 대한 정보를 최대한 흉내를 내는 방향으로 이루어지기 때문에, 여러 분야에서 인공지능망을 이용한 방법은 좋은 성능을 발휘하고 있다. 그렇다면 신경망이 실제로 가중치와 편향을 조절하는 방법은 어떤 원리일까?

위에 설명한 입력(x_1, x_2, x_3) 조합에서 y 출력을 얻는 과정을 순전파(Forward propagation)이라고 부른다. 신경망이 가중치와 편향을 조절하는 방법은 출력으로부터 입력으로 거슬러 올라가면서 가중치와 편향을 조절하는 형태를 띠기 때문에 역전파(Backward propagation)라고 부른다. 역전파의 간단한 원리는 다음과 같다. 우리가 (대형 상가: 4개, 복지 시설: 5개, 의료 시설: 7개)→(주택 가격: 12억)과 같은 정보를 가지고 있다고 가정하면 가중치와 편향이 올바르게 설정되지 않은 신경망은 잘못된 값을 예측할 것이다.



[그림 15] 단일 노드 신경망의 순전파 도식²¹⁾

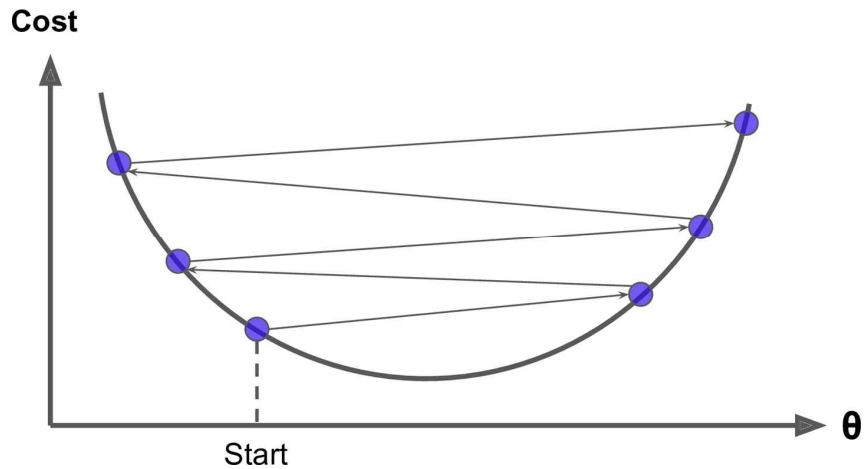
여기서 우리는 손실 함수²²⁾라는 것을 정의한다. 손실 함수는 출력 값이 우리가 가지고 있는 주택 가격(12억)과 얼마나 다른지에 대한 지표를 제공하는 함수이다. 가장 간단한 손실 함수로 “우리의 주택 가격(12억)-예측 가격”을, 즉 “실제값-예측값”을 사용할 수도 있지만²³⁾ 보통은 신경망 학습에 더 적합하게 고안된 함수인 교차 엔트로피 오차 손실 함수를 이용한다.

우리는 가지고 있는 모든 $X \rightarrow Y$ 대응 쌍에 대해서 손실 함수를 모두 더한 값이 0에 가까울수록 신경망이 입력에 대한 실제 값을 잘 예측한다고 생각할 수 있다. 따라서 역전파는 이 손실 함수를 모두 더한 값을 0에 수렴시키는 가중치와 편향의 방향을 찾아, 그 방향만큼 가중치와 편향을 갱신하는 형태로 이루어지게 된다. 여기서 손실 함수를 모두 더한 값의 평균을 비용 함수라고 하고, 가중치와 편향을 조절하여 비용 함수를 0으로 최소화시키는 역전파 과정을 경사 하강법이라고 부른다.

21) w^T 는 가중치 행렬의 전치 행렬이다.

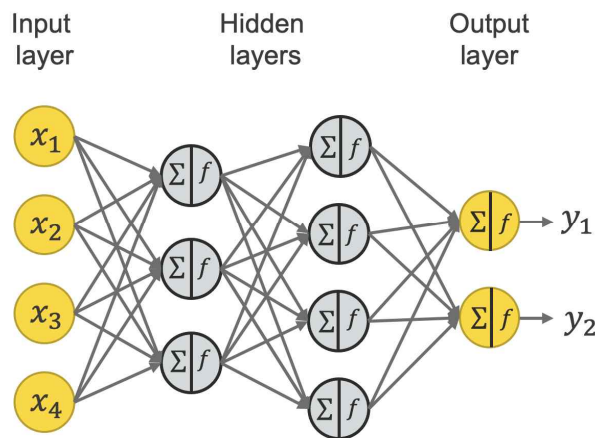
22) [그림 15]에서 \mathcal{L} 로 표기한다.

23) 이를 이용한 손실 함수를 절대 오차(Absolute error) 손실 함수라고 한다.



[그림 16] 경사 하강법의 기하적 표현

4.3.1. 심층 신경망 (Deep neural network)



[그림 17] 입력, 은닉, 출력층을 가지는 인공신경망

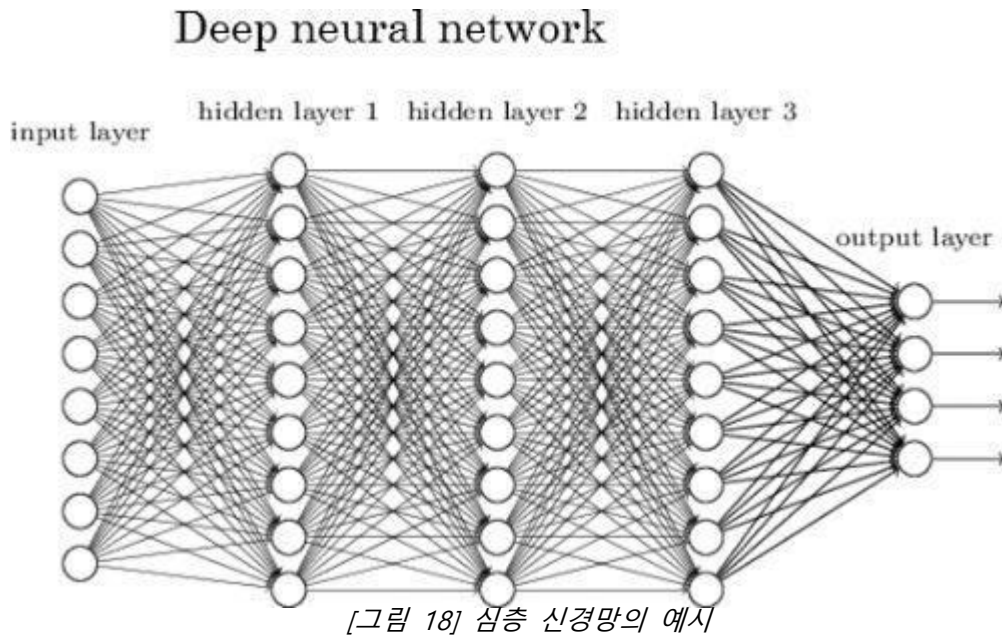
이전에 구성한 단일 노드만을 가지는 신경망은 가중치가 입력의 개수만큼, 편향은 노드의 개수만큼 가지게 된다. 따라서 이전의 주택 가격 예측 예시에서 사용하는 신경망 내부 변수는 입력 개수가 3개, 노드가 1개이므로 가중치 3개와 편향 1개로 총 4개의 변수를 가지게 된다.

그런데 이런 단순한 구조에서 아무리 많은 정보를 역전파 과정으로 학습시켜 신경망의 변수를 설정한다고, 유의미한 주택 가격을 예측할 수 있을까? 우리가 낮은 차수의 다항식(직선과 같은)이 고차 다항식에 비해서 단조로운 특징을 가지는 것을 수학에서 확인할 수 있듯이 이런 간단한 신경망은 복잡한 규칙을 학습하지 못한다.

따라서 이런 노드들 여러 개를 중첩적으로 신경망을 구성하는 데 사용함으로써, 인공 신경망은 복잡한 성질에 대해서도 학습할 수 있게 된다. [그림 17]에서처럼 각 노드는 이전에 배치된 노드의 출력값을 입력으로 받아 연산을 수행한 다음에 배치된 노드의 입력으로 자신의 출력을 넘겨주게 된다. 같은 입력을 받아들이는 노드들의 집합을 레이어라고 하고, 입력 부분을 제외한 레이어들을 은닉층이라고 부른다. 그리고 마지막으로 결괏

값을 출력하는 레이어를 출력층이라고 한다.

은닉층과 출력층 노드의 구성이나 개수, 활성화 함수 등은 인공신경망의 목적에 따라서 달라질 수 있으며 많은 수(2개 이상의)의 은닉층을 가지는 인공신경망을 심층 신경망(Deep neural network)이라고 한다.



4.3.2. 합성곱 신경망 (Convolution neural network, CNN)

4.3.2.1. 기존 인공신경망 구조의 문제점

기존의 인공신경망은 주택 가격 예측을 시도하는 문제와 같이 적은 수의 입력과 출력을 가지는 문제에서는 잘 적용될 수 있었다. 하지만 많은 개수의 입력을 가지는 문제에는 적용하기 힘들었다.

왜냐하면 신경망이 너무 복잡해지기 때문이다. 예를 들어 사진을 입력으로 받아, 그 사진이 고양이인지 아닌지 예측을 시도하는 것²⁴⁾과 같은 이미지 인식 문제에서는 입력이 사진의 화소 개수만큼 존재하게 된다.

회색조 이미지를 인식하는 문제의 경우 각 화소당 한 개의 회색조 강도 수치 하나를 가지게 되므로 이미지의 화소 개수와 같은 입력을 가지고, 컬러 이미지를 인식할 때는 각 화소당 빨강, 초록, 파랑의 색을 표현하는 3가지 수치를 가지게 되므로 화소 개수 * 3 개의 입력을 가지게 된다.

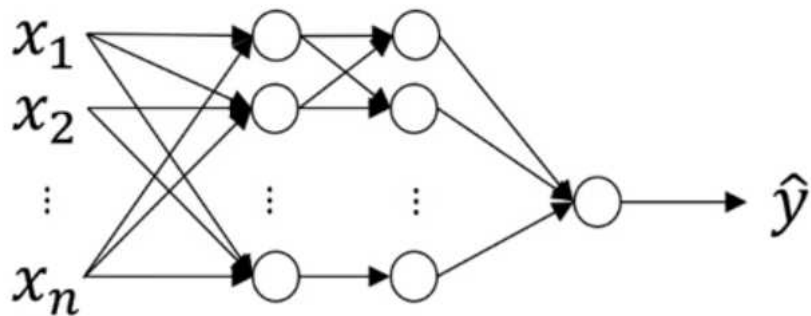
24) 출력은 0(고양이가 아님), 1(고양이임) 두 개의 숫자를 사용하여 나타낼 수 있다.



64x64

→ Cat? (0/1)

[그림 19] 64x64 해상도의 사진에서 고양이를 인식하는 경우



[그림 20] 사진을 입력으로 받아 고양이를 식별하는 신경망

다음과 같이 문제 상황을 가정해보자. 고양이 그려진 사진이 있고 이 사진을 신경망에 입력으로 주어서 출력 값으로 고양이이면 1, 아니면 0을 출력하게끔 학습시키려고 한다. 사진은 각 픽셀마다 RGB 정보를 가지고 있다. 이것을 입력으로 받는 신경망을 생각하면 고양이 사진이 64x64의 해상도일 때 64x64x3으로 12,288개의 입력을 받아서 1개의 출력이 나오게끔 신경망이 구성된다. 이 신경망에서 사용되는 가중치와 편향의 개수는 몇 개일까? 1000²⁵⁾개의 노드로 구성된 은닉층이 2개 있는 신경망이라고 했을 때 1,328만 9천 개의 가중치 변수²⁶⁾와 2001²⁷⁾개의 편향 값을 가지게 된다.



1000x1000

→ Cat? (0/1)

[그림 21] 1000x1000 해상도의 사진에서 고양이를 인식하는 경우

그럼 이번에는 고해상도의 고양이 사진인 1000x1000 해상도의 이미지를 입력으로 받는 상황을 고려해보자. 첫 번째 은닉층에서 사용되는 가중치 변수의 개수만 세도 30억²⁸⁾

25) 입력 개수에 비례하여 신경망의 유의미한 복잡도를 가질 수 있도록 임의로 가정한 레이어의 노드 개수

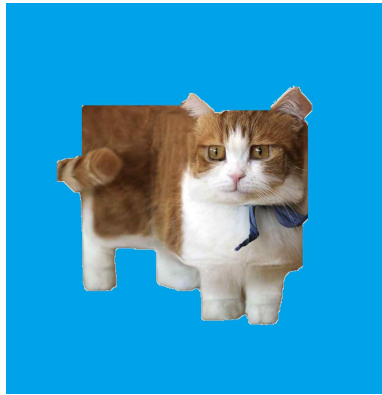
26) 12288*1000 + 1000*1000 + 1000

27) 1000*2+1

개가 된다. 이는 현대 컴퓨터의 저장 공간과 계산 성능으로도 처리하기 어려울 정도로 비대한 신경망이다. 이런 이미지에서의 신경망 사용 문제를 해결하기 위해서 합성곱 신경망이 만들어지게 되었다.

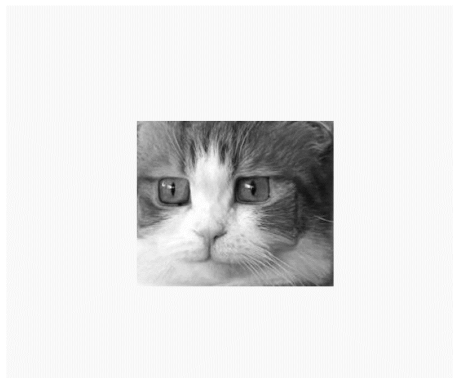
4.3.2.2. 합성곱 연산과 필터

이전 장에서 인공신경망을 이용하여 고양이를 검출하기 위해서는 현실적으로 감당할 수 없는 수준의 연산 비용이 필요하다고 다루었다. 그렇다면 이미지에서 고양이를 효율적으로 검출하려면 어떻게 해야 할까? 효율적인 고양이 검출법에 관해 설명하기 위해서 [3.2.절](#)에서 다루었던 내용을 조금 더 자세히 설명하겠다. 고양이는 사진에서 어떻게 표현될까? 고양이만의 색깔, 털 무늬, 세모난 귀 등의 다양한 특징이 있지만 조금 더 저수준의 특징을 생각해보자.



[그림 22] 빈 공간의 고양이

빈 공간에 고양이가 있는 그림을 상상하면, 사진은 고양이를 제외하고는 텅 빈 픽셀값을 가지다가, 고양이가 사진에 찍힌 위치에 다가가면 서서히 고양이의 모서리가 나타날 것이다. 모서리가 모여 고양이의 얼굴과 형체를 만들고 이를 공간과 구별할 것이다.

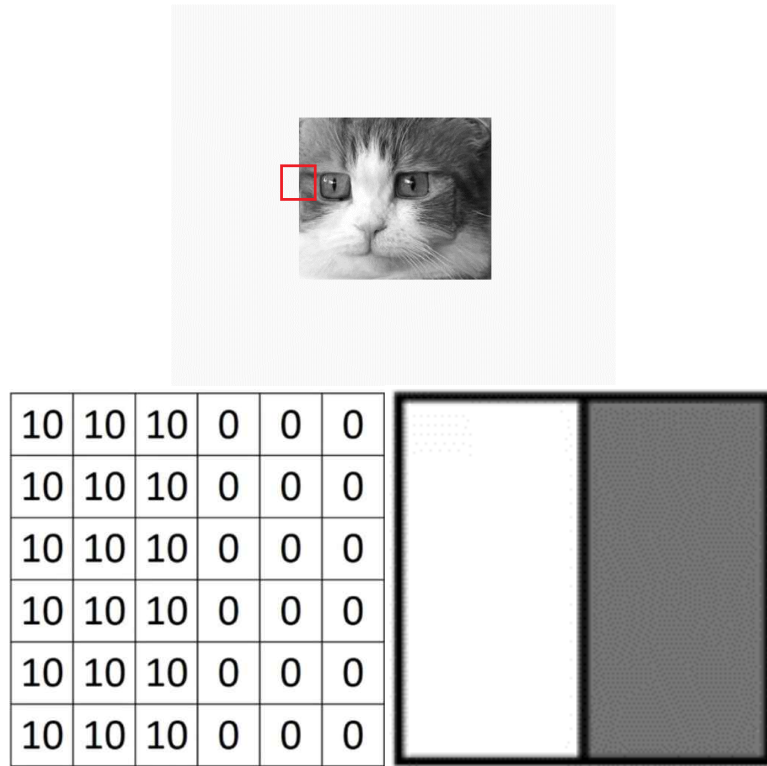


[그림 23] "빈 공간의 고양이"를 단순화한 사진

문제를 조금 더 단순화하기 위해서 [\[그림 22\]](#)의 고양이를 사각형으로 단순화하고 픽셀당 RGB 값을 가지는 것이 아니라 회색조 강도 값²⁸⁾ 하나만을 가진다고 가정하겠다. 이

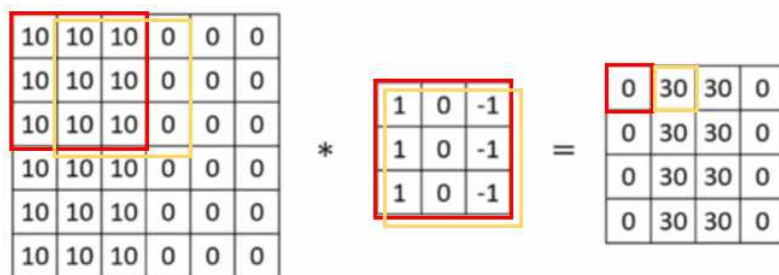
28) 1000*1000*3*1000

단순화된 고양이를 검출하기 위해서는 빈 공간에서 가로 모서리나 세로 모서리가 있는 곳을 찾아내면 될 것이다. 두 모서리 중에 세로 모서리만 고려하자. 고양이의 세로 모서리를 검출하려면 어떻게 해야 할까?



[그림 24] 고양이 얼굴 좌측 사각형의 픽셀 숫자 행렬과 간이 표현

위 고양이 사진 일부분의 회색조 강도 값을 숫자 행렬로 나타내면 [그림 24]와 같다.



[그림 25] 합성곱 연산 과정

여기서 [그림 25]의 3x3 숫자 행렬을 고양이 사진 일부분 숫자 행렬에 요소별로 곱한 뒤 더한 값³⁰⁾을 새로운 숫자 행렬에 나타내보자³¹⁾. 새로운 숫자 행렬의 중앙에 30의 값이 채워진 것을 확인할 수 있다. 이는 중앙에 모서리가 있다는 수치적 표현이자 특징 정

29) 밝을수록 높은 값을 가진다.

30) $10*1+10*0+10*-1+10*1+10*0+10*-1+10*1+10*0+10*-1=0$, 노란색 사각형 기준으로 $10*1+10*0+-1*0+10*1+10*0+-1*0+10*1+10*0+-1*0=30$

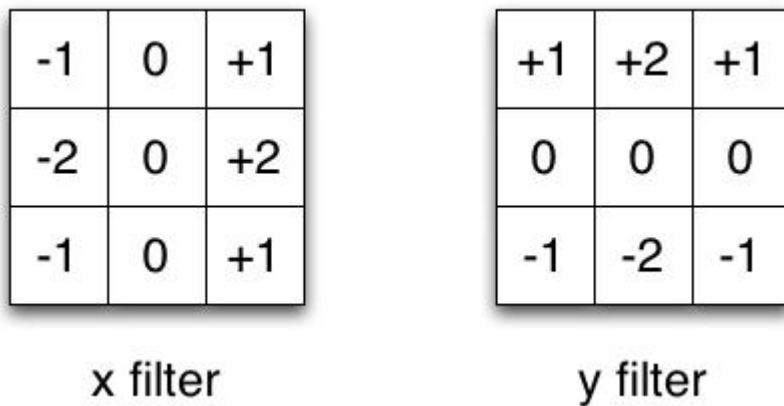
31) [그림 25]의 우측 행렬에 해당 값이 나타난다.

보가 된다. 여기서 이미지에 숫자 행렬을 곱한 뒤 합산하는 연산을 합성곱(Convolution) 연산이라고 한다.



[그림 26] 고양이의 좌/우 부분에 대한 합성곱 연산 결과

고양이의 좌/우 부분에 대해서 이 합성곱 연산을 시행하면, 모서리가 좌측에 있는지, 우측에 있는지를 판단할 수 있는 특징 정보를 가지는 숫자 행렬을 만들 수 있다. 여기서 다른 3x3 행렬을 필터라고 한다. 필터는 이미지에서 특징을 검출하는 숫자 행렬이다. 초창기 차선 인식 방법에서는 바로 이런 방식으로 이미지에서 특징을 검출해냈다.



[그림 27] 가로축과 세로축 모서리 검출에 사용하는 소벨 필터

3.3.절 차선 인식의 모서리 검출에 사용했던 소벨 필터가 바로 이 숫자 행렬을 의미한다. 이런 손수 만들어진 필터를 사용하여 세로, 가로 방향과 같은 간단한 모서리를 검출하는 것은 쉽지만, 대각선 모서리, 곡선 모서리 또는 이미지에 포함된 사물의 복잡한 특징 등을 검출할 수 있을까?

이미지에서 고양이를 검출하는 것만을 생각해도 수많은 필터의 가짓수가 있는 상황에서, 검출을 위한 모든 필터를 사람이 직접 만들기는 어려울 것이다. 그럼 특징을 검출하는 필터를 심층 신경망이 만들면 어떨까? 이것이 합성곱 신경망의 핵심 아이디어이다.

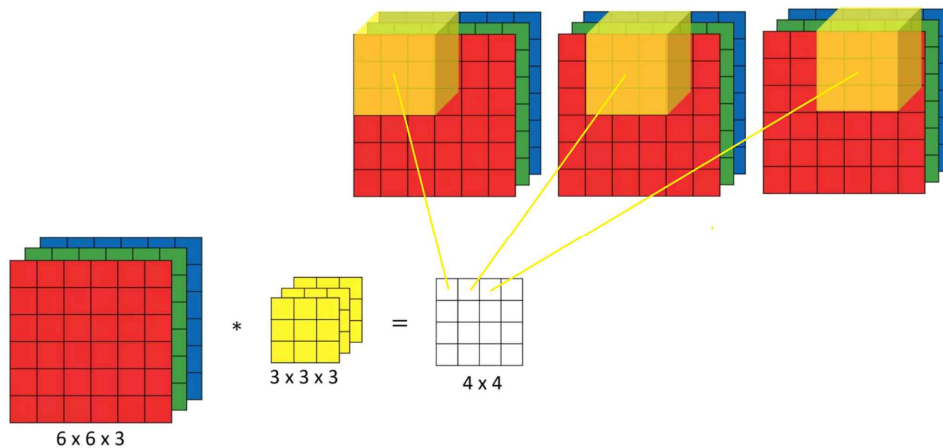
w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

[그림 28] 가중치 변수로 구성된 필터

신경망의 입력을 이미지로 받고, 위와 같은 3x3 필터를 각 숫자 행렬의 원소를 가중치로 설정한 뒤 이 가중치로 이루어진 필터가 고양이의 특징을 학습하게 만든다. 그리고 고양이 이미지로 신경망을 학습시키면 고양이를 검출하는 필터를 얻을 수 있게 된다. 이 필터는 각진 고양이가 아니라 일반적인 고양이도 검출할 수 있을 것이다.

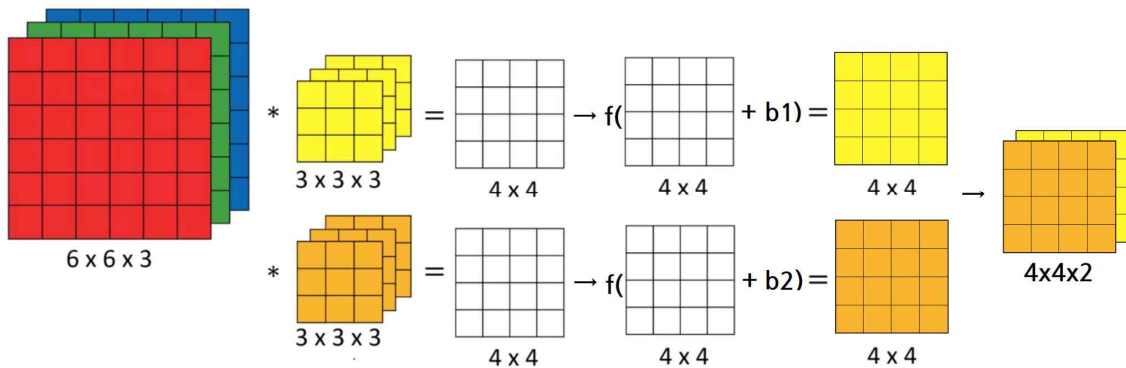
4.3.3. RGB 이미지에서의 필터와 합성곱 연산

[이전 절](#)에서 다룬 것과 같이 회색조 이미지가 아니라 일반적인 RGB 이미지에 대해서는 어떤 방식으로 적용되는지를 살펴보겠다.



[그림 29] 6x6 해상도를 가지는 RGB 이미지에 3x3x3 크기의 필터를 합성곱 연산하는 예시

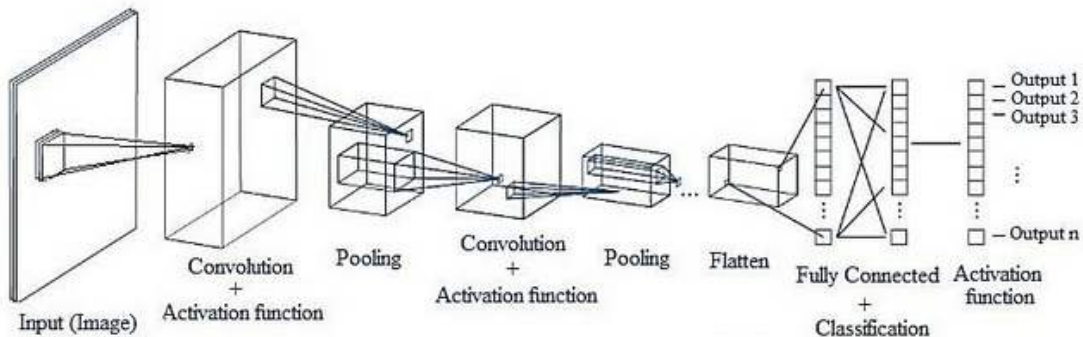
일반적인 RGB 이미지에 대해서 합성곱 연산을 수행하면 [그림 29](#)와 같이 각 RGB 숫자 행렬 3개에 대해서 3x3x3 필터를 합성곱하게 된다. 이는 마치 3차원 부피를 가지는 이미지에 작은 부피를 가지는 필터를 가져다 대면서 합성곱 결과를 계산하는 것과 같다. 여기서 겹쳐진 숫자 행렬의 개수를 채널 수라고 한다. 합성곱 연산을 위해서는 이미지의 채널과 필터의 채널 수가 같아야 한다.



[그림 30] 두 가지의 필터를 합성곱 연산한 뒤 4x4x2의 필터 출력을 계산하는 과정

합성곱 신경망의 한 레이어³²⁾는 어떠한 특징³³⁾을 검출하는 필터의 개수만큼 숫자 행렬을 만든다. 그리고 이 출력에 편향 값을 더해 활성화 함수에 넣은 값을 다음 레이어의 입력으로 넘겨주고, 다음 레이어에서는 이 필터를 처리한 결과인 특징 맵³⁴⁾을 입력으로 받아 다시 해당 레이어가 가지는 필터를 적용하여 연쇄적으로 다음 레이어로 넘겨주게 된다. 그리고 이 숫자 행렬은 대체로 주어진 입력 이미지의 크기보다 작게 나타나게 된다.³⁵⁾

4.3.3.1. 합성곱 신경망의 구조



[그림 31] 합성곱 신경망 기반 분류 모델의 구조³⁶⁾

합성곱 신경망은 이전 장에서 다루었던 것처럼 필터 처리 결과를 계속 다음 레이어에 입력으로 전달하면서 이미지의 특징을 필터를 통해 연쇄적으로 추출하고, 입력 이미지의 크기를 계속 줄여 나간다.

32) 심층 신경망의 레이어와 같은 개념이다.

33) [그림 30]와 같은 경우 가령 수직 모서리 검출 필터, 수평 모서리 검출 필터 등을 노란색, 주황색 필터로 사용할 수 있다.

34) 입력 이미지를 필터하여 특징을 추출한 것으로 이 필터 결과는 특징 맵(Feature map)이라고 불린다.

35) 수학적으로 이미지 크기-필터 크기+1로 계산된다. [그림 30]과 같은 경우 6-3+1=4로 계산할 수 있다.

36) [그림 31]의 Pooling은 이전 레이어 출력 결과에서 특징을 더욱 강조하거나 숫자 행렬 크기를 줄이는 효과를 가지는 연산이다.






[그림 32] 얼굴을 인식하도록 학습된 합성곱 신경망의 레이어별 특징 맵

학습 간 반복되는 역전파 과정을 거치면서 입력과 가까운 쪽에 있는 필터는 모서리와 같은 저수준 구조를 검출할 수 있게 학습되고 출력과 가까운 쪽에 있는 필터는 이전 필터에서 필터 처리된 결과(입력)에 대해서 고수준 구조를 검출할 수 있게 학습된다.

합성곱 신경망은 대체로 각 레이어별로 주어진 상황에 최적화된 필터 크기를 설정하여 레이어를 지날수록 받는 입력의 크기³⁷⁾는 줄이고, 채널의 수는 증가시키는 방향으로 설계된다.³⁸⁾ 중국에는 입력의 크기가 엄청나게 줄어들게³⁹⁾ 되고 채널의 개수만 많이 늘어나게 되는데, 이렇게 하는 이유는 방대한 크기의 최초 입력에서 작은 크기의 고차원 특징 맵으로 압축하기 위해서이다. 이렇게 압축된 특징 맵은 각 숫자 행렬 요소를 1차원 배열로 쭉 펴서 나열한 뒤 우리가 위에서 다루었던 심층 신경망 구조에 적절한 개수⁴⁰⁾의 입력 크기로 전달된다.

이러한 구조에서 학습이 진행되면, 합성곱 신경망이 저밀도의 정보 밀도를 가지는 원래의 이미지에서 고밀도의 정보 밀도를 가지는 이미지로 입력 수를 줄여주기 때문에, [그림 19]와 같이 1000x1000의 해상도를 가지는 이미지에 대해서도 고양이를 검출해낼 수 있게 된다.

	Encoding
	0
	1
	2

[그림 33] 고양이를 0, 강아지를 1, 얼룩말을 2로 표현하는 부호화 방식

또한 [그림 33]처럼 고양이뿐만 아니라 여러 개체에 대한 출력값을 지정하고 학습을 진행해서 사진 속의 대상이 어떤 분류에 속하는지 감지해낼 수도 있다.⁴¹⁾ 이런 작업을 해내기 위해서 설계된 합성곱 신경망을 합성곱 신경망 분류기(Convolutional neural network classifier)라고 한다.

37) 숫자 행렬의 크기

38) 이 과정을 수행하기 위해서 각 레이어 별로 필터의 개수, 크기, Pooling 등의 연산을 수행한다.

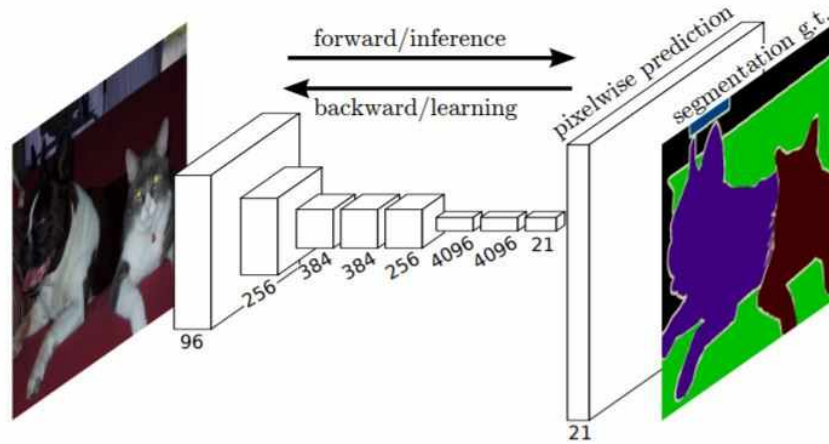
39) 신경망 구성에 따라 다르지만 7x7, 3x3 크기 수준으로 줄어든다.

40) 이미지 전체를 입력으로 사용하는 것과 달리 감당할 수 있는 수준으로 입력을 줄여서

41) 신경망 구성에 따라 여러 개체를 한 번에 찾는 것도 가능하다. 이런 분류기를 다중 라벨 분류기(Multi-label classifier)라고 한다.

4.3.4. 이미지 분할 (Image segmentation)

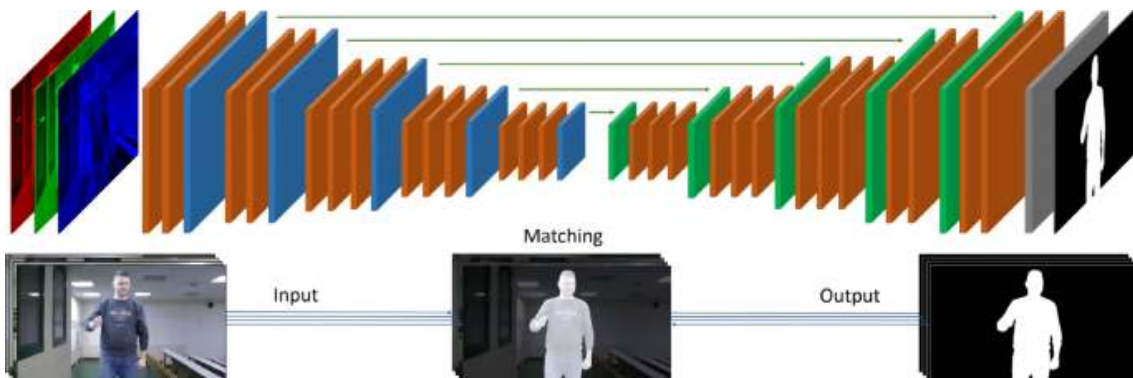
이전 장에서는 합성곱 신경망을 통해 이미지를 입력으로 받아 어떤 카테고리(고양이, 강아지, 얼룩말)에 속하는지를 분류하는 방법에 대해서 알아보았다. 이 장에서는 단순히 사진을 분류하는 것이 아니라, 사진에 어떤 영역에 어떤 물체가 존재하는지 분류하는 방법에 대해서 알아보겠다.



[그림 34] 합성곱 신경망 기반 이미지 분할 모델의 구조

[그림 31]에서 우리는 합성곱 신경망의 구조가 점점 입력의 크기를 줄여나가는 방식으로 이루어진다는 것을 다루었다. 우리는 최종적으로 숫자로 객체를 분류하기 위해서 특징을 추리고, 이 추린 특징에서 객체를 분류하는 방법을 사용했다. 그런데 이 과정을 반대로 해서 고차원 특징값을 다시 이전 단계의 위치에 복원하면서 입력의 크기를 키우면 어떻게 될까?

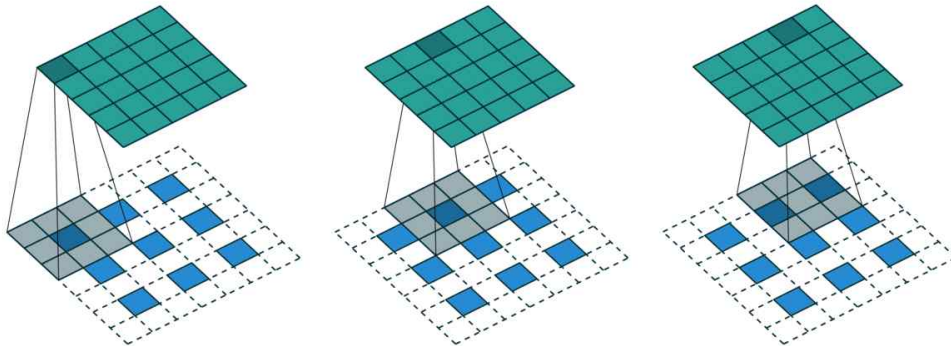
이미지의 고차원 특징값들이 복원된 출력 영역에 나타나면서 이미지의 같은 고차원 특징을 가진 영역이 고유한 값으로 표시될 것이다. 이렇게 이미지를 각 객체의 영역으로 나누는 것을 이미지 분할이라고 한다.



[그림 35] 인코더와 디코더 구조

이런 이미지 분할 모델의 구조는 크게 인코더와 디코더로 구성되어있다. 인코더는 출력 숫자 행렬의 크기를 줄이면서 고차원의 특징으로 입력을 압축하는 부분을 말하고, 디

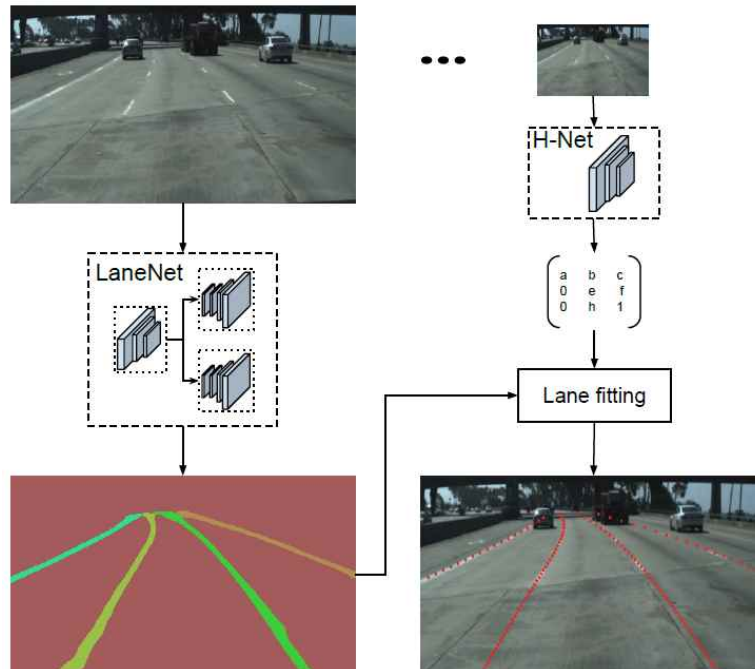
코더는 출력 숫자 행렬의 크기를 키우면서 고차원 특징을 이미지 영역에 표시하는 부분을 일컫는다.



[그림 36] 전치된 합성곱(Transposed convolution) 연산의 예

디코더 부분의 경우 출력 크기를 확장하기 위해 전치된 합성곱 연산을 주로 사용한다. [그림 36]의 전치된 합성곱 연산은, 아래의 4x4 크기의 숫자 행렬에 대해서 (흰색 부분), 3x3 크기의 패딩⁴²⁾을 넣고, 3x3 크기의 필터(모든 숫자가 1인)를 이용하여 합성곱 연산을 수행하여 입력 숫자 행렬의 크기를 5x5로 확장한다.

4.3.5. 차선 인식 신경망의 구조



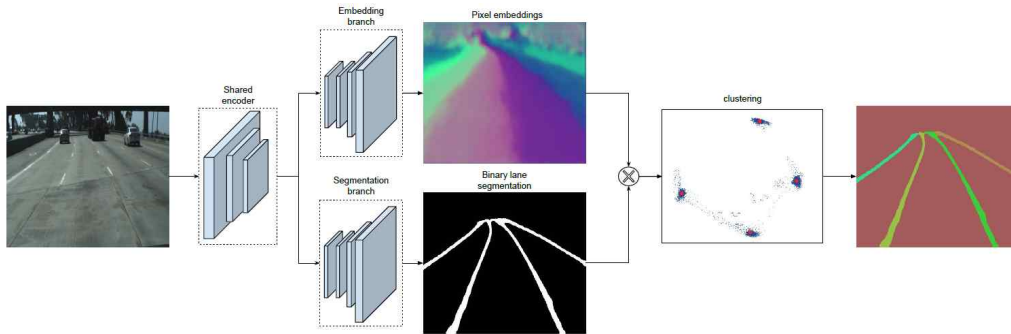
[그림 37] 차선 인식 신경망의 전체적인 구조

LaneNet 논문에서 다루는 차선 인식 신경망은 LaneNet과 H-Net이라고 불리는 두 개

42) 0의 값으로 구성된 숫자 행렬 영역

의 신경망으로 구성되어 있다. 이 장에서는 위에서 설명한 인공지능 기반 차선 인식에 사용되는 개념들을 바탕으로 LaneNet과 H-Net의 구조와 작동 원리에 대해서 알아볼 것이다.

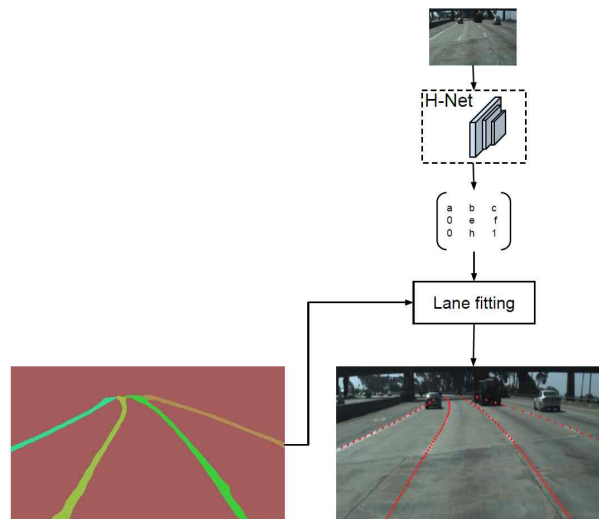
4.3.5.1. LaneNet



[그림 38] LaneNet의 구조

LaneNet은 합성곱 연산과 인코더/디코더 구조를 이용한 이미지 분할 신경망인 ENet의 구조를 기반으로 한다. LaneNet은 입력 이미지에 대해서 공유된 인코더로 차선의 특징 맵을 구성한 다음에 Embedding branch와 Segmentation branch의 두 분기 신경망으로 나누어진다. 이 두 개의 분기 신경망들은 각기 다른 방식의 디코더로 구성되어 서로 다른 결과를 생성하는데. Embedding branch는 각 차선 영역을 이미지 분할한 결과를 출력하고, Segmentation branch는 차선의 선 영역을 배경과 구분짓는 이미지 분할 결과를 출력한다. 이 두 네트워크의 “차선 영역을 이미지 분할”한 결과와, “차선의 선 영역을 배경과 구분 지어 이미지 분할”한 결과는 취합되어 후처리 기법에 따라서 각 차선이 고유하게 분리된 형태로⁴³⁾ 변환되어 출력된다.

4.3.5.2. H-Net



[그림 39] H-Net의 구조

43) Segmentation branch에서는 배경과 차선의 영역이 분리되었지만, 결과 이미지에서는 각 차선마다 개별 영역으로 분리되었다.

H-Net은 LaneNet 출력인 “각 차선별 픽셀 영역 모음”을 곡선 함수⁴⁴⁾로 변환하기 위한 수학적 변환 방법이 서로 다른 소실점⁴⁵⁾을 가지는 이미지에 대해서 오류를 가지는 문제점이 있어서, 각 입력 이미지에 대해서 LaneNet의 출력을 가장 정확하게 차선 정보⁴⁶⁾로 변환할 수 있는 변환 행렬을 구하기 위해서 고안된 신경망이다.

$$H = \begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & f & 1 \end{bmatrix}$$

[그림 40] H-Net의 출력으로 나오는 변환 행렬의 모습

이 신경망은 의도적으로 작은 규모로 구성되었으며, 각 레이어별 16~64개의 필터를 가지고 합성곱을 진행하는 방식으로 설계되었다.

4.3.5.3. 요약

여기서 다른 차선 인식 신경망은 입력 이미지에 대해서 H-Net의 변환 행렬 출력과 LaneNet의 차선별 픽셀 영역 모음 출력을 얻은 뒤, 변환 행렬을 통해 LaneNet의 출력을 차선 정보로 최종적으로 변환하여 차선 인식을 수행한다.

이 차선 인식 신경망은 50 FPS⁴⁷⁾의 성능을, TuSimple 데이터 세트의 성능 순위에서 4위⁴⁸⁾를 기록했다.

Performance analysis of existing lane detection approaches.

Methodology	Accuracy
Guassianfilter+Hough transform	89%
MSER+Hough	92%
HSV-ROI+Hough	95%
CNN Based	96%
Improved LD+Hough	96%
LaneNet	97%

[그림 41] 다양한 환경⁴⁹⁾을 포함한 데이터 세트에서의 차선 인식 방법 정확도 비교

또한 낮, 밤, 저조도⁵⁰⁾ 등의 다양한 환경을 포함한 데이터 세트에서는 허프(Hough) 변

44) 2차, 3차 다항식 등의

45) 소실점은 원근법에서 실제로는 평행선이 되어 있는 것을 평행이 아니게 그릴 때, 그 선이 사라지는 지점이다.

46) TuSimple, CULane에서 사용했던 것과 비슷한 형태의 차선 정보를 말한다.

47) 초당 50개의 이미지에 대해서 차선 인식을 수행할 수 있음을 말한다.

48) 2022년 5월 현시점, TuSimple 데이터 세트에 대해서 SCNN_UNet_ConvLSTM2(CNN의 변형인 SCNN과 또 다른 인공지능망 모델인 LSTM을 이용한 차선 인식 네트워크)가 98.19%의 정확도로 최첨단 알고리즘 모델이며, LaneNet의 현재 순위는 13위이다. 현시점에서 10위권 내의 차선 인식 알고리즘은 모두 인공지능망 기반 모델이 차지하고 있다.

49) 낮, 밤 등의 다양한 시간에서 Samsung C7 Pro 휴대폰과 자전거를 이용하여 찍은 다양한 환경을 포함한 데이터 세트 (Satti, Satish Kumar, et al. "A machine learning approach for detecting and tracking road boundary lanes." ICT Express 7.1 (2021): 99-103.)

50) 빛의 광자 수가 적은 어두운 상태

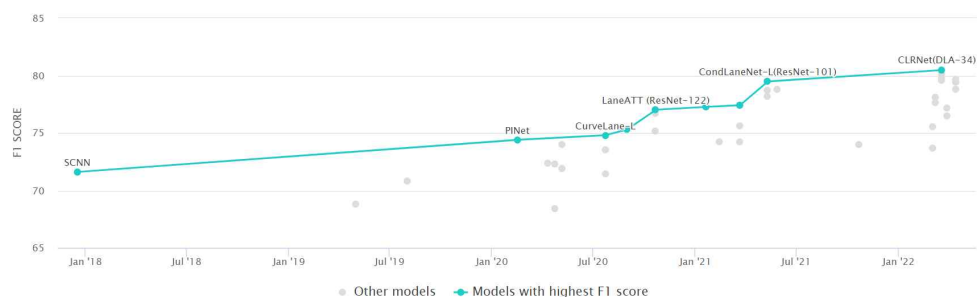
환을 이용한 기존의 차선 인식 방법과 CNN만을 이용한 방법에 비해 더 높은 정확도를 보여주었다.

4.4. 차선 인식 기술의 전망

이전 장까지 차선 인식의 기본적인 개념과 인공지능망을 이용한 차선 인식 방법에 대해서 알아보았다. 현재 차선 인식 기술 분야는 손수 만들어진 특징 검출기를 이용한 초창기의 방법보다, 더 좋은 성능과 장점이 있는 인공지능망 기술을 주로 하여 발전하고 있다.



[그림 42] TuSimple 데이터 세트에서의 차선 인식 방법별 성능⁵¹⁾



[그림 43] CULane 데이터 세트에서의 차선 인식 방법별 성능⁵²⁾

[그림 42]과 [그림 43]는 각각 TuSimple과 CULane 데이터 세트에 대해 적용한 차선 인식 방법들을 차선 인식 방법이 개발된 시기와 성능으로 나타낸 그래프이다. 그래프를 살펴보면 상위권에 있는 대부분 방법들이 인공지능망을 사용한 방법이므로 차선 인식 방법 이름에 인공지능망을 의미하는 “Net”이라는 접미어가 붙은 것을 확인할 수 있다.

최근 차선 인식 분야에서는 다양한 발견법적 방법과 조합하거나, 심층 신경망을 구성 하기에 최적화된 신경망 모델인 ResNet⁵³⁾을 이용하여 더 높은 성능을 가지는 차선 인식 신경망을 구성하는 연구가 진행되었다. 그리고 기존 합성곱 신경망에서 정확도를 희생하여 합성곱 연산의 연산량을 획기적으로 줄인 MobileNet과 같은 변형 합성곱 신경망을

51) 각 차선 인식 방법(논문)이 나온 시점과 해당 인식 방법의 정확도를 그래프로 나타낸 그림이다.

52) F1 Score는 인공지능망의 훈련 간 사용되는 데이터 세트와 검증 간 사용되는 데이터 세트의 정확도 관계를 고려한 인공지능망의 성능 평가 지표이다. 0~1의 사이를 가지며 이 값이 클수록 더 좋은 성능의 신경망으로 평가된다.

53) [그림 43]의 LaneATT(ResNet-122), CondLaneNet-L(ResNet-101) 모델이 이 ResNet을 이용한 방법들이다.

이용하여 실제 제품에 적용되기에 쉬운 차선 인식 신경망이 연구되기도 했다.

이렇게 정확도와 성능 두 마리의 토끼를 잡은 신경망 기반 차선 인식에 대한 전망과 수요는 밝으며, 실제로 최근의 많은 자동차 제조 회사에서는 이러한 기술을 적용한 차선 유지 보조 시스템(LKAS), 차선 이탈 경보 장치(LDWS) 등을 탑재한 제품을 개발하고 있다.

5. SWOT 분석

5.1. Strength

전용 지표나 신호를 따라가게 만드는 것이 아닌 기존에 그려진 차선을 인식하여 능동적으로 주행하는 것이므로, 도로 적응력이 높고, 차선 인식을 사용하지 않는 다른 주행 방법에 비해서 새로운 환경에서도 비교적 어렵지 않게 주행할 수 있다.

주변 환경을 복합적으로 인식하여 차선을 인식하는 경우, 정확도의 향상과 함께 주변의 사물이나 사람에 대한 정보도 수집하므로 운전자에게 유용한 정보를 제공할 수 있다.

5.2. Weakness

기술개발이 어느 정도 정점에 올랐으나 아직 완벽한 기술이 아니므로 일부 오판의 여지가 있다. 이에 기기가 오작동하여 교통사고를 유발하는 등의 문제가 생길 수 있다.

특히 기술 결함으로 재산 또는 인명 피해가 발생하면 이 책임을 누구에게 물어야 하는지 등의 현실적인 문제가 발생할 수 있다. 상술한 이유로 본 기술의 수혜자 또는 이용자가 될 운전자의 신뢰도가 하락할 수 있으며 이를 해결하는 것도 차선 인식과 자율주행 분야의 숙제가 될 것이다.

5.2.1. 해결 방안

첫 번째로 지적했던 오판의 문제는 위험 부분에서 이야기할 바와 같이 데이터 세트의 크기 및 다양성과 연관이 있을 수 있다. 즉, 데이터 세트의 크기와 다양성을 개선함으로써 일부 개선이나 해결이 가능할 것이다. 이러한 데이터 세트를 수집하는 단체로는 구글과 같이 지도 서비스를 운영하는 기업이나 도로 사정을 관리하는 지자체 등이 있다. 이러한 기업이나 지자체와 계약을 체결하여 데이터를 수집하는 방법이 있을 수 있다.

한편 소비자의 신뢰를 쌓기 위해 간단한 문제부터 차근차근 해결하는 것도 방법이 될 수 있다. 예를 들어 통상적으로 생각하는 자율주행은 도심에서의 자율주행이다. 그런데 사실 도심에서의 자율주행은 보행자나 주변 사물 등 변수가 많아 기술개발이 어려운 편이다. 반면에 고속도로에서의 화물 운송 등 변수가 통제된 환경에서 주행하는 차량의 경우 그 난이도가 낮아질 수 있다.

실제로 벤츠에서는 도심 속 자율주행보다 화물 트럭과 같이 고속도로를 달리는 화물 트럭에 대한 자율주행 기술에 힘쓰고 있다. 특히 화물 운송에서는 차선 인식뿐 아니라 군집 주행이라 일컫는 기술을 적용해 주변 상황을 판단할 수 있는 요소를 더해 안정성을 높였다. 이와 비슷하게 차선 인식이나 자율주행을 수행할 때 사용하는 데이터의 가짓

수⁵⁴⁾를 늘림으로써 안정성을 높이는 방안을 생각해볼 수 있다.

5.3. Opportunity

최근 자동차 시장의 흐름은 자동차의 출력이나 연비와 같은 기능적인 측면에서만 개선하는 것에 머무르지 않고 그 안에서 동작하는 소프트웨어를 함께 개선하고 있다. 예를 들어 테슬라에서 출시한 차는 자동차인 동시에 전자기기라는 평가를 종종 받는다. 그만큼 차량에 탑재되는 소프트웨어가 중요해졌다는 방증이다.

즉, 시간이 지날수록 자동차는 소프트웨어가 차지하는 비중이 커질 것이고 그 소프트웨어에는 자율주행도 포함된다. 이 자율주행 소프트웨어에 본문에서 이야기한 차선 인식 기술을 적용한다면 시장을 확보할 수 있다. 한편 자동차 시장의 흐름이 내연기관에서 전기모터로 전환됨에 따라서 배터리 기술이 주목받고 있다.

배터리 기술은 전기차를 구동하는 데 필수적으로 요구되는 기술이므로 전기차의 성능을 결정짓는 요소가 된다. 이때 자율주행차량은 운전자와는 달리 현재 차량의 상태를 자세히 파악할 수 있으므로 이를 이용해 배터리 효율을 높게 끌어낼 수 있을 것이다.

5.4. Threat

본 기술의 위협은 약점에서 언급했던 기술 결함과 관련이 있다. 약점에서 이야기했던 오판의 문제는 인공지능의 발전으로 해결할 수 있겠지만 그보다 중요한 것은 인공지능이 학습할 때 사용하는 데이터에 있다. 아무리 인공지능이 뛰어나다 하더라도 데이터에 결함이 있거나 수가 적으면 무용지물이다. 따라서 이 데이터의 확보는 인공지능에 아주 중대한 사안이다.

그러나 상술했던 테슬라를 비롯한 여러 기업들은 자신들의 소프트웨어에 사용된 인공신경망 모델이나 데이터 세트를 공개하지 않고 있다. 이는 결국 데이터 세트의 규모, 인공신경망 모델 등을 놓고 경쟁하는 꼴이 된다. 그런데 후발주자의 경우 선발주자가 수집한 데이터의 양과 질을 따라가기 힘들고, 이와 더불어 복잡한 인공신경망 모델을 개발하기도 힘들 것이다. 이는 기술 발전의 발목을 잡을 수 있다.

또한 위와 같은 이유로 특정 기업의 기술이 비약적으로 발전, 시장을 선점 및 독점하는 경우가 생길 수 있다. 이런 현상이 발생하면 후발주자의 시장 진입이 매우 어려워지며 기술 발전은 물론 소비자에게 제공되는 서비스의 질도 하락할 수 있다.

5.4.1. 해결 방안

위협에서는 선발주자의 독과점으로 인한 기술개발의 침체를 지적했다. 독과점이 발생하면 선발주자는 유리한 위치에 서게 되므로 기술개발 속도가 더뎌지고 후발주자는 시장 진입조차 어려워진다. 이러한 현상은 기업 간 협약 체결을 통해 해결할 수 있겠지만 이는 현실적으로 어렵다. 따라서 정부 기관과 같이 이를 조정할 수 있는 기관이 개입하여

54) BDD100K와 같이 다양한 데이터가 포함된 데이터 세트를 통해 차선을 인식하는 분야는 단순히 이미지만을 사용한 차선을 인식하는 분야보다 발전이 더디다. 비단 데이터 구축 뿐만 아니라, 복합 데이터 세트를 활용한 차선 인식 분야에 대한 연구와 지원도 같이 이루어져야 할 것이다.

해결할 수 있다. 정부 기관의 개입으로 인한 해결은 이미 자리를 잡은 선발주자를 제지하는 것보다는 후발 주자에게 지원해주는 방향으로 진행해야 할 것이다.

6. 참고자료

Precedence Research - Autonomous Vehicle Market

<<https://www.precedenceresearch.com/autonomous-vehicle-market>>

YouTube - This Is What Tesla's Autopilot Sees On The Road

<<https://www.youtube.com/watch?v=fKXztwtXaGo>>

Wikipedia - Vanishing point

<https://en.wikipedia.org/wiki/Vanishing_point>

Wikipedia - Sobel operator

<https://en.wikipedia.org/wiki/Sobel_operator>

Wikipedia - Deep learning

<https://en.wikipedia.org/wiki/Deep_learning>

Wikipedia - Hough transform

<https://en.wikipedia.org/wiki/Hough_transform>

TuSimple

<<https://www.tusimple.com>>

Papers With Code – Lane Detection on TuSimple

<<https://paperswithcode.com/sota/lane-detection-on-tusimple>>

CULane

<<https://xingangpan.github.io/projects/CULane.html>>

Papers With Code – Lane Detection on CULane

<<https://paperswithcode.com/sota/lane-detection-on-culane>>

BDD100K

<<https://www.bdd100k.com>>

Neven, Davy, et al. "Towards end-to-end lane detection: an instance segmentation approach." 2018 IEEE intelligent vehicles symposium (IV). IEEE, 2018.

Shirke, Suvarna, and R. Udayakumar. "Lane datasets for lane detection." 2019 International Conference on Communication and Signal Processing (ICCSP). IEEE, 2019.

Satti, Satish Kumar, et al. "A machine learning approach for detecting and tracking road boundary lanes." ICT Express 7.1 (2021): 99-103.