

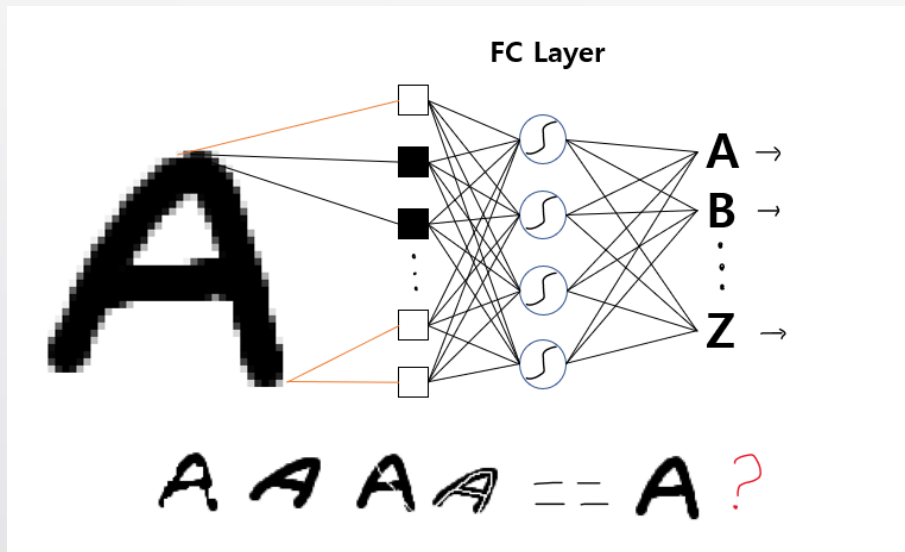


## CNN(Convolutional Neural Networks)

1. CNN은 이미지를 딥러닝 모델에 적용하기 위해 고안된 알고리즘
2. CNN은 이미지를 인식하기 위해 패턴을 찾는데 특히 유용
3. 데이터에서 직접 학습하고 패턴을 사용해 이미지를 분류 함. 즉, 특징을 수동으로 추출할 필요 없음
4. 자율주행자동차, 얼굴인식과 같은 객체인식이나 computer vision이 필요한 분야에 많이 사용

////////////////////


CNN이 나오기 이전, 이미지 인식은 2차원으로 된 이미지(채널까지 포함해서 3차원)를 1차원배열로 바꾼 뒤 FC(Fully Connected)신경망으로 학습시키는 방법 사용



그림과 같이 이미지의 형상은 고려하지 않고, raw data를 직접 처리하기 때문에 많은 양의 학습데이터가 필요하고 학습시간이 길어짐.

또한 이미지가 회전하거나 움직이면 새로운 입력으로 데이터를 처리 해줘야함. 이미지의 특성을 이해하지 못하고 단순 1D데이터로 보고 학습을 하는 것이 특징

이러한 방법은 이미지 데이터를 평면화 시키는 과정에서 공간정보가 손실될 수밖에 없음.  
즉, 신경망이 특징을 추출하고 학습하는데 있어 비효율적이고 정확도를 높이는데 한계 있음.  
이런 단점을 보완하여 **이미지의 공간정보를 유지한 상태로 학습**을 하게하는 모델이 **CNN** 임.



CNN(Convolutional Neural Network)은 기존 Fully Connected Neural Network와 비교하여 다음과 같은 차별성이 있음

- 각 레이어의 입출력 데이터의 형상 유지
- 이미지의 공간 정보를 유지하면서 인접 이미지와의 특징을 효과적으로 인식
- 복수의 필터로 이미지의 특징 추출 및 학습
- 추출한 이미지의 특징을 모으고 강화하는 Pooling 레이어
- 필터를 공유 파라미터로 사용하기 때문에, 일반 인공 신경망과 비교하여 학습 파라미터가 매우 적음

////////////////////

CNN의 가장 핵심적인 개념은 **이미지의 공간정보를 유지한채 학습**을 한다는 것.

## 1. Convolution

- Convolution의 사전적 정의는 합성곱
- Convolution은 처음 등장한 개념이 아니라 CNN이 등장하기 한참 전부터 이미지처리에서 사용되었던 개념

1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved  
Feature

Stride : gif처럼 한칸씩 이동시키면서 적용 시킬 건지,  
두칸씩 이동할 건지 정하는 것

이를 통해 이미지의 feature map을 만들 수 있음  
filter(또는 kernel)의 구성에 따라 이미지의 특징을  
뽑을 수 있음

## 2. Filter(Kernel)

- Filter는 이미지의 특징을 찾아내기 위한 공용 파라미터

예시 : Filter를 통해서 이미지의 feature를 확인

sobel filter : 이미지의 가로세로 feature를 확인 할 수 있는 필터



원본이미지



-1	0	+1
-2	0	+2
-1	0	+1

Sobel-X  
(vertical)

+1	+2	+1
0	0	0
-1	-2	-1

Sobel-Y  
(horizontal)



(왼쪽: sobel-x적용, 오른쪽: sobel-y적용)



feature

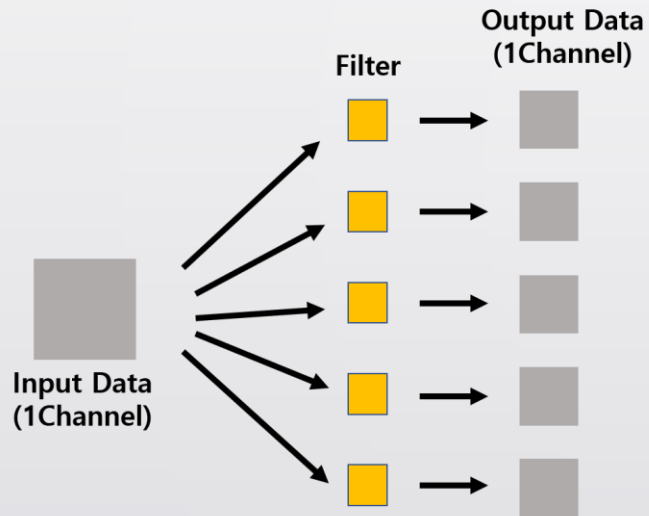
### <특이사항>

이미지 처리에서는  
sobel필터와 같이 유명한  
필터들을 직접 사용자가  
찾아서 사용 했으나,  
CNN은 신경망에서 학습을  
통해 자동으로 적합한  
필터를 생성해 줌

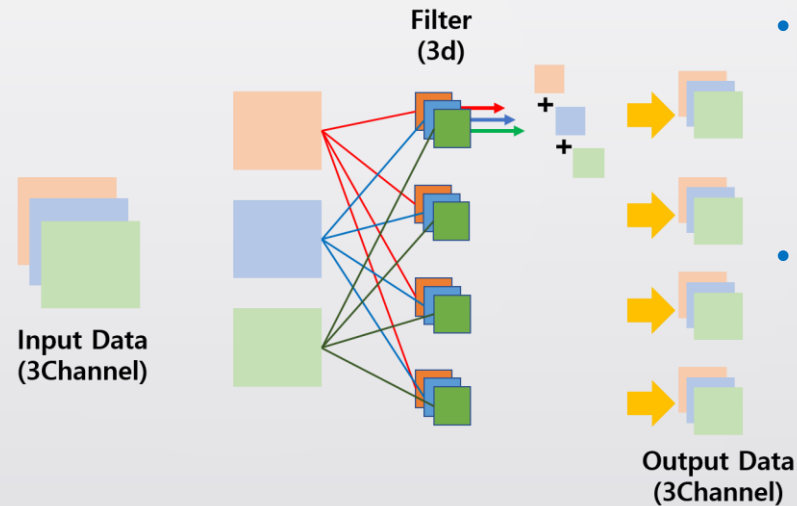


### 3. Channel

- color이미지는 red채널, blue채널, green채널이 합쳐진 이미지
- 즉, 하나의 color이미지는 3개의 채널로 구성
- 연산량을 줄이기 위해(오차를 줄이기 위해) 전처리에서 이미지를 흑백(Channel:1)으로 만들어 처리



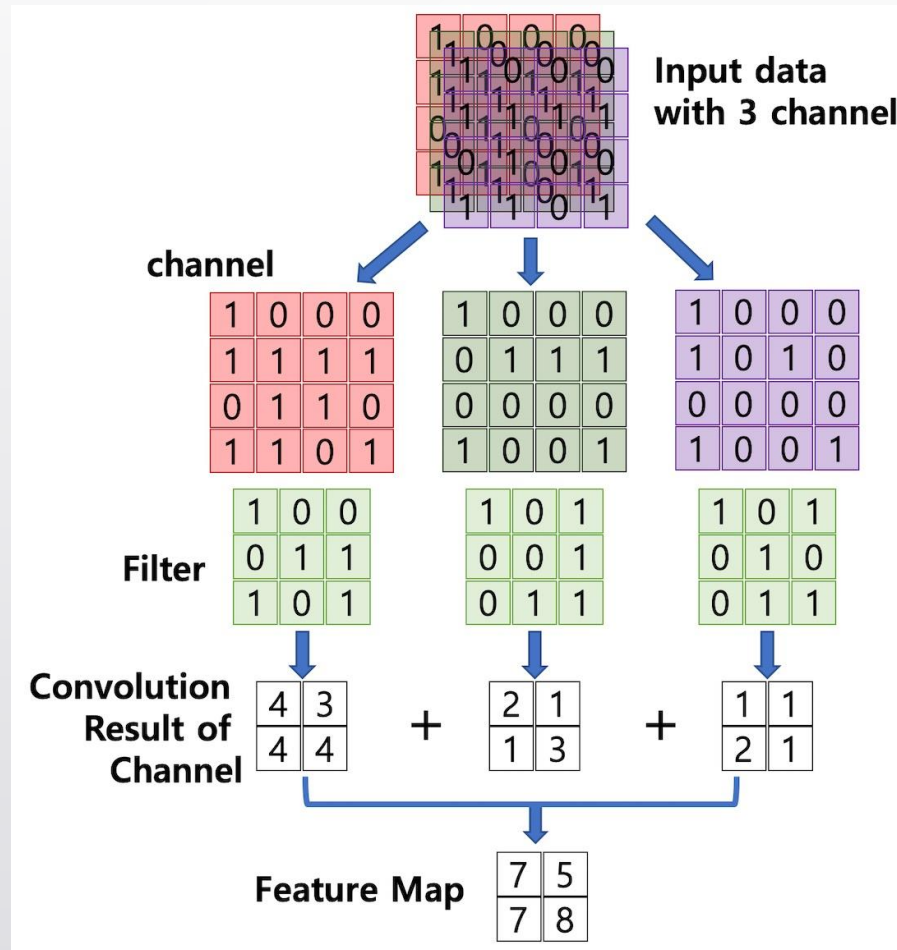
흑백이미지(1 channel)



컬러이미지(multi channel)

- input data의 channel수와 filter의 channel수는 같아야 함
- input data의 channel의 수와 관계없이 filter의 개수만큼 output데이터가 나옴

- Multi Channel 입력 데이터에 필터를 적용한 합성곱 계산 절차



### Feature Map(Activation Map)

Convolution Layer의 입력 데이터를  
필터가 순회하며 합성곱을 통해서  
만든 출력

## 4. Padding

- Convolution 레이어에서는 Filter와 Stride의 작용으로 Feature map의 크기는 입력데이터보다 작음
- 입력데이터보다 출력데이터가 작아지는 것을 방지하는 방법이 Padding
- padding을 하게 되면 convolution을 해도 크기가 작아지지 않음

0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	0	0	0	0	0	0

Image  
(+zero Padding)

패딩은 입력 데이터의 외곽에  
지정된 픽셀만큼 특정 값으로 채워  
넣는 것을 의미  
(보통 패딩값으로 0을 채움)



## 5. Pooling

- 풀링 레이어는 컨볼루션 레이어의 출력 데이터를 입력으로 받아서 출력 데이터(Activation Map)의 크기를 줄이거나 특정 데이터를 강조하는 용도로 사용
- 처리하는 방법 : Max Pooling, Average Pooling, Min Pooling
- 최댓값을 구하거나 평균을 구하는 방식으로 동작
- 일반적으로 pooling의 크기와 stride의 크기를 같게 설정하여 모든 원소가 한번씩은 처리가 되도록 설정

13	20	30	0
8	12	3	0
34	70	33	5
111	80	10	23

Activation Map

20	30
111	33

Max Pooling

13	8
66	18

Average Pooling

8	0
34	5

Min Pooling

<특징>

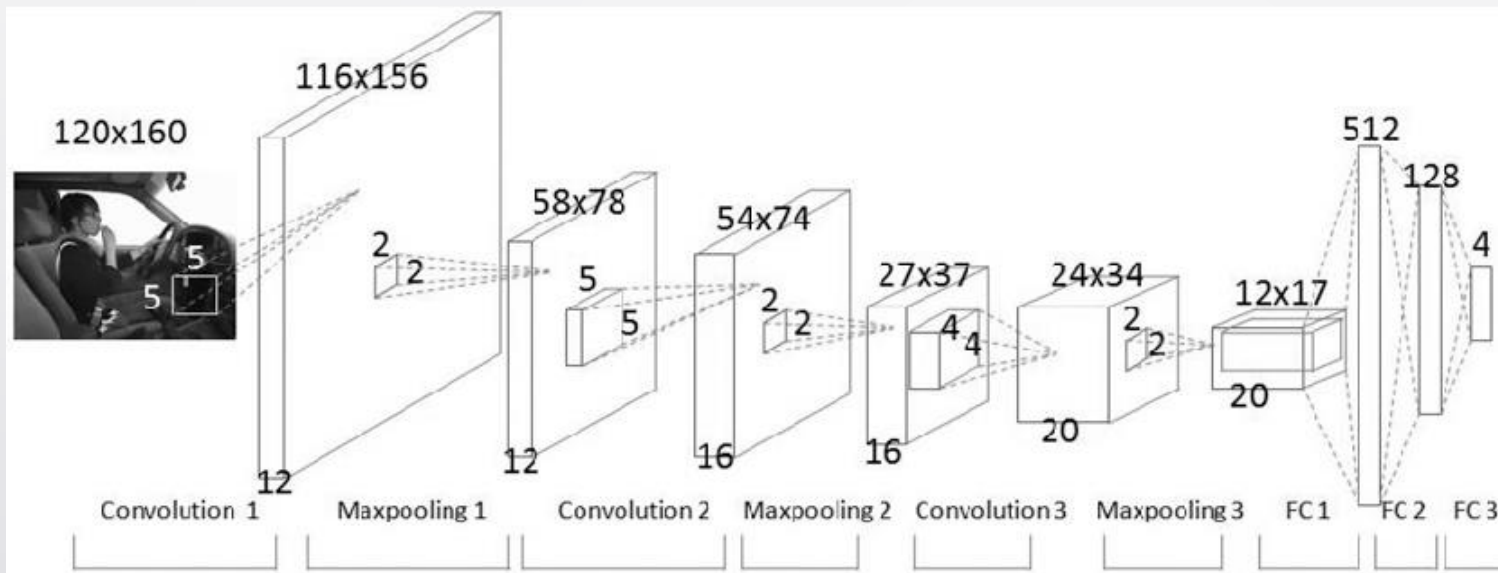
학습대상 파라미터가 없음

Pooling 레이어를 통과하면 행렬의 크기 감소

Pooling 레이어를 통해서 채널 수 변경 없음

## 6. 요약

- CNN은 Convolution과 Pooling을 반복적으로 사용하면서 불변하는 특징을 찾고, 그 특징을 입력데이터로 Fully-connected 신경망에 보내 Classification을 수행



## 전형적인 CNN 구성

Convolution Layer와 Max Pooling  
레이어를 반복적으로 stack을 쌓는  
특징 추출 (Feature Extraction) 부분과  
Fully Connected Layer를 구성하고  
마지막 출력층에 Softmax를 적용한  
분류 (Classification) 부분으로  
나누어짐