

## Lecture 5 | Convolutional Neural Networks

### [History]

Perceptron -> Back propagation 개념이 나오고

LeNet(1998)을 시초로 CNN의 개념이 적용됨

그 후 많은 양의 데이터를 사용할 수 있게 되고, LeNet보다 더 크고 깊은 AlexNet이 등장

이후 ConvNet은 Classification, Detection, Segmentation, Image Captioning, 자율 주행 차 등 많은 분야에서 사용되고 있음

### [Convolutional Neural Networks]

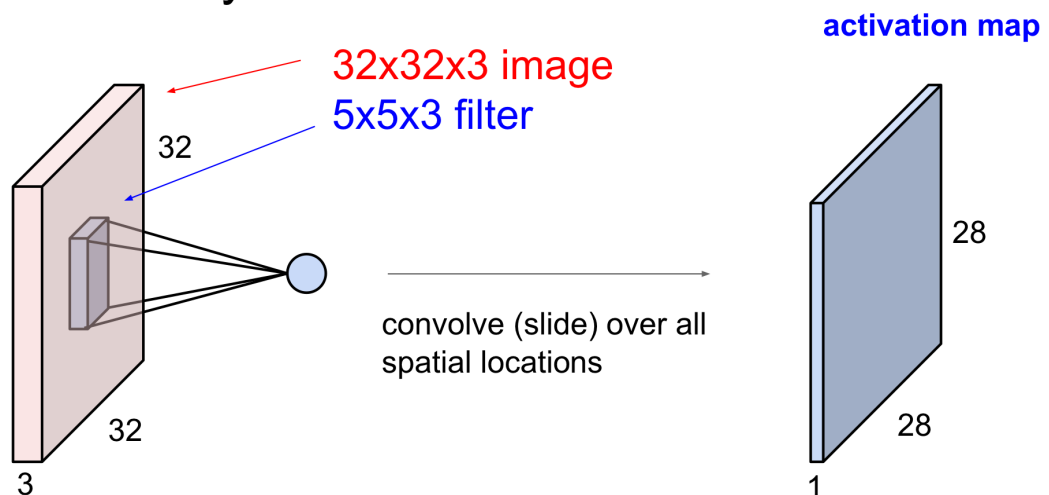
- FC Layer vs. Conv Layer

32x32x3 이미지가 있을 때,

FC Layer는 인풋 이미지를 stretch(to 3072x1)하여 사용하지만,

Conv Layer는 이미지 그 자체를 인풋으로 사용.

## Convolution Layer

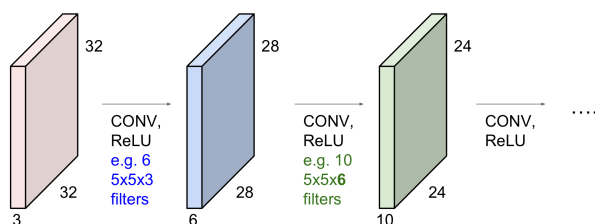


32x32x3 input image로부터 filter(5x5x3) 크기 만큼에 해당하는 이미지의 픽셀을 필터의 각 W와 곱한다. 이러한 filter를 슬라이딩하며 반복.

-> 여러 개의 필터를 사용하여 여러가지 activation map들을 얻을 수 있다.

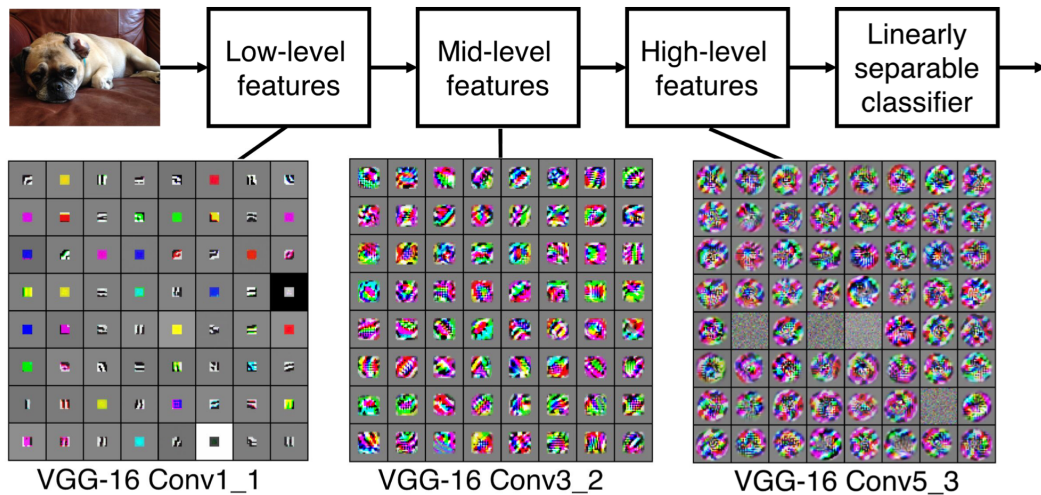
\*. 둘의 depth가 같아야 한다.

**Preview:** ConvNet is a sequence of Convolutional Layers, interspersed with activation functions



CONV, ReLU + (Pooling)를 반복하다 마지막에 FC-layer

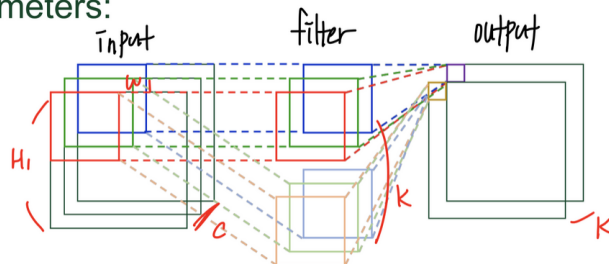
\*. Pooling을 통해 activation map의 사이즈를 줄여준다.



특징 추출 시각화

### Conv Layer 요약

Parameters:



- input feature map:  $W_1 \times H_1 \times C$
- Convolution layer의 4가지 hyperparameters:
  - filter의 개수 ( $K$ )
  - filter size ( $F$ )
  - stride ( $S$ )
  - zero padding ( $P$ )
- output feature map:  $W_2 \times H_2 \times K$ 
  - \*. output의 채널 수 = 필터의 개수
- parameter의 개수:  $F^2 C K$ 개의 weights와  $K$ 개의 bias
  - $F \times F \times C$  크기의 필터가  $K$ 개 있음 (bias는 3D 필터 하나 당 한개)

\*. 예시 -> 강의자료 참조

[Pooling Layer]

- 이미지 사이즈를 줄여주는 역할  
e.g. max pooling

[FC Layer]

: 마지막에 사용하여 값들 분류