**Paper review : Very Deep Convolutional Networks For LARGE-SCALE IMAGE RECOGNITION**

Abstract.

* Investigate the effect of the convolutional network **depth** on it’s accuracy in the large-scale image recognition setting

: cnn depth가 **large-scale image recognition의 setting에 미치는 영향**에 대해 탐구

* Main contribution : very small convolution filters( 3\*3 )

: significant improvement < by pushing the depth to 16-19 weight layers.

Introduction,

* A number of attempts have been made to improve the original architecture(Krizhevsky et ai 2012)

: ILSVRC utilized smaller receptive window size and smaller stride of the first convolutional layer.

* \*\*\* This paper address another important aspect of ConvNet architecture design- it’s depth.

**: 다른 파라미터의 값은 고정시킨 상태로, convolutional layers를 추가함에 따라 점진적으로 네트워크의 depth를 증가시켰으며, 이를 가능케 한 주 요인은 모든 layers에 굉장이 작은(3\*3) convolutional filters를 layers들에 적용시켰기 때문이다.**

**Prev : 위에서 언급한 바와 같이, 가장 큰 특징들을 정리해보자면,**

**-점진적으로 네트워크의 depth를 증가시킴**

**-굉장히 작은 convolutional filter를 적용 하였/다는 것인데, 이 각 특징에 해당하는 세부 이론을 정리해보도록 하겠다.**

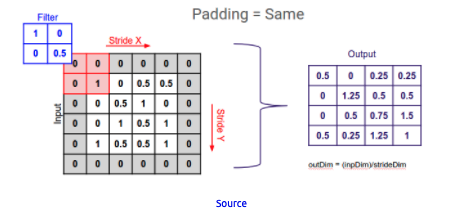
**: convolutional layer를 통해 이미지의 특징을 추출 및 해당 값이 신경망과 연결되게 된다.**

**: layer내에 다중 필터와 filter 와 activation function이 계층을 이룬 형태로 구조화 되어있다.**

**:stride의 개념**

* **필터를 적용하는 간격을 의미한다.**
* **해당 stride를 통해 filter를 적용시켜 얻은 값을 feature map or activation map이라 한다.**
* **이러한 single filter를 다중 필터로 적용함에 따라 activation maps가 형성되게 된다.**

**: padding의 개념**

****

* **패딩의 근본적인 목적은 feature map이 원본사이즈에 비해 작아짐에 따라 유실될 수 있는 특징을 보호하기 위해 원본의 크기대로 0값을 둘러싸준다. 이렇게 되면 Filter를 통과한 결과 값의 크기를 원본과 동일하게 가져갈 수 있음과 동시에 피쳐가 소실되는 것을 막을 수 있게 된다.**

**: pooling의 개념**

* **Pooling을 다시 풀이하면 sub sampling이라 볼 수 있다.**

**: Convolutional layer를 거쳐 추출된 특징들은 필요에 따라 sub sampling과정을 거친다.**

**Convolutional layer를 통해 특징이 추출되었?다면, 이 모든 특징 동반하여 판단할 필요가 없을 수 있기 때문이다.**

**Max pooling**

**: max pooling은 MxN의 크기로 잘라낸 후, 그 안에서 가장 큰 값을 뽑아내는 방법이다.**

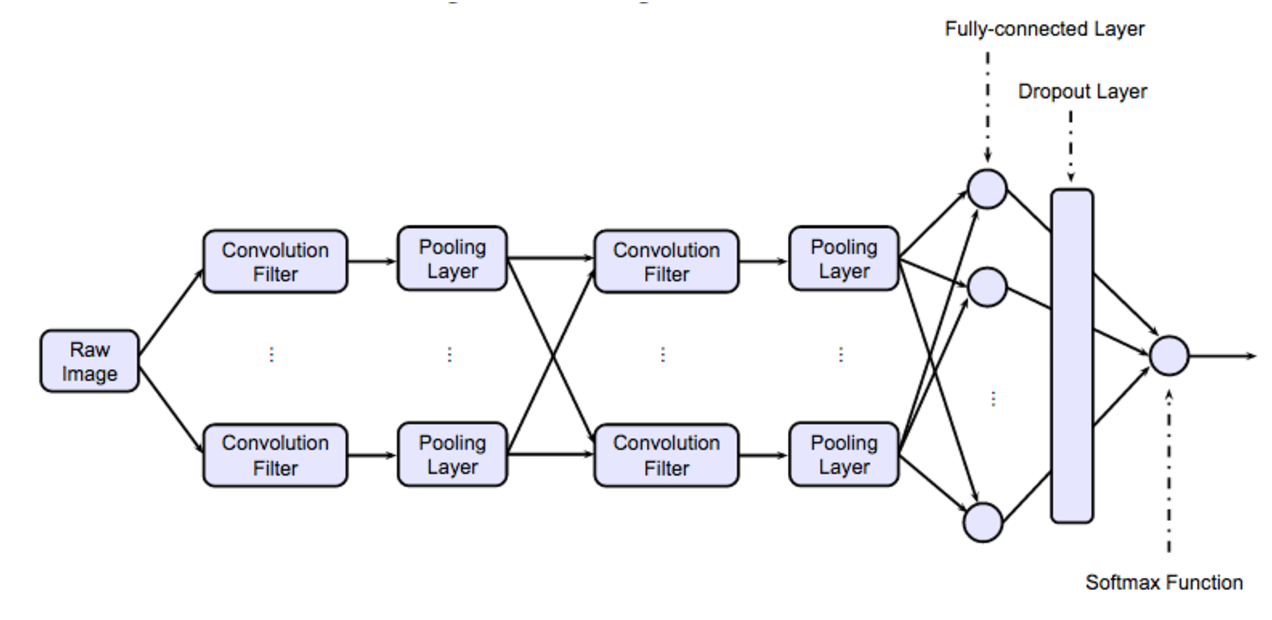
**\*\* 위의 내용을 정리하면 아래와 와 같다.**

테이블이(가) 표시된 사진

자동 생성된 설명

**: Filter > Activation Function > ReLU > Plling Layer를 반복적으로 조합하여 특징을 추출한다.**

**: Fully connected layer**



**: Drop out**

* **Fully connected network와 Softmax function의 중간에 해당 Dropout layer가 위치한다. 이는 Overfitting을 막기 위한 방법으로 learning 중 random 하게 특정 뉴런의 학습을 제외시키는 프로세스를 중간 삽입한다.**

**Architecture:**

* **During training, the input to our ConvNet is fixed-size 224 \* 224 rgb image.**
* **\*\* Subtracing the mean rgb value, computed on the training set, from each pixel.**

**: why?**

* **Get center the data.**
* **It is able to normalize to z score**
* **The reason** we do both of those things is because in the process of training our network, we're going to be multiplying (weights) and adding to (biases) these initial inputs in order to cause activations that we then backpropogate with the gradients to train the model. (**step for training moel**)
* **In one of configurations they also utilize 1 \* 1 convolution filters, which can be seen as a linear transformation of the input**
* Convolution stride : 1 pix