

##0315

2021년 3월 1일 월요일 오후 4:52

■ 7장. 합성곱 신경망

□ 7.5 CNN 구현하기[code]

File names : ch07 folder

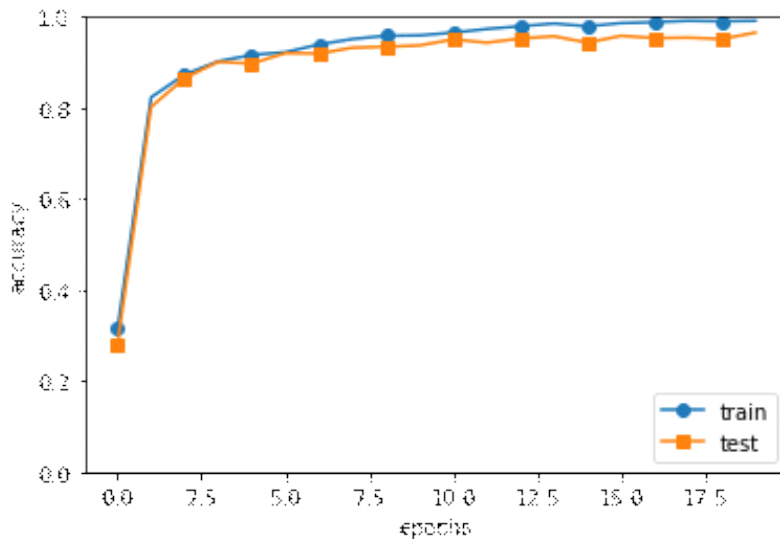
simple_convnet.py # CNN (단순 합성곱 + 풀링 구현)

layers.py #Layer(층) 관련 코드 ex. sigmoid, affine, convolution 등

train_convnet.py #실제 학습 코드(객체 생성 및 그래프 생성)

trainer.py #신경망 학습 코드

학습완료 후 출력되는 그래프



epochs(학습)가 증가할수록 accuracy 가 수렴하는 모습을 보인다.

학습시간이 너무 길어 trainset rows:5000, testset rows:1000 으로 한 결과

학습이 진행될수록 오버피팅이 소폭 발생하는 모습을 보여준다(variance)

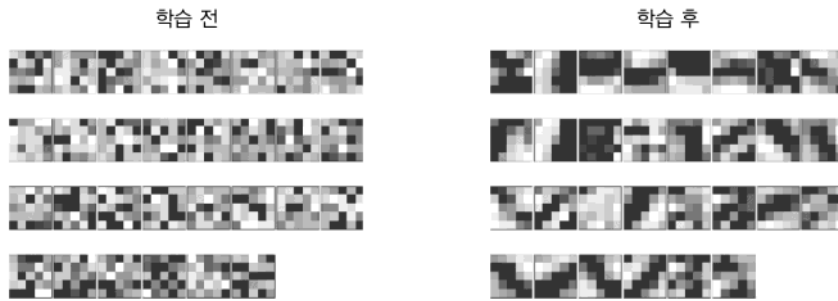
코드의 구현 순서



□ 7.6 CNN 시각화하기

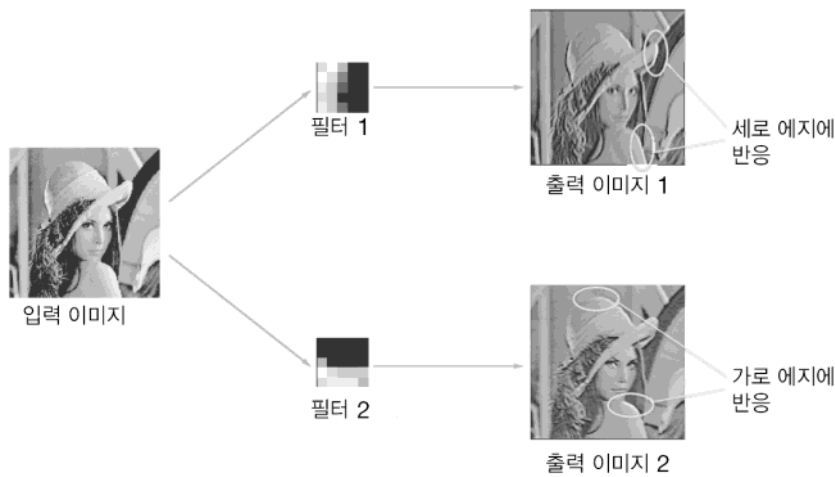
● 7.6.1 첫번째 층의 가중치 시각화하기

[그림 7-24]



- # 합성곱 계층(1층)의 학습전과 후(학습된 가중치 .pkl 포함)의 이미지
- # 합성곱 계층의 가중치의 형상 : (30, 1, 5, 5)
- # 필터 30개, 채널 1개, 5x5 필터크기
- # 필터에 규칙성이 생겼다 -----> 어떤 것을 학습해서? -----> trainset 의 Loss(오차)가 최소가 되는 기울기(가중치)
- # 위와 같은 규칙성이 의미하는 바는 무엇인가?

[그림 7-25]



- # 합성곱 계층의 필터는 에지나 블롭(덩어리) 등의 원시적인 정보를 추출할 수 있다
- # 합성곱 계층은 이미지를 인식하기 위한 원시정보를 파악

● 7.6.2 층 깊이에 따른 추출 정보 변화

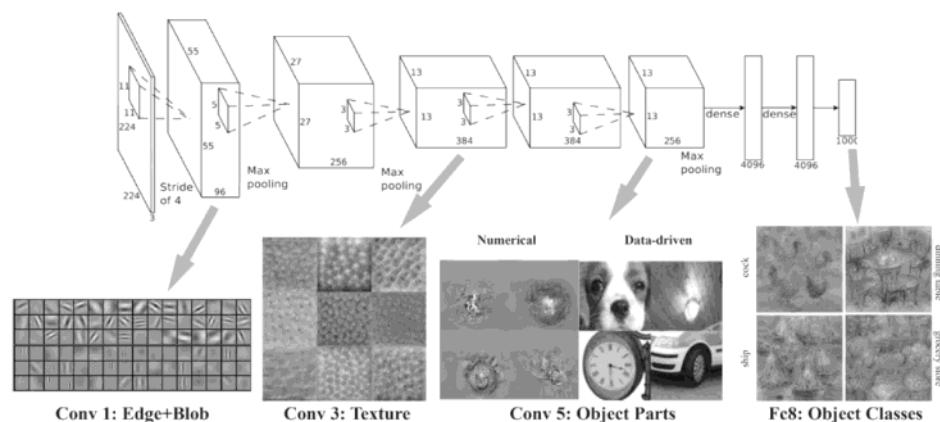
계층이 깊어질수록 추출되는 정보(강하게 반응하는 뉴런)은 더 추상화된다.

더 강하게 반응한다는 것은 어떤 의미인가??

계층이 깊어질수록 더 강한 신호(max pooling 을 계속 진행하므로서 가장 센 값이 끝까지 남게 됨)가 해당 합성곱 Layer 에 위치

Convolution - pooling ~ Relu ~ - Convolution - pooling - ~ Relu -

[그림 7-26]



- # 층이 깊어질수록 더 복잡한(raw 데이터 상의 값이 큰 값) feature 을 이해하는 모습을 보인다.
- # convolution 과 max pooling 이 진행될수록 추상적인 부분을 인식하는 것을 볼 수 있다.

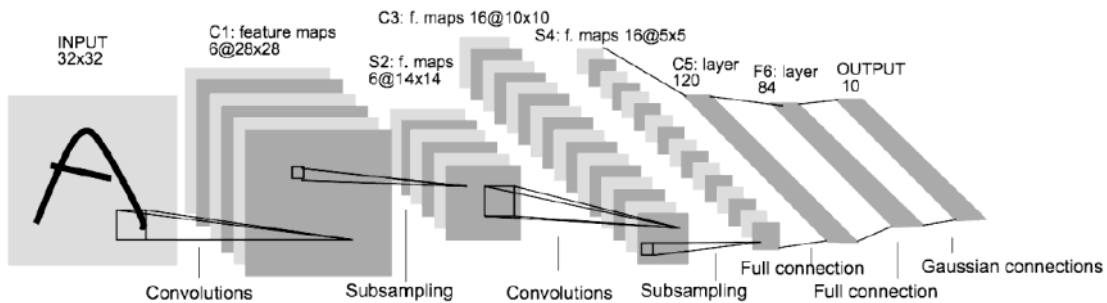
Affin 계층에서는 분류를 수행

□ 7.7 대표적인 CNN

● 7.7.1 Lenet

Lenet 은 1998년에 제안된 합성곱 계층 + 풀링 계층(subsampling)이다

[그림 7-27] Lenet



CNN과 비교되는 Lenet 의 차이점

	Lenet	AlexNet
활성화함수	sigmoid	Relau
Pooling method	Subsampling (중간데이터의 단순크기 감소)	Max pooling (최대값 추출)
etc		드롭아웃 사용
		LRN 국소적 정규화 실시 계층 사용

LRN : Local Response Normalization

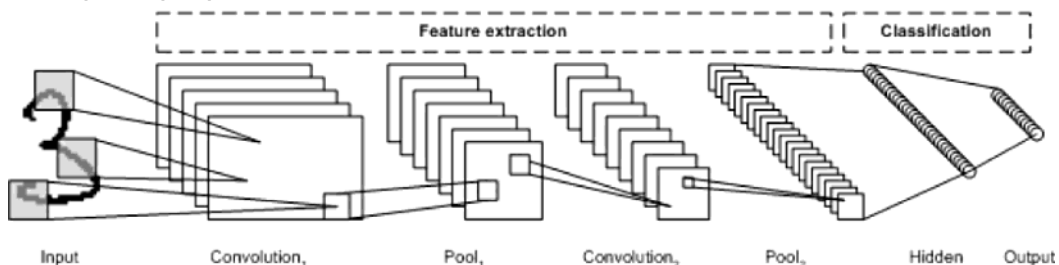
● 7.7.2 AlexNet

2012 년도에 발표

병렬 계산 특화된 GPU 보급 및 활용 가능 환경 (Lenet ---> AlexNet)

□ 추가학습

CNN 의 대략적인 구조

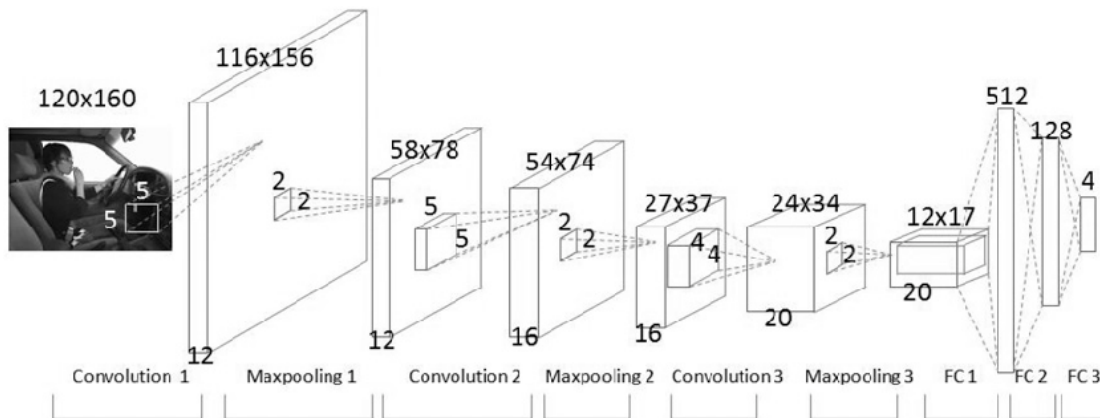


CNN 은 크게 Feature extraction(특징 추출) + Classification(분류) 로 구분

Affin(완전연결 계층) 을 사용하는 이유 : 분류

Convolution 과 Pooling 을 사용하는 이유 : Feature extraction(특징 추출)

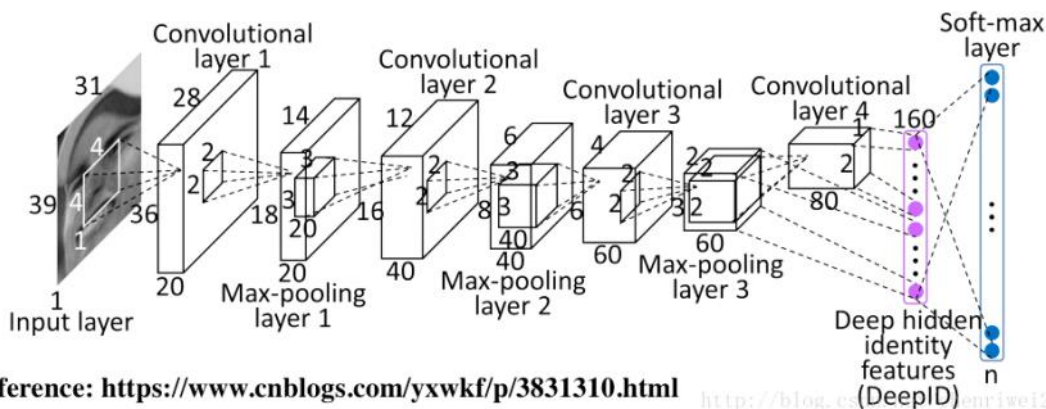
전형적인 CNN 구성



FC1, FC2, FC3 : Classification

else : Feature extraction

CNN의 실제 구현모습



Reference: <https://www.cnblogs.com/yxwxf/p/3831310.html>

합성곱 추출의 층수(Layer)가 깊어질수록 filter 크기가 작아지는 것을 관찰가능

layer4 이후 Flatten 을 수행하여 Classification 하는 모습 ----> Affin + soft-max layer

책에서는 분류 과정에서도 위와 동일한 구조를 취하고 있음

Summary

CNN(Convolutional Neural Network)은 이미지의 공간 정보를 유지하면서 인접 이미지와의 특징을 효과적으로 인식하고 강조하는 방식으로 이미지의 특징을 추출하는 부분과 이미지를 분류하는 부분으로 구성됩니다. 특징 추출 영역은 Filter를 사용하여 공유 파라미터 수를 최소화하면서 이미지의 특징을 찾는 Convolution 레이어와 특징을 강화하고 모으는 Pooling 레이어로 구성됩니다.

CNN은 Filter의 크기, Stride, Padding과 Pooling 크기로 출력 데이터 크기를 조절하고, 필터의 개수로 출력 데이터의 채널을 결정합니다.

CNN는 같은 레이어 크기의 Fully Connected Neural Network와 비교해 볼 때, 학습 파라미터량은 20% 규모입니다.

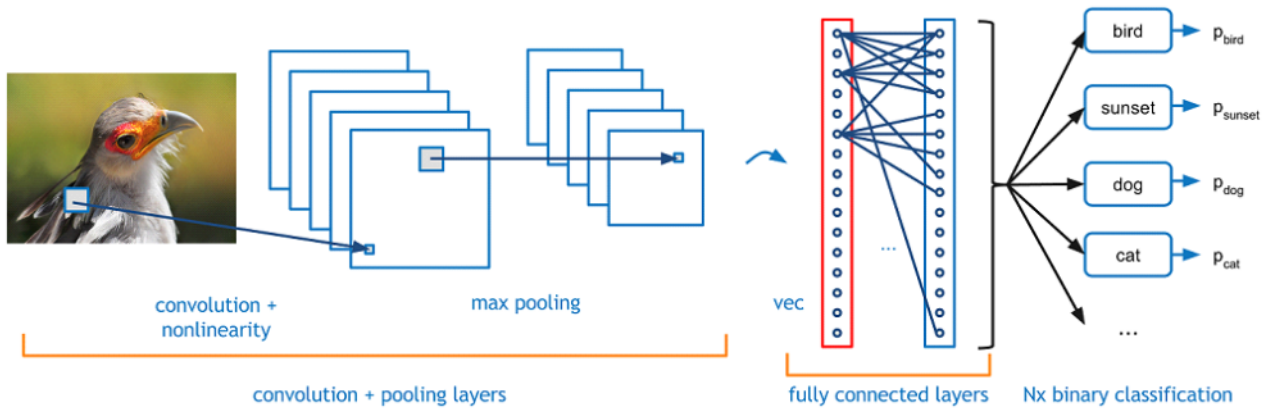
은닉층이 깊어질 수록 학습 파라미터의 차이는 더 벌어집니다.

CNN은 Fully Connected Neural Network와 비교하여 더 작은 학습 파라미터로 더 높은 인식률을 제공합니다.

추가학습 참조 url : <http://taewan.kim/post/cnn/>

□ 그 외 추가 img 자료

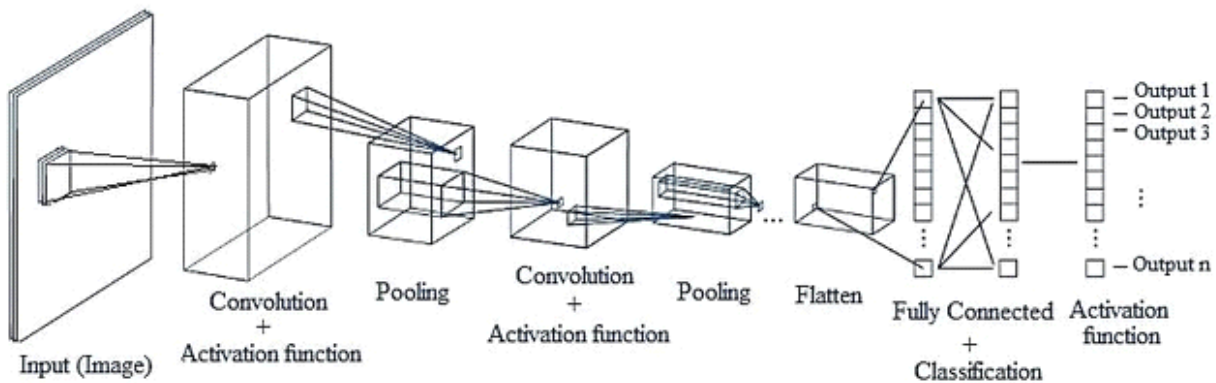
[1]



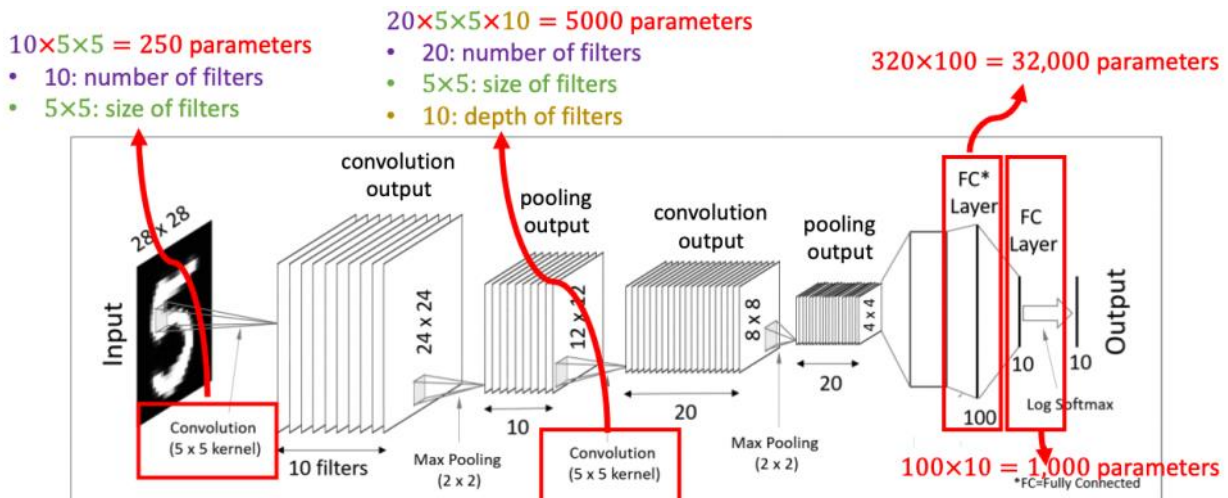
<https://sites.google.com/site/bimprinciple/in-the-news/peiseubug-uidibleoningcnnngibanopeunsosseubeon-yeoggisulcssl>

[2] # 매우 자세한 설명 # 전체의 과정이 이해가 안된다면 추천

[2-1] CNN 구조

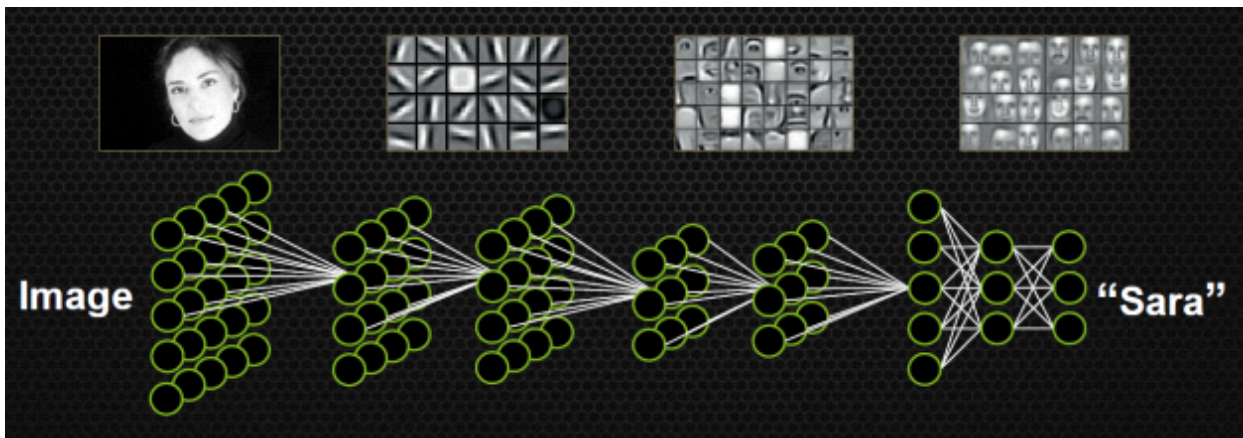


[2-2] 학습 가능한 매개변수의 수



<https://halfundecided.medium.com/%EB%94%A5%EB%9F%AC%EB%8B%9D-%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-cnn-convolutional-neural-networks-%EC%89%BD%EA%B2%8C-%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0-836869f88375>

[3] # convoulution Layer 별 이미지 인식 예시1

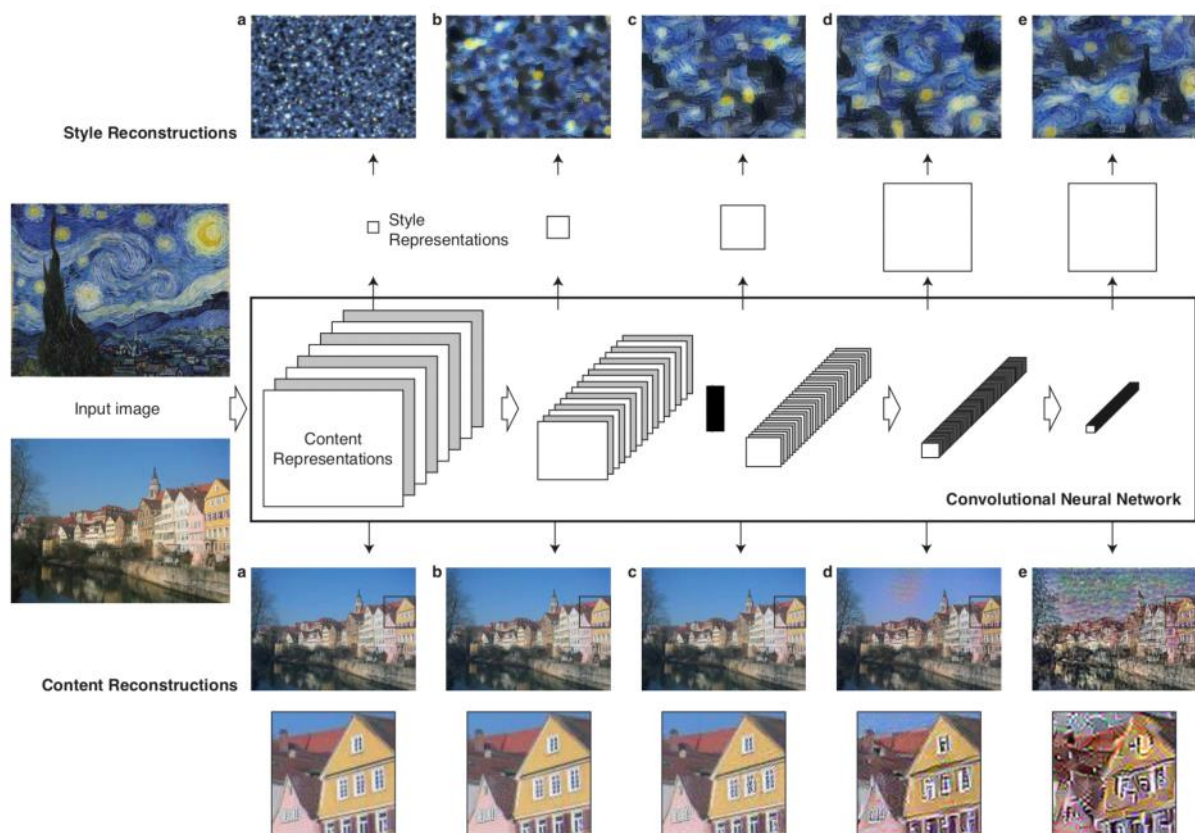


<https://developer.nvidia.com/blog/accelerate-machine-learning-cudnn-deep-neural-network-library/>

깊은 convolution layer 에서 추상화가 일어나고 있는 것을 시각화한 자료

맨 끝장의 그림은 압축된 feature 을 바탕으로 classification(분류)를 수행하는 과정

[4] Neural style transfer Algorithm # 화풍변환

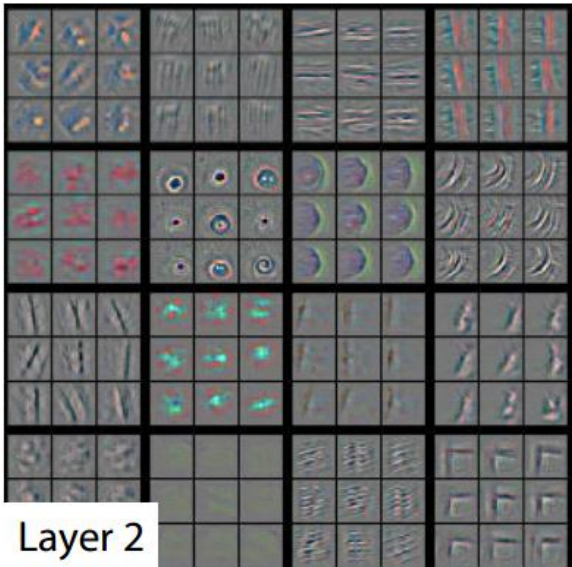


https://mjgim.me/2018/10/09/style_transfer.html

주어진 이미지의 윤곽과 같은 객체 정보를 유지한 채 다른 이미지의 스타일을 주는 알고리즘

Loss 함수와 대략적인 개념은 위 사이트 참고

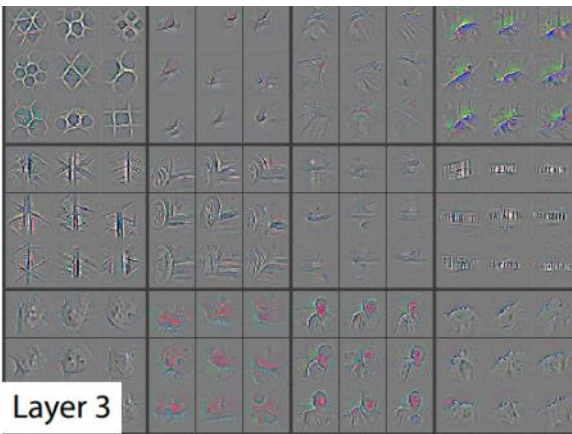
[5] Layer 에 따른 feature extraction 예시



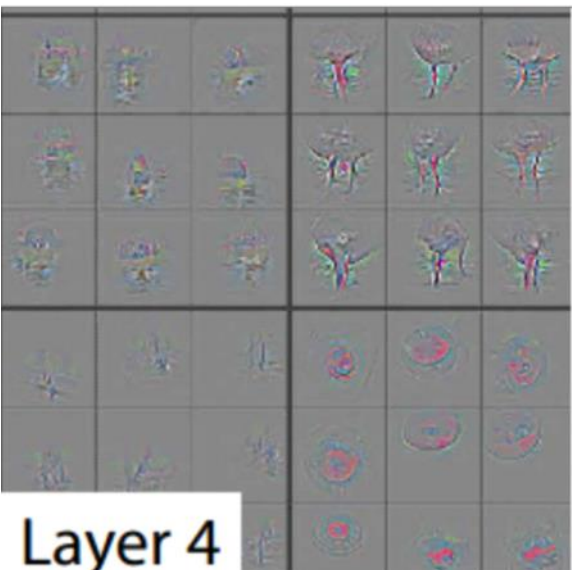
Layer 2

edge, bolum

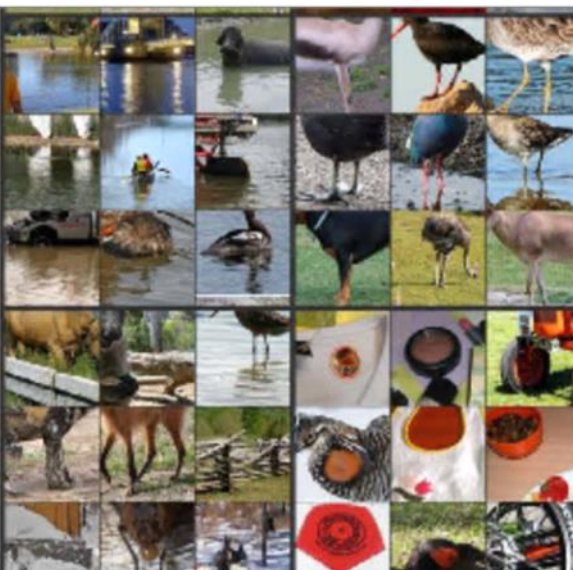
선, 덩어리를 인식

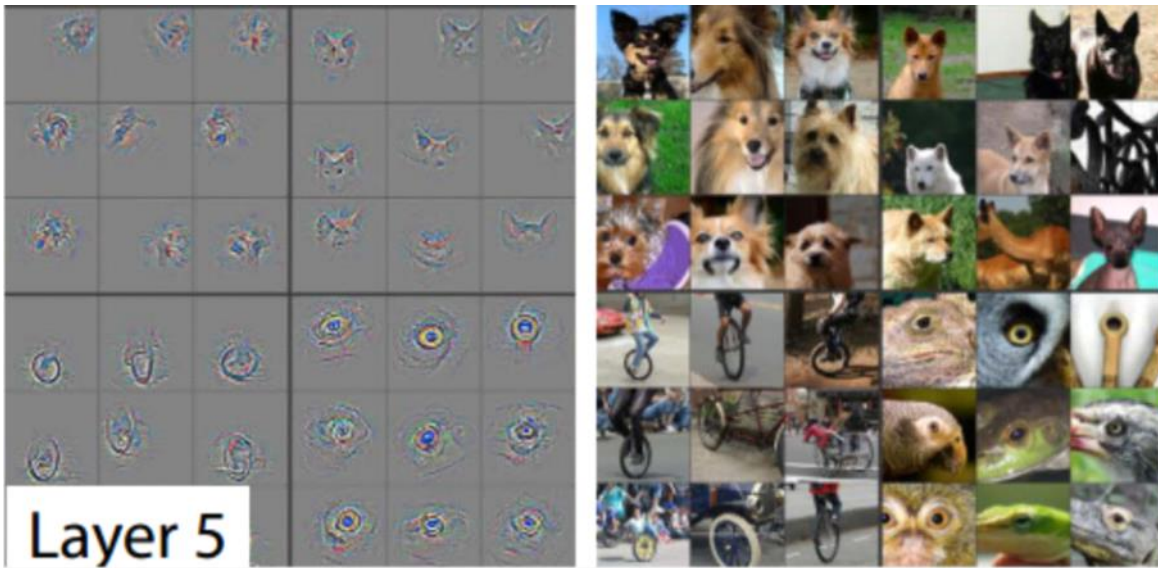


Layer 3



Layer 4





Layer 가 깊어질수록 각 이미지의 고유 feature(추상화) 이미지를 추출(학습)

<https://towardsdatascience.com/deep-learning-2-f81ebe632d5c>

#####endLine=====