VGG 논문리뷰

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

2021 보아즈 분석 학기세션
DETECTION

16기 김영은

여긴 목차 페이지입니다

01. INTRO-DUCTION

"MUCH DEEPER NETWORK, MUCH SMALLER FILTERS"

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

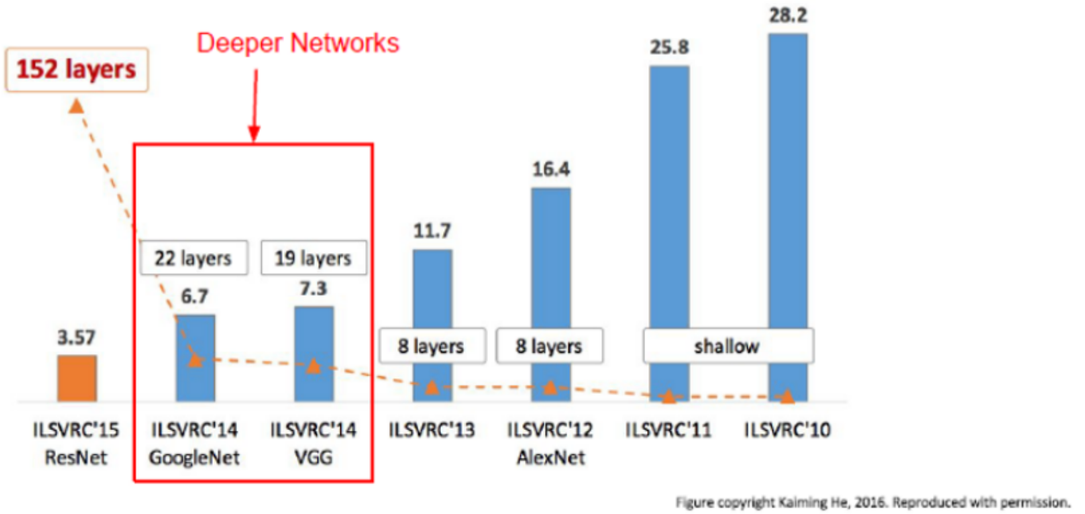


Figure copyright Kaiming He, 2016. Reproduced with permission.

2014년 처음으로 씐 더 깊은 deeper 네트워크 등장

from oxford

VGG : Visual Geometry Group

16 : 레이어수

ImageNet에서 AlexNet의 오차율을 절반으로 줄임

Localization 1위 & Classification 2위

Top-5 테스트 정확도 92.7%

=> 어떻게 VGG 16-19 레이어와 같이 깊은 신경망 모델의 학습을 성공했을까?

사진참고 : CS231N CNN

02. CONVET CONFIGURATIONS

VGG-16 ARCHITECTURE 구성

모든 con layer에서 3*3 필터 사용한다!

13 Convolution Layer + 3 Fully-Connected Layer + Softmax

Input : 224*224 RGB

3*3 Convoultion Filters (좌우, 위아래, 중앙의 정보를 구별할 수 있는 가장 작은 형태)

stride : 1

padding : 1 -> 사이즈 줄지 않게!

5번의 2*2 max pooling (stride : 2)로 이미지 줄이기

RELU for nonliearlity

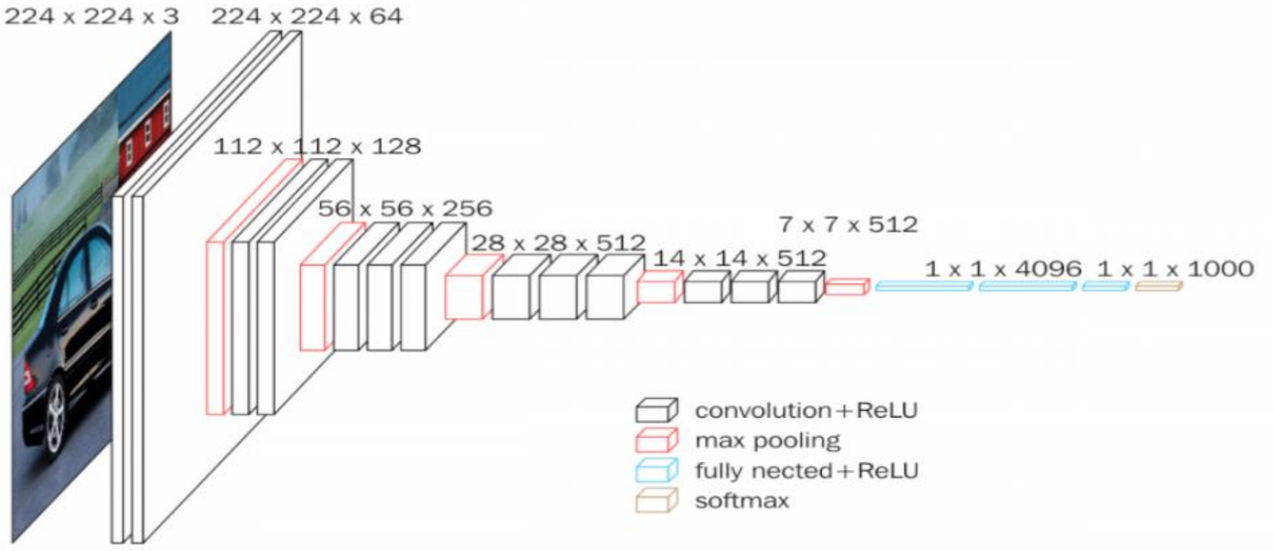


Table 1: **ConvNet configurations** (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as "conv⟨receptive field size⟩-⟨number of channels⟩". The ReLU activation function is not shown for brevity.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

사진 참고 : HTTPS://NEUROHIVE.IO/EN/POPULAR-NETWORKS/VGG16/

(not counting biases)

INPUT: [224x224x3] memory: 224*224*3=150K params: 0
CONV3-64: [224x224x64] memory: **224*224*64=3.2M** params: (3*3*3)*64 = 1,728
CONV3-64: [224x224x64] memory: **224*224*64=3.2M** params: (3*3*64)*64 = 36,864
POOL2: [112x112x64] memory: 112*112*64=800K params: 0
CONV3-128: [112x112x128] memory: 112*112*128=1.6M params: (3*3*64)*128 = 73,728
CONV3-128: [112x112x128] memory: 112*112*128=1.6M params: (3*3*128)*128 = 147,456
POOL2: [56x56x128] memory: 56*56*128=400K params: 0
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*128)*256 = 294,912
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*256)*256 = 589,824
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*256)*256 = 589,824
POOL2: [28x28x256] memory: 28*28*256=200K params: 0
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*256)*512 = 1,179,648
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*512)*512 = 2,359,296
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*512)*512 = 2,359,296
POOL2: [14x14x512] memory: 14*14*512=100K params: 0
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
POOL2: [7x7x512] memory: 7*7*512=25K params: 0
FC: [1x1x4096] memory: 4096 params: 7*7*512*4096 = **102,760,448**
FC: [1x1x4096] memory: 4096 params: 4096*4096 = 16,777,216
FC: [1x1x1000] memory: 1000 params: 4096*1000 = 4,096,000

TOTAL memory: 24M * 4 bytes ~= 96MB / image (only forward! ~*2 for bwd)
TOTAL params: 138M parameters

Note:

Most memory is in early CONV

Most params are in late FC

VGG16

Common names

03. DISCUSSION

3*3 FILTER

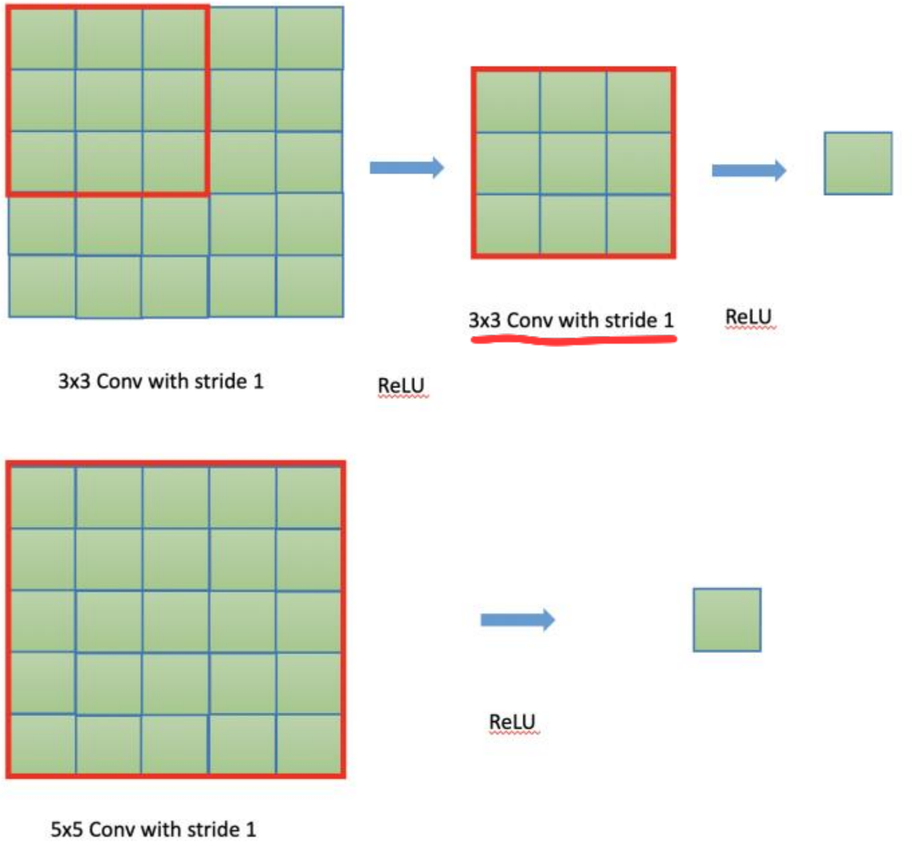
효과

3*3 Conv layer 2개 = 5*5 Conv layer 효과

3*3 Conv layer 3개 = 7*7 Conv layer 효과

5*5 이미지가 있다고 했을 때, 3*3 filter를 먼저 적용하면(stride 1) 3*3 feature map이 생성되고, 3*3을 3*3 kernel을 또 한번 통과하면 처음 이미지에서 5*5 filter를 한번 통과시킨 것과 같은 결과를 볼 수 있다.

stride가 1일 때, 3차례의 3*3 Conv 필터링을 반복한 특징맵은 한 픽셀이 원본 이미지의 7*7 Receptive field의 효과를 볼 수 있다.



3*3 FILTER

7*7 필터로 1번 VS 3*3 필터로 3번 Conv

1. 결정함수의 비선형성 증가

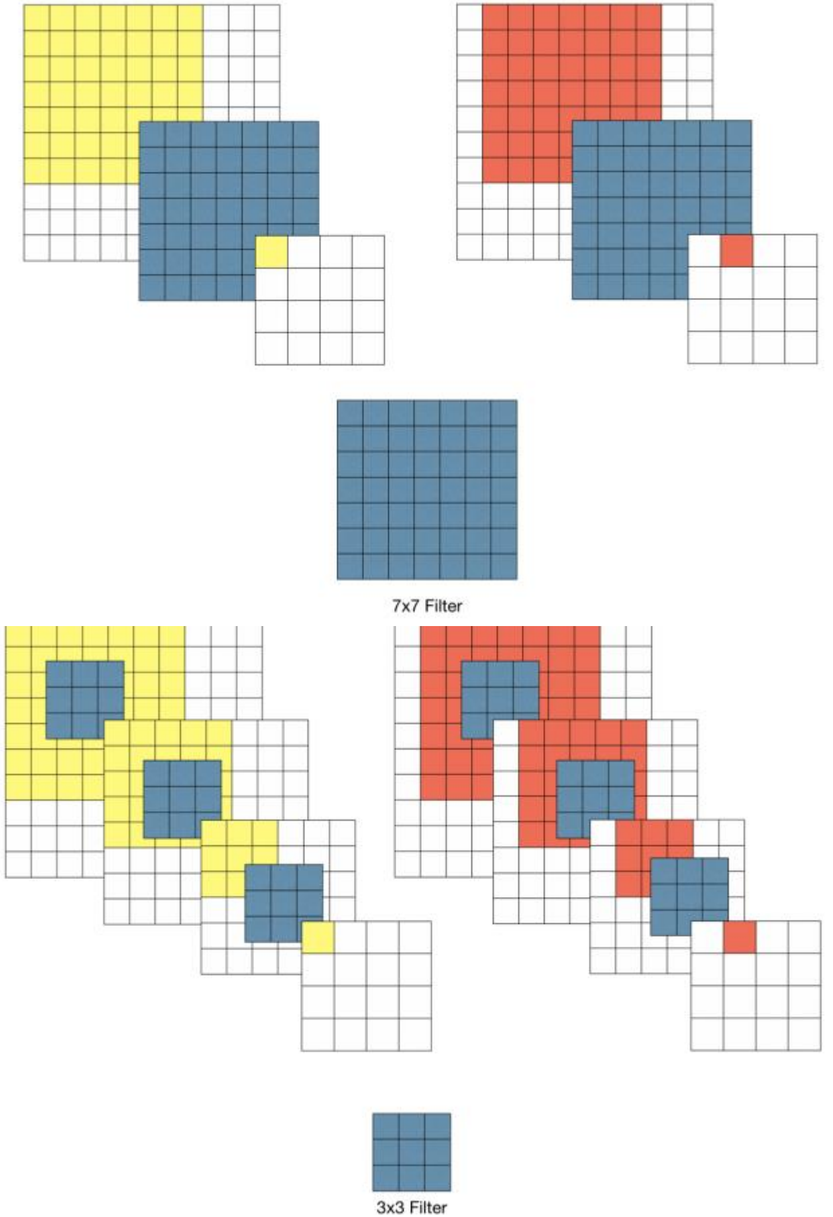
   각 Convoultion은 Relu 함수 포함한다.

   모델의 특징 식별성 증가로 이어진다.

2. 학습 파라미터 수의 감소

   7**7 필터 1개에 대한 학습 파라미터 수는 49(7*7)

   3*3 필터 3개에 대한 학습 파라미터 수는 27(3*3*3) -> 45% 절약 가능!

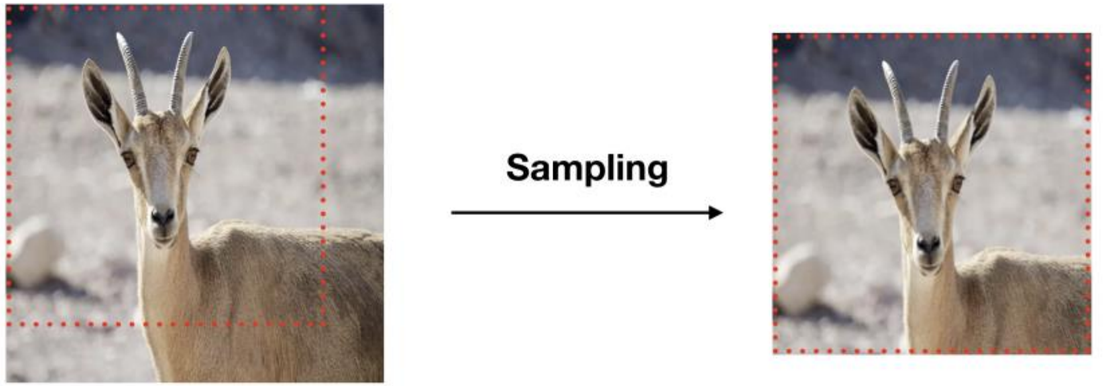


04. 알고리즘

학습 이미지 크기

Training시 입력 이미지의 크기는 모두 224*224로 고정하였다.

학습 이미지는 각 이미지에 대해 256*256 ~ 512*512 내에서 임의의 크기로 변환하고, 크기가 변환된 이미지에서 개체의 일부가 포함된 224*224 이미지를 Crop하여 사용하였다.



학습 이미지 크기

학습 데이터를 다양한 크기로 변환하고 그 중 일부분을 샘플링해 사용함으로써

1. 한정적인 데이터의 수를 늘릴 수 있다. Data augmentation
2. 하나의 오브젝트에 대한 다양한 측면을 학습 시 반영시킬 수 있다.

=> Overfitting 방지

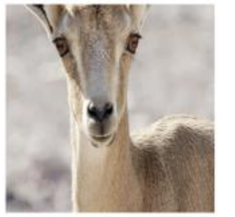
512x512

256x256

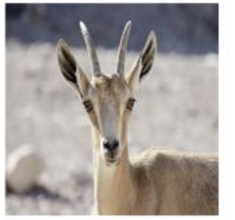
224x224

224x224

224x224



224x224



224x224

224x224

224x224

224x224

05.

결론

Table 3: **ConvNet performance at a single test scale.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	**25.5**	**8.0**

ILSVRC-2012 dataset에 대한 실험결과, 다양한 type의 VGG들의 성능은 확실히 19 layers를 가지고 있는 E type으로 갈수록 좋다.

Table 7: **Comparison with the state of the art in ILSVRC classification**. Our method is denoted as "VGG". Only the results obtained without outside training data are reported.

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	**23.7**	**6.8**	**6.8**
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-	7.9	
GoogLeNet (Szegedy et al., 2014) (7 nets)	-	**6.7**	
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

Ensemble을 사용하지 않은 단일 모델의 성능으로만 보면, VGG가 더 좋다.

THANK YOU