VGGNet

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

신훈철

1. INTRODUCTION

- Conv net이 깊어지는 것에 대하여 실험함.
- 비교적 간단한 구조로 SOTA 성능을 냄.
- 3*3 필터로 Conv. 인풋의 사이즈를 지켜줌.

2. CONVNET CONFIGURATIONS

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.

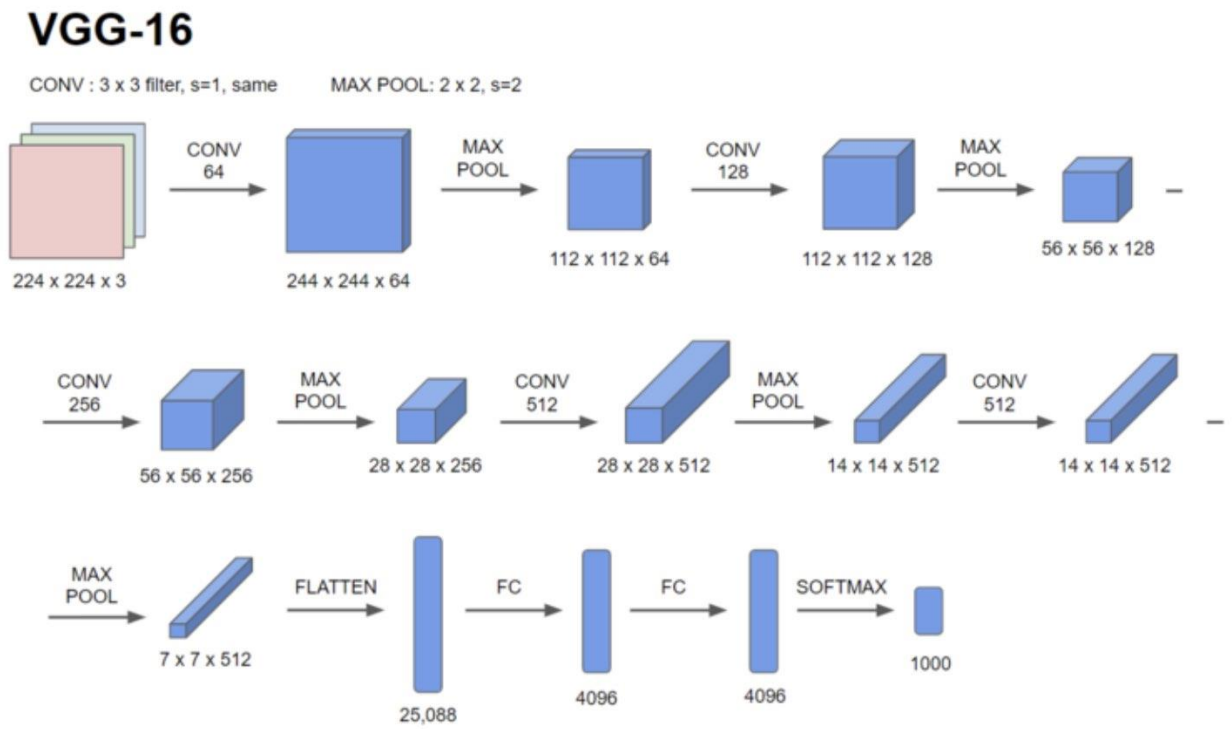
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	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					

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

- 데이터 전처리는 각 픽셀별로 RGB 평균을 빼는것으로 끝.
- LRN 효과가 없어 B부터 제외함.
- 3*3 필터로 가중치 개수 줄임.
- 3*3 필터로 같은 receptive field 에 discriminative 해짐.
- Conv1 은 조정역할, 큰 효과는 없었다.
- 깊이가 깊어질수록 성능 향상을 보임.
- 더 큰 사이즈 데이터에는 더 깊이가도 괜찮다고 추측함.
- GoogLE NET 에 비해 매우 매우 쉬운 구조]
- 맥스풀링 만으로 사이즈를 줄인다.

2. CONVNET CONFIGURATIONS



VGG-16

- 이 모델의 특징은 모든 합성곱 연산은 3 x 3 의 필터를 가지고 패딩 크기는 2, 스트라이드는 1로 하고, 2 x 2 픽셀씩 최대 풀링하는 것입니다..
- 산출값의 높이와 넓이는 매 최대 풀링 마다 1/2씩 줄어들며, 채널의 수는 두배 혹은 세배로 늘어나게 만드는 것이 VGG 모델의 체계적인 점입니다.
- 다만, 훈련시킬 변수의 개수가 많아 네트워크의 크기가 커진다는 단점이 있습니다.

3. CLASSIFICATION FRAMEWORK

Train 하이퍼 파라미터 설정

SGD(m=0.9)
L2(5e-4)
Dropout(0.5) for first 2 FC
Learning_rate(e-2)
Acc 멈추면 10씩 나눠줌. 3번
74에폭쯤에서 멈춤.
아마, A를 활용한 초기화(4, 3)와 깊이와 작은 필터의 조화로 빨리 끝낸것일듯

부가적으로..
NVIDIA Titan Black 4gpus 2-3 weeks per a single net
각 지피유가 계산한 gradient를 평균내어 합쳐준다. 이는 하나의 지피유가 일한것과 동일.
C++ Caffe 사용

3. CLASSIFICATION FRAMEWORK

Training image size.

Training scale을 'S' 로 표시.

Single-scale training 의 경우,
256/384 로 무작위로 자른 후 224 사이즈 선택
384에 256 가중치 적용

Multi-scale training의 경우,
256-512 사이즈에 대해서 인풋 데이터를 취함
384의 가중치를 가져옴.

1.3 M / 50K / 100K

Testing image size.

Testing scale을 'Q' 로 표시.

Multi-crop 방식을 활용함. 여러 스케일로부터 각 코너와 중앙, 뒤집기등을 적용하여 데이터 augmentation 150장.

Dense evaluation 적용.
큰사이즈의 이미지에 자르지 않고 일정한 픽셀간격으로 인풋을 만든다?

4. CLASSIFICATION 실험

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**

4. CLASSIFICATION 실험

Table 4: **ConvNet performance at multiple test scales.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
B	256	224,256,288	28.2	9.6
C	256	224,256,288	27.7	9.2
	384	352,384,416	27.8	9.2
	[256; 512]	256,384,512	26.3	8.2
D	256	224,256,288	26.6	8.6
	384	352,384,416	26.5	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**
E	256	224,256,288	26.9	8.7
	384	352,384,416	26.7	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**

4. CLASSIFICATION 실험

Table 5: **ConvNet evaluation techniques comparison.** In all experiments the training scale S was sampled from $[256; 512]$, and three test scales Q were considered: $\{256, 384, 512\}$.

ConvNet config. (Table 1)	Evaluation method	top-1 val. error (%)	top-5 val. error (%)
D	dense	24.8	7.5
	multi-crop	24.6	7.5
	multi-crop & dense	**24.4**	**7.2**
E	dense	24.8	7.5
	multi-crop	24.6	7.4
	multi-crop & dense	**24.4**	**7.1**

4. CLASSIFICATION 실험

Table 6: **Multiple ConvNet fusion results.**

Combined ConvNet models	Error		
	top-1 val	top-5 val	top-5 test
ILSVRC submission			
(D/256/224,256,288), (D/384/352,384,416), (D/[256;512]/256,384,512) (C/256/224,256,288), (C/384/352,384,416) (E/256/224,256,288), (E/384/352,384,416)	24.7	7.5	7.3
post-submission			
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), dense eval.	24.0	7.1	7.0
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop	23.9	7.2	-
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop & dense eval.	**23.7**	**6.8**	**6.8**

4. CLASSIFICATION 실험

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	-