GNU

Lab Seminar: 2022. 08. 02.

Very Deep Convolutional Networks For Large-Scale Image Recognition

(Simonyan et al., 15' ICLR)

IDEALAB

Improving lives through learning

JongHyeon Kim

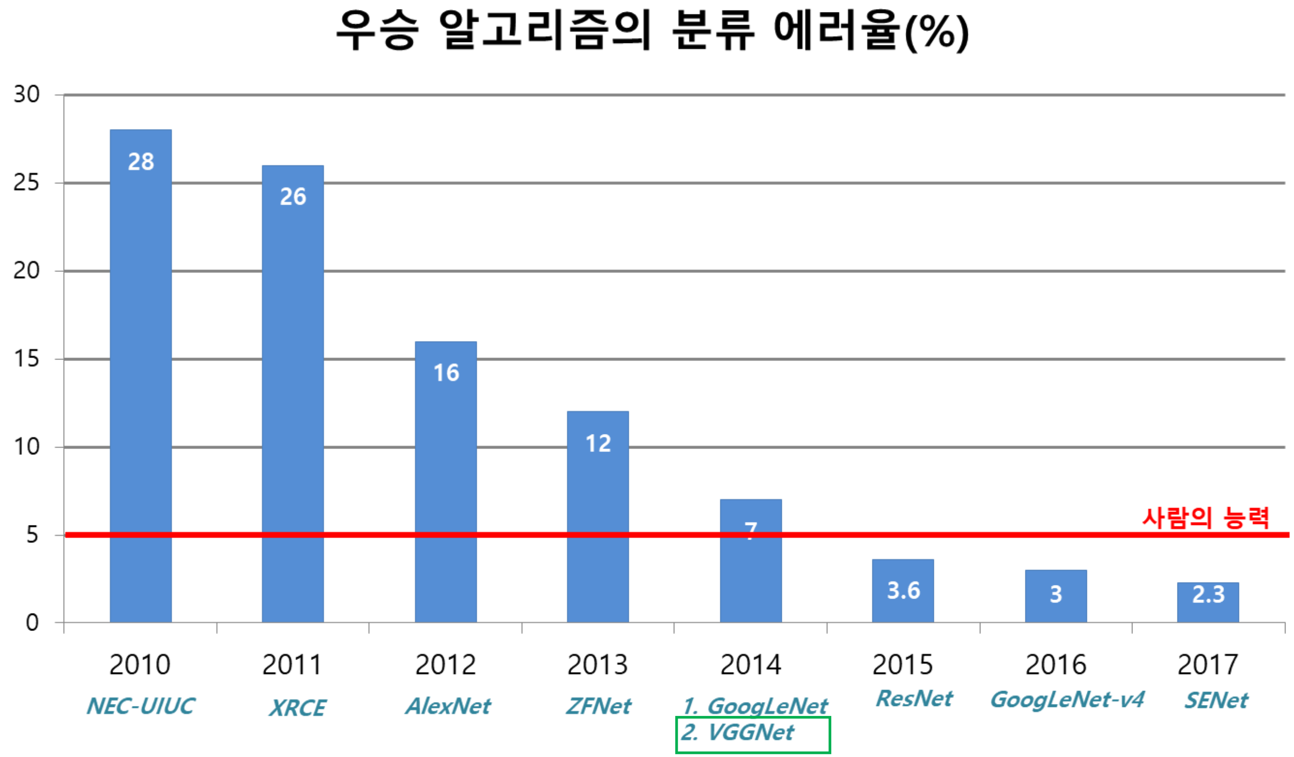
School of Computer Science/Department of AI Convergence Engineering

Gyeongsang National University (GNU)

GNU 경상국립대학교 Gyeongsang National University

Contents

Introduction

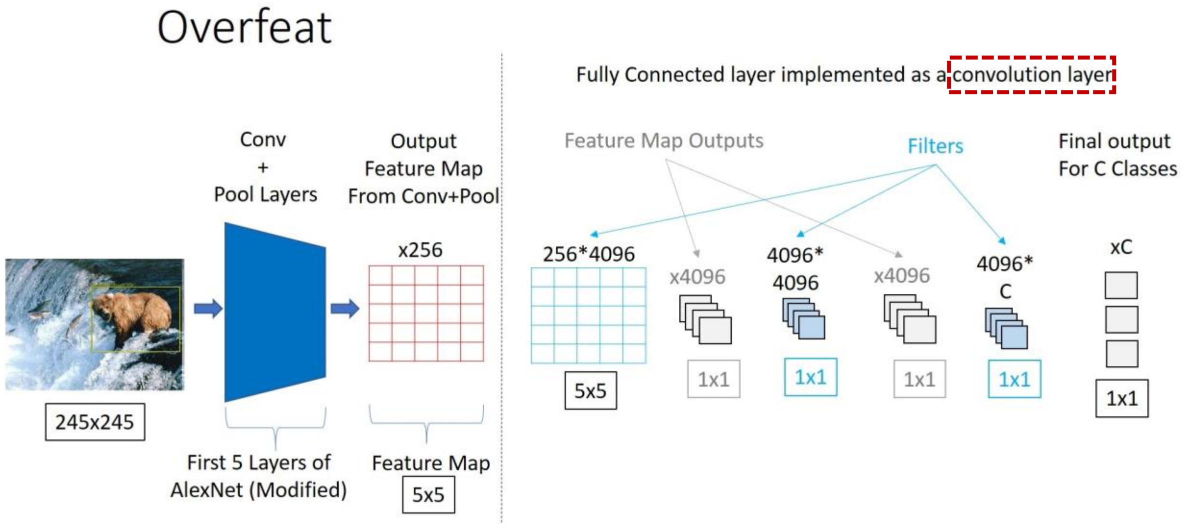


Introduction

- Increased network depth
  - Adding more convolutional layers with 3x3 conv. Filters
- Simple pipeline -> SOTA on ILSVRC 2014 (localization)
- Released two best-performing models
  - For further research

Related Work

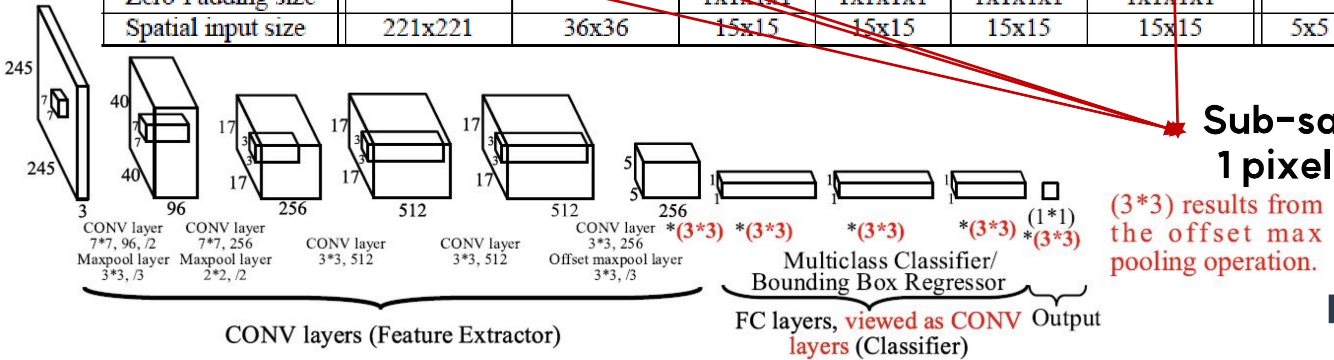
- **OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks (Sermanet et al., ICLR 2014)**
  - FC Layer -> Conv. Layer (for multi-scale input)



Related Work

- **OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks (Sermanet et al., ICLR 2014)**
  - **Architecture**

Layer	1	2	3	4	5	6	7	8	Output 9
Stage	conv + max	conv + max	conv	conv	conv	conv + max	full	full	full
# channels	96	256	512	512	1024	1024	4096	4096	1000
Filter size	7x7	7x7	3x3	3x3	3x3	3x3	-	-	-
Conv. stride	2x2	1x1	1x1	1x1	1x1	1x1	-	-	-
Pooling size	3x3	2x2	-	-	-	3x3	-	-	-
Pooling stride	3x3	2x2	-	-	-	3x3	-	-	-
Zero-Padding size	-	-	1x1x1x1	1x1x1x1	1x1x1x1	1x1x1x1	-	-	-
Spatial input size	221x221	36x36	15x15	15x15	15x15	15x15	5x5	1x1	1x1



Sub-sampling: 2x3x2x3=36
1 pixel -> encode 36 pixels

Related Work

- OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks (Sermanet et al., ICLR 2014)
  - Multi-Scale Classification

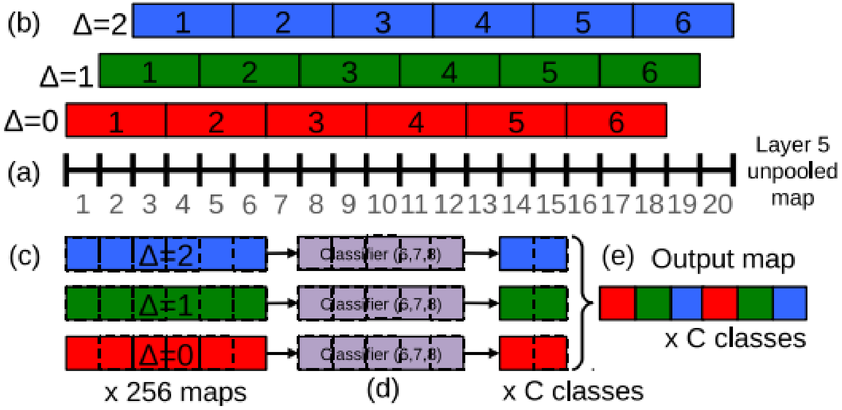


Figure 3: 1D illustration (to scale) of output map computation for classification, using y-dimension from scale 2 as an example (see Table 5). (a): 20 pixel unpooled layer 5 feature map. (b): max pooling over non-overlapping 3 pixel groups, using offsets of $\Delta = \{0, 1, 2\}$ pixels (red, green, blue respectively). (c): The resulting 6 pixel pooled maps, for different Δ. (d): 5 pixel classifier (layers 6,7) is applied in sliding window fashion to pooled maps, yielding 2 pixel by C maps for each Δ. (e): reshaped into 6 pixel by C output maps.

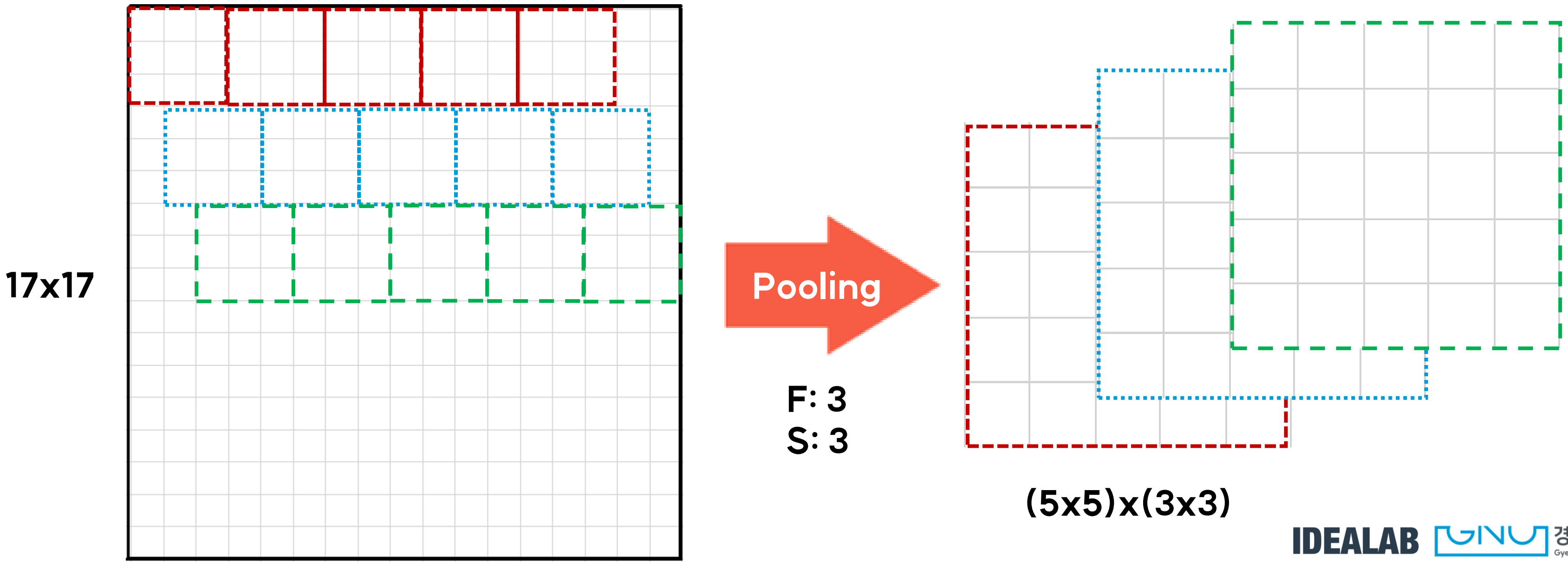
Related Work

- **OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks (Sermanet et al., ICLR 2014)**
  - **Multi-Scale Classification**

Scale	Input size	Layer 5 pre-pool	Layer 5 post-pool	Classifier map (pre-reshape)	Classifier map size
1	245x245	17x17	(5x5)x(3x3)	(1x1)x(3x3)xC	3x3xC
2	281x317	20x23	(6x7)x(3x3)	(2x3)x(3x3)xC	6x9xC
3	317x389	23x29	(7x9)x(3x3)	(3x5)x(3x3)xC	9x15xC
4	389x461	29x35	(9x11)x(3x3)	(5x7)x(3x3)xC	15x21xC
5	425x497	32x35	(10x11)x(3x3)	(6x7)x(3x3)xC	18x24xC
6	461x569	35x44	(11x14)x(3x3)	(7x10)x(3x3)xC	21x30xC

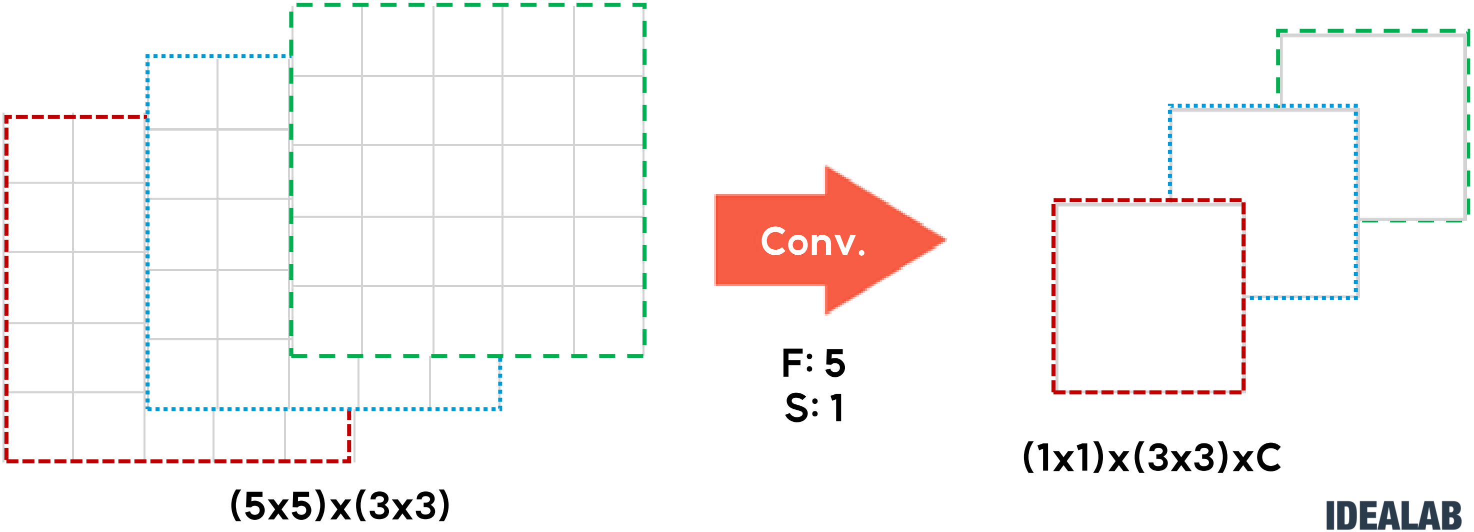
Related Work

- **OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks (Sermanet et al., ICLR 2014)**
  - **Multi-Scale Classification (6th layer)**



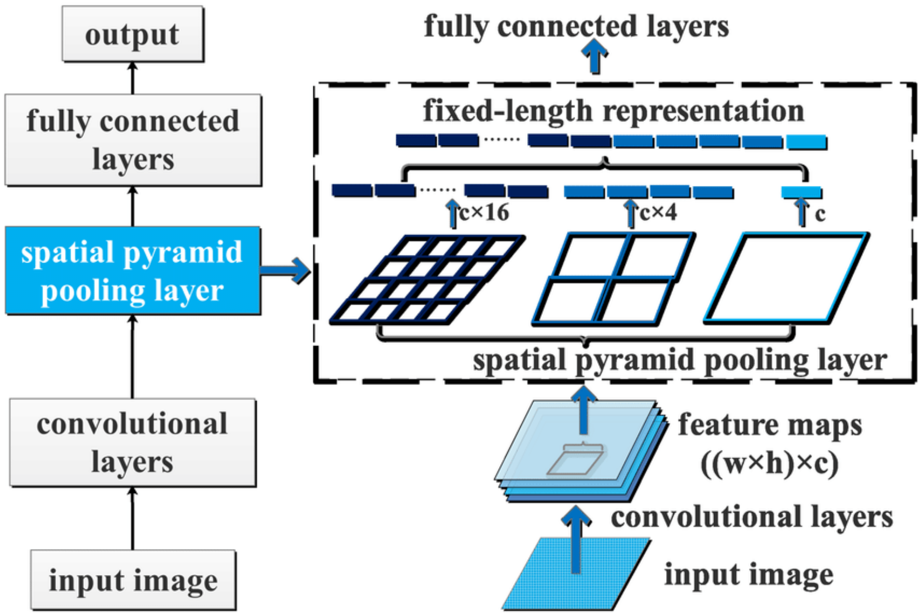
Related Work

- OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks (Sermanet et al., ICLR 2014)
  - Multi-Scale Classification (7th layer)



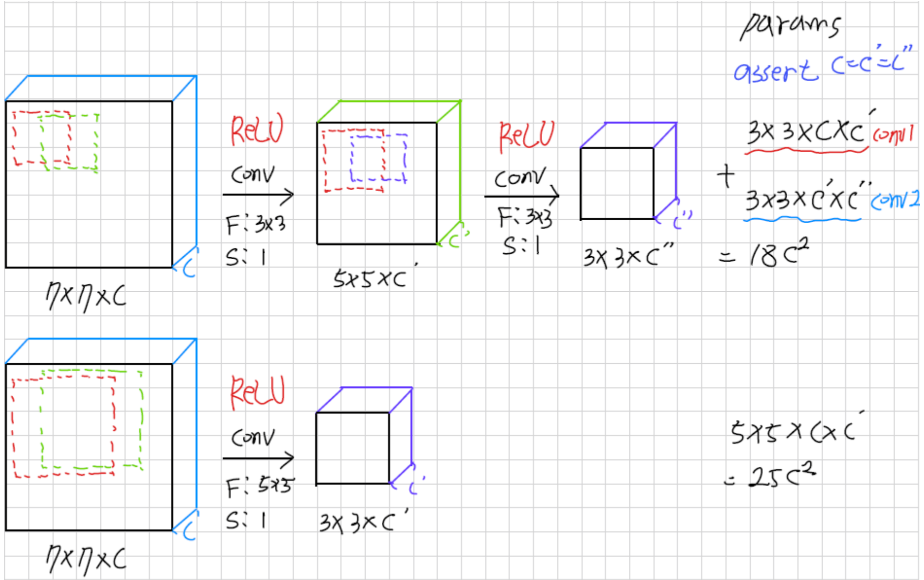
Related Work

- **Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition (Kaiming et al., CoRR 2014; SPPNet)**
  - **Spatial Pyramid Pooling (SPP)**
    - Spatial bins
      - 50 bin = [6x6, 3x3, 2x2, 1x1]
      - 30 bin = [4x4, 3x3, 2x2, 1x1]
      - 21 bin = [4x4, 2x2, 1x1]
  - **Output dimension (For FC)**
    - k*M (k: conv5 num of filters, M: bins)
      - e.g.) 21 bin: 256 feature map * 21



Main Ideas

- 3 x 3 convolution
  - Stack 3 x 3 convolutional layers -> more non-linearities (ReLUs)
  - Parameter reduction



Main Ideas

- 1 x 1 convolution
  - From *Network in Network* (Lin et al.; 2014)
  - Not for parameter decrease (GoogLeNet, NiN); for non-linearity
    - projection onto same space (same dimension)

Architecture

- Preprocessing: Mean subtraction (AlexNet, GoogLeNet)
- Small Filters (3 x 3, 1 x 1)
  - 3 x 3: smallest size to capture feature of left/right, up/down, center
  - 1 x 1: non-linearity (*Network in Network*)
- Spatial Pooling
  - Performed max-pooling some of the conv. Layers (F: 2, S: 2)
- ReLU -> after all of the conv. layers: For non-linearity

Architecture

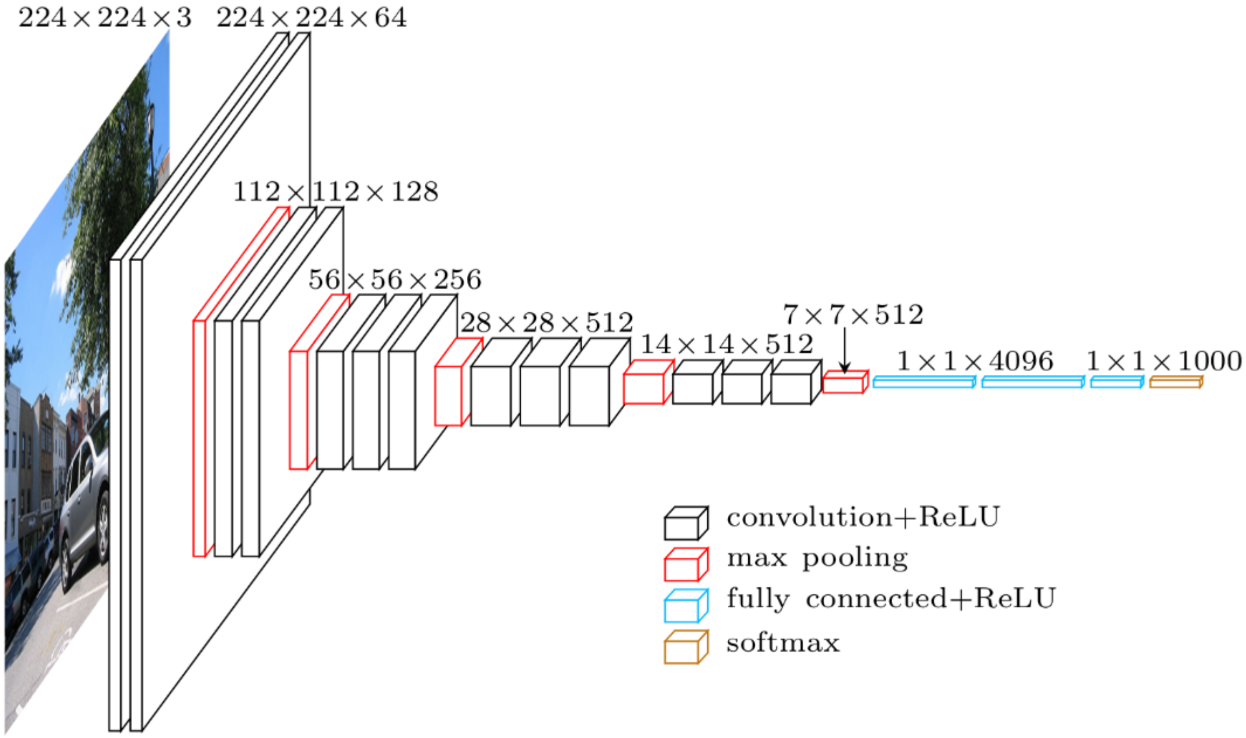
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128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

Architecture

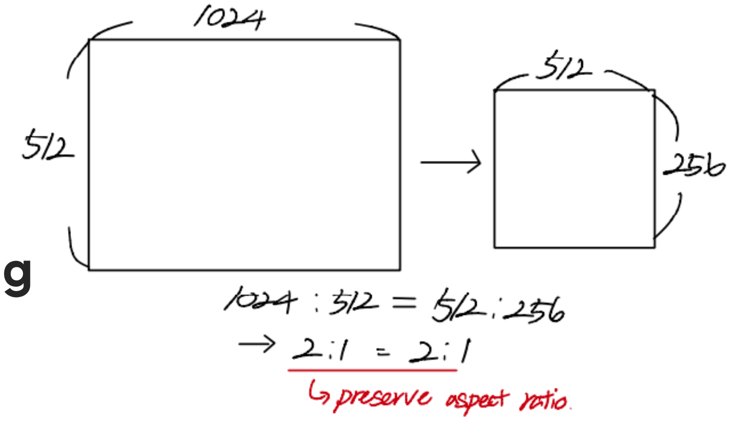
- **VGG-16 (D)**



D
16 weight layers
conv3-64 conv3-64
conv3-128 conv3-128
conv3-256 conv3-256 **conv3-256**
conv3-512 conv3-512 **conv3-512**
conv3-512 conv3-512 **conv3-512**

Framework (Train)

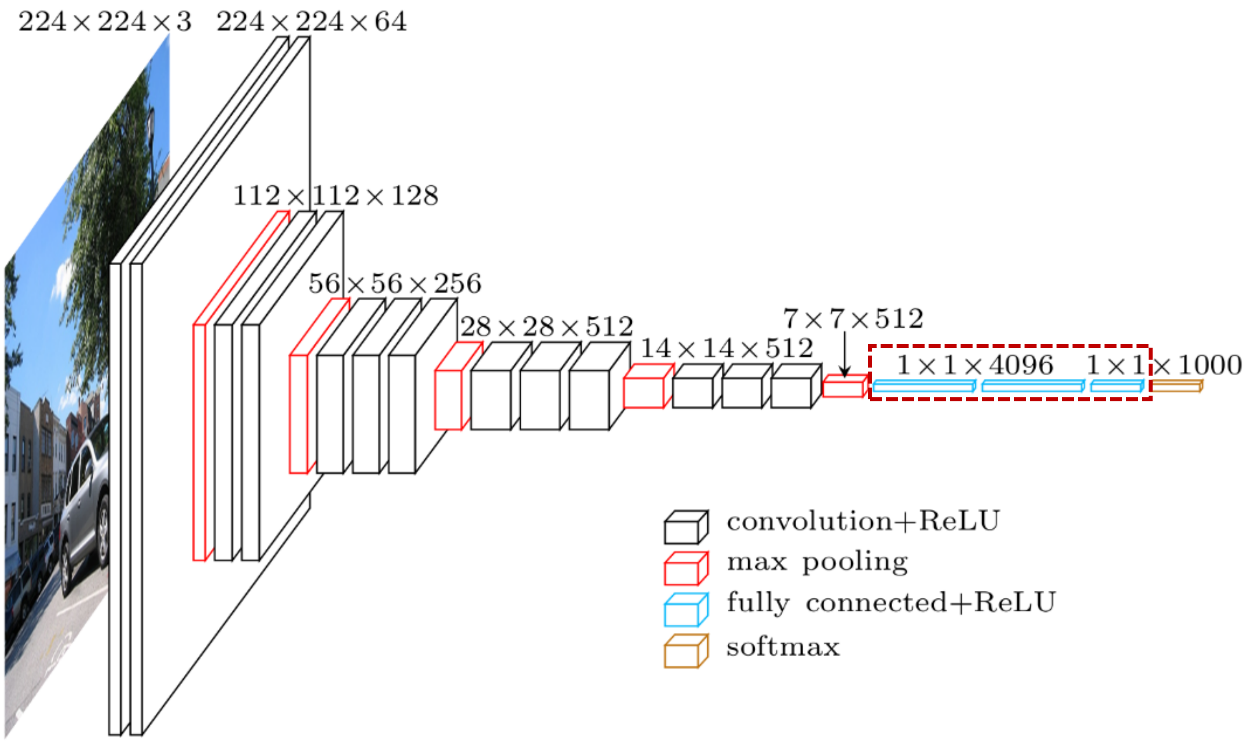
- Image Rescaling
  - Scale Jittering (Isotropically-rescaled)
    - Rescale to S (shorter side of origin image)
  - RandomCrop(224x224) after scale jittering



- Scaling
  - Single-scale training
    - Fix S to 256 or 384
  - Multi-scale training
    - Random S: [256; 512] -> various scale image

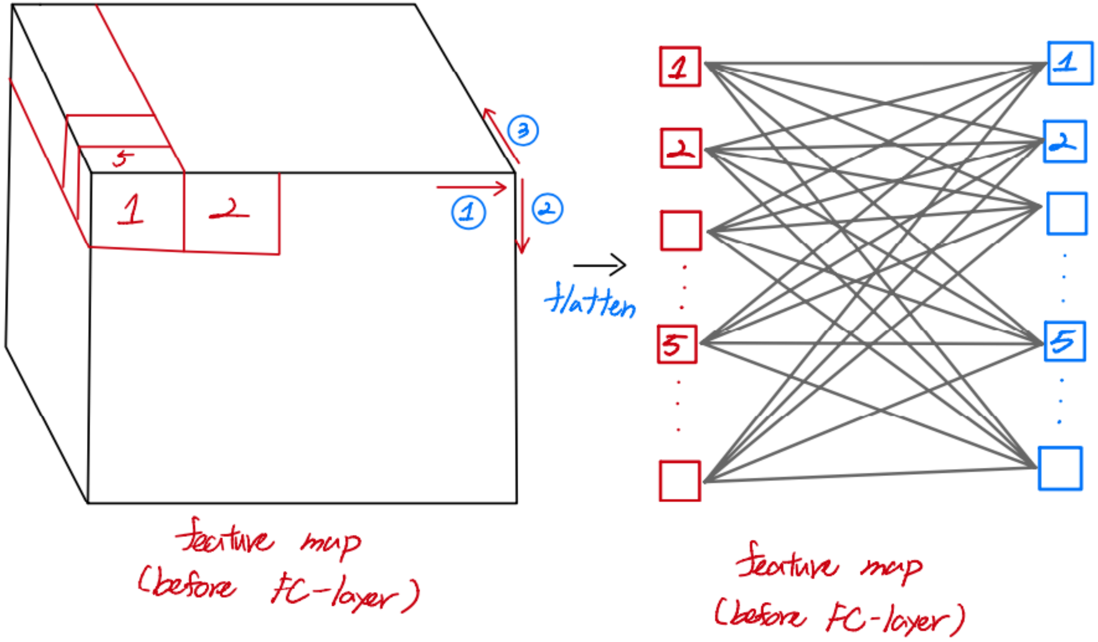
Framework (Test)

- FC layer -> Conv. layer



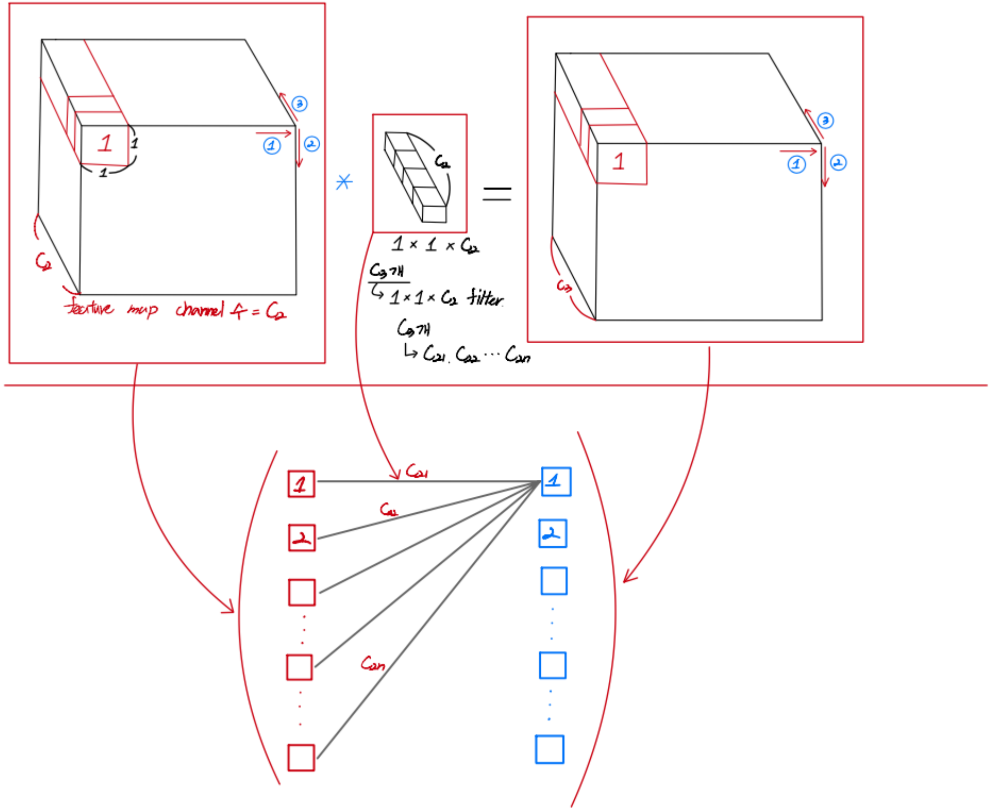
Framework (Test)

- FC layer -> Conv. layer



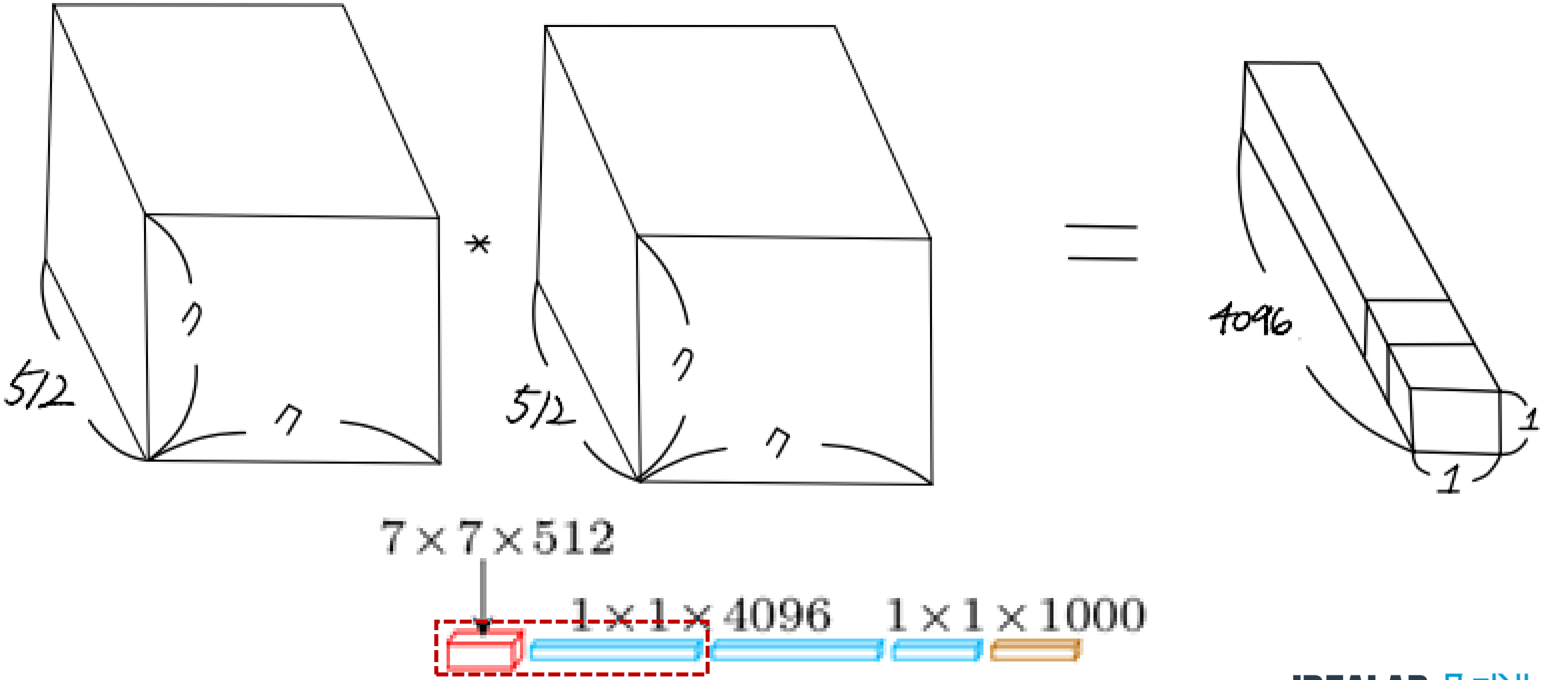
Framework (Test)

- FC layer -> Conv. layer



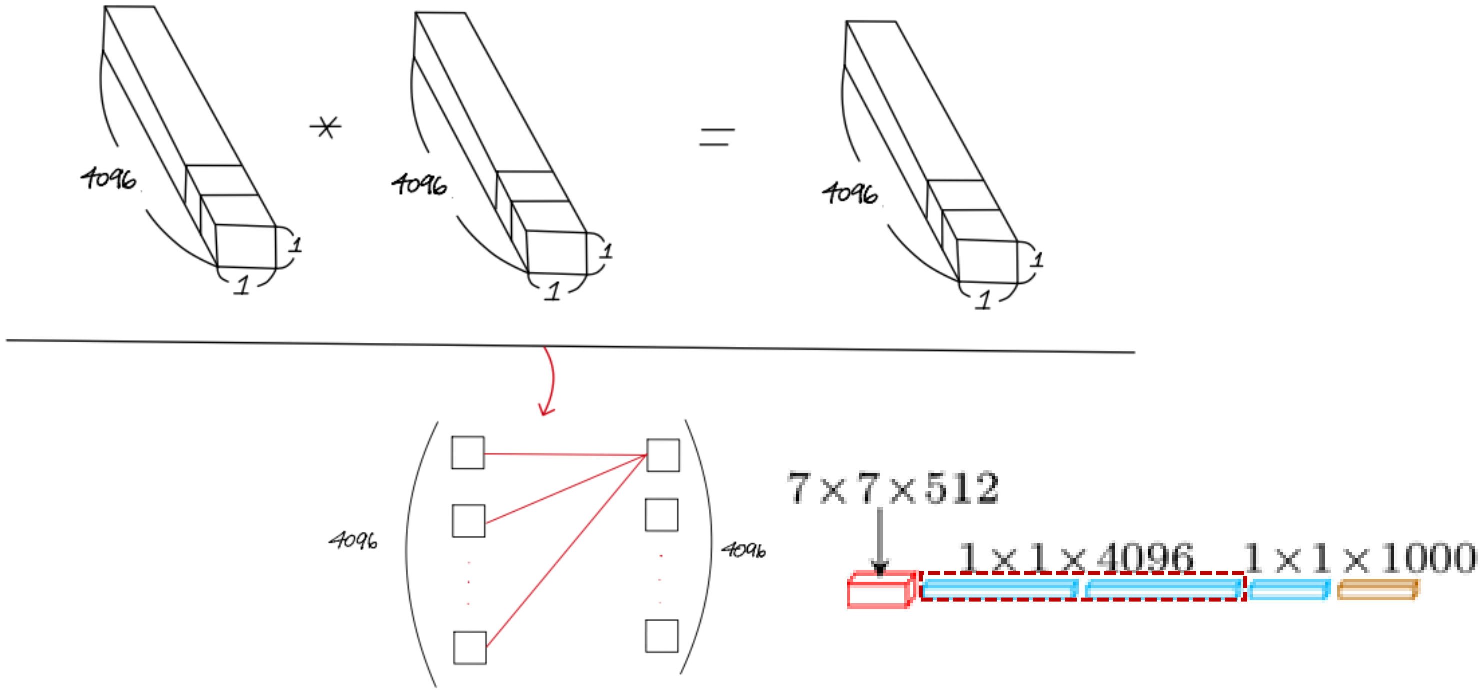
Framework (Test)

- FC layer -> Conv. layer



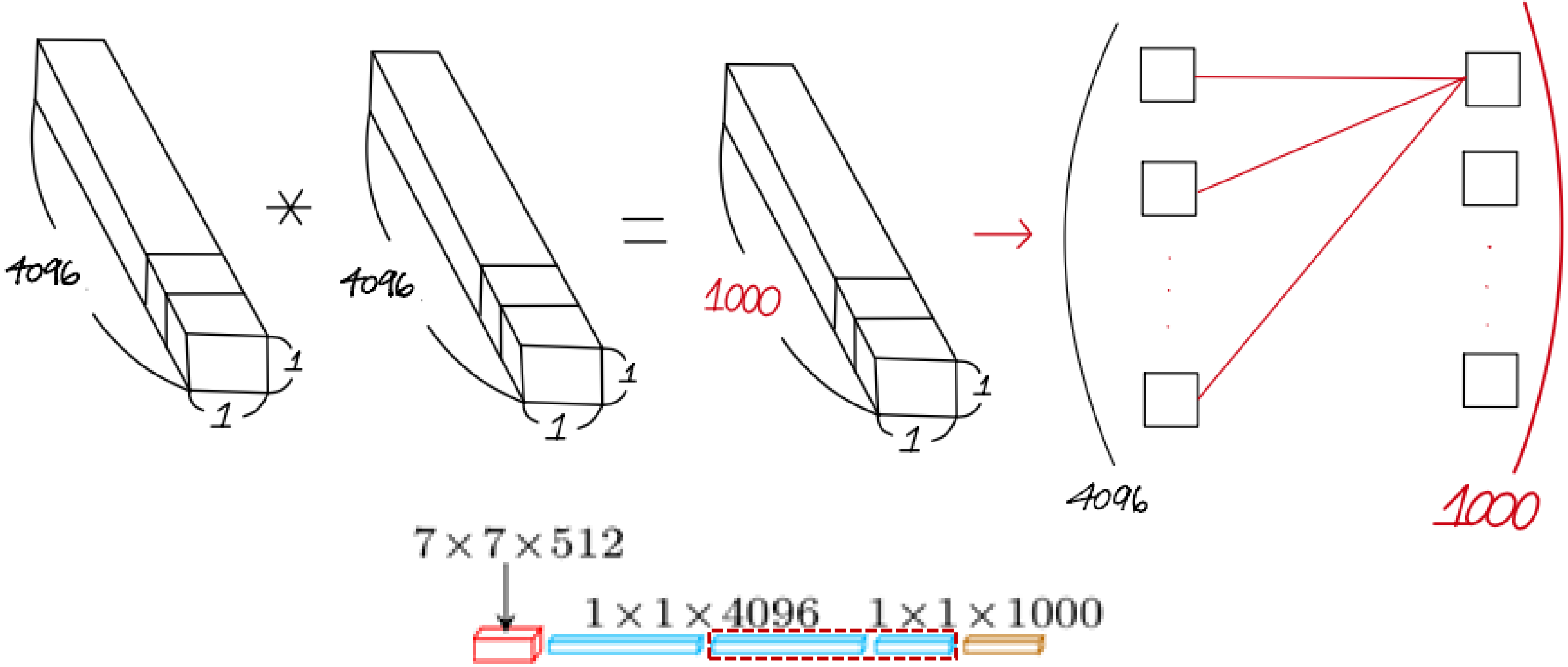
Framework (Test)

- FC layer -> Conv. layer



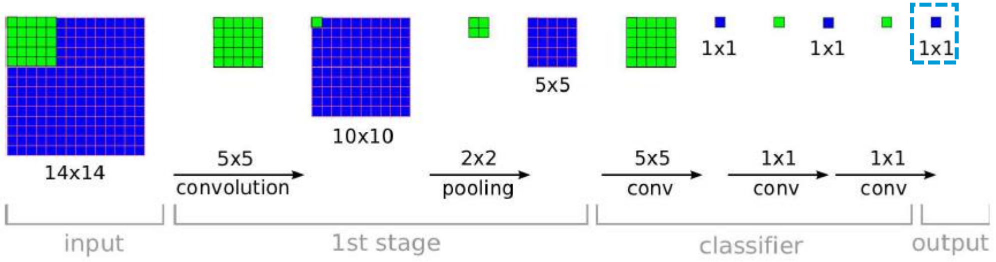
Framework (Test)

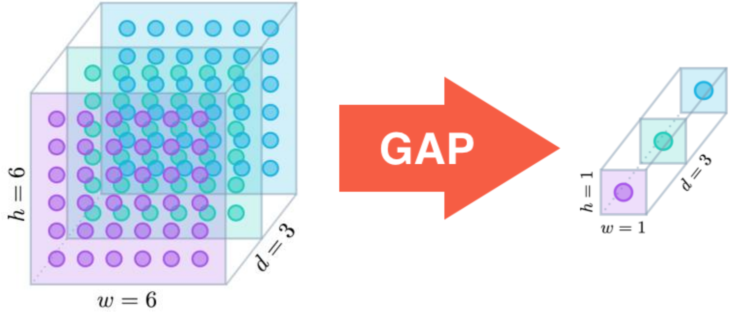
- FC layer -> Conv. layer



Framework (Test)

- FC layer -> Conv. layer





Training Details

- Optimizer: SGD (momentum 0.9, weight decay: 5e-4; AlexNet)
- First two fc layer(4960 dim): dropout 0.5
- Lr: initial 1e-2, divided by 10 when validation acc stop
- Trained for 370K iters (74 epochs)
- RGB mean subtraction = Normalization (AlexNet, GoogLeNet)

Training Details

- Regularization
  - Implicit: 3x3 conv(parameter reduction)
  - Explicit: dropout

- Initialization
  - First 4 Conv. Layers, Last 3 FC Layers (VGG-A; 11 Layers)
    - Gaussian Distribution (mean: 0, std: 1e-2) -> weight, bias = 0
  - Xavier Initialization(*Glorot & Bengio, 2010*): does not need to pretrain
    - After paper submission

Experiments

- **Single Scale Evaluation**
  - **Test image size: 0.5* (S_{min}, S_{max}) -> 0.5 * (256 + 512) = 384**

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 **1x1 conv. 16 layers**	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D **3x3 conv. 16 layers**	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**

Experiments

- **Multi-Scale (Dense) Evaluation**
  - **Test image size**
    - Q = {S − 32, S, S + 32} -> fixed S
    - Q = $\{S_{min}, 0.5(S_{min} + S_{max}), S_{max}\}$ -> random S ($[S_{min}; S_{max}]$)
  - **Averaging the result class posteriors (due to different *Q*)**

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**

Experiments

- **Multi Crop Evaluation (AlexNet, GoogLeNet)**
  - **Test image size**
    - Q = {256, 384, 512} -> S = [256; 512]
    - Q = {S_{min} , $0.5(S_{min} + S_{max})$, S_{max}} -> random S ([S_{min}; S_{max}])
  - **Averaging dense evaluation + multi crop evaluation (softmax prob.)**

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**

Results

- **Ensemble (Multiple ConvNet fusion)**

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

Combined ConvNet models	Error		
	top-1 val	top-5 val	top-5 test
ILSVRC submission **(7 models)**			
(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 **(2 models)**			
(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**

Results

- **Comparison with the SOTA in ILSVRC classification**

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	-

Conclusions & Discussions

- Conclusions
  - Deep conv. Networks -> up to 19 layers
  - Representation depth -> beneficial for the classification acc.

- Discussions
  - Large computation (parms; VGG-19: 144M, GoogLeNet: 11M)
  - Gradient Vanishing -> Deep conv. networks

GNU

경상국립대학교

Gyeongsang National University

Improving lives through learning

IDEALAB