여러가지 CNN 구조

초기모델에서 전이학습까지

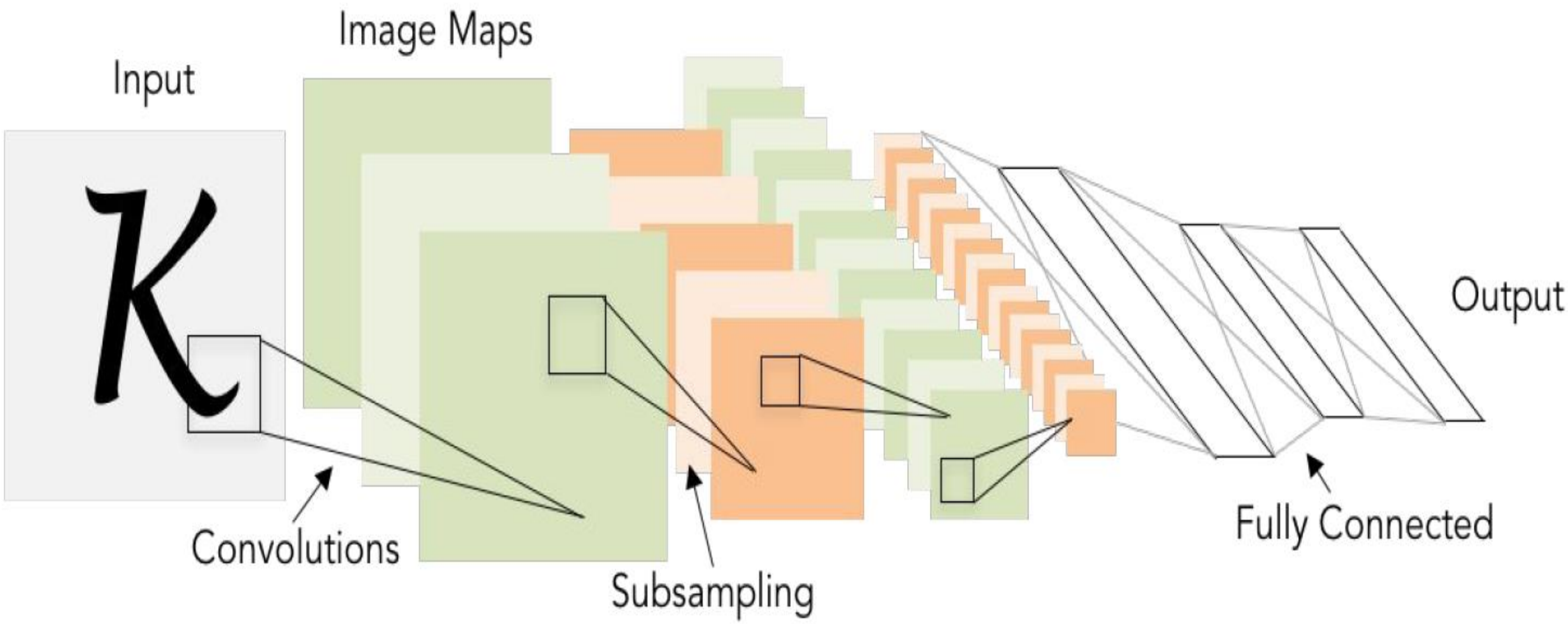
Dr. Rhee

May 2019

LeNet-5: 원조 CNN

LeNet-5

[LeCun et al., 1998]



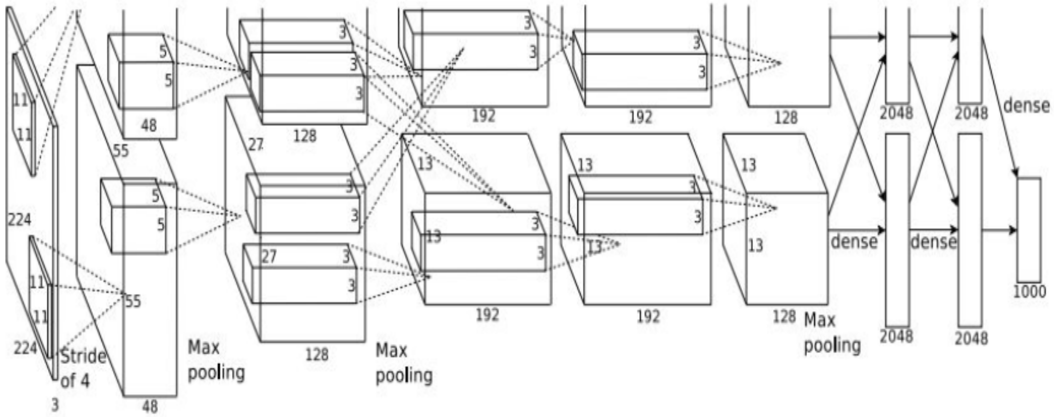
Conv filters were 5x5, applied at stride 1
Subsampling (Pooling) layers were 2x2 applied at stride 2
i.e. architecture is [CONV-POOL-CONV-POOL-FC-FC]

AlexNet

AlexNet

Case Study: AlexNet

[Krizhevsky et al. 2012]



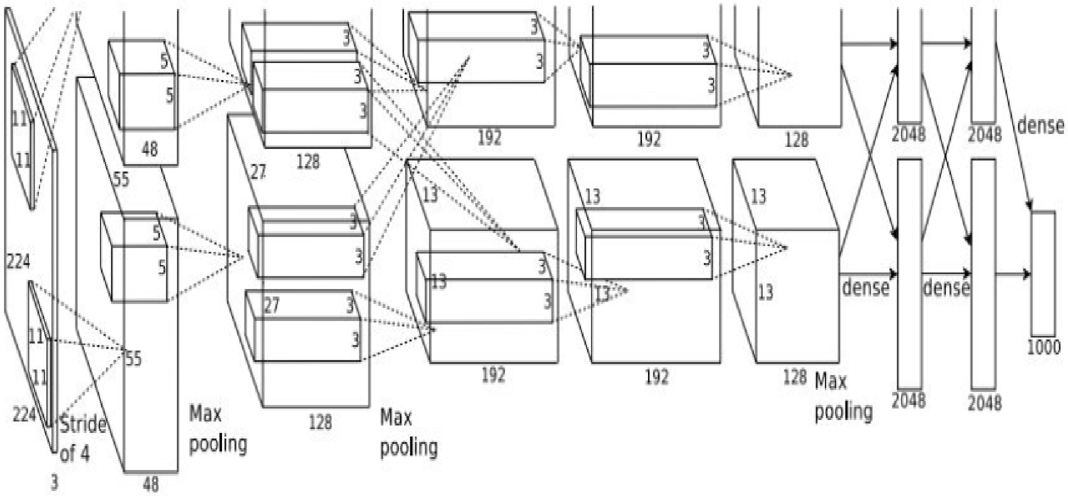
Architecture:
CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

AlexNet

Case Study: AlexNet

[Krizhevsky et al. 2012]



Input: 227x227x3 images

First layer (CONV1): 96 11x11 filters applied at stride 4

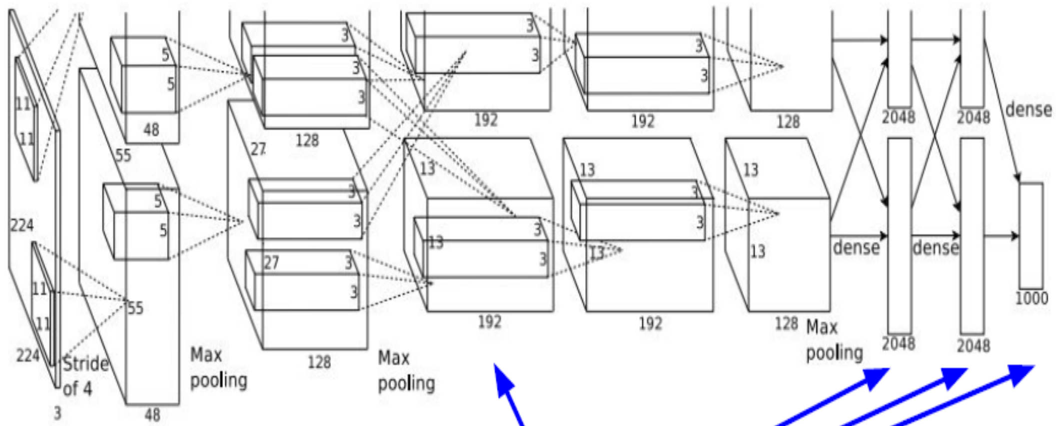
=>

Q: what is the output volume size? Hint: (227-11)/4+1 = 55

AlexNet

Case Study: AlexNet

[Krizhevsky et al. 2012]

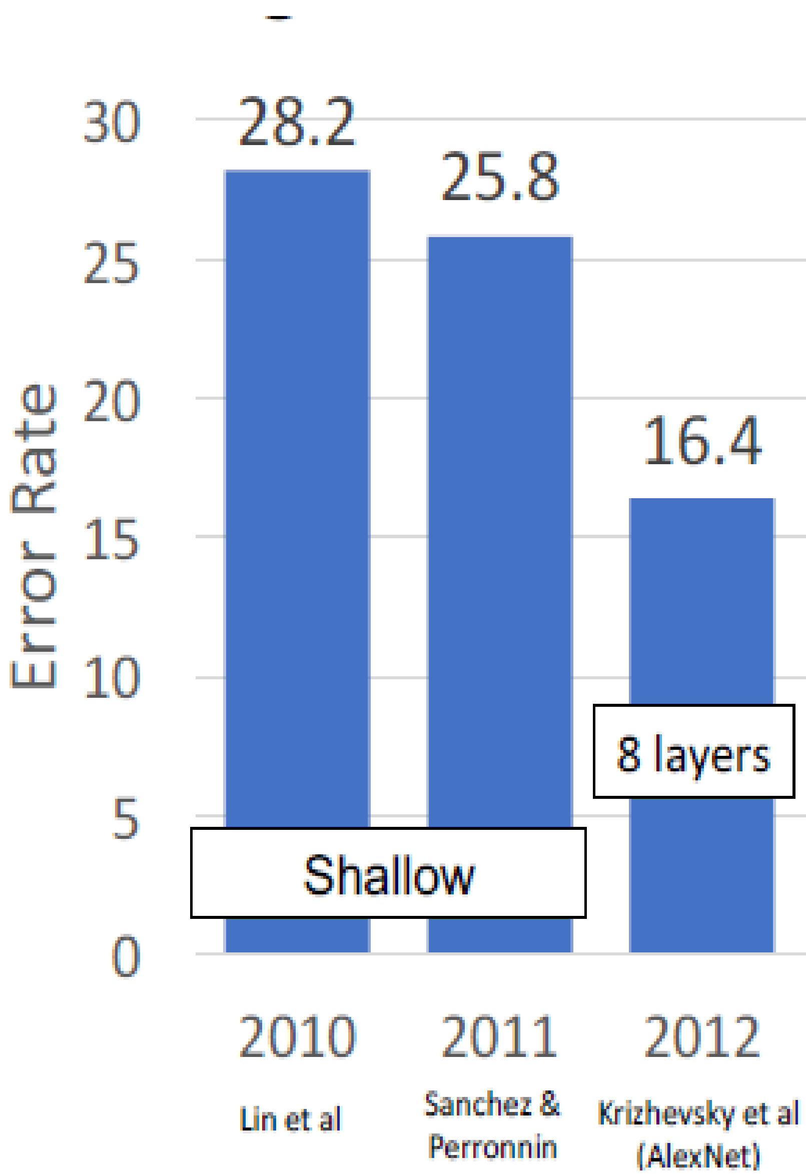


CONV3, FC6, FC7, FC8:
Connections with all feature maps in preceding layer, communication across GPUs

Full (simplified) AlexNet architecture:
[227x227x3] INPUT
[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization layer
[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization layer
[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)

Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

이미지넷 분류 경연대회

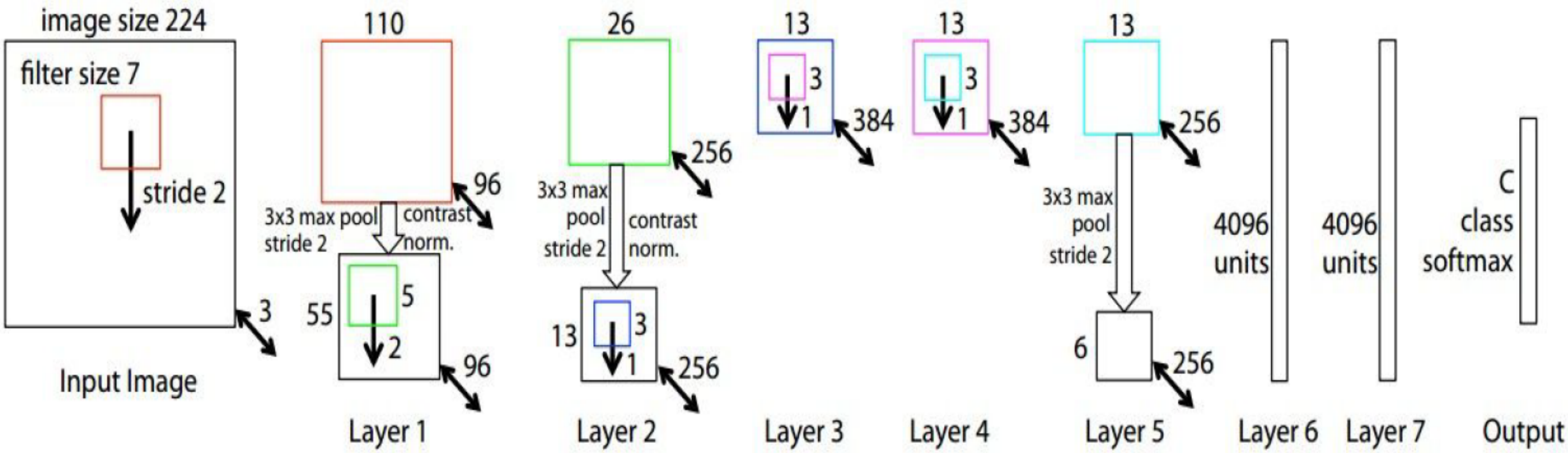


Error Rate
30
25
20
15
10
5
0
28.2
25.8
16.4
8 layers
Shallow
2010
2011
2012
Lin et al
Sanchez & Perronnin
Krizhevsky et al (AlexNet)

AlexNet

ZFNet

[Zeiler and Fergus, 2013]

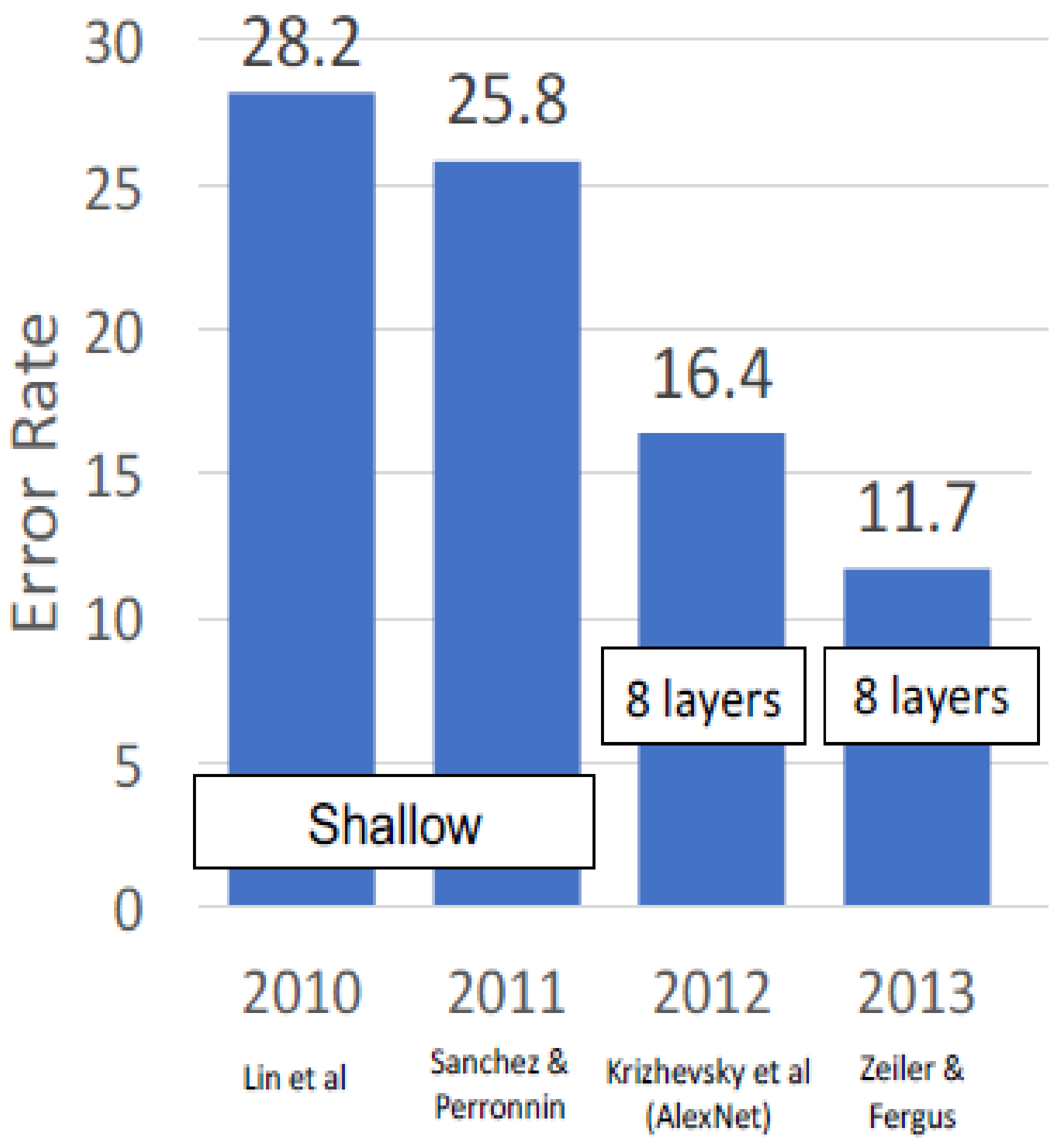


TODO: remake figure

AlexNet but:
CONV1: change from (11x11 stride 4) to (7x7 stride 2)
CONV3,4,5: instead of 384, 384, 256 filters use 512, 1024, 512

ImageNet top 5 error: 16.4% -> 11.7%

이미지넷 분류 경연대회


Error Rate
30
25
20
15
10
5
0
28.2
25.8
16.4
11.7
Shallow
8 layers
8 layers
2010
2011
2012
2013
Lin et al
Sanchez & Perronnin
Krizhevsky et al (AlexNet)
Zeiler & Fergus

VGGNet

VGGNet

Case Study: VGGNet

[Simonyan and Zisserman, 2014]

Small filters, Deeper networks

8 layers (AlexNet)
-> 16 - 19 layers (VGG16Net)

Only 3x3 CONV stride 1, pad 1
and 2x2 MAX POOL stride 2

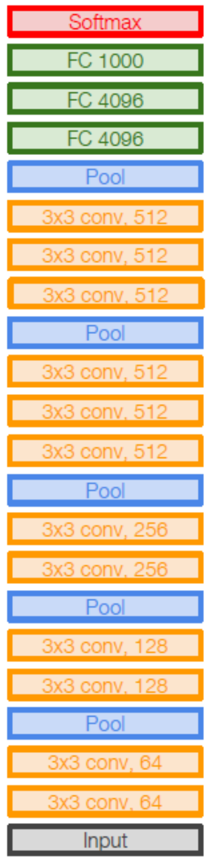
11.7% top 5 error in ILSVRC'13
(ZFNet)
-> 7.3% top 5 error in ILSVRC'14

VGGNet

(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

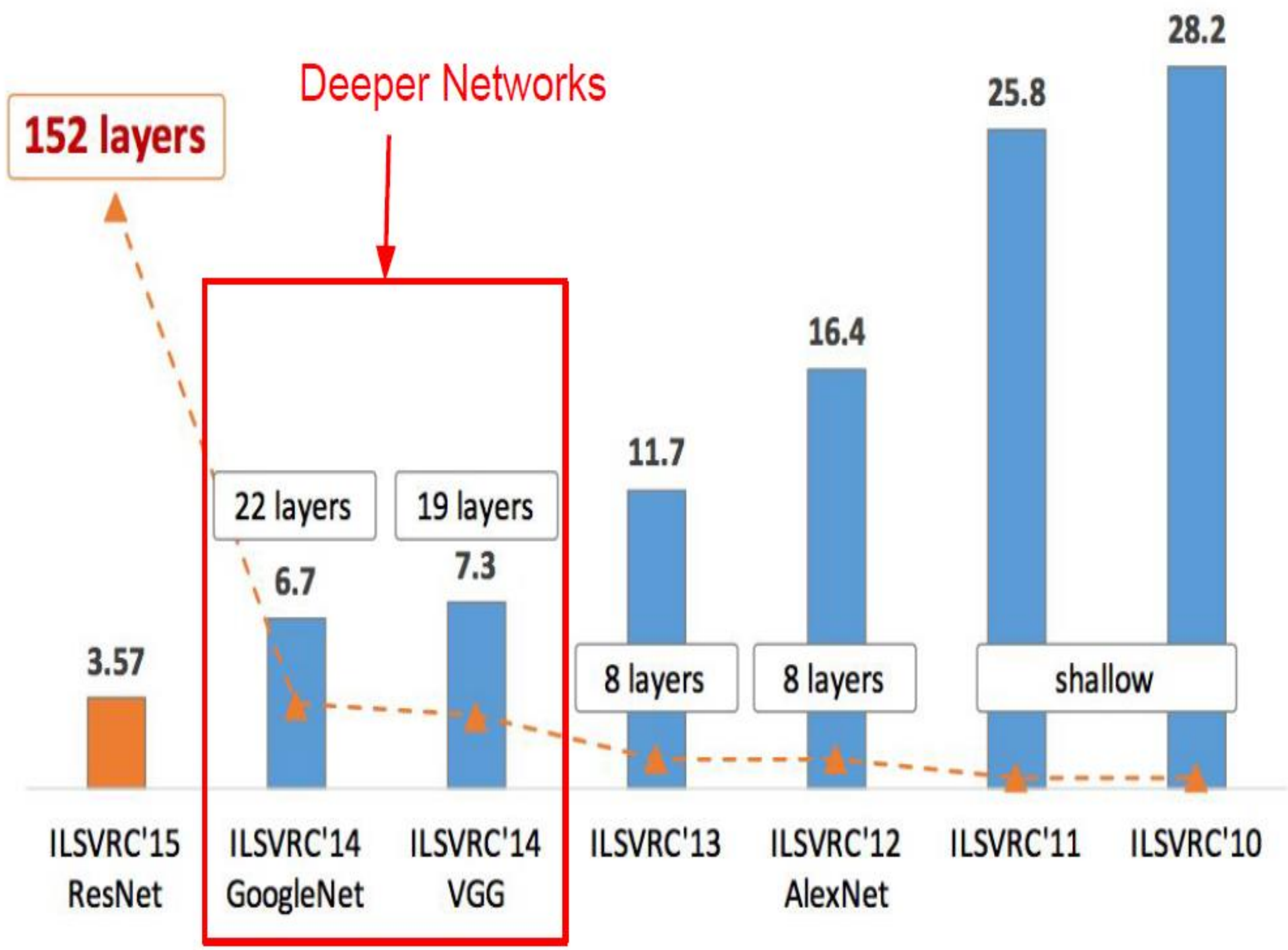
TOTAL memory: 24M * 4 bytes ~= 96MB / image (only forward! ~*2 for bwd)
TOTAL params: 138M parameters



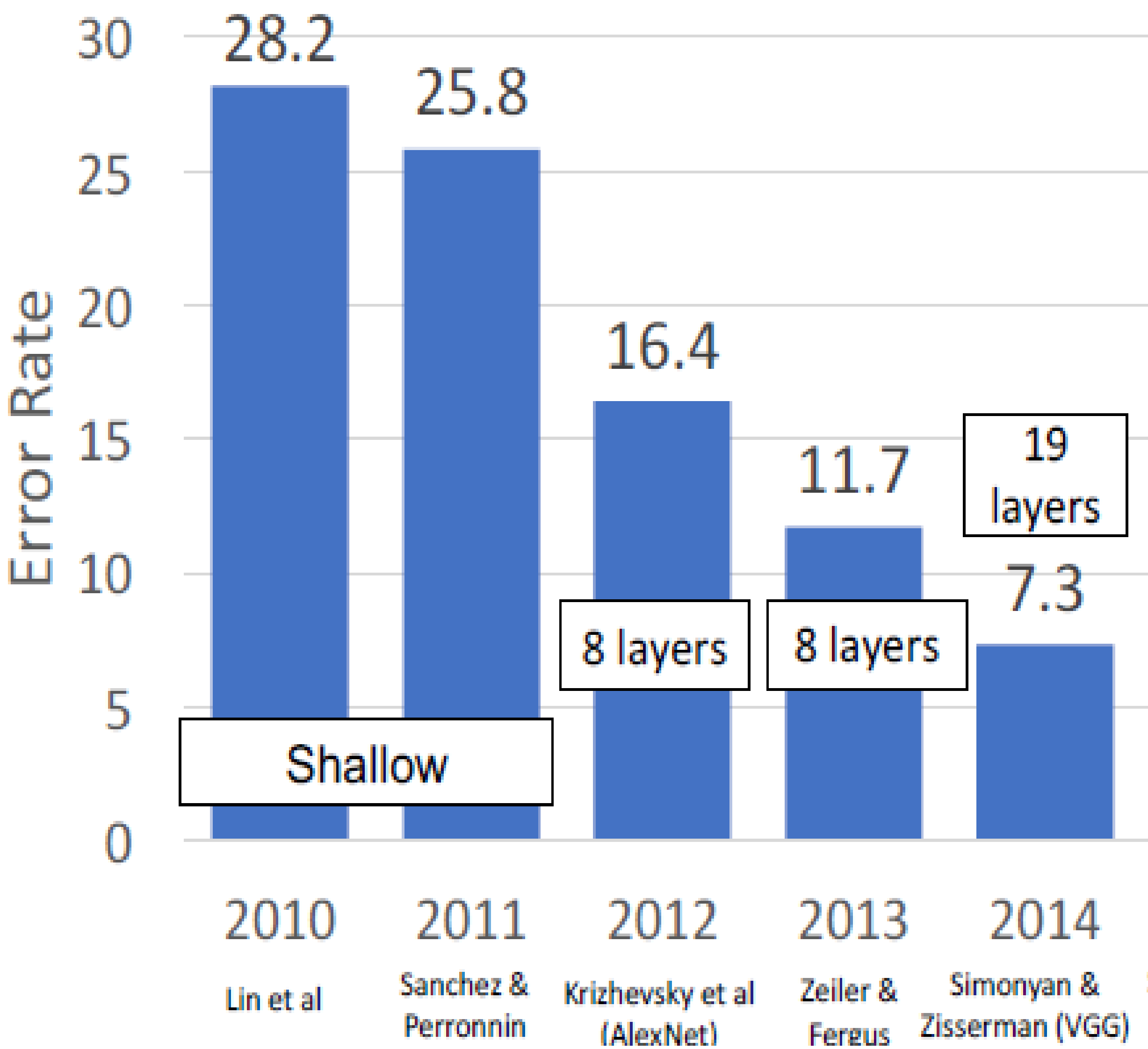
VGG16

VGGNet

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



이미지넷 분류 경연대회



Error Rate
30
25
20
15
10
5
0
28.2
25.8
16.4
11.7
7.3
Shallow
8 layers
8 layers
19 layers
2010
2011
2012
2013
2014
Lin et al
Sanchez & Perronnin
Krizhevsky et al (AlexNet)
Zeiler & Fergus
Simonyan & Zisserman (VGG)

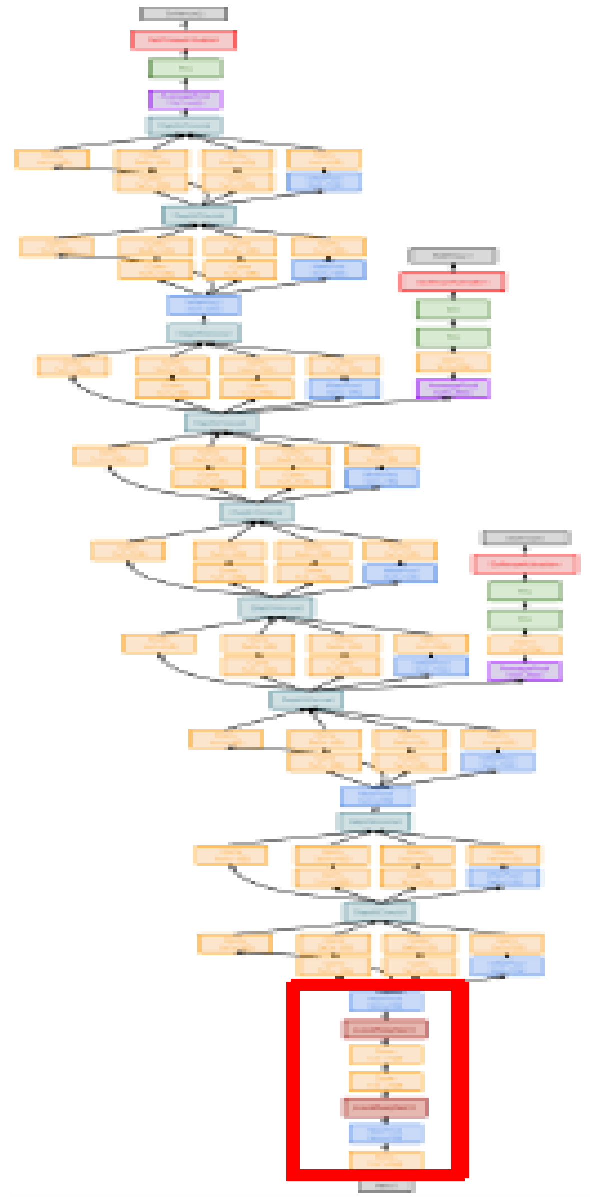
GoogleNet

GoogleNet

- 적극적 스템: 신경망의 시작부분에서 입력을 크게 다운샘플링

- VGG16의 경우 많은 계산이 시작부분에서 일어난다.

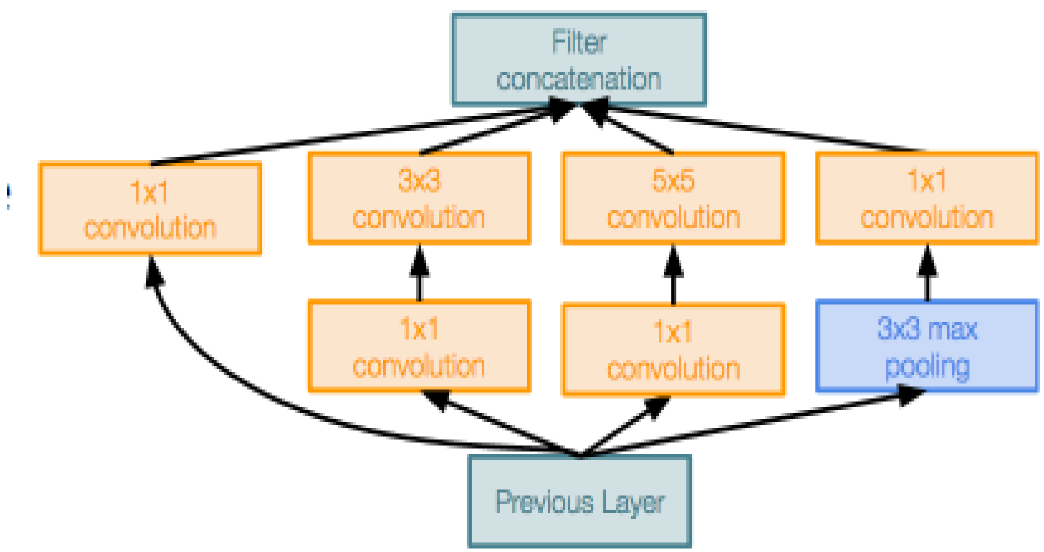
- VGG16과 비교했을 때 17.8배 연산을 감소

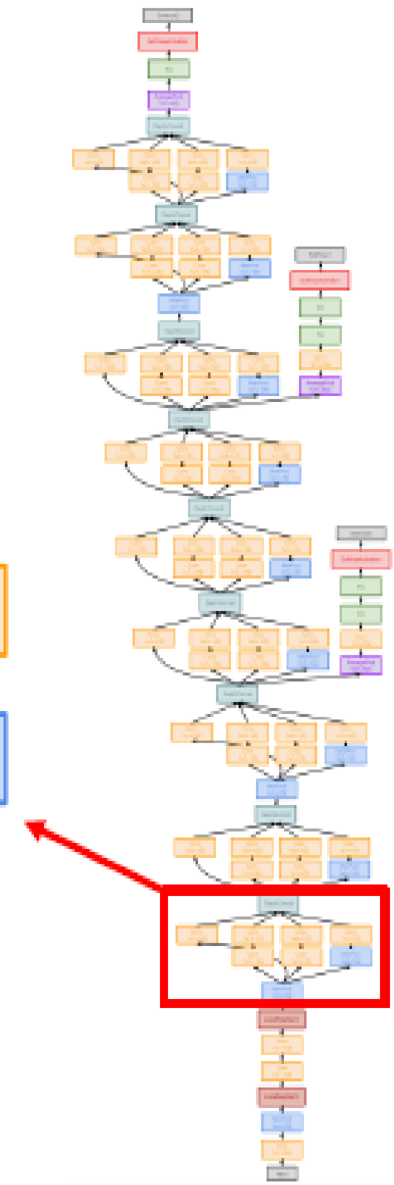


GoogleNet

- 인셉션 모듈(Inception module): 병렬 가지를 가진 로컬 유닛

\- 로컬 유닛이 신경망에서 여러번 반복된다.

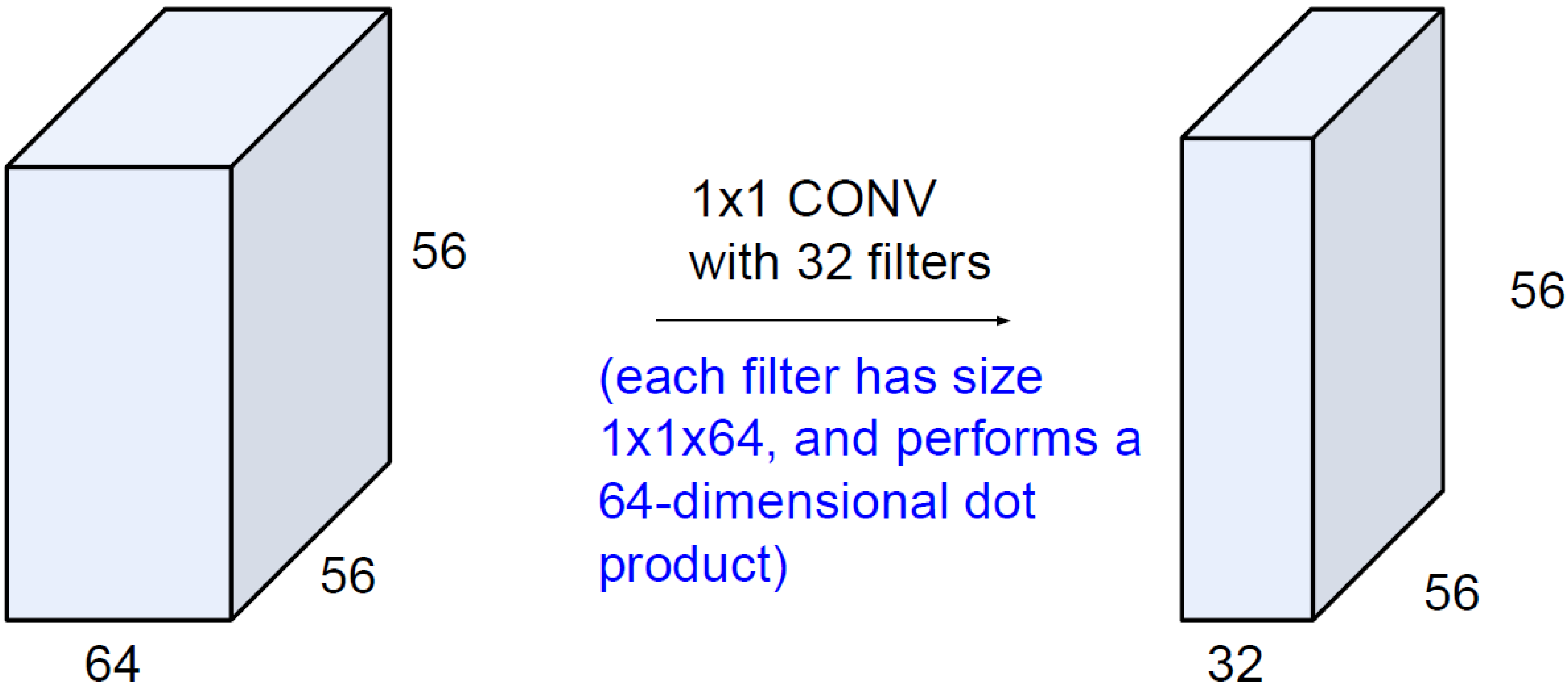




CVPR 2015

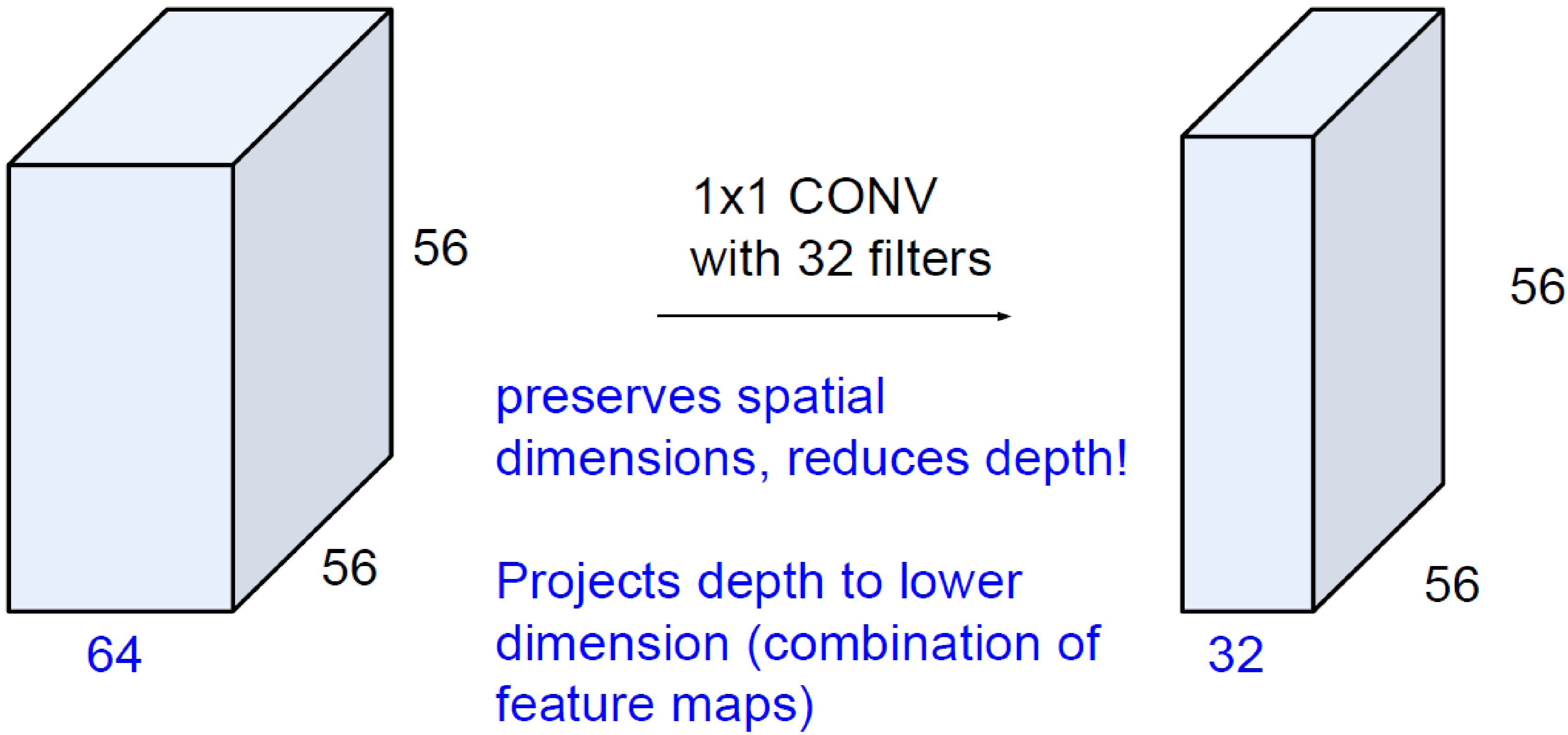
GoogleNet

Reminder: 1x1 convolutions



GoogleNet

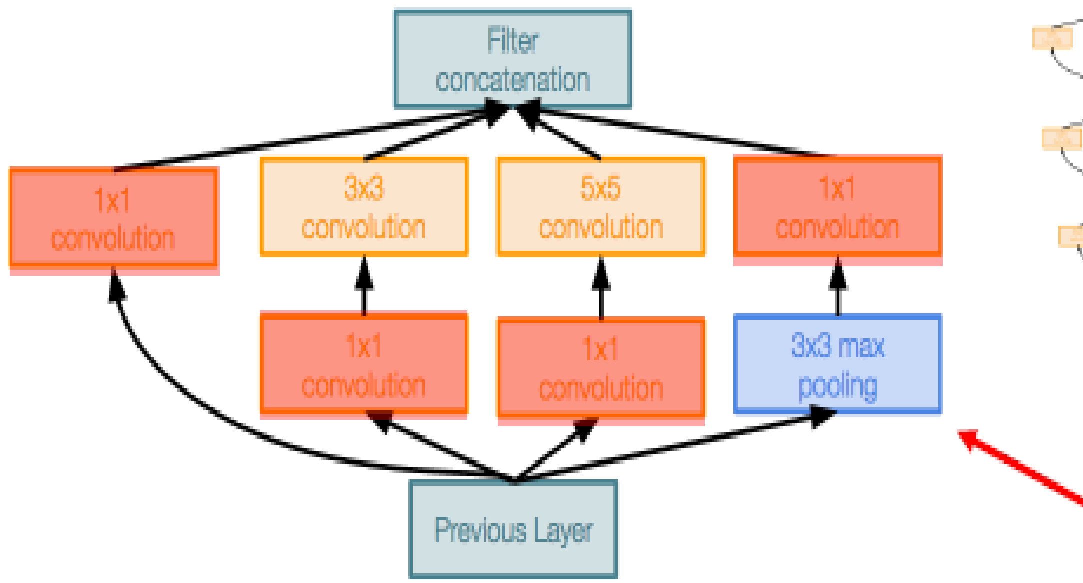
Reminder: 1x1 convolutions

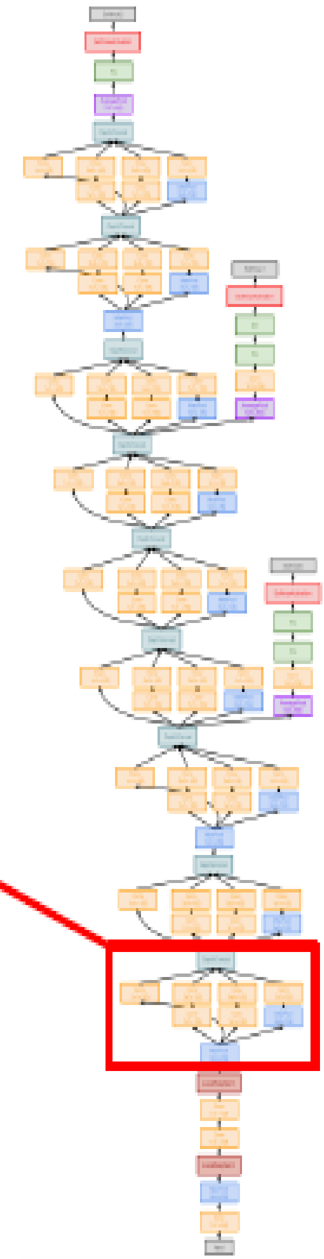


GoogleNet

- 인셉션 모듈(Inception module):

- 1x1 버틀넥(bottleneck) 층은 합성곱연산이 일어나기 전에 채널 차원을 감소시킨다.

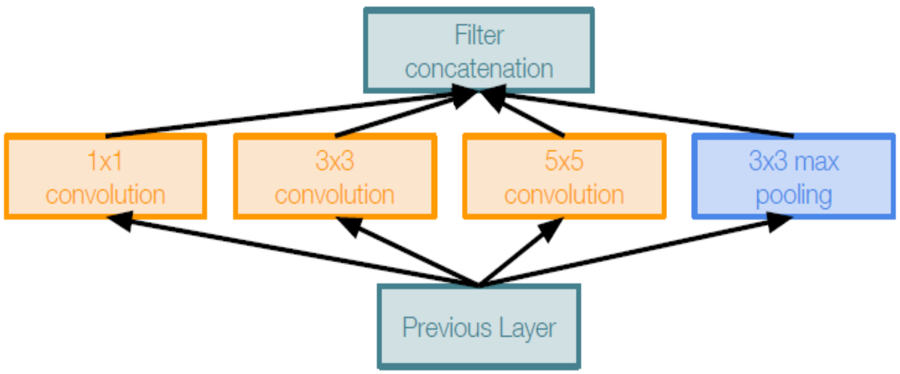




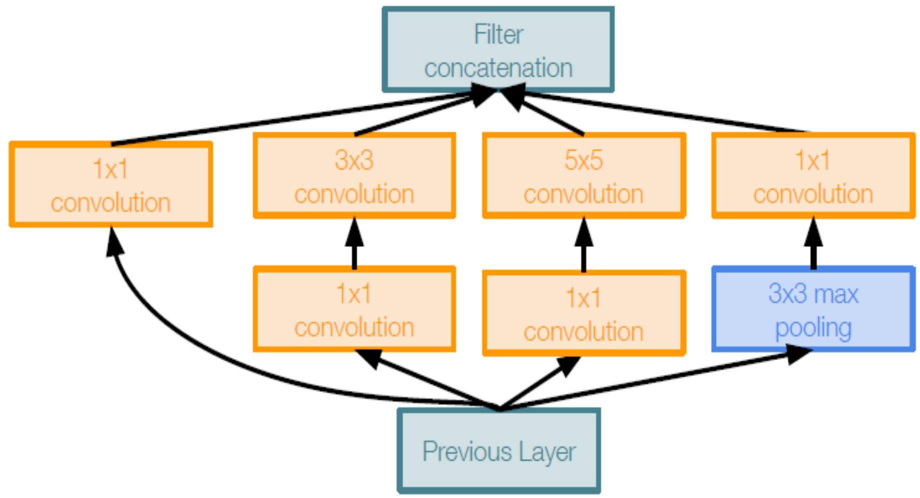
GoogleNet

Case Study: GoogLeNet

[Szegedy et al., 2014]



Naive Inception module



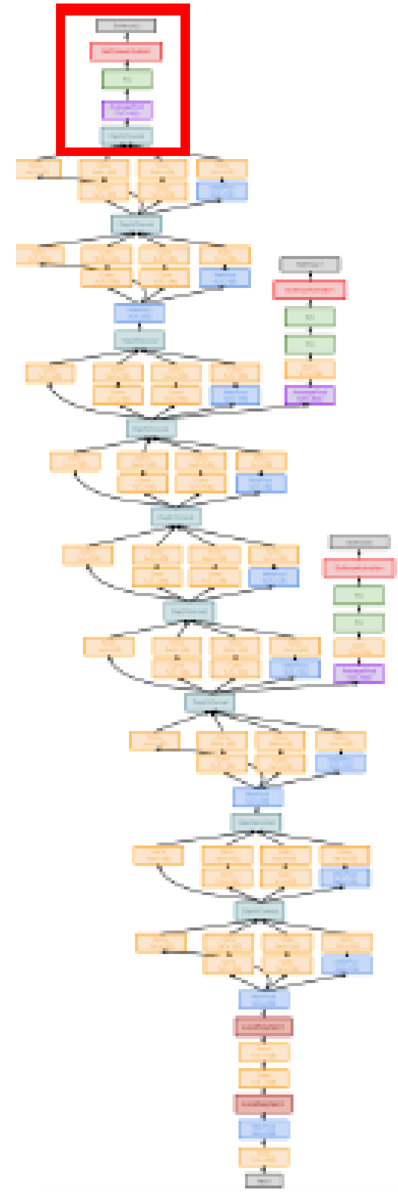
Inception module with dimension reduction

GoogleNet

- 글로벌 평균 풀링(GAP: Global Average Pooling)

- 신경망의 마지막 부분에서 계산량이 많은 FC 대신 GAP을 사용해 차원을 감소시키고, 최종적으로 점수를 구하기 위해 단 하나의 선형층을 사용한다.

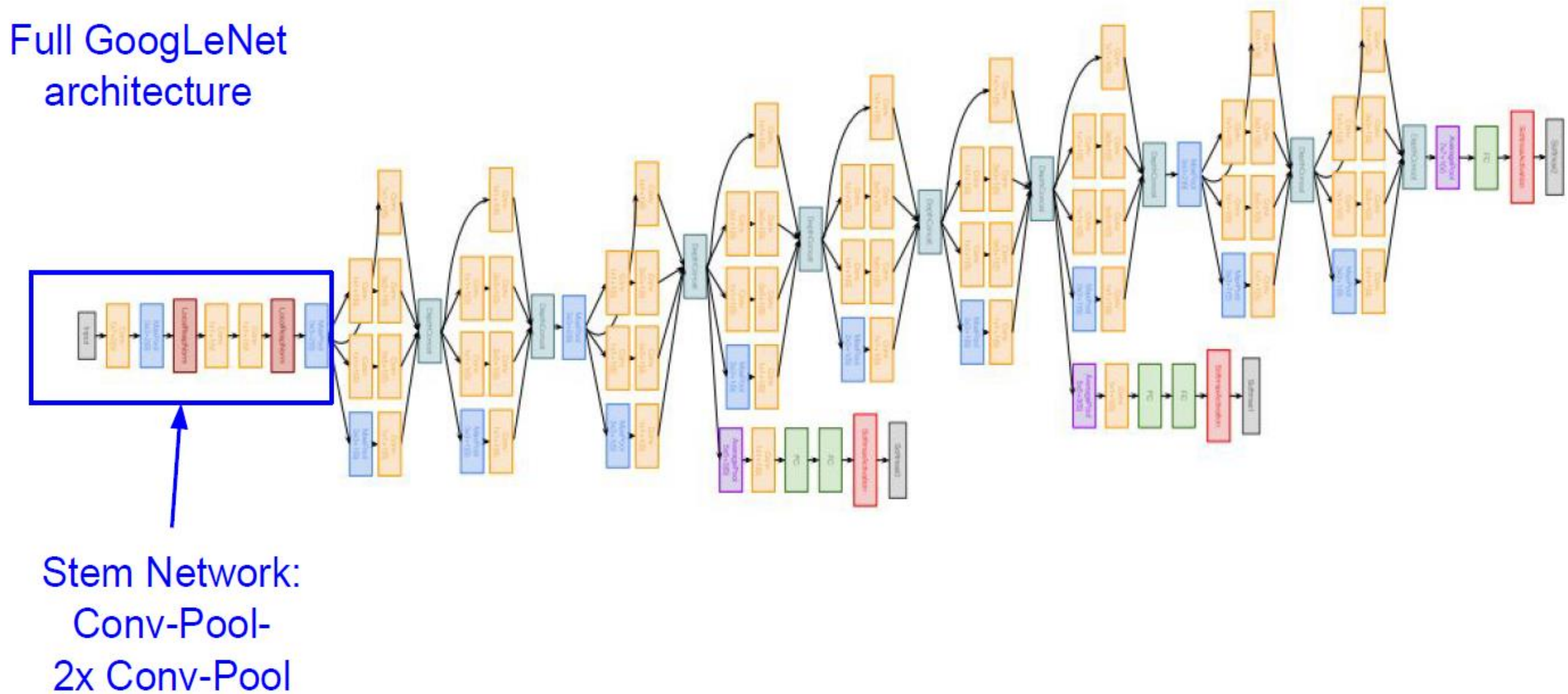
- VGG16의 경우 FC층에서 많은 파라미터를 계산해야 했다.



GoogleNet

Case Study: GoogLeNet

[Szegedy et al., 2014]

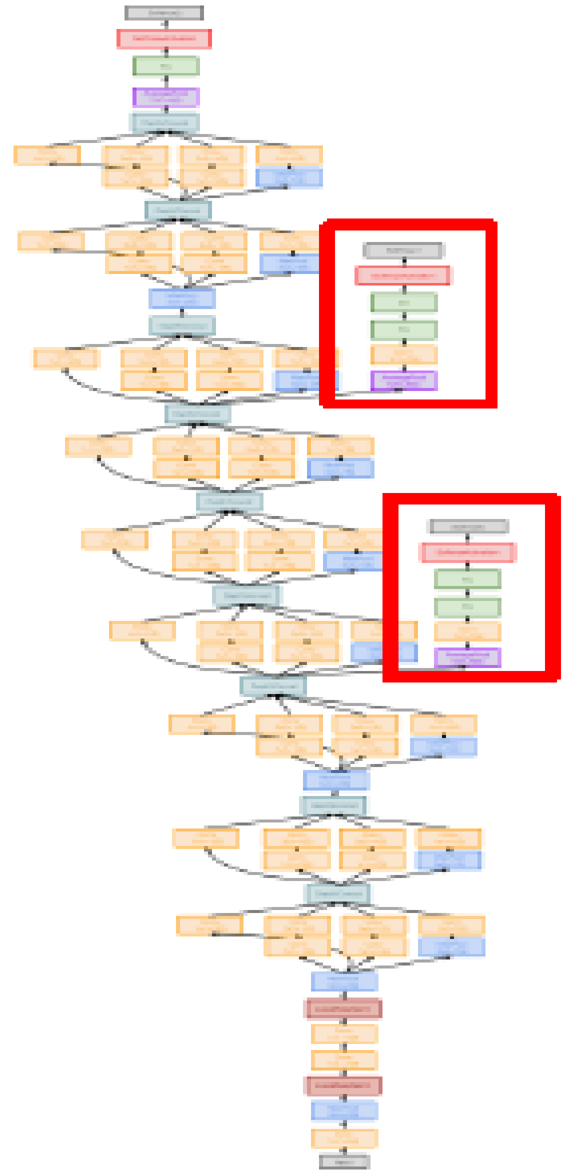


GoogleNet

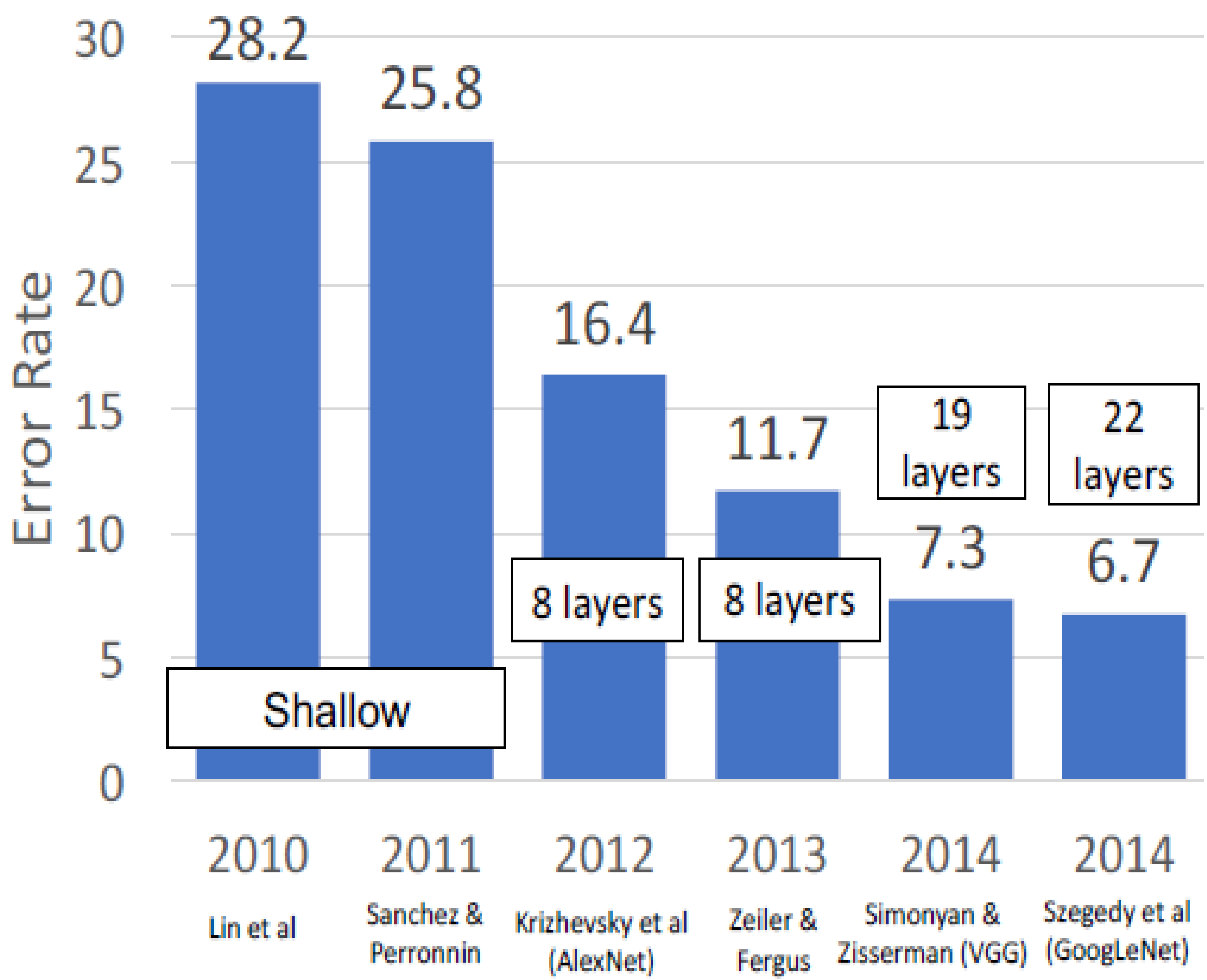
- 보조 분류기

- 신경망이 너무 커져 최종 손실로 학습이 잘 안됨 즉 역전파가 깨끗하게 안됨.

- 배치정규화 등장 이후 더 이상 이 방법은 사용하지 않는다.



GoogleNet



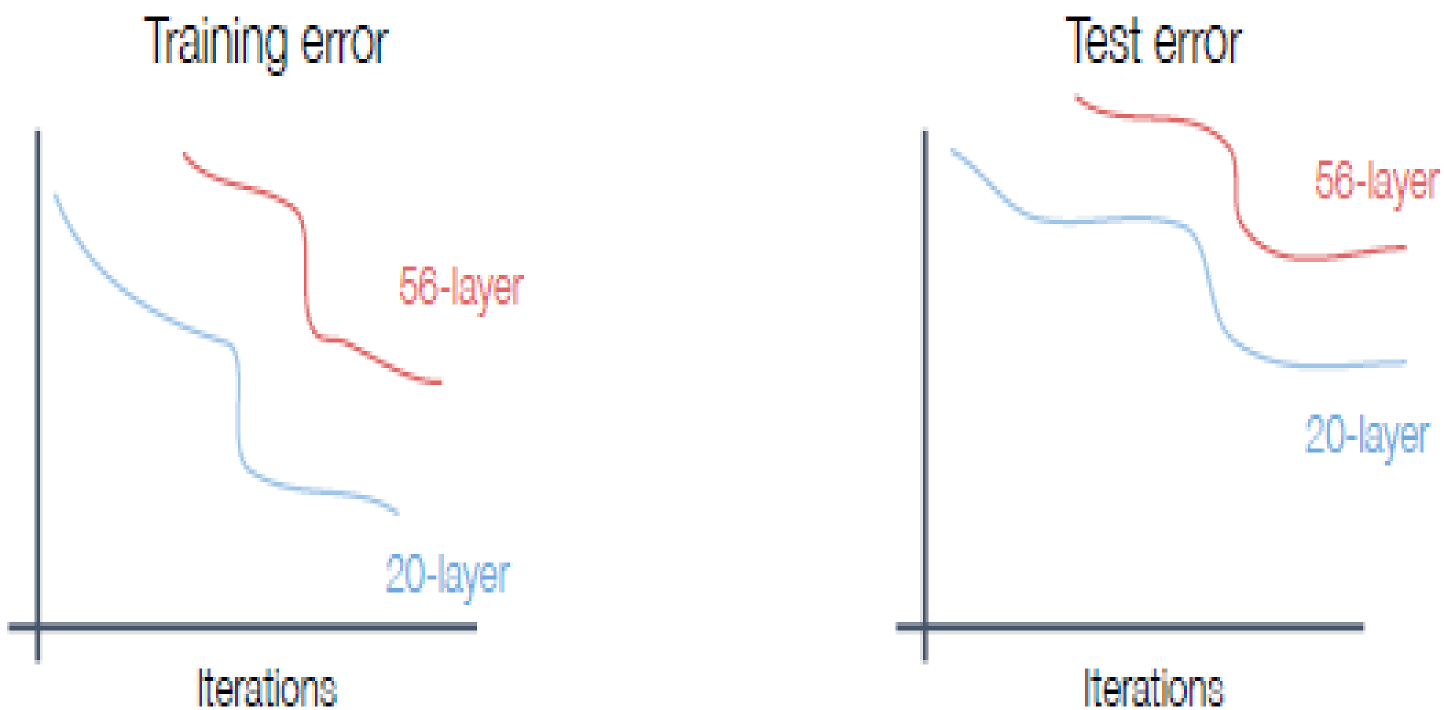
Error Rate
30
25
20
15
10
5
0
28.2
25.8
16.4
11.7
7.3
6.7
Shallow
8 layers
8 layers
19 layers
22 layers
2010
Lin et al
2011
Sanchez & Perronnin
2012
Krizhevsky et al (AlexNet)
2013
Zeiler & Fergus
2014
Simonyan & Zisserman (VGG)
2014
Szegedy et al (GoogLeNet)

ResNet

ResNet의 등장

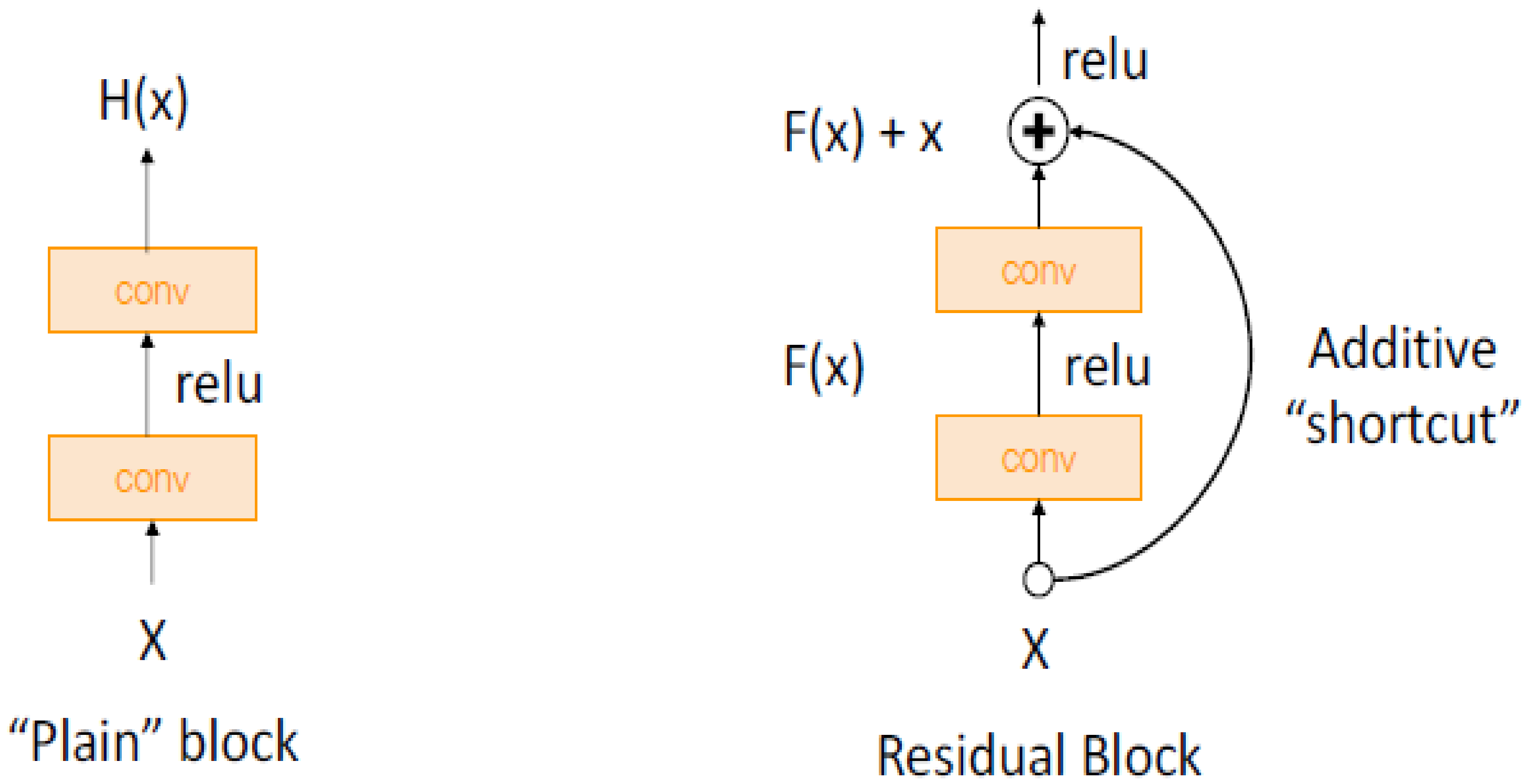
- 문제: 더 깊은 망이 과적합으로 테스트 오차는 클 수 있지만, 훈련오차도 더 크게 나온다. 이것은 뭔가 잘못된 거이 아닐까?

- 더 복잡한 깊은 망의 과소적합 문제 ????



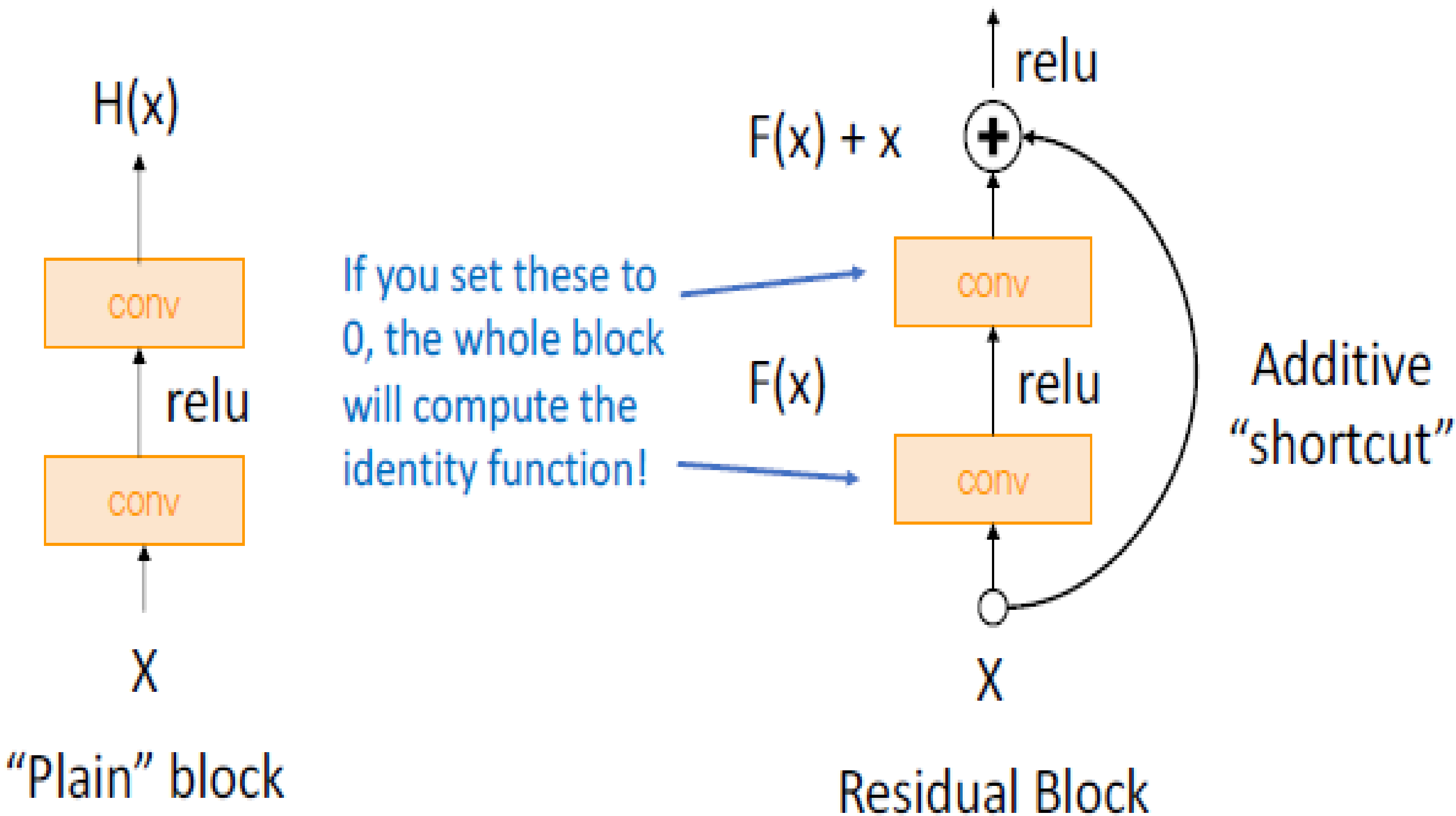
ResNet의 등장

- 아이디어: 깊은 신경망이 낮은 신경망을 복제할 수 있도록 항등함수를 설정하고 추가 정보만 망을 통해서 학습한다.



ResNet

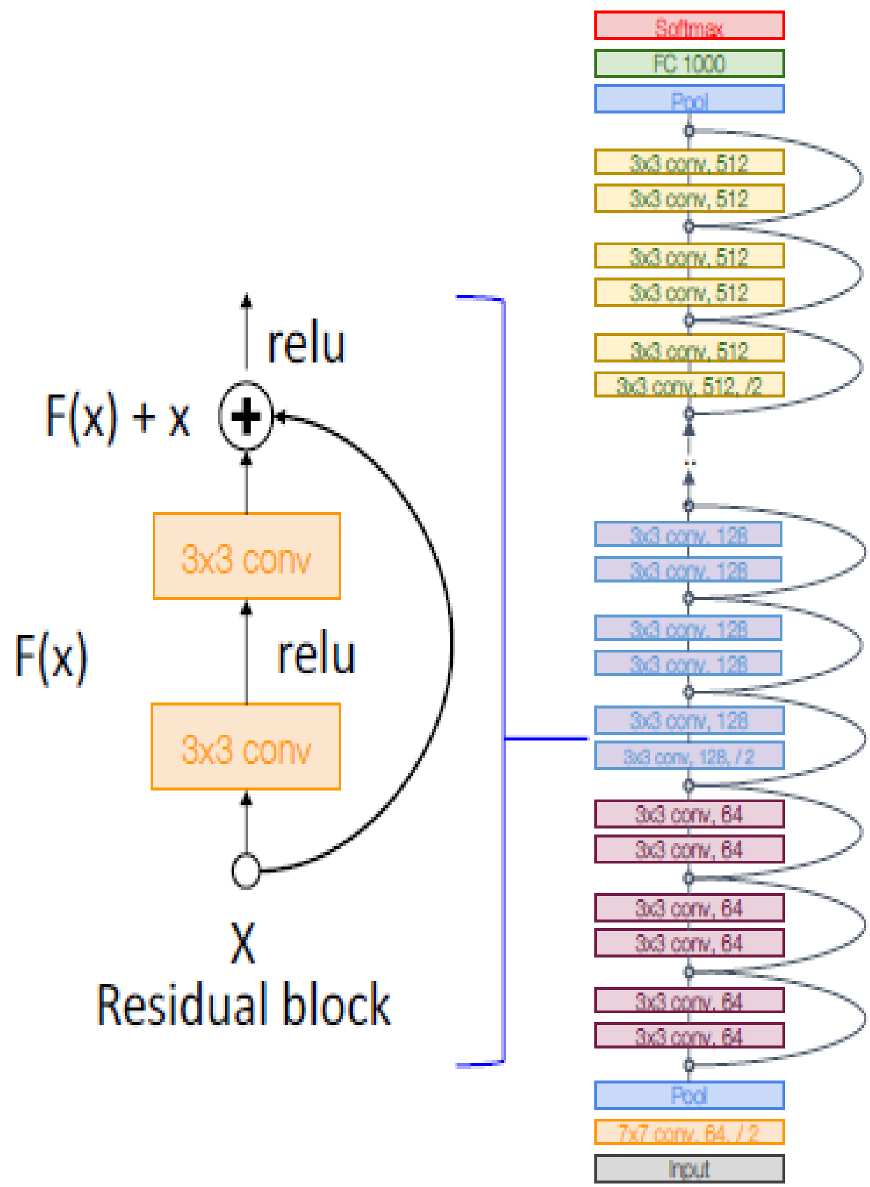
- 아이디어: 깊은 신경망이 낮은 신경망을 복제할 수 있도록 항등함수를 설정하고 추가 정보만 망을 통해서 학습한다.



ResNet

- ResNet은 잔차블록을 쌓은 것이다.
- VGG처럼 각 잔차블록은 2개의 3x3 conv를 갖는다.
- 네트워크는 스테이지(stage)로 구성된다.

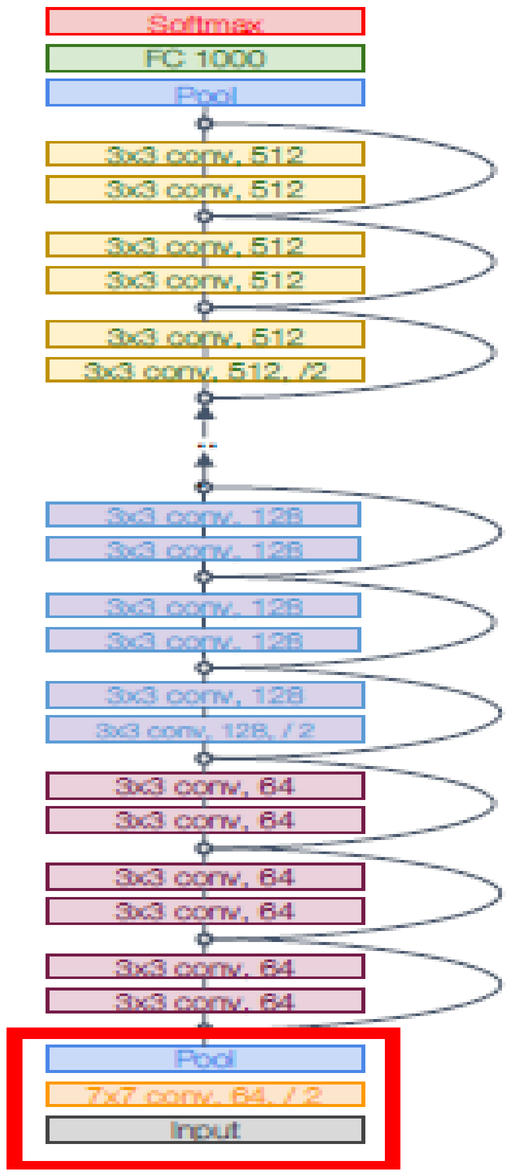
- 각 스테이지의 첫째 블록은 stride-2 conv로 크기를 ½로 줄이고, 채널을 2배로 늘린다.



ResNet

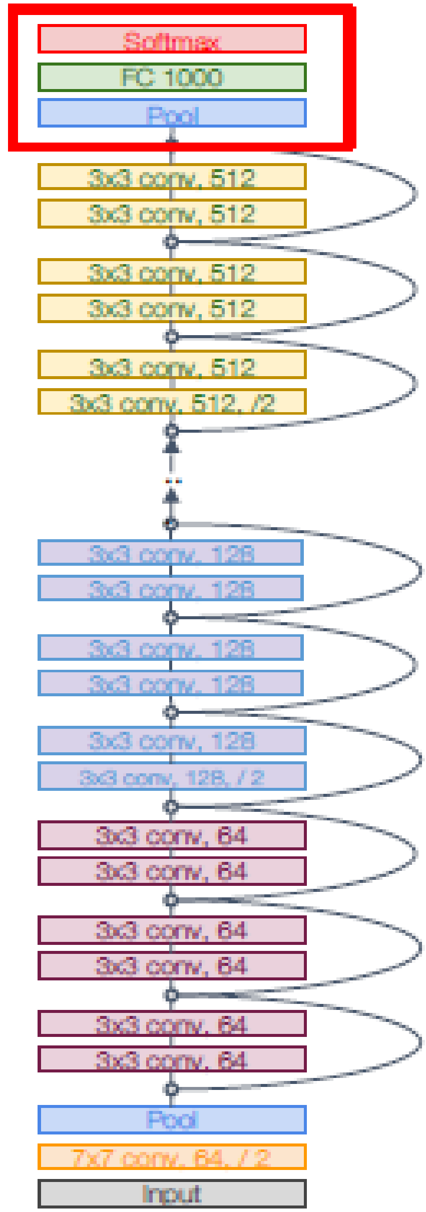
- 구글넷처럼 시작부분에 적극적 스템을 사용

- 잔차블록을 적용하기 이전에 4x로 다운샘플링



ResNet

- 구글넷처럼 마지막 부분에 GAP를 사용하고, 선형층을 최종적으로 적용한다.



ResNet

- VGG보다 깊이는 더 깊어지고 계산량은 오히려 줄었다.

Residual Networks

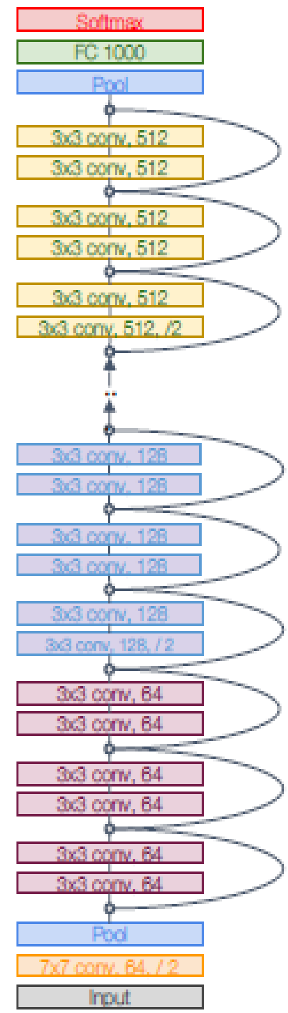
ResNet-18:
Stem: 1 conv layer
Stage 1 (C=64): 2 res. block = 4 conv
Stage 2 (C=128): 2 res. block = 4 conv
Stage 3 (C=256): 2 res. block = 4 conv
Stage 4 (C=512): 2 res. block = 4 conv
Linear

ImageNet top-5 error: 10.92
GFLOP: 1.8

ResNet-34:
Stem: 1 conv layer
Stage 1: 3 res. block = 6 conv
Stage 2: 4 res. block = 8 conv
Stage 3: 6 res. block = 12 conv
Stage 4: 3 res. block = 6 conv
Linear

ImageNet top-5 error: 8.58
GFLOP: 3.6

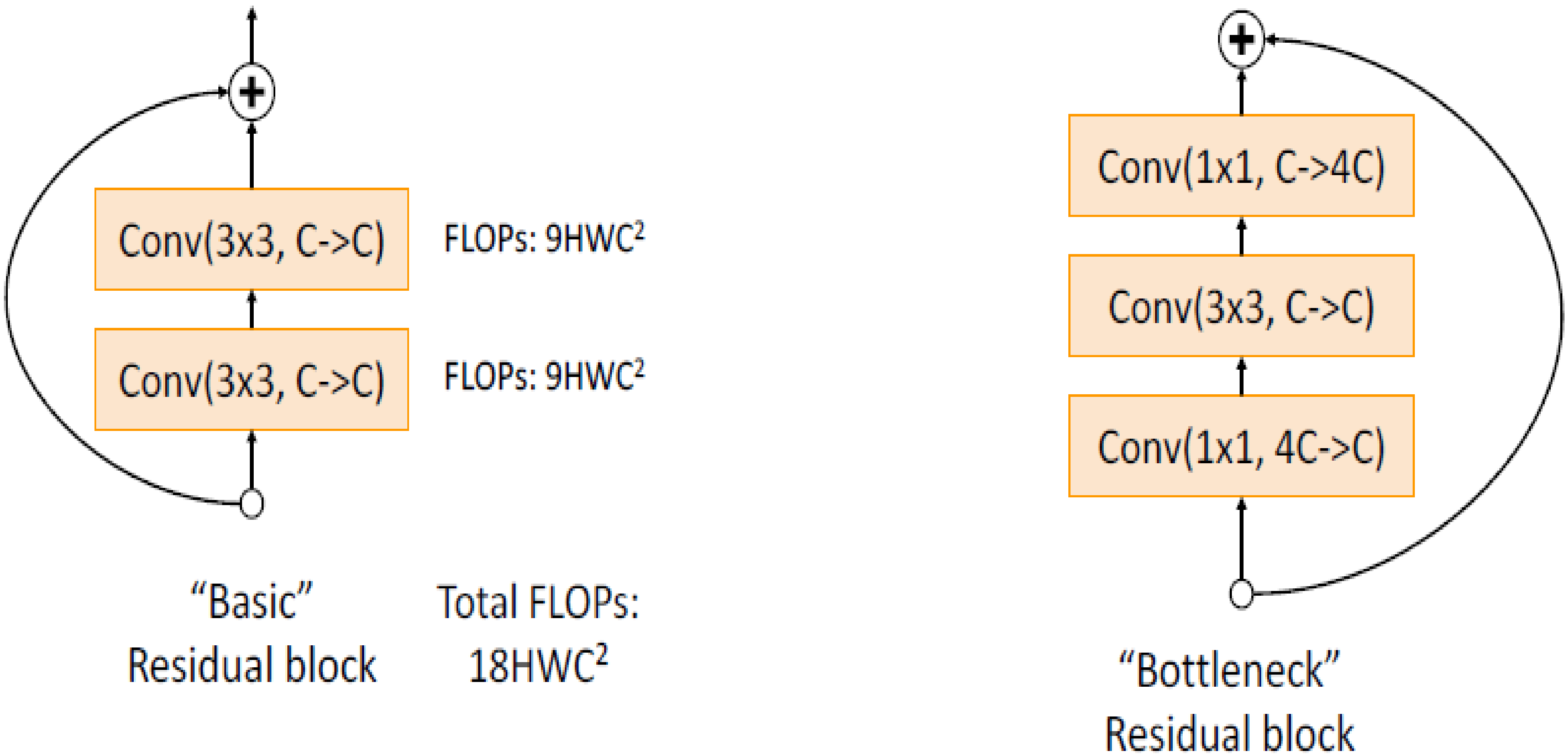
VGG-16:
ImageNet top-5 error: 9.62
GFLOP: 13.6



He et al, "Deep Residual Learning for Image Recognition", CVPR 2016
Error rates are 224x224 single-crop testing, reported by torchvision

ResNet

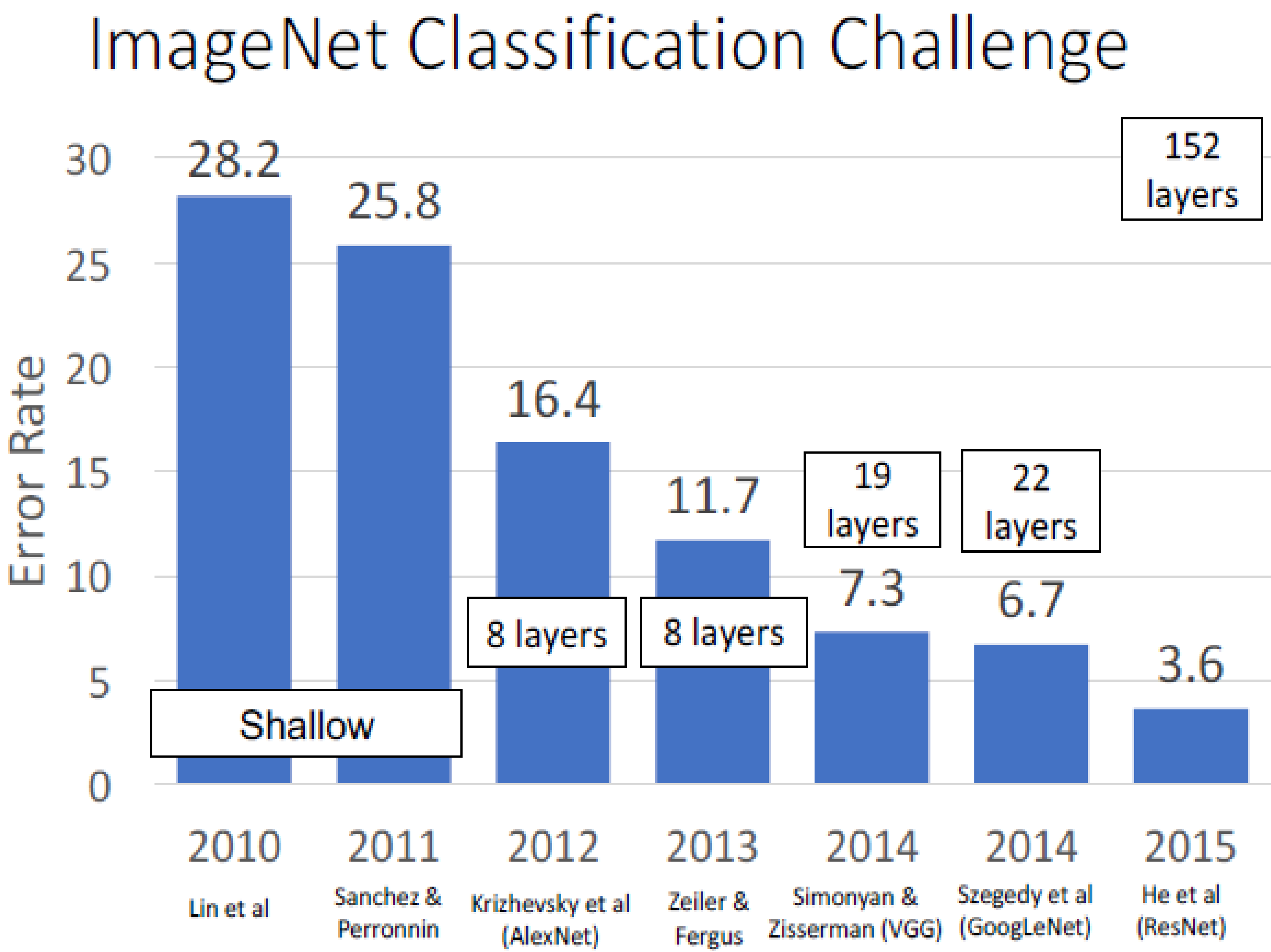
- 버틀넥 사용: 더욱 계산량을 감소한다.



ResNet

- 이제야 깊은 망이 얇은 망보다 더 좋은 성과를 내기 시작한다. ResNet은 ILSRVC와 COCO2015 모두에서 1등을 휩쓸었다.

이미지넷 경연대회 (Resnet의 출현까지)

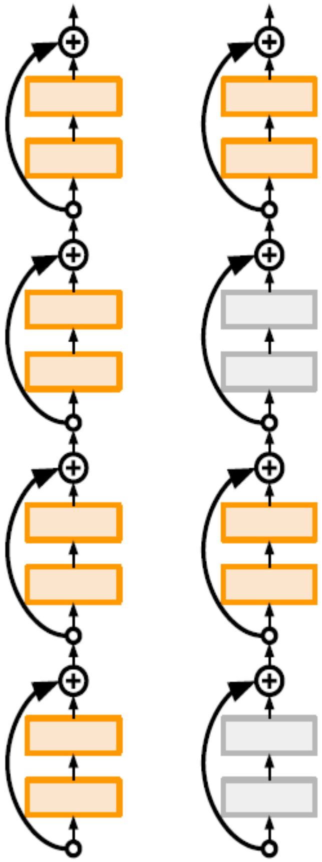


확률적 깊이(Stochastic Depth): ResNets의 개량

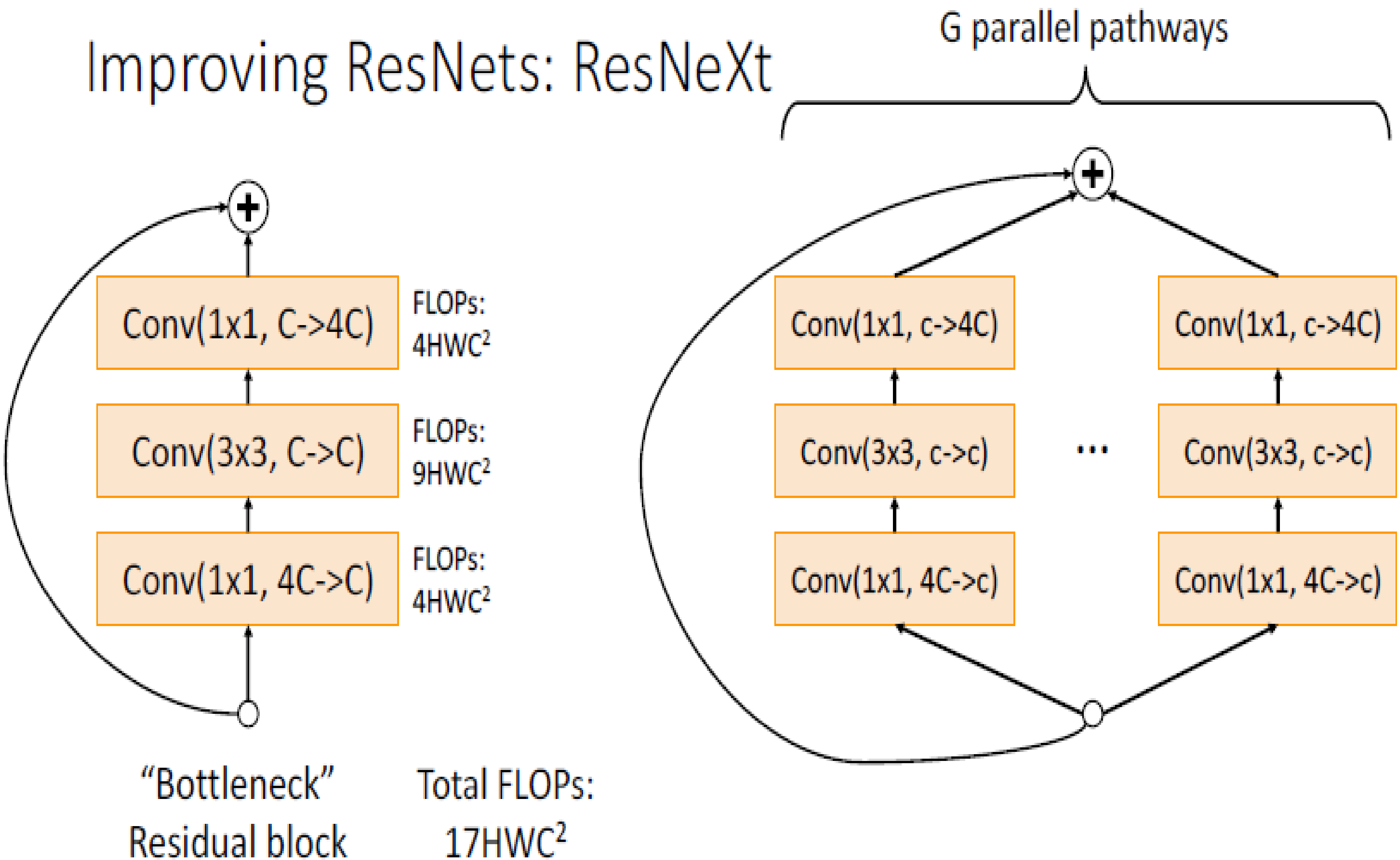
Deep Networks with Stochastic Depth

[Huang et al. 2016]

- Motivation: reduce vanishing gradients and training time through short networks during training
- Randomly drop a subset of layers during each training pass
- Bypass with identity function
- Use full deep network at test time

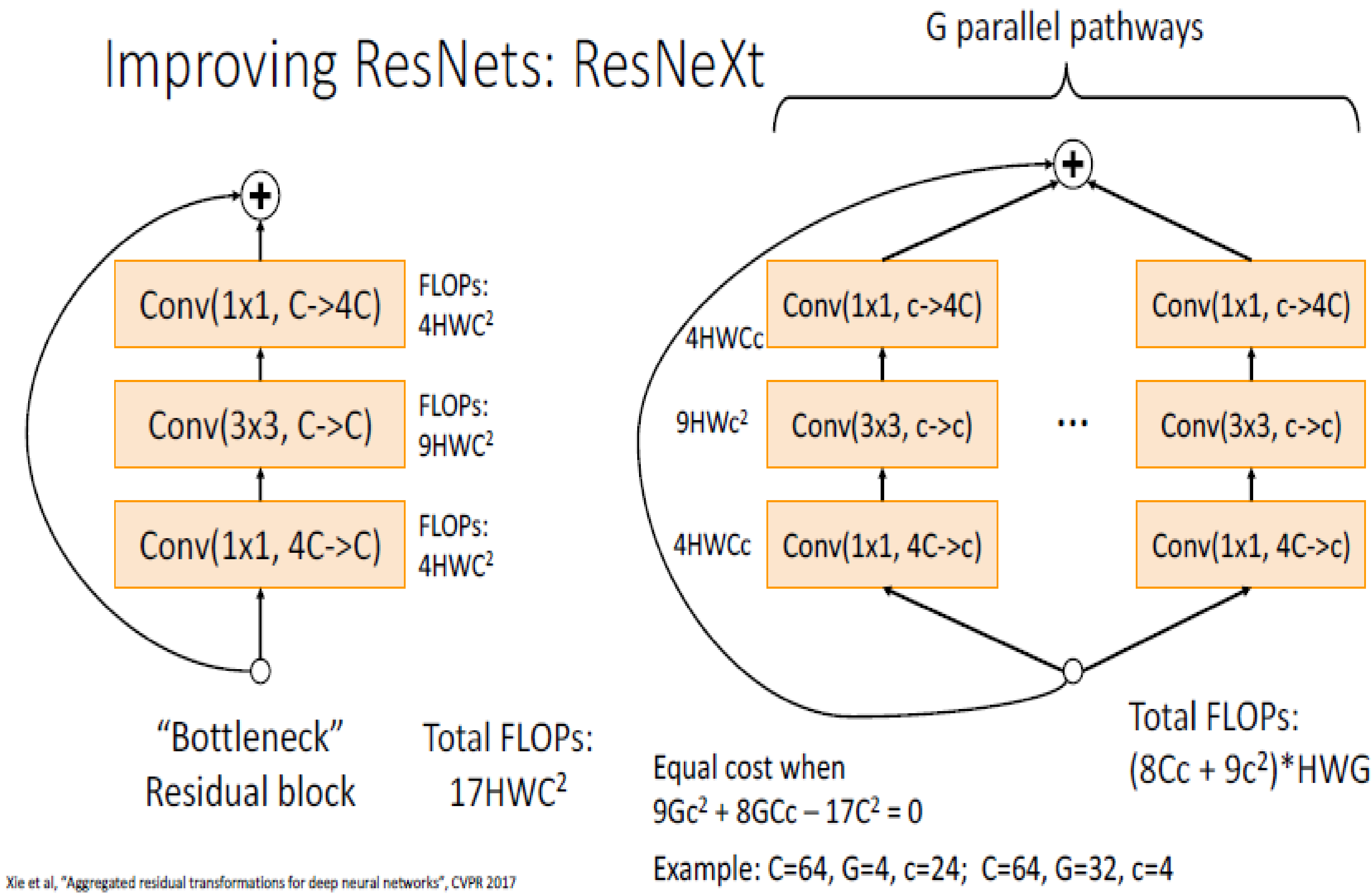


ResNet의 개선 ResNeXt

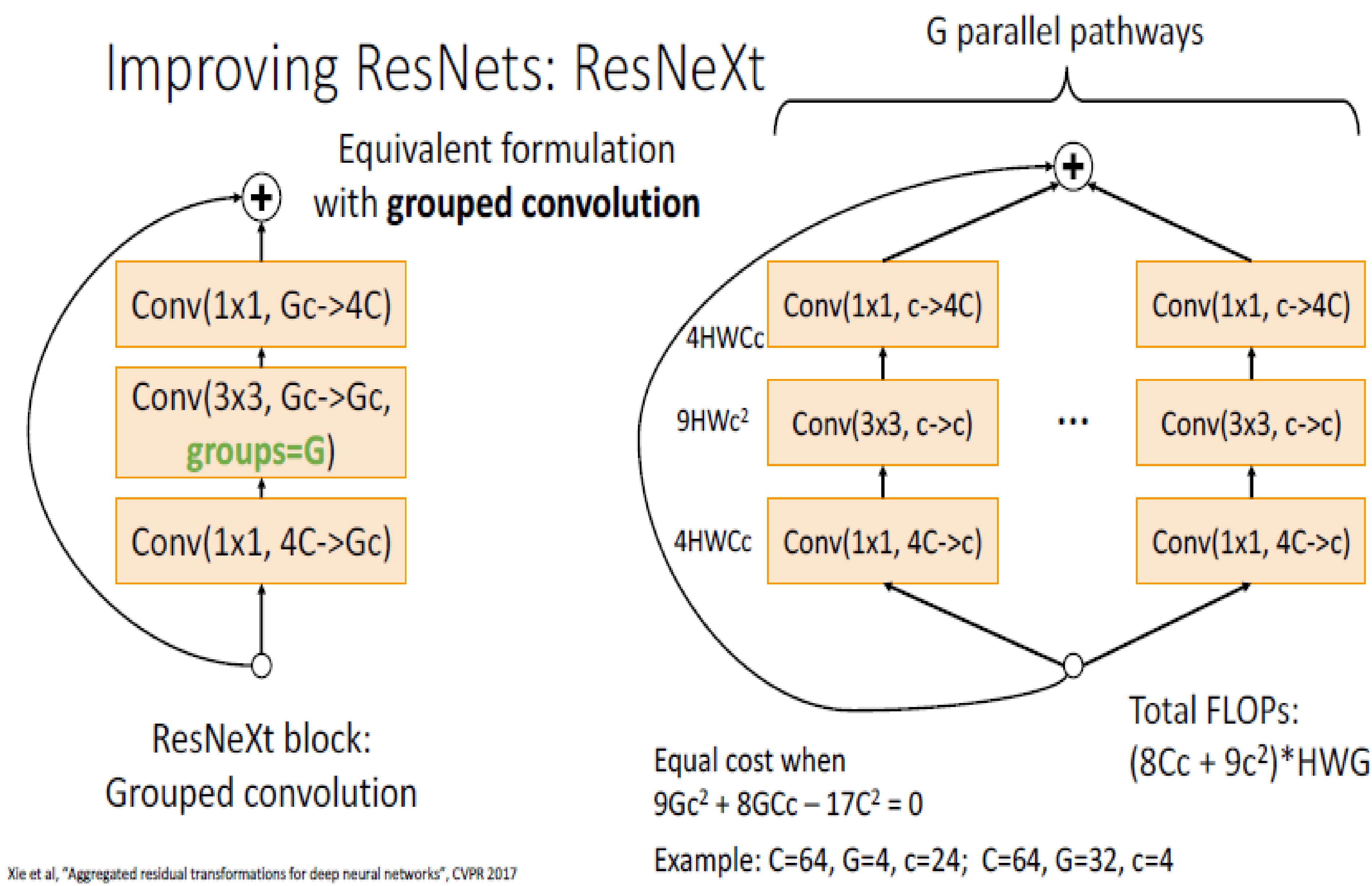


Xie et al, "Aggregated residual transformations for deep neural networks", CVPR 2017

ResNet의 개선 ResNeXt



ResNet의 개선 ResNeXt

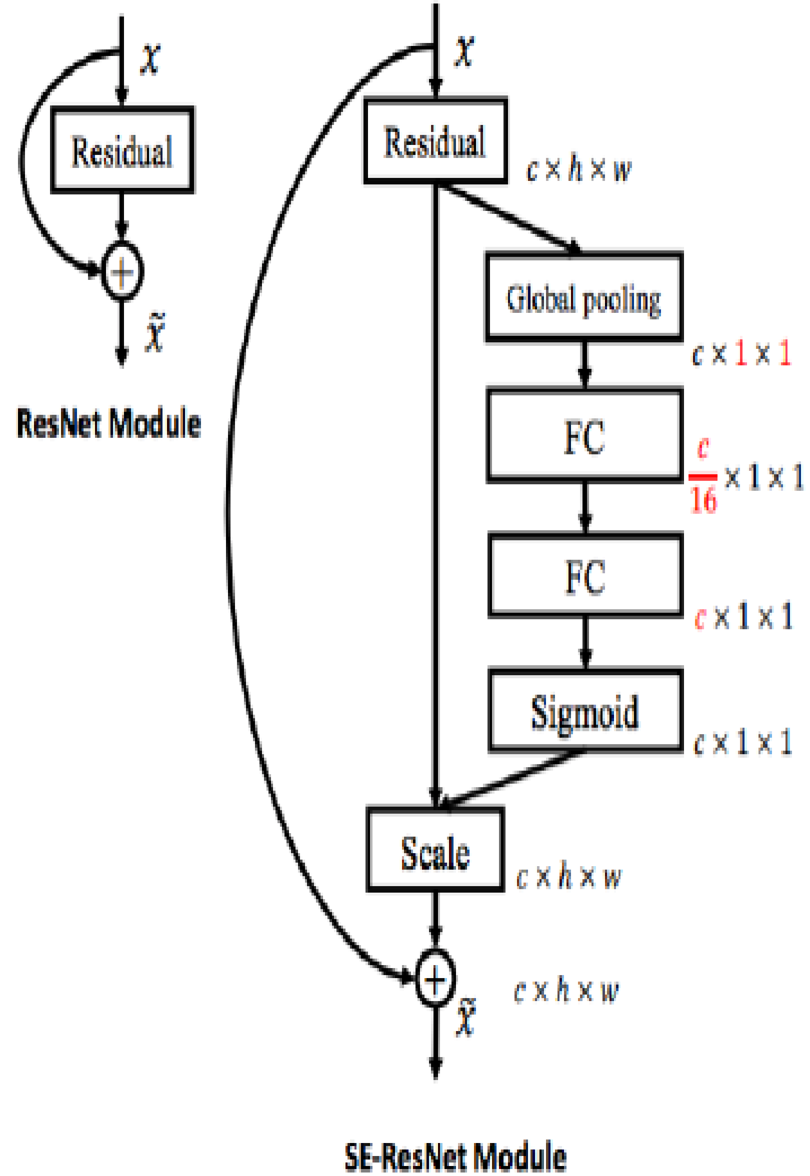


SE 네트워크 (Squeeze and Excitation Network)

Adds a "Squeeze-and-excite" branch to each residual block that performs global pooling, full-connected layers, and multiplies back onto feature map

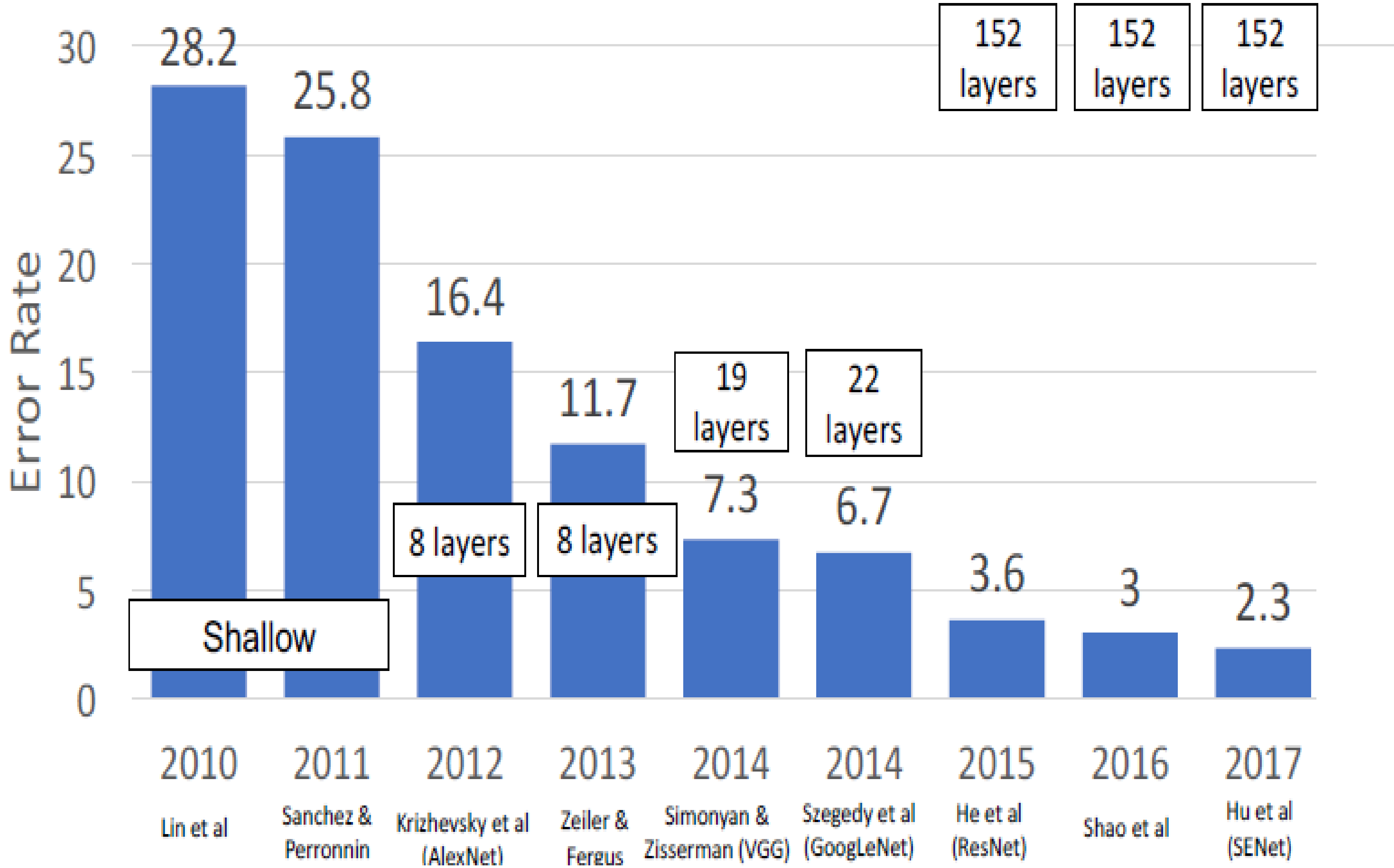
Adds **global context** to each residual block!

Won ILSVRC 2017 with ResNeXt-152-SE



Hu et al, "Squeeze-and-Excitation networks", CVPR 2018

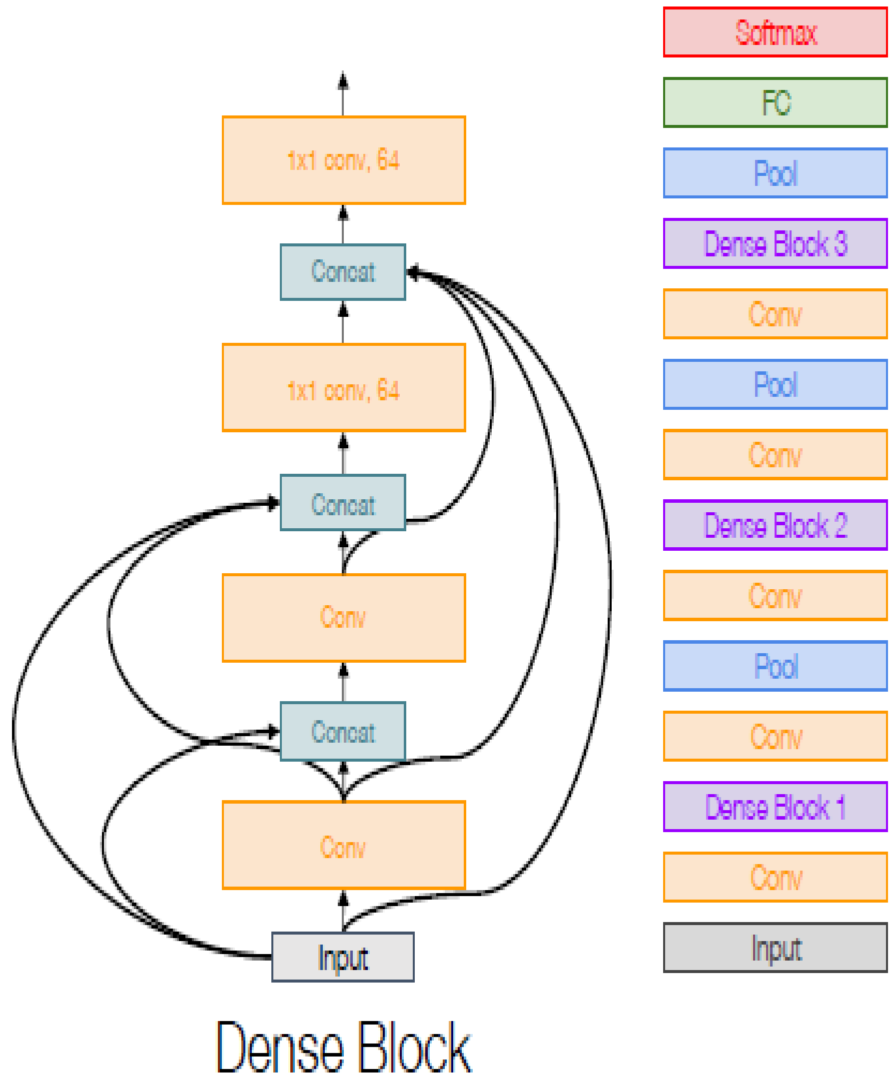
ImageNet 분류 경연대회 (2017년이후에는 Kaggle로 이전)



밀집결합 신경망(Densely Connected Network)

Dense blocks where each layer is connected to every other layer in feedforward fashion

Alleviates vanishing gradient, strengthens feature propagation, encourages feature reuse

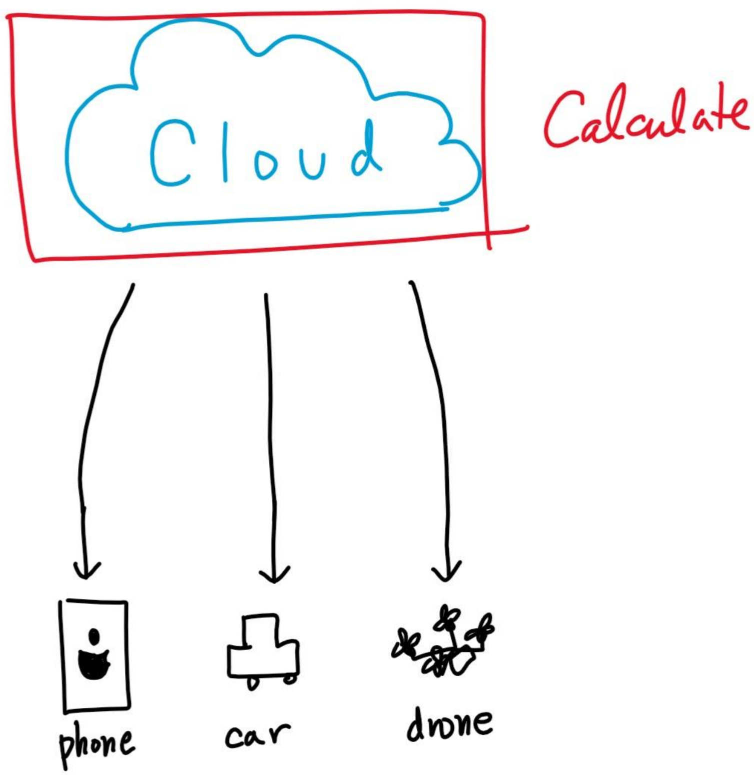


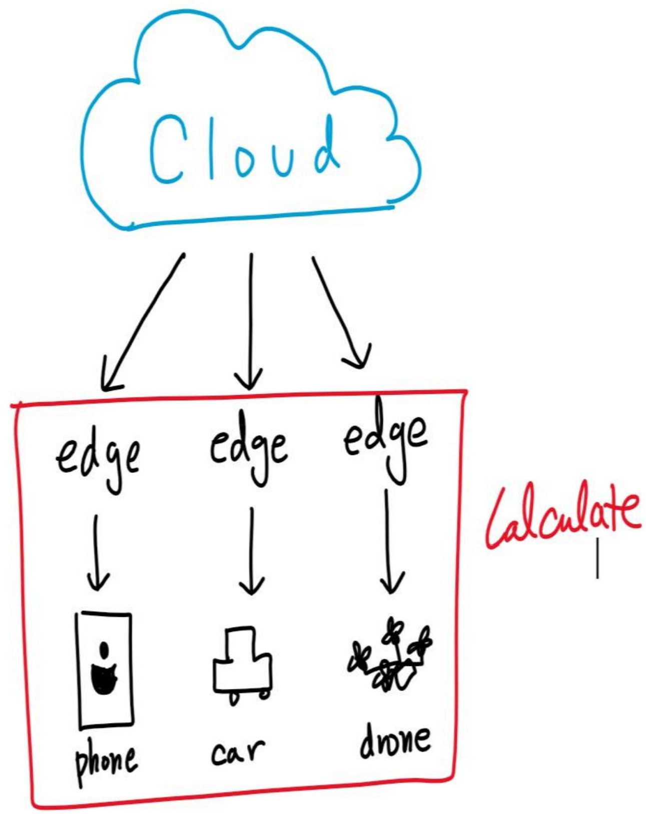
Huang et al, "Densely connected neural networks", CVPR 2017

ResNet

모바일넷의 필요성

- 이슈: 클라우드 대 모바일





작은 딥신경망 기법

- 작은 딥신경망을 위한 기법들이 개발되고 있다.

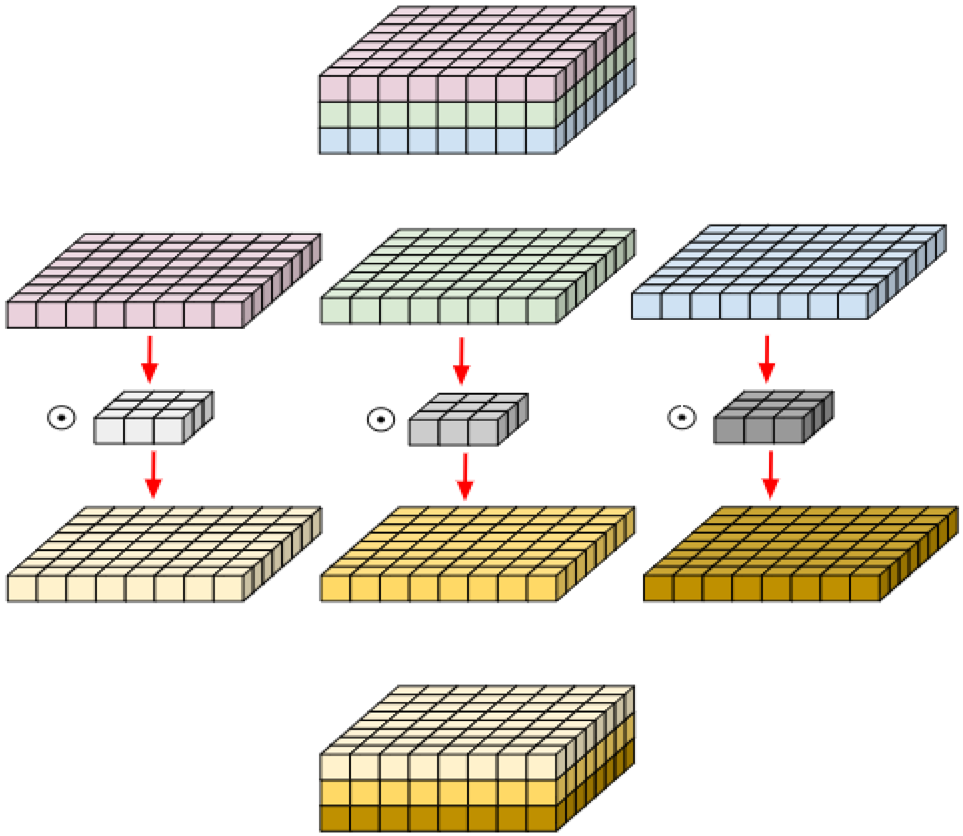
Techniques for Small Deep Neural Networks

- Remove Fully-Connected Layers
- Kernel Reduction (3x3 → 1x1)
- Channel Reduction
- Evenly Spaced Downsampling
- Depthwise Separable Convolutions
- Shuffle Operations
- Distillation & Compression

Slide Credit : Small Deep Neural Networks - Their Advantages, and Their Design by Forrest Iandola

채널별 합성곱

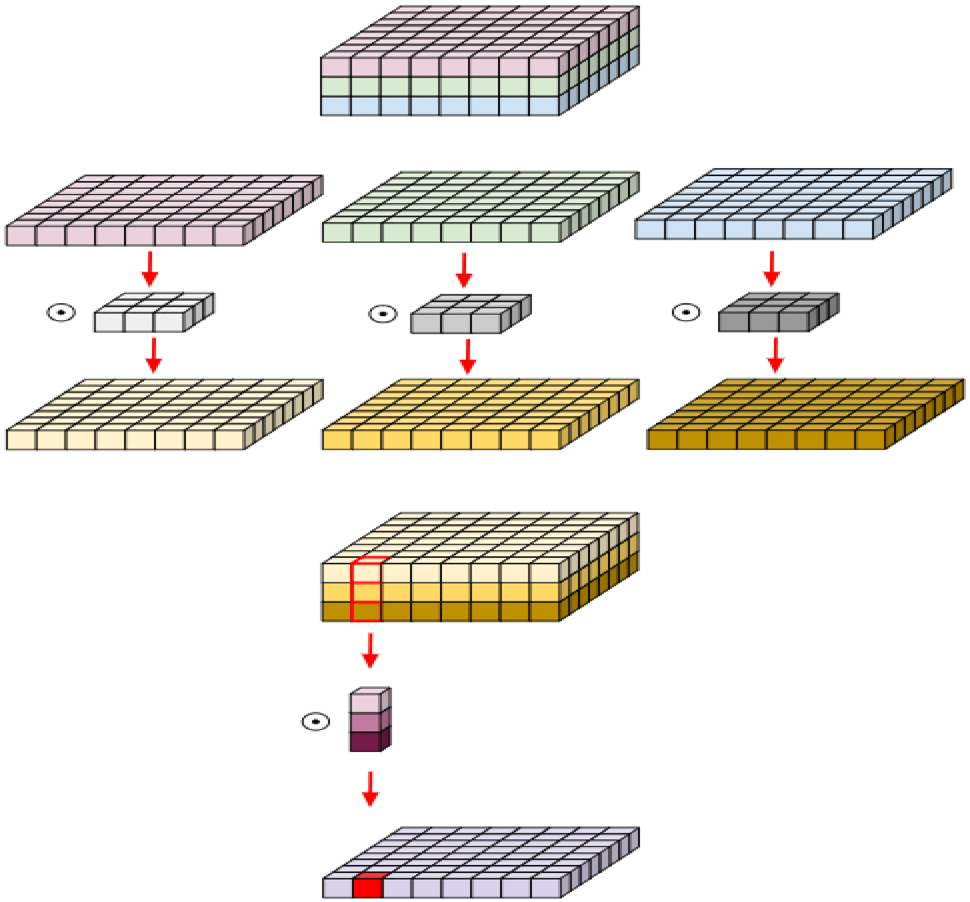
- 채널별 합성곱 (Depthwise Convolution)



- 채널별로 합성곱 실행 즉 채널방향이 아니라 공간 방향으로만 합성곱실행

채널별 합성곱 + 채널방향 합성곱

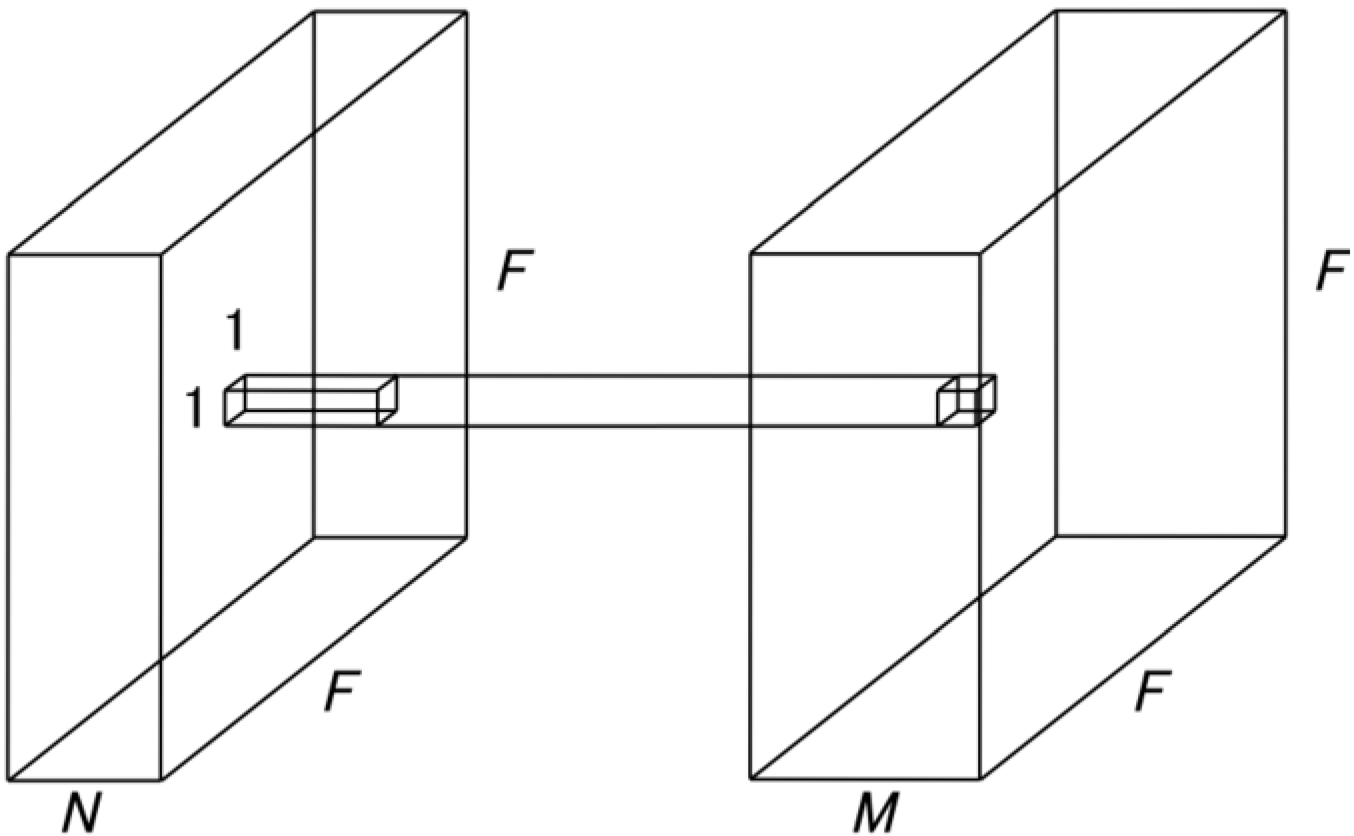
- 채널별 합성곱 이후 채널방향 개별 포인트 합성곱 (Depthwise Separable Convolution)



- 1단계에서는 공간방향으로 2단계에서는 채널방향으로 합성곱 실행한다.

개별 포인트 합성곱

- 개별 포인트 합성곱 (Pointwise Convolution)

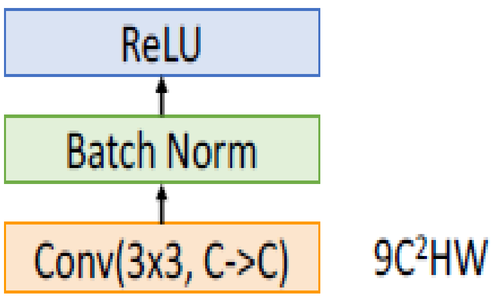


- 채널 방향으로 압축을 해 채널 수를 줄인다.

모바일넷 (MobileNets): 모바일 장치들을 위한 작은 신경망

Standard Convolution Block

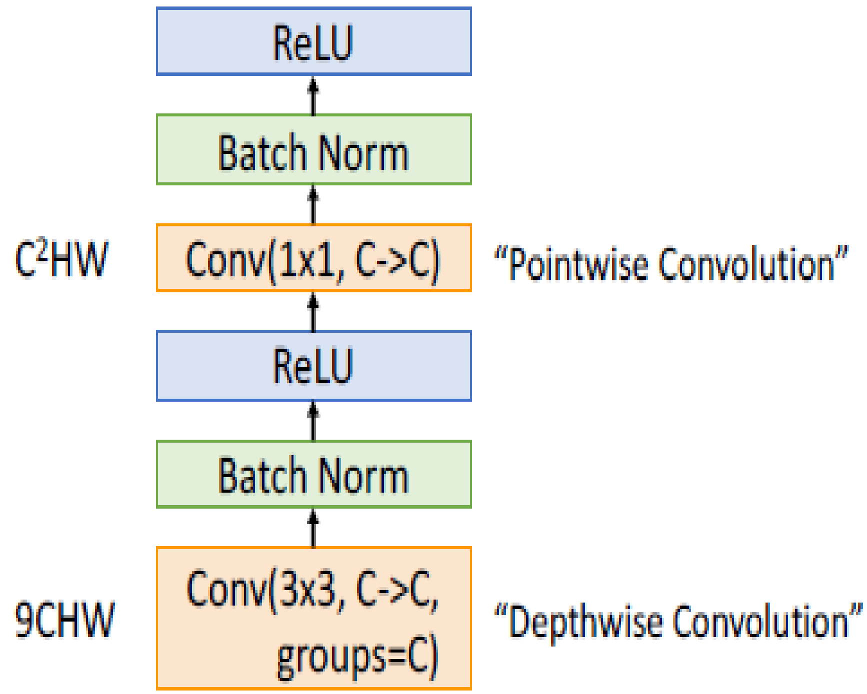
Total cost: $9C^2HW$



Speedup = $9C^2/(9C+C^2)$
$= 9C/(9+C)$
$\Rightarrow 9$ (as C->inf)

Depthwise Separable Convolution

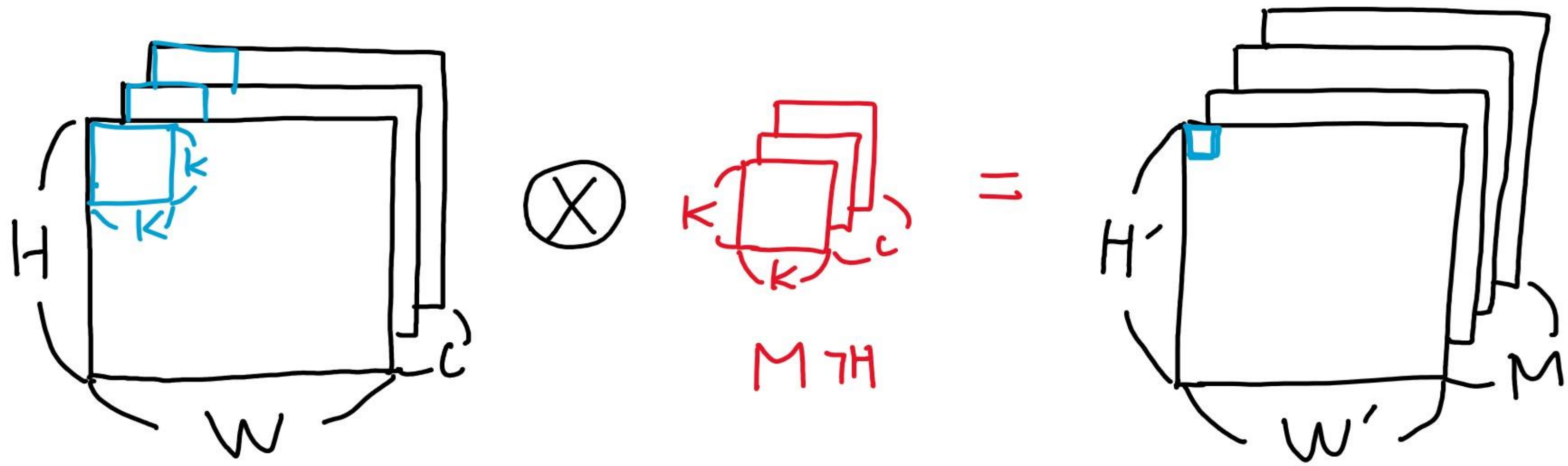
Total cost: $(9C + C^2)HW$

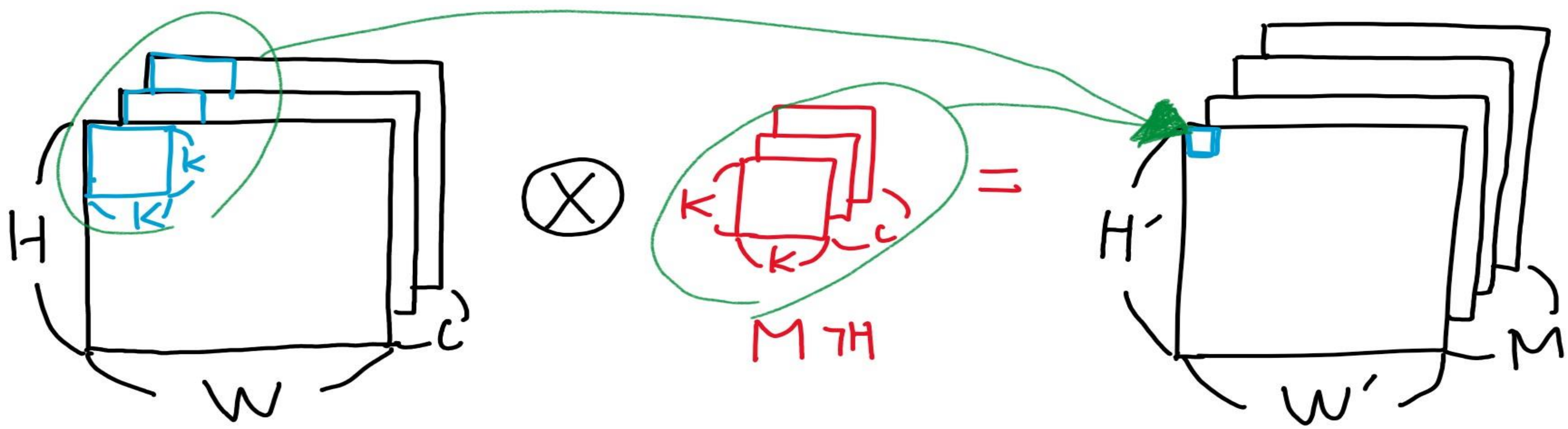


Howard et al, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", 2017

[참고] 모바일 넷이얼마나 기존 CNN 계산보다 절약하는가

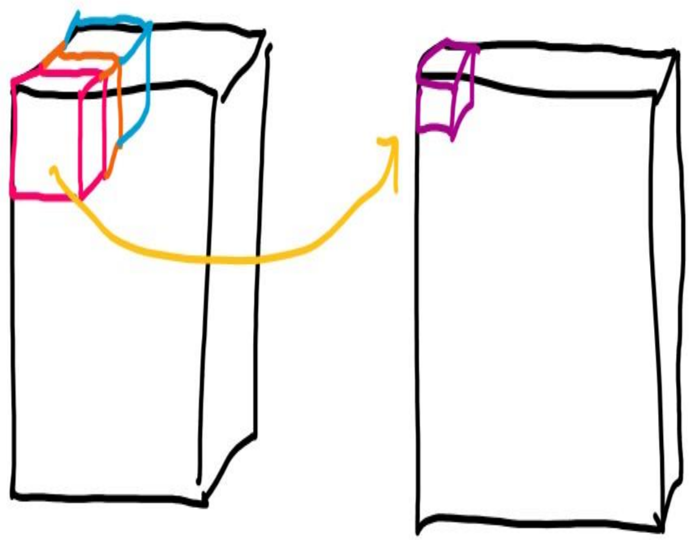
- 기존 CNN 계산의 연산은 여전히 매우 크다.

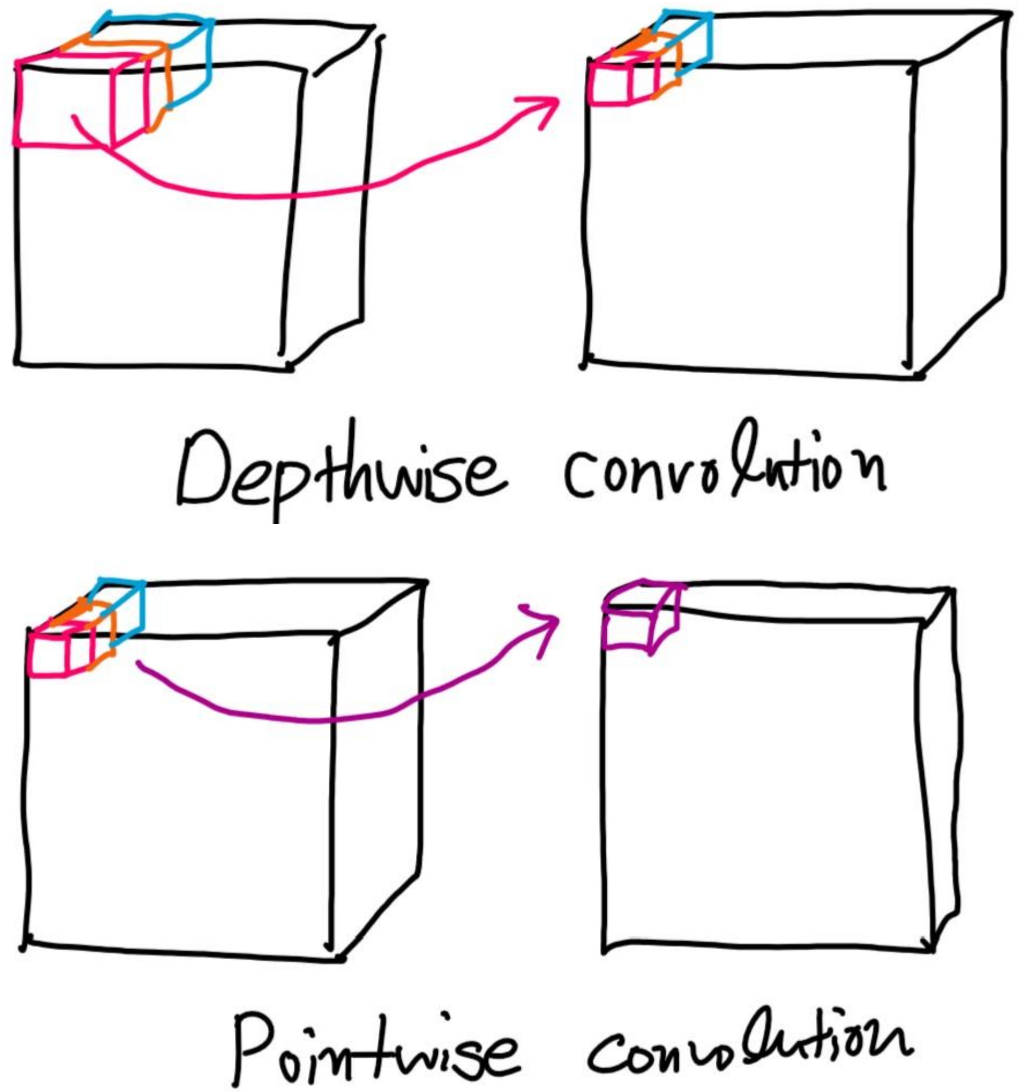




[참고] 모바일넷: 깊이별과 점별 합성곱 분리

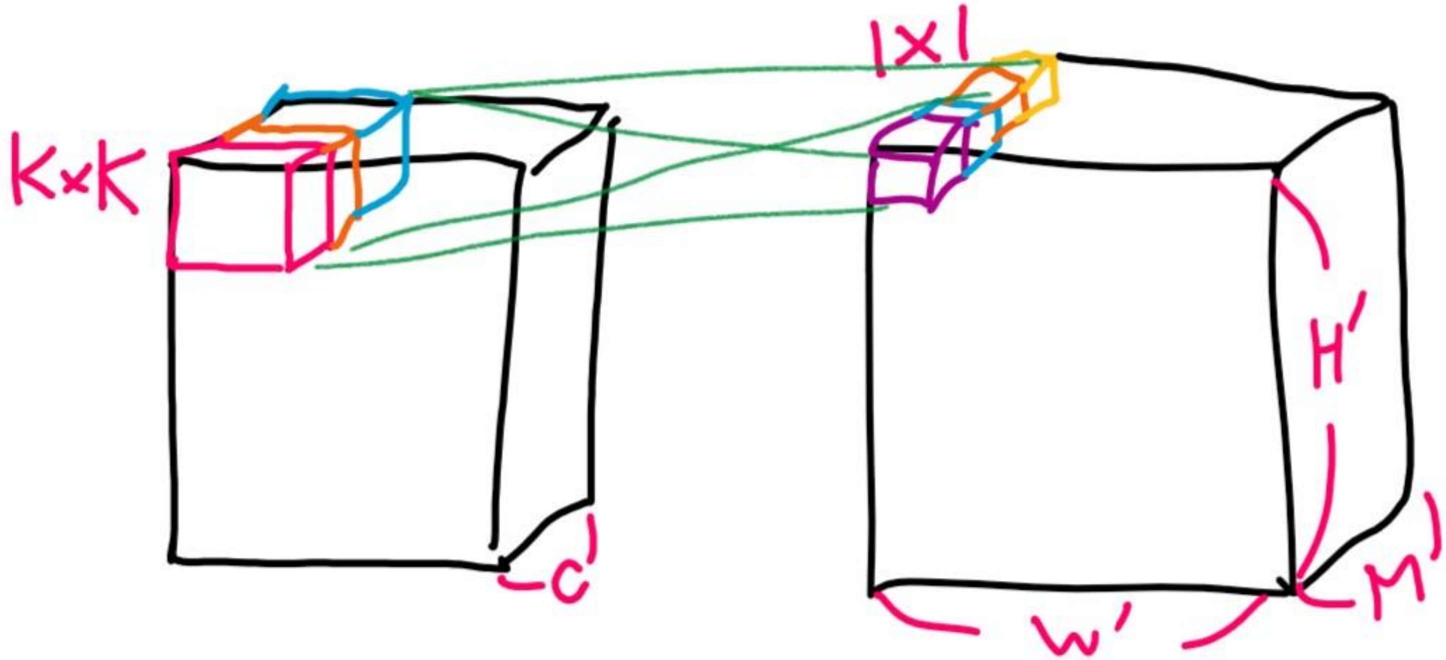
- 입체적으로 하던 연산을 깊이별 그리고 나서 점별 계산으로 분리함으로써 연산 절약





[참고] 모바일넷: 깊이별과 점별 합성곱 분리

- 기존 CNN의 계산 비용

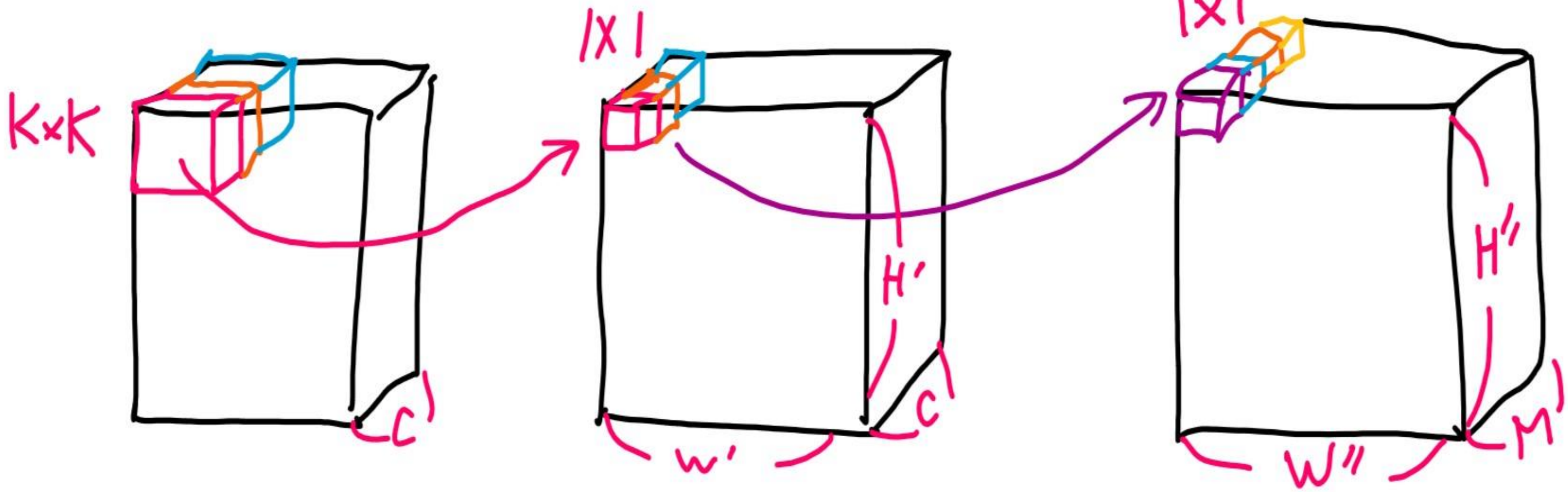


Parameter : $\boxed{K^2CM}$

Calculate Cost : $\boxed{K^2CM}\ H'W'$

[참고] 모바일넷: 깊이별과 점별 합성곱 분리

- 모바일넷의 계산비용: 깊이별 + 점별 계산으로의 분리로 연산 절약



Depthwise convolution + Pointwise convolution

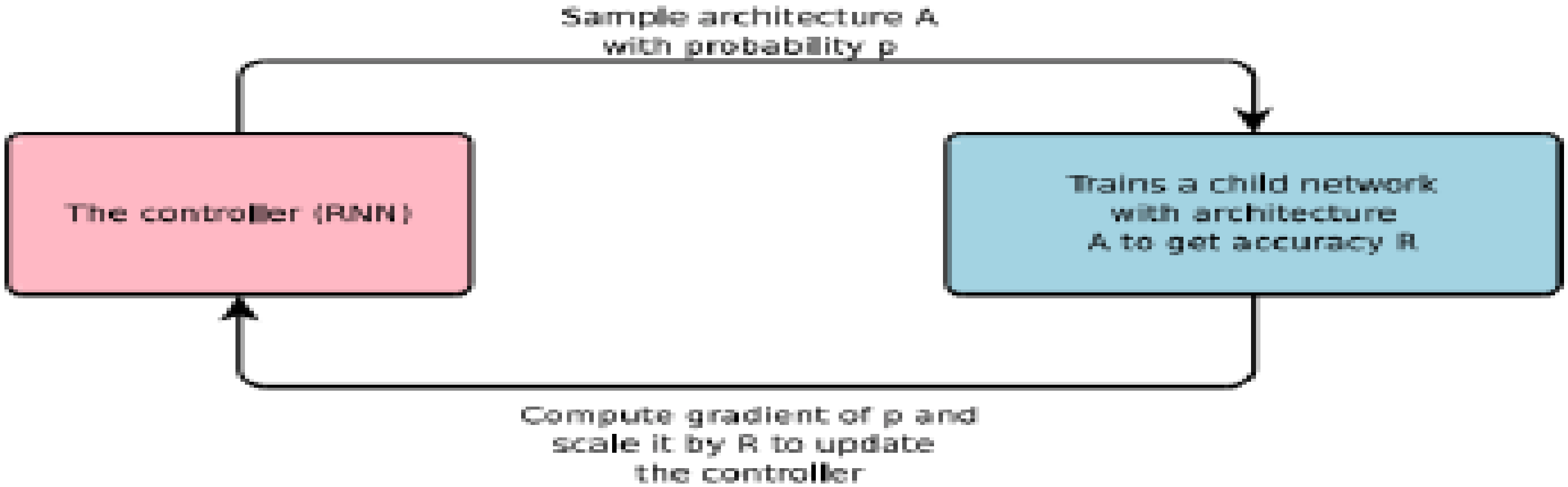
Parameter : $K^2C + CM$

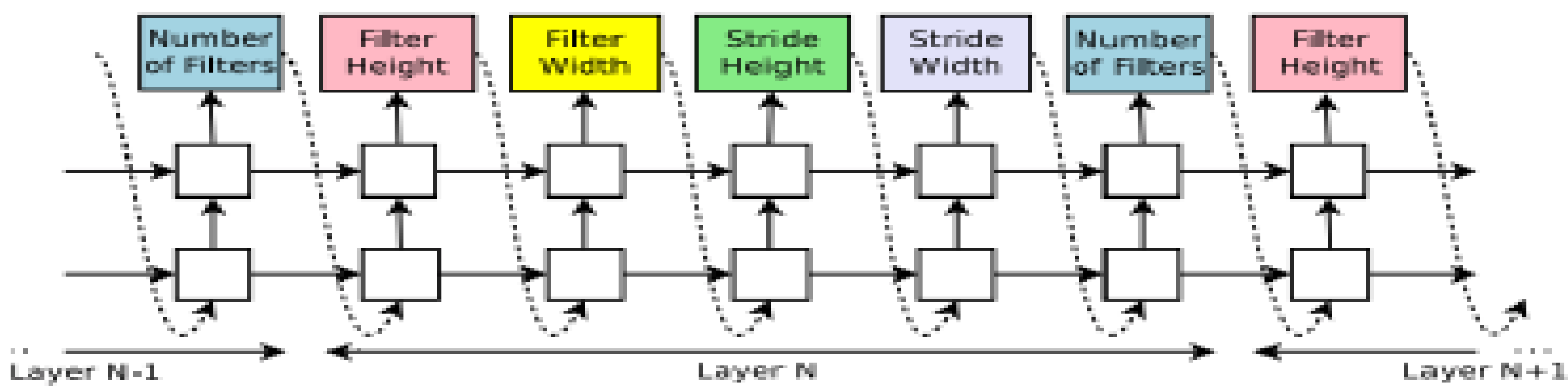
Calculate Cost : $K^2CW'H' + CMW''H''$

(if $W'=W''$, $H'=H''$. $W'H'C(K^2+M)$)

신경망 구조 탐색

신경망 구조 탐색 (NAS: Neural Architecture Search)





전이 학습

전이학습 (Transfer Lerning)

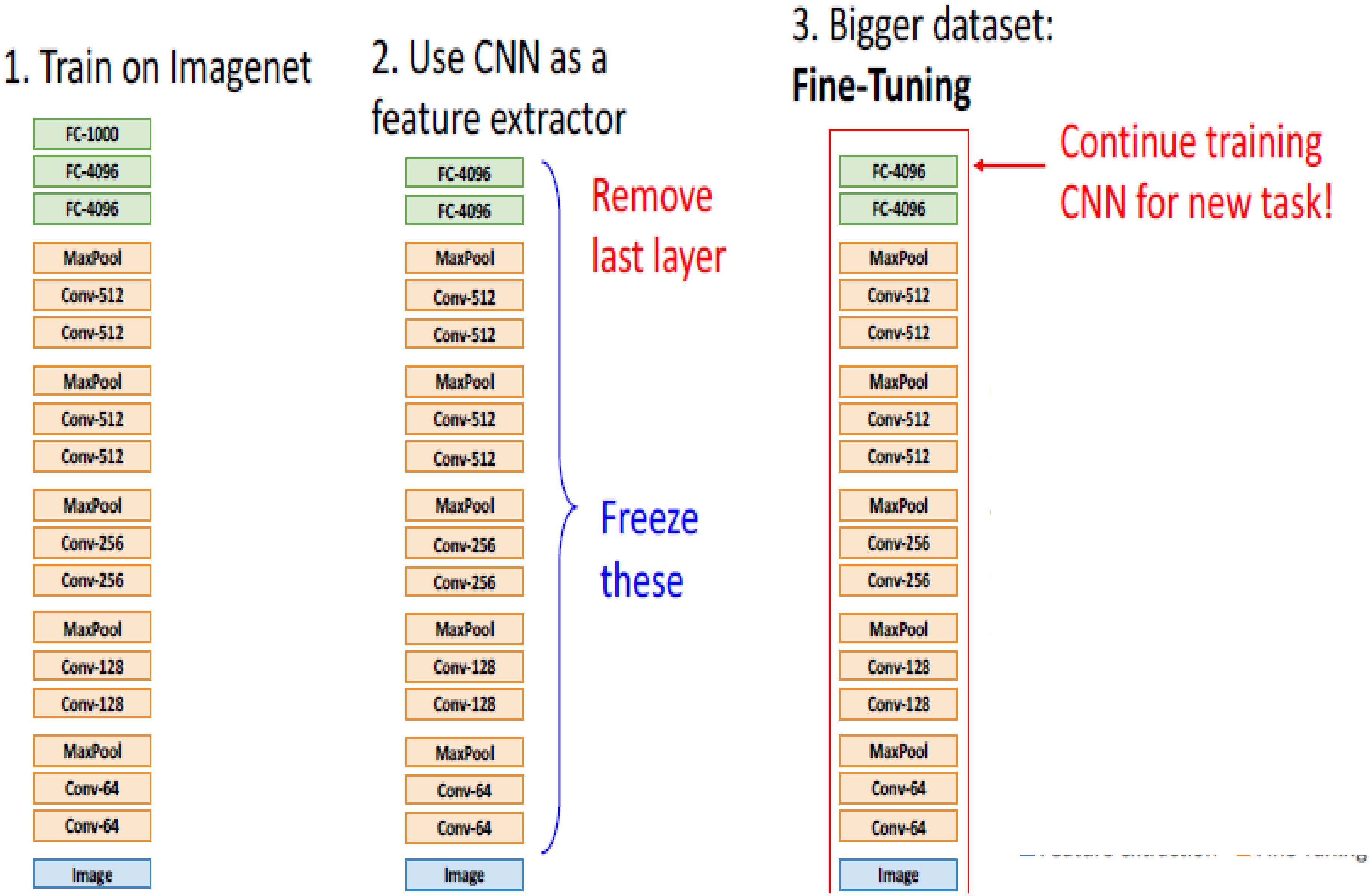
CNN의 학습과 사용을 위해서는
엄청난 데이터가 필요하다.

전이학습 (Transfer Lerning)

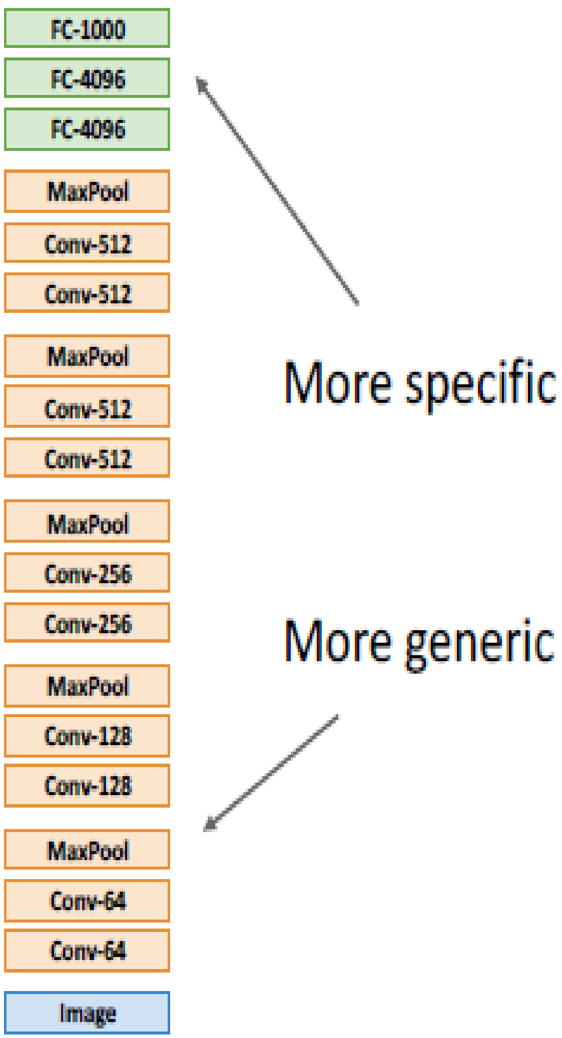
CNN의 학습과 사용을 위해서는
엄청난 데이터가 필요하다.

아니다.

전이학습 (Transfer Lerning) - CNN



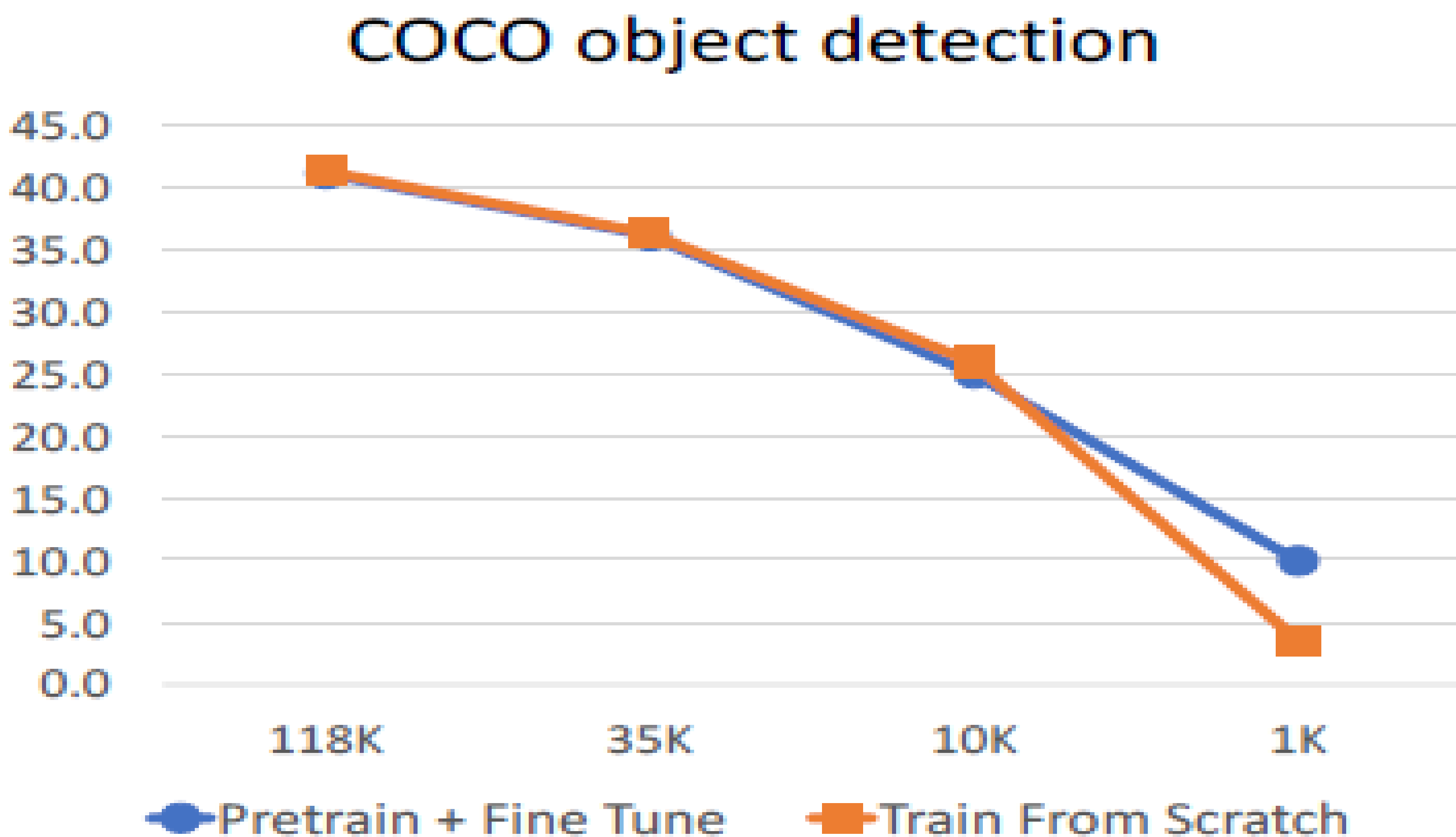
전이학습 (Transfer Lerning) - CNN



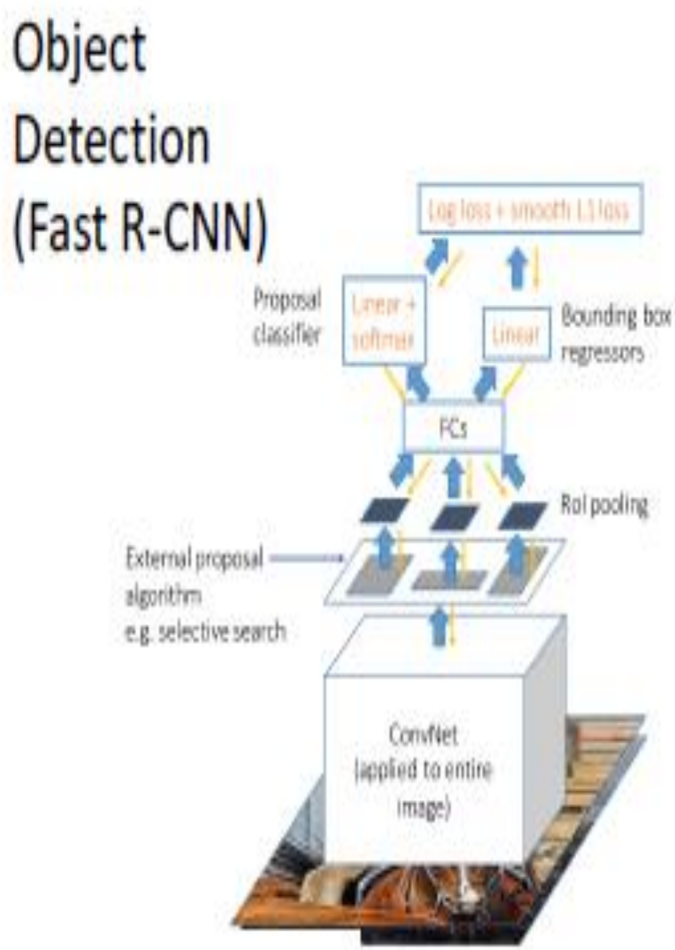
	Dataset similar to ImageNet	**Dataset very different from ImageNet**
very little data (10s to 100s)	Use Linear Classifier on top layer	You're in trouble… Try linear classifier from different stages
quite a lot of data (100s to 1000s)	Finetune a few layers	Finetune a larger number of layers

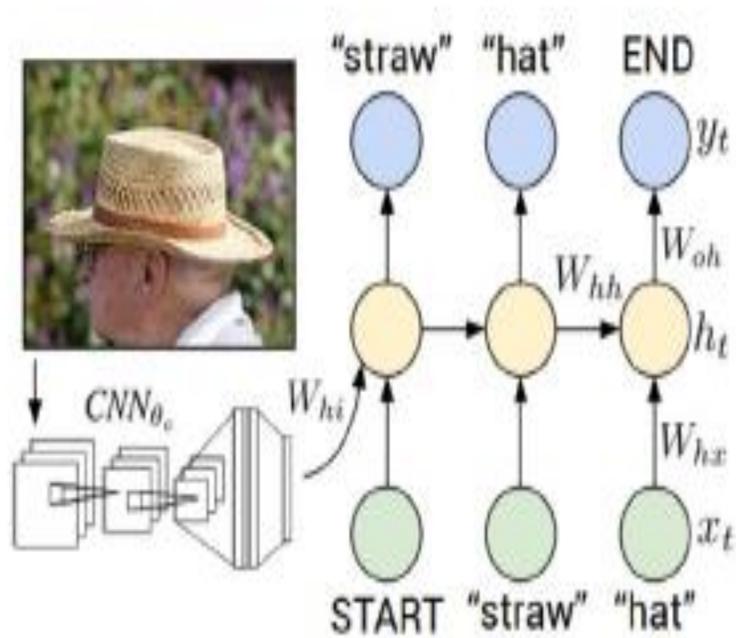
전이학습 (Transfer Lerning)는 기본이지 예외가 아니다.

데이터 크기가작을 때는 사전학습+전이학습이 처음부터 학습하는 것의 성과를 추월한다.

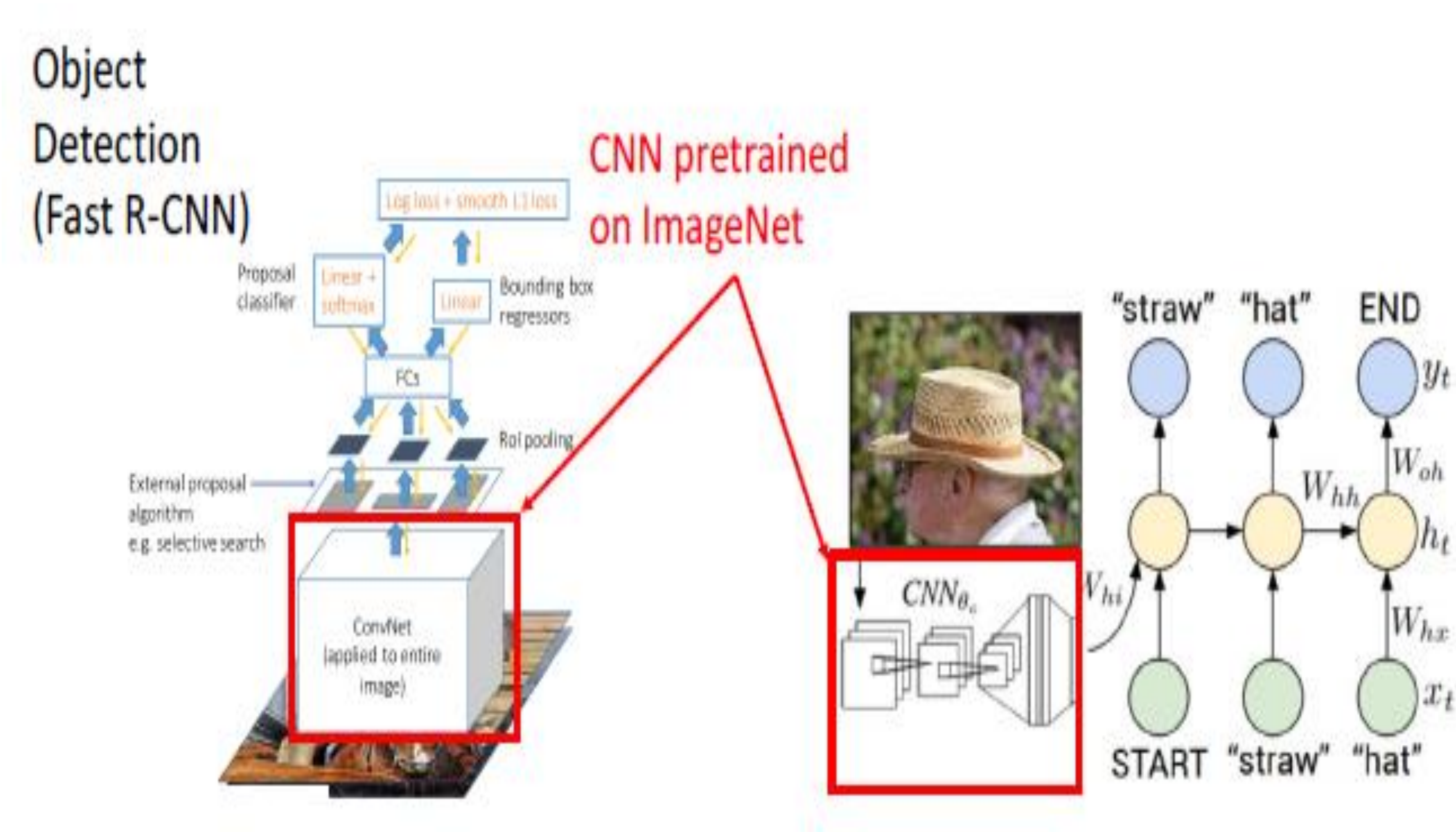


전이학습 (Transfer Lerning)는 기본이지 예외가 아니다.





전이학습 (Transfer Lerning)는 기본이지 예외가 아니다.



전이학습 (Transfer Lerning)는 기본이지 예외가 아니다.