



2023 D&A

Deep Session 6차시

CNN 심화 II

(GoogleNet, ResNet)



CONTENTS

/ 01

Review

- LeNet
- AlexNet
- VGG

/ 02

GoogleNet

- Intro
- Overall Architecture
- GoogleNet의 구조
- Outro

/ 03

ResNet

- Intro
- Overall Architecture
- ResNet의 구조
- Outro

/ 04

정리



Review

LeNet, AlexNet, VGG

■ LeNet

- 최초의 CNN 모델

■ AlexNet

- 227x227 size, RGB 3 channel
- 병렬구조
- LRN(Local Response Normalization)

■ VGG

- 3x3 filter 사용



CONTENTS

/ 01

Review

- LeNet
- AlexNet
- VGG

/ 02

GoogleNet

- Intro
- Overall Architecture
- GoogleNet의 구조
- Outro

/ 03

ResNet

- Intro
- Overall Architecture
- ResNet의 구조
- Outro

/ 04

정리



GoogleNet(Going deeper with Convolutions)

Intro – GoogleNet의 철학

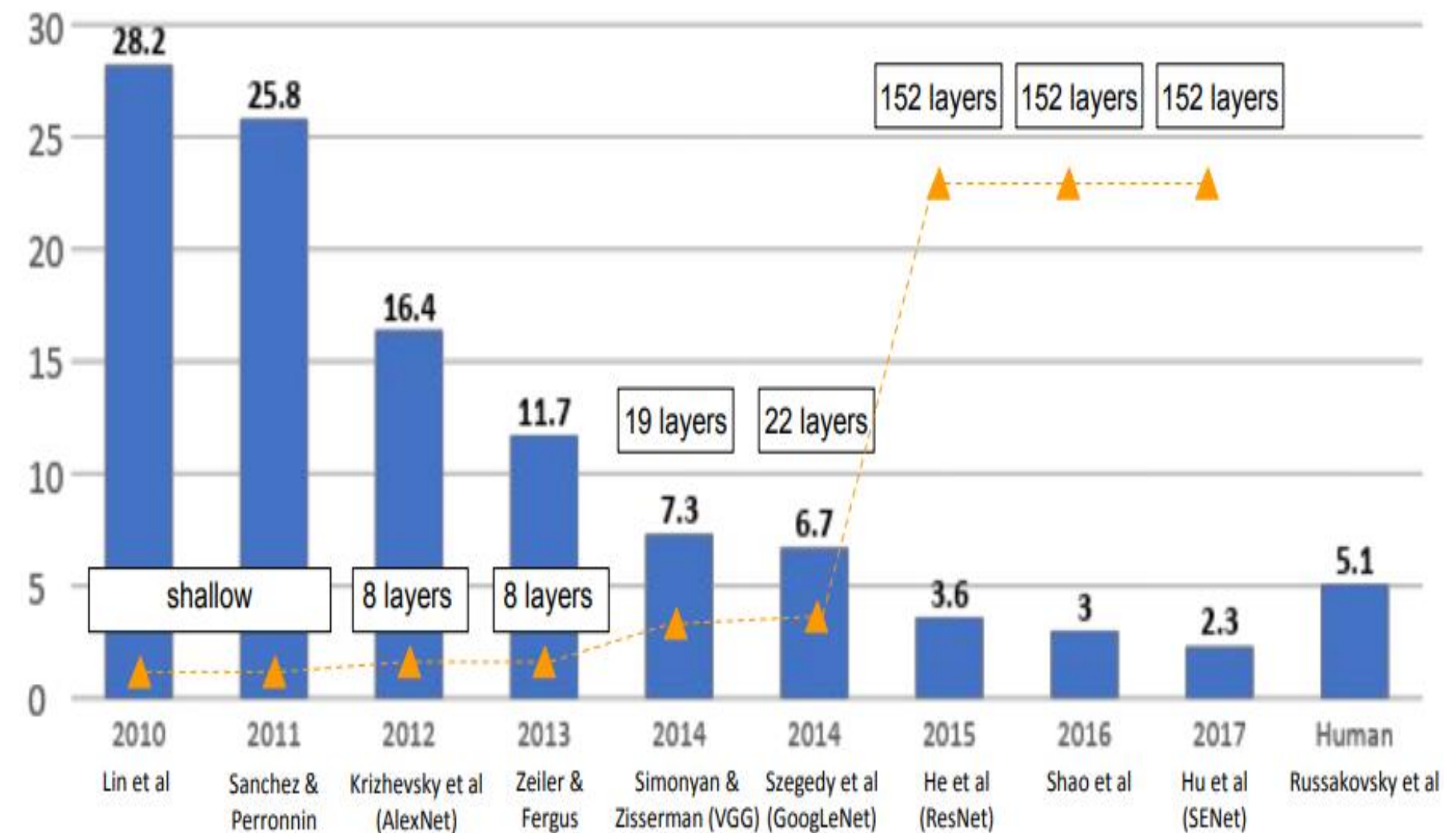
■ 딥러닝의 성능을 높이는 방법은 무엇일까?

1. layer를 깊게(depth)
2. width를 증가(channel)

■ 네트워크의 증가로 인한 문제점

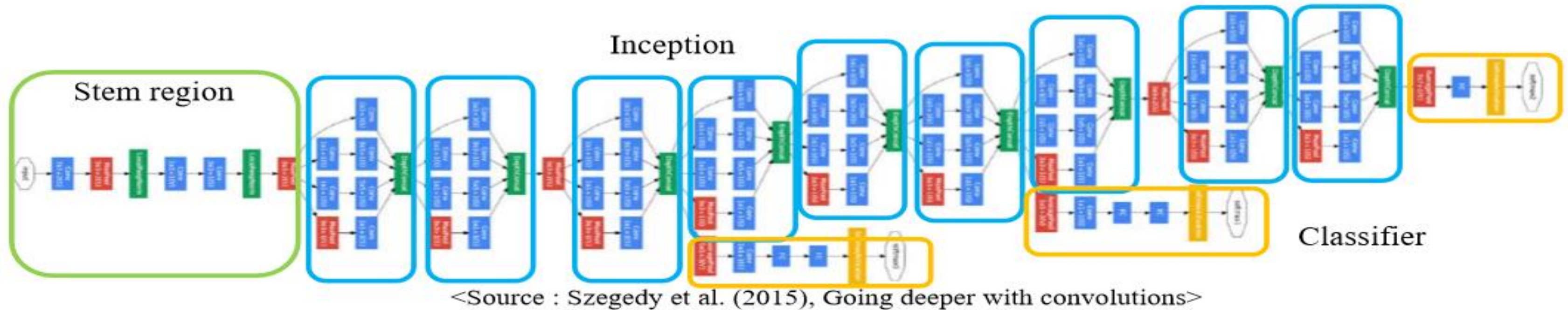
1. Overfitting 발생 -> 학습 데이터에 과하게 학습됨
2. Gradient vanishing 발생

=> layer를 깊게 쌓되, 효율적으로 깊게 쌓자!



GoogleNet

Overall Architecture



주요 특징들

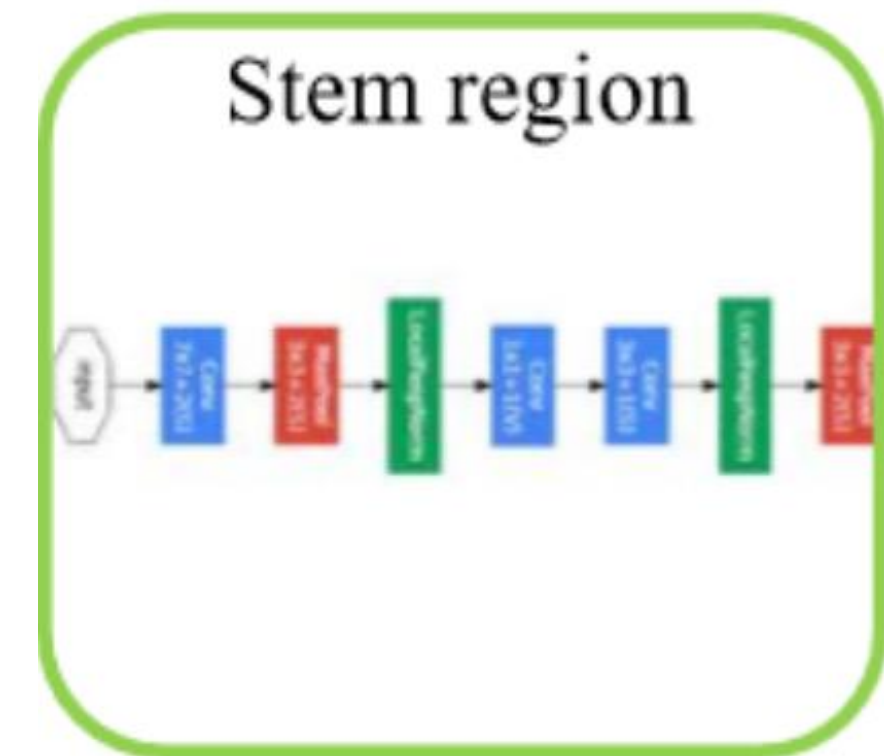
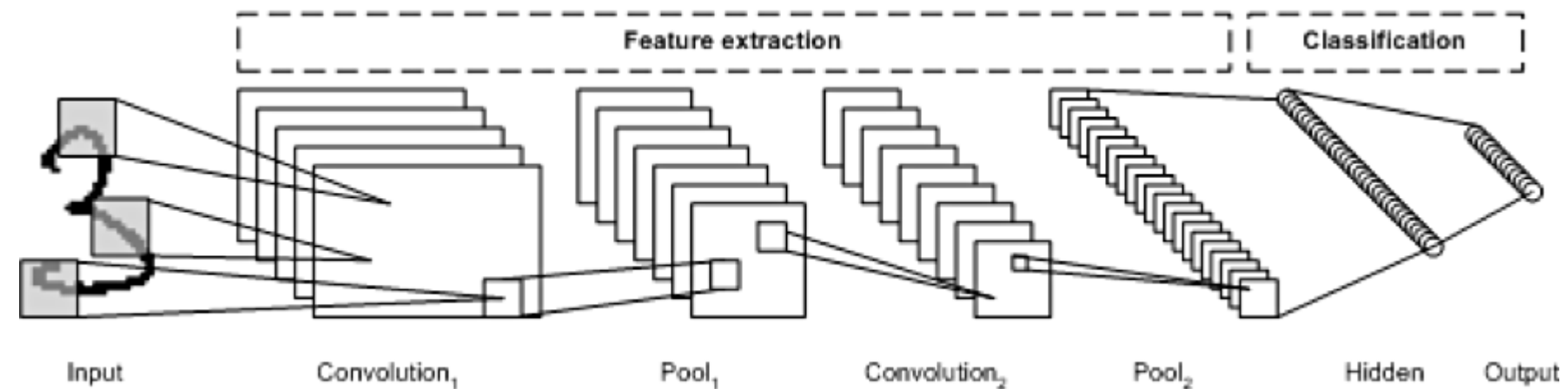
1. 기본 CNN 사용
2. 9개의 inception - 차원 축소 1x1 conv layer 적용
3. Auxiliary classifier(보조 분류기)
4. Global Average Pooling

GoogleNet

GoogleNet의 구조 – level 1

■ 기본적인 CNN 구조 활용

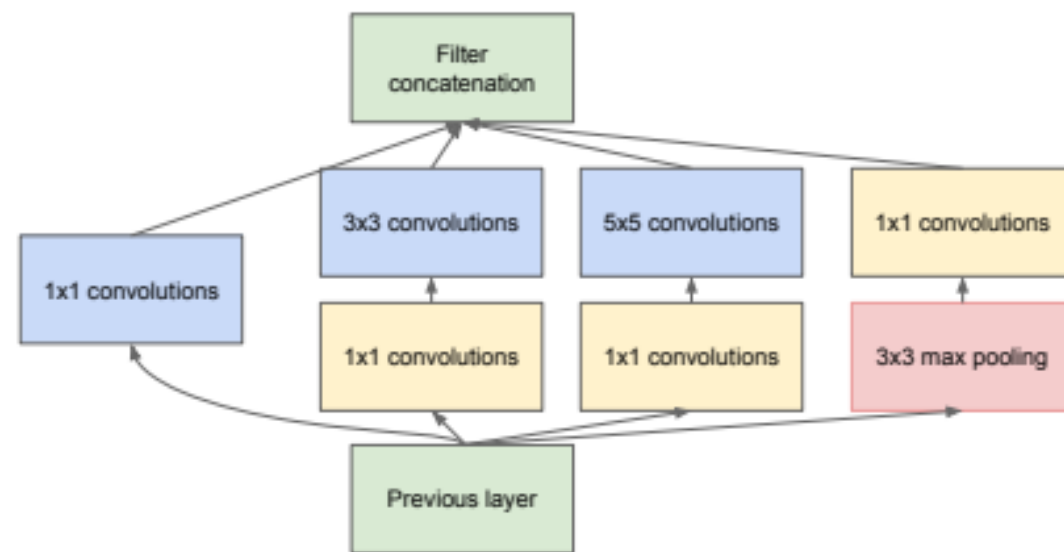
- 효율적인 메모리 사용을 위해 낮은 layer에서는 기본 CNN 구조 사용
- Conv -> pooling -> Conv -> pooling



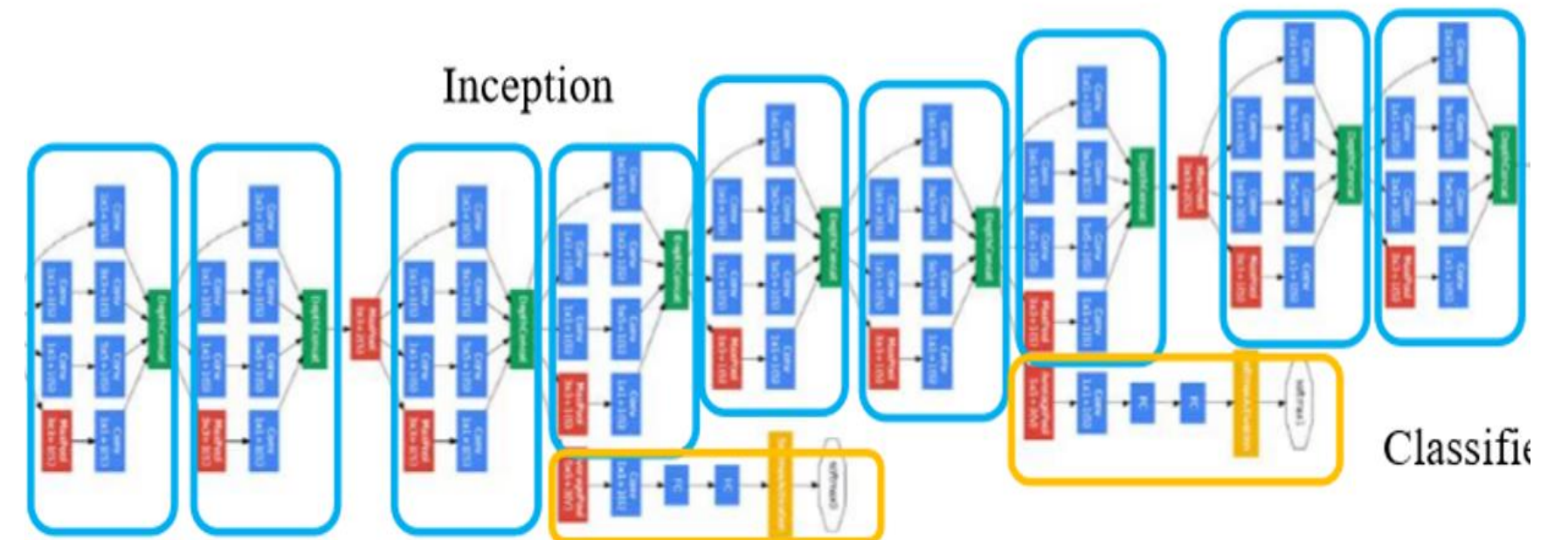
GoogleNet

GoogleNet의 구조 – level2

9개의 inception module



(b) Inception module with dimension reductions

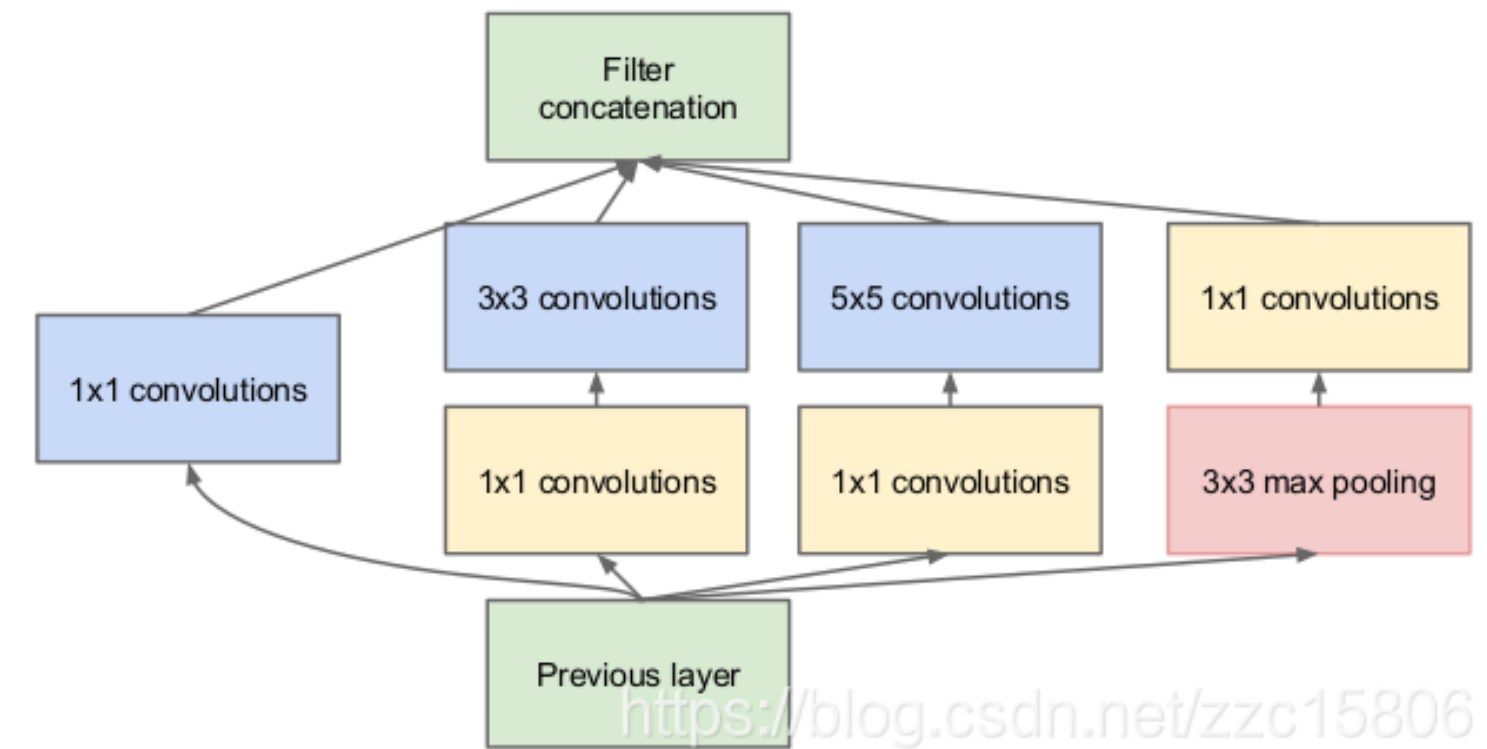


GoogleNet

GoogleNet의 구조 – level2

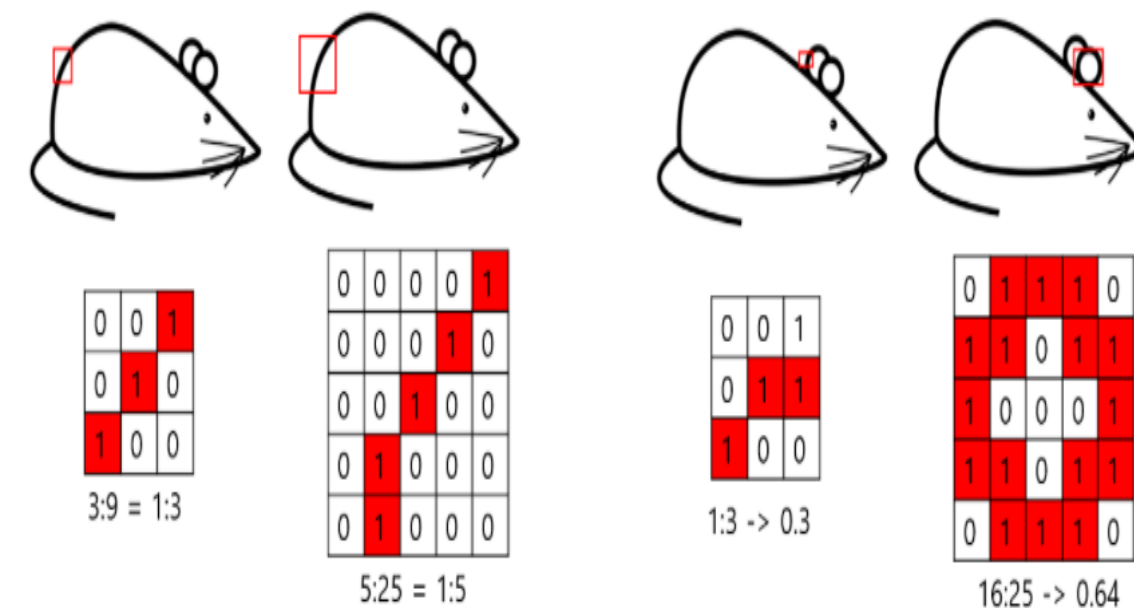
1. 다양한 Conv layer의 병렬 사용

- 1x1, 3x3, 5x5 layer
- 다양한 layer로 처리하기 때문에 다양한 정보 학습 가능



2. 3x3 max-pooling

- conv network에 필수요소로 생각
- 있으면 더 좋다

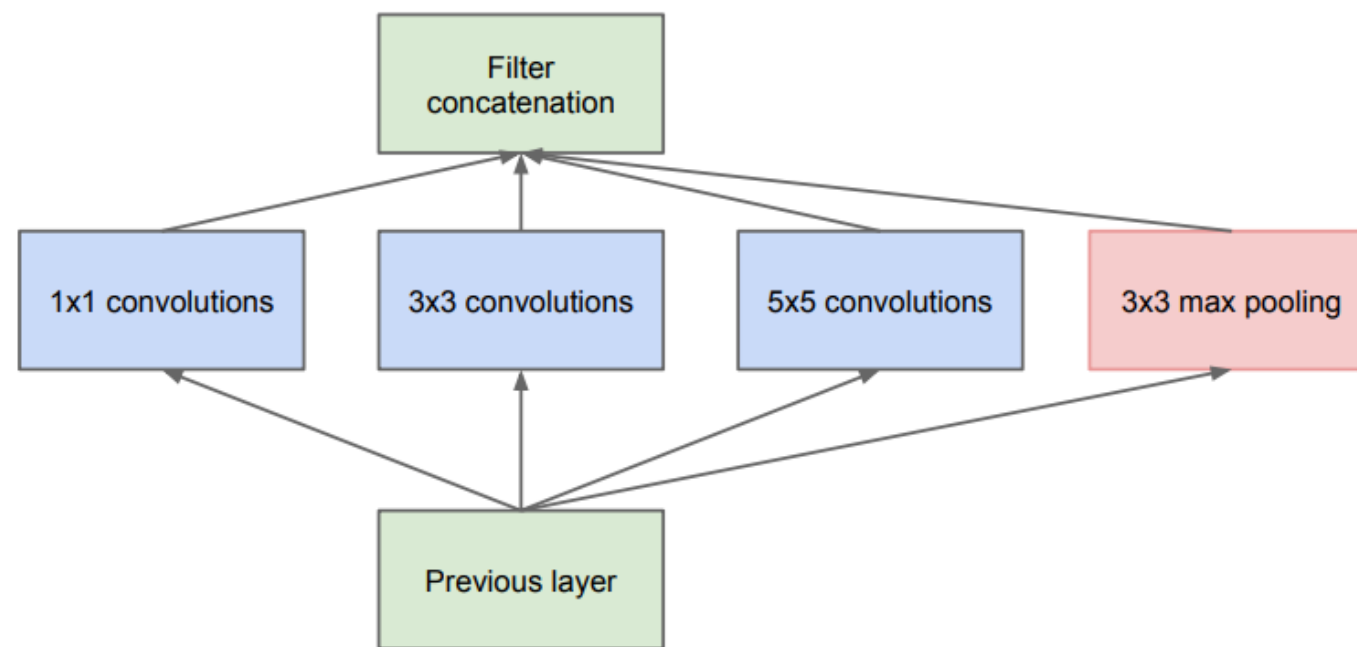


GoogleNet

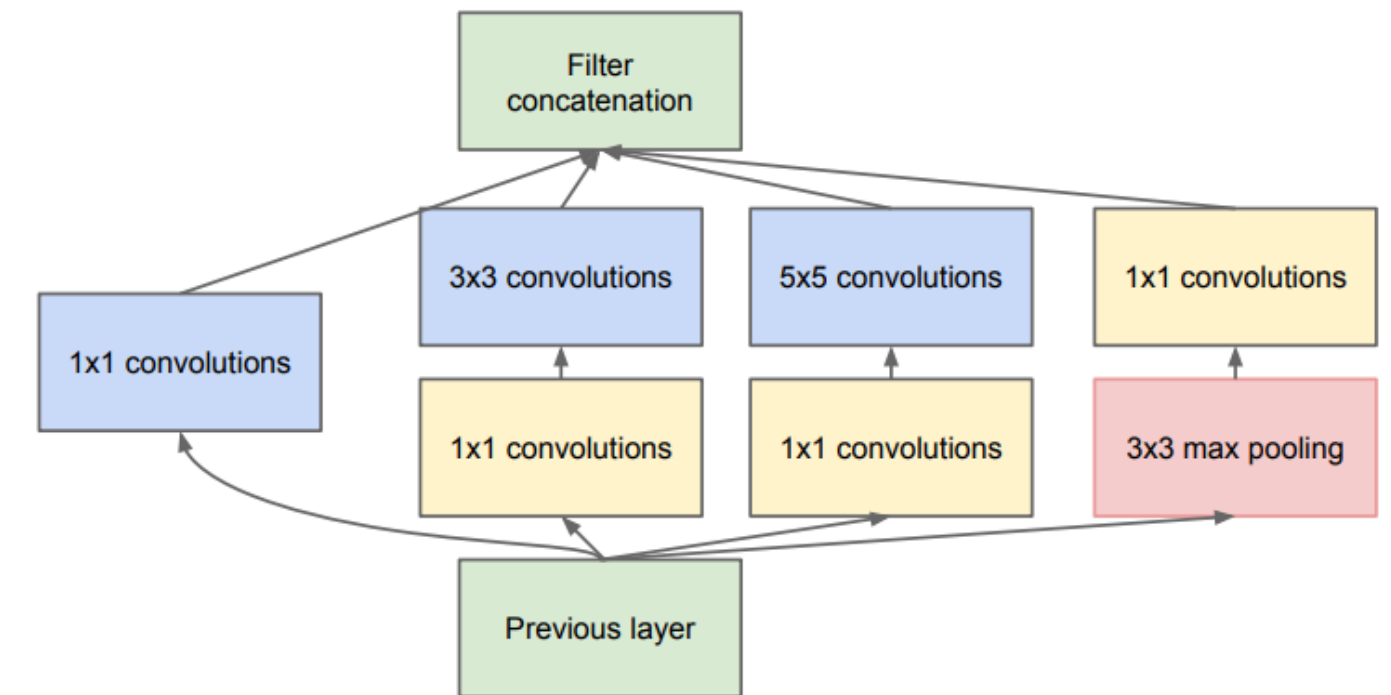
GoogleNet의 구조 – level2

3. 차원 축소 1x1 conv layer

- 연산량 줄이기 위해 사용



(a) Inception module, naïve version



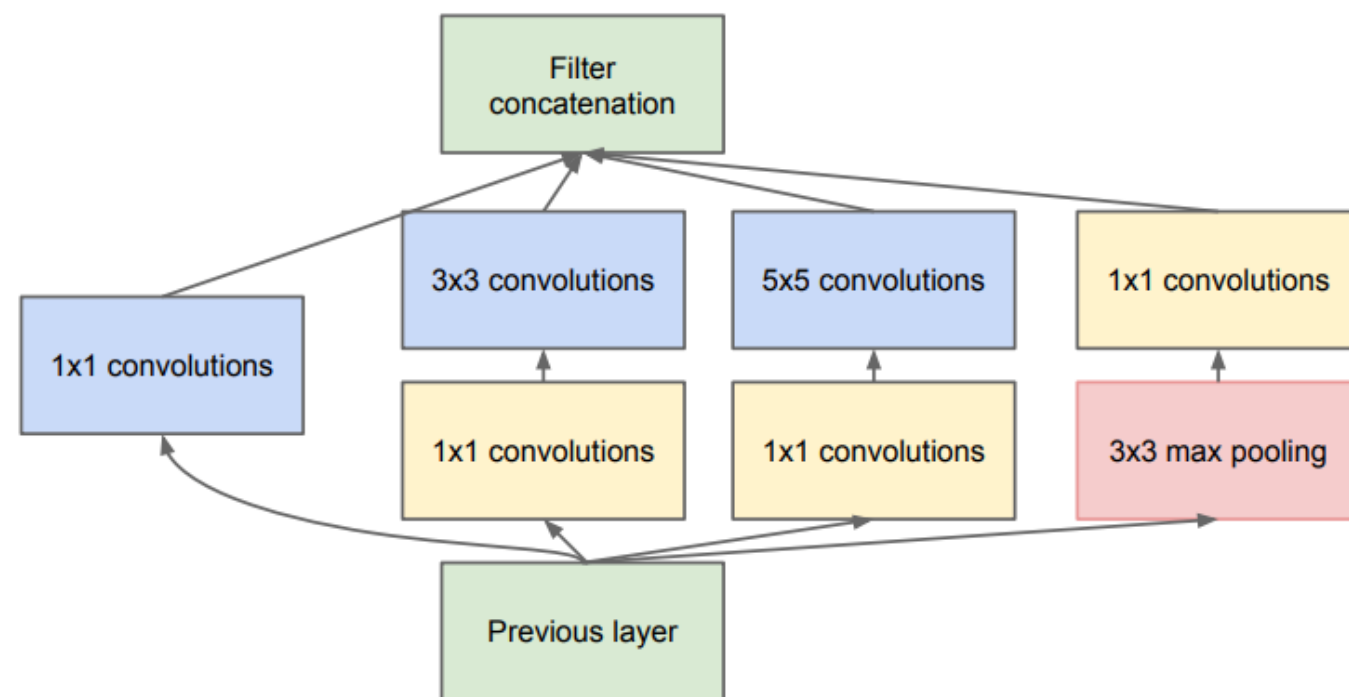
(b) Inception module with dimension reductions

GoogleNet

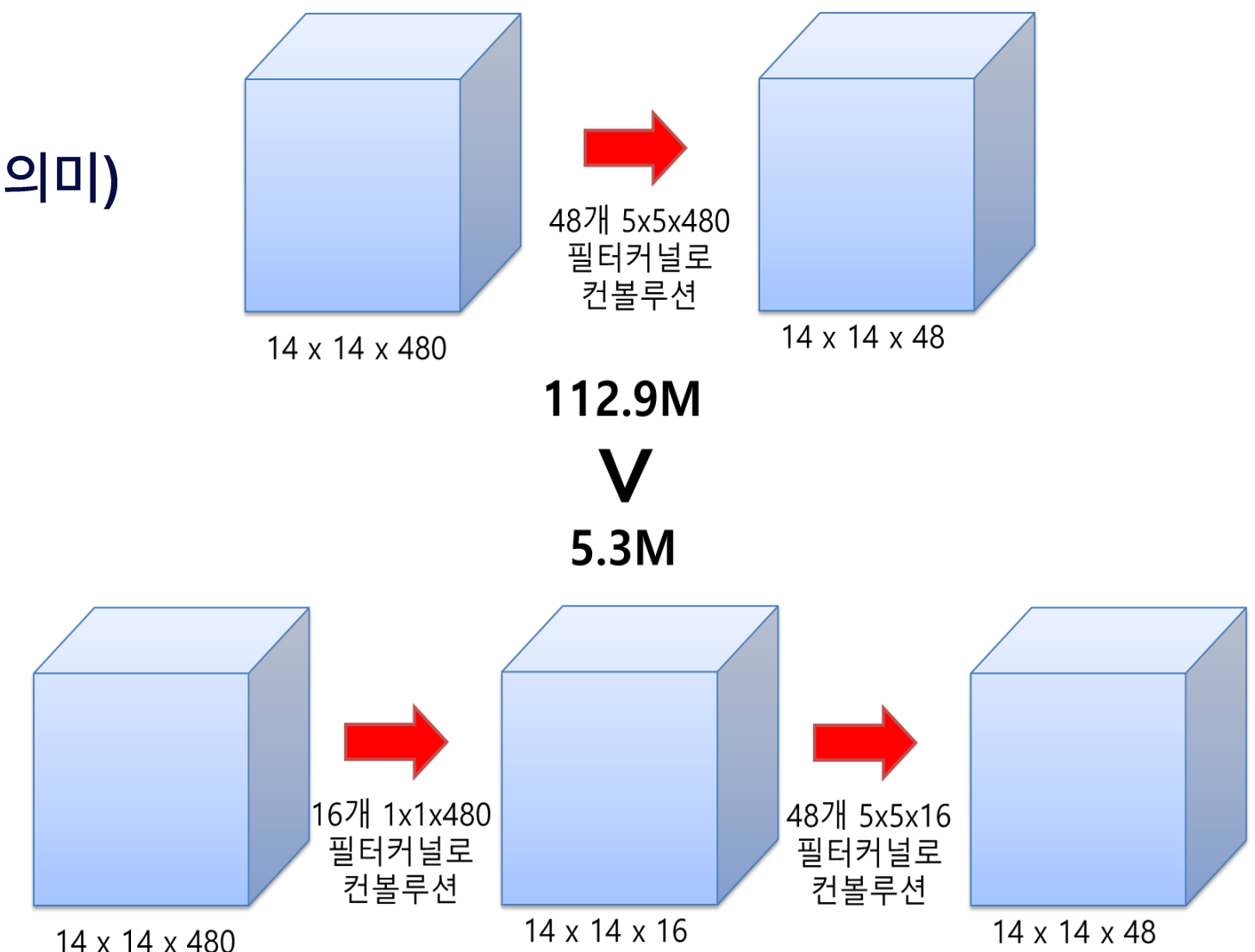
GoogleNet의 구조 – level2

3. 차원 축소 1x1 conv layer

- 연산량 줄이기 위해 사용
- > **Bottle Neck**이라고 표현 (병의 주둥이와 같이 좁은 곳을 통과한다는 의미)



(b) Inception module with dimension reductions

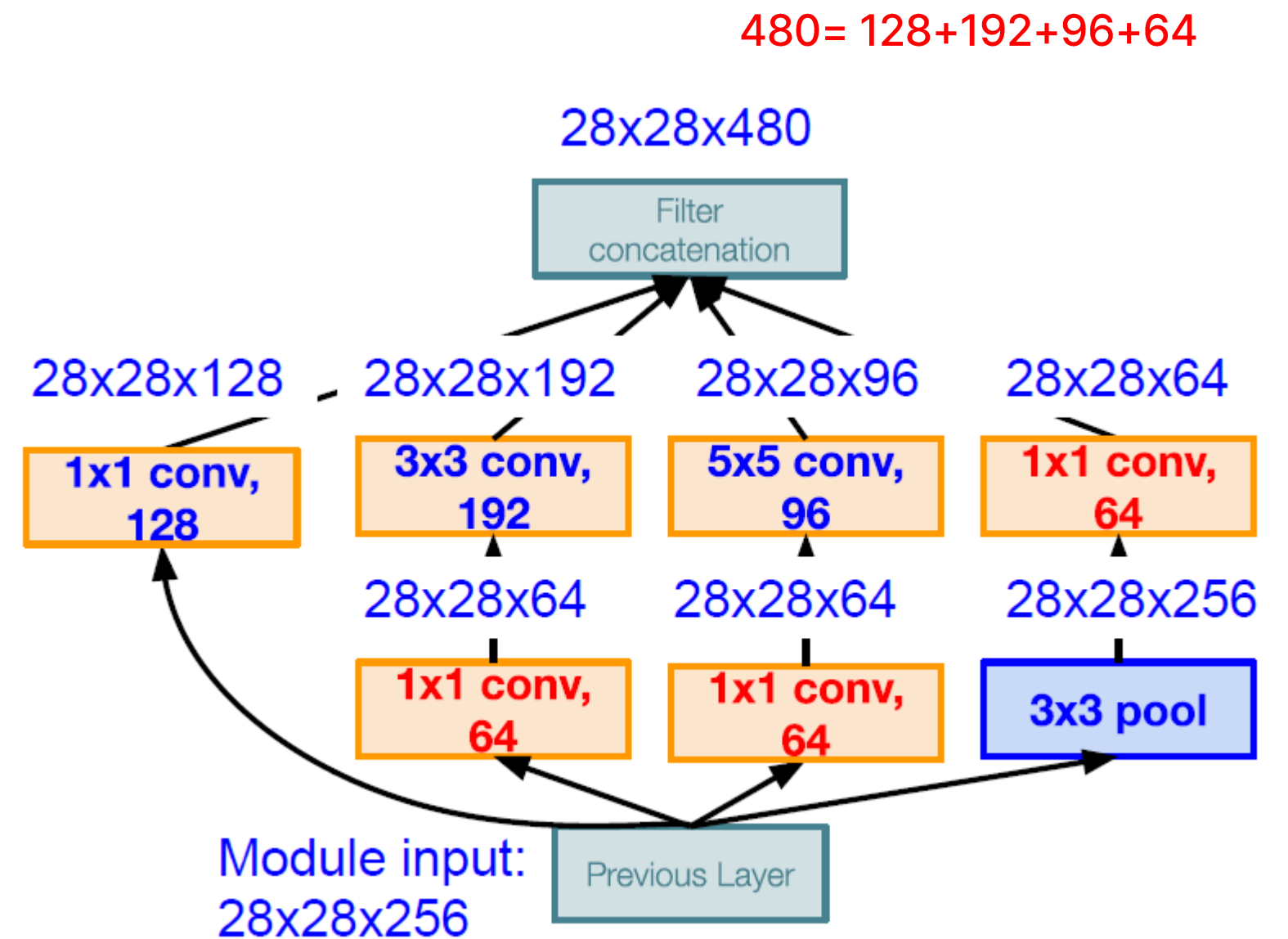
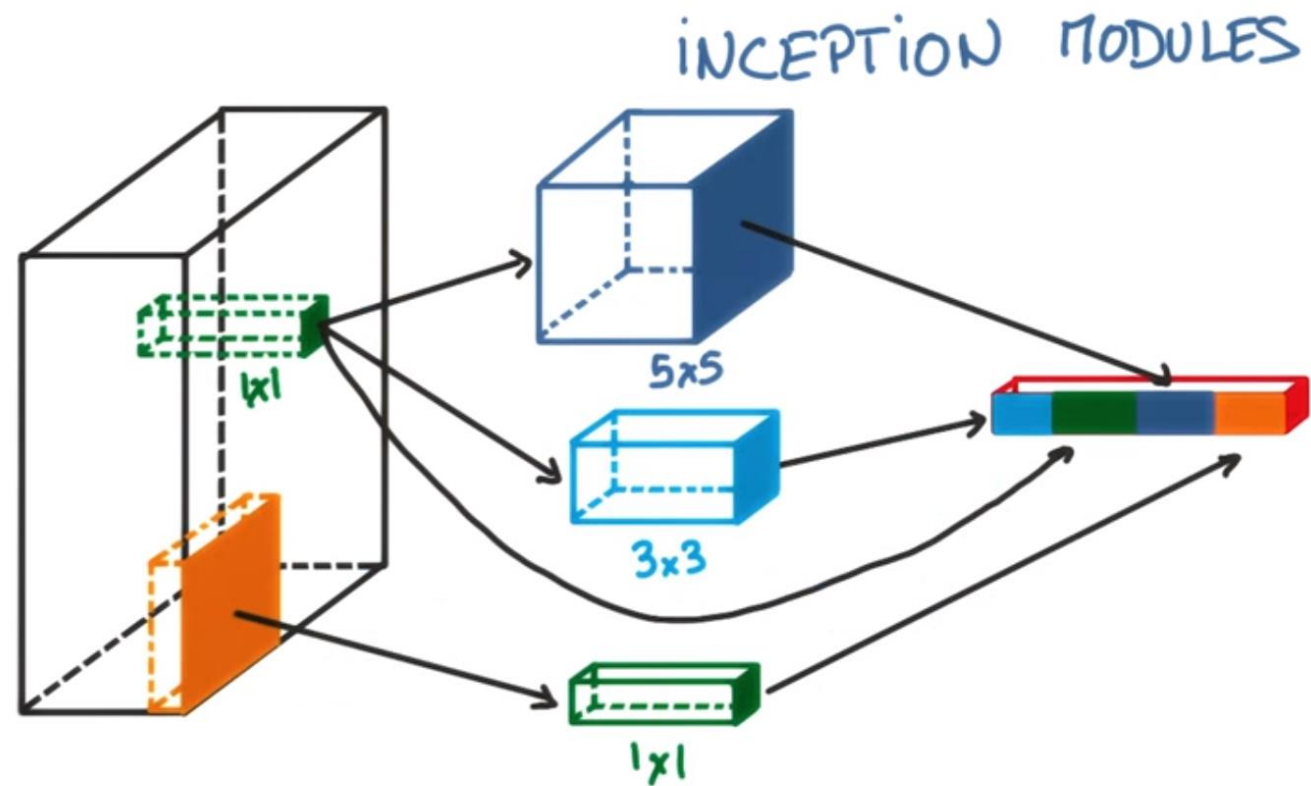


GoogleNet

GoogleNet의 구조 - level2

4. Concat

- 네 개의 특징 맵(output)들을 하나로 **겹쳐주는 역할**
- 각 분기의 결과는 크기를 하나로 통일하게 설계해야 concat이 가능

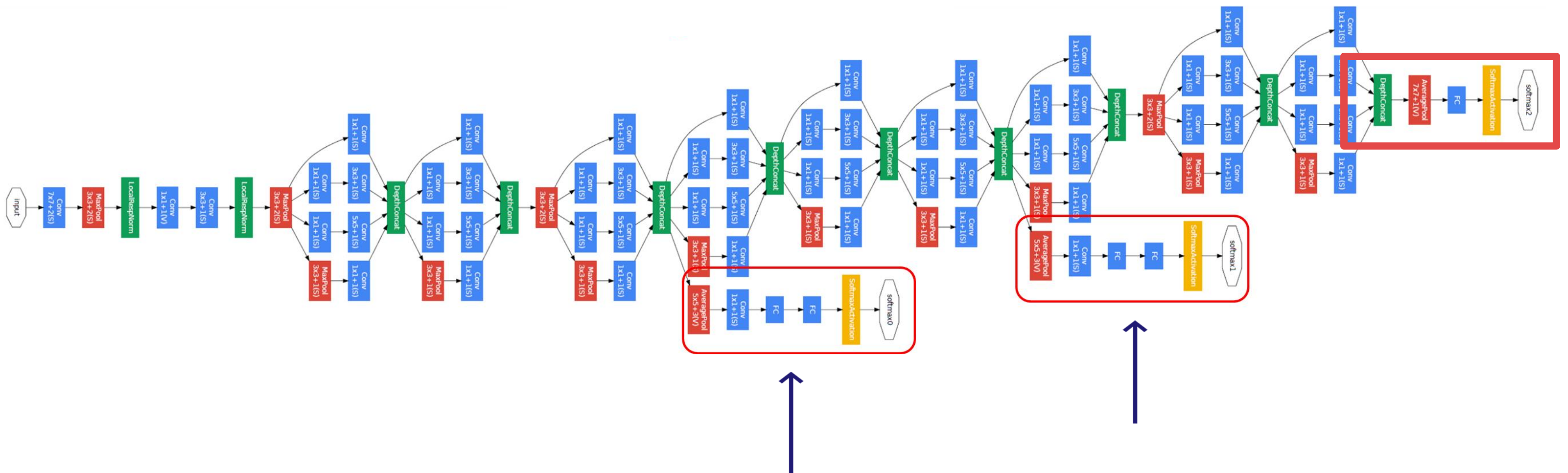


GoogleNet

GoogleNet의 구조 – level 3

Auxiliary classification(보조 분류기)

- 보조 분류기 2개 사용



GoogleNet

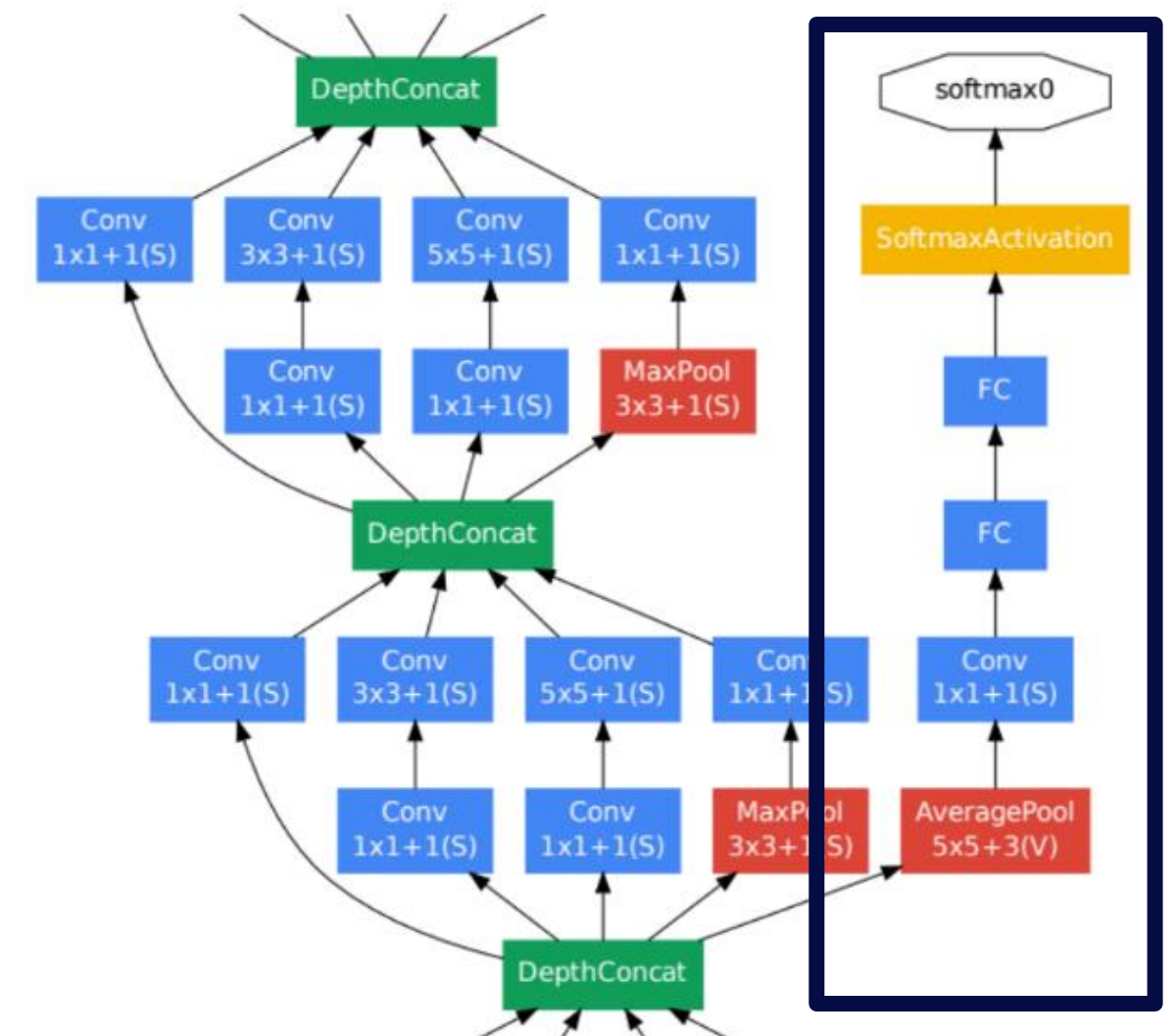
GoogleNet의 구조 – level 3

Auxiliary classification(보조 분류기)

- 기존의 경우, 마지막 층의 softmax만을 통해 gradient를 구했음
- layer가 깊어질수록 학습이 잘 진행되지 않는 Gradient Vanishing(기울기 소실) 문제 발생

⇒ 따라서 layer 중간중간 softmax 함수를 출력하여 gradient 역전파를 진행

*0.3을 곱해서 적용(가중치), test 시에는 제거함

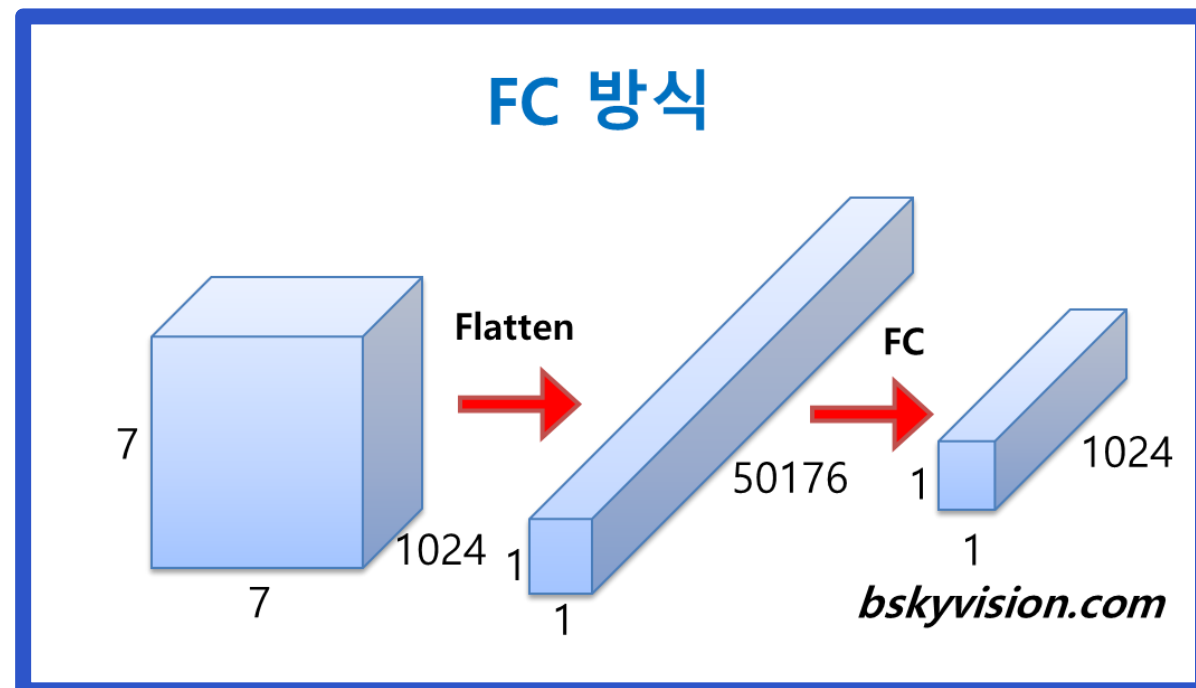


GoogleNet

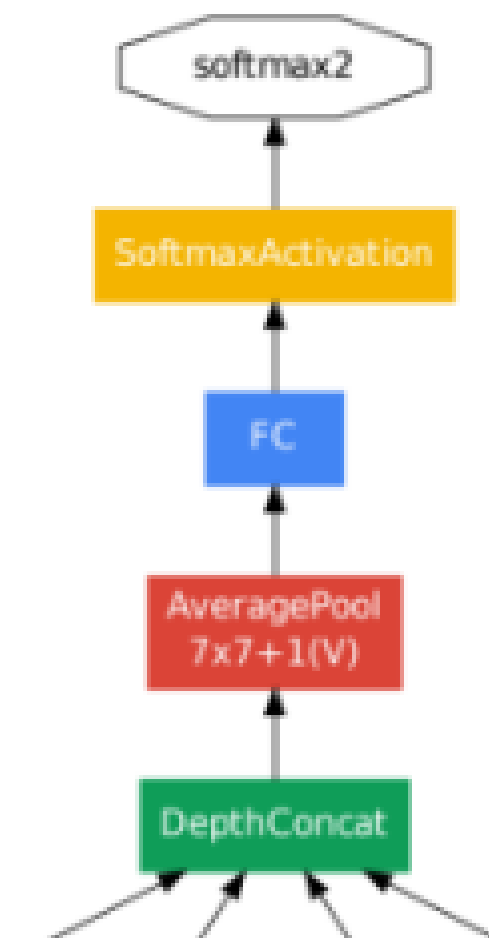
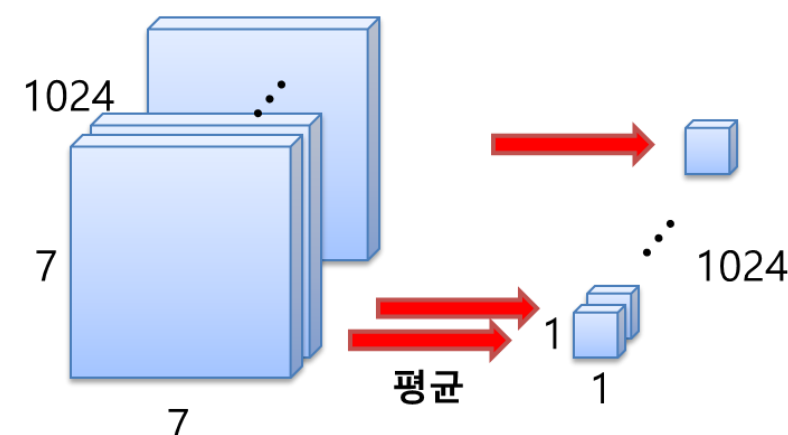
GoogleNet의 구조 – level 4

Global Average Pooling

- 기존의 경우
: 1차원으로 flatten 시킨 후, FC를 거쳐 $1 \times 1 \times 1024$ 의 벡터로 생성
-> 50176×1024 의 연산 필요
* $50176 = 7 \times 7 \times 1024$



Global average pooling

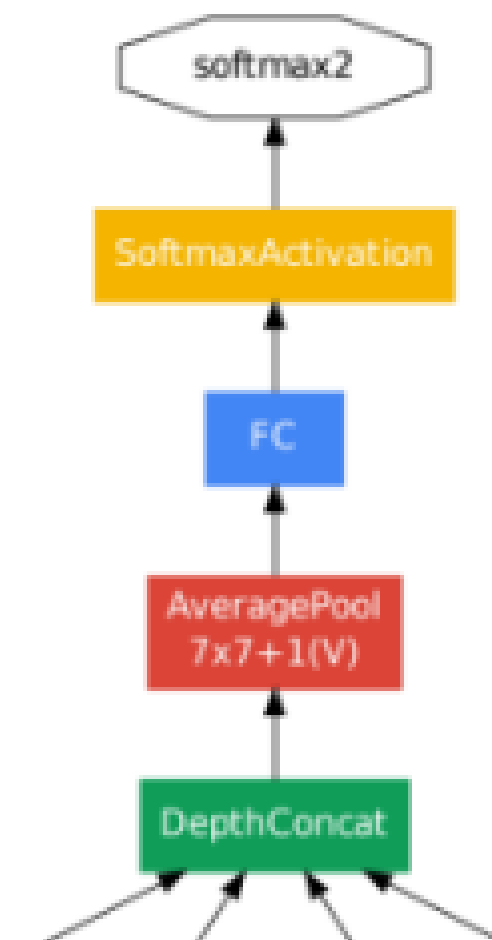
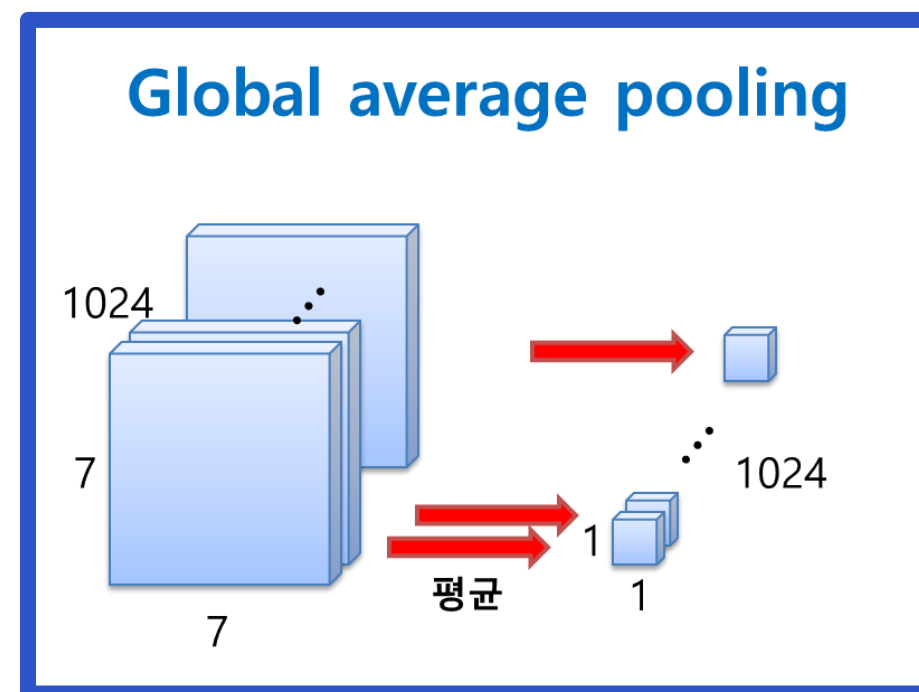
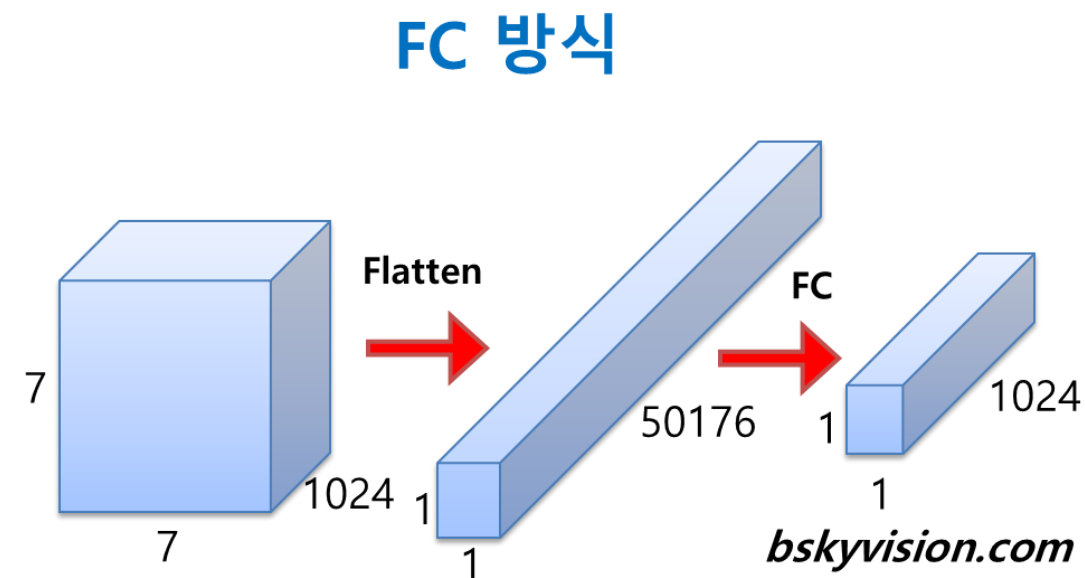


GoogleNet

GoogleNet의 구조 – level 4

Global Average Pooling

- GAP의 경우
: 각각의 특성맵 하나하나를 평균 내어 1024개의 벡터를 연결해줌
-> 평균을 내기 때문에 파라미터 계산이 필요 x

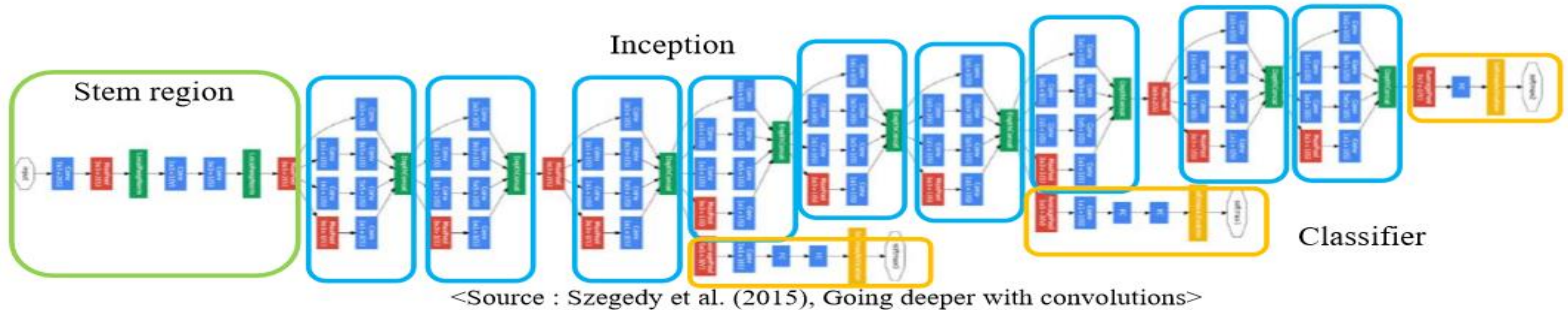


GoogleNet Intro

Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no



GoogleNet Outro



주요 특징들

1. 기본 CNN 사용
2. 9개의 inception- 차원 축소 1x1 conv layer 적용
3. Auxiliary classifier(보조 분류기)
4. Global Average Pooling

CONTENTS

/ 01

Review

- LeNet
- AlexNet
- VGG

/ 02

GoogleNet

- Intro
- Overall Architecture
- GoogleNet의 구조
- Outro

/ 03

ResNet

- Intro
- Overall Architecture
- ResNet의 구조
- Outro

/ 04

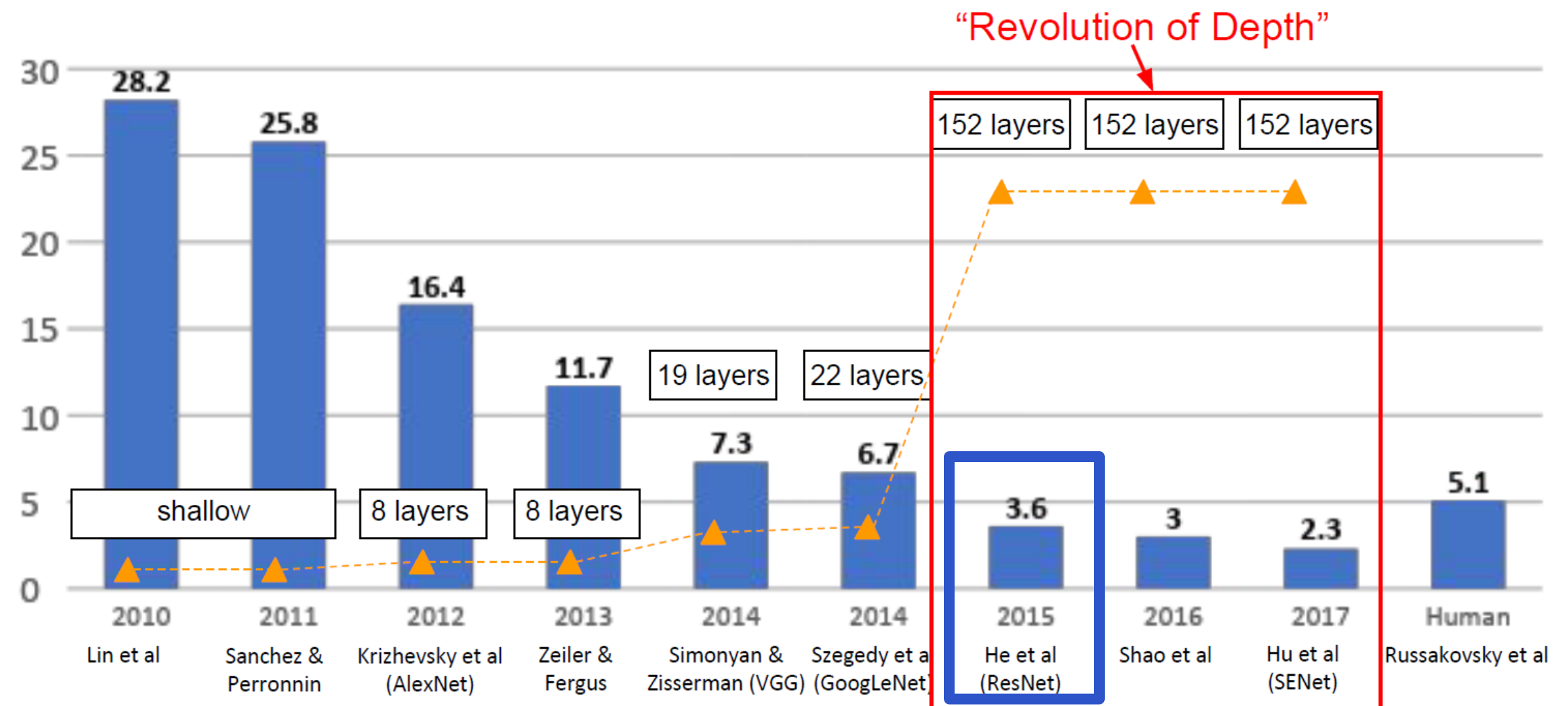
정리



ResNet(Residual neural network)

Intro

- VGG보다 8배 깊은 152개의 layer 사용
- 3.57%의 error를 보여줌
- ILSVRC 2015에서 1위 차지함
- ImageNet detection, ImageNet localization, COCO detection, COCO segmentation에서도 1위를 차지



ResNet

Intro

■ 깊이가 깊어질수록 네트워크의 성능은 무조건 향상될까?

- Layer가 증가할수록 오히려 더 높은 error 발생
- > gradient vanishing, overfitting, 연산량 증가 등의 문제
- > 연산이 진행되면서 원래 정보를 잃게 됨

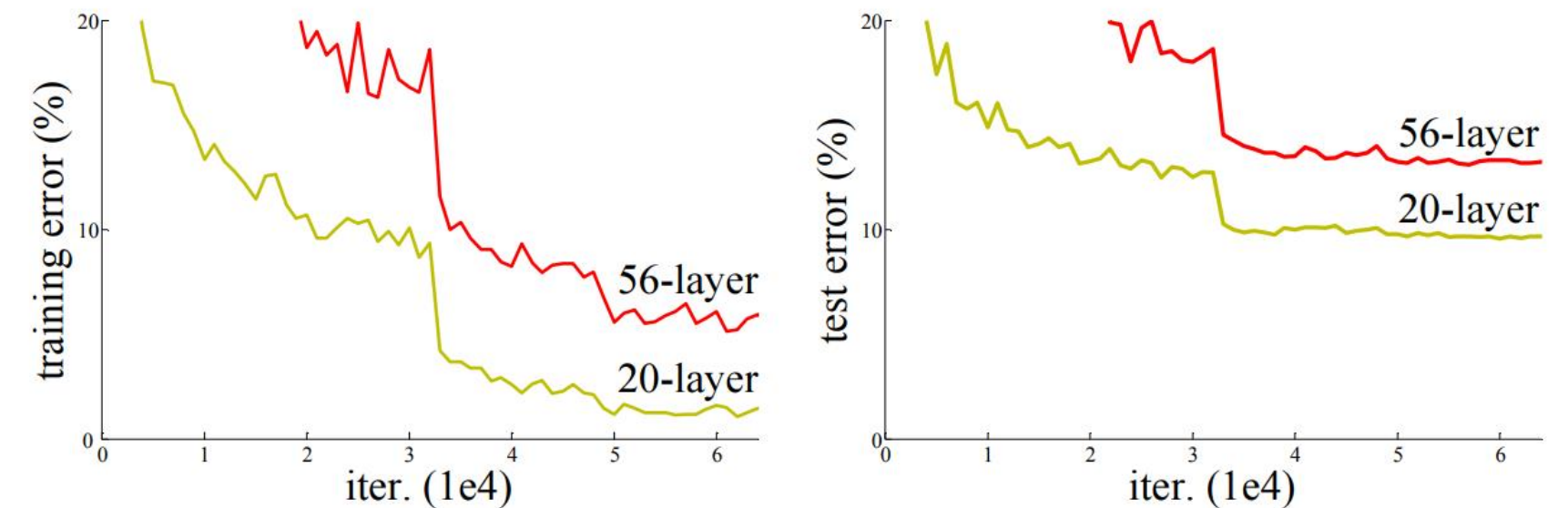


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

- ⇒ Layer는 깊게 하되, 학습하는 양을 줄이자!
- ⇒ Layer를 거친 값에 x 를 더해줘 기존 정보를 살리자!

ResNet

Overall Architecture

주요 특징

- Residual Learning
- Short cut(skip) connection

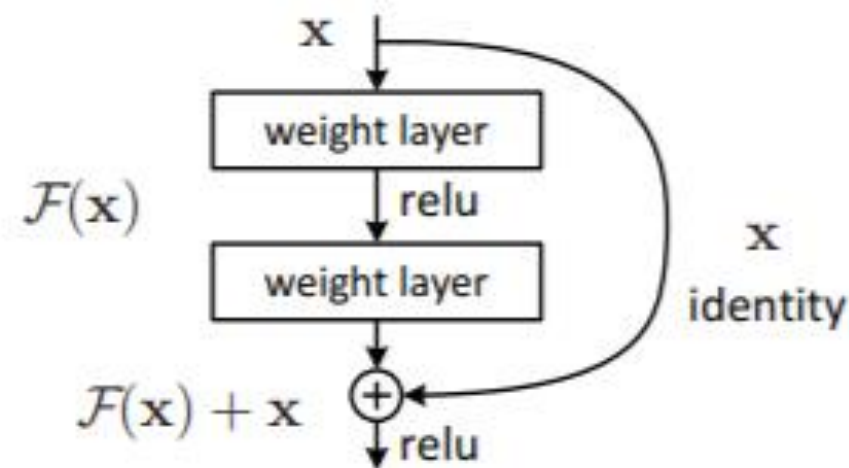
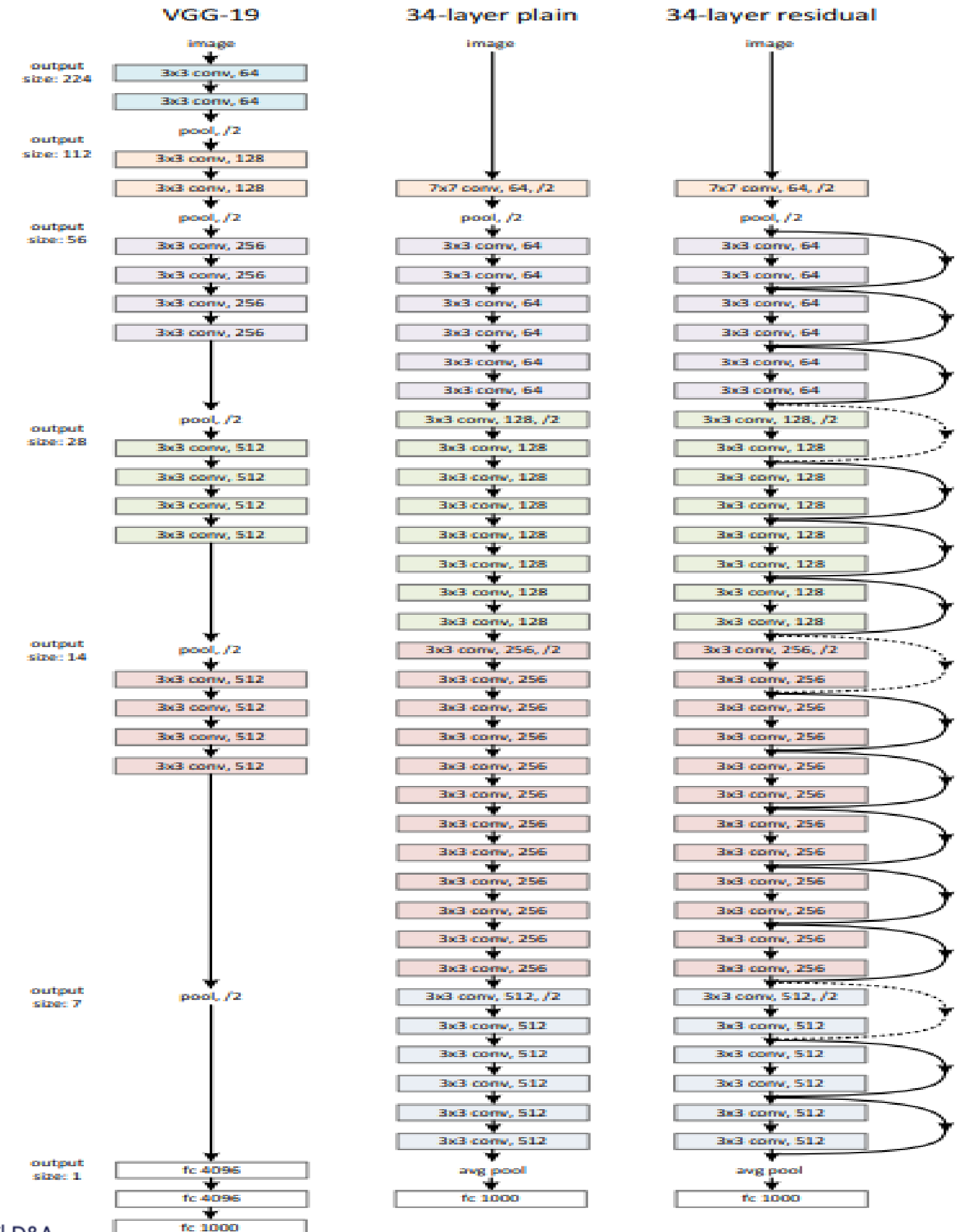


Figure 2. Residual learning: a building block.



ResNet

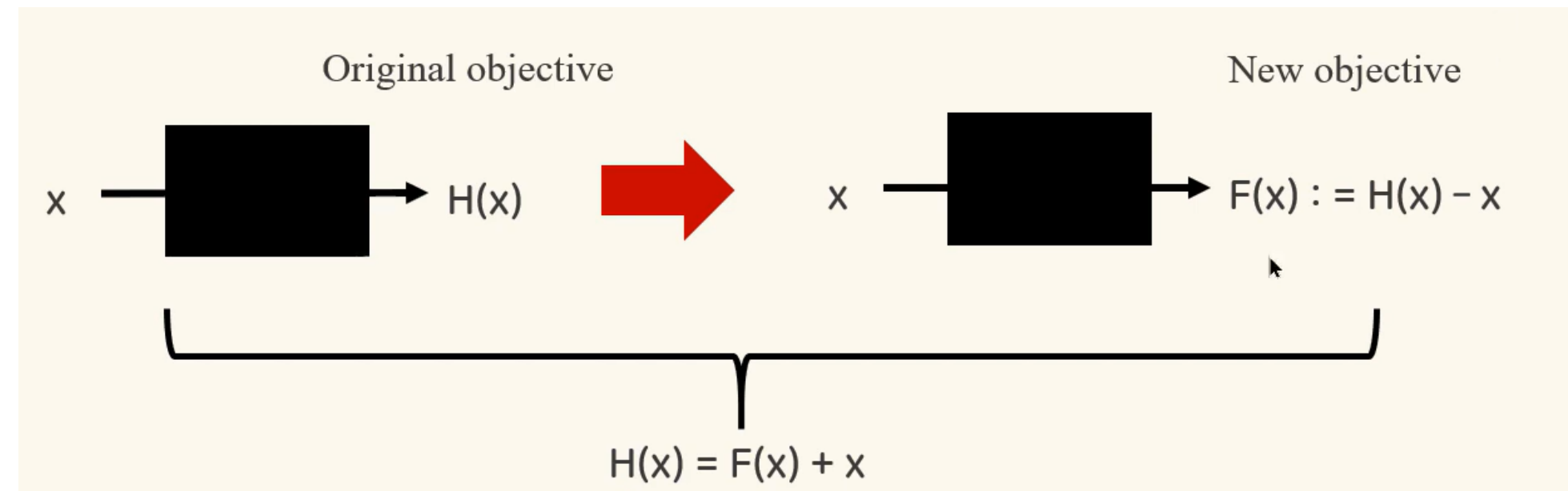
ResNet의 구조 - 잔차 학습

Residual Learning

1. Plain layers

- Layer가 깊어짐에 따라, 정보 손실이 발생

⇒ 최적 $H(x)$ 값을 구하기가 어려움



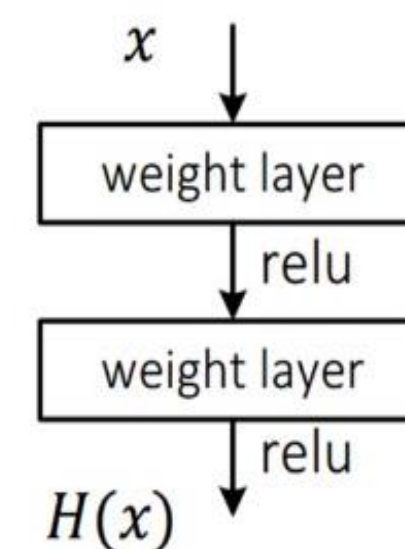
2. Residual block

- $F(x) = H(x) - x$
- 앞서 학습된 기존 정보(x)에 잔여한 정보($F(x)$)를 더하자! (정보 손실 방지)
- $F(x)$ 만 추가적으로 학습하면 됨 (연산량 감소)
- $H'(x) = F'(x) + 1$ 이므로 적어도 기울기가 1로 유지됨 (기울기 소실 문제 해소)

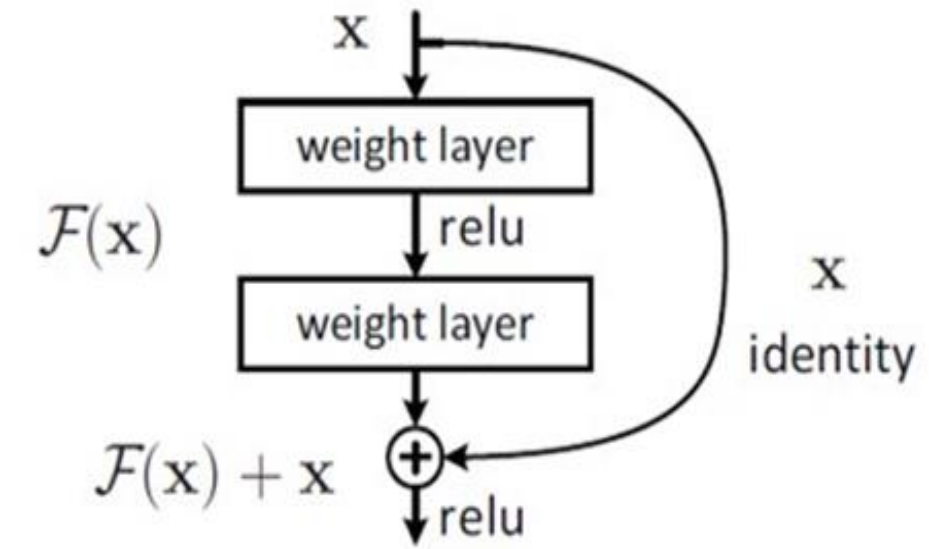
⇒ $F(x)$ 를 $H(x) - x$ 로 변형시켜 $H(x)$ 에 근사하도록 학습시키는 것이 목표

⇒ $F(x)$ 가 0에 가까워지도록 학습시킴 (잔차의 개념)

Plain layers



Residual block



ResNet

ResNet의 구조 – Short cut connection

Short cut(skip) connection

- 두 개의 conv layer마다 화살표가 나오는 것을 확인할 수 있음
- 레이어를 뛰어 넘어서 값 전달
- Shortcut(skip) 구조라고 표현

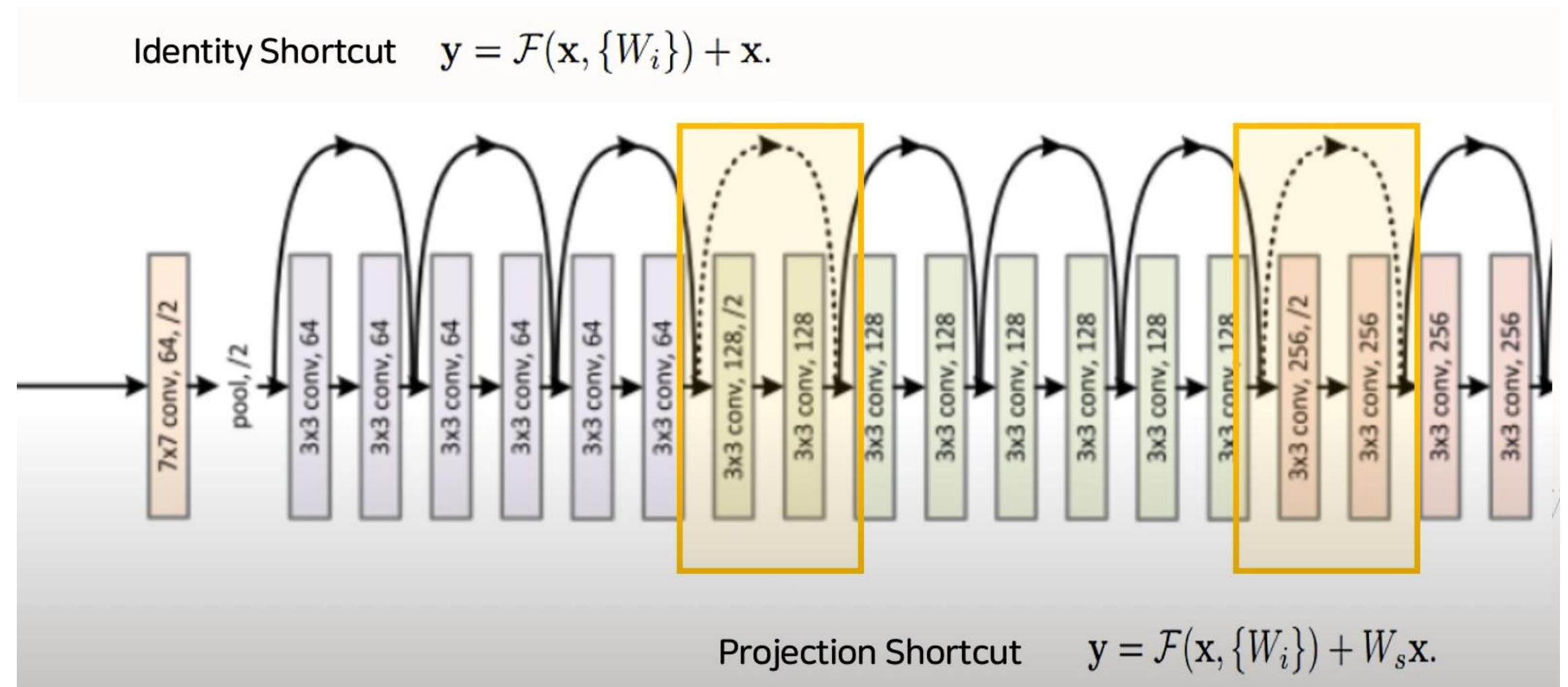
1) Identity Shortcut(일반적인 구조)

$$\Rightarrow y = F(x, W_i) + x$$

2) Projection Shortcut

$$\Rightarrow y = F(x, W_i) + W_s x$$

* (W_s 는 size 맞춰주는 행렬)



ResNet Intro

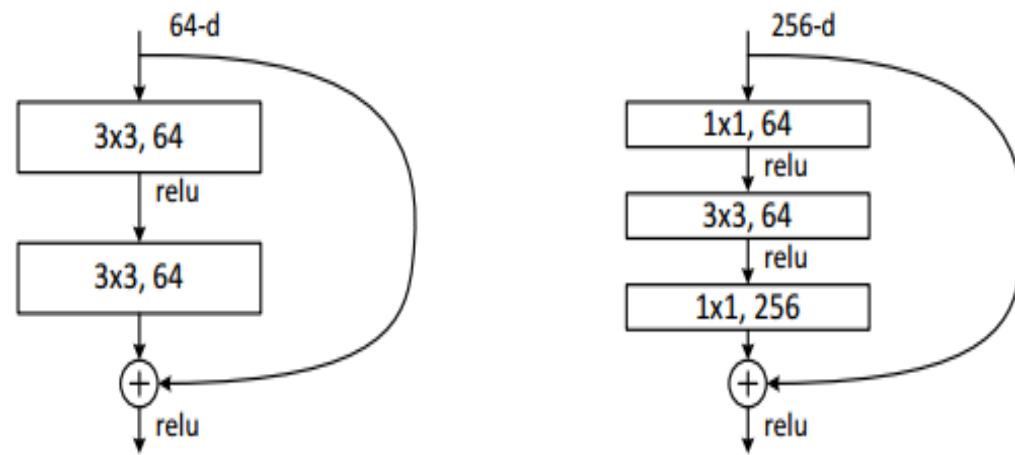


Figure 5. A deeper residual function \mathcal{F} for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a “bottleneck” building block for ResNet-50/101/152.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	$7 \times 7, 64, \text{stride } 2$				
		$3 \times 3 \text{ max pool, stride } 2$				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

- Layer가 50개 이상이 될 때 1x1 conv를 통해 연산량을 줄여줌
- **Bottle Neck 구조** (병의 주둥이와 같이 좁은 곳 통과한다는 의미)
- * GoogleNet의 inception module 과 유사

ResNet Intro

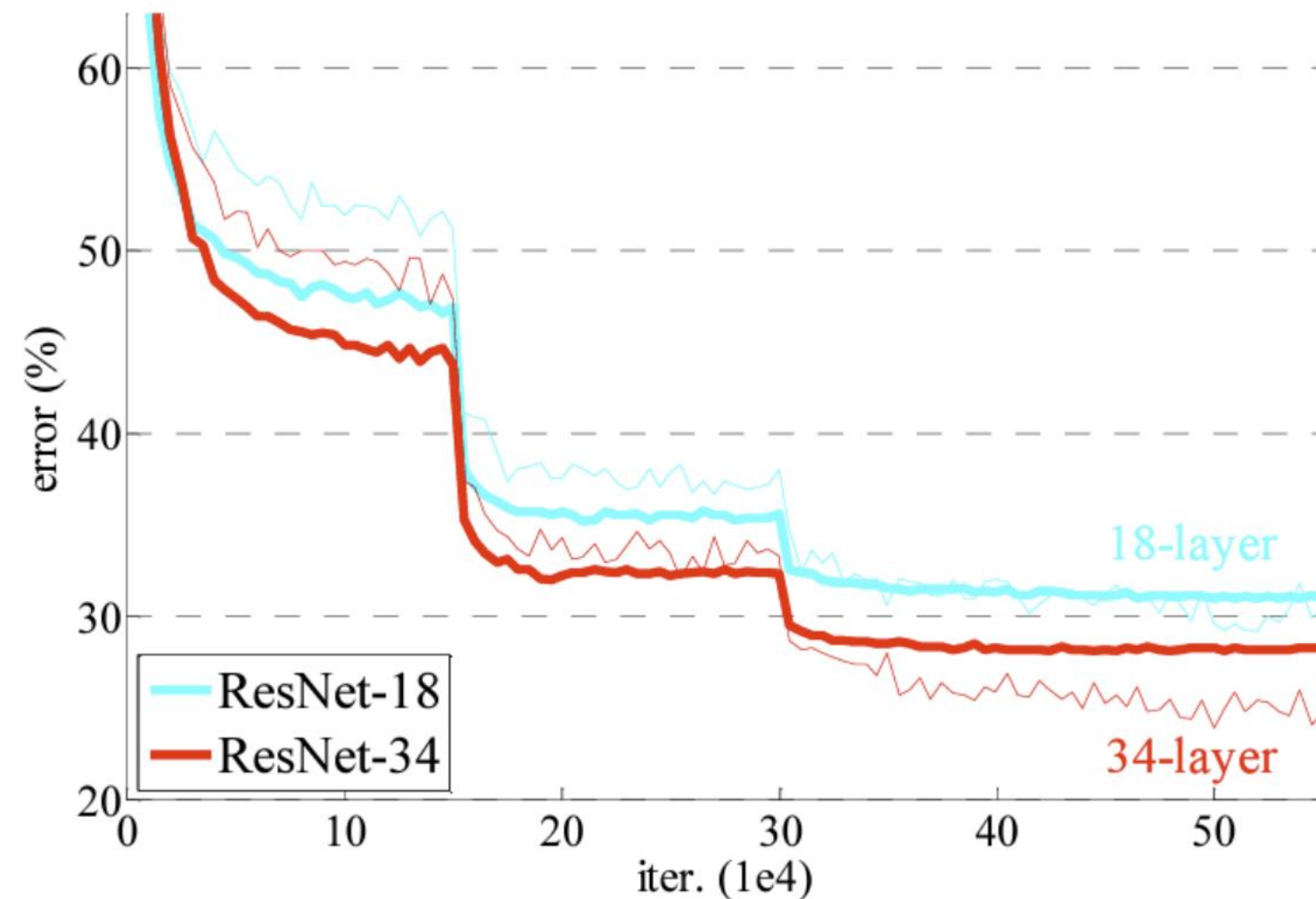


그림 8. Training on ImageNet

- 18 layer보다는 34 layer에서 훨씬 낮은 에러율을 보였고,
- 18 layer에서는 plain과 ResNet과 에러율은 비슷했으나, 학습 파라미터 수가 ResNet이 더 적어 학습 속도가 더 빨랐음

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	25.03

Table 2. Top-1 error (% , 10-crop testing) on ImageNet validation. Here the ResNets have no extra parameter compared to their plain counterparts. Fig. 4 shows the training procedures.

ResNet 정리

주요 특징

- Residual Learning
- Short cut(skip) connection

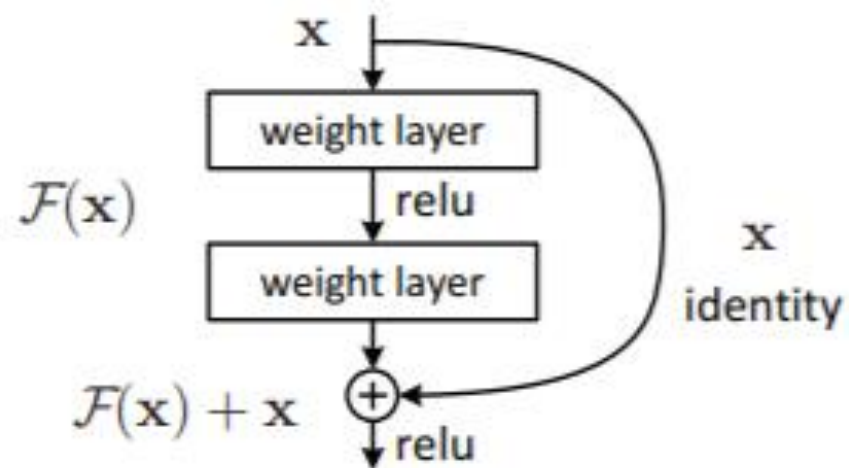
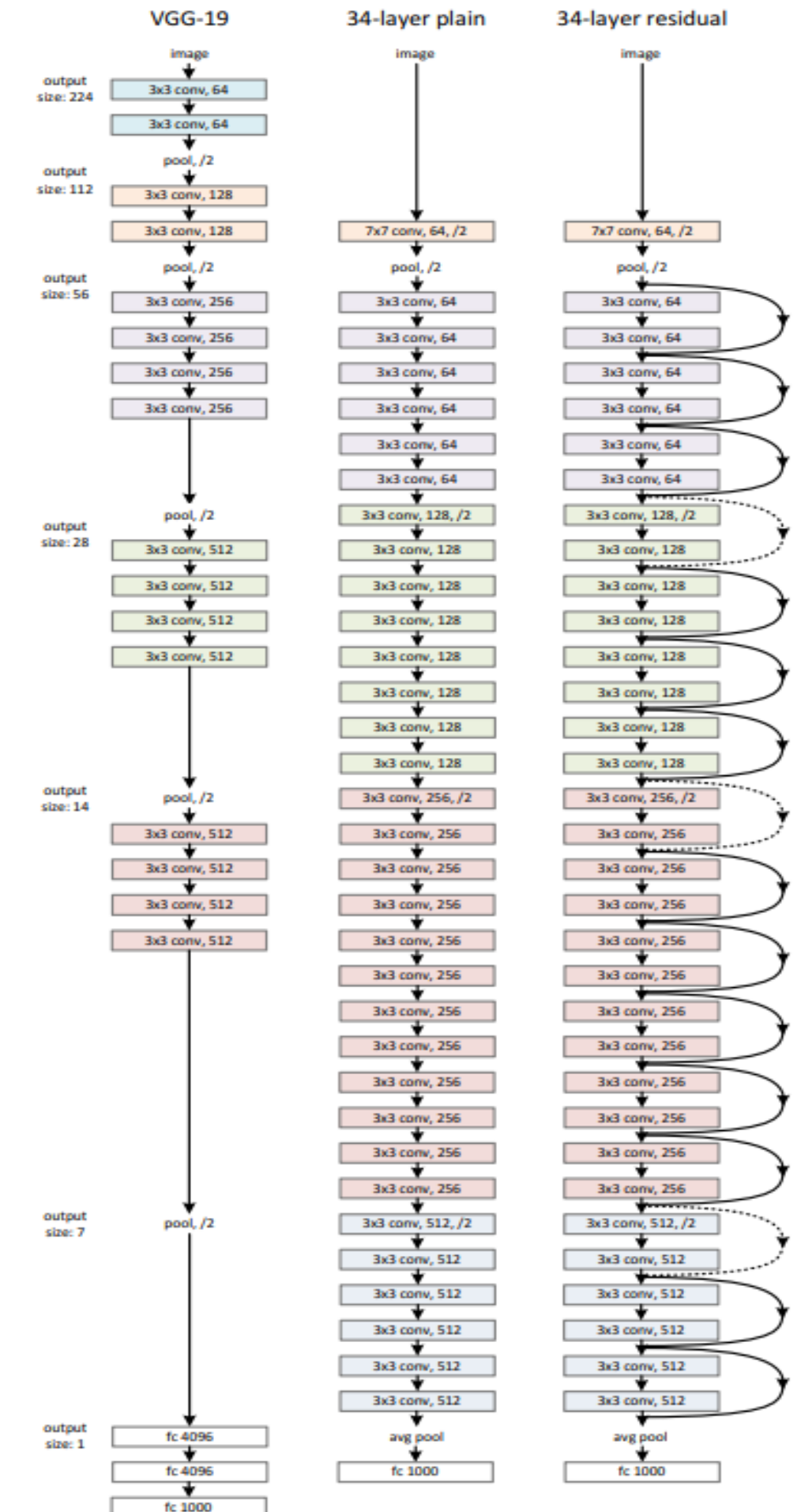
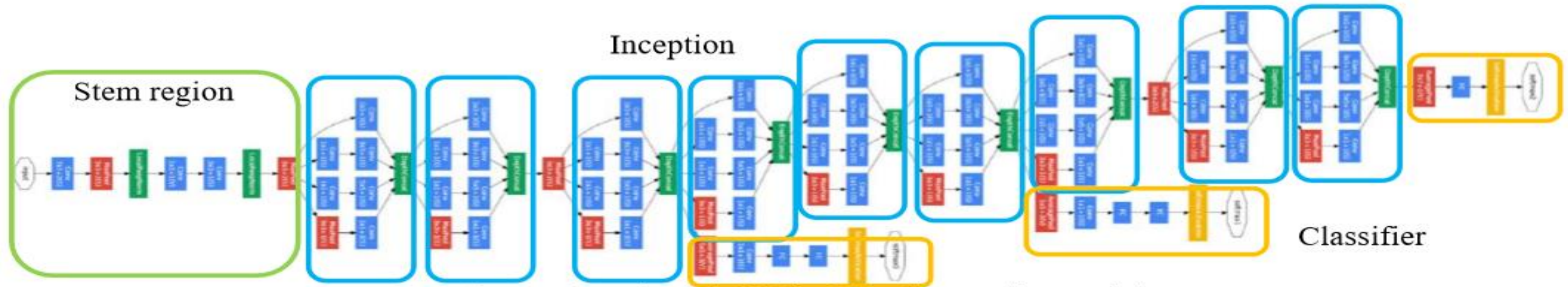


Figure 2. Residual learning: a building block.



GoogleNet 정리



<Source : Szegedy et al. (2015), Going deeper with convolutions>

주요 특징들

1. 기본 CNN 사용
2. 9개의 inception- 차원 축소 1×1 conv layer 적용
3. Auxiliary classifier(보조 분류기)
4. Global Average Pooling

과제

1. GoogleNet 주석 달기
2. ResNet 논문 review





2023 D&A

Basic Session 6차시

THANK YOU

