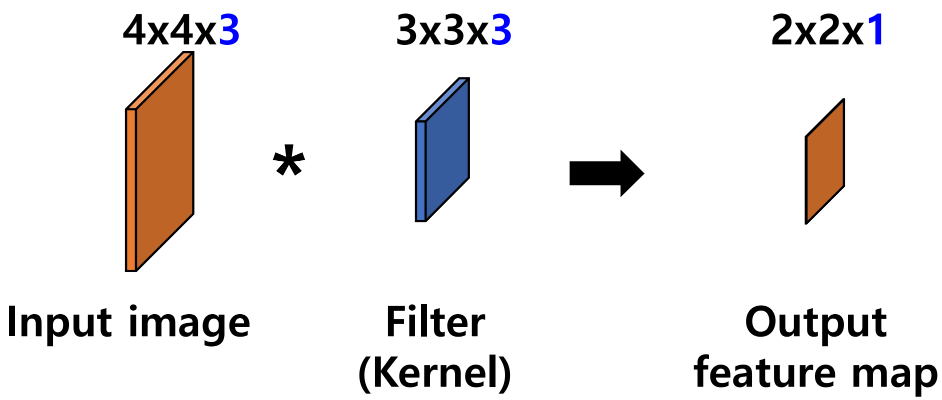
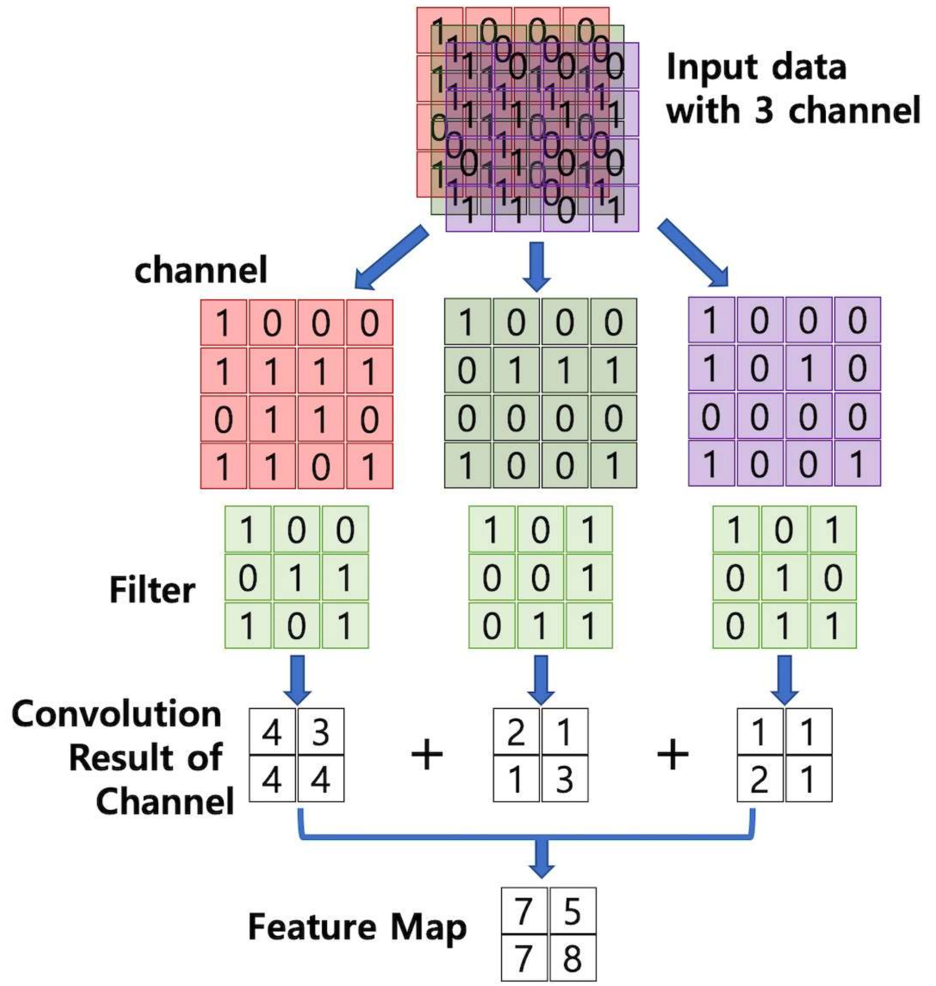
# 12주차 이론
# (CNN 주요설계모듈)

'인공지능 4대 선구자'로 꼽히는 얀 르쿤, 제프리 힌턴, 요슈아 벤지오, 앤드류 응(왼쪽부터). [자료=KAIST]

# Convolutional Neural Networks - 3D 데이터의 Convolution 연산

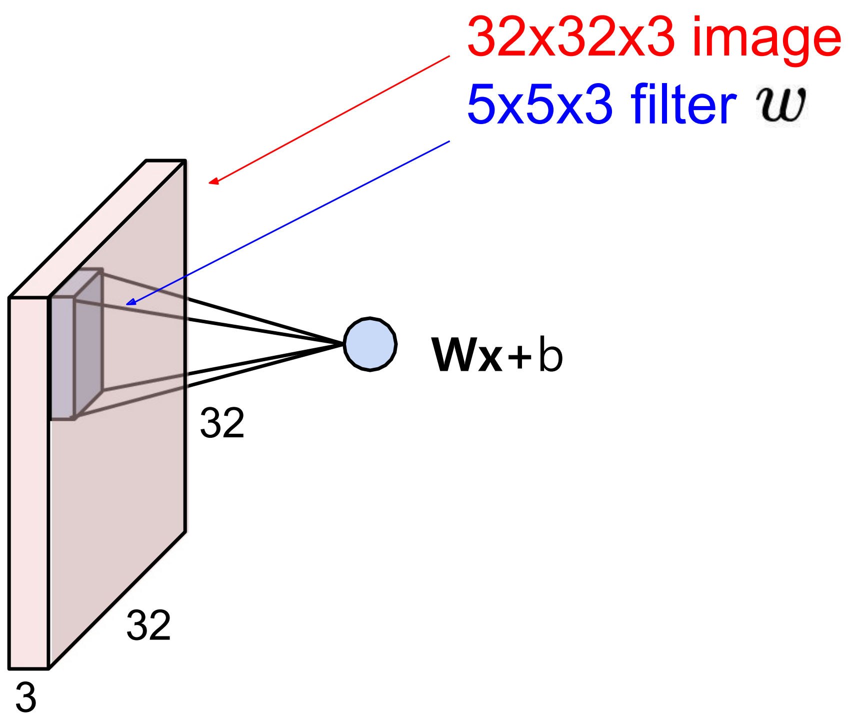
- 3D 이미지 (RGB) 입력에 대한 2D convolution 연산

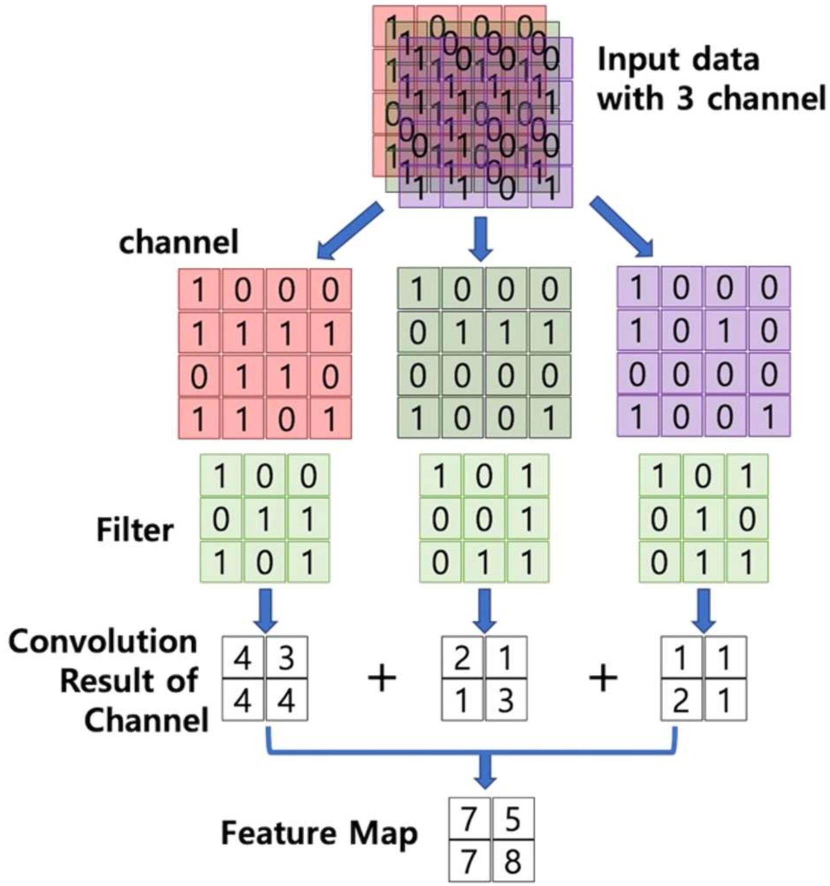


# Convolution Layer

- **Overview**

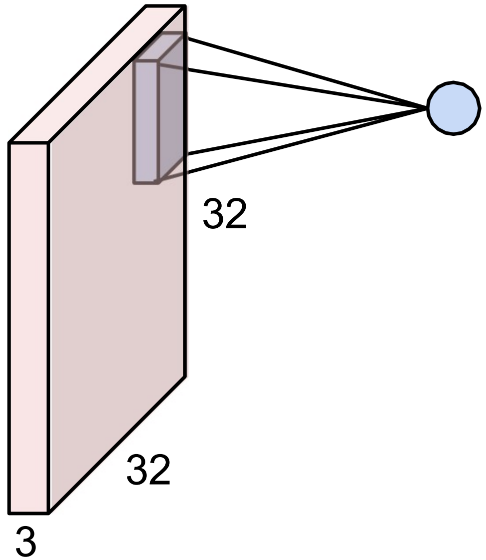



Fei-Fei Li & Justin Johnson & Serena Yeung
출처: cs231n_2017_lecture5 April 18, 2017

# Convolution Layer

- **Overview**

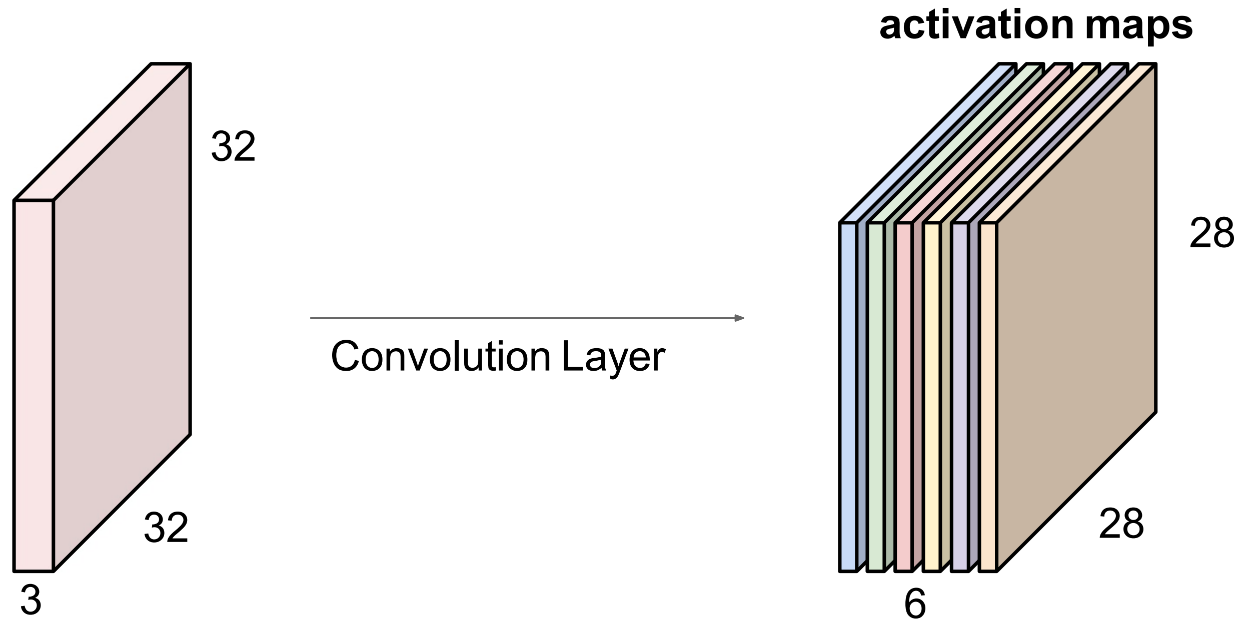



One number = **ReLU(Wx+**b)

Fei-Fei Li & Justin Johnson & Serena Yeung
출처: cs231n_2017_lecture5 April 18, 2017

# Convolution Layer

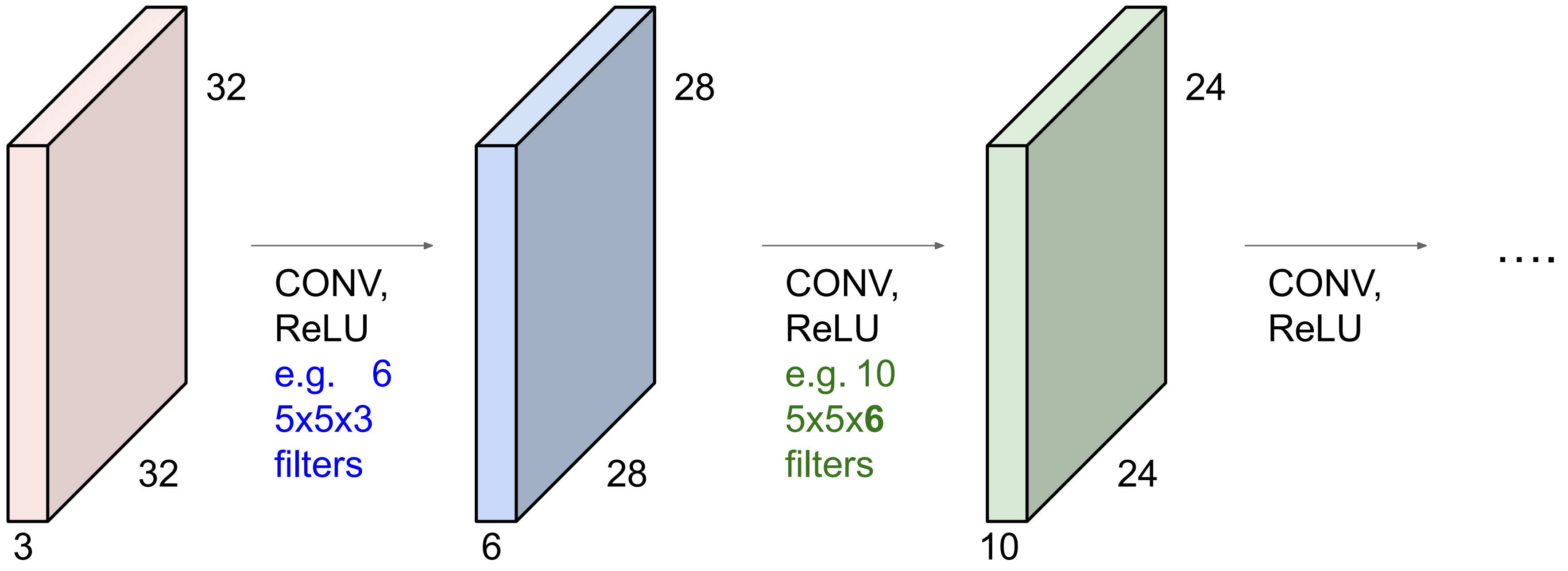
- **Overview**
  - For example, if we had 6 5x5x3 filters, we`ll get 6 separate activation maps




Fei-Fei Li & Justin Johnson & Serena Yeung
출처: cs231n_2017_lecture5 April 18, 2017

# Convolution Layer

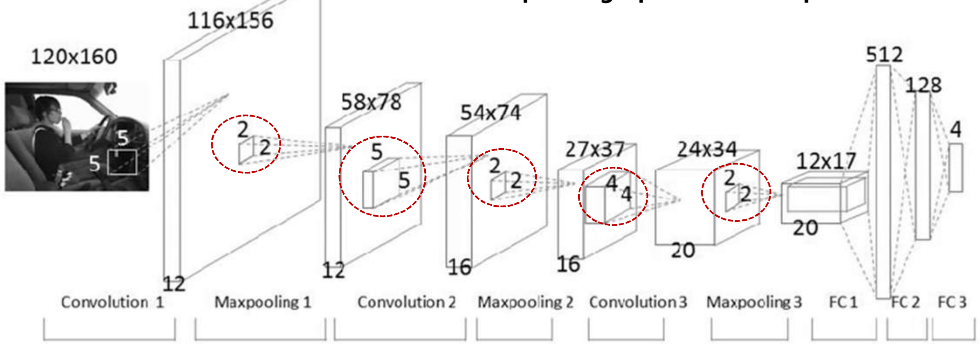
- **Overview**
  - ConvNet is a sequence of Convolutional Layers, interspersed with activation functions



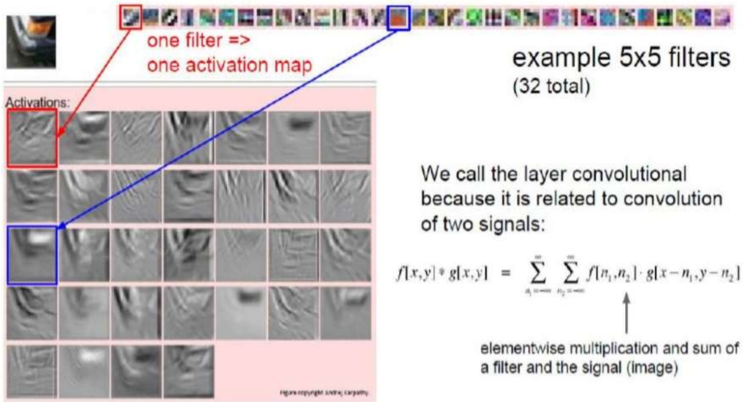

Fei-Fei Li & Justin Johnson & Serena Yeung
출처: cs231n_2017_lecture5 April 18, 2017

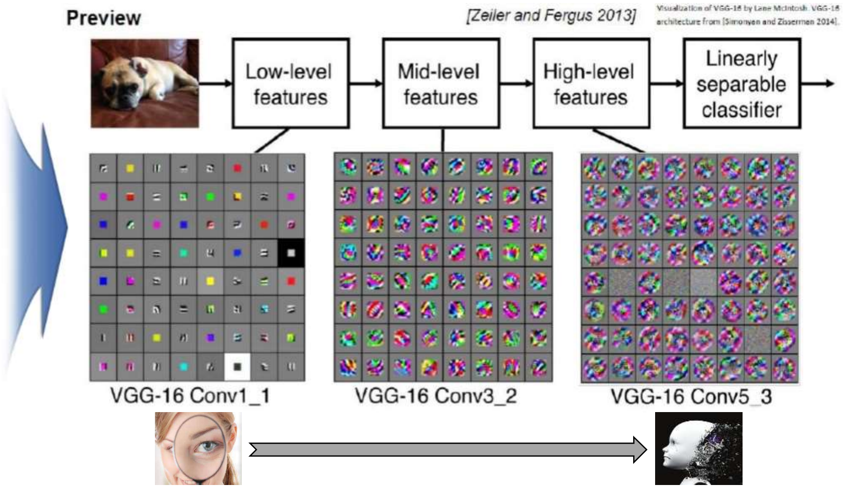
# Deep Learning: CNN Visualization

▪ Example of graphical CNN representation




▪ CNN trains the multiple filters (Kernel).

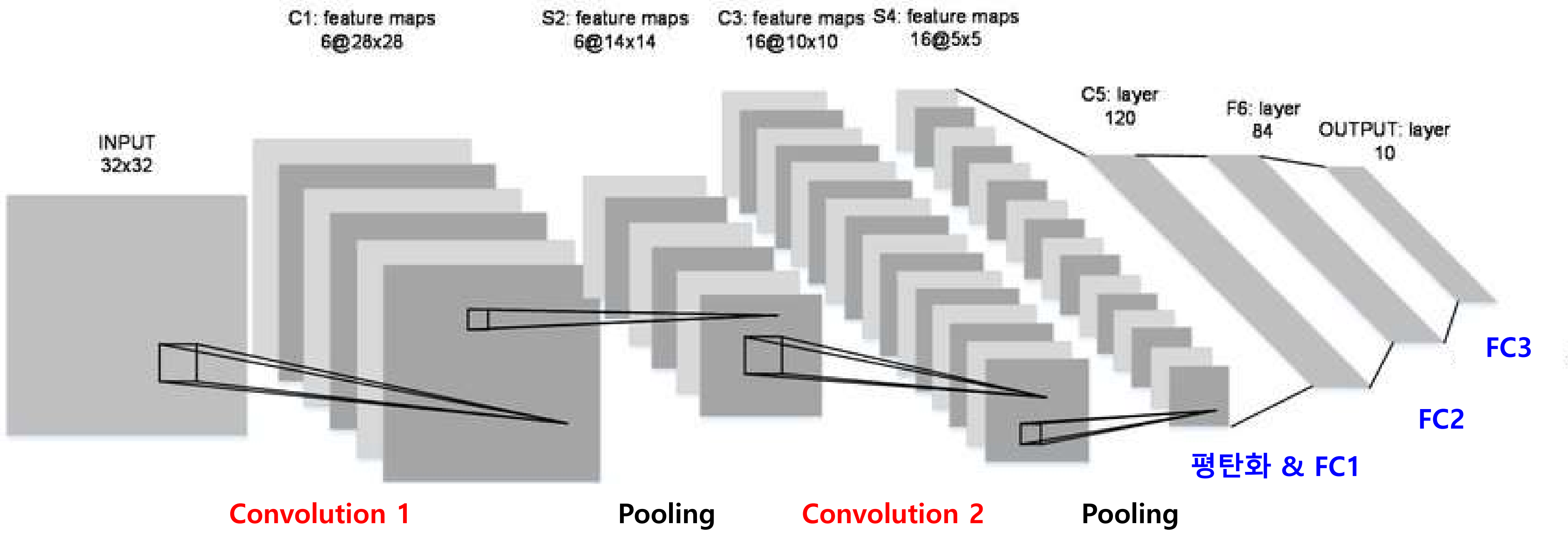



동아대학교
DONG-A UNIVERSITY

# Convolutional Neural Network (CNN) 이론

- **CNN을 이용한 classification model 설계 시 주의사항**
  - 일반적으로 CNN의 feature map을 평탄화 한 이후 fully connected layer에 입력함




**LeNet-5 구조**

Ref.: [Proc. on IEEE 1998] Gradient-based learning applied to document recognition

# LeNet-5 구조

| | |
|---|---|
| **C1** | 훈련해야할 파라미터 개수: (가중치*입력맵개수 + 바이어스)*특성맵개수 = (5*5*1 + 1)*6 = 156 |
| **S2** | 훈련해야할 파라미터 개수: (가중치 + 바이어스)*특성맵개수 = (1 + 1)*6 = 12 |
| **S4** | 훈련해야할 파라미터 개수: (가중치 + 바이어스)*특성맵개수 = (1 + 1)*16 = 32 |
| **C5** | 훈련해야할 파라미터 개수: (가중치*입력맵개수 + 바이어스)*특성맵 개수 = (5*5*16 + 1)*120 = 48120 |
| **F6** | 훈련해야할 파라미터 개수: 연결개수 = (입력개수 + 바이어스)*출력개수 = (120 + 1)*84 = 10164 |



Ref.: [Proc. on IEEE 1998] Gradient-based learning applied to document recognition

# LeNet-5 C3

채널

| 입력 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | X | | | | X | X | X | | | X | X | X | X | | X | X |
| 1 | X | X | | | | X | X | X | | | X | X | X | X | | X |
| 2 | X | X | X | | | | X | X | X | | | X | | X | X | X |
| 3 | | X | X | X | | | X | X | X | X | | | X | | X | X |
| 4 | | | X | X | X | | | X | X | X | X | | X | X | | X |
| 5 | | | | X | X | X | | | X | X | X | X | | X | X | X |

연속된 3장의 입력으로
convolution.
6개의 feature map 생성

연속된 4장의 입력으로
convolution.
6개의 feature map 생성

불연속된 4장의 입력으로
convolution.
3개의 feature map 생성

연속된 6장의 입력으로
convolution.
1개의 feature map 생성

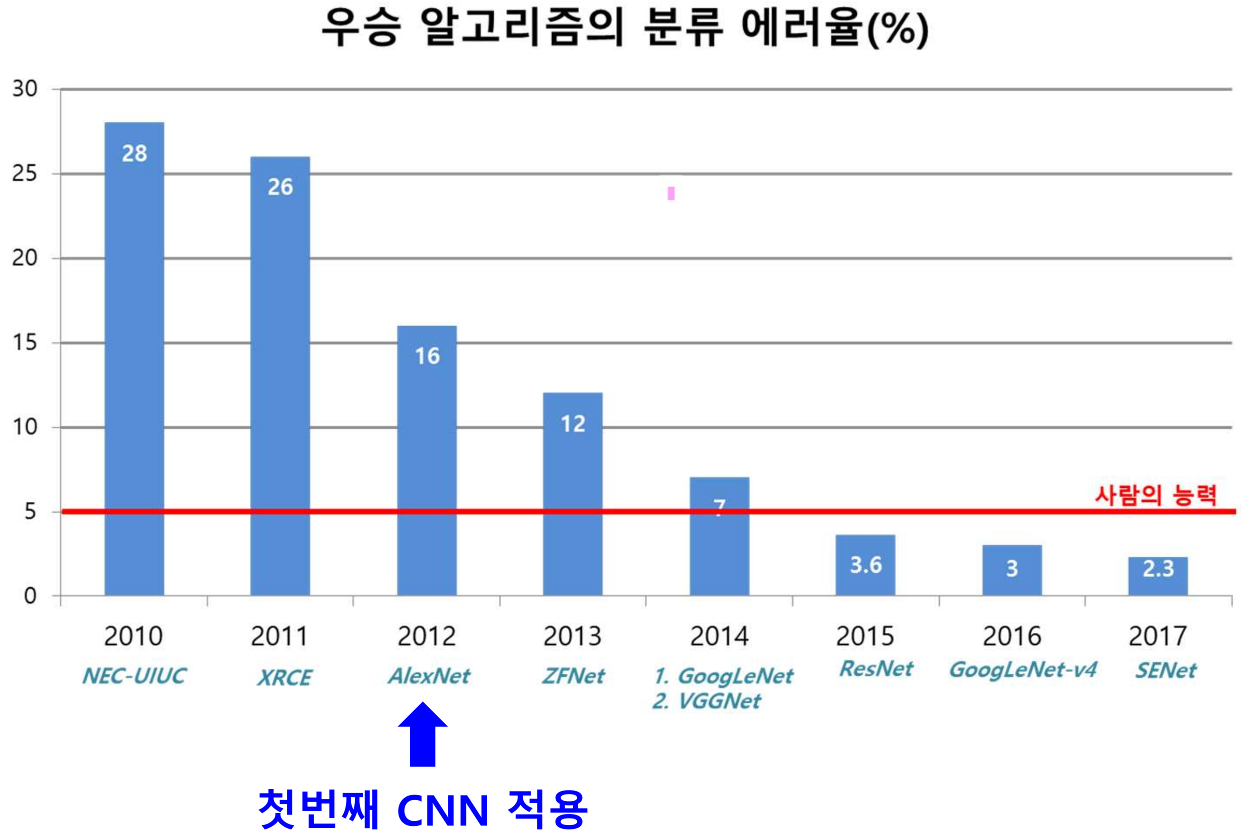
# 딥러닝 주요모델 구성요소

- **Skip connection (ResNet, 2015)**
- **Dense connection (DenseNet, 2017)**
- **Channel attention (SENet, 2018)**
- **Bottleneck Layer**
- **구성요소 적용 예시**

# 딥러닝 모델 구성요소

- **ImageNet Large Scale Visual Recognition Challenge (ILSVRC)**
  - 대용량의 이미지셋 (1000개의 클래스) 에 대한 이미지 분류 알고리즘 성능 평가 대회

**Top-5 error**




Ref.: https://bskyvision.com/425

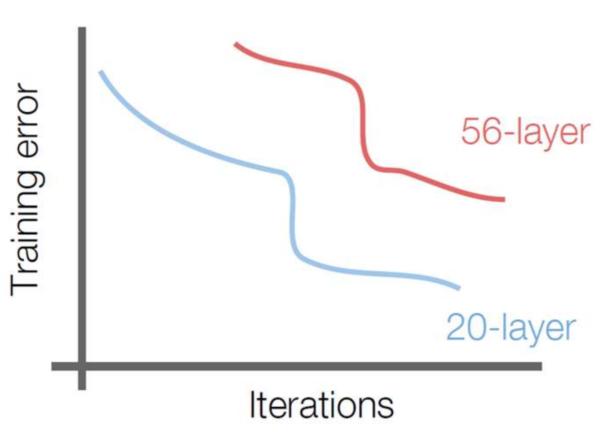
# 딥러닝 모델 구성요소

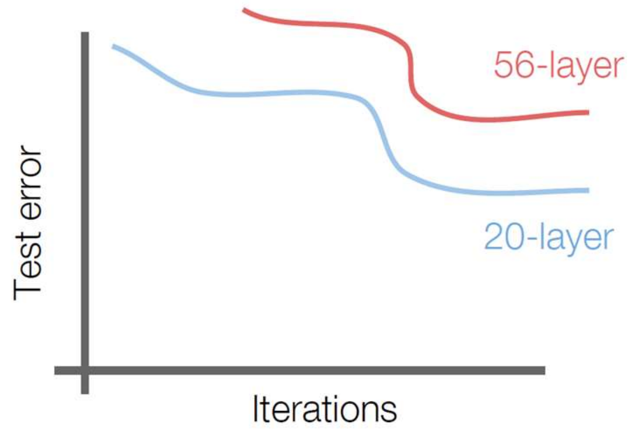
- **Skip connection (ResNet, 2015)**
- **Dense connection (DenseNet, 2017)**
- **Channel attention (SENet, 2018)**
- **Bottleneck Layer**
- **구성요소 적용 예시**

# 딥러닝 모델 구성요소 - Skip connection

- **[CVPR 2015] Deep Residual Learning for Image Recognition (Kaiming He, Microsoft Research)**
  - **ImageNet dataset에 대해 20-layer, 56-layer 모델의 성능 비교**

    **→ 깊은 모델의 성능이 더 떨어지는 것을 확인**

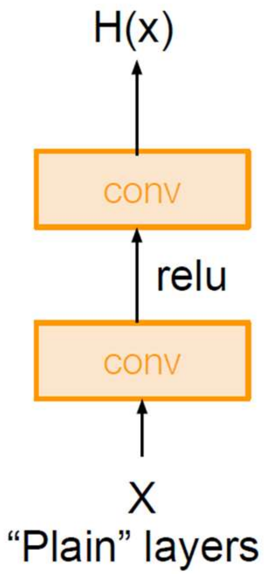


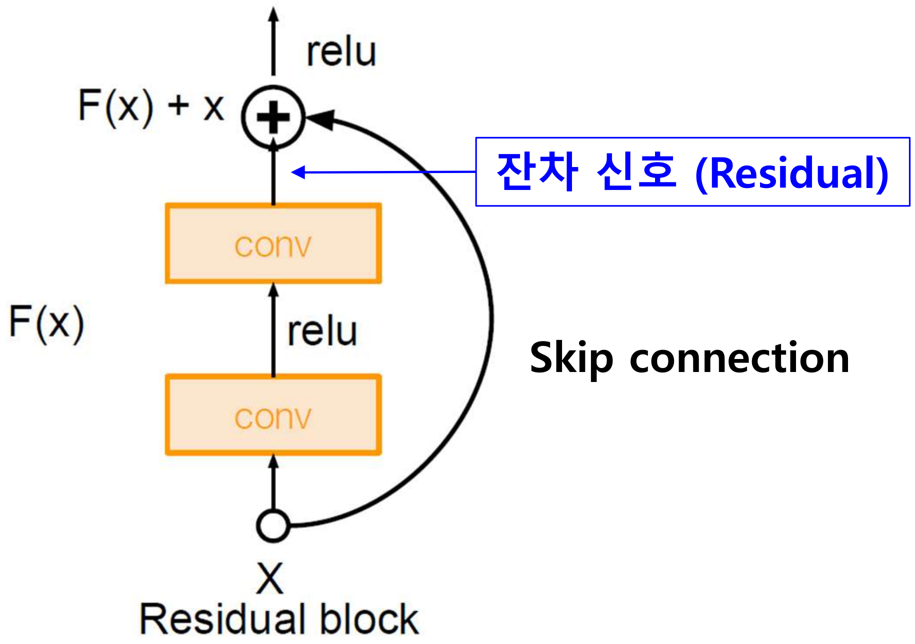




Ref.: cs231n.stanford.edu, Lecture 9

# 딥러닝 모델 구성요소 - Skip connection

- **[CVPR 2015] Deep Residual Learning for Image Recognition (Kaiming He, Microsoft Research)**
  - **잔차 신호 (Residual)을 학습 하게 설계 함으로써 문제 해결 시도**



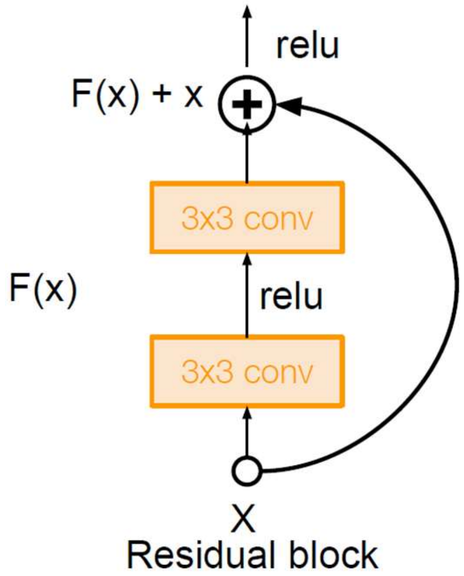



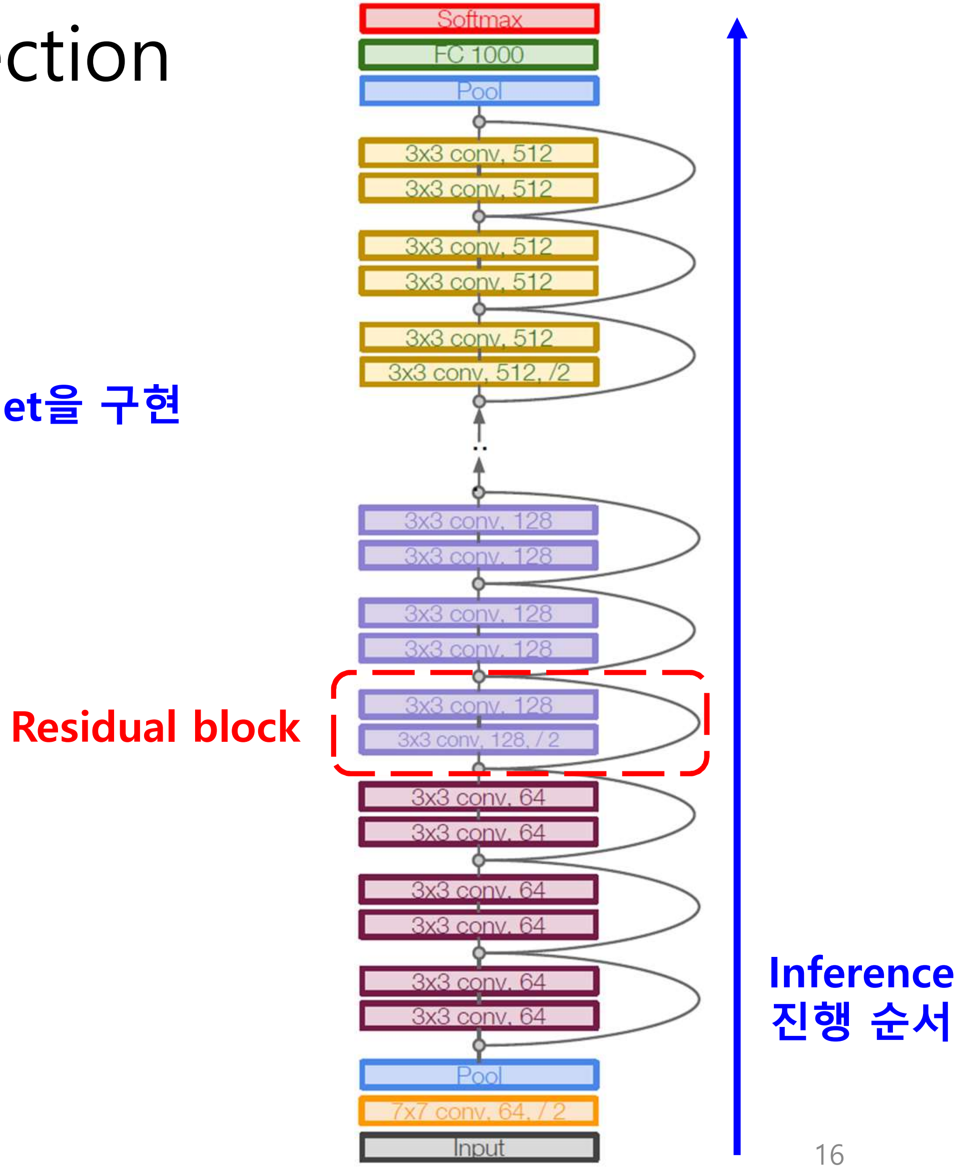



Ref.: cs231n.stanford.edu, Lecture 9

# 딥러닝 모델 구성요소 - Skip connection

- **[CVPR 2015] ResNet (Kaiming He, Microsoft Research)**
  - **3x3 convolution 2개,**
    **skip connection으로 구성된 Residual block 제안**
  - **여러 개의 Residual block을 이용해 제안 기법인 ResNet을 구현**





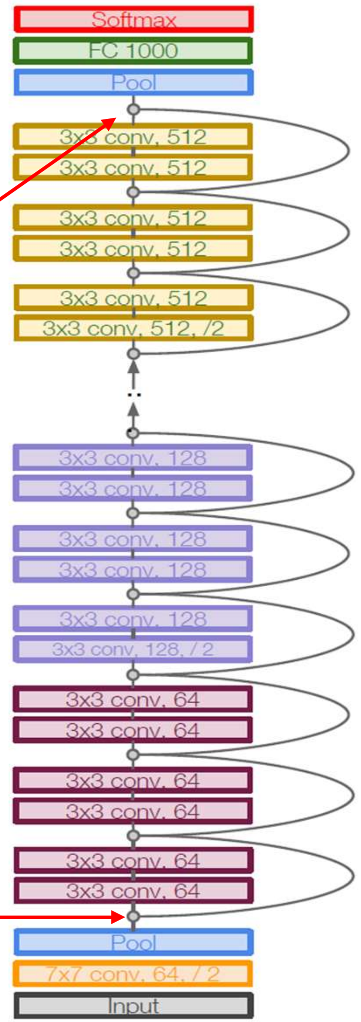

Ref.: cs231n.stanford.edu, Lecture 9

# 딥러닝 모델 구성요소 - Skip connection

- **[CVPR 2015] ResNet (Kaiming He, Microsoft Research)**
  - **Inference가 진행됨에 따라**
    - feature map의 width, height은 감소됨
    - feature map의 channel은 증가됨

Feature map size: 7x7x512

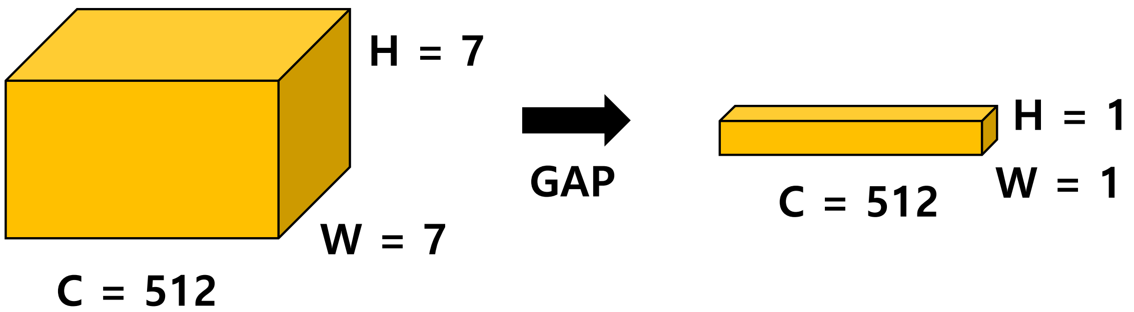
Feature map size: 112x112x64



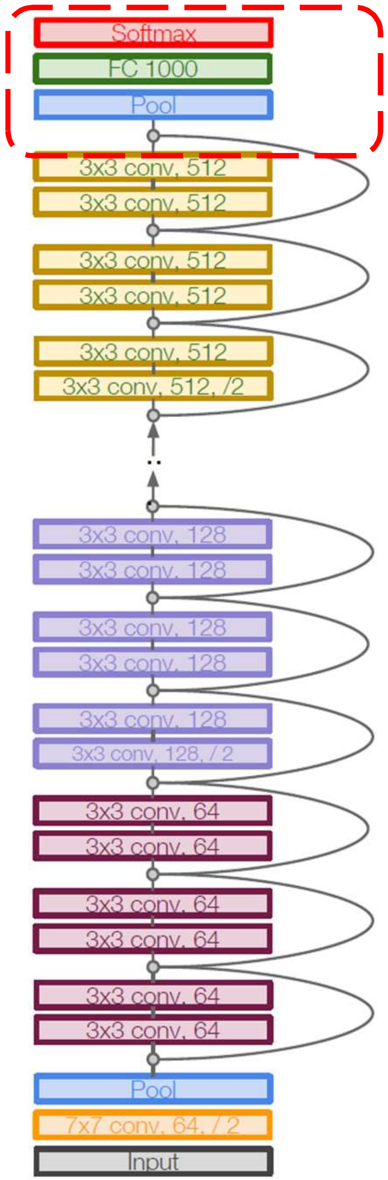

Ref.: cs231n.stanford.edu, Lecture 9

# 딥러닝 모델 구성요소 - Skip connection

- **[CVPR 2015] ResNet (Kaiming He, Microsoft Research)**
  - **Convolution layer의 최종 출력은**
    **Global Average Pooling (GAP)을 통해**
    **Fully connected layer에 입력됨**

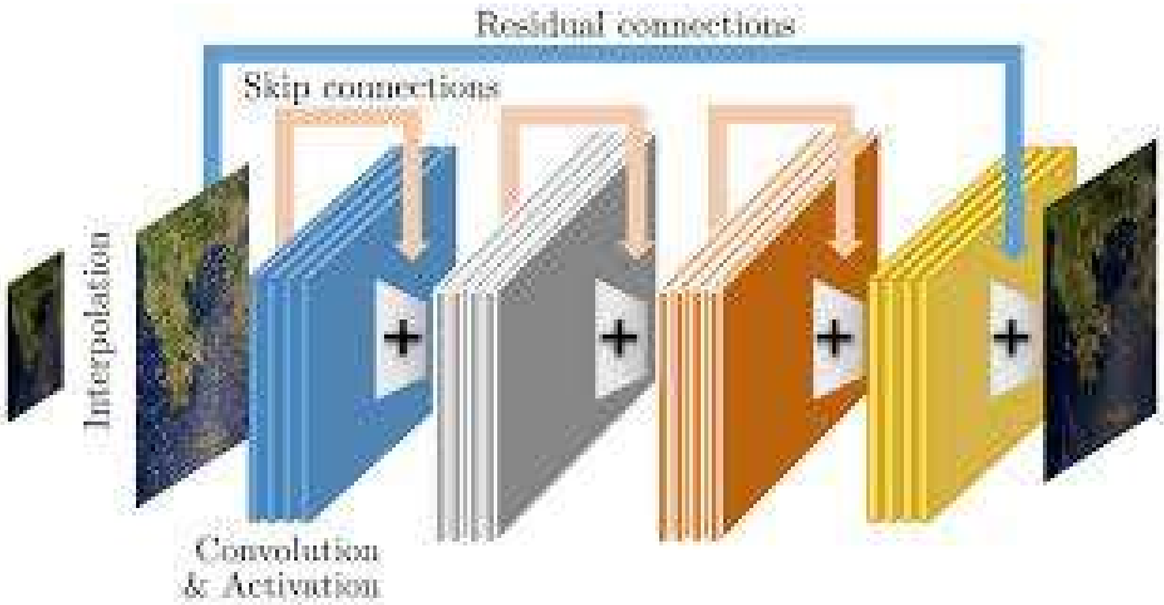



❖ C: Channel
❖ W: Width
❖ H: Height
❖ GAP: Global Average Pooling




Ref.: cs231n.stanford.edu, Lecture 9

# Skip Connection (ResNet)



# 딥러닝 모델 구성요소 - Skip connection

- **[CVPR 2015] ResNet (Kaiming He, Microsoft Research)**
  - **2015년 ImageNet 대회에서는 여러 개의 Layer에 대해 실험한 결과를 제안함**

**5개의 모델에 대해 실험 결과를 제시**

| layer name | output size | 18-layer | 34-layer | 50-layer | 101-layer | 152-layer |
|---|---|---|---|---|---|---|
| conv1 | 112×112 | 7×7, 64, stride 2 | | | | |
| conv2_x | 56×56 | 3×3 max pool, stride 2 | | | | |
| | | $\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$ | $\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$ | $\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$ | $\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$ | $\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$ |
| conv3_x | 28×28 | $\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$ | $\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$ | $\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$ | $\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$ | $\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$ |
| conv4_x | 14×14 | $\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$ | $\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$ | $\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$ | $\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$ | $\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$ |
| conv5_x | 7×7 | $\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$ | $\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$ | $\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$ | $\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$ | $\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$ |
| | 1×1 | average pool, 1000-d fc, softmax | | | | |
| FLOPs | | $1.8\times10^9$ | $3.6\times10^9$ | $3.8\times10^9$ | $7.6\times10^9$ | $11.3\times10^9$ |

**각 모델에 대한 복잡도**

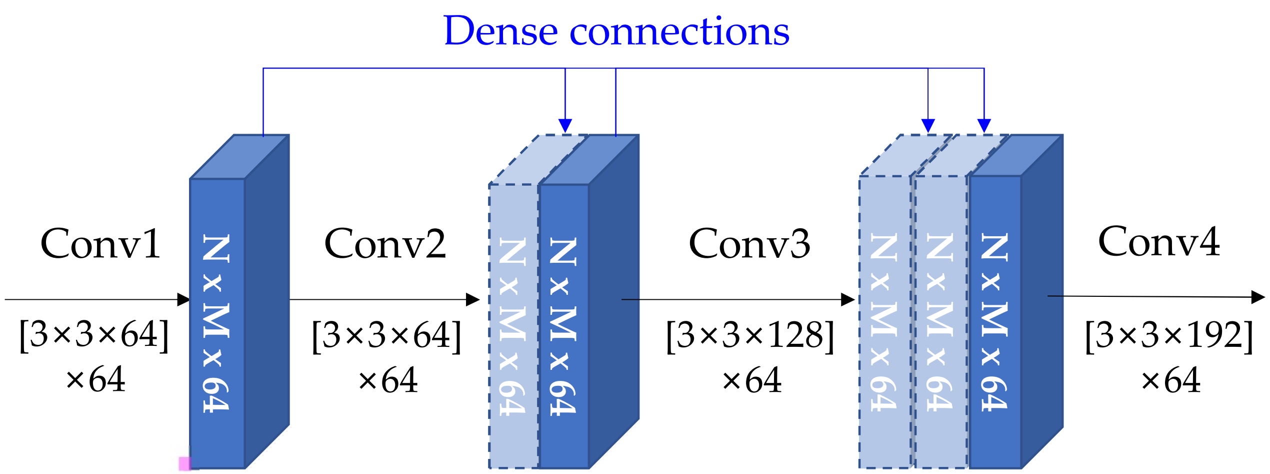
Ref.: cs231n.stanford.edu, Lecture 9

# 딥러닝 모델 구성요소

- **Skip connection (ResNet, 2015)**
- **Dense connection (DenseNet, 2017)**
- **Channel attention (SENet, 2018)**
- **Bottleneck Layer**
- **구성요소 적용 예시**

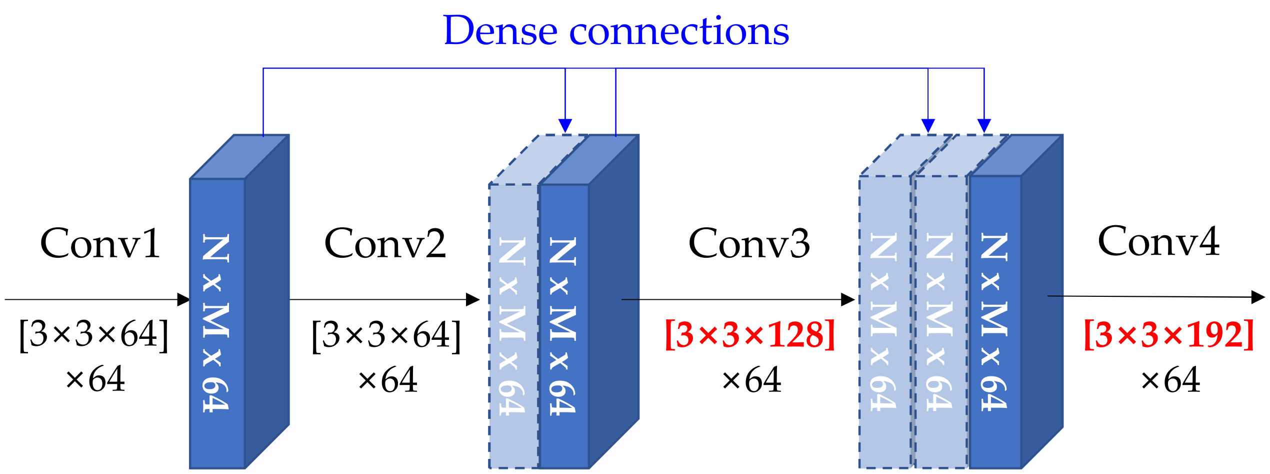
# 딥러닝 모델 구성요소 - Dense connection

- **[CVPR 2017] Densely Connected Convolutional Networks (Gao Huang, Cornell University)**
  - **이전 Layer의 출력 feature map을 이후 layer에서 재사용**



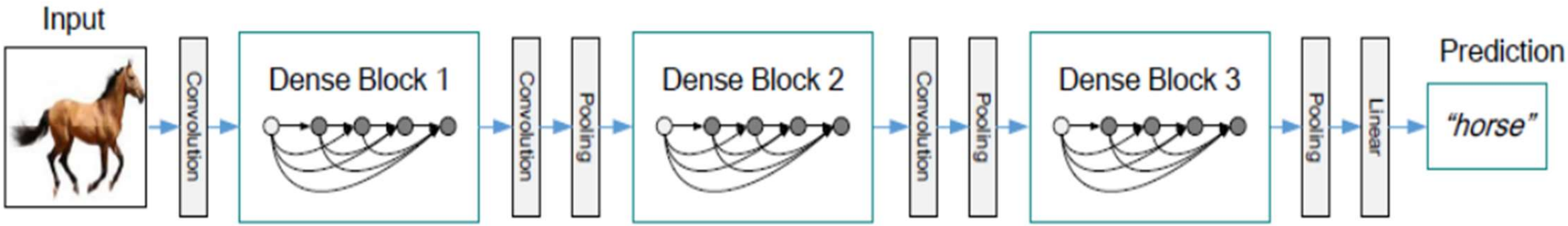
# 딥러닝 모델 구성요소 - Dense connection

- **[CVPR 2017] Densely Connected Convolutional Networks (Gao Huang, Cornell University)**
  - **이전 Layer의 출력 feature map을 이후 layer에서 재사용**
  - **Layer가 깊어짐에 따라 파라미터의 개수가 증가함**



# 딥러닝 모델 구성요소 - Dense connection

- **[CVPR 2017] Densely Connected Convolutional Networks (Gao Huang, Cornell University)**
  - **여러 개의 Convolution layer를 가지는 Dense Block을 정의,**
    **Dense Block 들로 구성되는 DenseNet을 제안**




**DenseNet 네트워크 구조 예시**

# Dense Connection

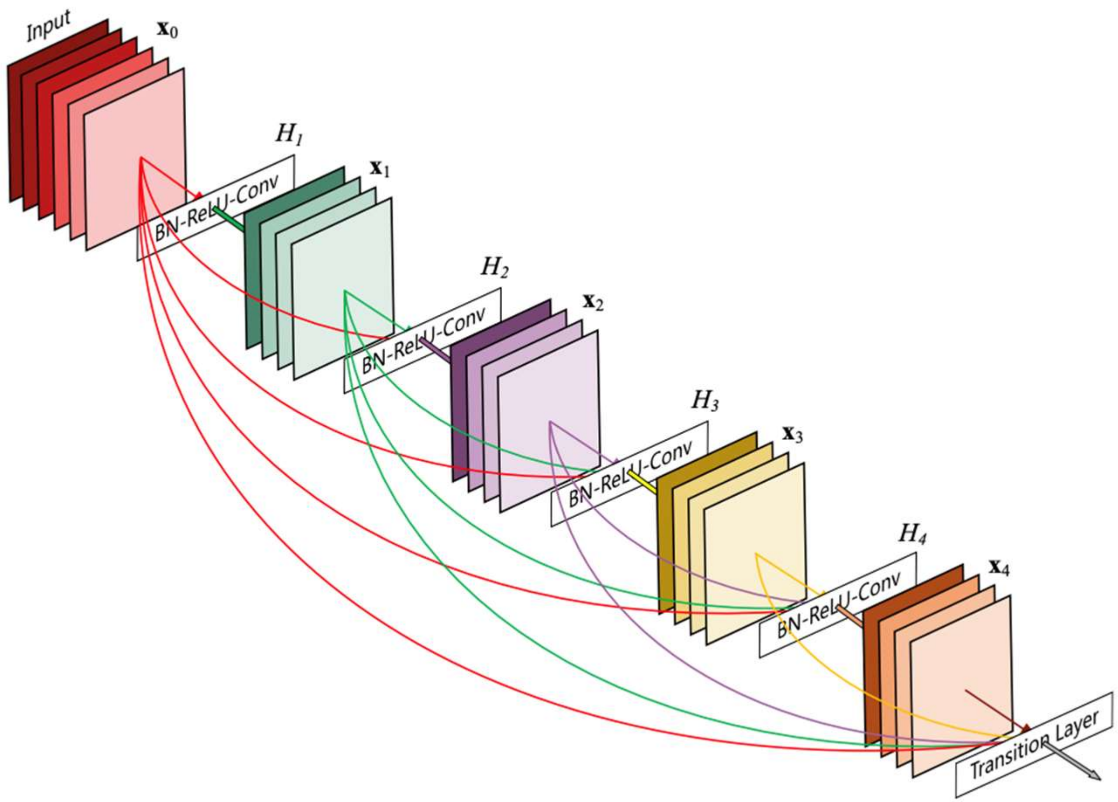



**Figure 1:** A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

# 딥러닝 모델 구성요소 - Dense connection

- **[CVPR 2017] Densely Connected Convolutional Networks (Gao Huang, Cornell University)**
  - **ResNet보다 깊은 구조에 대해 학습을 진행**

**5개의 모델에 대해 실험 결과를 제시**

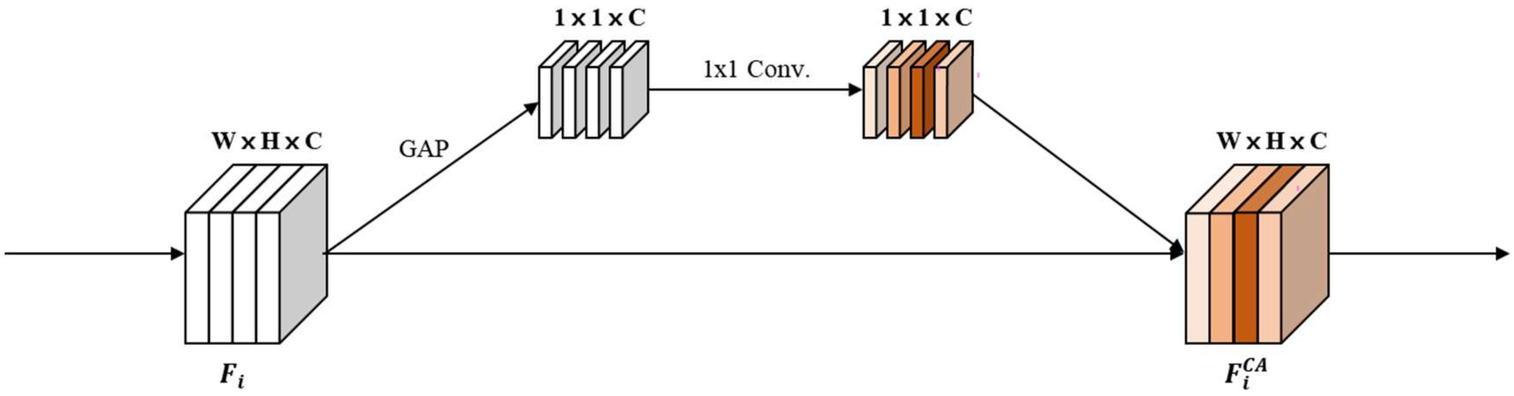
| Layers | Output Size | DenseNet-121 | DenseNet-169 | DenseNet-201 | DenseNet-264 |
|---|---|---|---|---|---|
| Convolution | $112 \times 112$ | $7 \times 7$ conv, stride 2 | | | |
| Pooling | $56 \times 56$ | $3 \times 3$ max pool, stride 2 | | | |
| Dense Block (1) | $56 \times 56$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$ |
| Transition Layer (1) | $56 \times 56$ | $1 \times 1$ conv | | | |
| | $28 \times 28$ | $2 \times 2$ average pool, stride 2 | | | |
| Dense Block (2) | $28 \times 28$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ |
| Transition Layer (2) | $28 \times 28$ | $1 \times 1$ conv | | | |
| | $14 \times 14$ | $2 \times 2$ average pool, stride 2 | | | |
| Dense Block (3) | $14 \times 14$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$ |
| Transition Layer (3) | $14 \times 14$ | $1 \times 1$ conv | | | |
| | $7 \times 7$ | $2 \times 2$ average pool, stride 2 | | | |
| Dense Block (4) | $7 \times 7$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$ |
| Classification Layer | $1 \times 1$ | $7 \times 7$ global average pool | | | |
| | | 1000D fully-connected, softmax | | | |

# 딥러닝 모델 구성요소

- **Skip connection (ResNet, 2015)**
- **Dense connection (DenseNet, 2017)**
- **Channel attention (SENet, 2018)**
- **Bottleneck Layer**
- **구성요소 적용 예시**

# 딥러닝 모델 구성요소 - Channel attention

- **[CVPR 2018] Squeeze-and-Excitation Networks**
  - **Feature map의 각 채널에 대해 중요도를 부여**

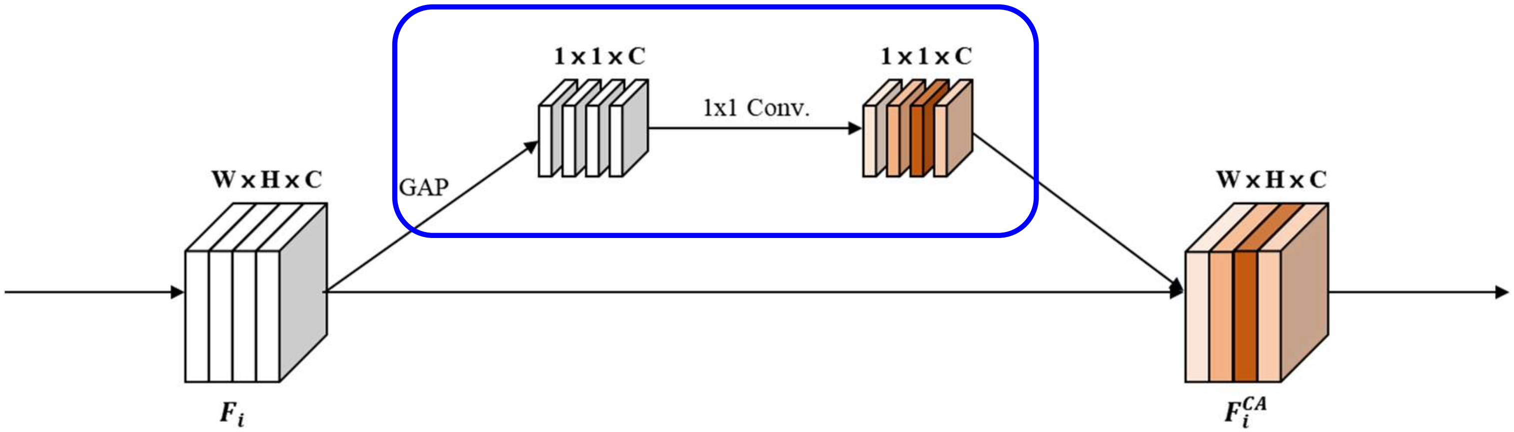



**일반적인 feature map**

**Attention을 통해 중요도가 부여된 feature map**

# 딥러닝 모델 구성요소 - Channel attention

- **[CVPR 2018] Squeeze-and-Excitation Networks**
  - **일반적으로 GAP를 통해 작아진 feature map에 대해 Fully connected layer 또는 1x1 convolution이 사용됨**



# 딥러닝 모델 구성요소 - Channel attention

- **[CVPR 2018] Squeeze-and-Excitation Networks**
  - **기존에 제안된 모델들에 대해 Channel attention을 적용하여 결과 제시**

| | re-implementation | | | SENet | | |
|---|---|---|---|---|---|---|
| | top-1 err. | top-5 err. | GFLOPs | top-1 err. | top-5 err. | GFLOPs |
| ResNet-50 [13] | 24.80 | 7.48 | 3.86 | $23.29_{(1.51)}$ | $6.62_{(0.86)}$ | 3.87 |
| ResNet-101 [13] | 23.17 | 6.52 | 7.58 | $22.38_{(0.79)}$ | $6.07_{(0.45)}$ | 7.60 |
| ResNet-152 [13] | 22.42 | 6.34 | 11.30 | $21.57_{(0.85)}$ | $5.73_{(0.61)}$ | 11.32 |
| ResNeXt-50 [19] | 22.11 | 5.90 | 4.24 | $21.10_{(1.01)}$ | $5.49_{(0.41)}$ | 4.25 |
| ResNeXt-101 [19] | 21.18 | 5.57 | 7.99 | $20.70_{(0.48)}$ | $5.01_{(0.56)}$ | 8.00 |
| VGG-16 [11] | 27.02 | 8.81 | 15.47 | $25.22_{(1.80)}$ | $7.70_{(1.11)}$ | 15.48 |
| BN-Inception [6] | 25.38 | 7.89 | 2.03 | $24.23_{(1.15)}$ | $7.14_{(0.75)}$ | 2.04 |
| Inception-ResNet-v2 [21] | 20.37 | 5.21 | 11.75 | $19.80_{(0.57)}$ | $4.79_{(0.42)}$ | 11.76 |

**Channel attention을 통한 성능 향상**

# 딥러닝 모델 구성요소 - Channel attention

- **[CVPR 2018] Squeeze-and-Excitation Networks**
  - **기존에 제안된 모델들에 대해 Channel attention을 적용하여 결과 제시**

| | re-implementation | | | SENet | | |
|---|---|---|---|---|---|---|
| | top-1 err. | top-5 err. | GFLOPs | top-1 err. | top-5 err. | GFLOPs |
| ResNet-50 [13] | 24.80 | 7.48 | 3.86 | $23.29_{(1.51)}$ | $6.62_{(0.86)}$ | 3.87 |
| ResNet-101 [13] | 23.17 | 6.52 | 7.58 | $22.38_{(0.79)}$ | $6.07_{(0.45)}$ | 7.60 |
| ResNet-152 [13] | 22.42 | 6.34 | 11.30 | $21.57_{(0.85)}$ | $5.73_{(0.61)}$ | 11.32 |
| ResNeXt-50 [19] | 22.11 | 5.90 | 4.24 | $21.10_{(1.01)}$ | $5.49_{(0.41)}$ | 4.25 |
| ResNeXt-101 [19] | 21.18 | 5.57 | 7.99 | $20.70_{(0.48)}$ | $5.01_{(0.56)}$ | 8.00 |
| VGG-16 [11] | 27.02 | 8.81 | 15.47 | $25.22_{(1.80)}$ | $7.70_{(1.11)}$ | 15.48 |
| BN-Inception [6] | 25.38 | 7.89 | 2.03 | $24.23_{(1.15)}$ | $7.14_{(0.75)}$ | 2.04 |
| Inception-ResNet-v2 [21] | 20.37 | 5.21 | 11.75 | $19.80_{(0.57)}$ | $4.79_{(0.42)}$ | 11.76 |

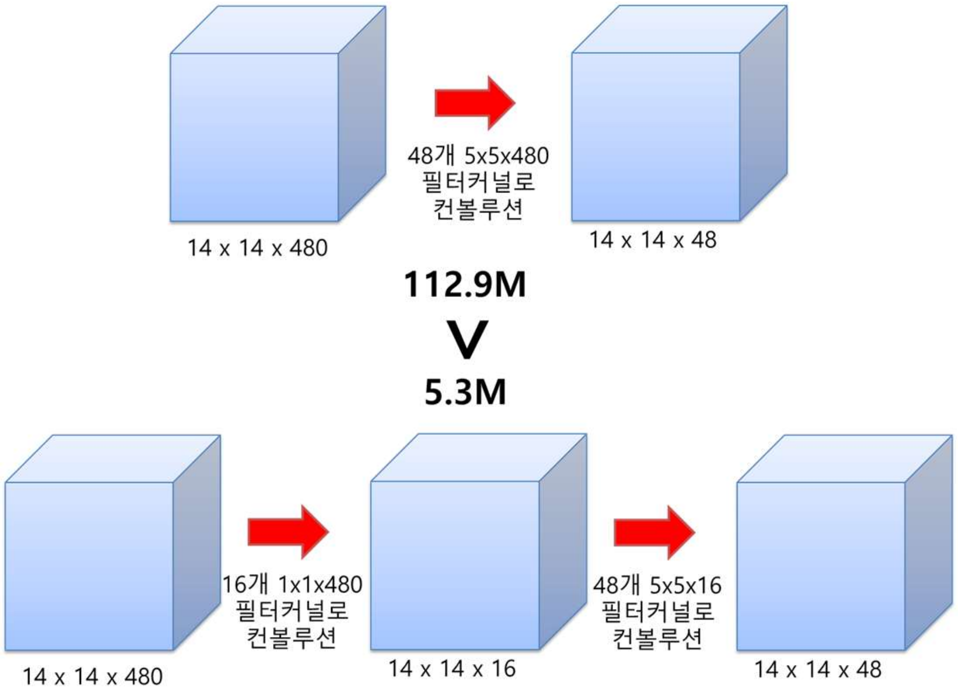
복잡도는 거의 증가되지 않음

# 딥러닝 모델 구성요소

- **Skip connection (ResNet, 2015)**
- **Dense connection (DenseNet, 2017)**
- **Channel attention (SENet, 2018)**
- **Bottleneck Layer**
- **구성요소 적용 예시**

# Bottleneck Layer

- **1x1 컨볼루션 사용 ➔ feature map의 개수를 줄이는 목적으로 사용**




- ✓ **Memory: [(5 x 5 x 480) + 1] x 48 + (14 x 14 x 48)**
- ✓ **연산횟수: (14 x 14 x 48) x (5 x 5 x 480) = 약 112.9M**

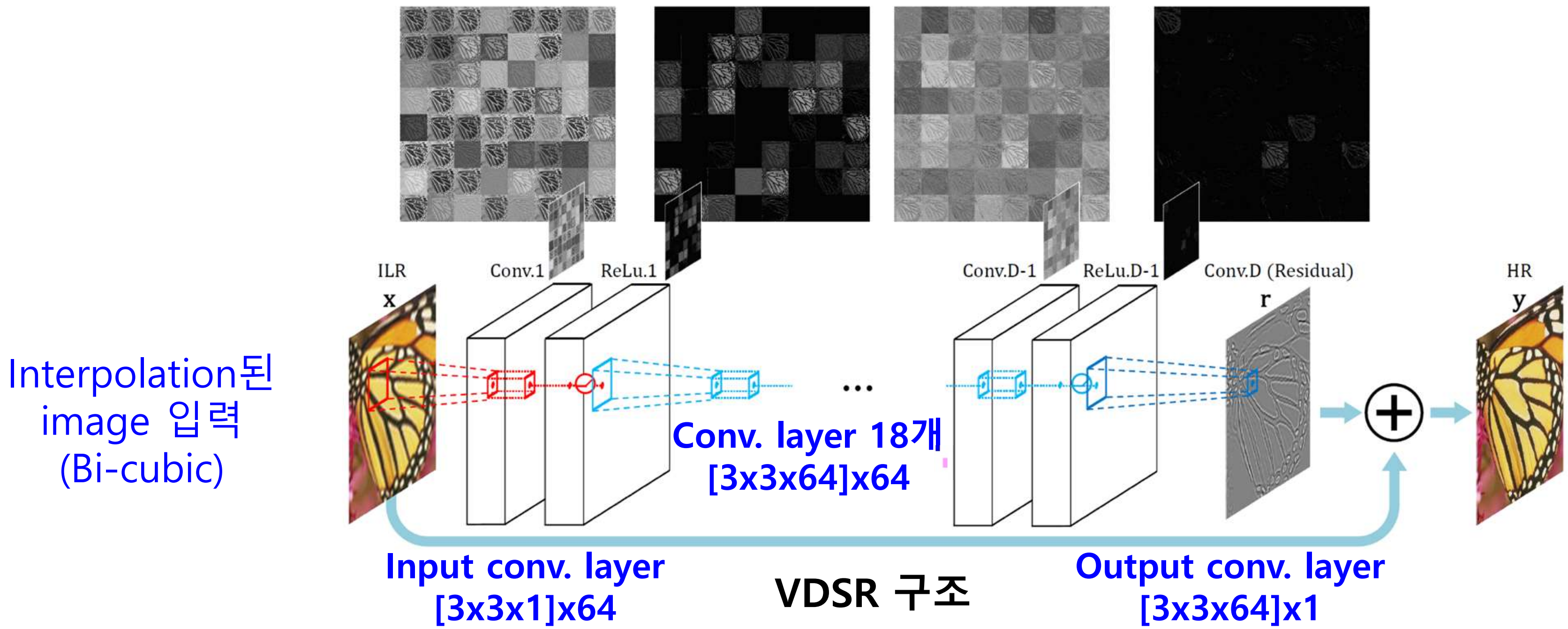
- ✓ **Memory: [(1 x 1 x 480) + 1] x 16 + [(5 x 5 x 16) + 1] x 48 + (14 x 14 x 16) + (14 x 14 x 48)**
- ✓ **연산횟수: (14 x 14 x 16)*(1 x 1 x 480) + (14 x 14 x 48)*(5 x 5 x 16) = 약 5.3M**

# 딥러닝 모델 구성요소

- **Skip connection (ResNet, 2015)**
- **Dense connection (DenseNet, 2017)**
- **Channel attention (SENet, 2018)**
- **Bottleneck Layer**
- **구성요소 적용 예시**

# 딥러닝 모델 구성요소 - 구성요소 적용 예시

- [CVPR 2016] Accurate Image Super-Resolution Using Very Deep Convolutional Networks (VDSR)
  - 3x3 Convolution 20개 사용, SR 분야에 대해 최초로 Residual Learning 적용




VDSR 구조

# 딥러닝 모델 구성요소 - 구성요소 적용 예시

- [CVPR 2017] Image Super-Resolution Using Dense Skip Connections (SR-DenseNet)
  - Dense connection을 이용해 Dense block 8의 출력 단은 1,000개 이상의 feature map을 사용