Video classification with keras

딥러닝스터디
2020-10-21
이 예 림

Video classification vs. Image classification

Video? A series of individual images!

Video classification = *N* images classification
(*N* = # of frames)

Video classification vs. Image classification

Video? A series of individual images!

Video classification = *N* images classification
(*N* = # of frames)

Video classification vs. Image classification

Video? A series of individual images!

Video classification = *N* images classification
(*N* = # of frames)

Subsequent frames are correlated with *semantic contents*.
> Temporal dimension > video classification results improval
(i.e., LSTM, RNN ⋯)

Video classification vs. Image classification

Video? A series of individual images!

Video classification = *N* images classification
(*N* = # of frames)

Subsequent frames are correlated with *semantic contents*.
> Temporal dimension > video classification results improval
(i.e., LSTM, RNN ⋯)

Rolling prediction averaging!

Image classification (CNN) > Video classification (Rolling averaging)

Video classification vs. Image classification

Image classification

1. Input : Image
2. Prediction
3. The largest corresponding probability >> The label

Video classification

1. Input : Frame
2. Loop over all frames
3. Prediction *individually* and *independently*
4. The largest corresponding probability >> The label

"Prediction flickering"

Video classification vs. Image classification

Image classification

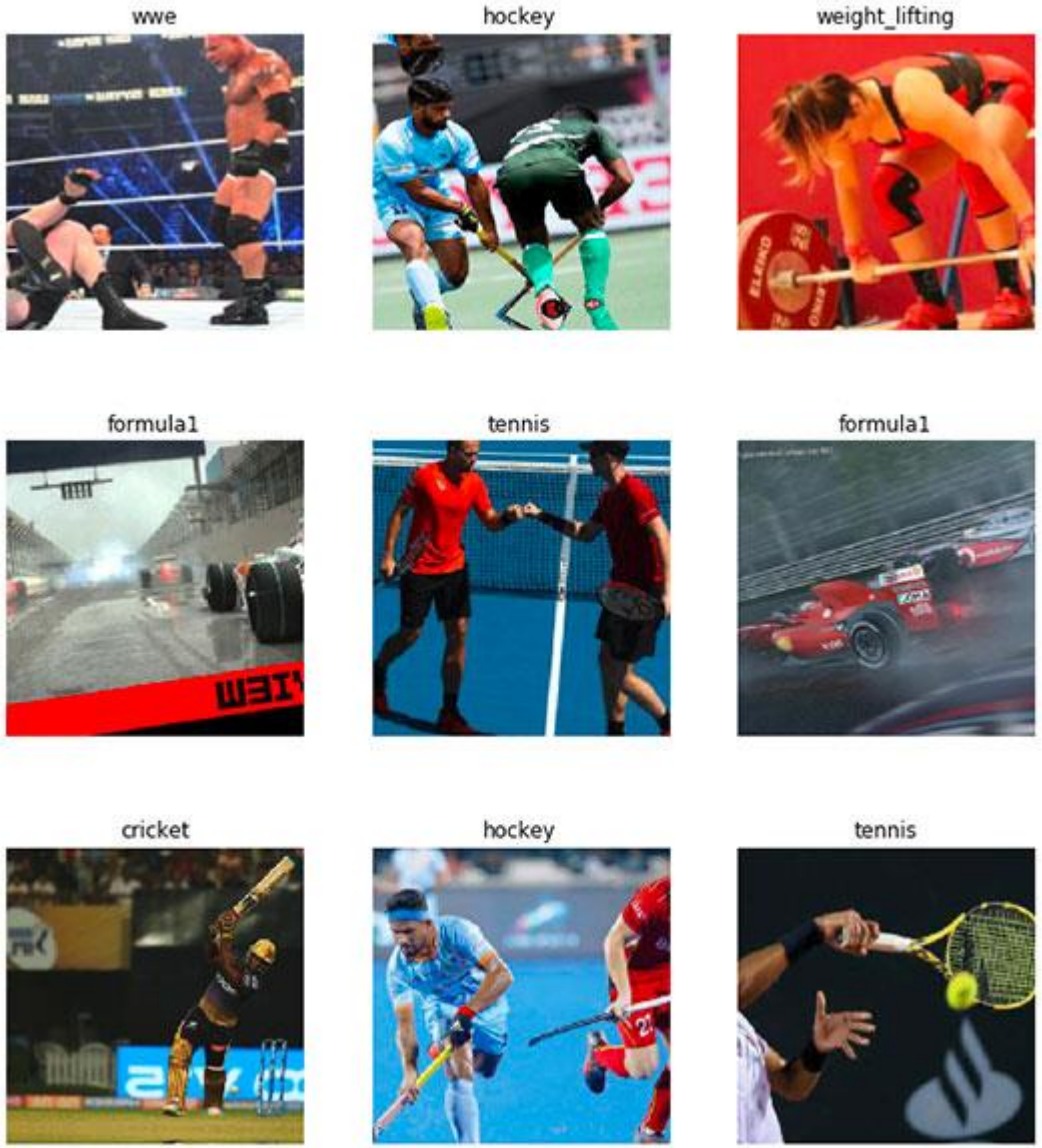
1. Input : Image
2. Prediction
3. The largest corresponding probability >> The label

Video classification

1. Input : Frame
2. Loop over all frames
3. Prediction *individually* and *independently*
4. A list of the last *K* predictions
5. The average of the last *K* predictions
6. The largest corresponding probability >> The label

"Smooting out"

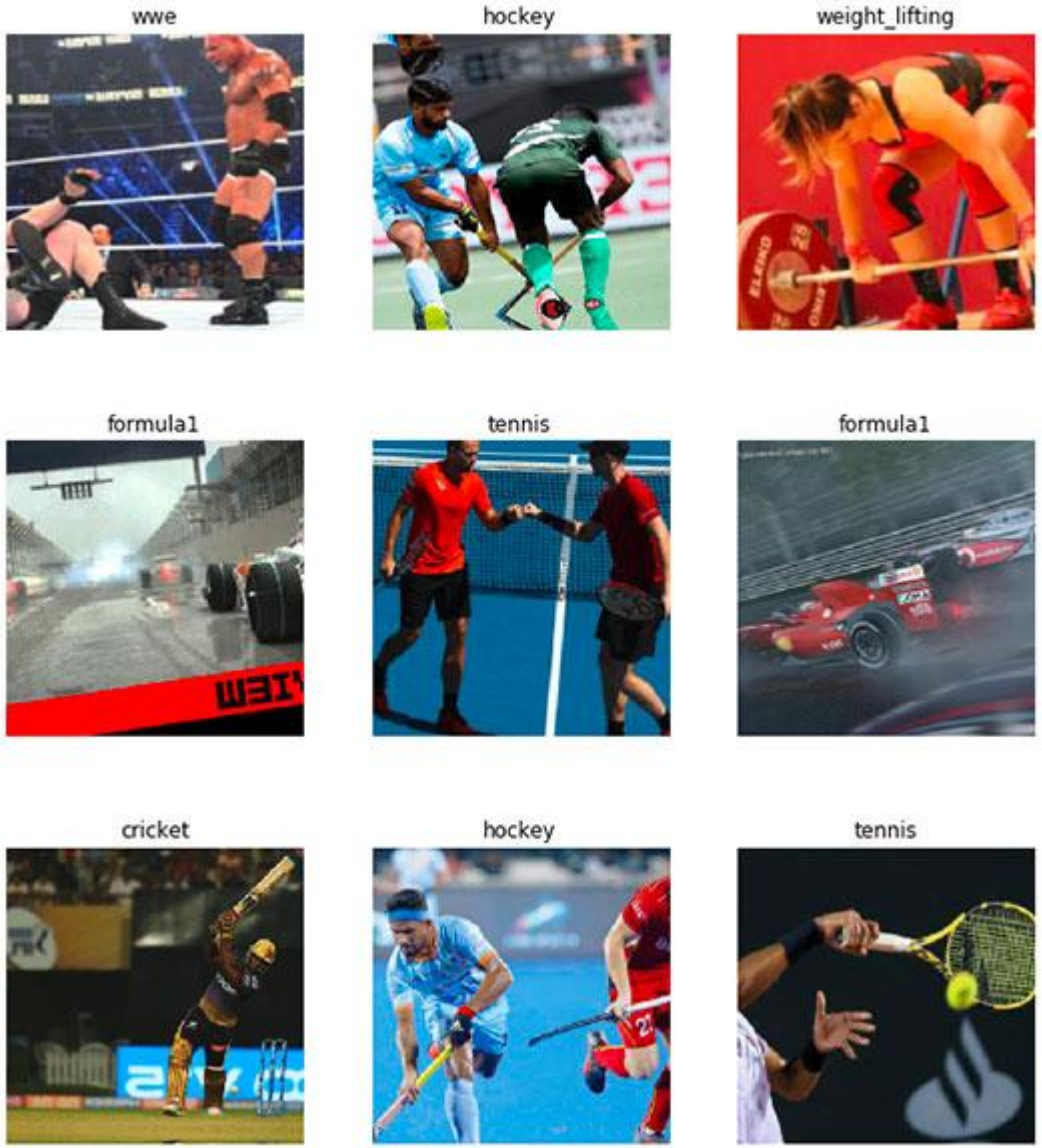
The sports classification dataset



1.Swimming
2.Badminton
3.Wrestling
4.Olympic Shooting
5.Cricket
6.Football
7.Tennis
8.Hockey
9.Ice Hockey
10.Kabaddi
11.WWE
12.Gymnasium
13.Weight lifting
14.Volleyball
15.Table tennis
16.Baseball
17.Formula 1
18.Moto GP
19.Chess
20.Boxing
21.Fencing
22.Basketba

https://github.com/anubhavmaity

The sports classification dataset



1.Swimming
2.Badminton
3.Wrestling
4.Olympic Shooting
5.Cricket
6.Football
7.Tennis
8.Hockey
9.Ice Hockey
10.Kabaddi
11.WWE
12.Gymnasium
13.Weight lifting
14.Volleyball
15.Table tennis
16.Baseball
17.Formula 1
18.Moto GP
19.Chess
20.Boxing
21.Fencing
22.Basketba

https://github.com/anubhavmaity

Transfer learning

1. Take a network pre-trained on a dataset.
2. Utilize the network to recognize image/object categories it was not trained on.

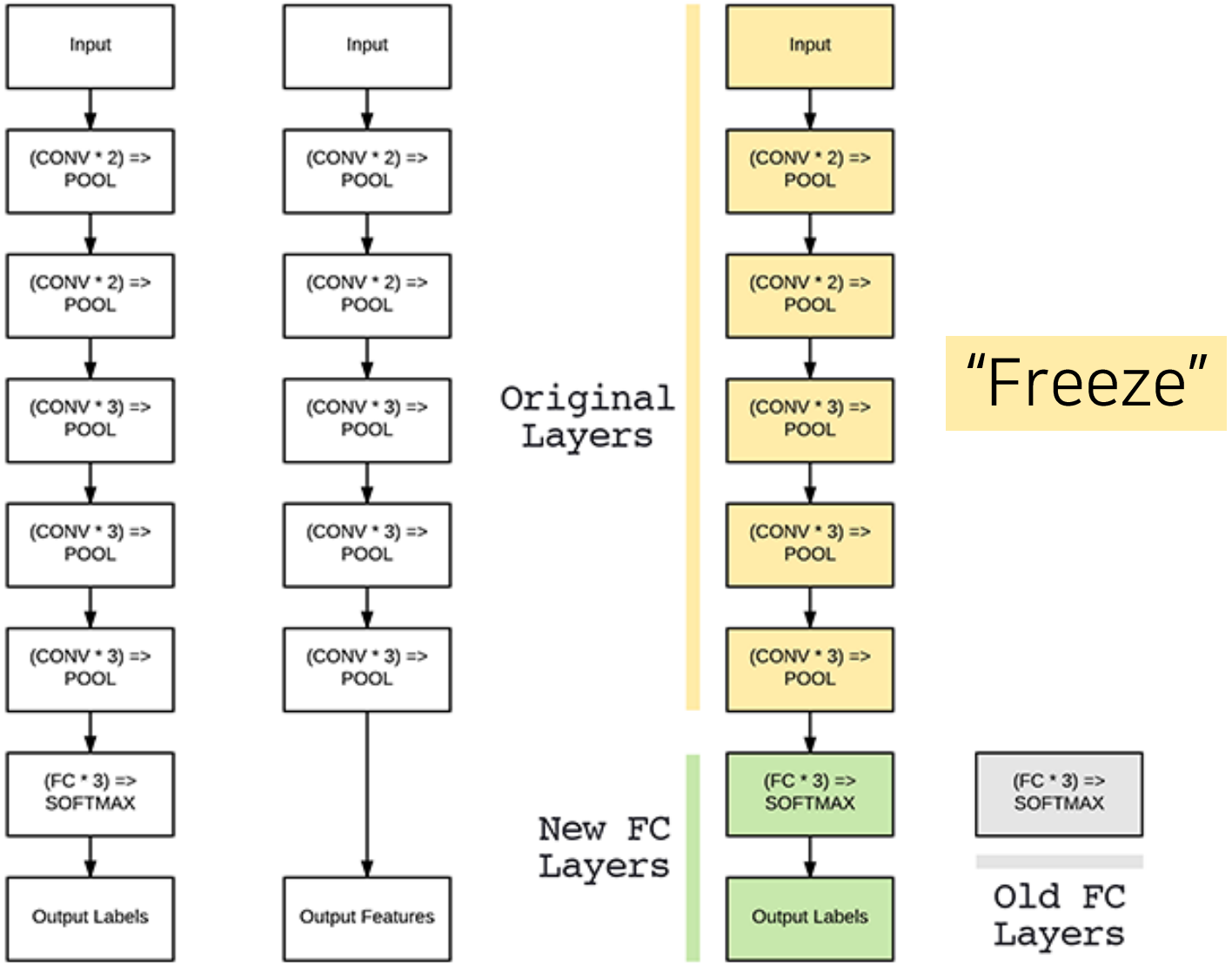
Advantages?
- Train a network with a new dataset > Cost + time loss, Not working
- Hundred of parameters

Two steps of transfer learning
1. Via feature extraction
    - Pre-trained network = Feature extractor
2. Via fine-tuning

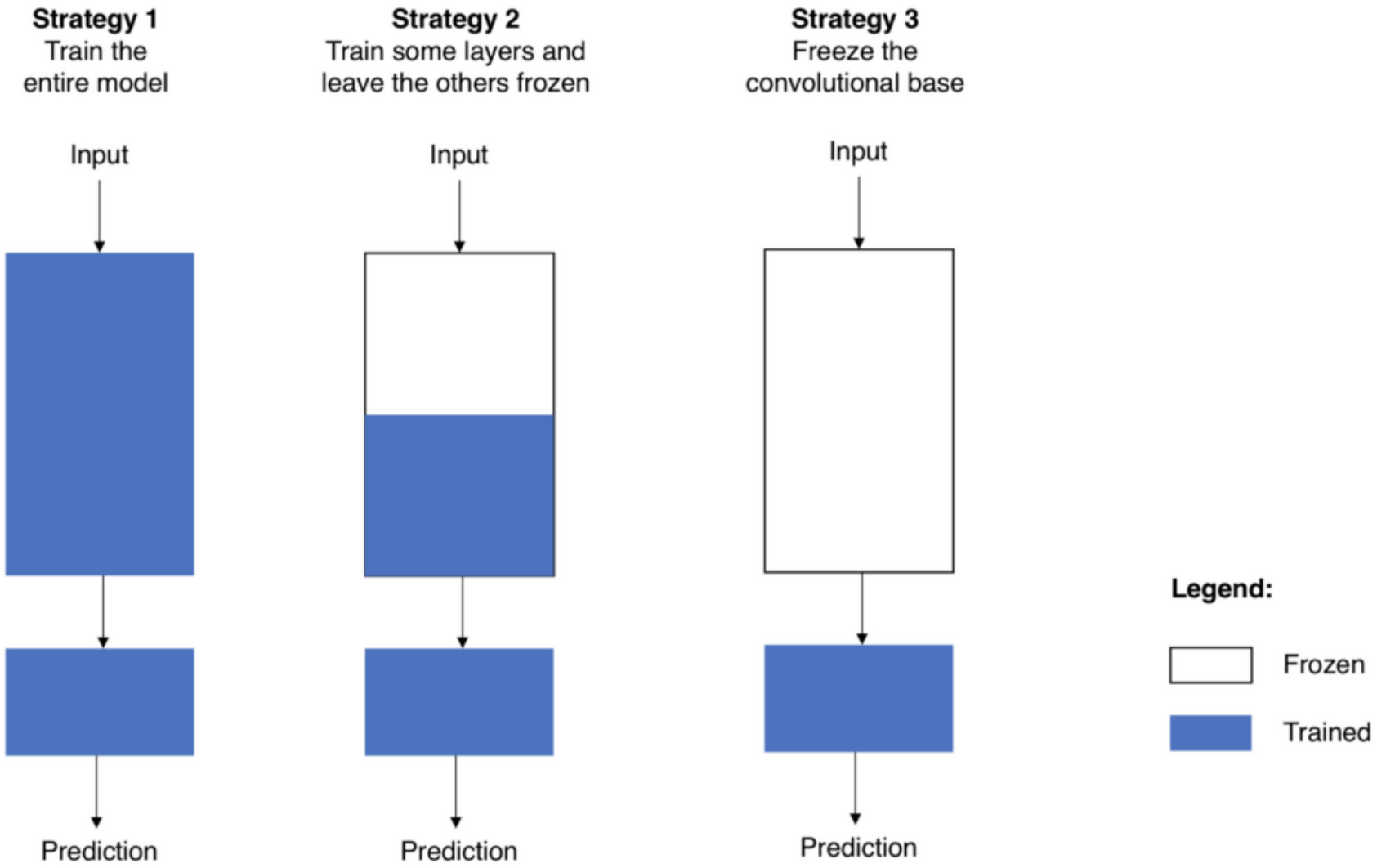
Q. 무조건 좋을까?
Feature extraction이 어떻게 되는 것에 따라 다르다. 데이터의 분포(특성)이 다를 경우

Transfer learning

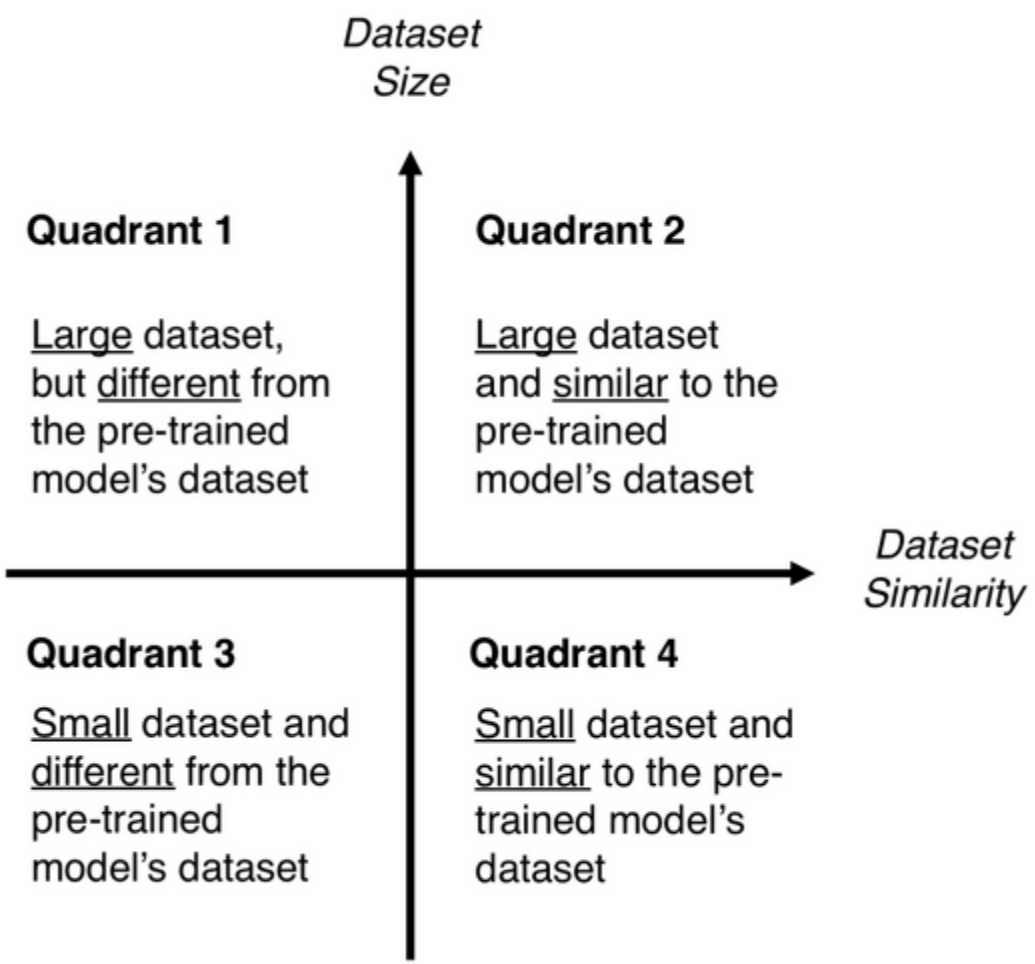
VGG16

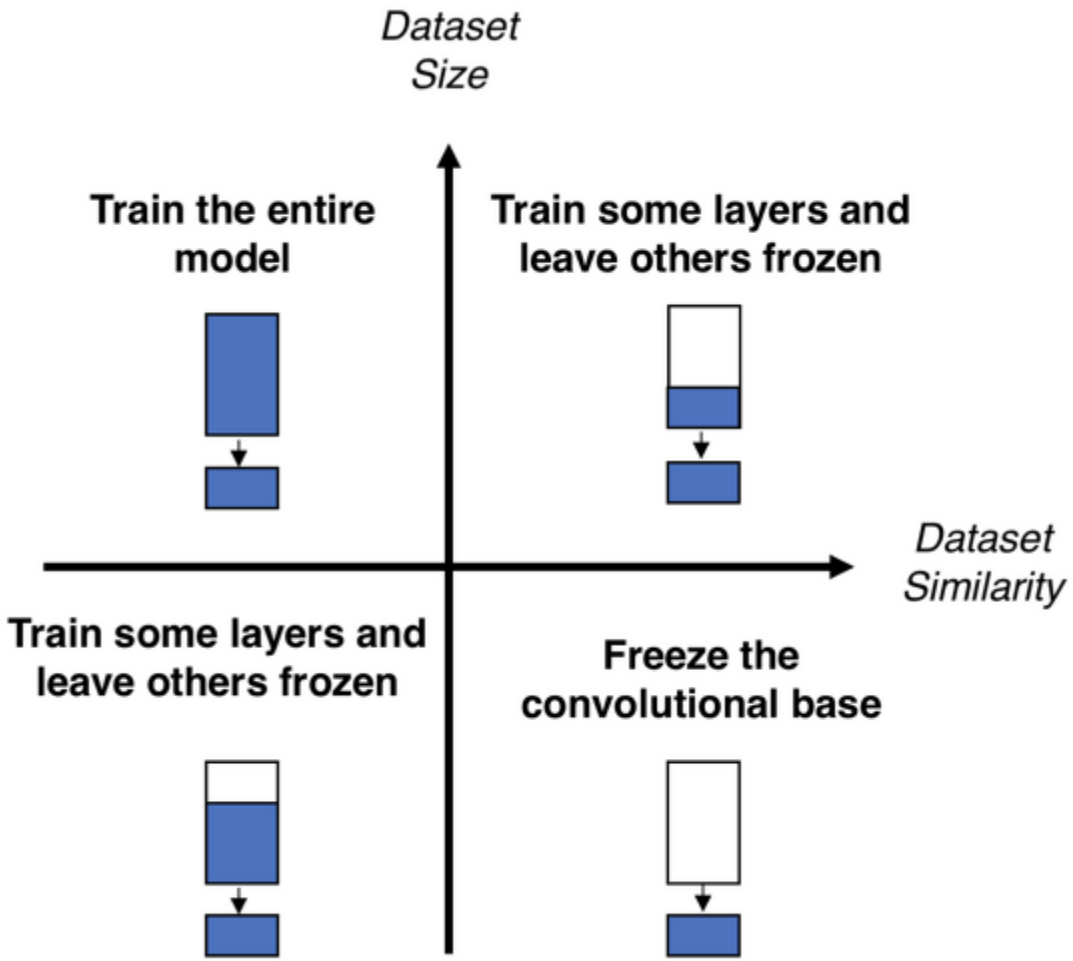


Transfer learning



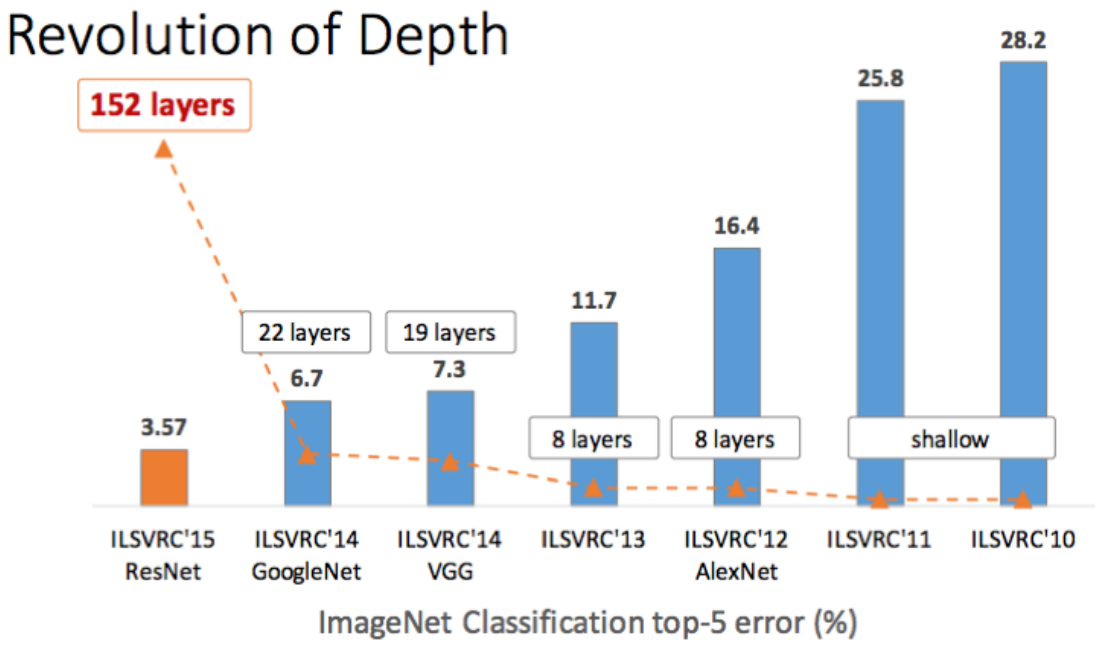
Transfer learning

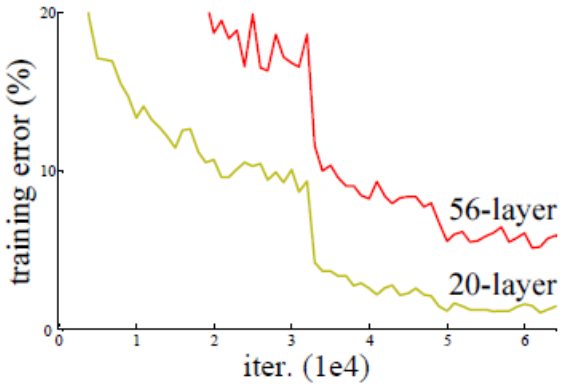


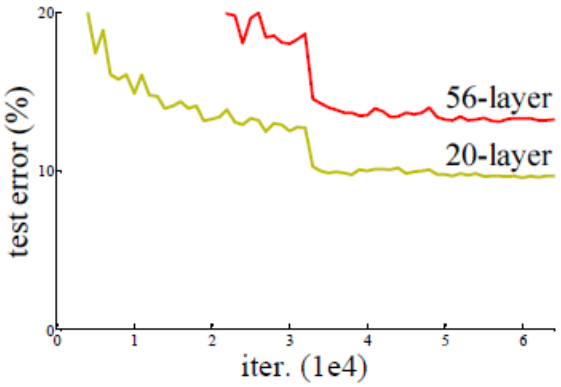


ResNet-50

- 2015 ImageNet Large Scale Visual Recognition Challenge(ILSVRC) winner.
- Developed by Microsoft
- "Deep Residual Learning for Image Recognition" (Kaiming He et al.,)
- Gradient vanishing >> Residual block (skip connection)

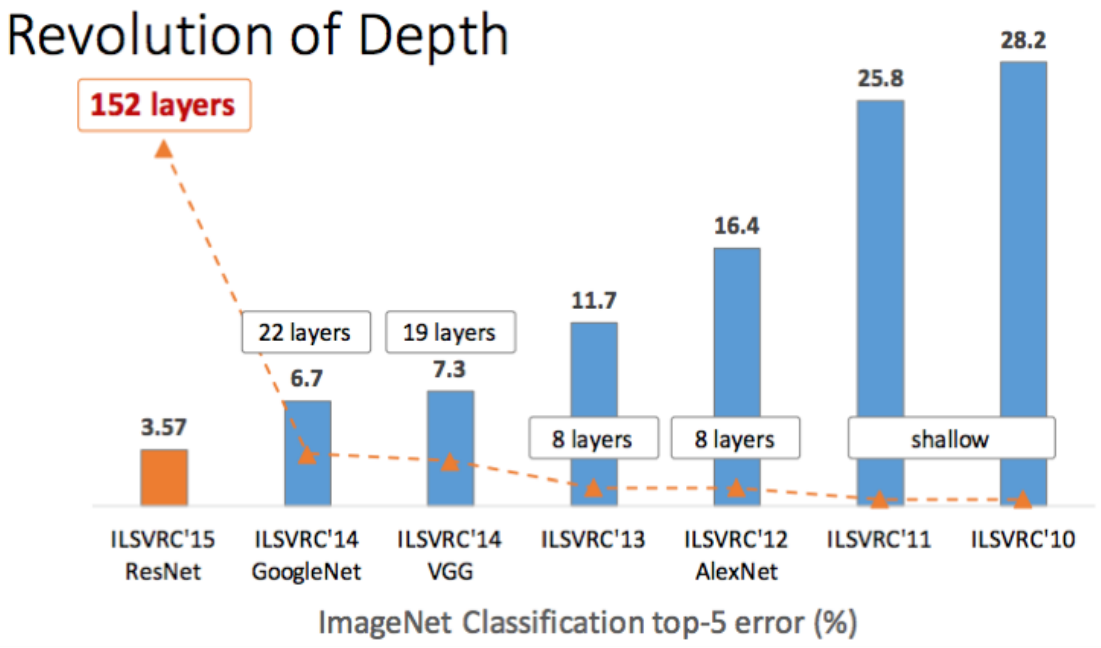


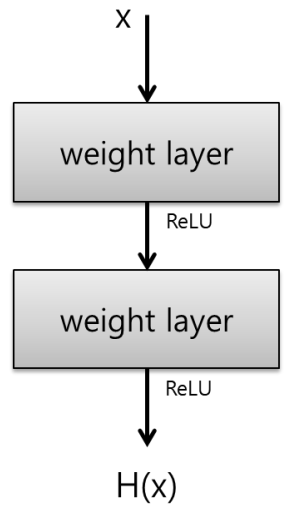




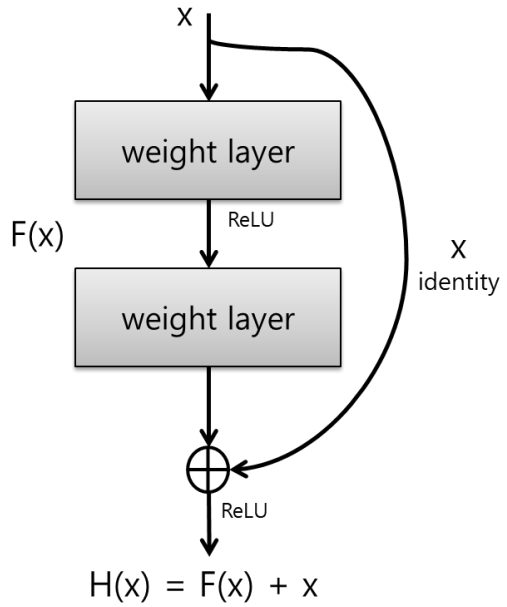
ResNet-50

- 2015 ImageNet Large Scale Visual Recognition Challenge(ILSVRC) winner.
- Developed by Microsoft
- "Deep Residual Learning for Image Recognition" (Kaiming He et al.,)
- Gradient vanishing >> Residual block (skip connection)



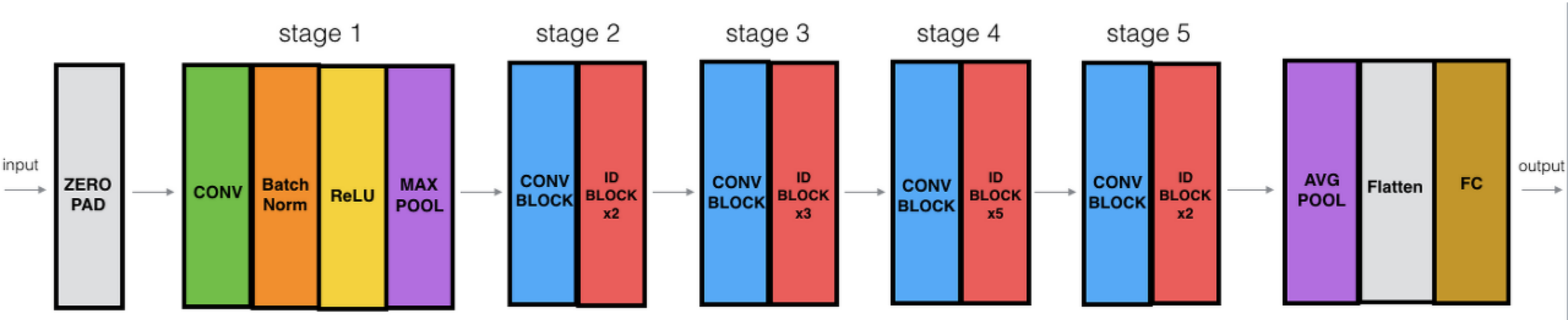


기존 방식



Residual block

ResNet-50



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
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		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9