

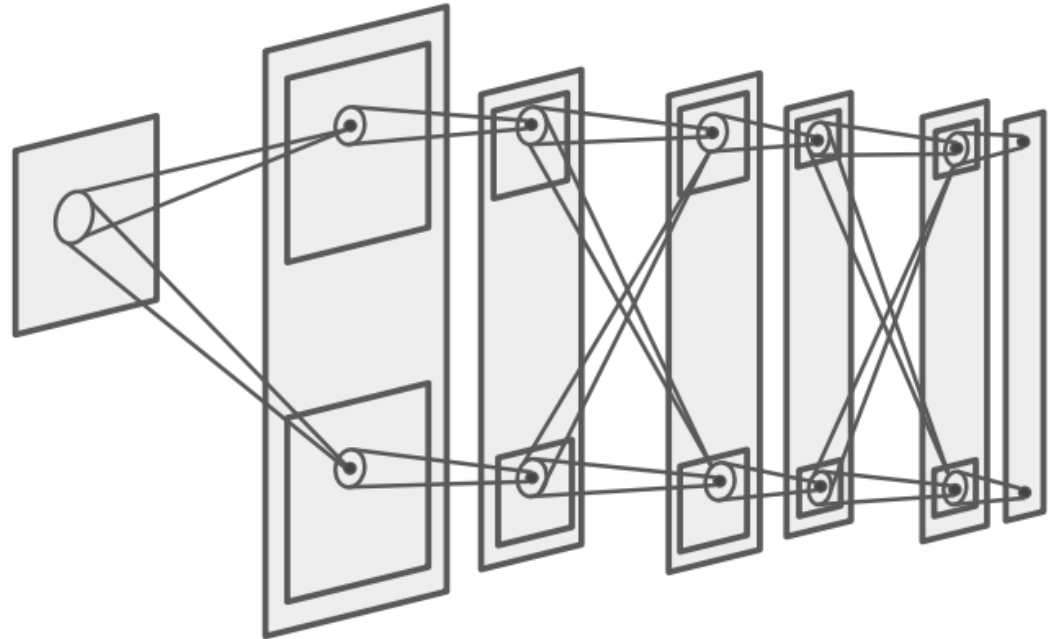
Bài 3:

Giới thiệu về mạng tích chập Conv Neural Networks

Lịch sử CNNs

Neocognitron *[Fukushima 1980]*

“sandwich” architecture (SCSCSC...)
simple cells: modifiable parameters
complex cells: perform pooling

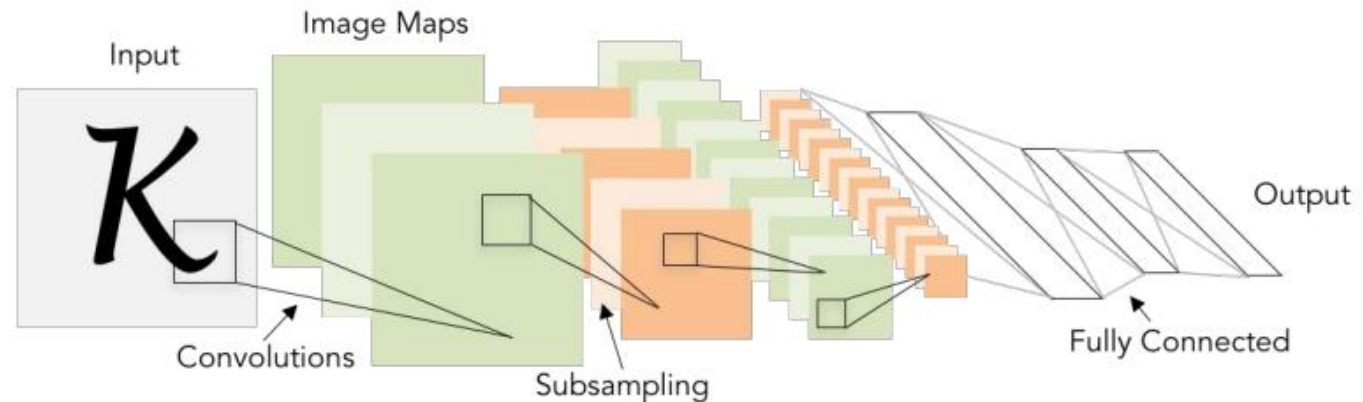


- Ý tưởng CNNs xuất phát đầu tiên từ công trình của Fukushima năm 1980

Lịch sử CNNs

Gradient-based learning applied to document recognition

[LeCun, Bottou, Bengio, Haffner 1998]



- Năm 1998, LeCun áp dụng BackProp huấn luyện mạng CNNs cho bài toán nhận dạng văn bản

Lịch sử CNNs

ImageNet Classification with Deep Convolutional Neural Networks [Krizhevsky, Sutskever, Hinton, 2012]

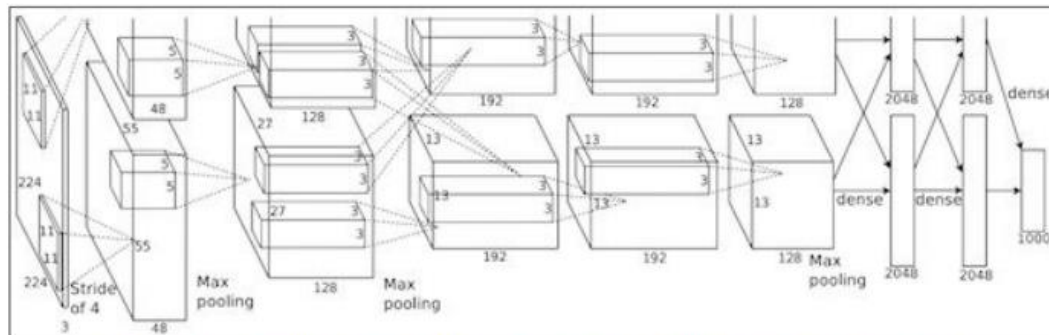


Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

“AlexNet”

- Năm 2012, CNNs gây tiếng vang lớn khi vô địch cuộc thi ILSRC 2012, vượt xa phương pháp đứng thứ 2 theo cách tiếp cận thị giác máy tính truyền thống.

Lịch sử CNNs

Classification



Retrieval

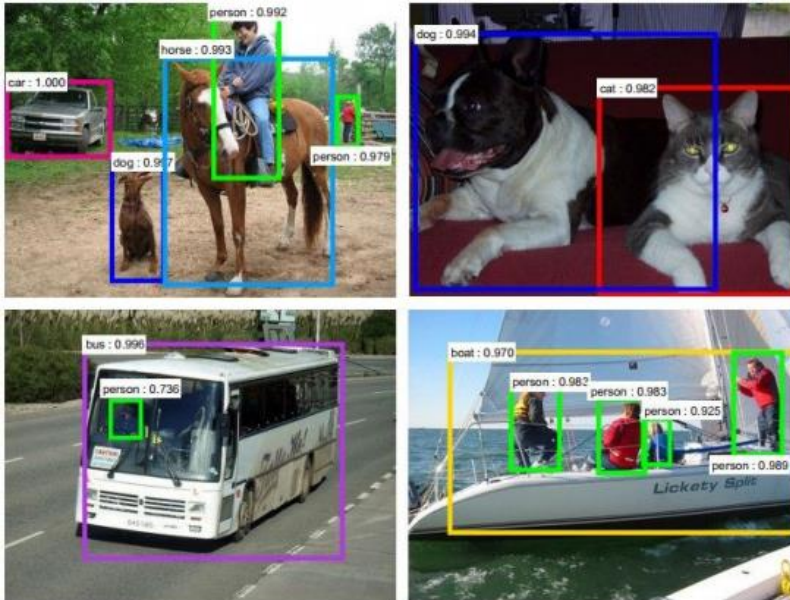


Figures copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

- Hiện nay CNNs ứng dụng khắp nơi, ví dụ trong bài toán phân loại ảnh, truy vấn ảnh

Lịch sử CNNs

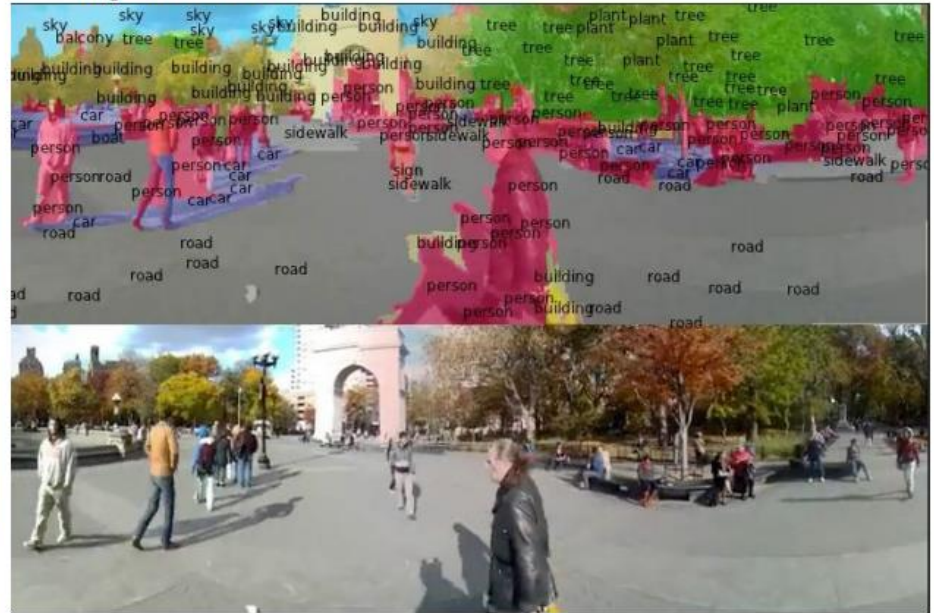
Detection



Figures copyright Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, 2015. Reproduced with permission.

[Faster R-CNN: Ren, He, Girshick, Sun 2015]

Segmentation



Figures copyright Clement Farabet, 2012.
Reproduced with permission.

[Farabet et al., 2012]

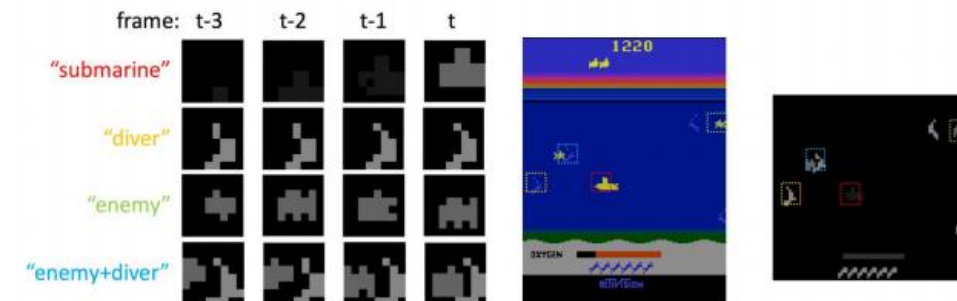
- Ứng dụng CNNs trong bài toán phát hiện đối tượng, phân đoạn ảnh

Lịch sử CNNs

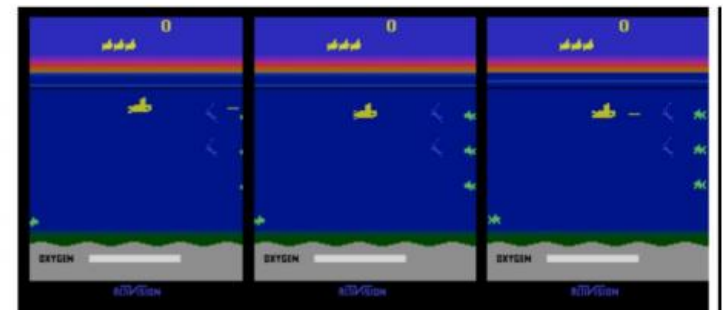


Images are examples of pose estimation, not actually from Toshev & Szegedy 2014. Copyright Lane McIntosh.

[Toshev, Szegedy 2014]



[Guo et al. 2014]

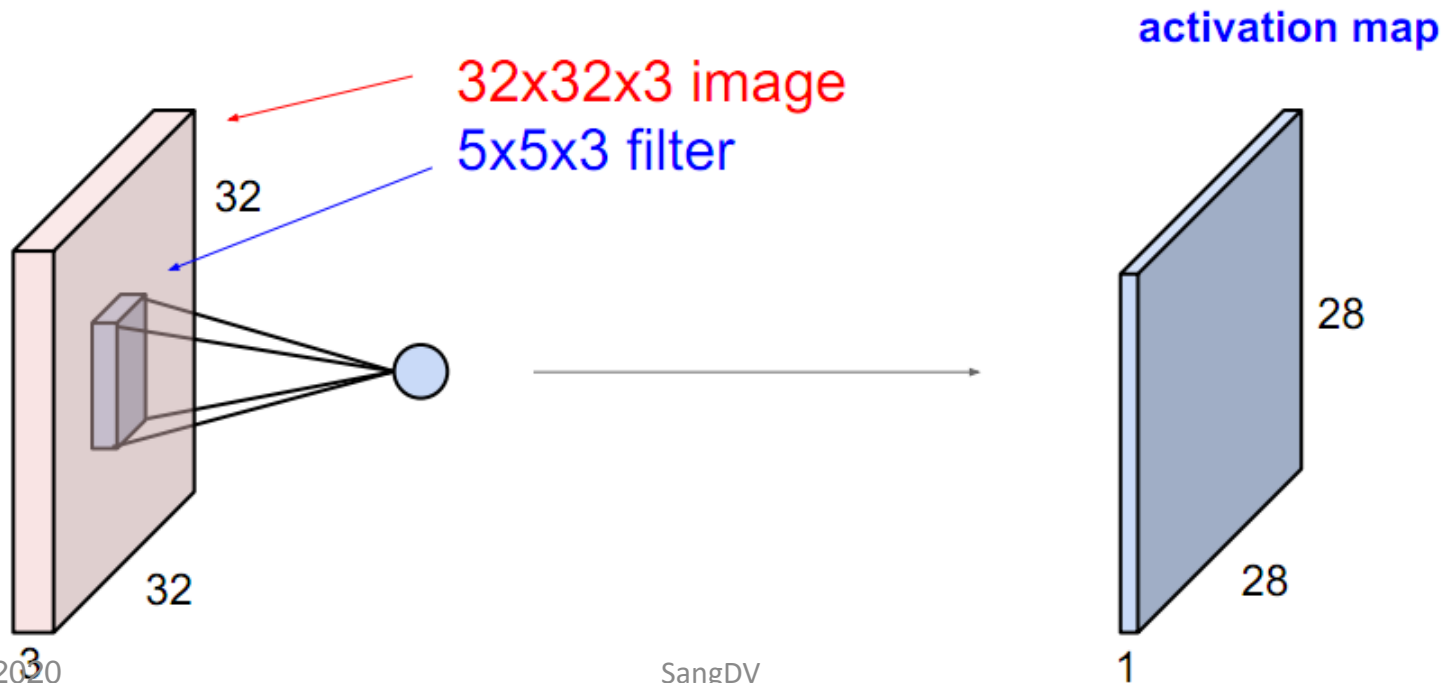


Figures copyright Xiaoxiao Guo, Satinder Singh, Honglak Lee, Richard Lewis, and Xiaoshi Wang, 2014. Reproduced with permission.

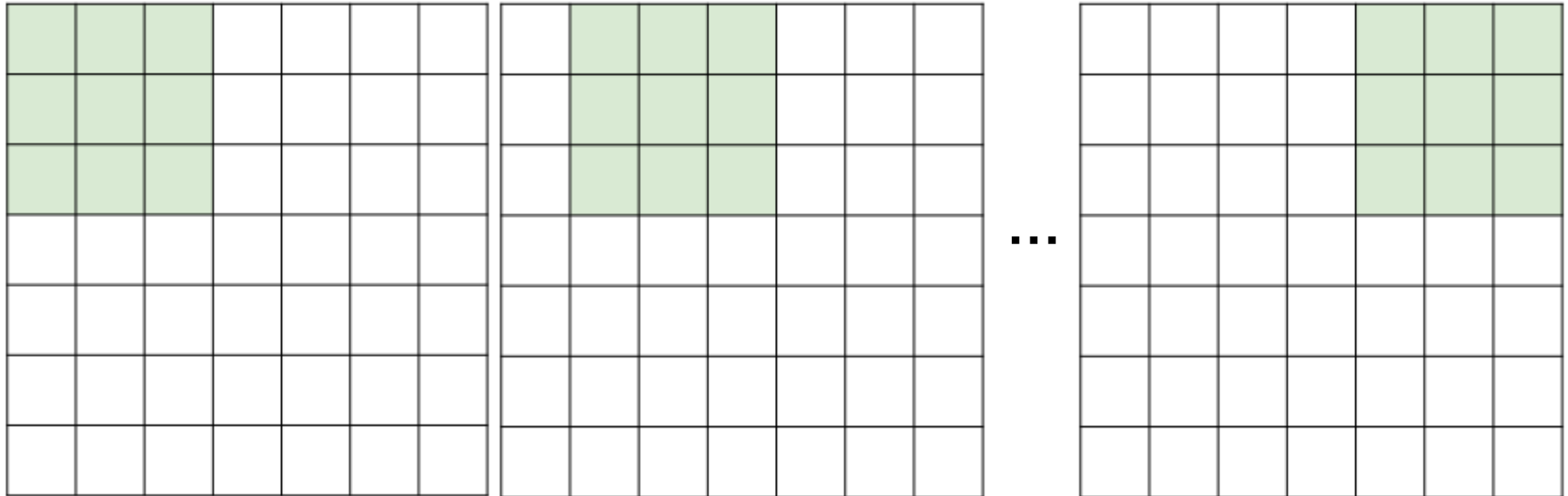
- Ứng dụng CNNs trong nhận dạng dáng người (human pose), trong trò chơi...

Lớp tích chập

- Khác với nơ-ron kết nối đầy đủ, mỗi nơ-ron tích chập (filter) chỉ kết nối cục bộ với dữ liệu đầu vào
- Nơ-ron tích chập trượt từ trái sang phải và từ trên xuống dưới khối dữ liệu đầu vào và tính toán để sinh ra một bản đồ kích hoạt (activation map)
- Chiều sâu của nơ-ron tích chập bằng chiều sâu của khối dữ liệu đầu vào

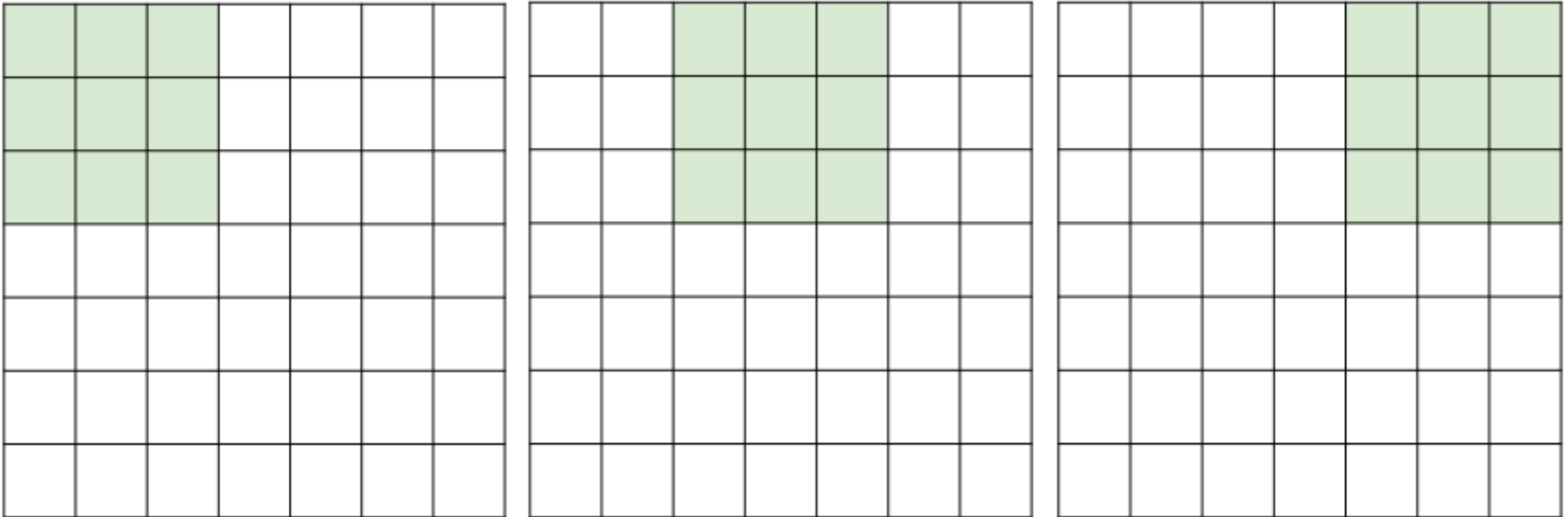


Lớp tích chập



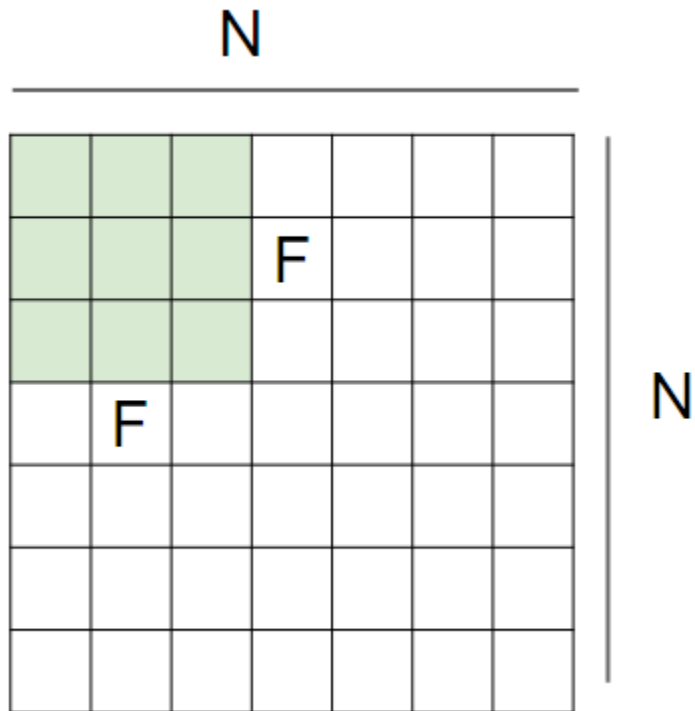
- Bước nhảy stride = 1
- Đầu vào kích thước 7x7, nơ-ron kích thước 3x3
- Đầu ra kích thước 5x5

Lớp tích chập



- Bước nhảy stride = 2
- Đầu vào kích thước 7x7, nơ-ron kích thước 3x3
- Đầu ra kích thước 3x3

Lớp tích chập



Output size:

$$(N - F) / \text{stride} + 1$$

e.g. $N = 7, F = 3$:

stride 1 $\Rightarrow (7 - 3)/1 + 1 = 5$

stride 2 $\Rightarrow (7 - 3)/2 + 1 = 3$

stride 3 $\Rightarrow (7 - 3)/3 + 1 = 2.33 \therefore \backslash$

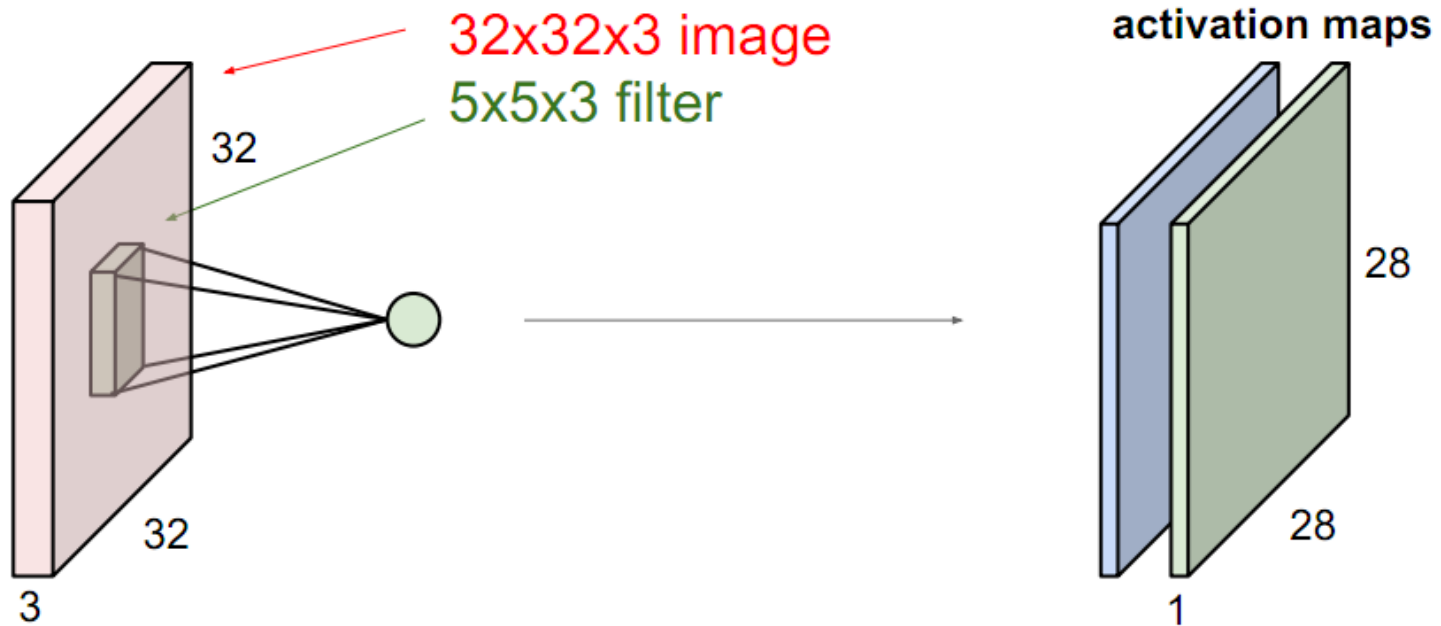
Lớp tích chập

- Để bảo toàn kích thước thường thêm viền bởi các số 0 (zero padding).
- Ví dụ: đầu vào kích thước 7×7 , nơ-ron kích thước 3×3 , bước nhảy stride 1, padding viền độ rộng 1.
- Khi đó kích thước đầu ra là 7×7

0	0	0	0	0	0			
0								
0								
0								
0								

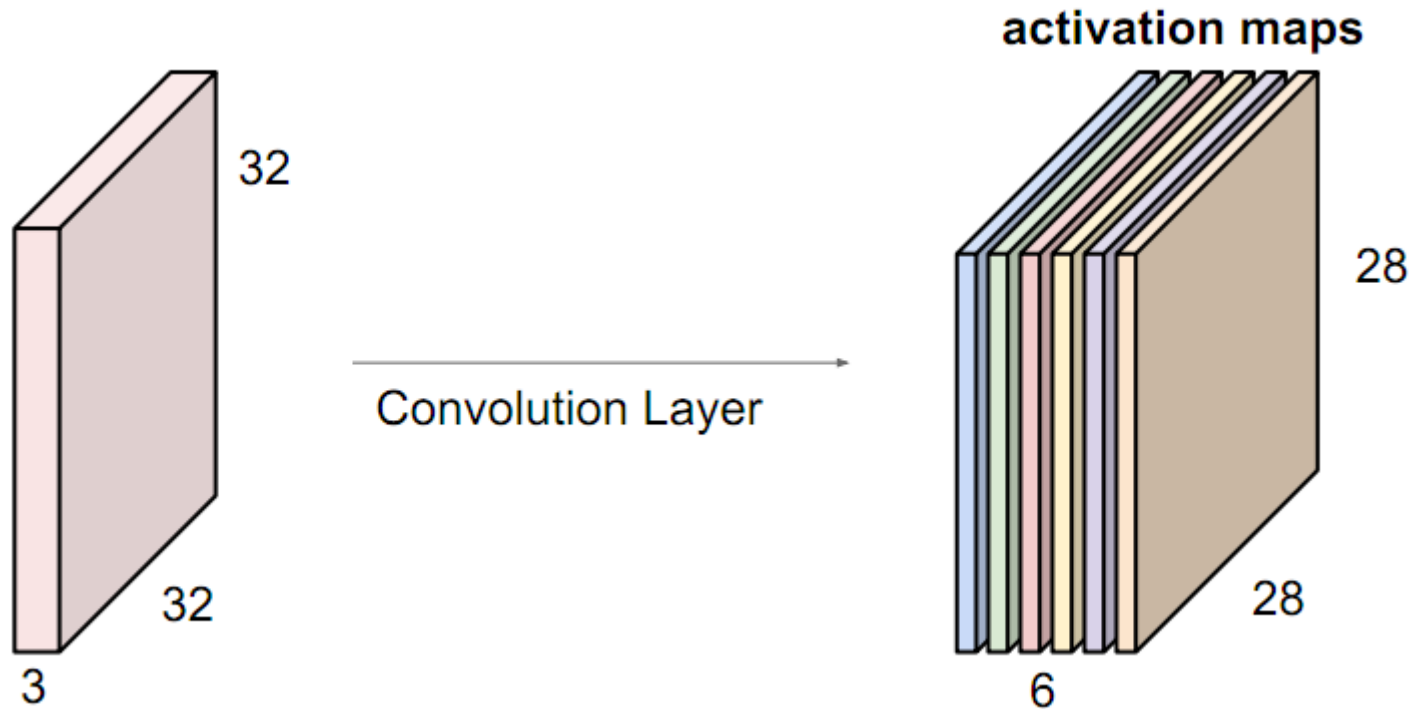
Lớp tích chập

- Giả sử có thêm nơ-ron tích chập khác thì nó cũng hoạt động tương tự và sinh ra bản đồ kích hoạt thứ hai
- Lưu ý trọng số của các nơ-ron tích chập là khác nhau



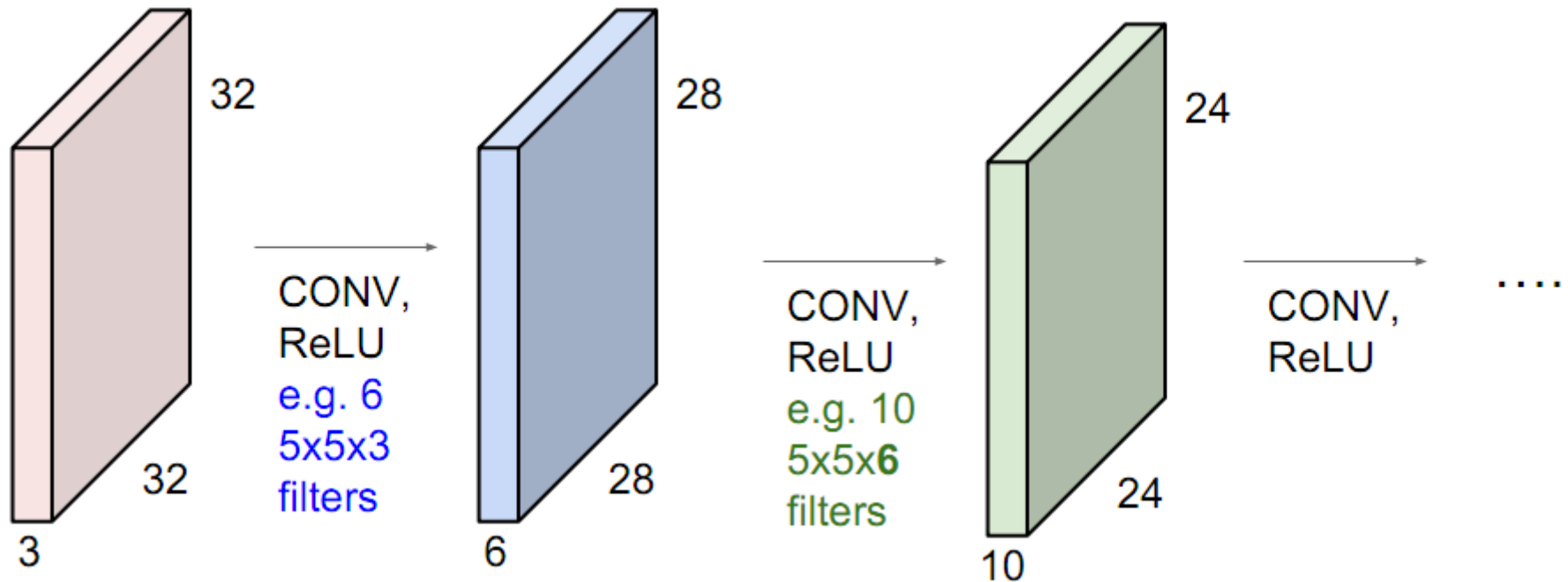
Lớp tích chập

- Giả sử có 6 nơ-ron tích chập sẽ sinh ra 6 bản đồ kích hoạt
- Các bản đồ kích hoạt ghép với nhau thành một “ảnh mới”



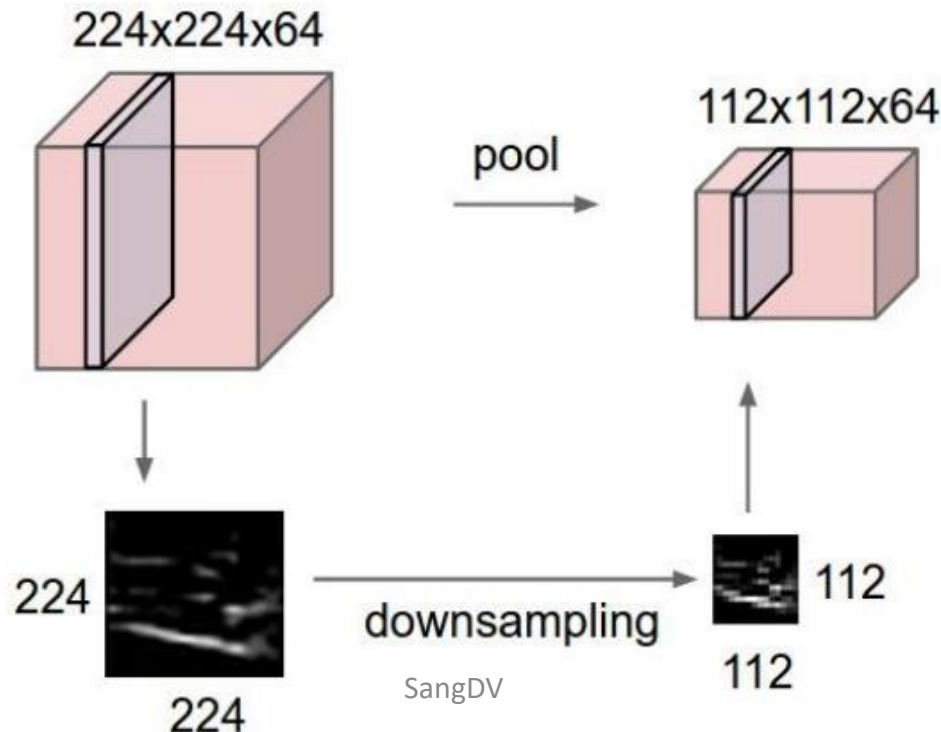
CNNs

- Mạng nơ-ron tích chập là một dãy các lớp tích chập nối liền tiếp nhau xen kẽ bởi các hàm kích hoạt (ví dụ ReLU)



Lớp gộp (pooling layer)

- Giúp giảm độ phân giải của khối dữ liệu để giảm bộ nhớ và khối lượng tính toán
- Hoạt động độc lập trên từng bản đồ kích hoạt
- Lớp gộp max pooling giúp mạng biểu diễn bất biến đối với các thay đổi tịnh tiến (translation invariance) hoặc biến dạng (deformation invariance) của dữ liệu đầu vào



Lớp gộp max pooling

Single depth slice

x

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

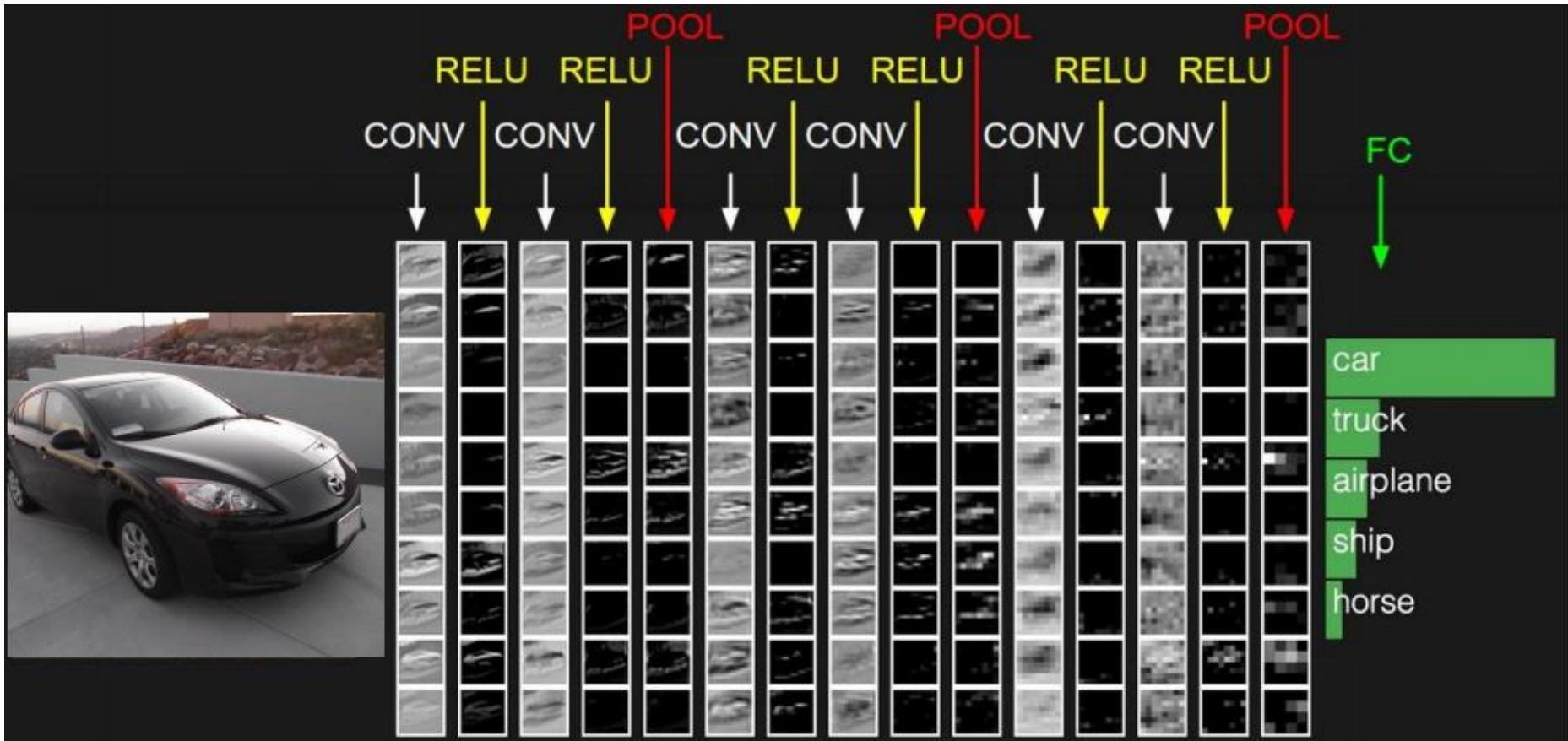
y

max pool with 2x2 filters
and stride 2



6	8
3	4

CNNs

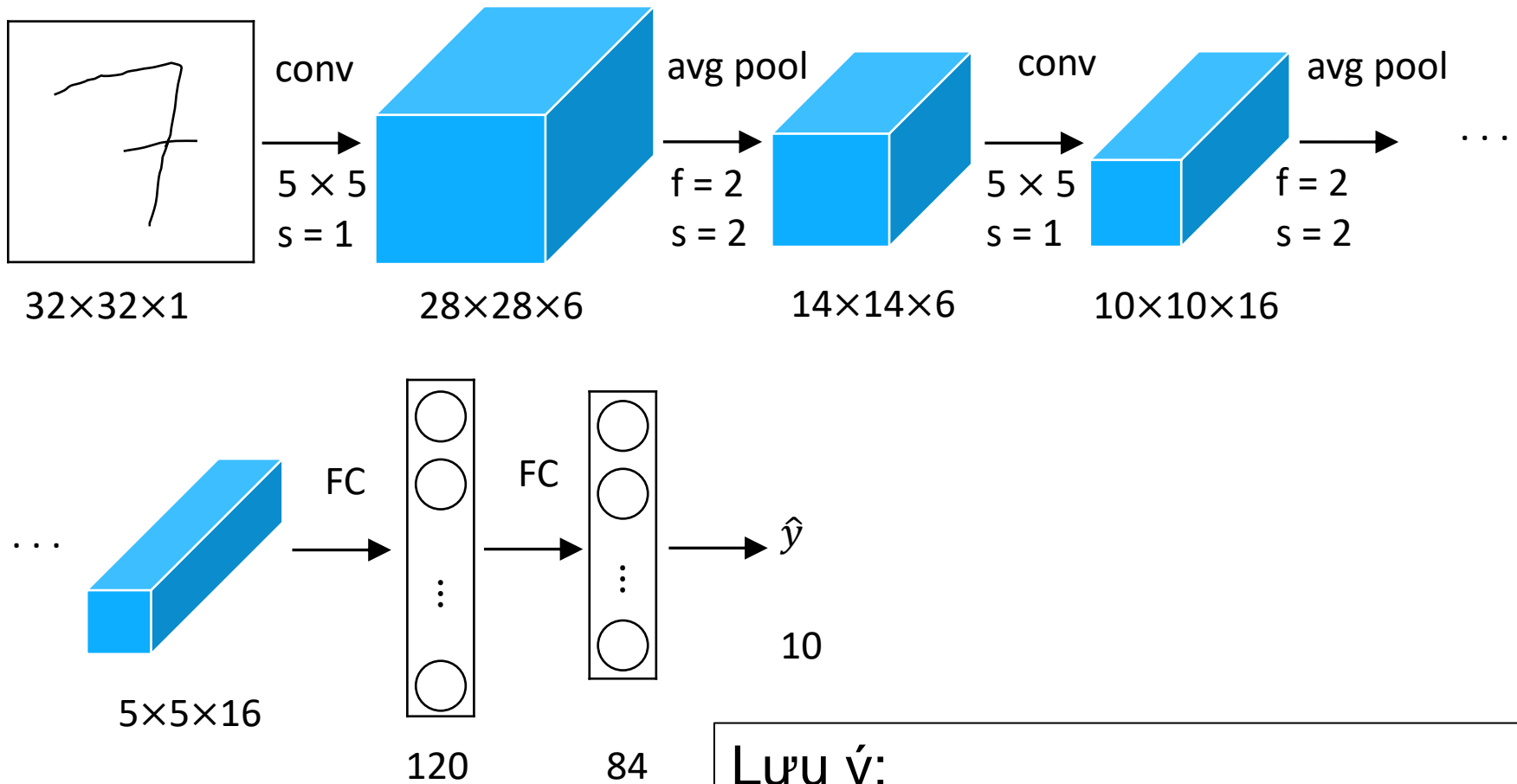


Một số mạng CNNs cơ bản



- LeNet-5
- AlexNet
- VGG
- GoogleNet
- ResNet

LeNet-5



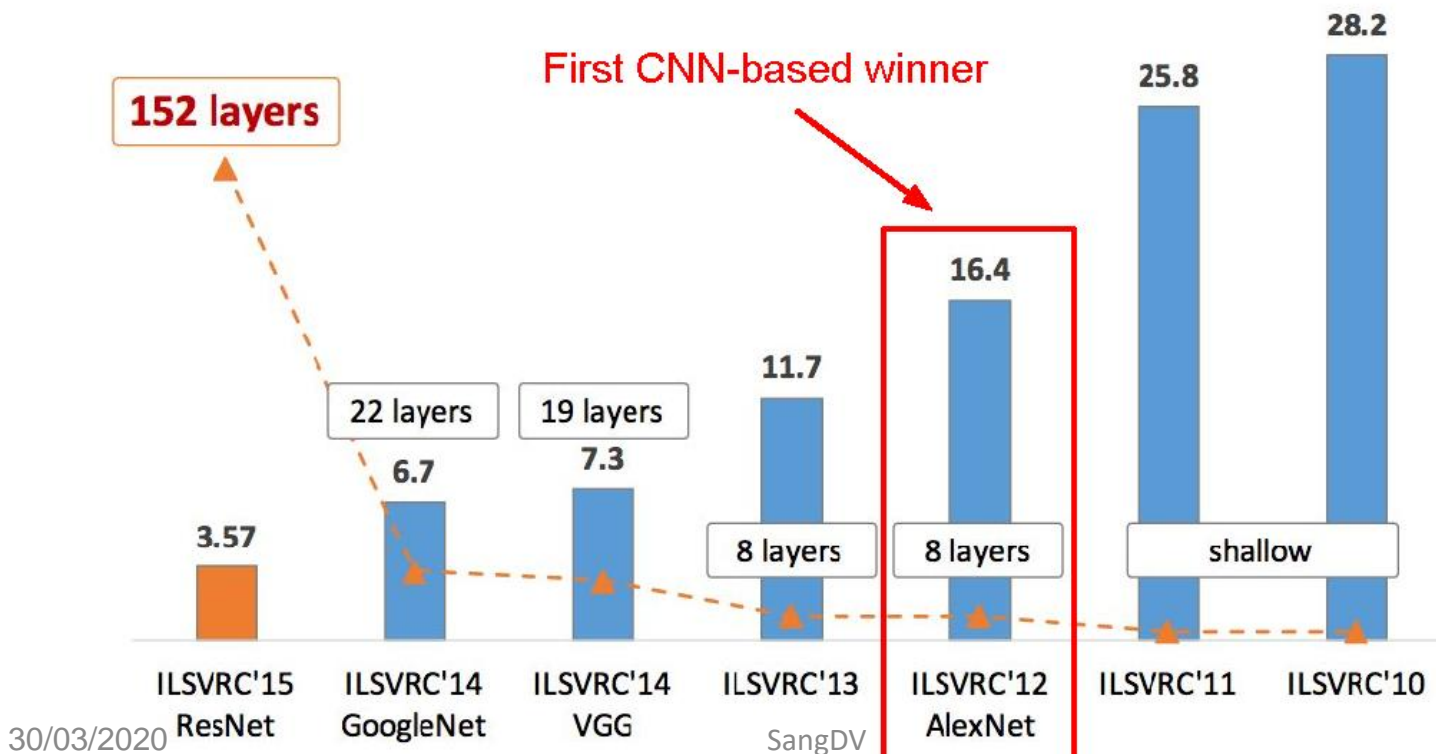
Lưu ý:
Output size = $(N+2P-F)/\text{stride} + 1$

AlexNet

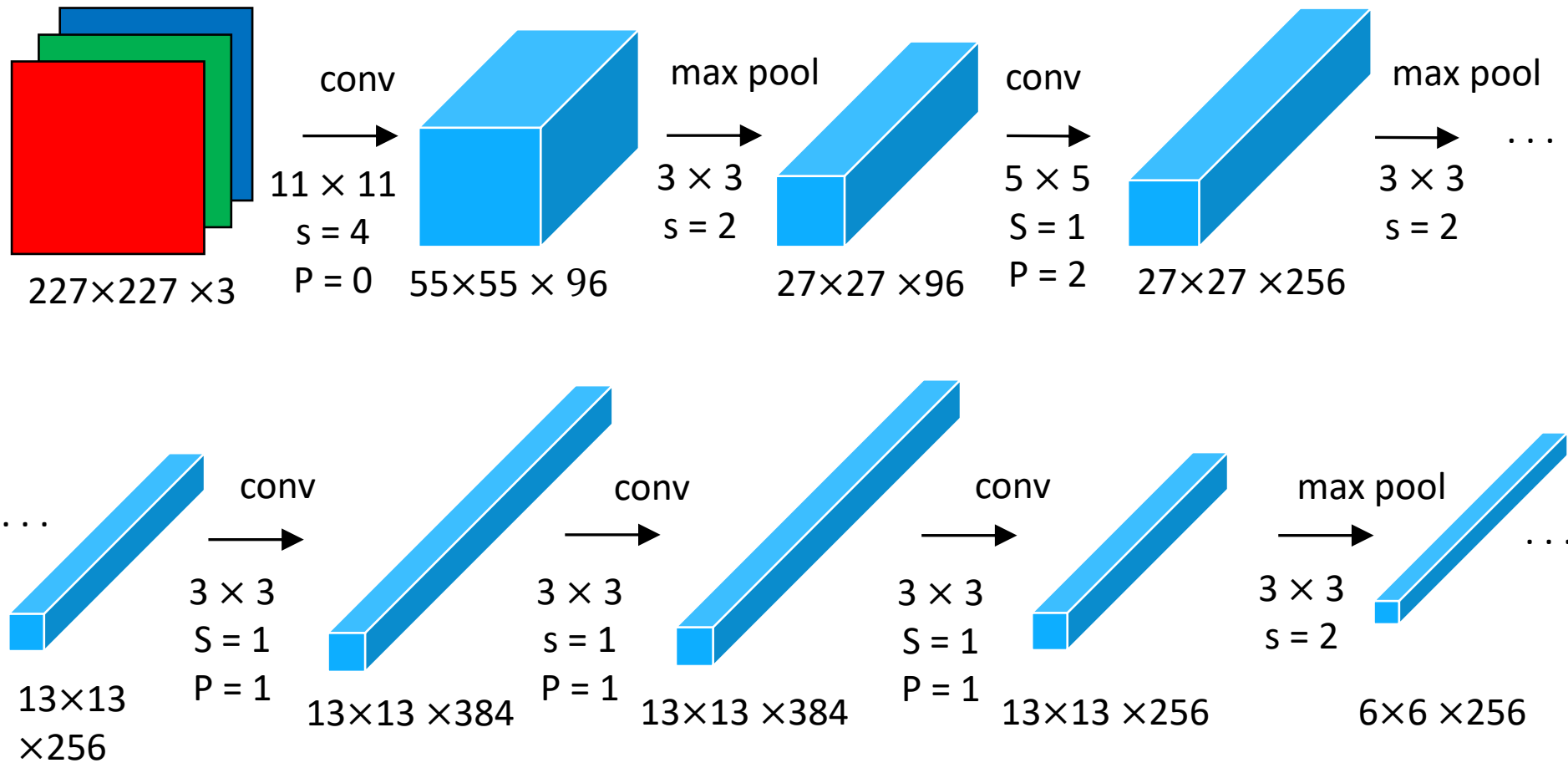
- ImageNet Classification with Deep Convolutional Neural Networks - Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton; 2012
- Một trong những mạng CNNs lớn nhất tại thời điểm đó
- Có 60M tham số so với 60k tham số LeNet-5

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

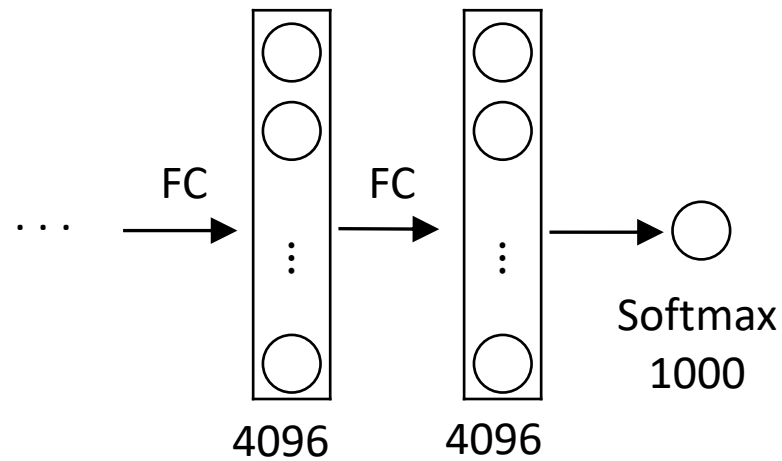
- “Olympics” thường niên về lĩnh vực thị giác máy tính.
- Các teams khắp thế giới thi đấu với nhau để xem ai là người có mô hình CV tốt nhất cho các bài toán như phân loại ảnh, định vị và phát hiện đối tượng trong ảnh



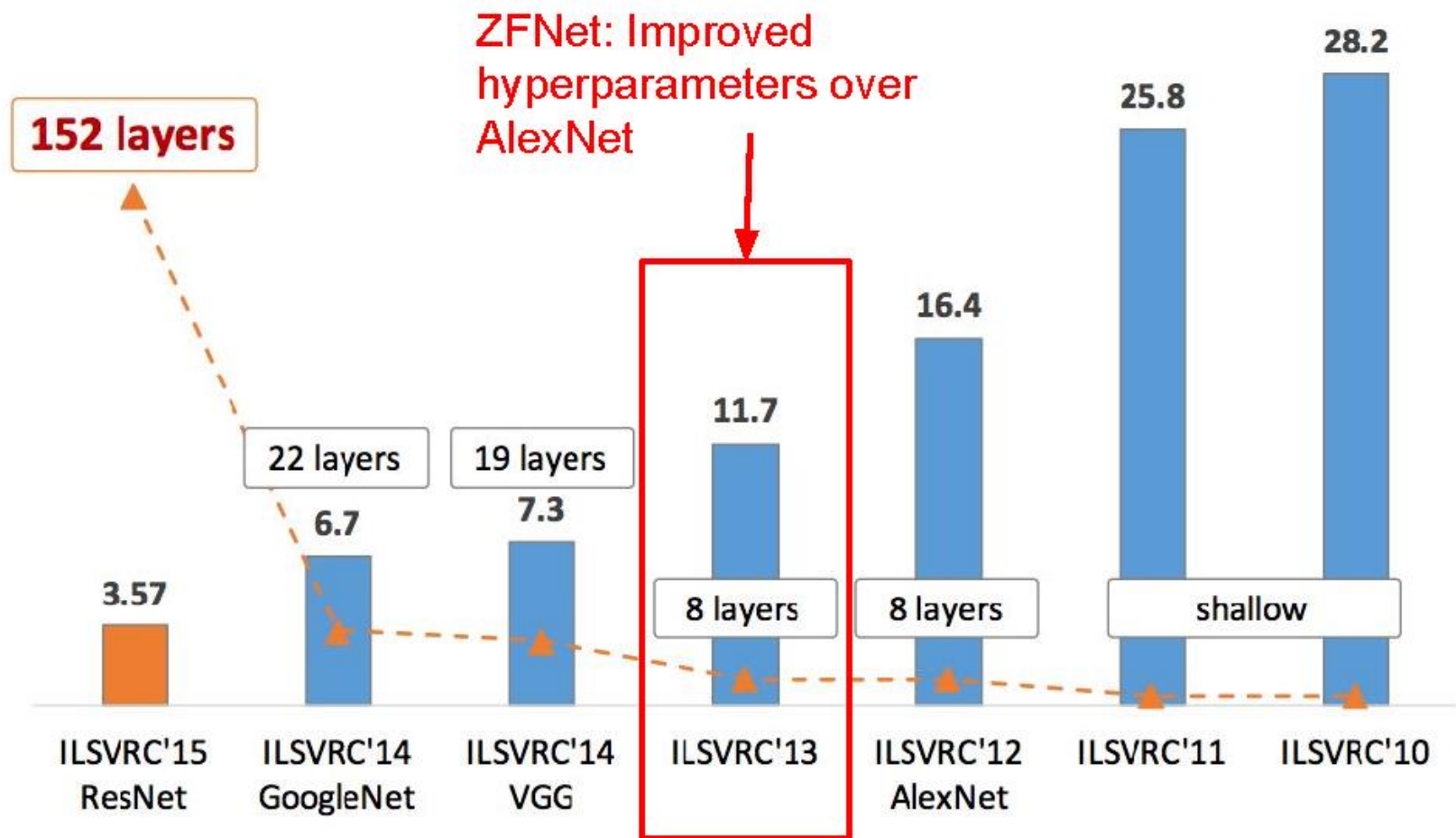
AlexNet



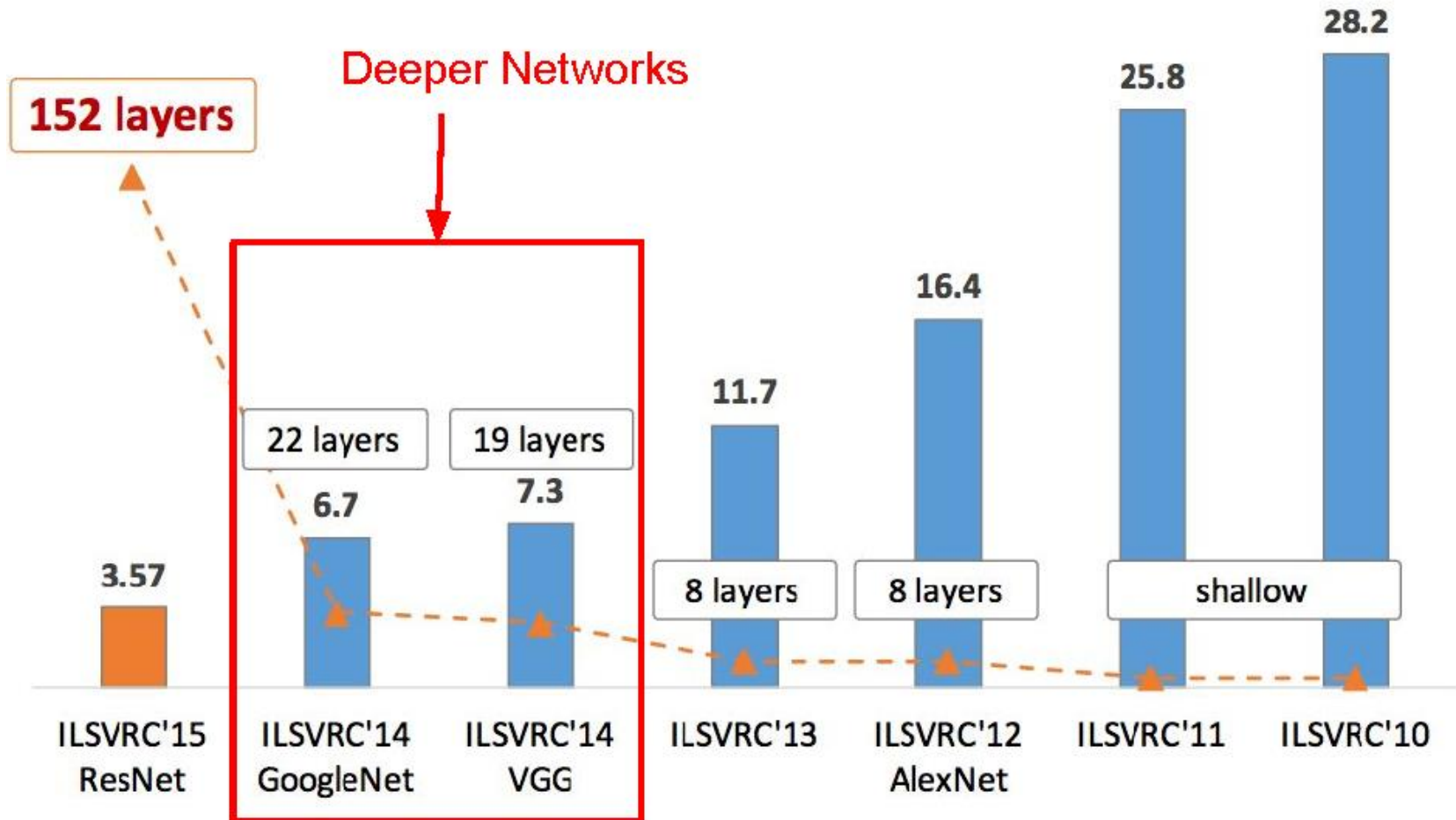
AlexNet



ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



VGGNet

- Very Deep Convolutional Networks For Large Scale Image Recognition - Karen Simonyan and Andrew Zisserman; 2015
- Á quân tại cuộc thi ILSVRC 2014
- Sâu hơn rất nhiều so với AlexNet
- 140 triệu tham số

Input

3x3 conv, 64

3x3 conv, 64

Pool 1/2

3x3 conv, 128

3x3 conv, 128

Pool 1/2

3x3 conv, 256

3x3 conv, 256

Pool 1/2

3x3 conv, 512

3x3 conv, 512

3x3 conv, 512

Pool 1/2

3x3 conv, 512

3x3 conv, 512

3x3 conv, 512

Pool 1/2

FC 4096

FC 4096

FC 1000

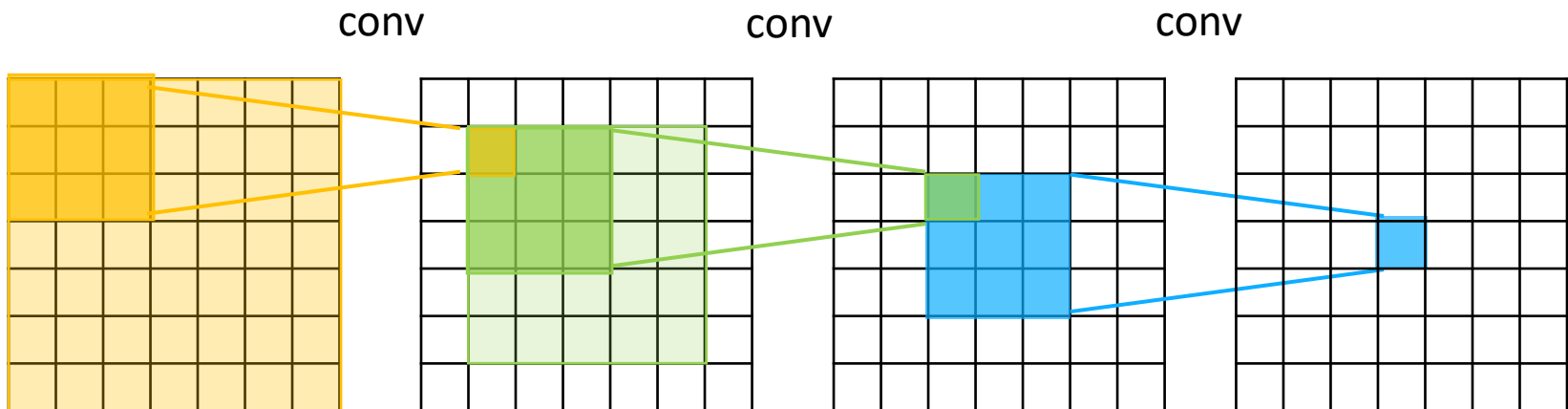
Softmax

VGGNet

- Neuron kích thước bé
Chỉ dùng conv 3x3, stride 1, pad 1
và 2x2 MAX POOL, stride 2
- Mạng sâu hơn
AlexNet: 8 lớp
VGGNet: 16 - 19 lớp
- ZFNet: 11.7% top 5 error in ILSVRC'13
- VGGNet: 7.3% top 5 error in ILSVRC'14

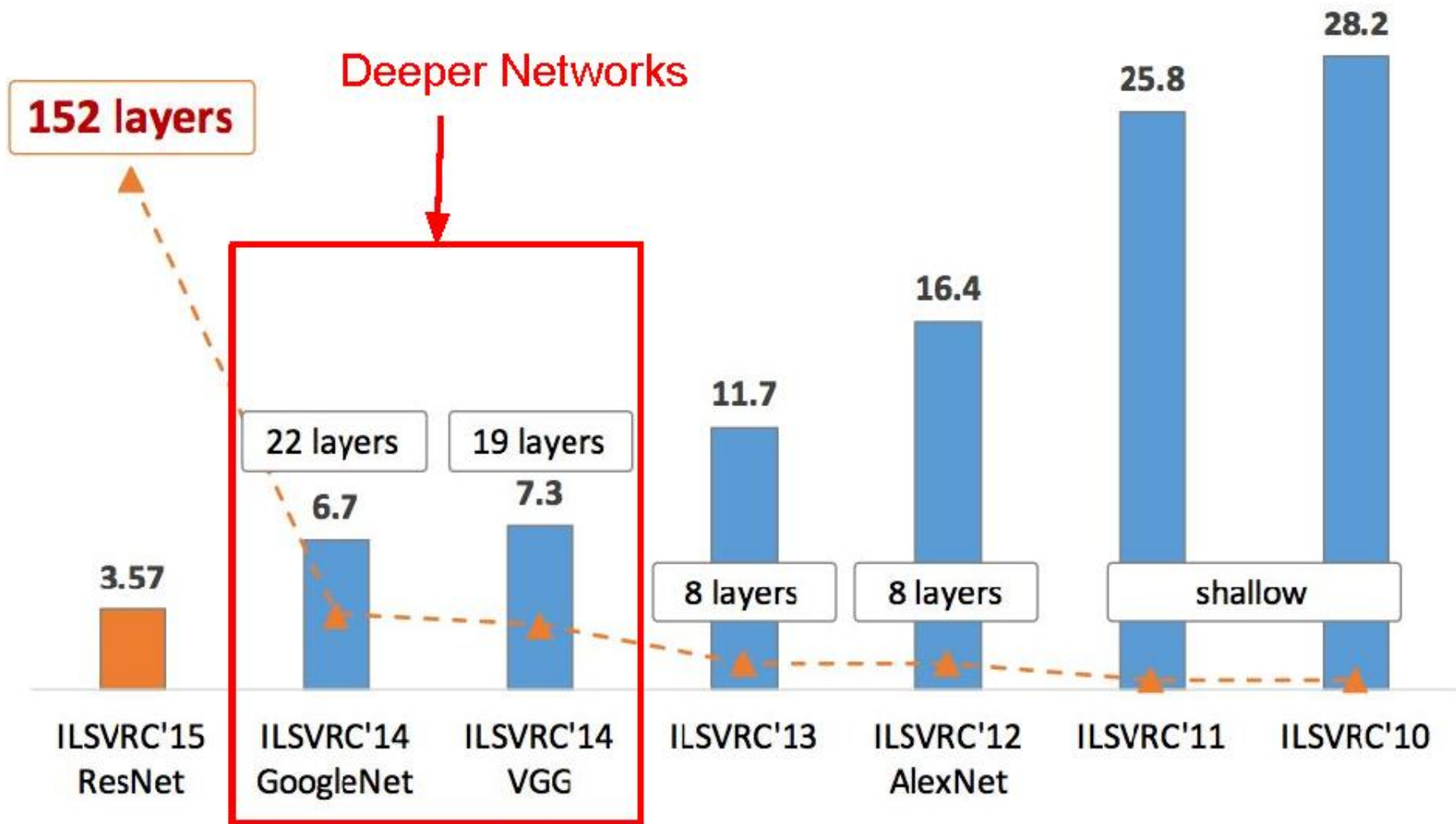
VGGNet

- Tại sao dùng filter bé? (3x3 conv)
- Chồng 3 lớp 3x3 conv (stride 1) có cùng hiệu quả thu nhận thông tin như một lớp 7x7 conv.
- Nhưng sâu hơn, nhiều lớp phi tuyến hơn
- Và ít tham số hơn: $3 * (3^2 C^2)$ vs. $7^2 C^2$ với C là số kênh của mỗi lớp



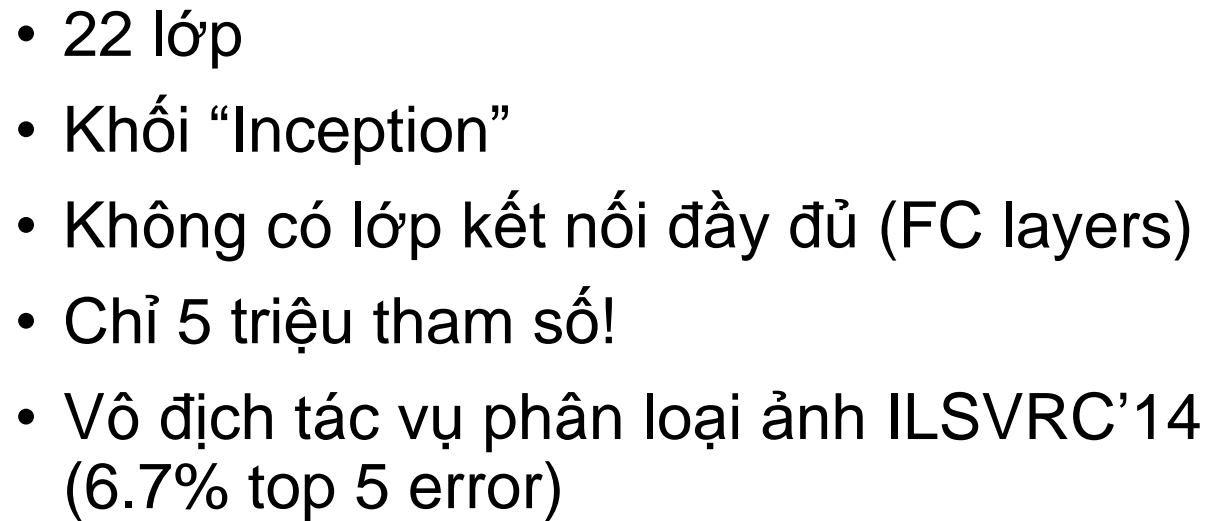
Input	memory: $224*224*3=150K$	params: 0
3x3 conv, 64	memory: $224*224*64=3.2M$	params: $(3*3*3)*64 = 1,728$
3x3 conv, 64	memory: $224*224*64=3.2M$	params: $(3*3*64)*64 = 36,864$
Pool	memory: $112*112*64=800K$	params: 0
3x3 conv, 128	memory: $112*112*128=1.6M$	params: $(3*3*64)*128 = 73,728$
3x3 conv, 128	memory: $112*112*128=1.6M$	params: $(3*3*128)*128 = 147,456$
Pool	memory: $56*56*128=400K$	params: 0
3x3 conv, 256	memory: $56*56*256=800K$	params: $(3*3*128)*256 = 294,912$
3x3 conv, 256	memory: $56*56*256=800K$	params: $(3*3*256)*256 = 589,824$
3x3 conv, 256	memory: $56*56*256=800K$	params: $(3*3*256)*256 = 589,824$
Pool	memory: $28*28*256=200K$	params: 0
3x3 conv, 512	memory: $28*28*512=400K$	params: $(3*3*256)*512 = 1,179,648$
3x3 conv, 512	memory: $28*28*512=400K$	params: $(3*3*512)*512 = 2,359,296$
3x3 conv, 512	memory: $28*28*512=400K$	params: $(3*3*512)*512 = 2,359,296$
Pool	memory: $14*14*512=100K$	params: 0
3x3 conv, 512	memory: $14*14*512=100K$	params: $(3*3*512)*512 = 2,359,296$
3x3 conv, 512	memory: $14*14*512=100K$	params: $(3*3*512)*512 = 2,359,296$
3x3 conv, 512	memory: $14*14*512=100K$	params: $(3*3*512)*512 = 2,359,296$
Pool	memory: $7*7*512=25K$	params: 0
FC 4096	memory: 4096	params: $7*7*512*4096 = 102,760,448$
FC 4096	memory: 4096	params: $4096*4096 = 16,777,216$
FC 1000	memory: 1000	params: $4096*1000 = 4,096,000$

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



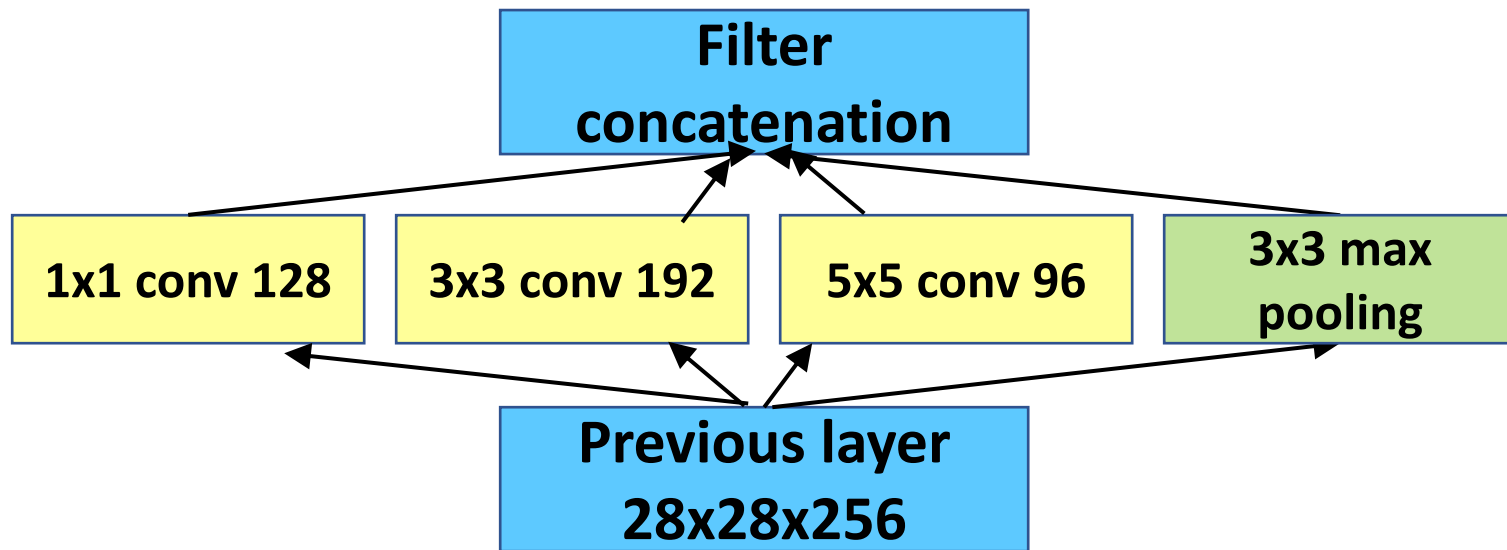
GoogleNet

- Going Deeper with Convolutions - Christian Szegedy et al.; 2015
- Vô địch ILSVRC 2014
- Sâu hơn nhiều so với AlexNet
- Số tham số ít hơn 12 lần so với AlexNet
- Tập trung vào giảm độ phức tạp tính toán



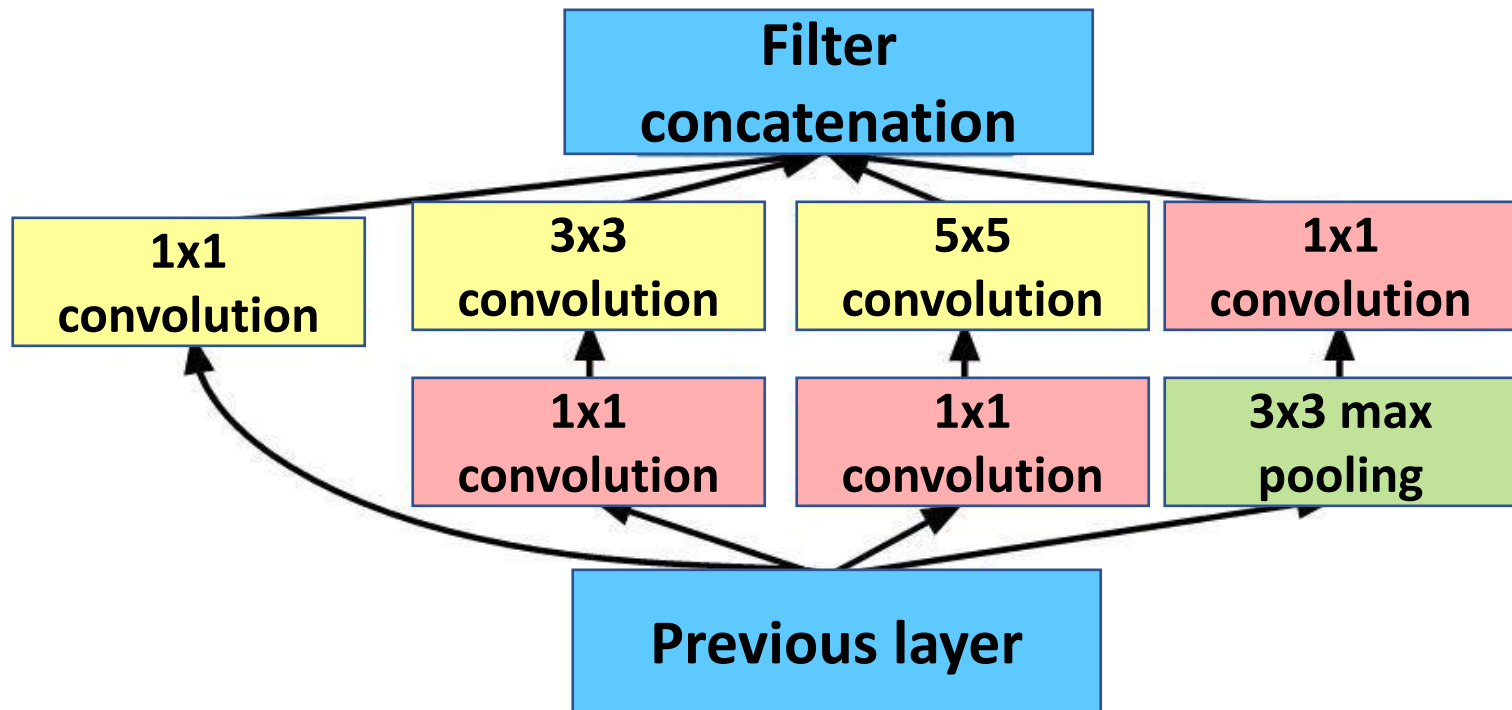
GoogleNet - Naïve Inception Model

- Số lượng phép tích chập:
- 1x1 conv, 128: $28 \times 28 \times 128 \times 1 \times 1 \times 256$
- 3x3 conv, 192: $28 \times 28 \times 192 \times 3 \times 3 \times 256$
- 5x5 conv, 96: $28 \times 28 \times 96 \times 5 \times 5 \times 256$
- Tổng cộng: 854M ops ==> **Tính toán rất nặng!**

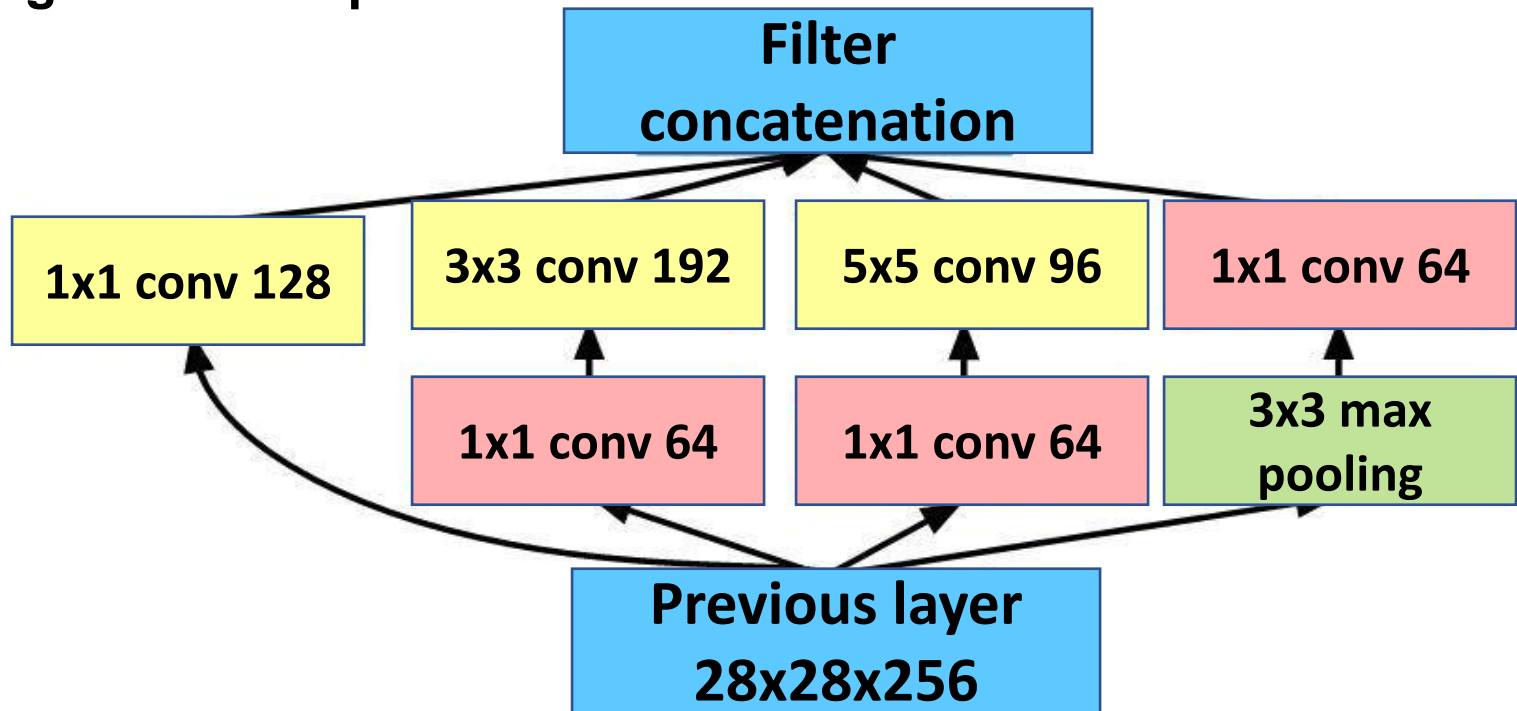


GoogleNet

- Giải pháp: lớp nút cổ chai “bottleneck” sử dụng conv 1x1 để giảm chiều sâu khối dữ liệu.

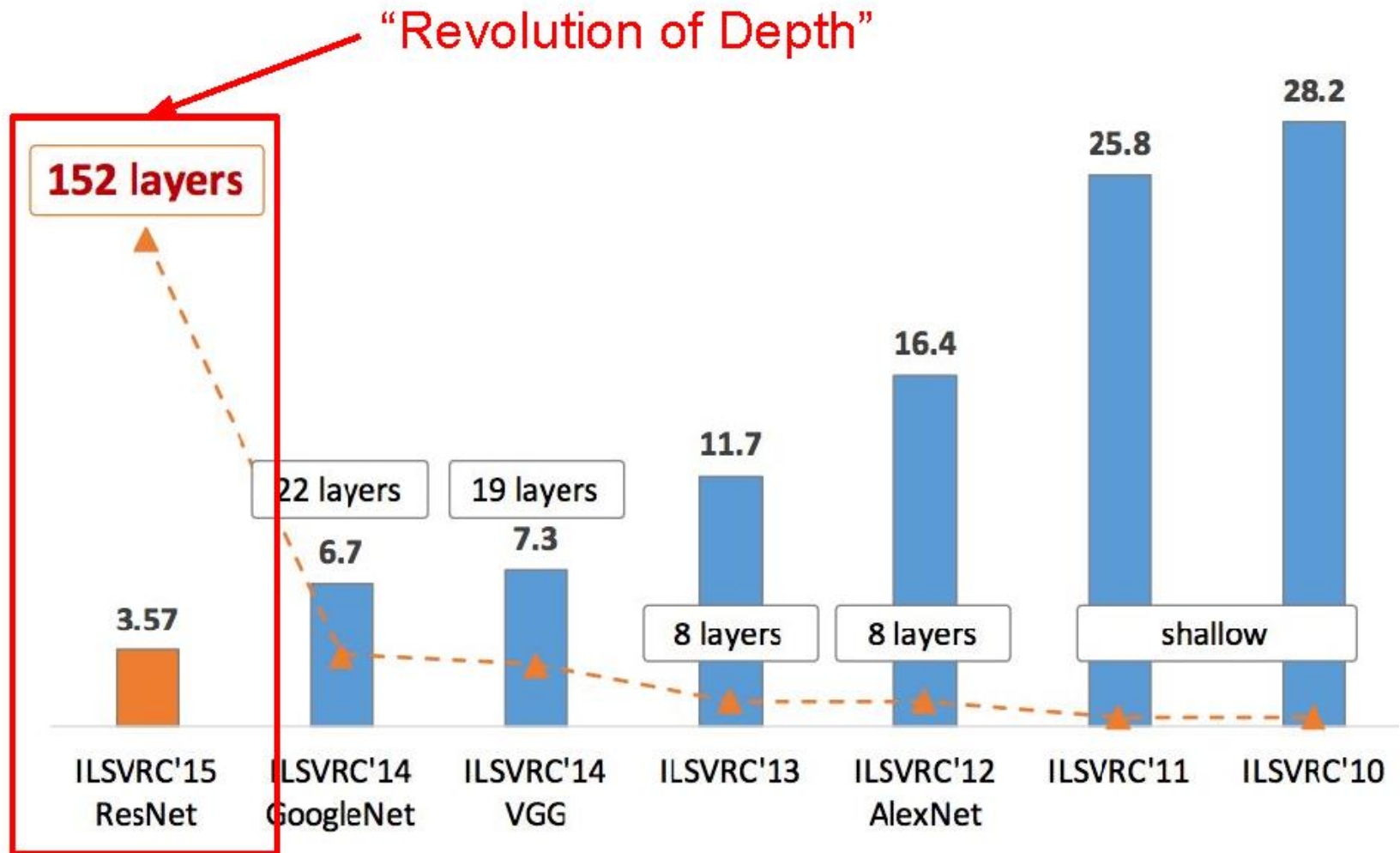


- **Số lượng phép toán tích chập:**
 - 1x1 conv, 64: $28 \times 28 \times 64 \times 1 \times 1 \times 256$
 - 1x1 conv, 64: $28 \times 28 \times 64 \times 1 \times 1 \times 256$
 - 1x1 conv, 128: $28 \times 28 \times 128 \times 1 \times 1 \times 256$
 - 3x3 conv, 192: $28 \times 28 \times 192 \times 3 \times 3 \times 64$
 - 5x5 conv, 96: $28 \times 28 \times 96 \times 5 \times 5 \times 264$
 - 1x1 conv, 64: $28 \times 28 \times 64 \times 1 \times 1 \times 256$
- **Tổng số: 353M ops**



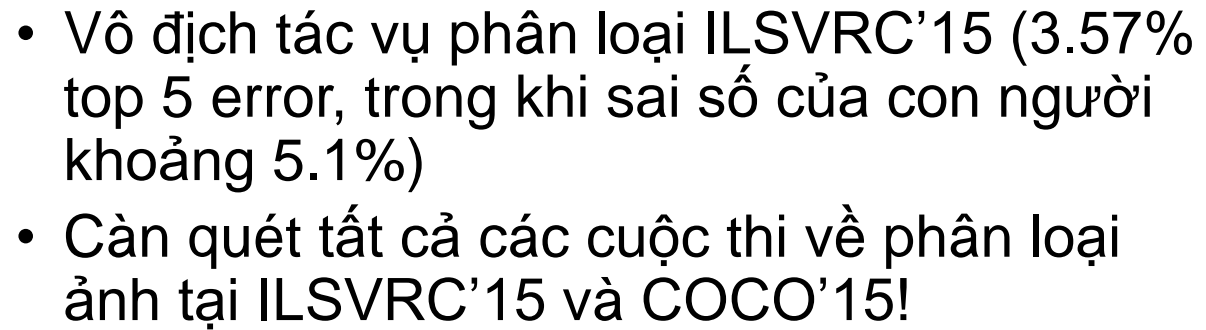
- So với 854M ops với khối inception thường

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



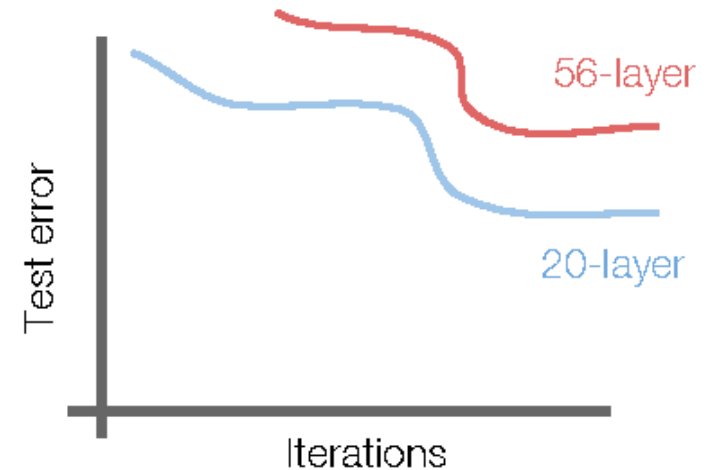
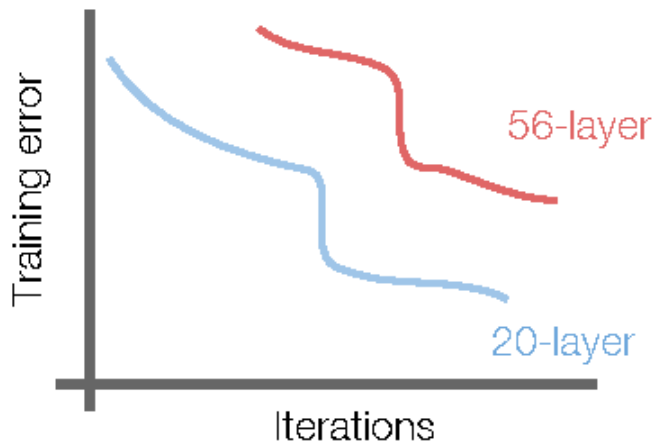
ResNet

- Deep Residual Learning for Image Recognition - Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun; 2015
- Mạng rất sâu, tới 152 lớp
- Mạng càng sâu càng khó huấn luyện.
- Mạng càng sâu càng chịu nhiều ảnh hưởng của vấn đề triệt tiêu và bùng nổ gradient.
- ResNet đề xuất phương pháp học phần dư (residual learning) cho phép huấn luyện hiệu quả các mạng sâu hơn rất nhiều so với các mạng xuất hiện trước đó.



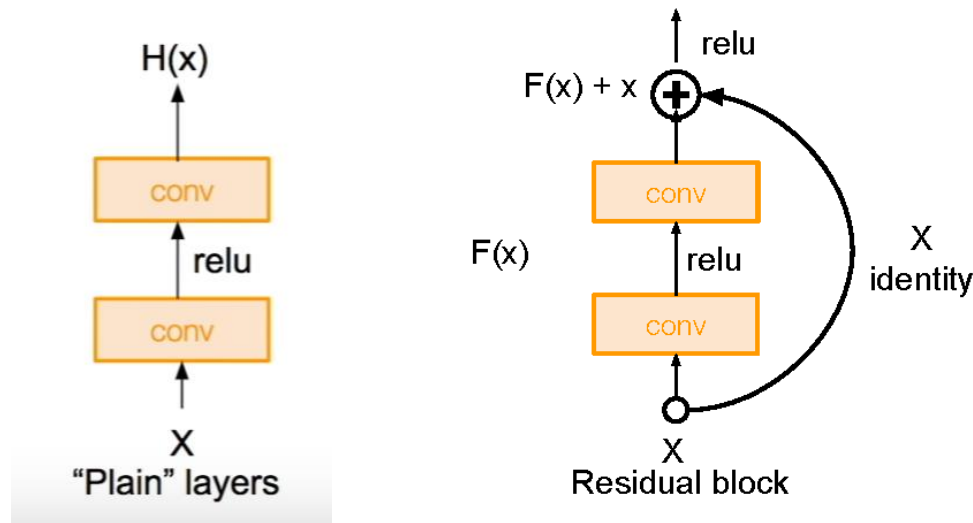
ResNet

- Điều gì xảy ra khi chúng ta tăng độ sâu mạng nơ-ron?
- Mạng 56 lớp làm việc kém hơn cả trên tập huấn luyện lẫn tập test (không phải do overfitting gây ra)
- Hiện tượng suy biến của các mạng sâu



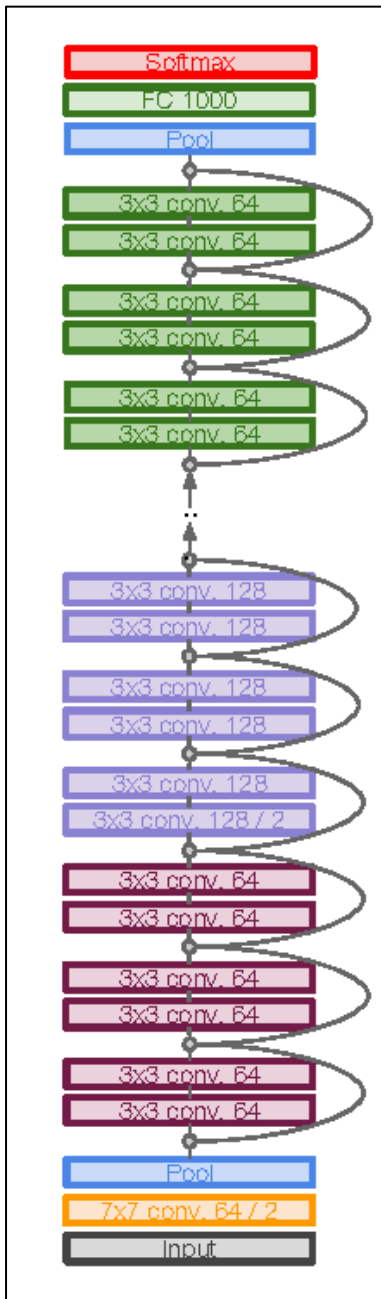
ResNet

- Giả thiết: Vấn đề ở chỗ bài toán tối ưu. Mạng rất sâu sẽ khó hơn để tối ưu.
- Giải pháp: Dùng các lớp mạng để học biểu diễn phần dư (sự sai khác giữa đầu ra và đầu vào) thay vì học trực tiếp đầu ra như trước.
- Học biểu diễn phần dư $F(x) = H(x) - x$ thay vì học trực tiếp $H(x)$



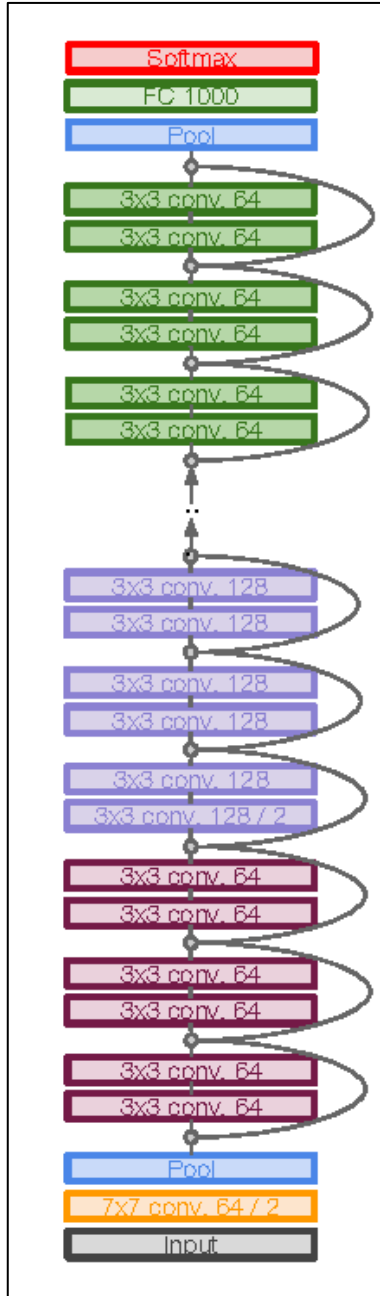
ResNet

- Kiến trúc ResNet đầy đủ:
- Chồng các khối phần dư residual blocks
- Mỗi khối có hai lớp 3x3 conv
- Định kỳ tăng gấp đôi số lượng filter và giảm độ phân giải bằng conv bước nhảy stride 2
- Lớp conv phụ ở đầu mạng
- Không có lớp FC ở cuối (chỉ có lớp FC 1000 để xuất ra kết quả phân loại 1000 lớp)



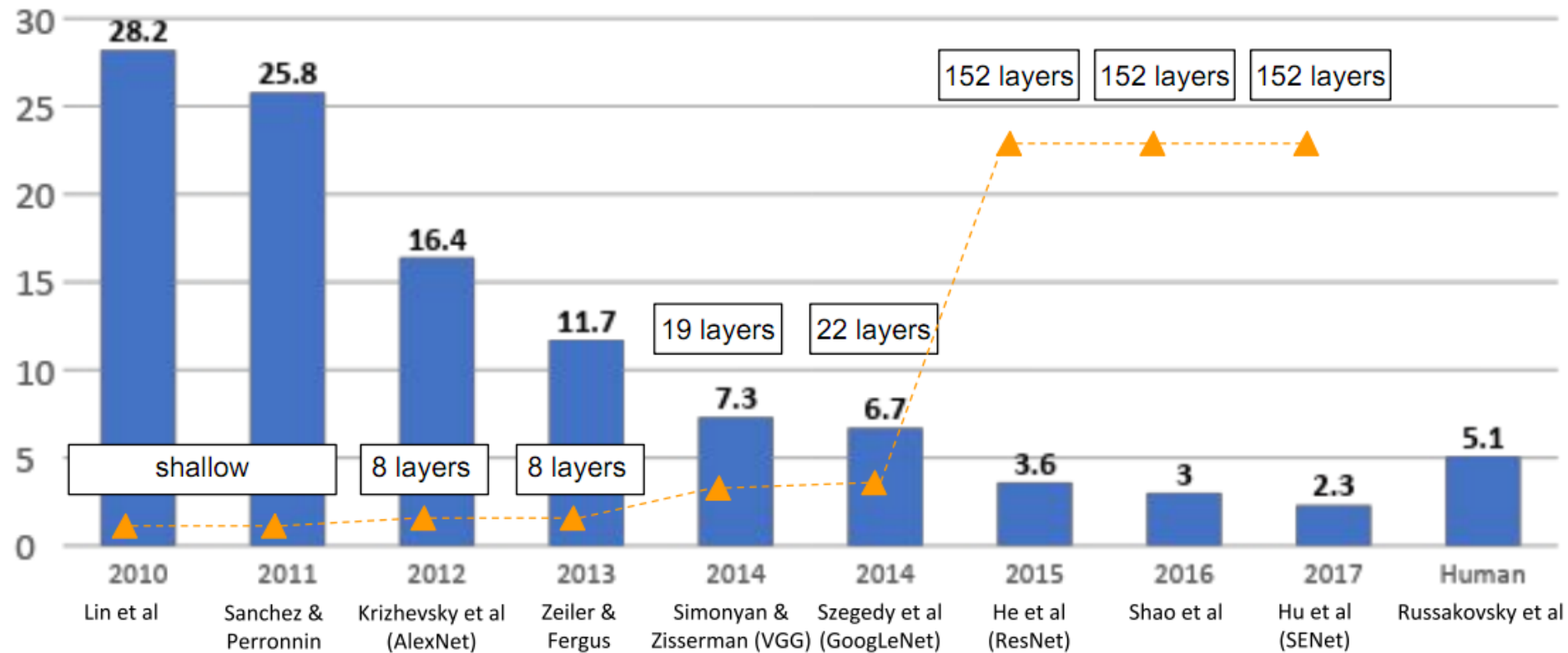
ResNet

- Độ sâu của mạng khi tham gia cuộc thi ImageNet: 34, 50, 101, 152
- Với các mạng sâu (ResNet-50+), tác giả dùng lớp “bottleneck” để tăng hiệu quả (tương tự như GoogLeNet)

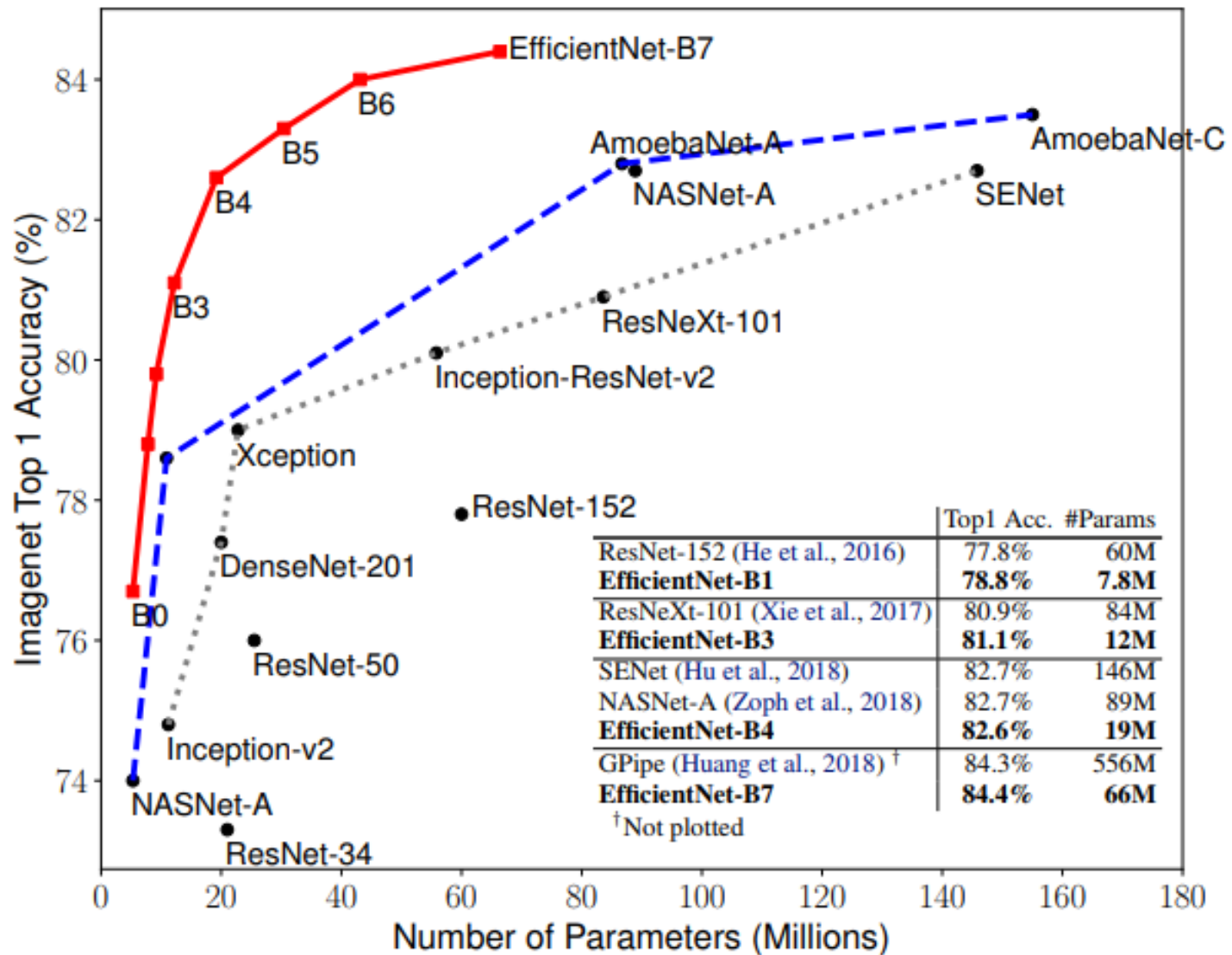


Recent SOTA

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



Recent SOTA



Accuracy comparison

