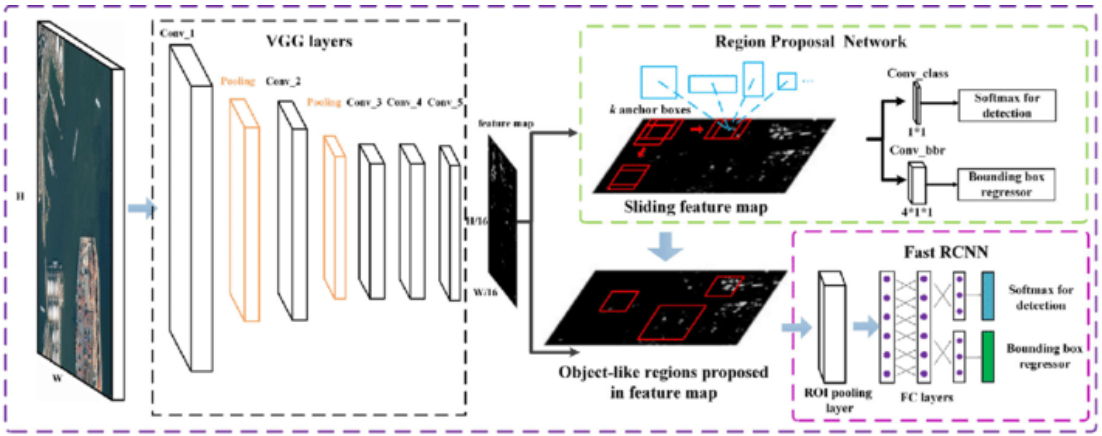
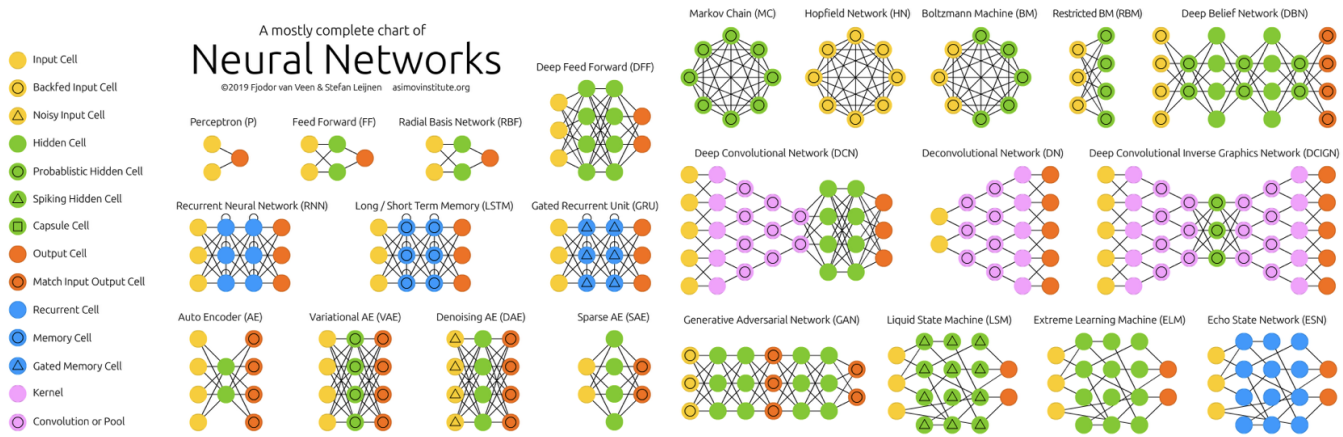


CNN Architectures

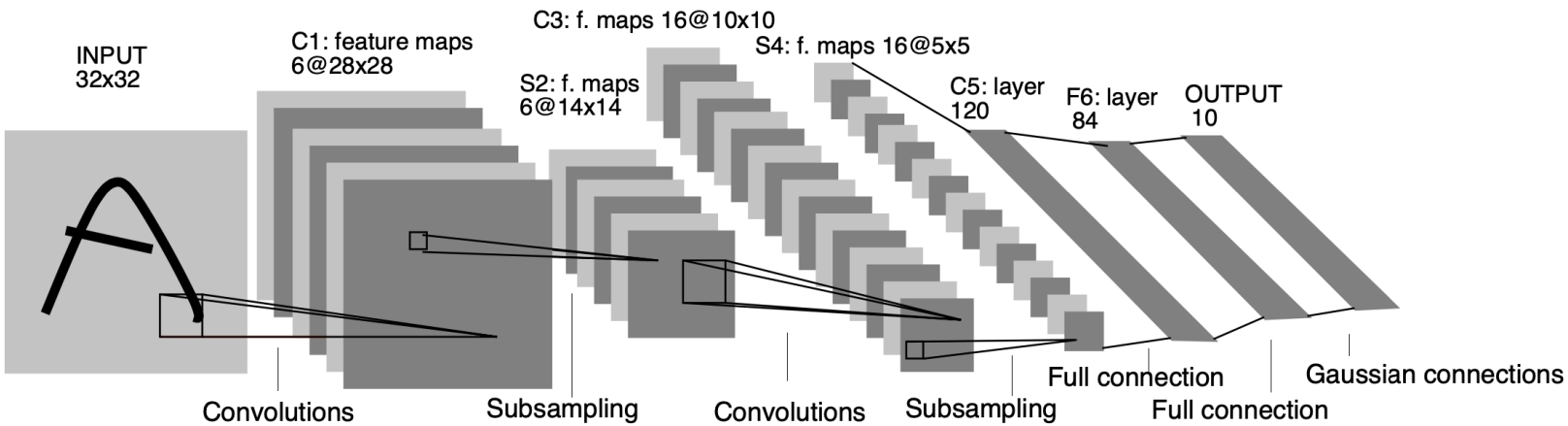
Vincent Barra
LIMOS, UMR 6158 CNRS, Université Clermont Auvergne

INTRODUCTION

- Since 2012, a HUGE collection of CNN models
- How/why the well known models have been built ?
- Finding right architeture : active area of research



LeNet-5 (1998)


INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections

LeNet-5 (1998)

Features

Architecture:

Conv((5× 5),6) - ReLU, Max Pooling - Conv((5× 5),16) - ReLU, Max Pooling - FC(120)-ReLU - FC (84) - ReLU - FC(10) - Softmax

Features

One of the precursors

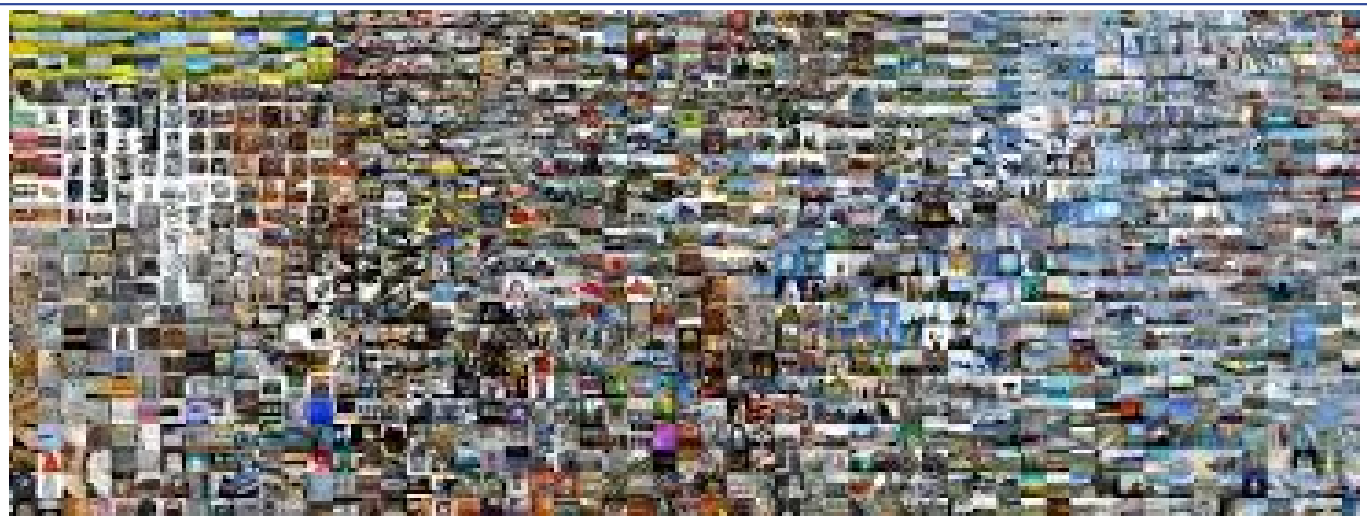
- First CNN to apply backpropagation
- 61 706 trainable parameters
- Number of connections limited
- Used for character recognition.

LeNet-5 (1998)

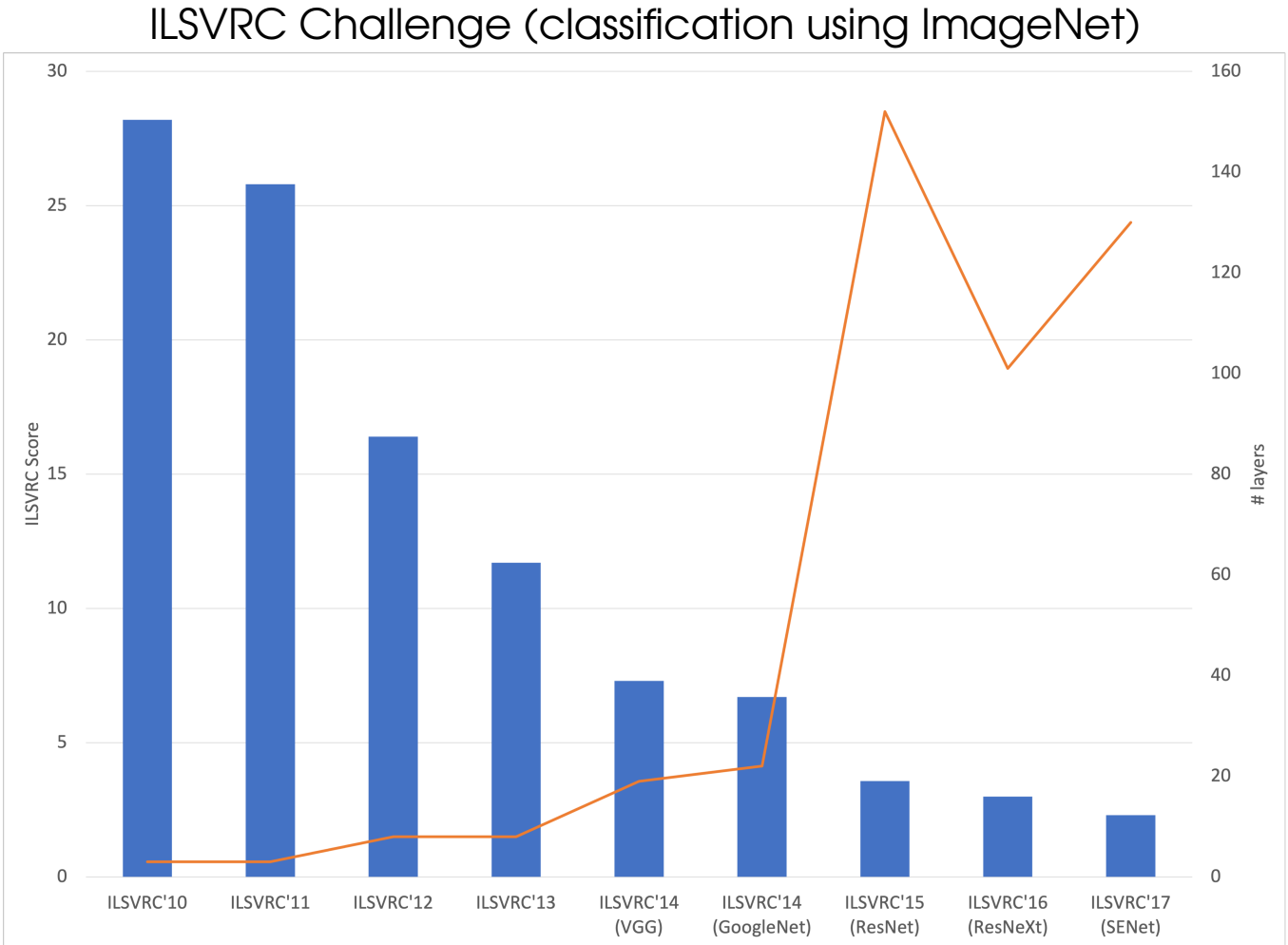
Interlude

Need for data

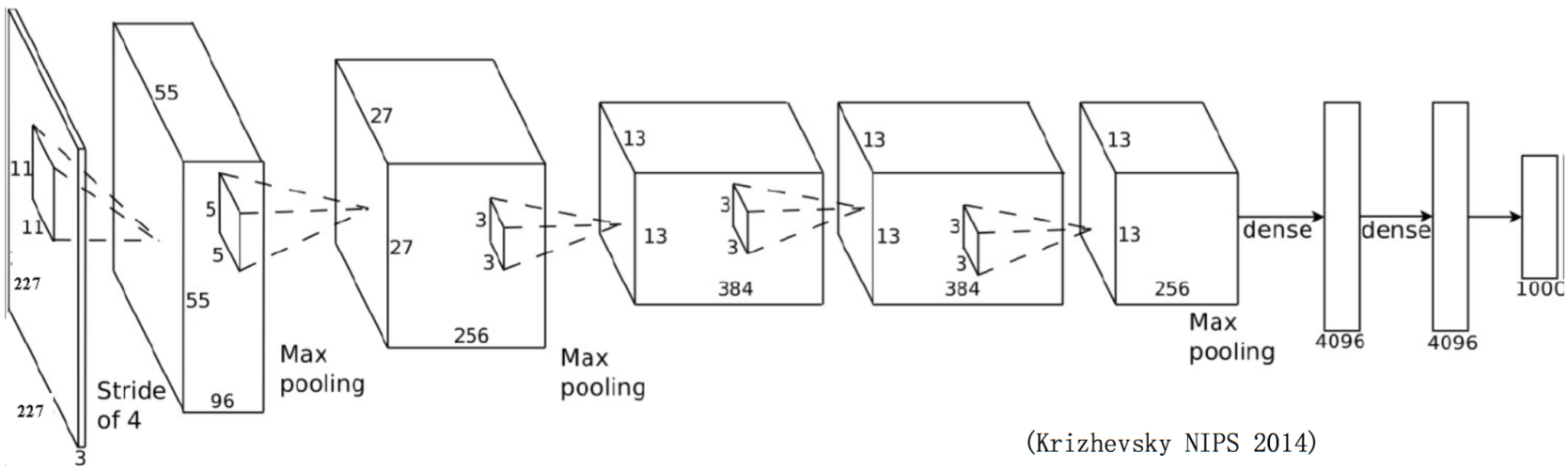
Can't add too much depth because no relevant training set available.... until 2012
$\Rightarrow$ ImageNet (10,000,000 labeled images depicting 10,000+ object categories for training).
(today 14,197,122 images, 21841 synsets indexed).



Interlude


ILSVRC Challenge (classification using ImageNet)
ILSVRC Score
layers
0
5
10
15
20
25
30
0
20
40
60
80
100
120
140
160
ILSVRC'10
ILSVRC'11
ILSVRC'12
ILSVRC'13
ILSVRC'14 (VGG)
ILSVRC'14 (GoogleNet)
ILSVRC'15 (ResNet)
ILSVRC'16 (ResNeXt)
ILSVRC'17 (SENet)

AlexNet (2012)



(Krizhevsky NIPS 2014)

AlexNet (2012)

Features

Architecture: (simplified)

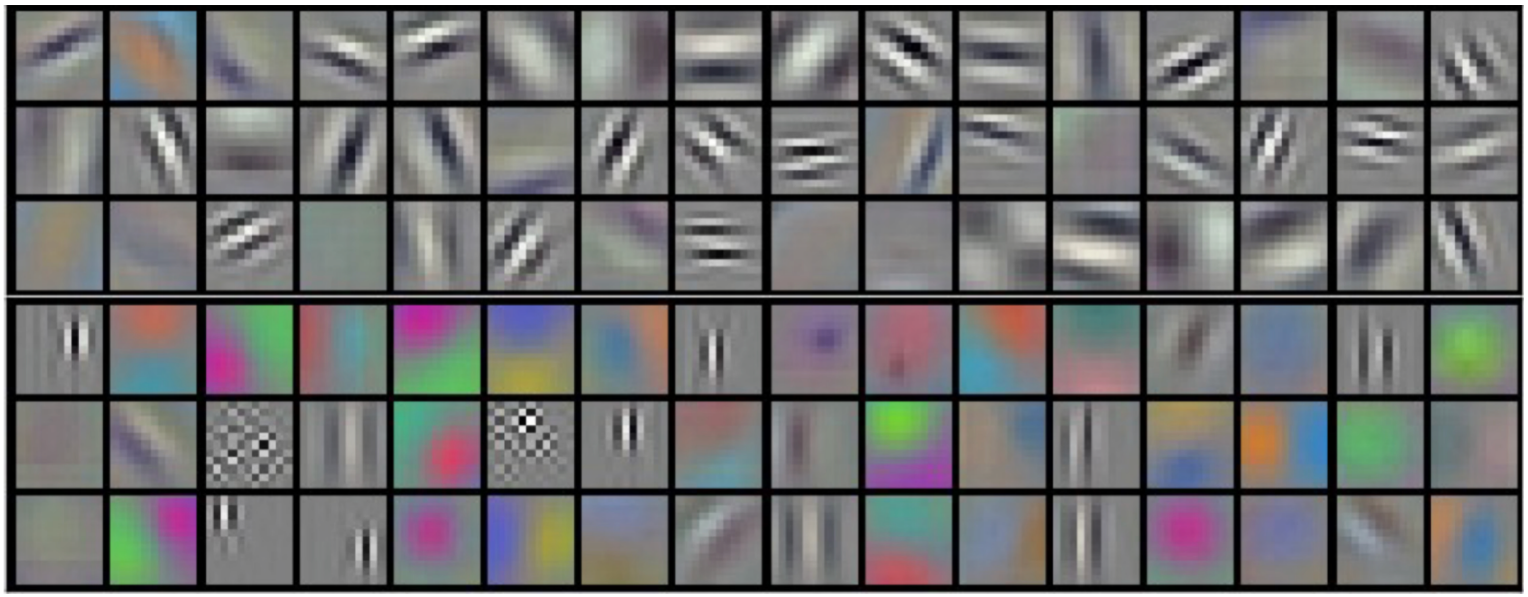
Conv(11×11,96) - ReLU - Max Pooling - Conv(5×5,192) - ReLU - Max Pooling - Conv(3×3,384) - ReLU - Conv(3×3,256) - ReLU - Conv(3×3,256) - ReLU - MaxPool - Softmax

Features

- ReLU, local response normalization, data augmentation, dropout
- More than 62.10^6 trainable parameters
- stochastic gradient descent with momentum

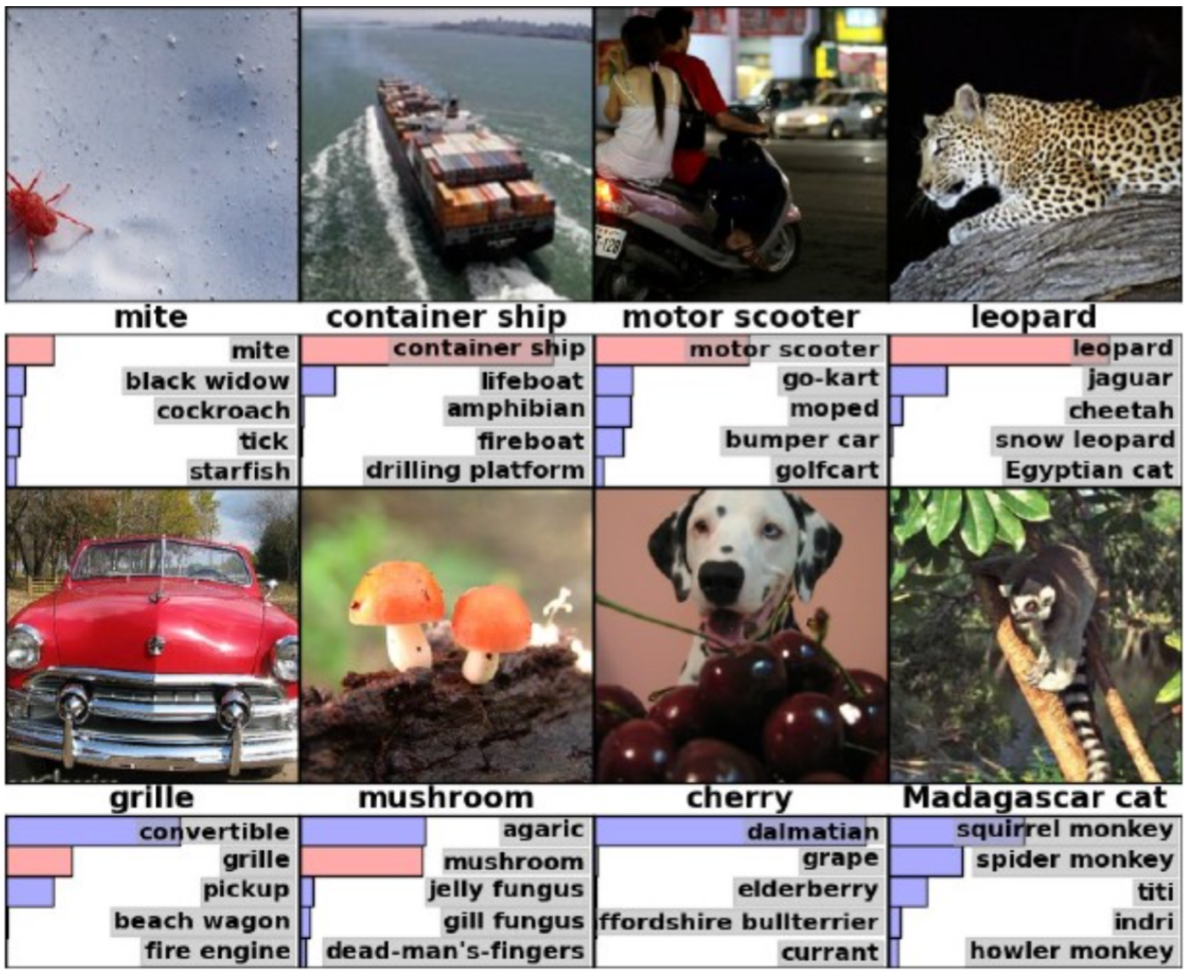
AlexNet (2012)

Layer 1 - Learned filters

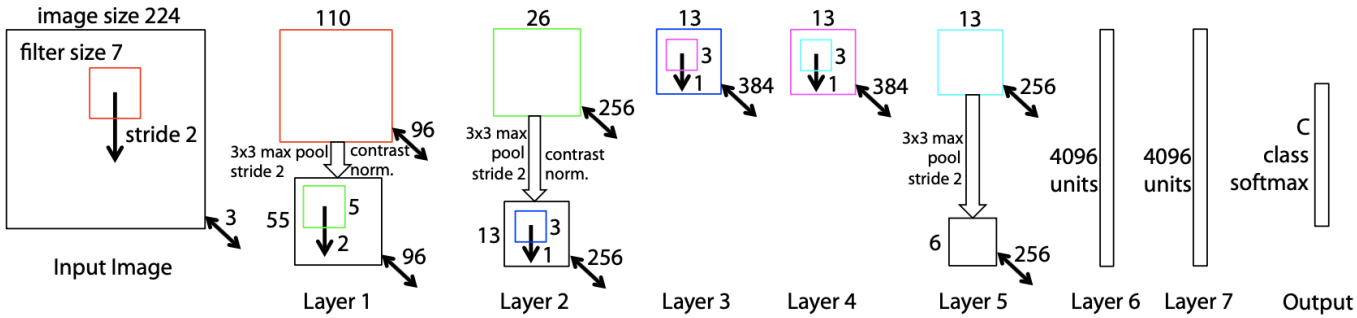


ALEXNET (2012)

AlexNet Results.



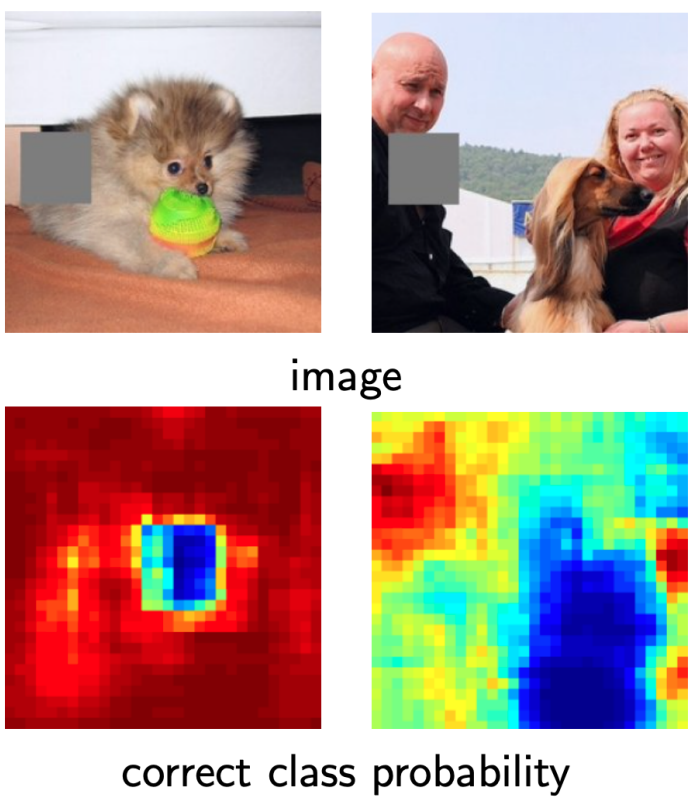
ZEILER AND FERGUS (2014)



Features

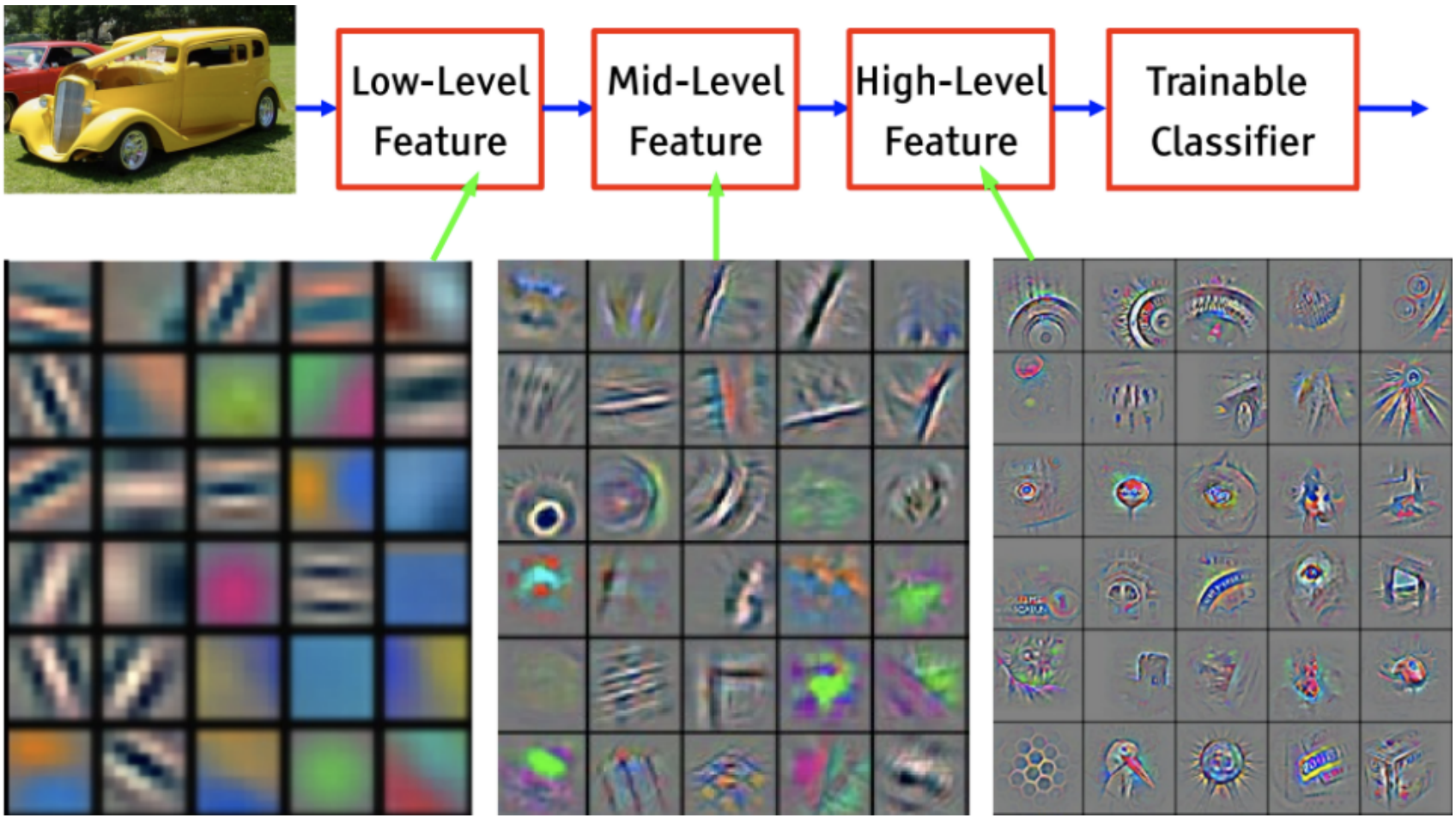
- Refinement of AlexNet
- More than 107.10^6 trainable parameters
- Layer widths adjusted by cross-validation $\Rightarrow$ Depth matters

ZEILER AND FERGUS (2014)



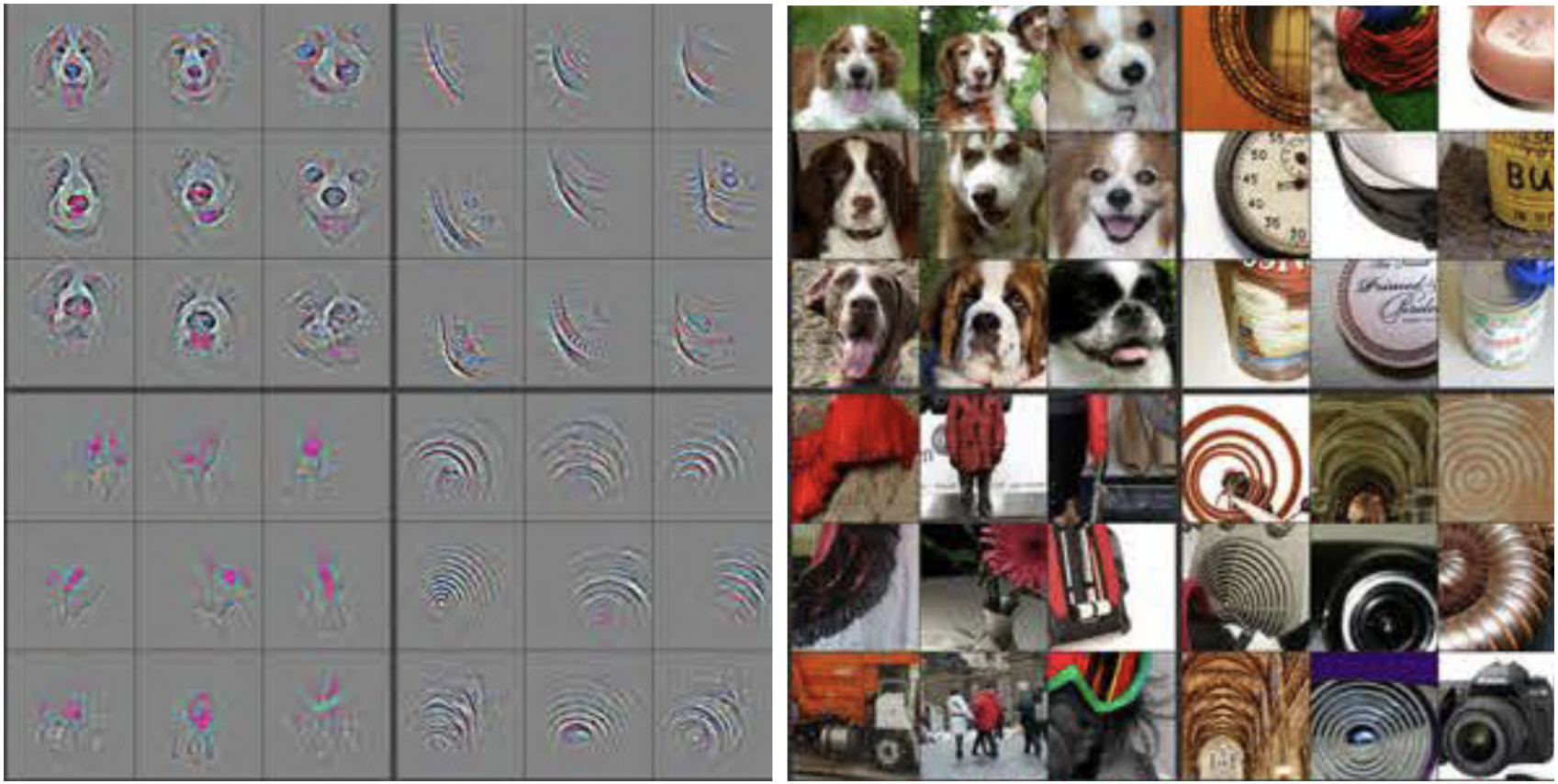
- image occluded by gray square at different positions (x, y)
- correct class probability : function of (x, y)

ZEILER AND FERGUS (2014)



Hierarchical representation

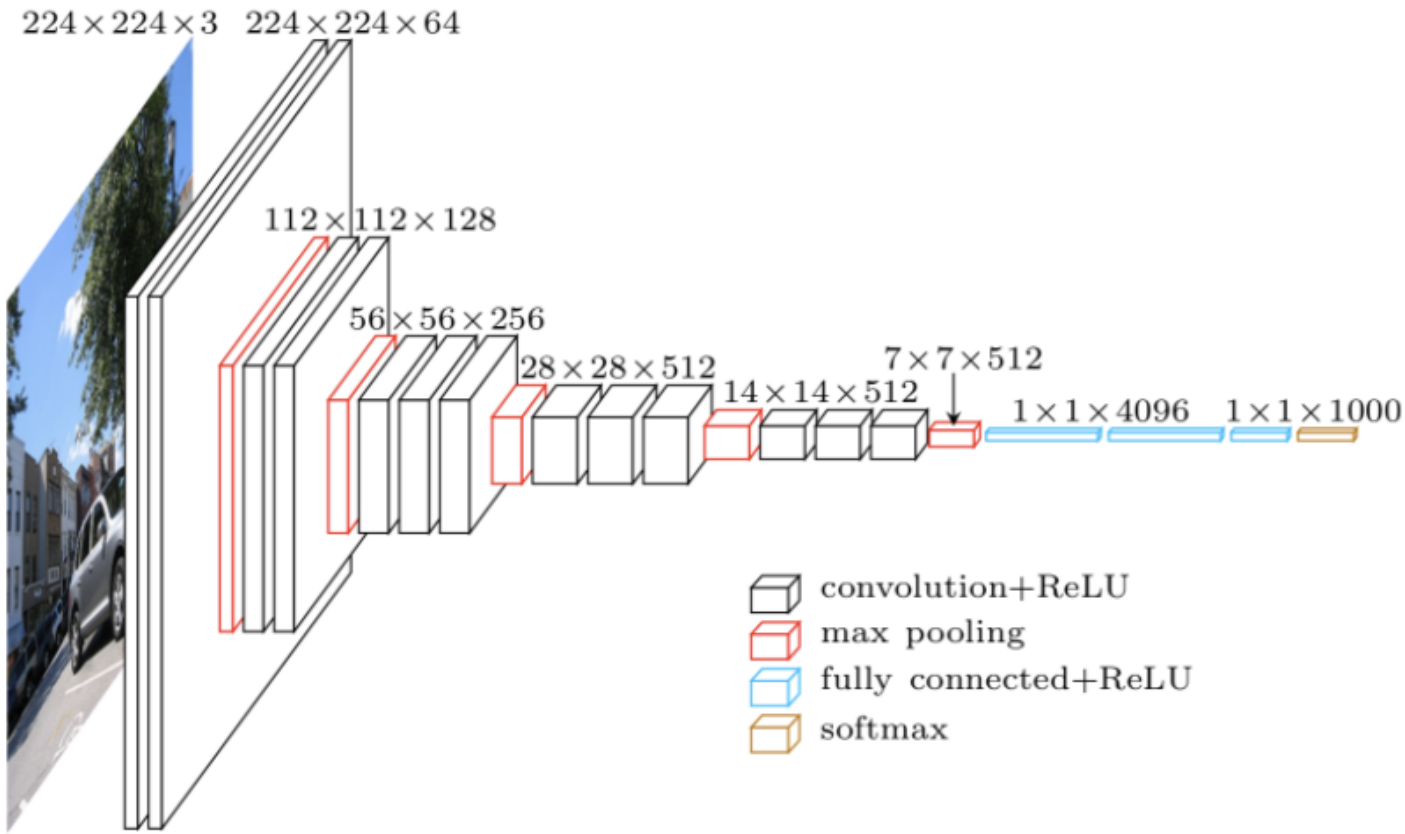
Zeiler and Fergus (2014)



Reconstructed patterns from top 9 activations of selected features of layer 4 and corresponding image patches

VGG (2015)

Very Deep CNN for Large-Scale Image Recognition. Family of networks.



VGG (2015)

Very Deep CNN for Large-Scale Image Recognition

A	A-LRN	B	C	D	E
11 layers	11 layers	13 layers	16 layers	16 layers	19 layers
133 M param	133 M param	133 M param	134 M param	138 M param	144M param
Input: 224×224 RGB image					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
max pooling					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
max pooling					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
max pooling					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
max pooling					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
max pooling					
FC layer 4096 neurons					
FC layer 4096 neurons					
FC layer 1000 neurons					
Softmax					

VGG (2015)

Features

- 3×3 filters: smallest possible filters to capture 4-connectivity
- 2 3×3 filters = receptive field of a 5×5 filter
- 3 3×3 filters = receptive field of a 7×7 filter

⇒ A large filter can be replaced by a deeper stack of successive smaller filters
⇒ More nonlinearities (ReLU), fewer parameters (3×3×K).2 = 18K / (5×5×K).1 = 25K)
⇒ Extension to 1×1 filters+nonlinearity : increase nonlinearities without affecting receptive field sizes.

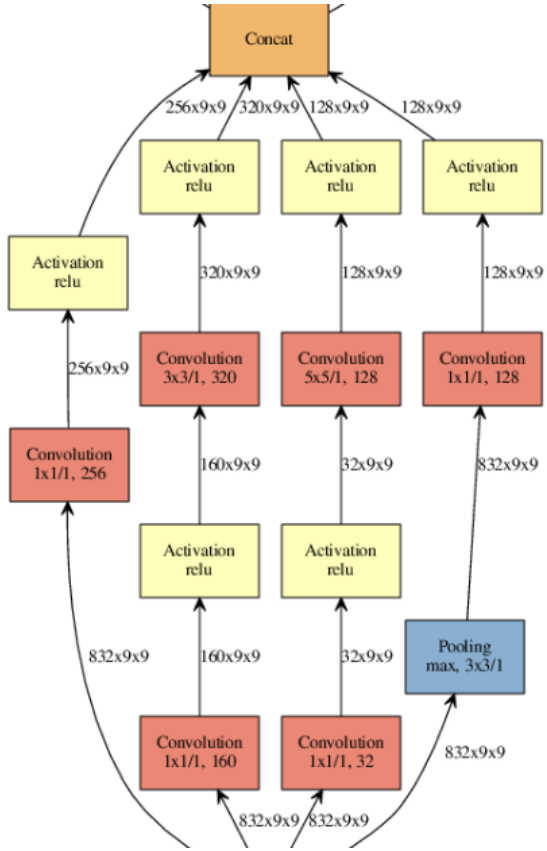
Result: 7.3% error rate in ImageNet (18.2% for AlexNet)

Although deeper, number of weights is not exploding.

GoogLeNet (2015)

Features

- Incarnation of the Inception architecture
- Multiple feedforward pathways



Inception family

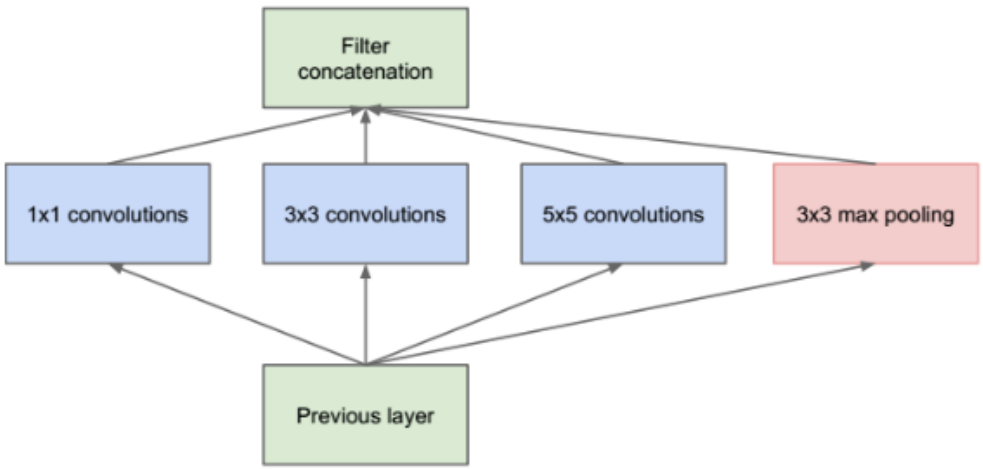
Idea

- Salient parts may have great variation in sizes
- The receptive field shoud vary in size
- Stacking CNN is expensive & may overfit.

Inception family

Idea

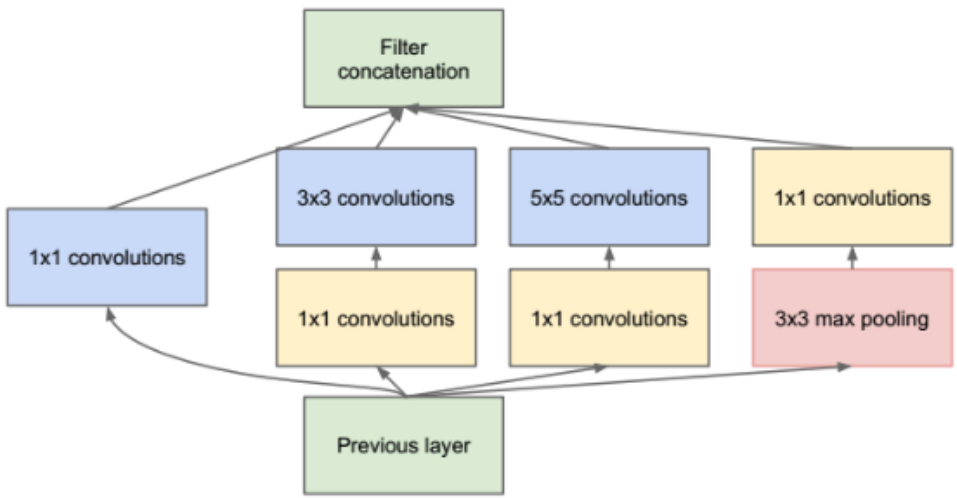
- Multiple kernel filters of different sizes⇒ Expensive !



Inception family

Idea

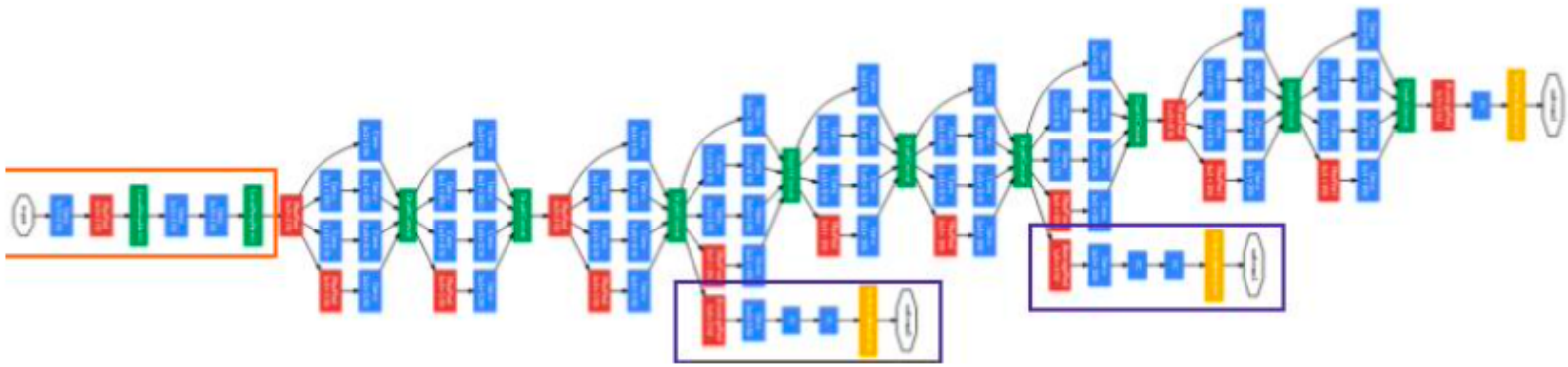
- Multiple kernel filters of different sizes⇒ Expensive !
- Solution: add 1×1 convolutions



INCEPTION FAMILY

Idea

- Multiple kernel filters of different sizes⇒ Expensive !
- Solution: add 1×1 convolutions
- 9 inception modules, 27 layers



Inception family

Idea

- Multiple kernel filters of different sizes⇒ Expensive !
- Solution: add 1×1 convolutions
- 9 inception modules, 27 layers

6.67% Imagenet error, compared to 18.2% of Alexnet, 25 times less parameters and faster than AlexNet

Inception family

Idea

- Multiple kernel filters of different sizes⇒ Expensive !
- Solution: add 1×1 convolutions
- 9 inception modules, 27 layers

The network was too deep :-(⇒ vanishing gradient

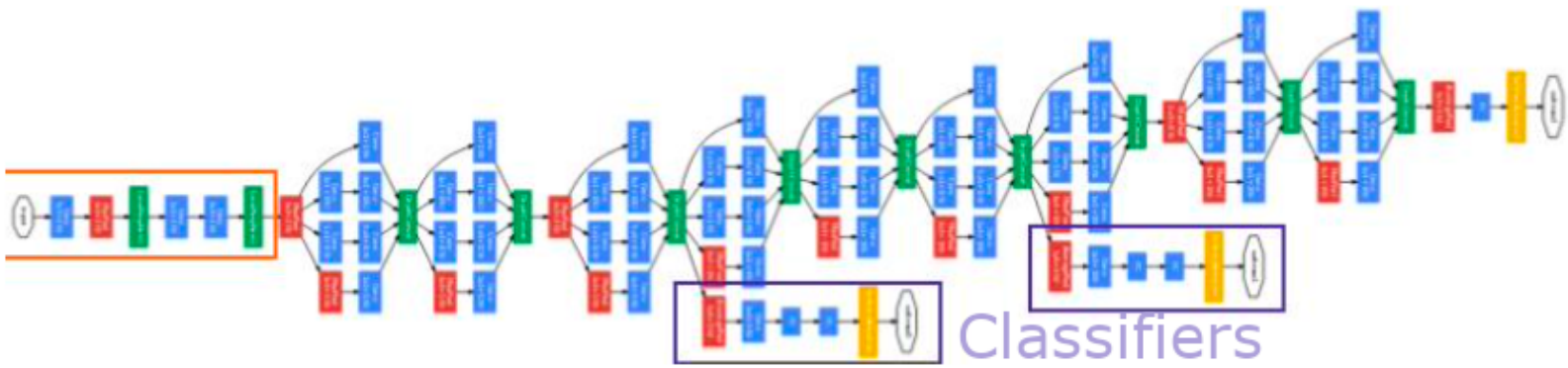
Additional ressource

See Slides "Vanishing gradient".

Inception family

Idea

- Multiple kernel filters of different sizes⇒ Expensive !
- Solution: add 1× 1 convolutions
- 9 inception modules, 27 layers

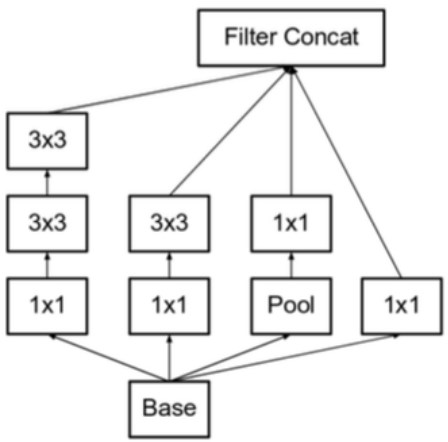


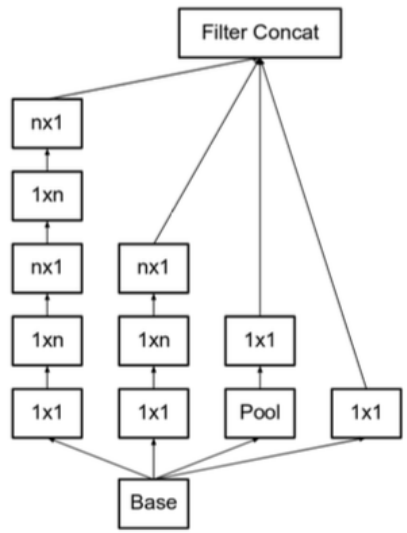
Solution: Intermediate classifiers

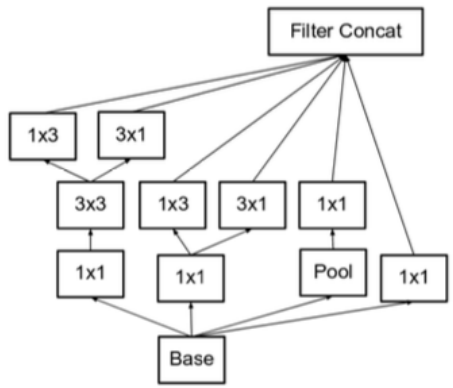
Inception family

v2,v3,v4,v5

- Factorization of 5×5 in 2 3×3
- Factorization of n×n in 2 n×1 and 1×n
- Optimization function (RMSProp)
- Batch normalization...







LIMOS Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes

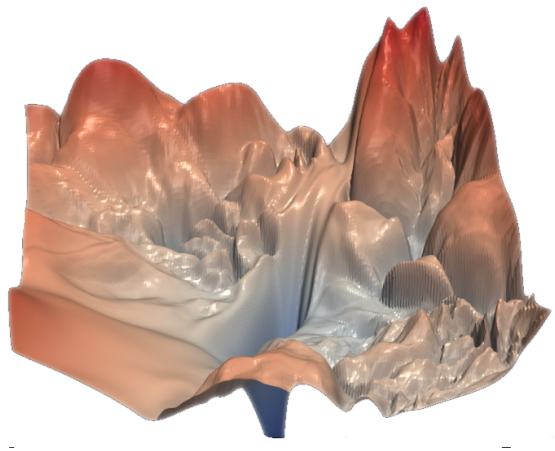
cnrs

UCA Université Clermont Auvergne

INTRODUCTION

Some facts

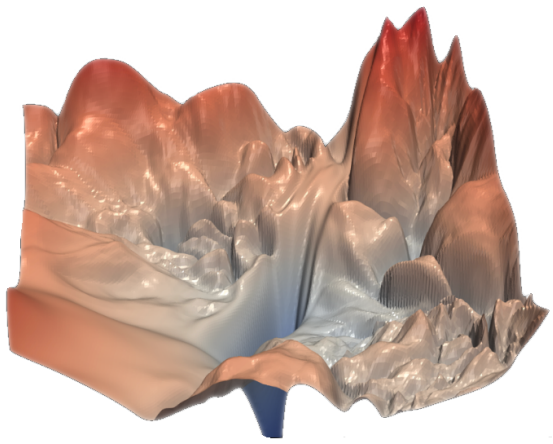
- Very deep networks stop learning after some time
- max accuracy→ saturation→ unlearning
- causes : not only overfitting and vanishing gradients but harder optimization



INTRODUCTION

Some facts

- Very deep networks stop learning after some time
- max accuracy→ saturation→ unlearning
- causes : not only overfitting and vanishing gradients but harder optimization



Some facts

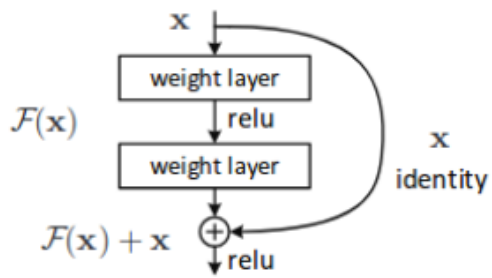
- Although identity is representable, learning it proves difficult for optimization methods
- Intution: tweak the network so it doesn't have to learn identity connections

Residual block

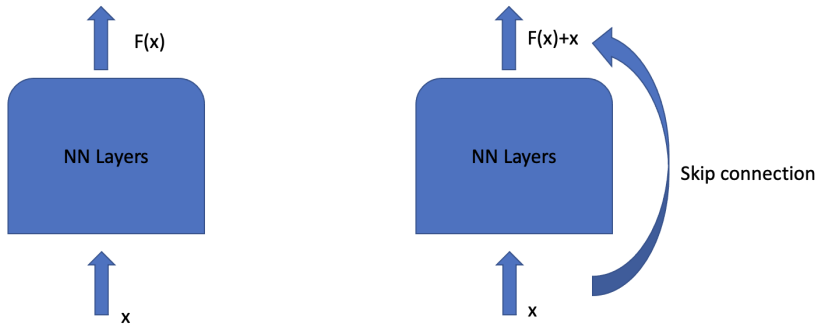
Residual idea

Learn residual functions rather than direct mappings.
If x is the input to the nonlinear layer, and $\mathcal{F}$ the stacked nonlinearities

$$\mathcal{F}(x) = \mathcal{R}(x) - x \Rightarrow \mathcal{R}(x) = \mathcal{F}(x) + x$$



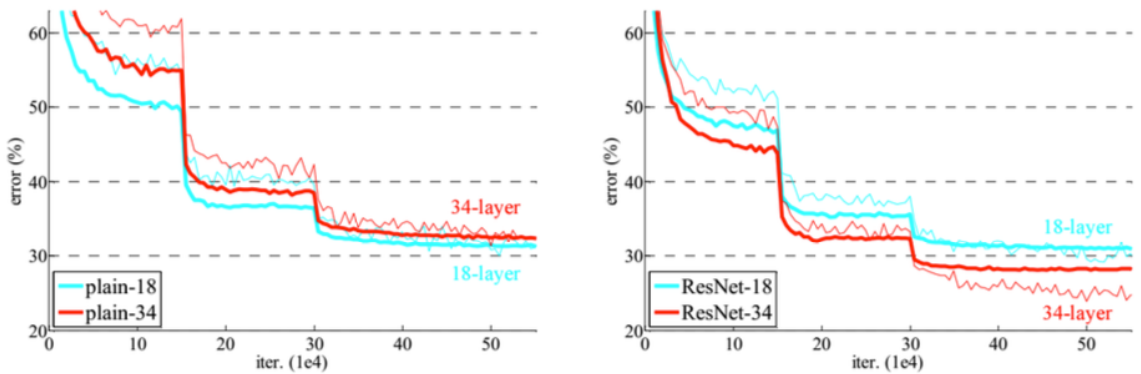
RESIDUAL BLOCK



Hard to get F(x)=x and
Make y=x identity mapping

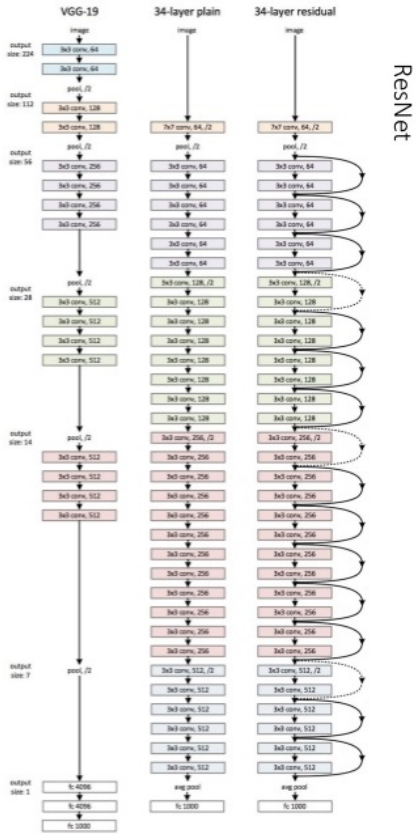
Easy to get F(x)=0 and
Make y=x identity mapping

With residual connections, deeper nets improve their scores (here on ImageNet)



Residual block

- Very low errors on imageNet (all challenges)
- Up to 1202 layers
- Skip connections $\Rightarrow$ faster training
- Batch norm compulsory (vanishing gradient)
- Comparison with VGG-D
  # parameters: 25M / 138M
- Comparison with VGG-D
  computational complexity: 3.8B / 15.3B Flops.



Residual networks

Why do they work ?

- ResNets $\approx$ implicitly ensembling shallower networks: with r residual modules there are 2^r possible subsets of modules
- Able to learn unrolled iterative refinements

Residual networks

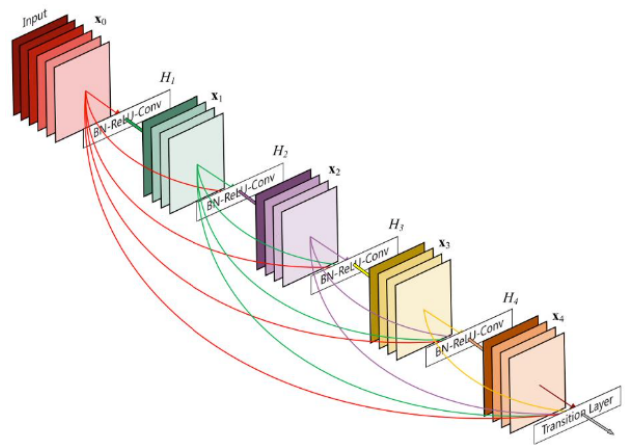
HighwayNet

$\approx$ ResNets + gate with learnable parameters on the importance of each skip connection

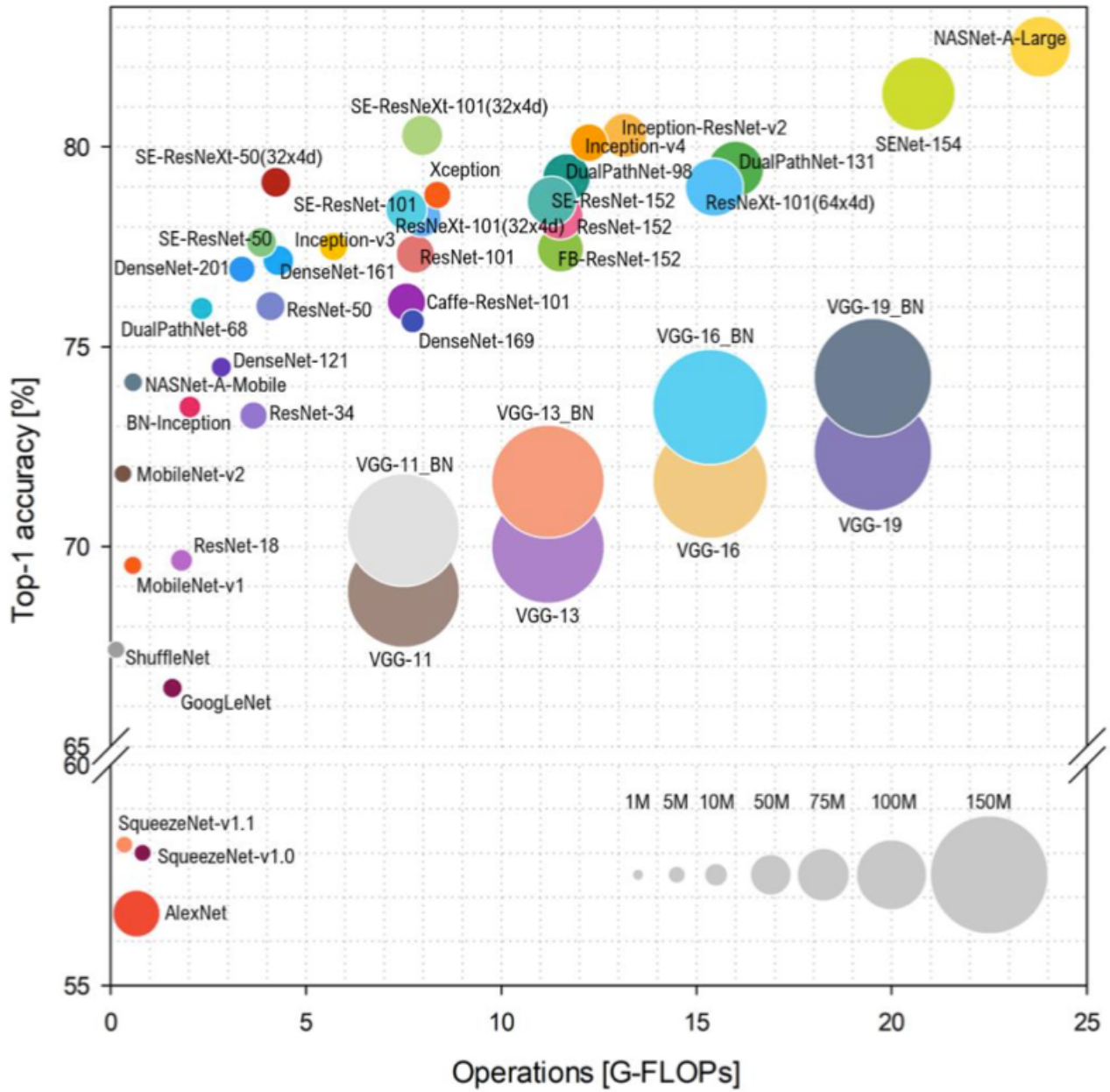
$$y = T(x, W_t).\mathcal{R}(x, W_h) + (1 - T(x, W_t)).x$$

DenseNet

- Add skip connections to multiple forward layers
- Each layer is conneted to every subsequent layer
- Interest: combinaison of detected patterns



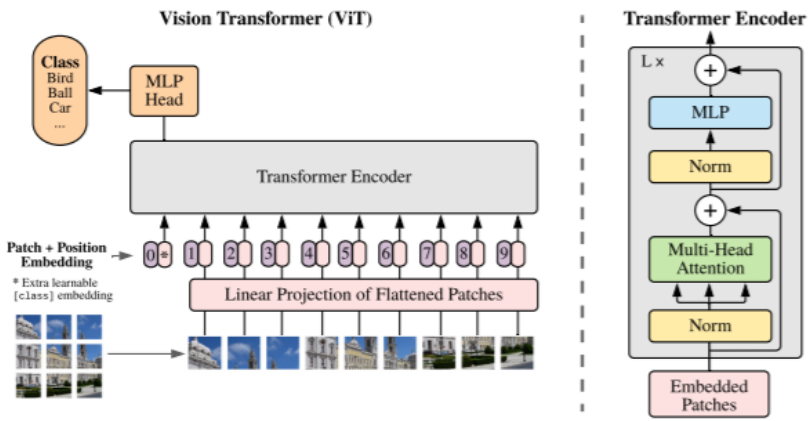
STATE-OF-THE-ART



Current trends (change rapidly...)

Some issues

- Reduce filter sizes (except possibly at the lowest layer), factorize filters
- 1×1 convolutions to reduce and expand the number of feature maps
- Use skip connections
- Attention mechanisms
- Visual transformers
- NAS (Network Architecture Search)



State-of-the-art

- All these models are already trained on huge databases (ImageNet, CIFAR...).
- For a "close" problem, no need to train the CNN from scratch...
- Subject of a further lecture (teasing :-))

See....

Lecture "Transfer Learning".