Convolutional neural networks

- History and background
    - Simple cells and complex cells
- Basic structures of CNNs
    - Convolution and pooling
- Training CNNs
- Recent designs of CNNs

CNN : Convolutional Neural Networks

- Neocognitron [Fukushima80]
- LeNet [LeCun+89]
  - Backpropagation Applied to Handwritten Zip Code Recognition, 1989
- Based on findings in neuroscience
  - Simple cell/complex cell [Hubel-Wiesel]
  - Local receptive field)

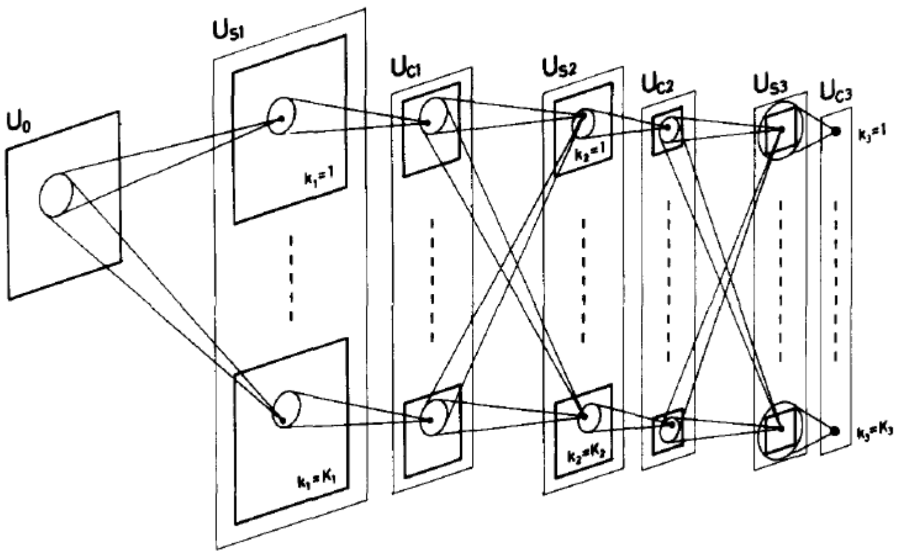


Fig 4 Schematic diagram illustrating the interconnections between layers in the neocognitron

[Fukushima+83]

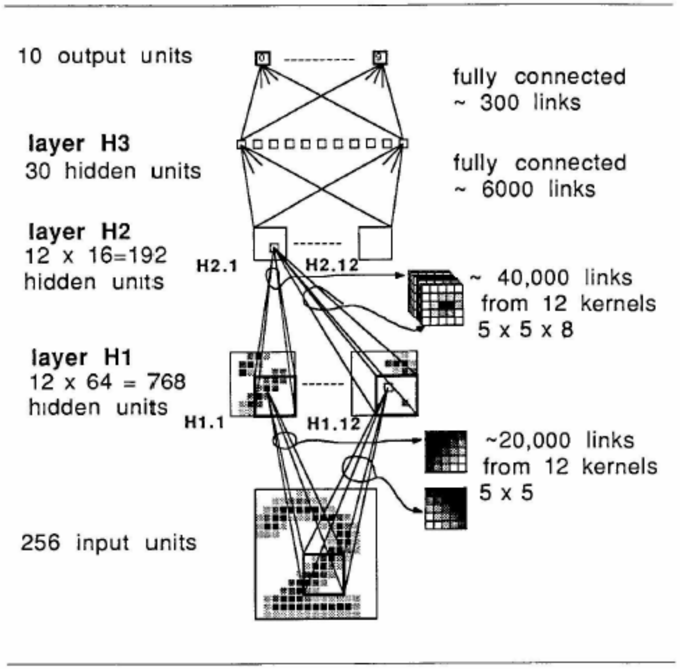
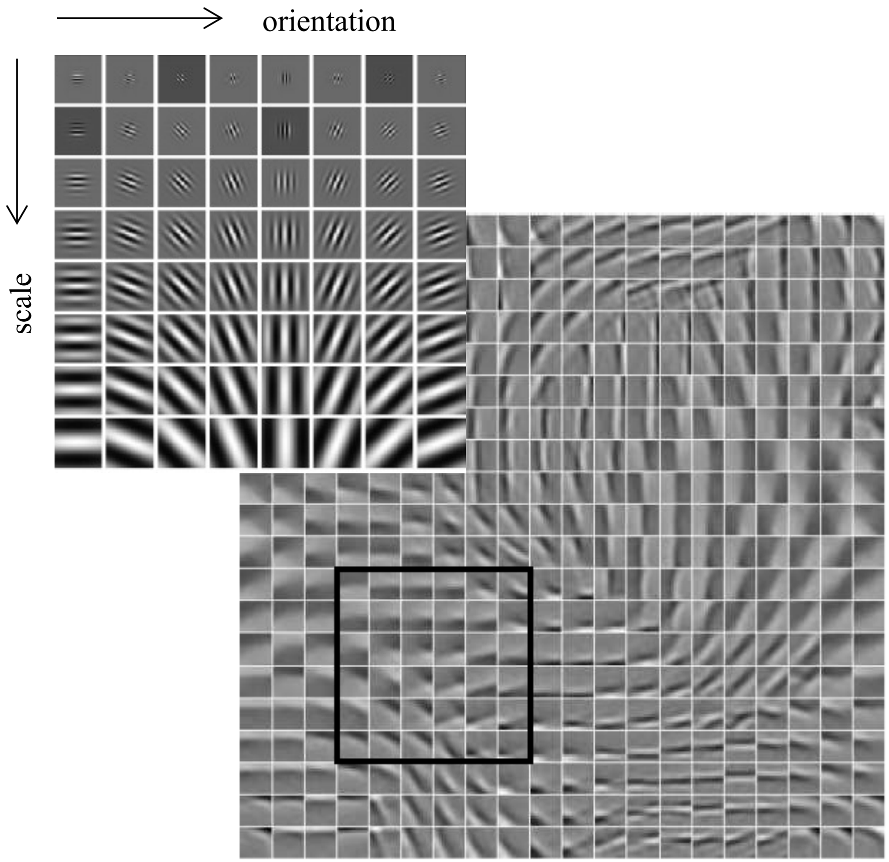


Figure 3 Log mean squared error (MSE) (top) and raw error rate (bottom) versus number of training passes

[LeCun+89]

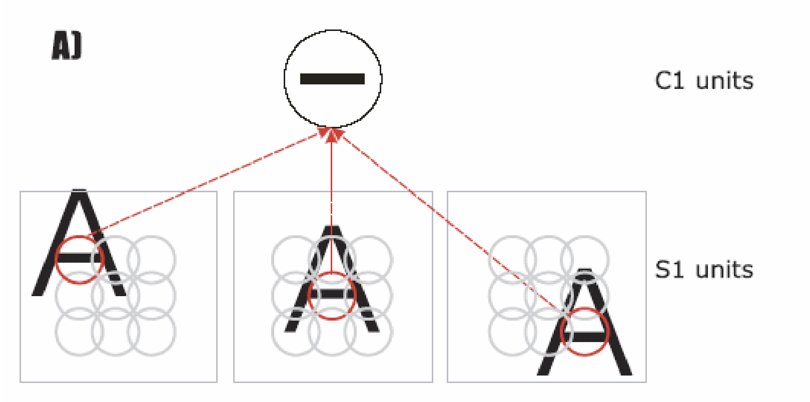
V1 area of visual cortex and simple cells/complex cells

- Gabor wavelets
  - Tuned to position/orientation/scale
  - Topographic map



Kavukcuoglu, Ranzato, Fergus, LeCun, Learning Invariant Features through Topographic Filter Maps, CVPR09

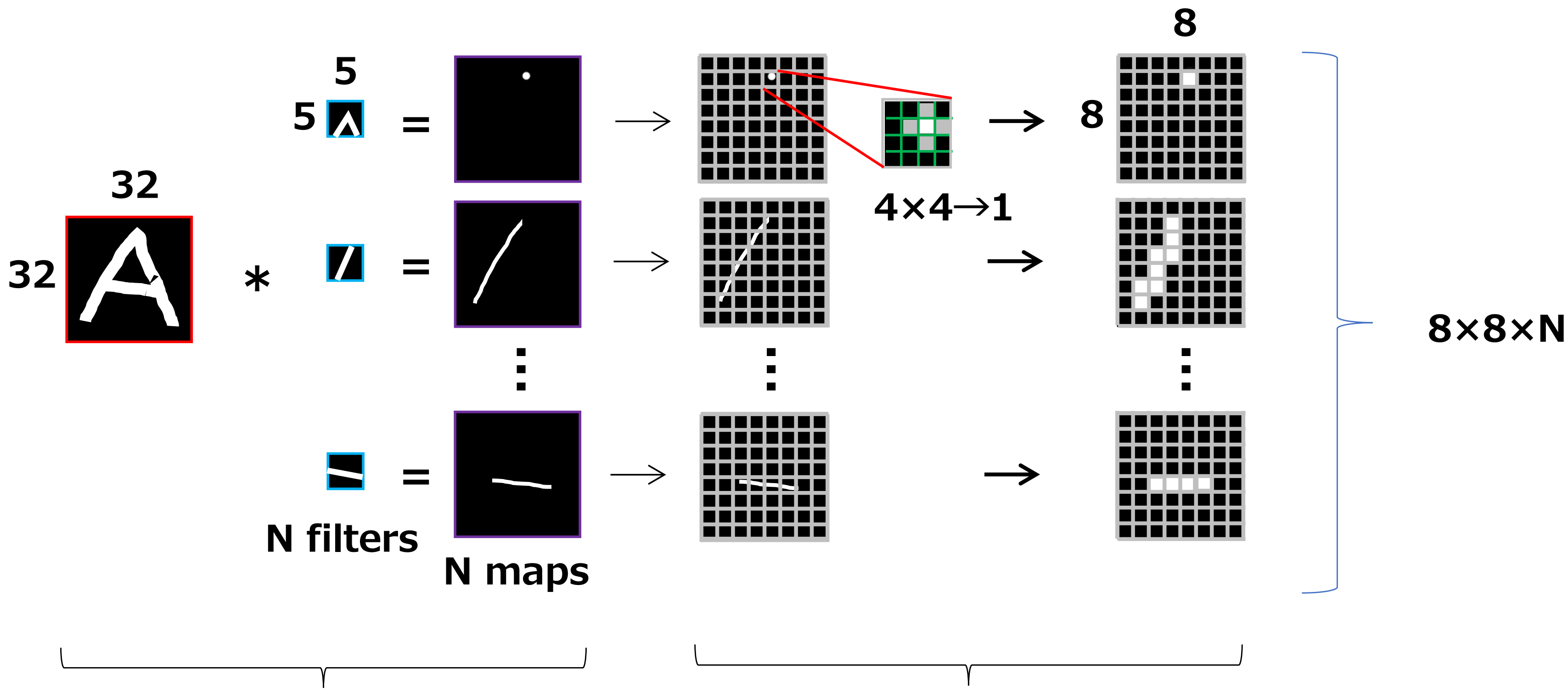
- Simple cells/complex cells [Huber-Wiesel59]



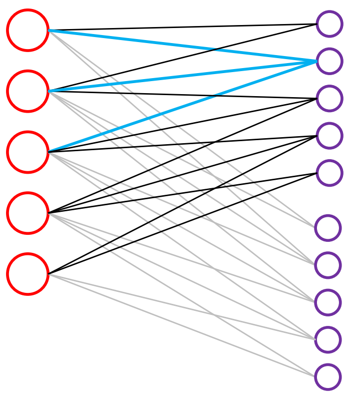
Serre et al, Object Recognition with Features Inspired by Visual Cortex, CVPR05

- Slow feature analysis [Berkes-Wiskott05]
- Gabor quadrature pair [Jones - Palmer87]

Two operations: convolution & pooling

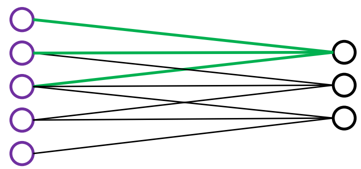


Convolution



- Shared weights
- Sparse connection

Pooling



- Fixed weights
- Sparse connection

Convolution

$$u_{ij} = \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{i+p,j+q} h_{pq}$$

Input **Filter** **Output (map)**

77	80	82	78	70	82	82	140
83	78	80	83	82	77	94	151
87	82	81	80	74	75	112	152
87	87	85	77	66	99	151	167
84	79	77	78	76	107	162	160
86	72	70	72	81	151	166	151
78	72	73	73	107	166	170	148
76	76	77	84	147	180	168	142

$x_{i+p,j+q}$

$\otimes$

0.01	0.08	0.01
0.08	0.62	0.08
0.01	0.08	0.01

h_{pq}

$=$

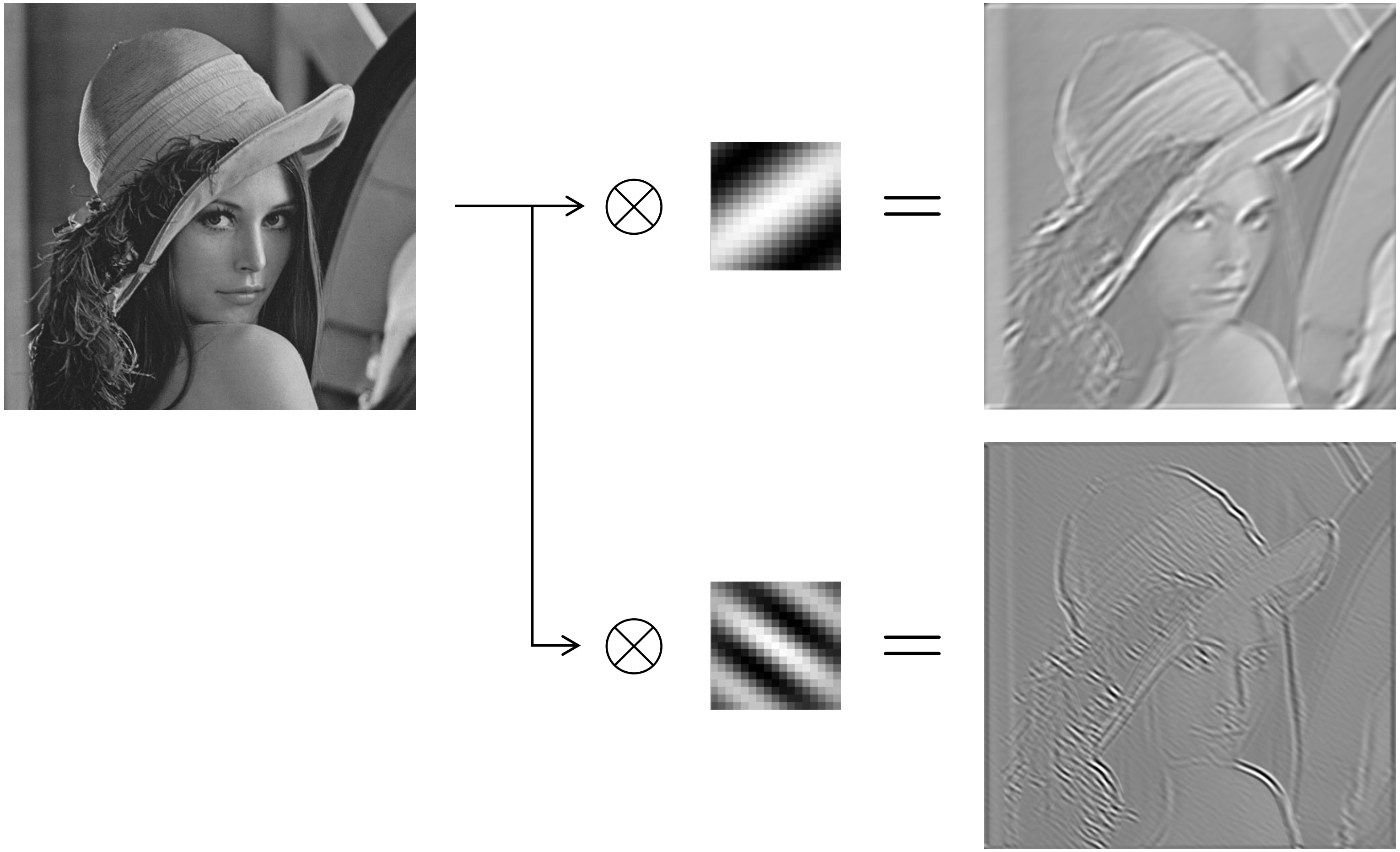
79	80	81	79	79	98
82	81	79	75	81	114
85	83	77	72	99	144
79	77	77	79	112	155
73	71	73	89	142	162
73	73	77	110	160	166

u_{ij}

Convlution

$$u_{ij} = \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{i+p,j+q} h_{pq}$$

Input **Filter** **Output (map)**



$x_{i+p,j+q}$ h_{pq} u_{ij}

Pooling

- Choose and transmit a single value from a small region in input
  - The small regions are sampled with margins, resulting in smaller resolution (or size) of the output

Input

62	71	72	69	65	71	79	107
73	79	80	81	79	79	98	128
76	82	81	79	75	81	114	132
77	85	83	77	72	99	144	145
74	79	77	77	79	112	155	142
74	73	71	73	89	142	162	137
69	73	73	77	110	160	166	134
60	67	68	78	124	154	148	116

$\rightarrow$

Output

82	82	114	132
85	85	155	155
85	110	166	166
79	124	166	166

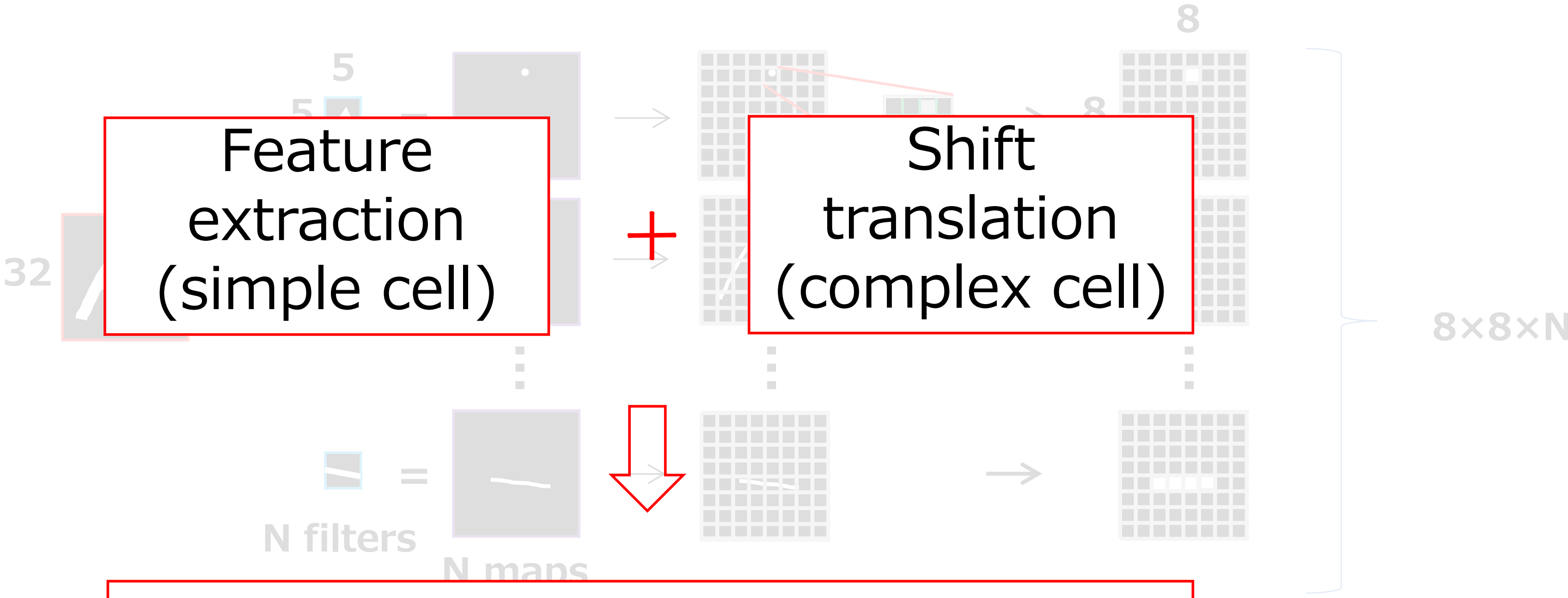
Max pooling

$$u_{ijk} = \max_{(p,q)\in P_{ij}} z_{pqk}$$

Average pooling

$$u_{ijk} = \frac{1}{H^2} \sum_{(p,q)\in P_{ij}} z_{pqk}$$

Two operations: convolution & pooling

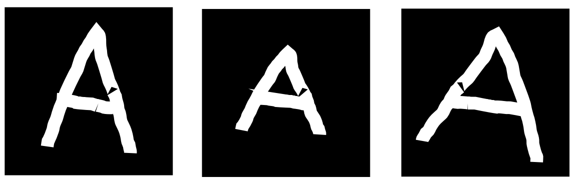


Feature extraction (simple cell)

+

Shift translation (complex cell)

Invariance to small geometric changes



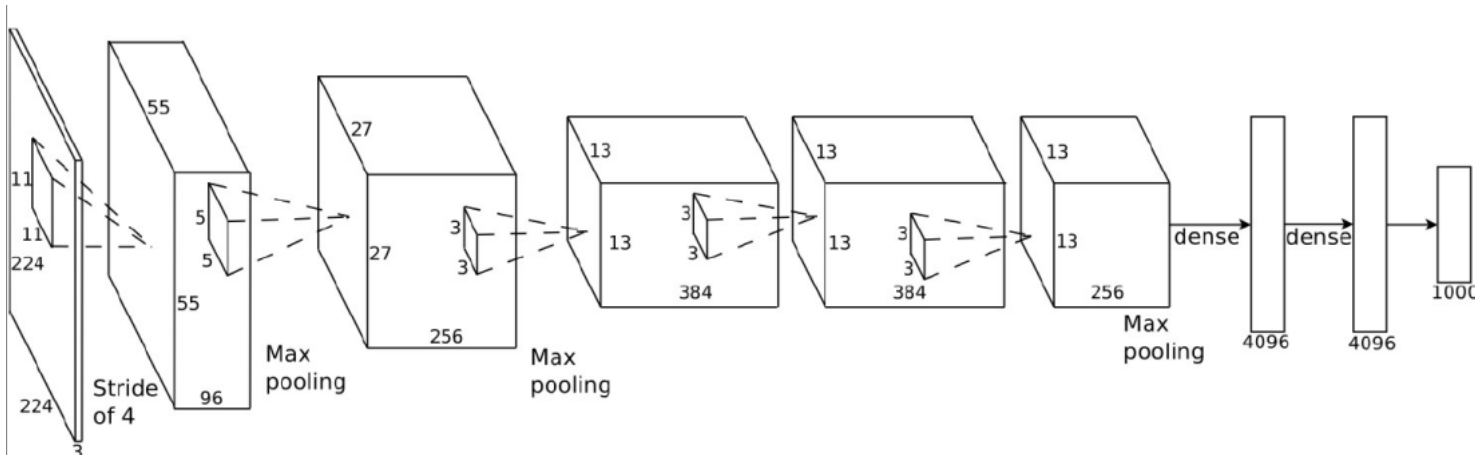
…

connecti
on

- Fixed weights
- Sparse connection

Deep CNNs

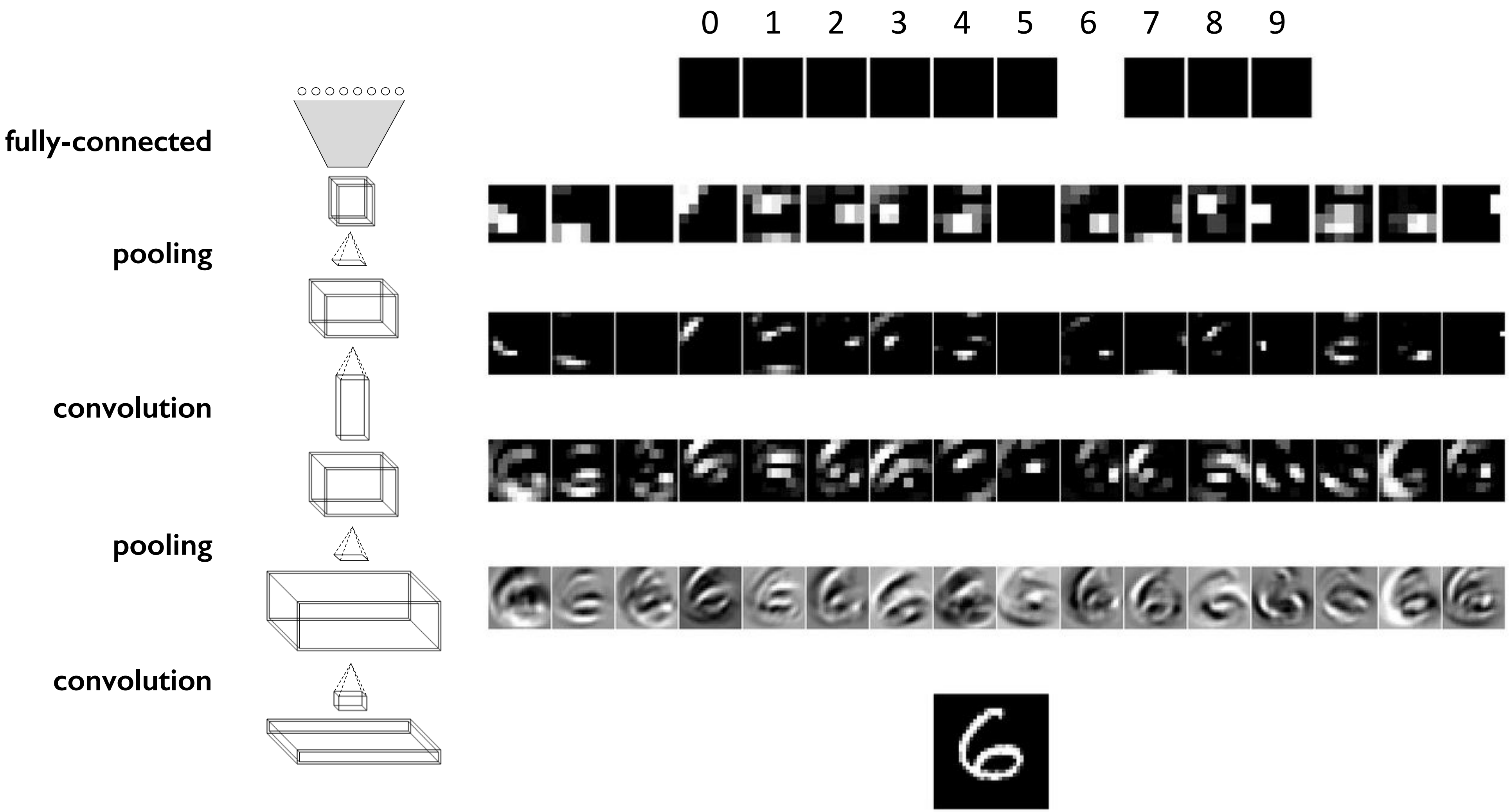
- Feed-forward nets with alternated repetition of conv. layer(s) and a pooling layer followed by fully-connected layer(s)



ILSVRC12のCNN [Krizhevsky+12]

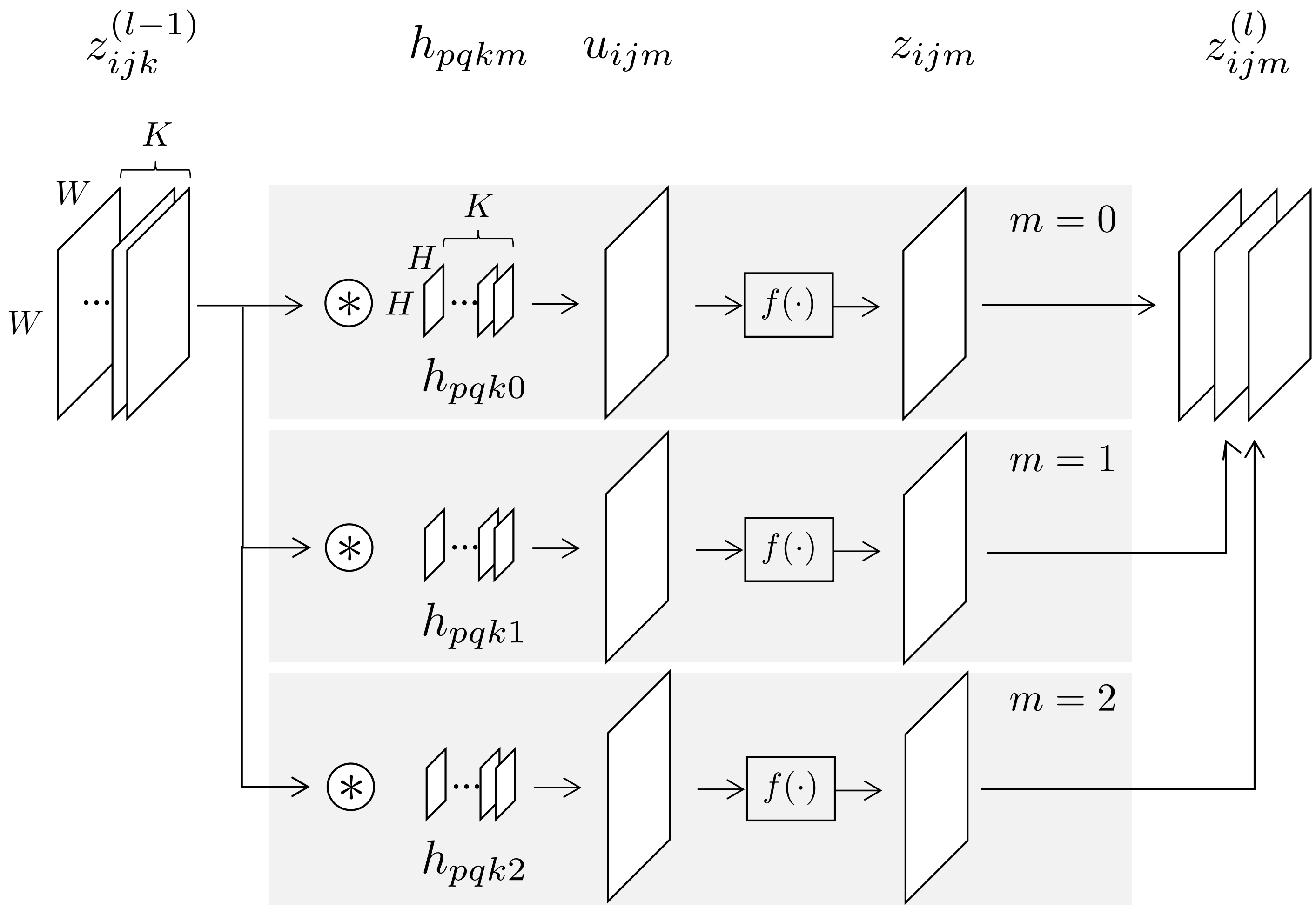
“Alexnet”

Example: CNN behaviors



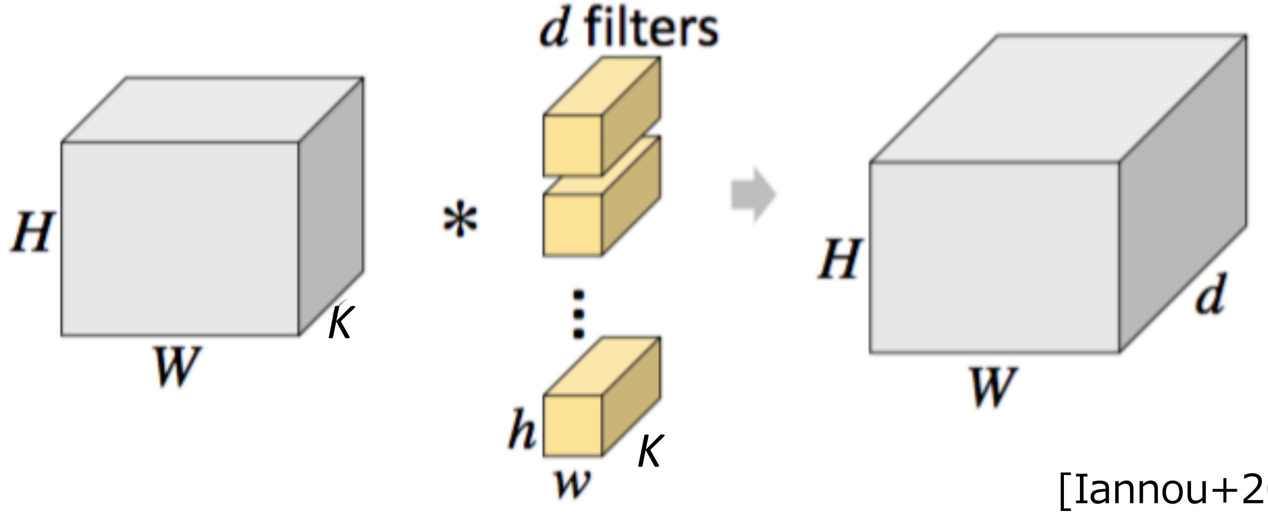
Details of convolution layers

- Multi-channel inputs and outputs



Details of convolution layers

- Computation of sum & product between 3rd order tensors

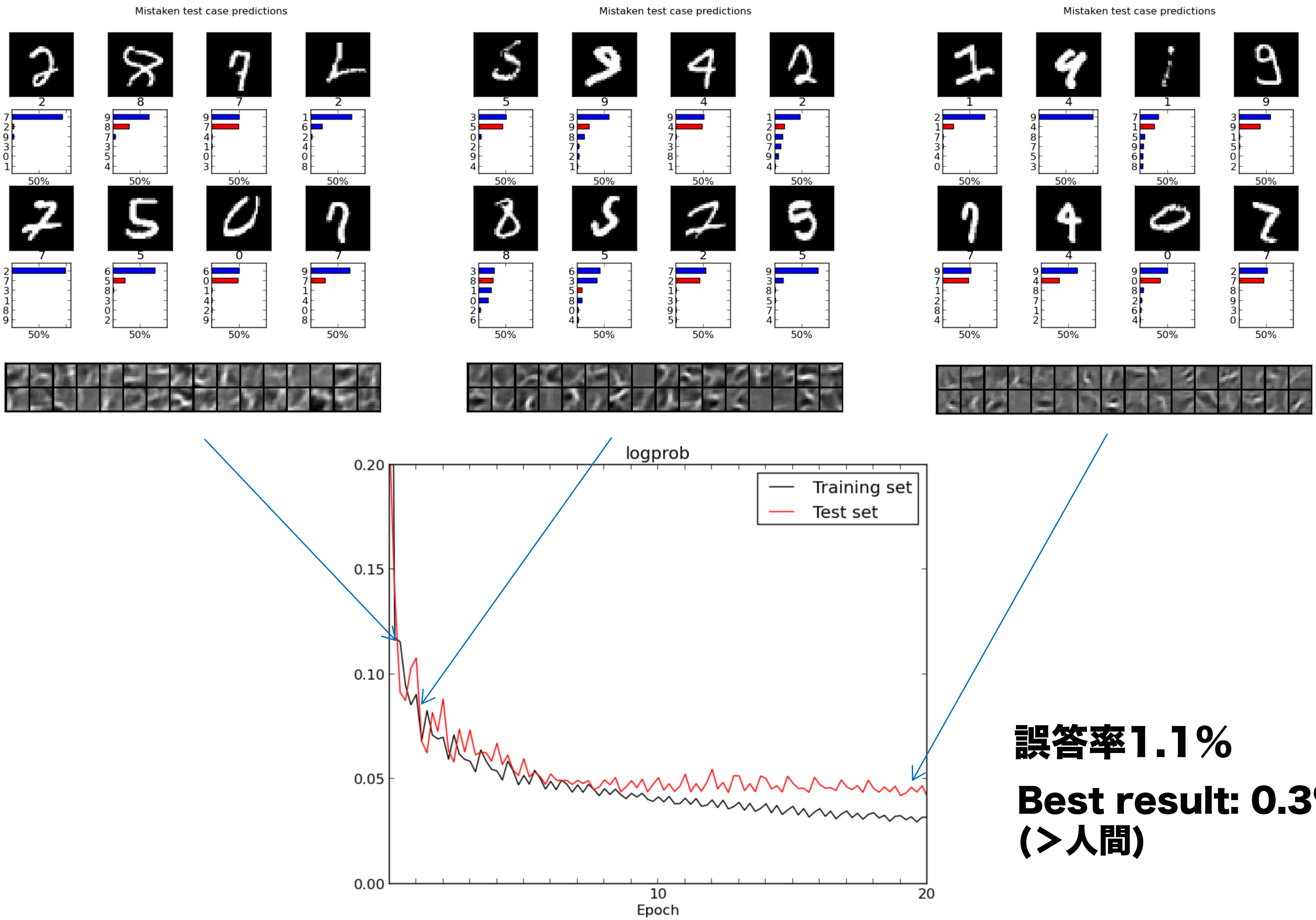


[Iannou+2016]

Training CNNs

- The same as ordinary FF nets
  - SGD with computation of gradients by backprop
- Backprop of deltas in pooling layers
  - Max pooling: the selected input pixel (unit) in the forward process is memorized; a propagated delta is simply transmitted to that pixel (equivalently, a unit weights connection to it)

Example: training of hand-written digits



誤答率1.1%

Best result: 0.3%
(>人間)

Object category recognition

→ Computer → ‘lion’

⇒ ‘lion’

⇒ ‘acoustic guitar’

⇒ ‘electric guitar’

⇒ ‘chambered nautilus’

⇒ ‘table lamp’

⇒ ‘Blenheim spaniel’

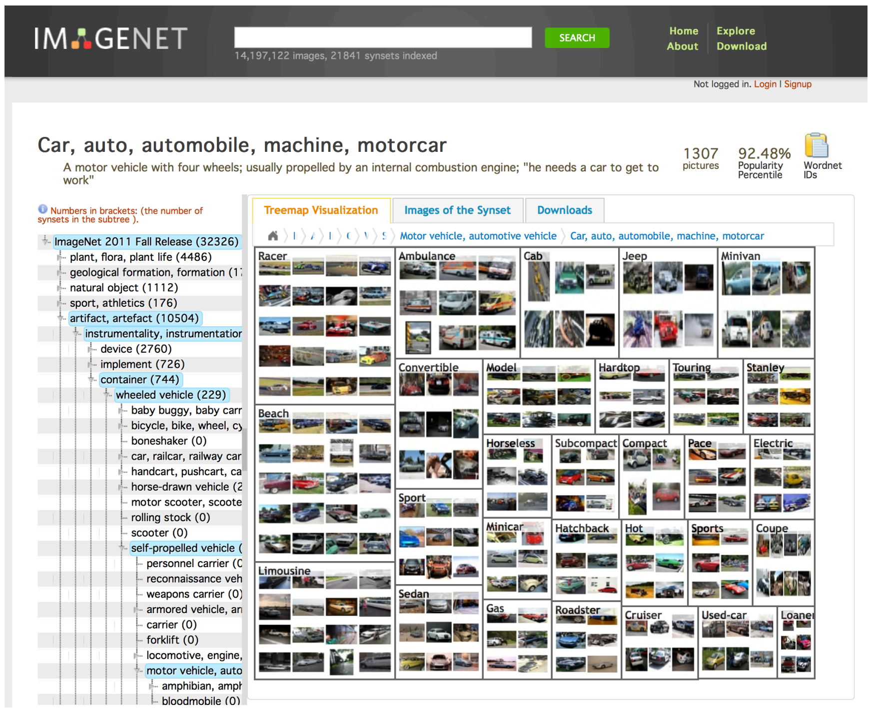
⇒ ‘Japanese spaniel’

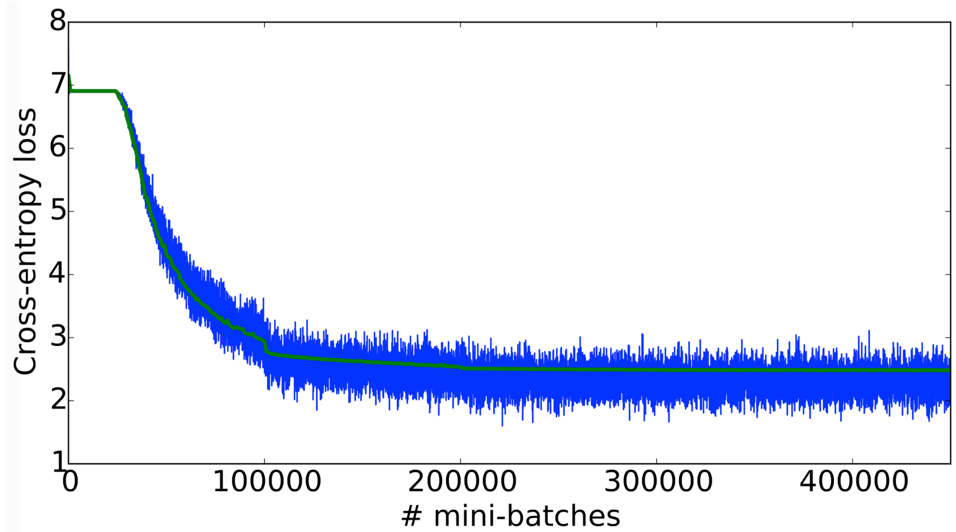
⇒ ‘crane’

Example: training of 1000 object categories

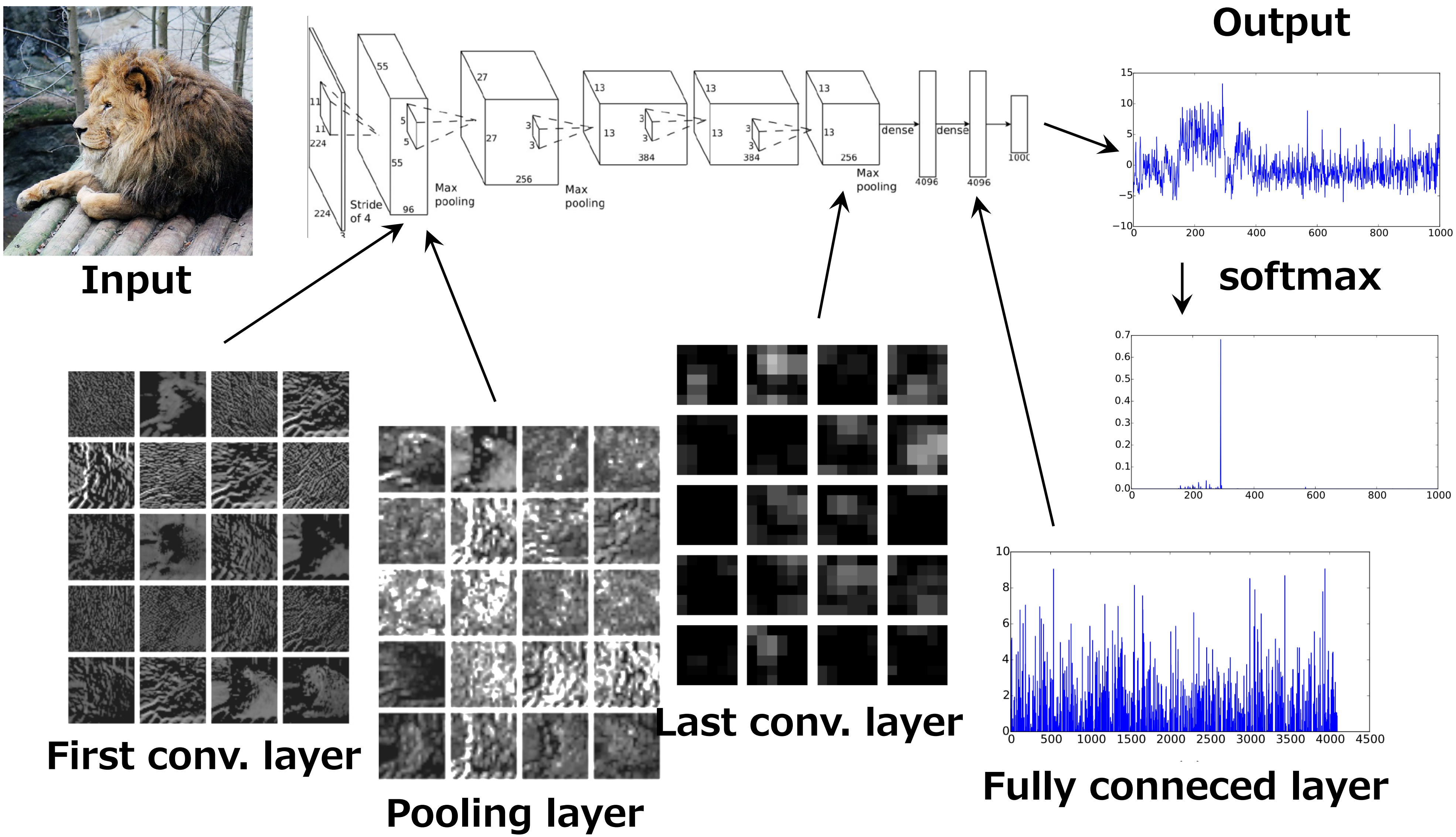
- More than one million training
  - Moer than 1000 images per category
- Training takes days to weeks even using the latest GPUs
- CNNs now surpass human vision in terms of accuracy

(He+, Delving deep into rectifier, 2015)

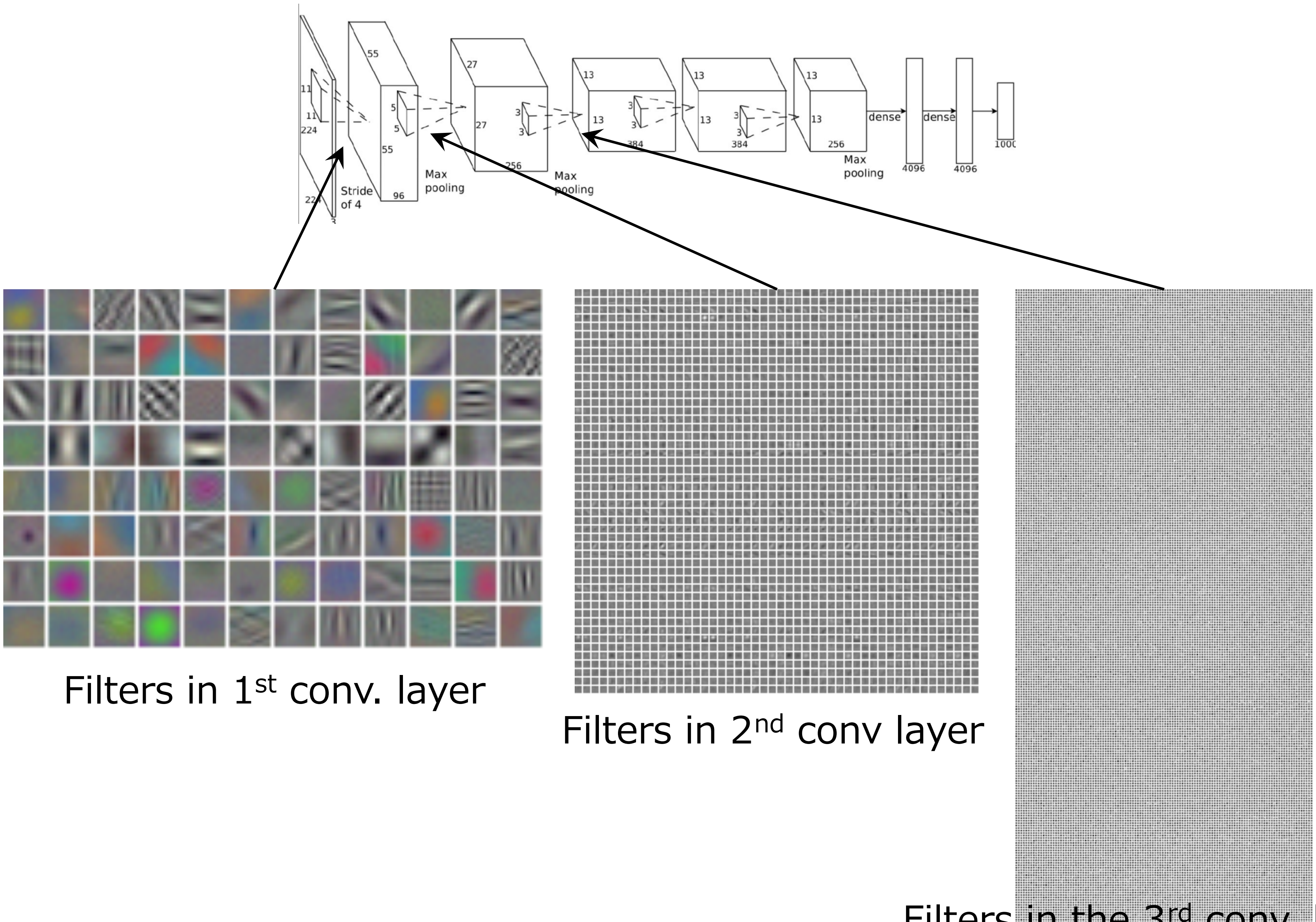




How a trained CNN processes an input



Trained features



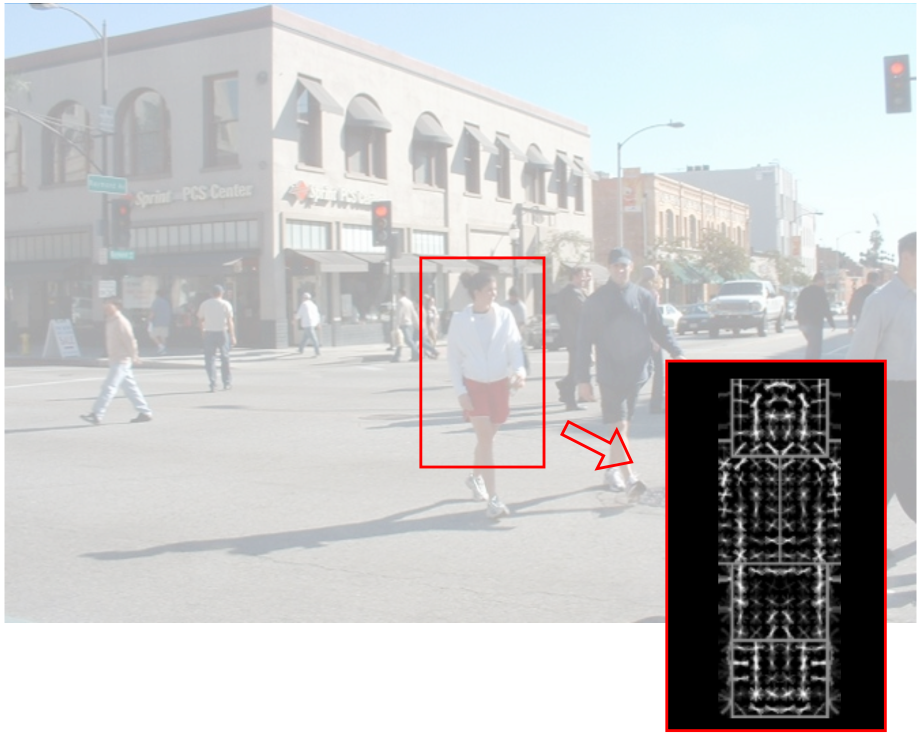
Filters in 1^{st} conv. layer

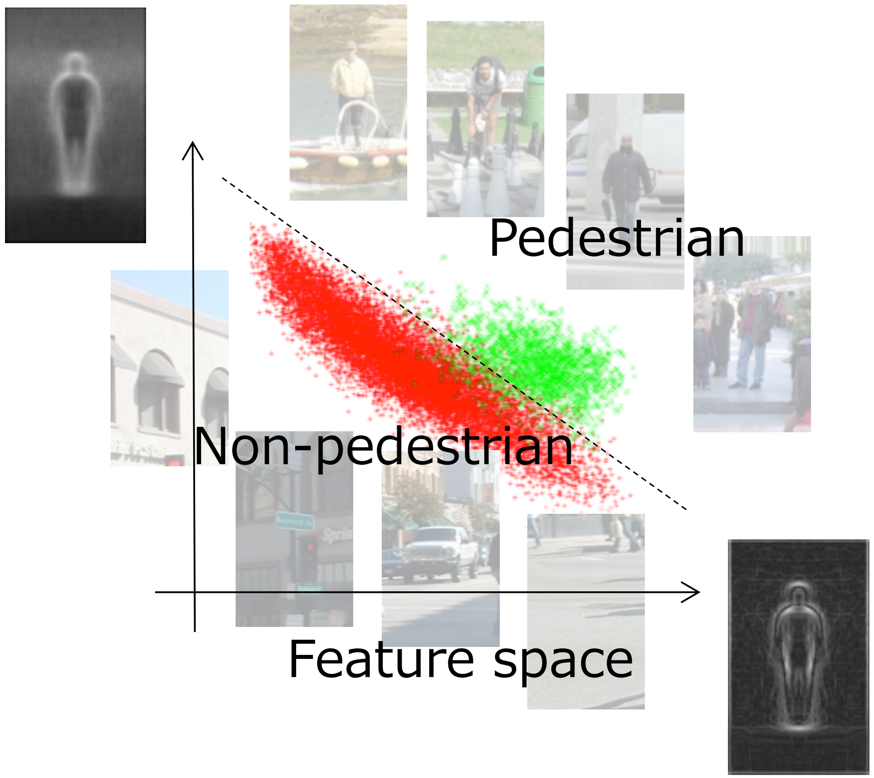
Filters in 2^{nd} conv layer

Filters in the 3^{rd} conv. layer

Standard pipeline of visual recognition

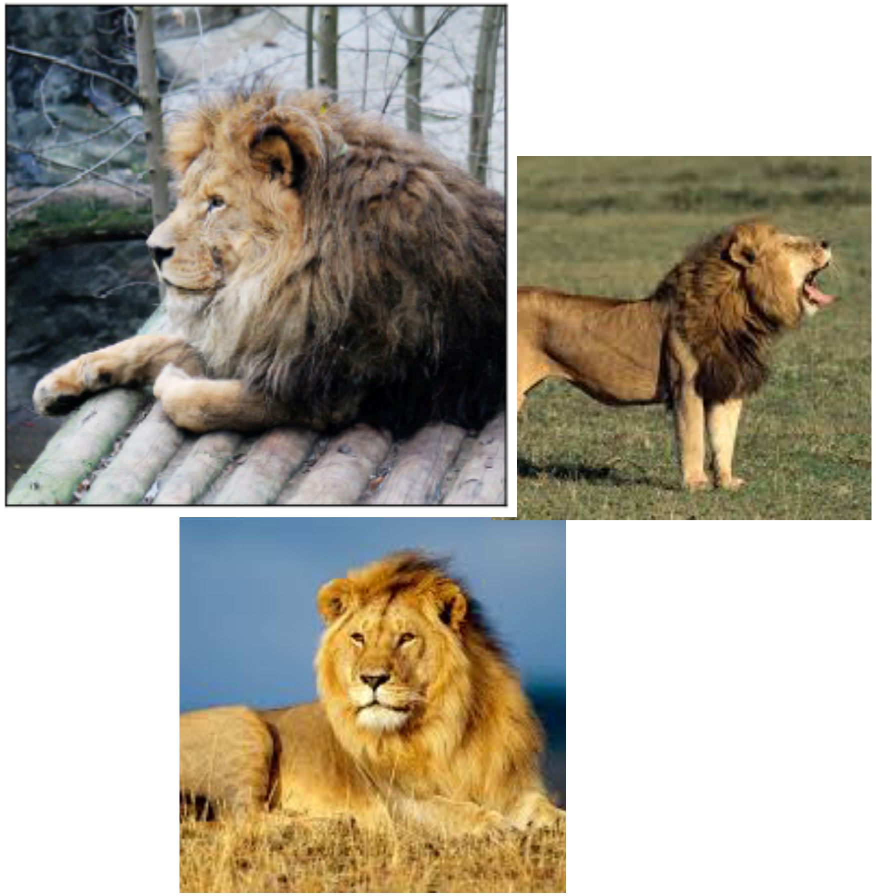
Image → Feature extraction → Feature → Classification → Result





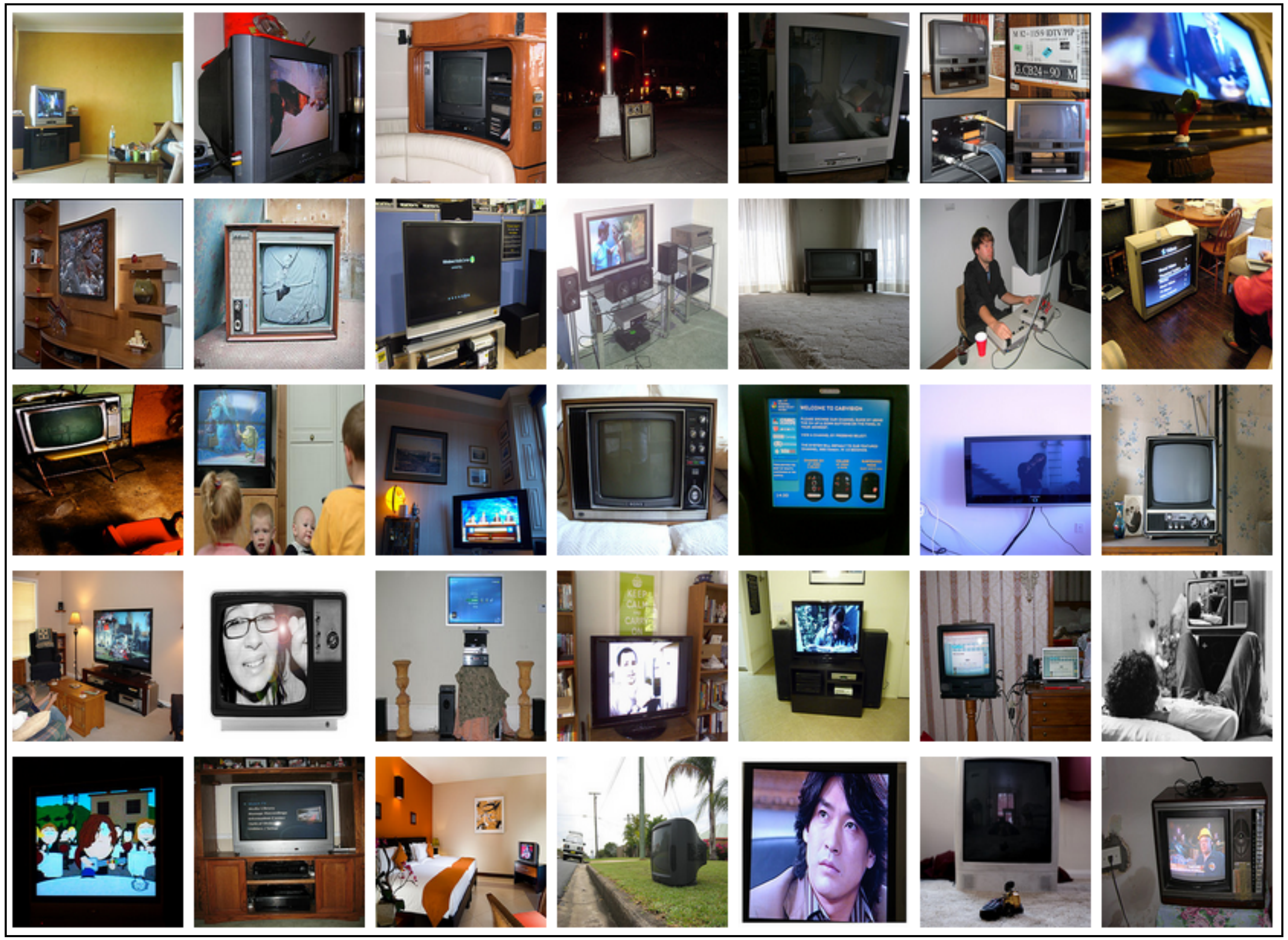
Difficulties with the task

Image → Feature extraction → Feature → Classification → Result



Difficulties with the task

- Invariance: Extracted features need to be tolerant to all sorts of variation in the same category

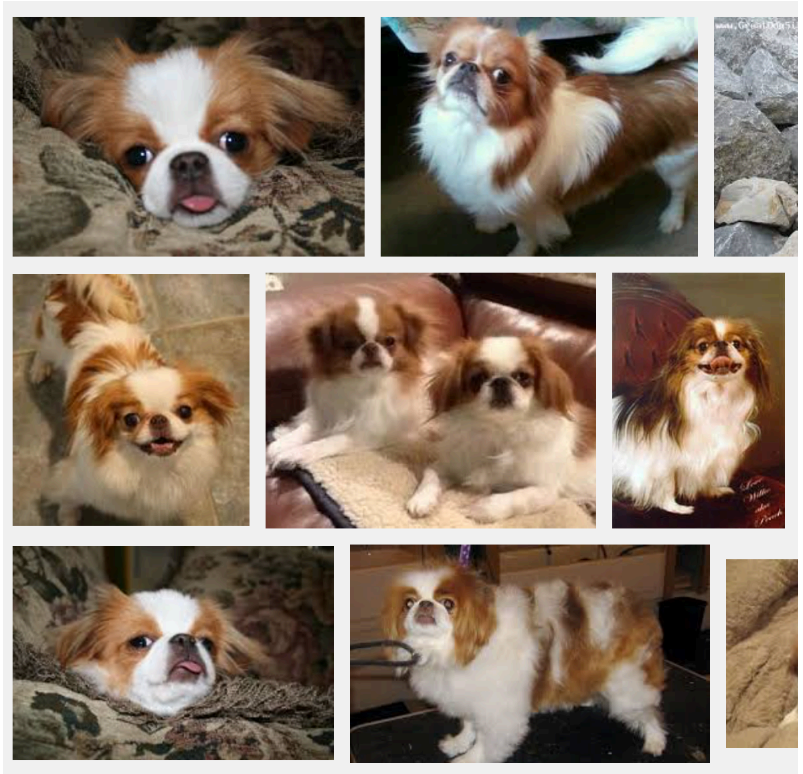


“Television set”

Difficulties with the task

- Discriminability: Extracted features need to be sensitive to small differences between categories

'Blenheim spaniel'



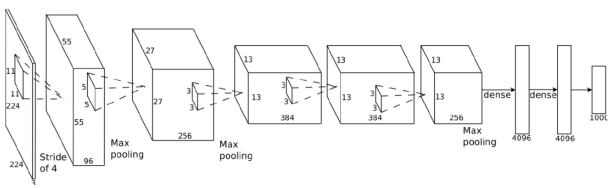
'Japanese spaniel'

Recent design of CNNs

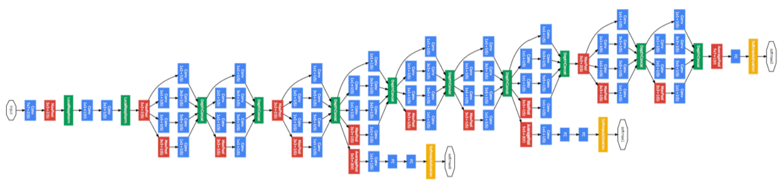
表 1　代表的なモデルのパラメータ数および演算回数．畳込み層および全結合層での合計と総計．

モデル		Alexnet	VGGNet	GoogLeNet	ResNet	
Conv.	層	5	13	21	151	Layers
	重み	380 万	0.15 億	580 万	-	Weights
	演算	10.8 億	153 億	15 億	113 億	Operations
FC	層	3	3	1	1	Layers
	重み	0.59 億	1.24 億	100 万	200 万	Weights
	演算	0.59 億	1.24 億	100 万	200 万	Operations
Total	重み	0.62 億	1.38 億	680 万	-	Weights
	演算	11.4 億	155 億	15 億	113 億	Operations

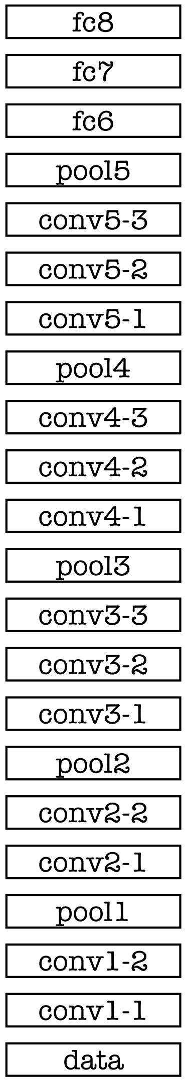
AlexNet [Krizhevsky+12]



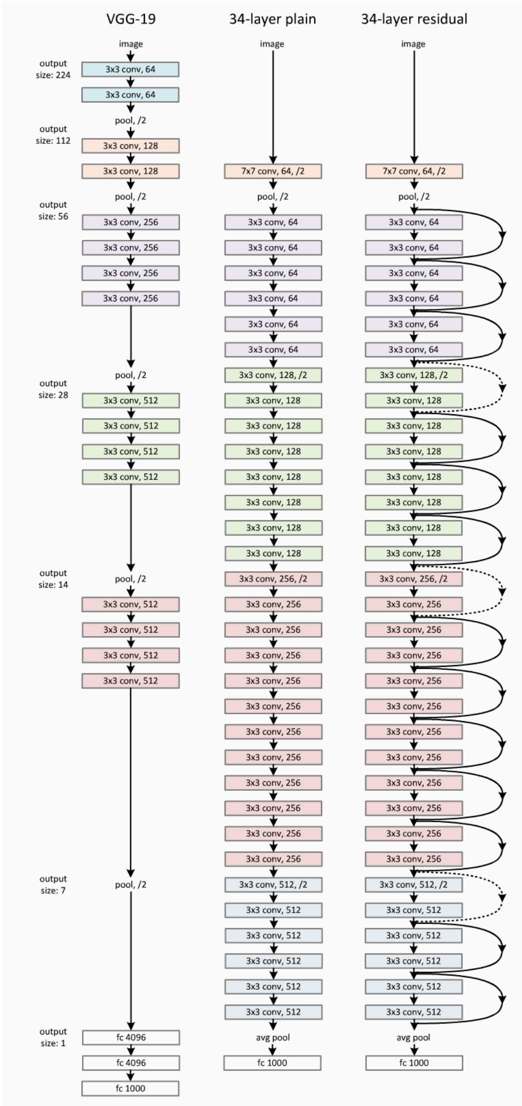
GoogLeNet [Szegedy+14]



VGGNet [Simonyan+14]



ResNet [He+15]



5th (and final) assignments

1. Briefly explain your research for your thesis (MSc/PhD), i.e., what you (or your lab) are studying now.
2. Explain how you think machine learning(ML) including deep learning can be used to solve some of the problems you (or your lab) are tackling now.
   - If ML or DL has already been used, explain how it can be better used for the problem.
3. If you think the use of ML is irrelevant for your research, find a problem to which ML has not yet been applied and explain how ML can be used for it.