

Flexible, High Performance Convolutional Neural Networks for Image Classification

Dan C. Cireşan, Ueli Meier, Jonathan Masci, Luca M. Gambardella, Jürgen Schmidhuber

IDSIA, USI and SUPSI
Manno-Lugano, Switzerland
{dan, ueli, jonathan, luca, juergen}@idsia.ch



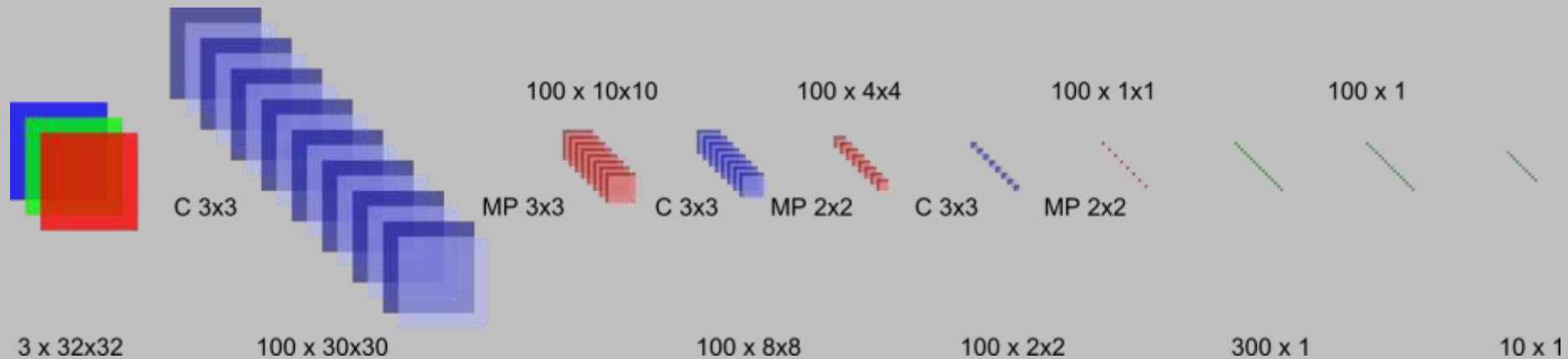
Introduction

- Image recognition: text, objects.
- Features or raw pixels? Learning features.
- Convolutional Neural Networks.
- How to train huge nets? GPUs ...
- Evaluation protocol for standard benchmarks.

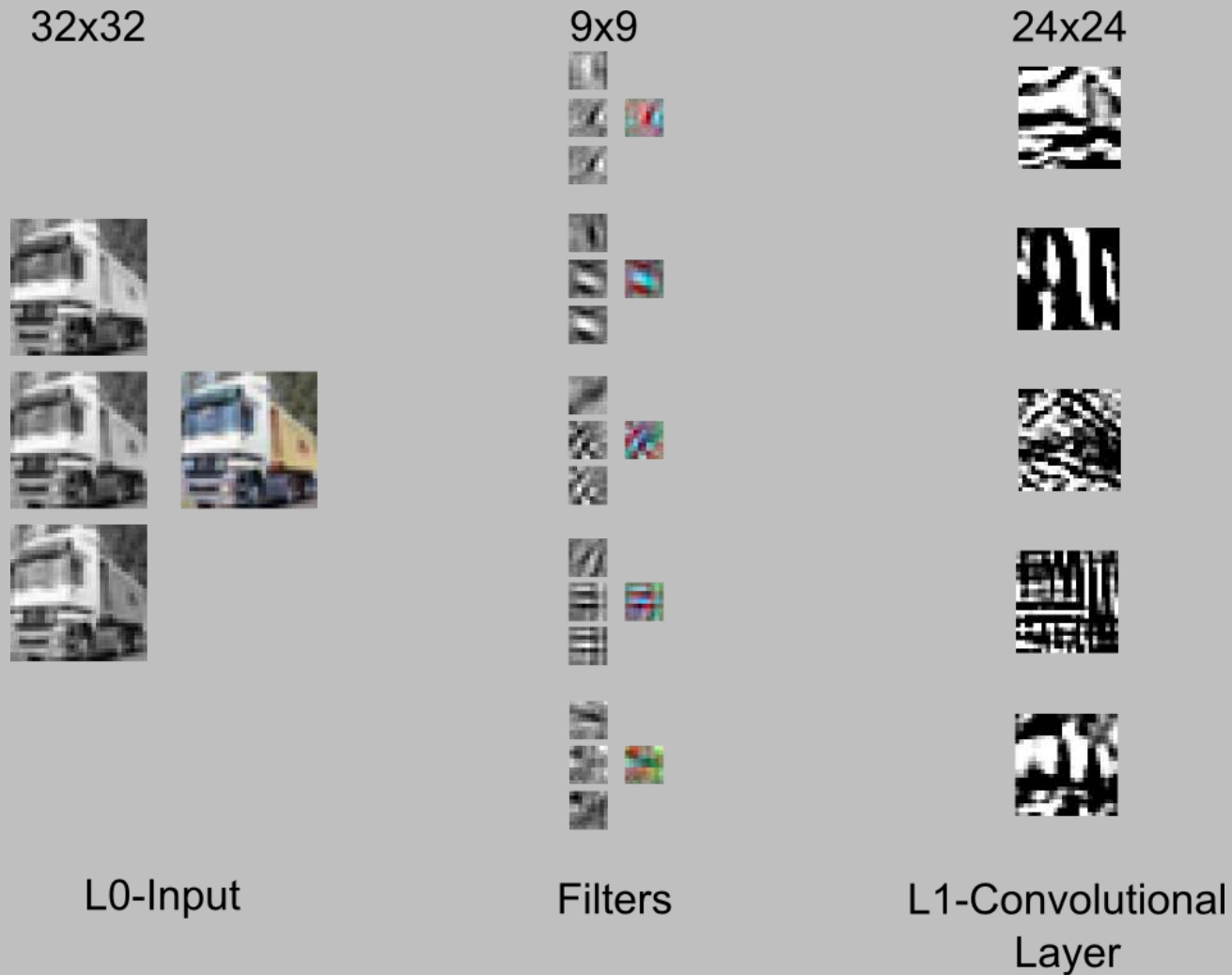


Convolutional Neural Networks (CNNs)

- Hierarchical architecture for image recognition, loosely inspired from biology.
- Introduced by Fukushima (80) and refined by LeCun et al.(98), Riesenhuber et al.(99), Simard et al.(03), Behnke (03).
- Fully supervised, with randomly initialized filters, trained minimizing the misclassification error.
- Flexible architecture.



Convolutional layer



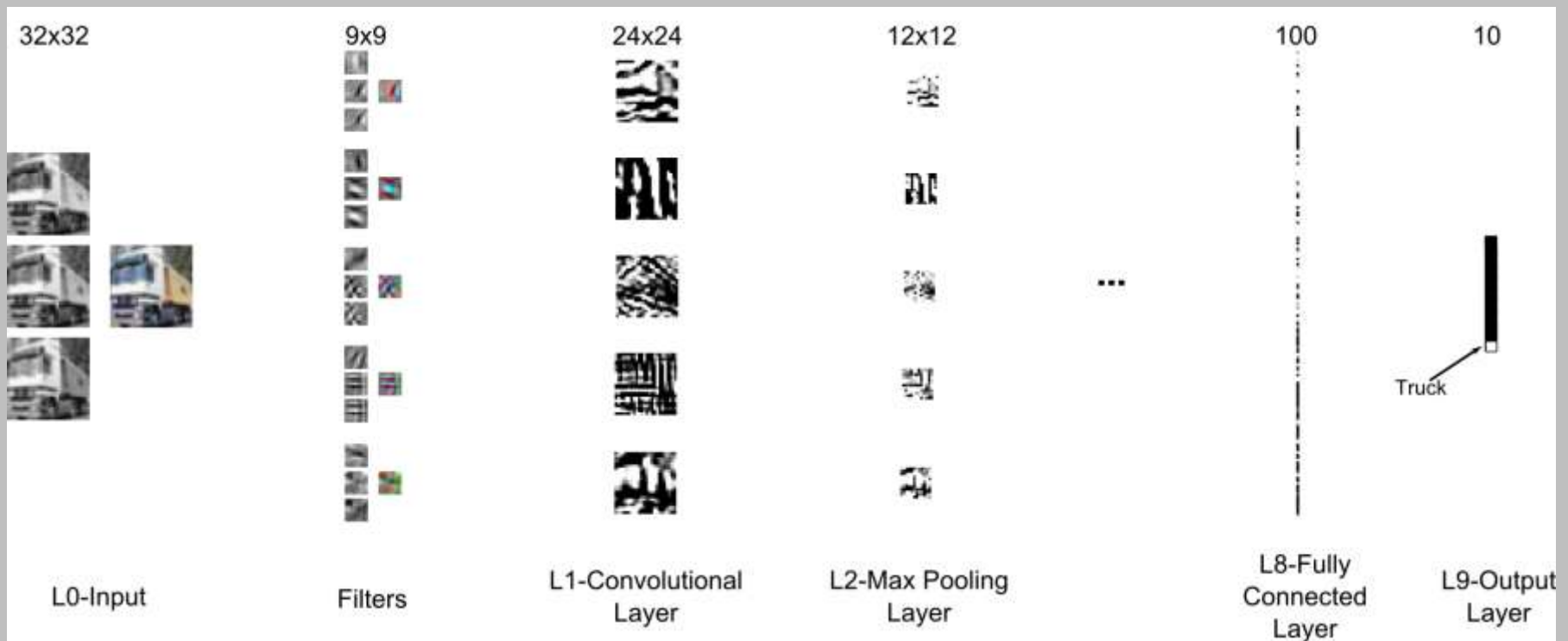
Max-pooling layer

- Introduces small translation invariance
- Improves generalization



Fully connected layer

- One output neuron per class normalized with soft-max activation function



Graphics processing units (GPUs)

- 8 x GTX 480/580 1.5GB RAM
- >12 TFLOPS (theoretical speed)
- 40-80x speed-up compared with a single threaded CPU version of the CNN program (one day on GPU instead of two months on CPU)

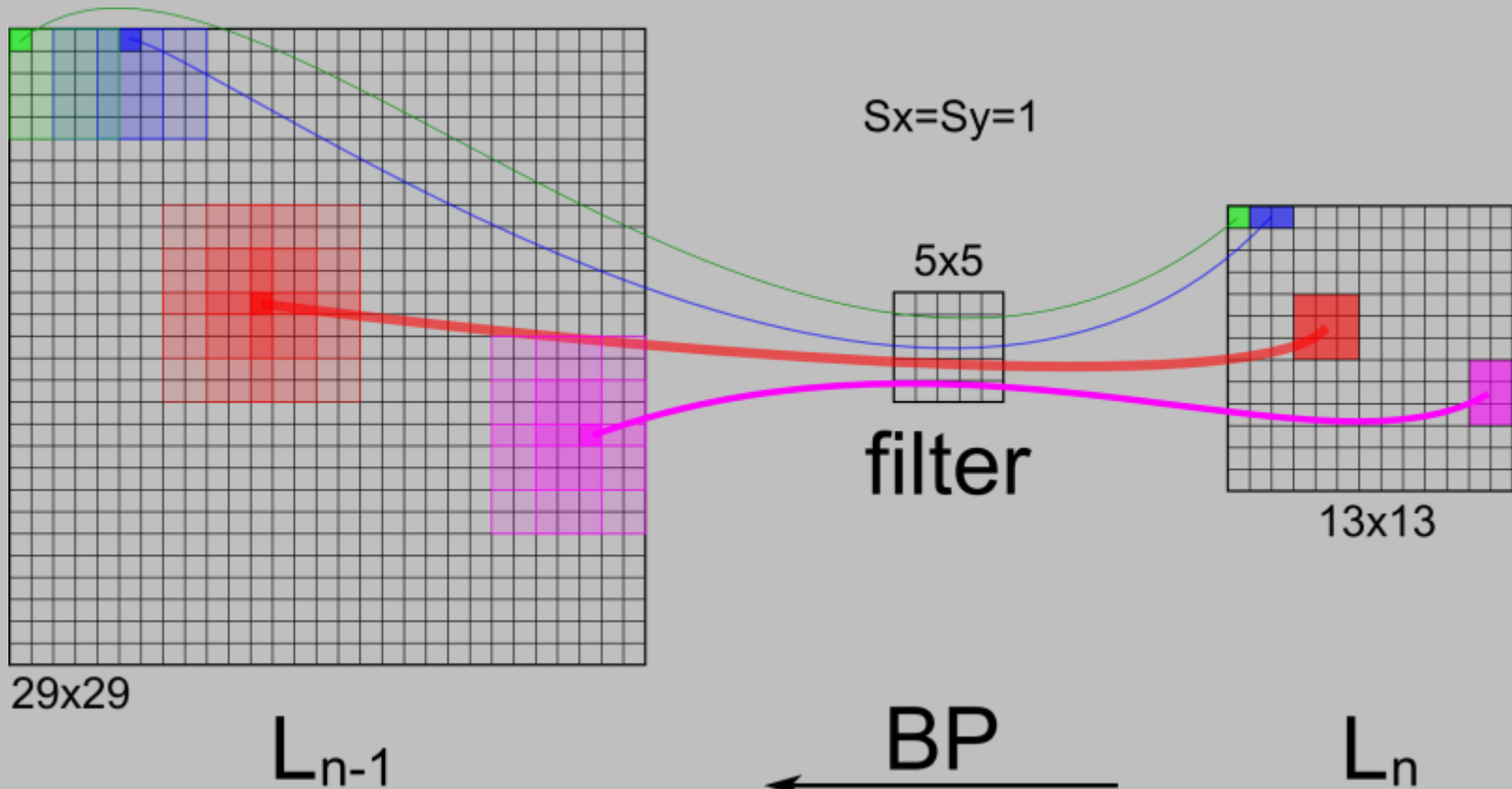


Back-propagation of deltas

- Uses pooling of deltas.

$$\max\left(\left\lfloor \frac{i - K_x + 1}{S_x + 1} \right\rfloor, 0\right) \leq x \leq \min\left(\left\lfloor \frac{i}{S_x + 1} \right\rfloor, M_x - 1\right)$$

$$\max\left(\left\lfloor \frac{i - K_y + 1}{S_y + 1} \right\rfloor, 0\right) \leq y \leq \min\left(\left\lfloor \frac{i}{S_y + 1} \right\rfloor, M_y - 1\right)$$



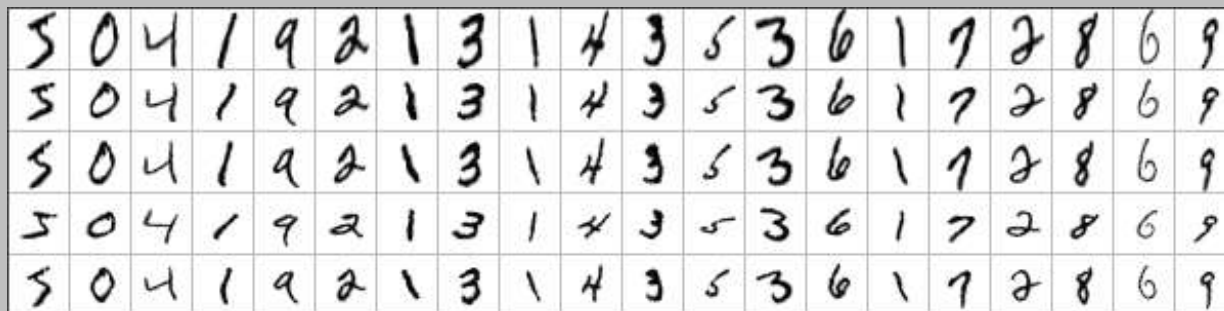
Experiments

- Distorting images
- Datasets:
 - Handwritten digits: MNIST
 - 3D models: NORB
 - Natural images: CIFAR10
- Evaluation protocol
 - Repeat experiment and compute mean and standard deviation



Distortions

- MNIST
 - Translation
 - Rotation
 - Scaling
 - Elastic
- NORB and CIFAR10: only translation (random for both axes, maximum 5%)
- Greatly improves generalization and recognition rate
- Border effects



MNIST

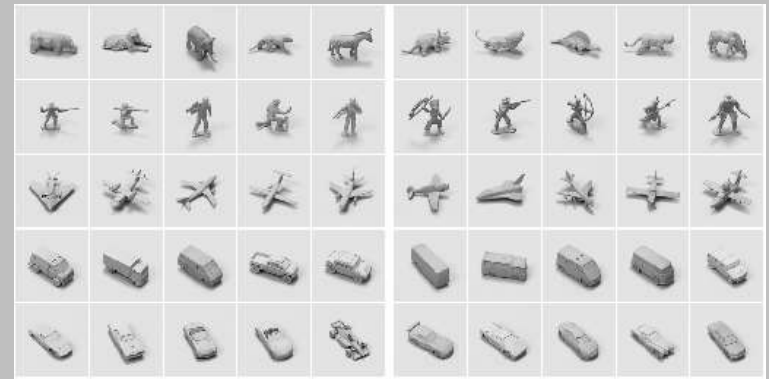
- 28x28 grayscale images
- 60000 for training and 10000 for testing
- Simard et al. (2003) – 0.40%, Ciresan et al. (2010) – 0.35%
- 30 out of 35 digits have a correct second prediction

#M, #N in Hidden Layers	Tfbv [%]
20M-60M	1.02
20M-60M-150N	0.55
20M-60M-100M-150N	0.38
20M-40M-60M-80M-100M-120M-150N	0.35



Small NORB

- 48600 96x96 stereo images
- 5 classes with 10 instances
- 5 instances for training and 5 for testing
- bad/challenging dataset, only 5 instances/class, some instances from test set are completely different than the one from training set
- IP maps (Mexican hat) are needed only for this data set
- previous state of the art: Behnke et al. 2.87%



trans. [%]	IP	TfbV [%]	runs	time/epoch [s]
0	no	7.86 ± 0.55	50	1143
5	no	4.71 ± 0.57	50	1563
0	yes	3.94 ± 0.48	50	1658
5	yes	2.53 ± 0.40	100	2080



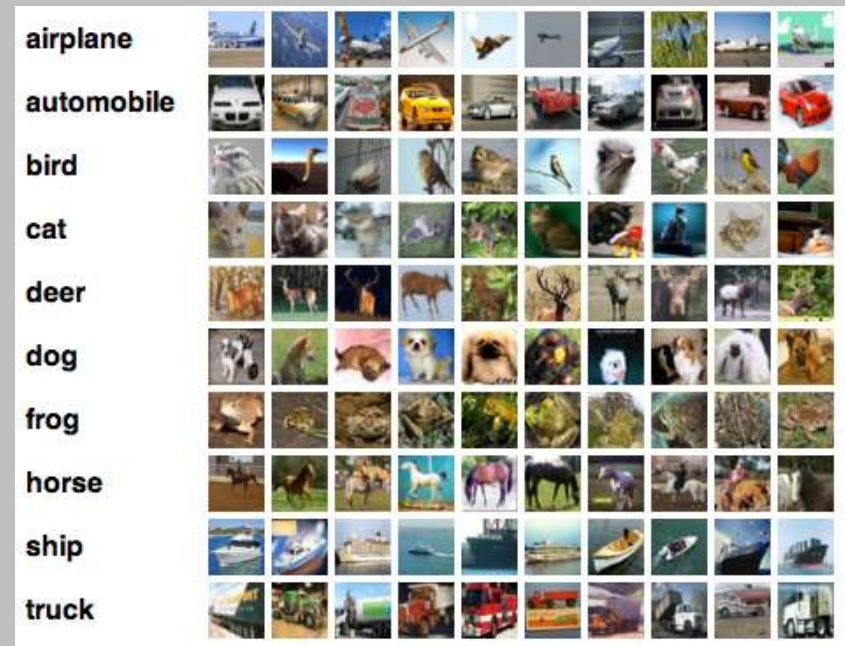
	mammal	human	plane	truck	car	all
mammal	0	22	13	2	3	40
human	35	0	0	0	0	35
plane	88	0	0	60	37	185
truck	19	0	0	0	7	26
car	79	0	0	246	0	325



CIFAR10

- small, 32x32 pixels color images
- complex backgrounds
- 10 classes
- 50000 training images
- 10000 test images
- 19.51% error rate (state of the art)

trans. [%]	IP	TfbV [%]	Runs	time/epoch [s]
0; 100M	no	28.87 ± 0.37	11	93
0; 100M	edge	29.11 ± 0.36	15	104
5; 100M	no	20.26 ± 0.21	11	111
5; 100M	edge	21.87 ± 0.57	5	120
5; 100M	hat	21.44 ± 0.44	4	136
5; 200M	no	19.90 ± 0.16	5	248
5; 300M	no	19.51 ± 0.18	5	532
5; 400M	No	19.54 ± 0.16	5	875



first layer filters



Conclusions

- Our big deep nets combining CNN and other ideas are now state of the art for many image classification tasks.
- No need to extract handcrafted features.
- Supervised training with simple gradient descent training is best. No need for unsupervised pre-training (e.g. autoencoders) in case of sufficient training samples.
- Distorting the training set improves recognition rate on unseen data.
- CPUs are not enough anymore, use GPUs which are 2 orders of magnitude faster.
- Robust (smallest error rates) and fast enough (10^3 - 10^4 images/s) for immediate industrial applications.



What is next?

- Results on all benchmarks already improved 20-50% compared with this paper.
- Test the CNNs on different datasets:
 - already done:
 - Chinese characters: 3755 classes, >1M characters, 6.5% error rate, first place at ICDAR 2011 competition
 - Traffic signs: 43 classes, <1% error rate, first place at IJCNN 2011 competition
 - All Latin alphabet (NIST SD 19, >0.8M characters): state of the art results (ICDAR 2011)
 - next:
 - CALTECH 101 & 256, ImageNet, cluttered NORB
 - medical images
- Use CNNs for general scene understanding (segmentation).
- Robot vision applications (detect type, position and orientation of objects in a scene).
- Computer vision applications (de-blurring, segmentation, similarity check etc.).
- Try to reach and surpass human image classification performance.
- More at www.idsia.ch/~ciresan

