

# 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





Timisoara, Romania  
PhD at UPT

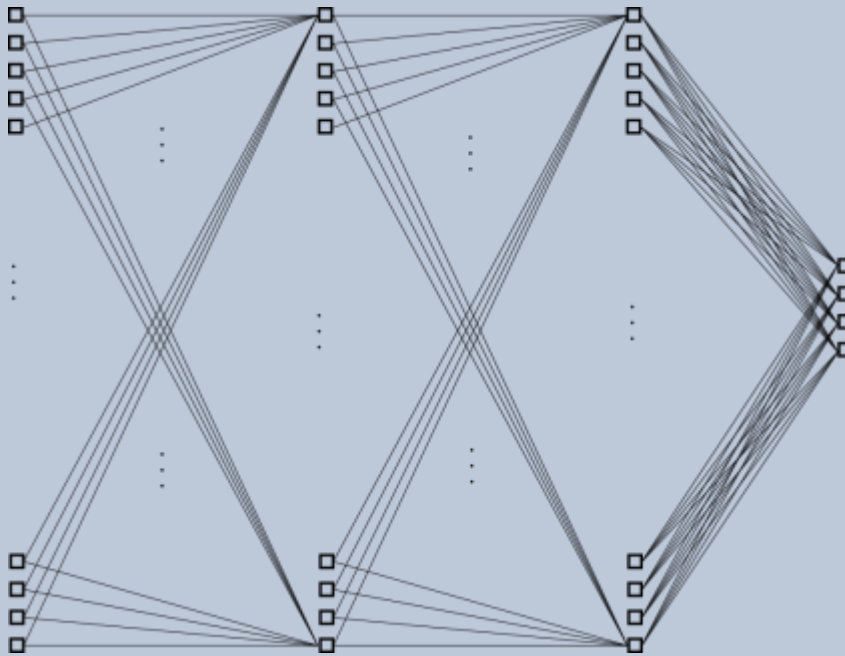
Lugano, Switzerland  
postdoc at IDSIA

# Introduction

- Image recognition: text, objects.
- Convolutional Neural Networks.
- How to train huge nets? GPUs ...
- Evaluation protocol for standard benchmarks.
- Other applications.



# Multi Layer Perceptrons



- simple, uniform architecture
- fully connected
- many weights, one weight per connection
- very general
- disregards 2D information (neighboring info)
- not good for classifying complex images
- We trained the first big and deep MLP, breaking the record on MNIST (Neural Computation 2010)

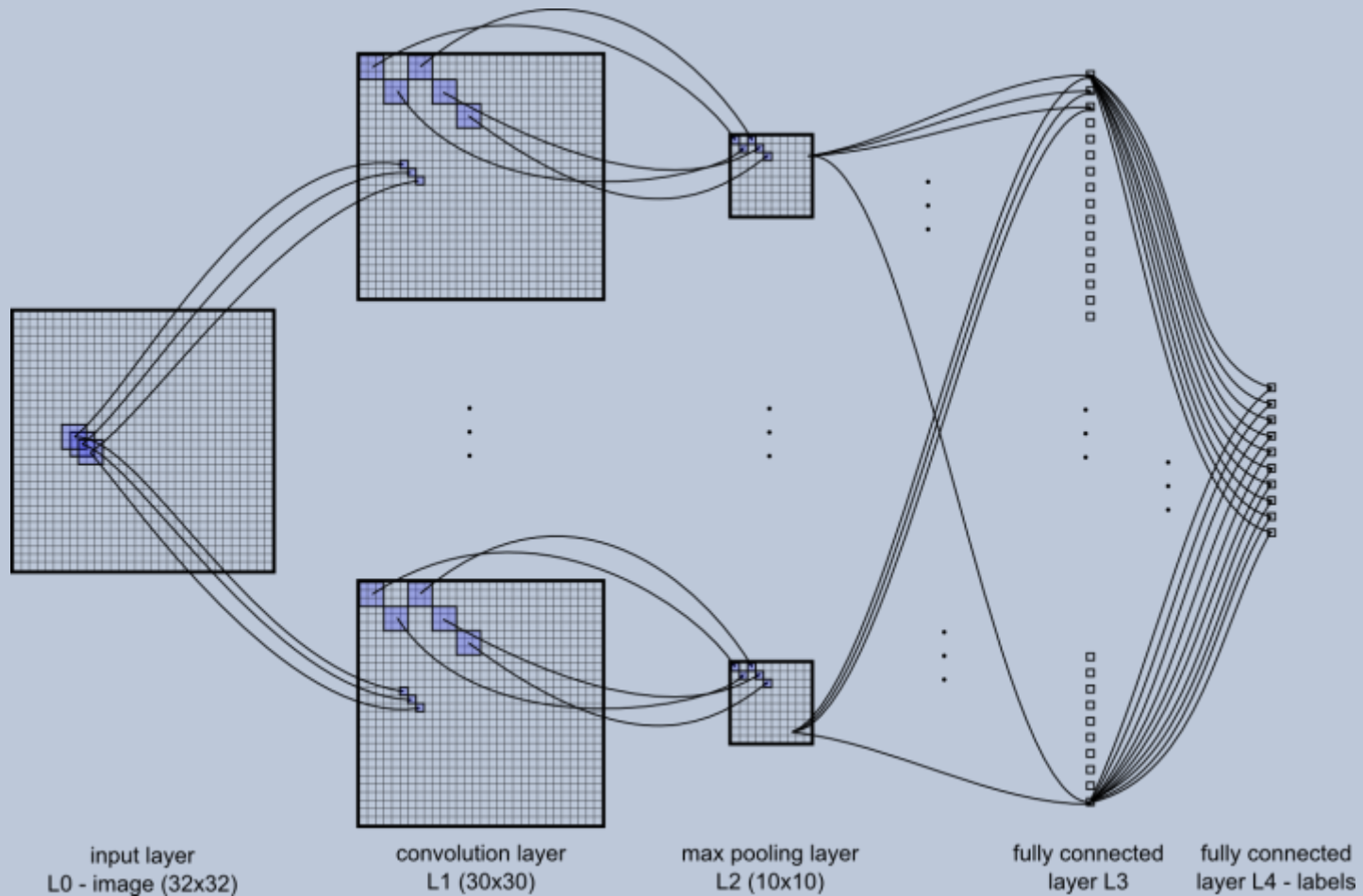


# Convolutional Neural Networks (CNNs)

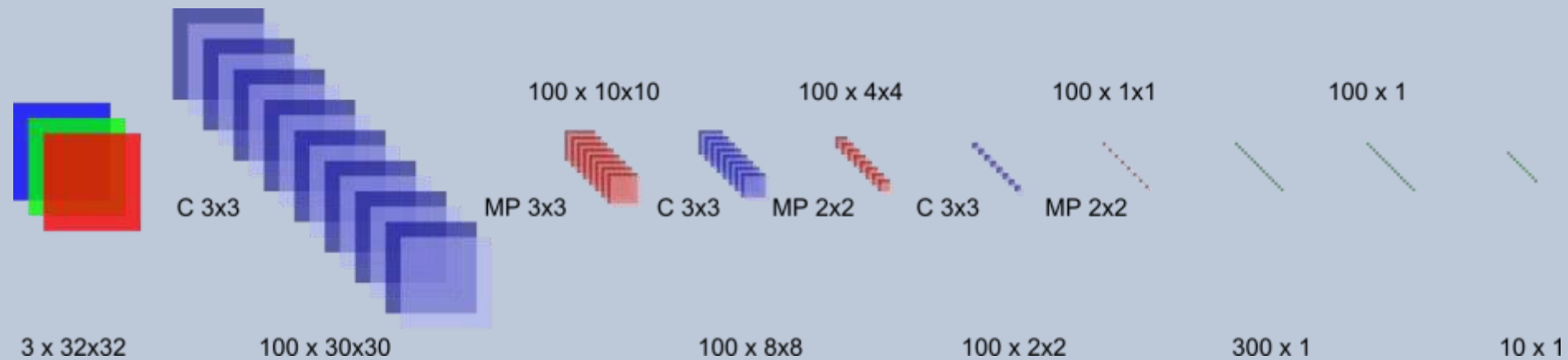
- Hierarchical architecture designed for image recognition, loosely inspired from biology.
- Introduced by Fukushima (80) and refined by LeCun et al.(98), Riesenhuber et al.(99), Simard et al.(03), Behnke (03).
- Uses the neighboring information (preserves the 2D information).
- Shared weights.
- Fully supervised, with randomly initialized filters, trained minimizing the misclassification error.
- Flexible architecture.



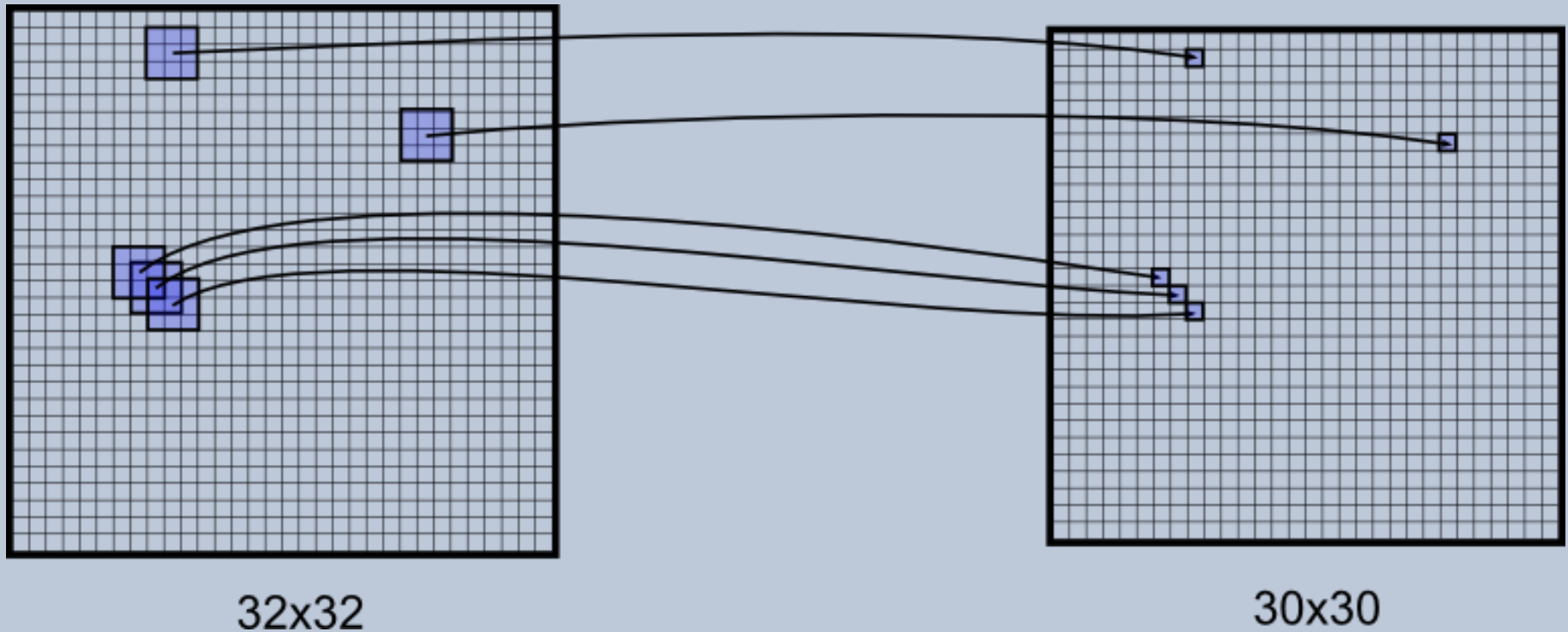
# Convolutional Neural Networks (CNNs)



# Convolutional Neural Networks (CNNs)



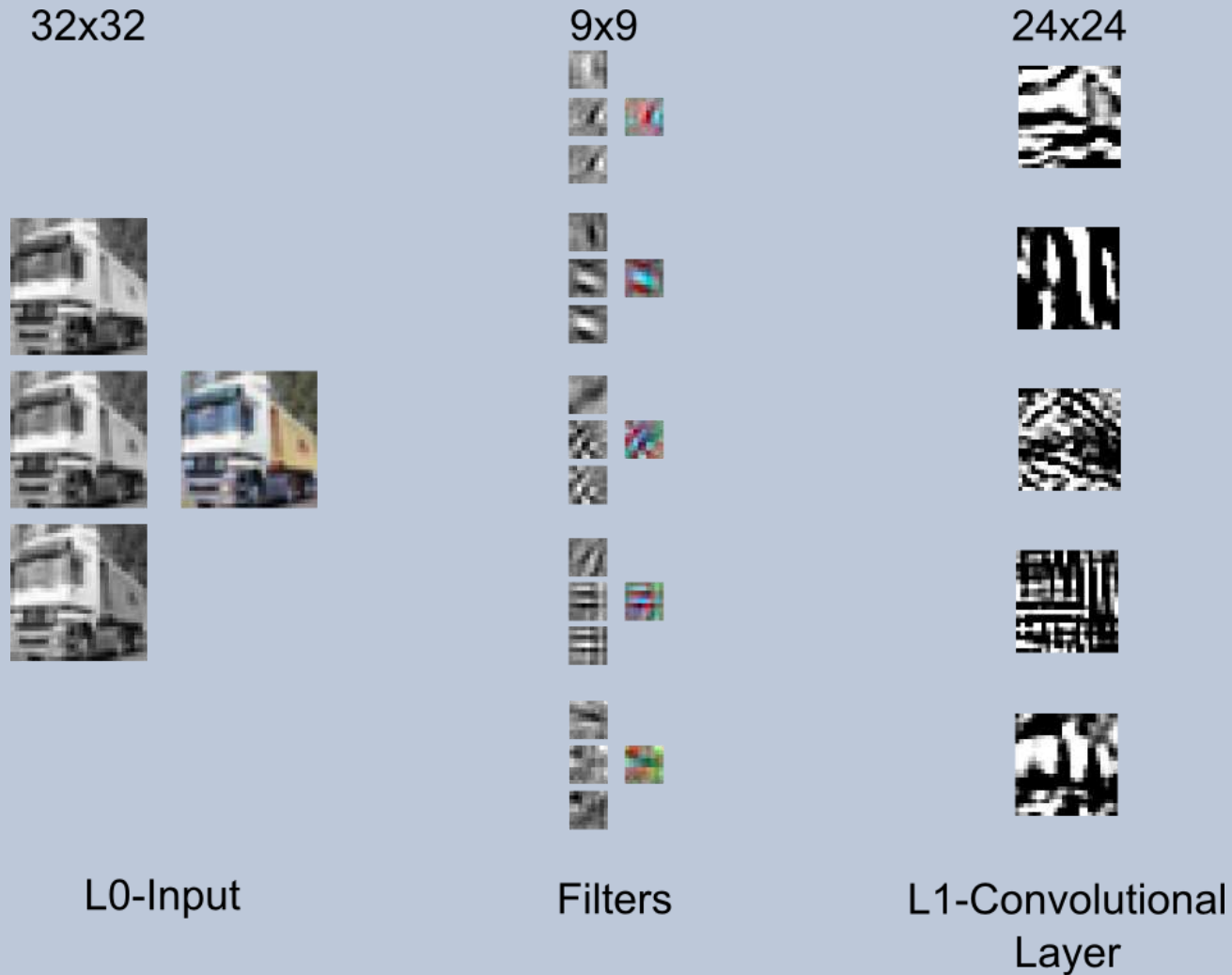
# Convolutional layer



- 3x3 kernel (filter)  $\Rightarrow$  only 10 weights ( $3 \times 3 + 1$ ) shared by all 900 neurons
- each neuron has 9 source neurons
- $900 \times (9 + 1) = 9000$  connections



# Convolutional layer

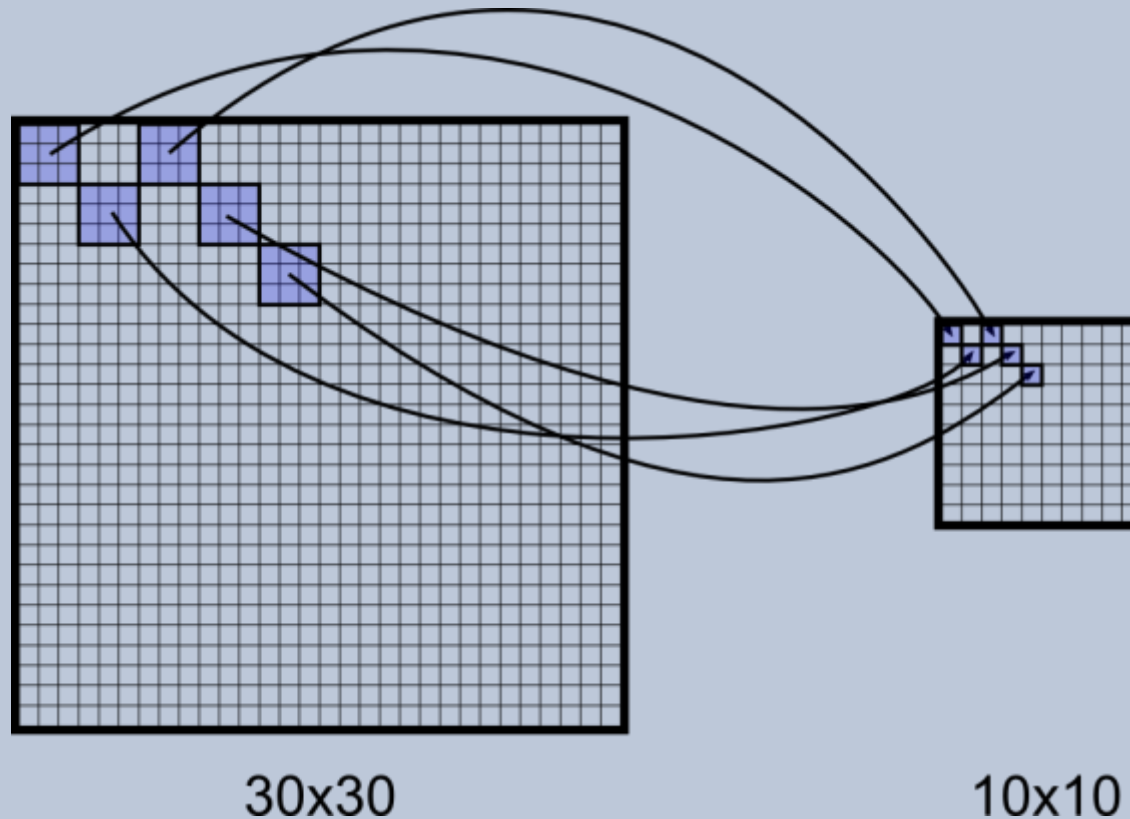


# Convolutional layer



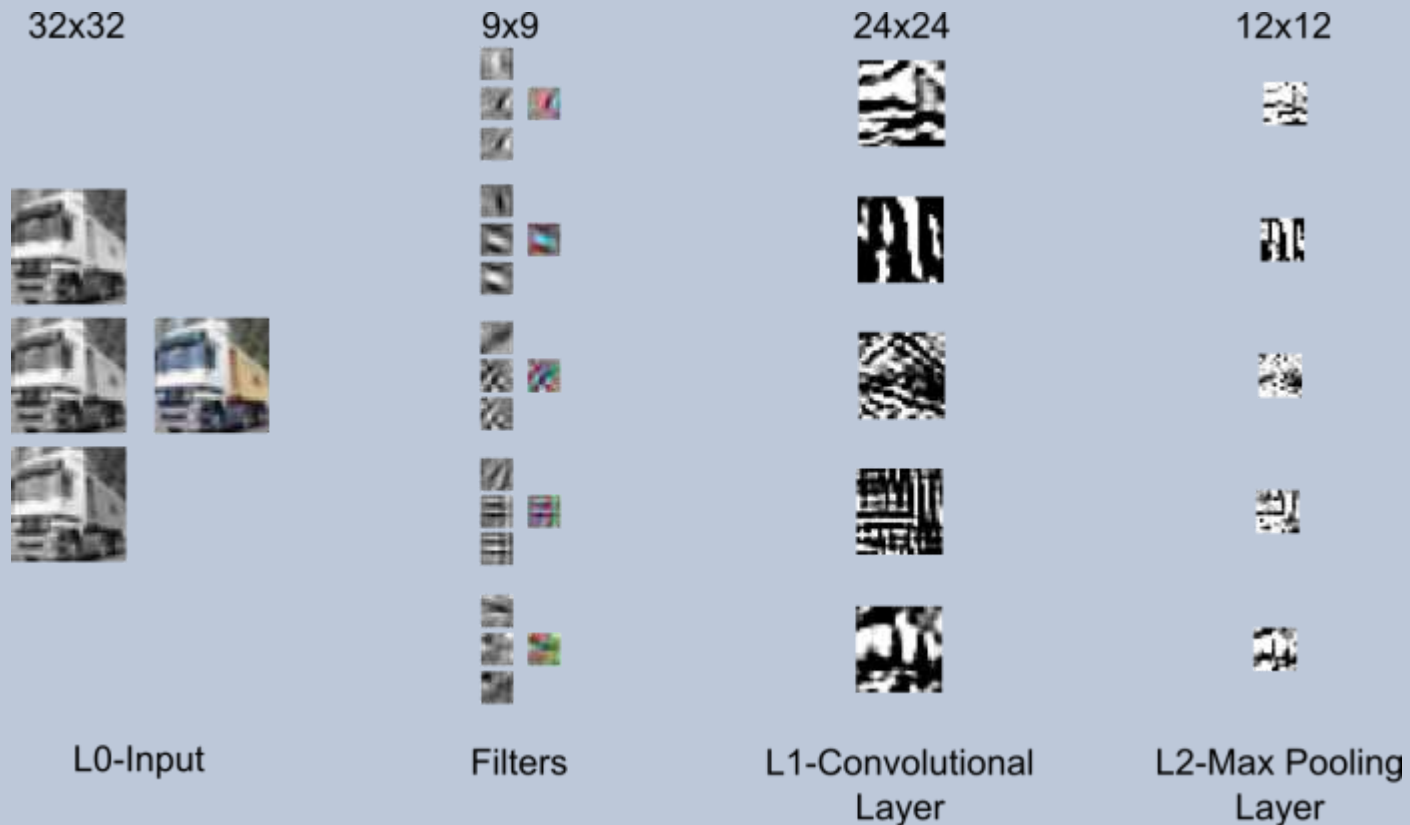
# Max-pooling layer

- Introduces small translation invariance
- Improves generalization



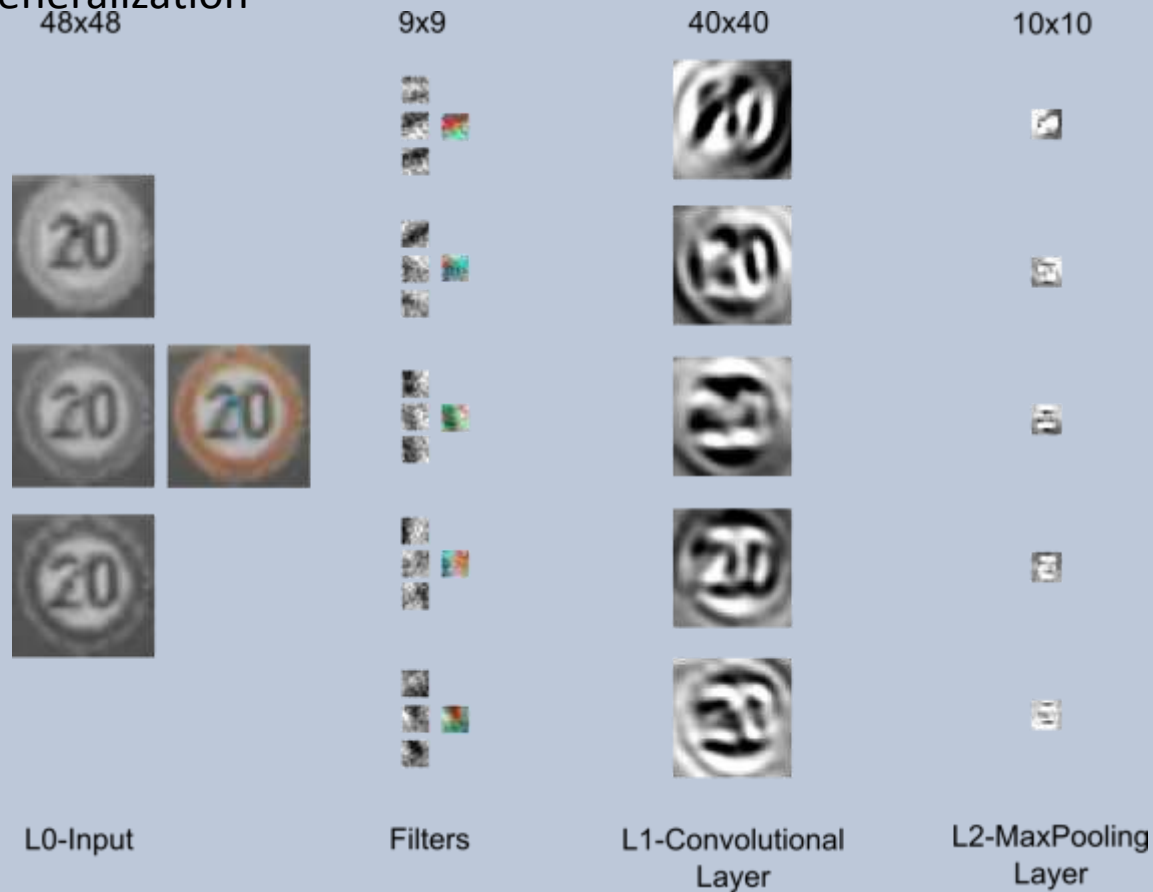
# Max-pooling layer

- Introduces small translation invariance
- Improves generalization



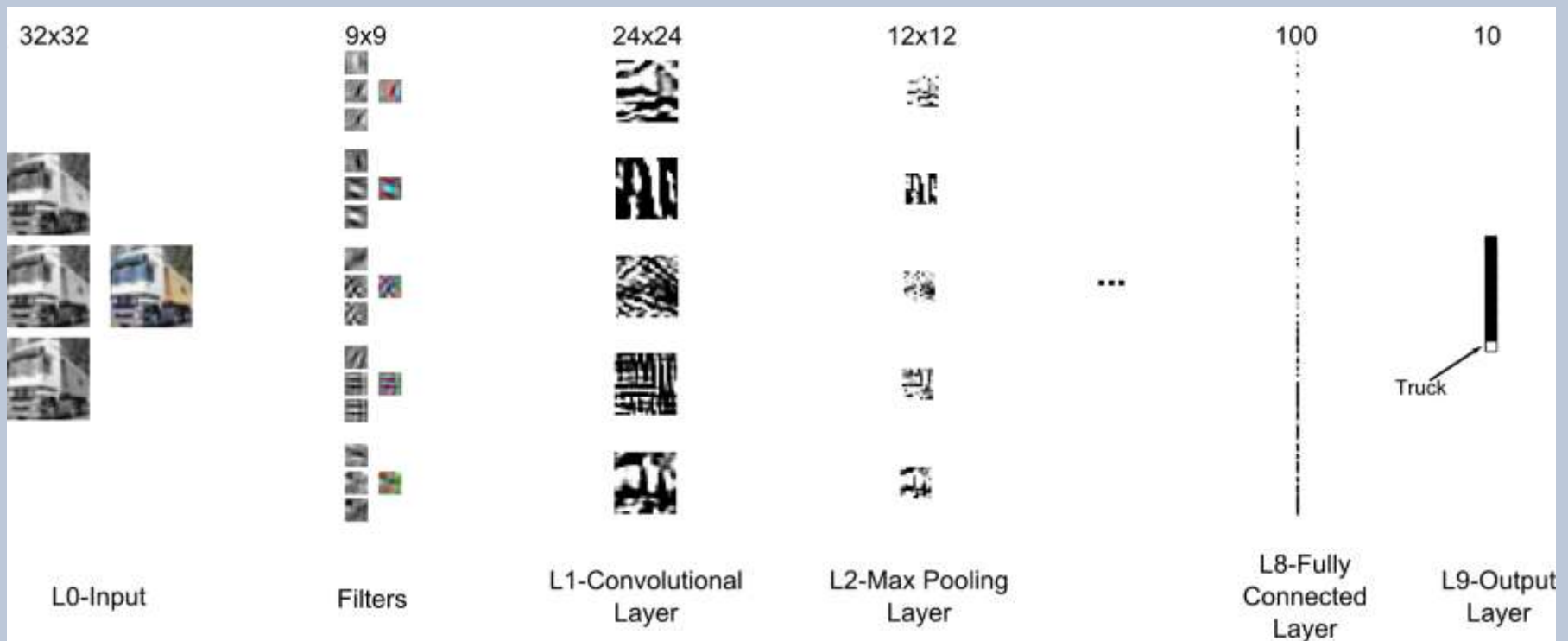
# Max-pooling layer

- Introduces small translation invariance
- Improves generalization



# Fully connected layer

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

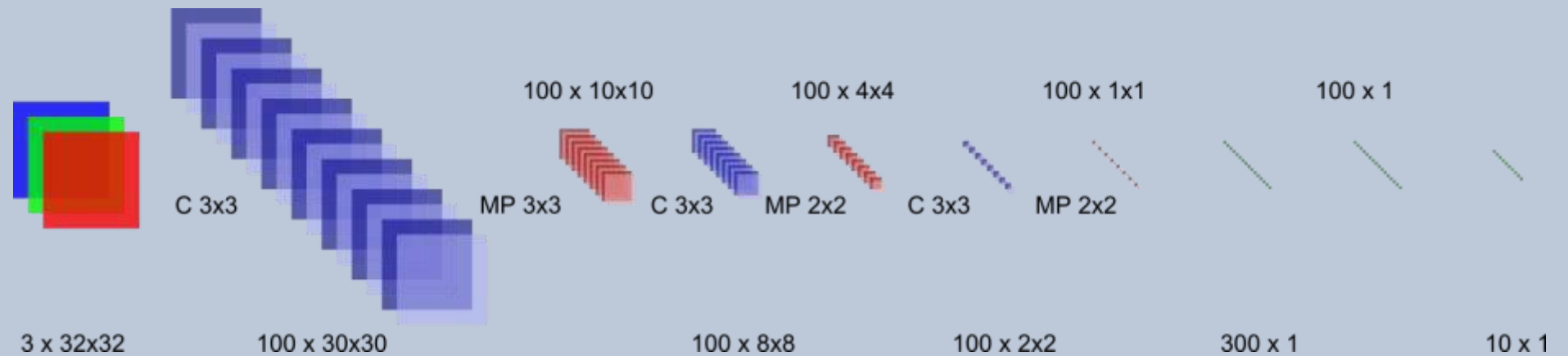


# Fully connected layer

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



# Training is computational intensive



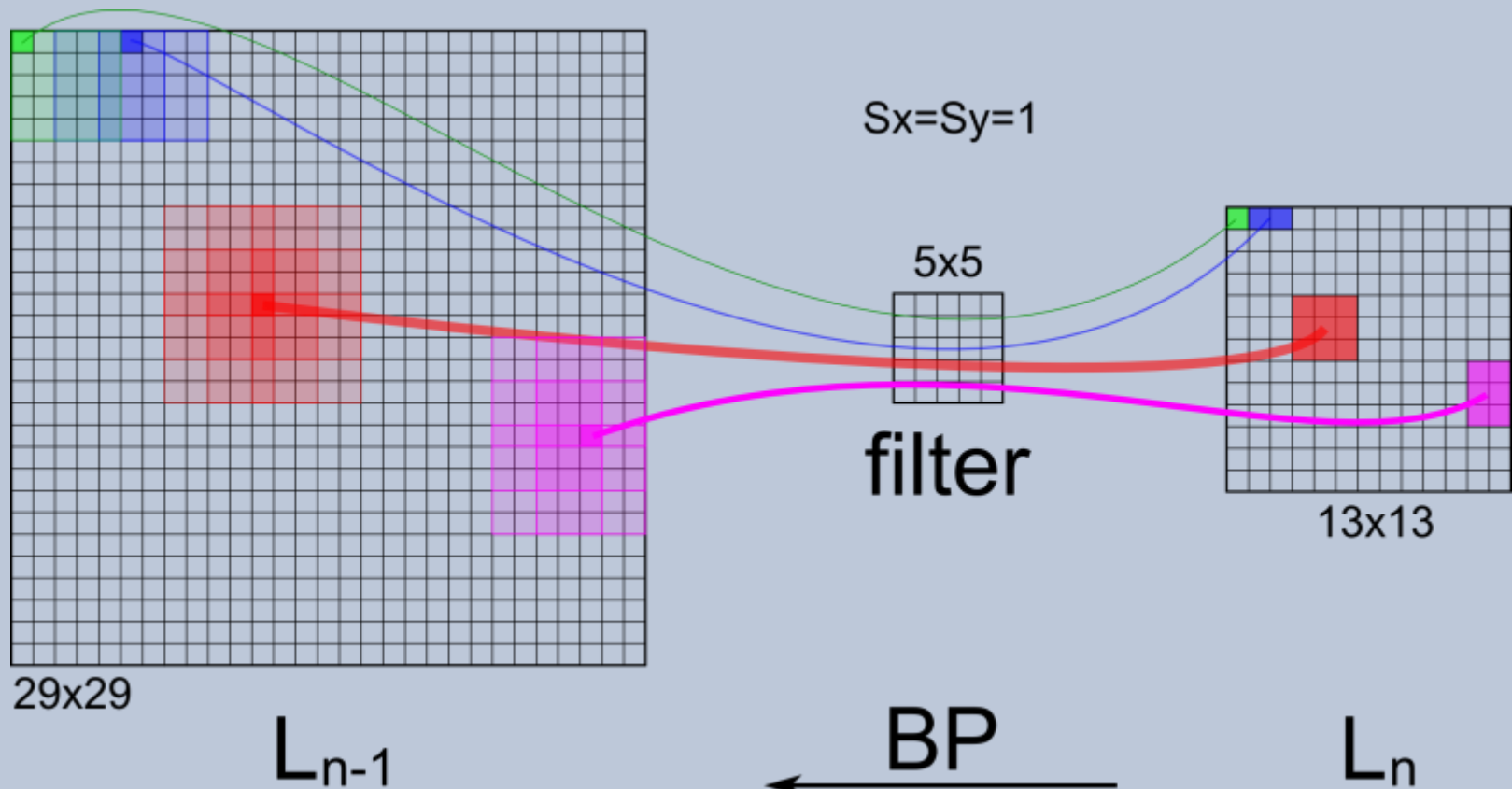
# Graphics processing units (GPUs)

- 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)
- ~3000\$ for one workstation with 4 GPUs



# Back-propagation of errors

- Uses pooling of errors (deltas).



# Experiments

- Distorting images
- Datasets:
  - Handwritten digits: MNIST
  - Handwritten Latin characters: NIST SD 19
  - Handwritten Chinese characters
  - 3D models: NORB
  - Natural images: CIFAR10
  - Traffic signs



# Distortions

- MNIST
  - affine: translation, rotation, scaling
  - elastic

5	0	4	1	9	2	1	3	1	4	3	5	3	6	1	7	2	8	6	9	Original
5	0	4	1	9	2	1	3	1	4	3	5	3	6	1	7	2	8	6	9	Epoch 0
5	0	4	1	9	2	1	3	1	4	3	5	3	6	1	7	2	8	6	9	Epoch 1
5	0	4	1	9	2	1	3	1	4	3	5	3	6	1	7	2	8	6	9	Epoch 2
5	0	4	1	9	2	1	3	1	4	3	5	3	6	1	7	2	8	6	9	Epoch 3

- Elastic only for characters and digits
- Greatly improves generalization and recognition rate
- CIFAR10/traffic signs have non-uniform backgrounds => border effects



# MNIST

- Very competitive dataset
- Handwritten digits
- 28x28 grayscale images
- 60000 for training and 10000 for testing
- Tens of papers: <http://yann.lecun.com/exdb/mnist/>



# MNIST

- Simard et al. (2003) – 0.40%, Ciresan et al. (2010) – 0.35% (big deep MLP)
- Big deep CNN – 0.35% (2011), far less weights than the MLP
- 30 out of 35 digits have a correct second prediction

#M, #N in Hidden Layers	Test error [%]
20M-60M	1.02
20M-60M-150N	0.55
20M-60M-100M-150N	0.38
<b>20M-40M-60M-80M-100M-120M-150N</b>	<b>0.35</b>

9 8 9	5 3 5	4 4 6	2 1 7	4 6 4	9 5 9	2 3 2	2 3 5	4 4 9	5 6 5	9 9 4	9 4 9
4 4 9	0 8 2	1 2 1	4 9 4	0 0 6	4 8 4	7 2 7	1 7 1	3 2 9	3 5 3	4 4 9	4 4 9
2 2 7	5 3 5	7 1 4	1 6 1	1 6 1	4 9 4	2 2 7	8 5 8	1 4 1	5 6 5	0 6 9	

Label

First prediction

Second prediction

35 errors



# NIST SD 19

- More than 800000 handwritten digits and letters
- our CNN have state of the art results on all tasks

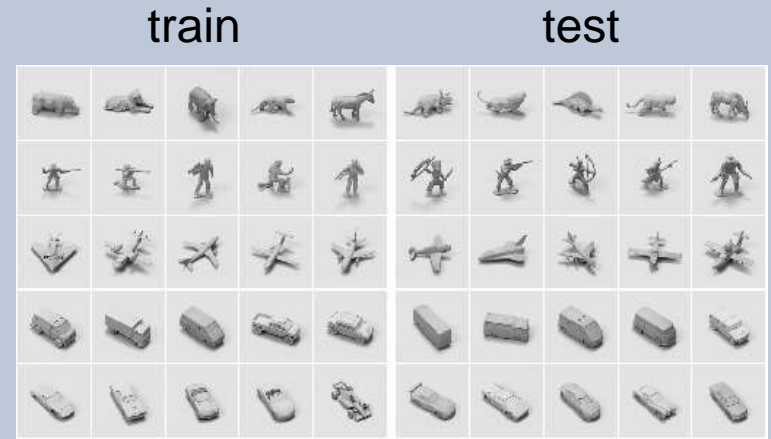


Task	Test error [%]
Small letters	7.71
Big letters	1.91
Case insensitive	7.58
Merged letters (37 classes)	8.21

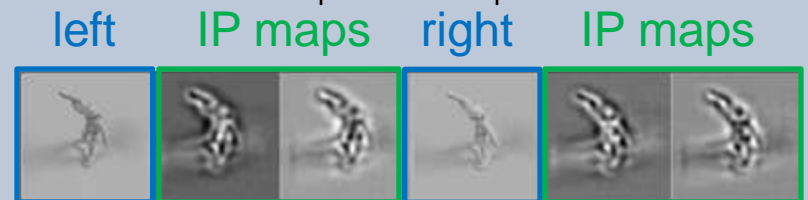


# Small NORB

- 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%
- 40% of the errors are cars erroneously classified as trucks



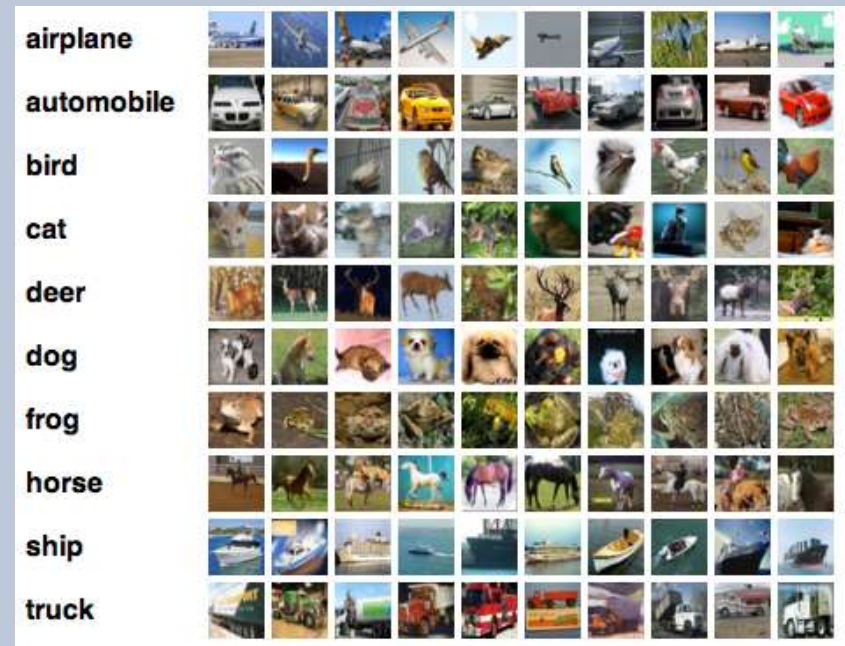
translation [%]	IP maps	test [%]
0	no	7.86 ± 0.55
5	no	4.71 ± 0.57
0	yes	3.94 ± 0.48
<b>5</b>	<b>yes</b>	<b>2.53 ± 0.40</b>



# CIFAR10

- small, 32x32 pixels color images
- complex backgrounds
- 10 classes
- 50000 training images
- 10000 test images

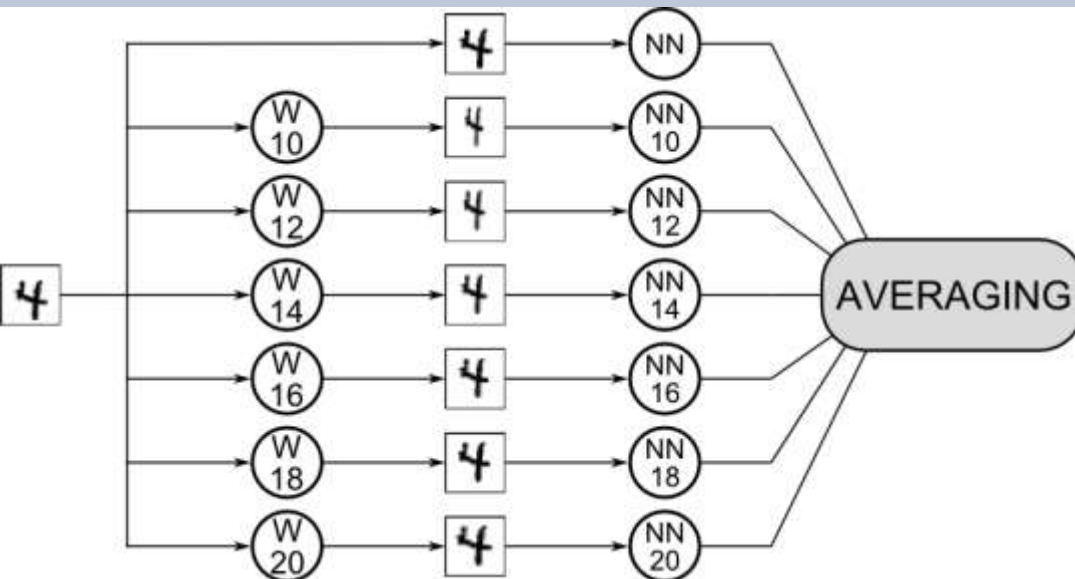
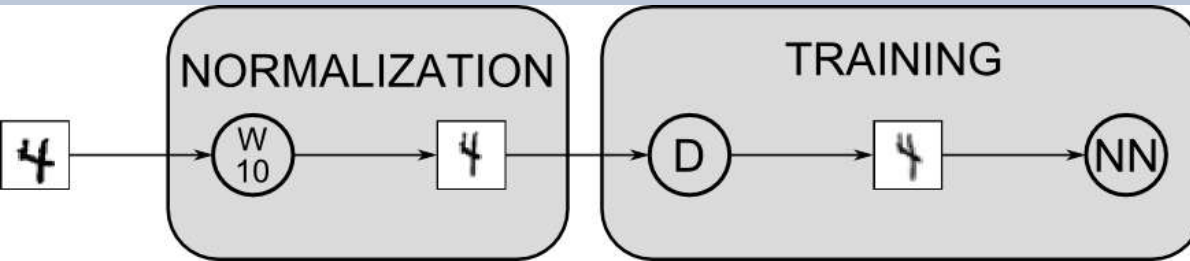
trans. [%]	IP maps	TfbV [%]
0; 100M	no	28.87 ± 0.37
0; 100M	edge	29.11 ± 0.36
5; 100M	no	20.26 ± 0.21
5; 100M	edge	21.87 ± 0.57
5; 100M	hat	21.44 ± 0.44
5; 200M	no	19.90 ± 0.16
<b>5; 300M</b>	<b>no</b>	<b>19.51 ± 0.18</b>
5; 400M	no	19.54 ± 0.16



first layer filters



# Committees



- Extremely simple idea
- Easy to compute
- Averaging the corresponding outputs of many nets
- Decrease the error with 20-80%
- Work better with preprocessing in case of handwritten characters



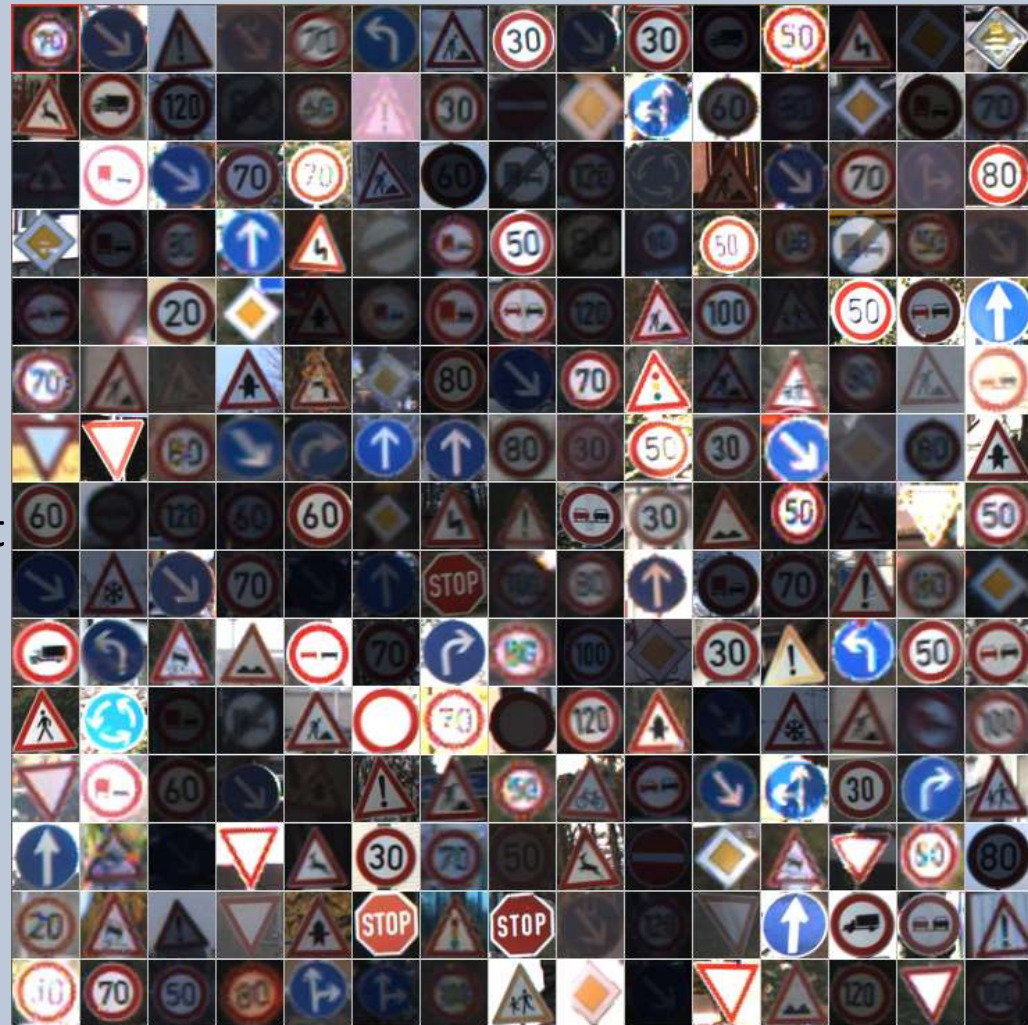
# Chinese Handwriting Recognition Competition ICDAR 2011

- Offline Chinese Character Recognition (Task 1)
- 3GB of data
- 3755 classes
- >1M characters
- 48x48 pixels grayscale images
- 270 samples / class
- 9 teams
- No knowledge of Chinese
- First place at ICDAR 2011 competition
  - 92.18% first prediction is correct
  - 99.29% CR10

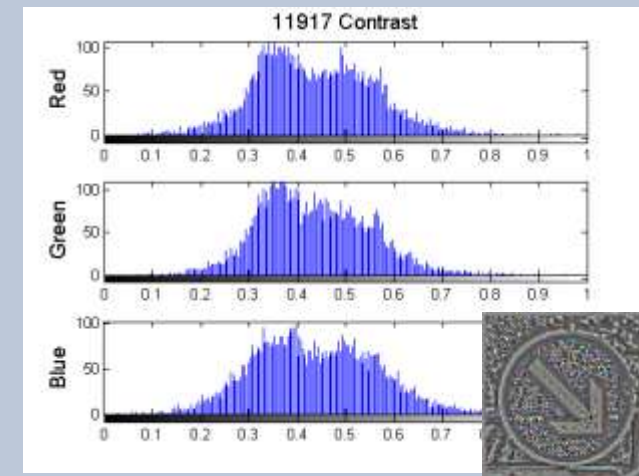
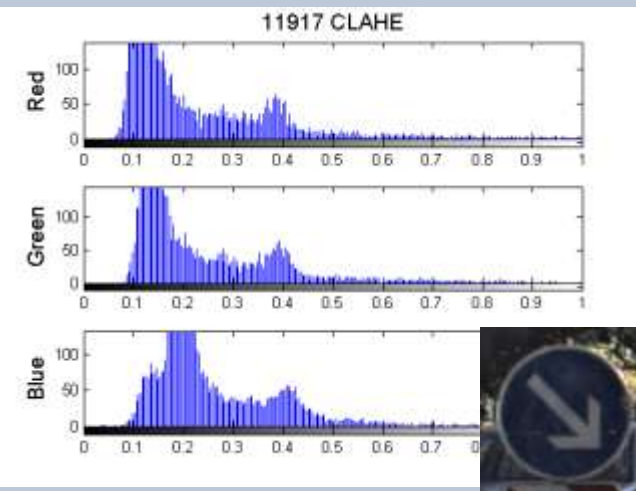
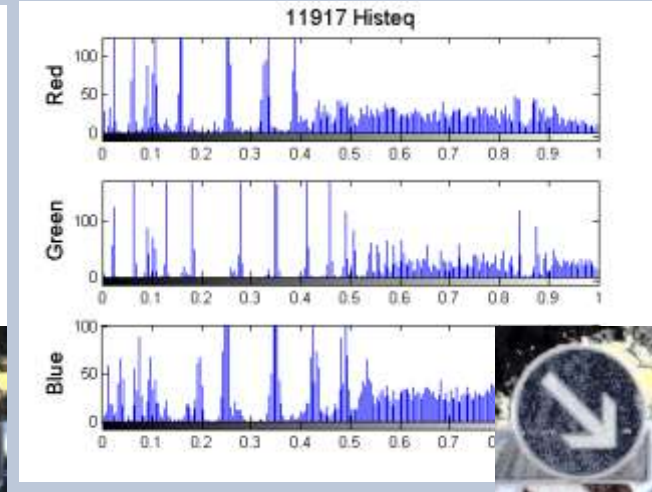
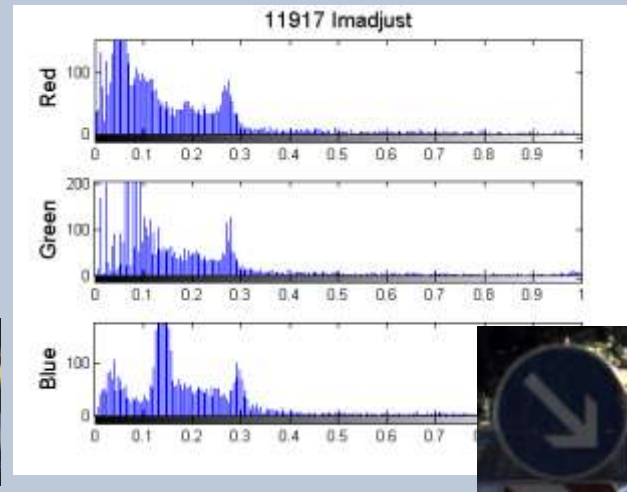
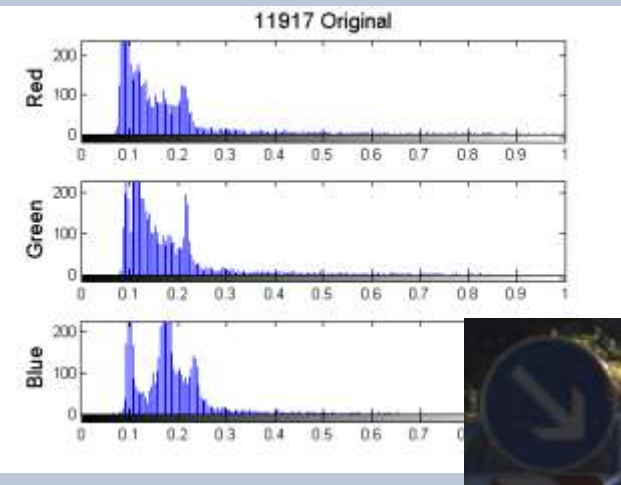


# German Traffic Sign Recognition Competition

- Initial phase:
  - More than 10 teams
  - We won 1<sup>st</sup> place with 98.98%
- 40000 color images 15x15 to 250x250
- 43 different signs: 26640 training images, 12569 testing images Input: Extract Region of Interest (ROI)
- Enhance contrast with four different methods
- Resize up/down to 48x48 pixels
- Great variation in details, contrast and illumination
- <http://benchmark.ini.rub.de/>
- Final phase: last Wednesday at IJCNN 2011



# New normalizations



# Normalized images

- original
- imadjust
- histeq
- clahe
- contrast



# Distortions

- At the beginning of every epoch, each image in the training set is distorted
- The original training set is used only for validation
- Type:
  - Translation (random for both axes, maximum 10%)
  - Rotation (random, maximum 10 degrees)
  - Scaling (random for both axes, maximum 10%)
- Improves generalization and recognition rate on test set
- Non-uniform backgrounds => border effects



# CNN trained on preprocessed color images

- Original + 4 preprocessed datasets
- Train 5 nets for each
- Build a 25-net committee
- First place with 99.43% recognition rate
- The only team with better than human (98.84%) results



first layer filters

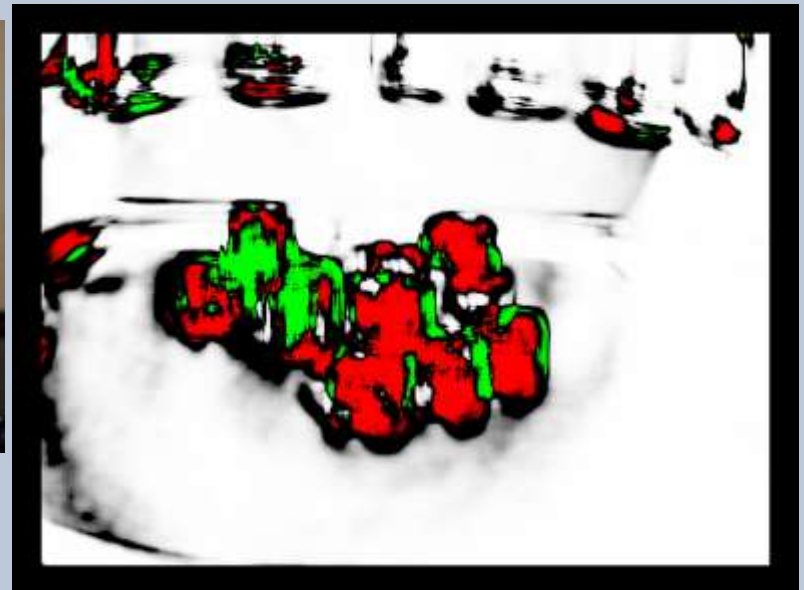
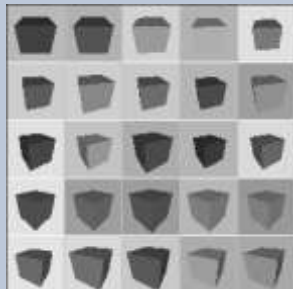
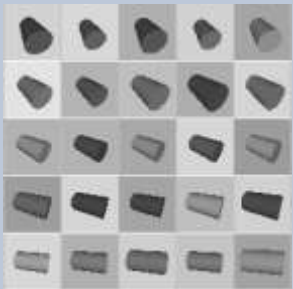
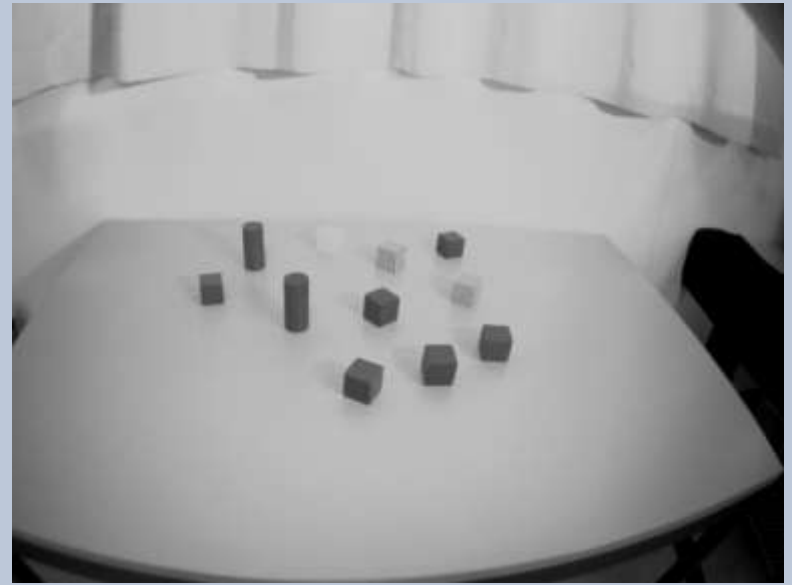
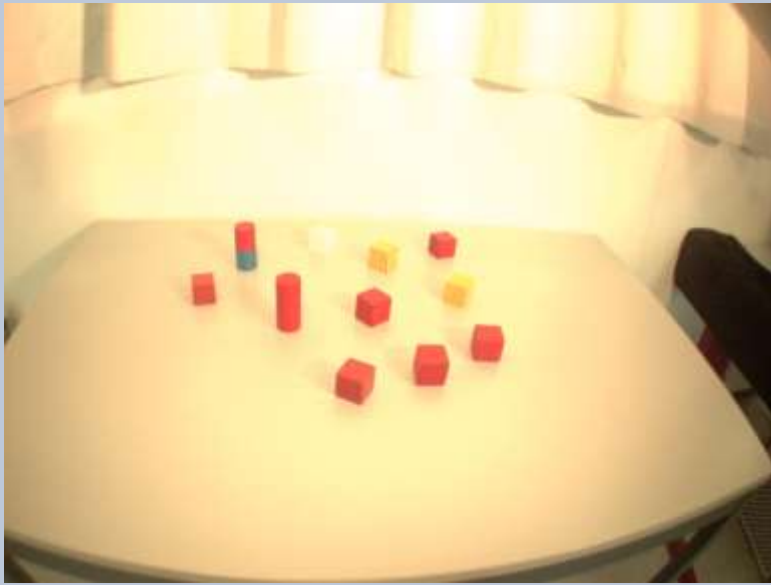


# Summary

Dataset	Best result by others (error [%])	Our result (error [%])	Decrease [%]
MNIST	0.39	0.23	41%
NIST SD 19	all tasks	all tasks	20-80%
Chinese	10.01	6.5	35%
NORB	2.87	2.01	30%
CIFAR10	18.81	11.75	37%
Traffic signs	1.69	0.54	72%

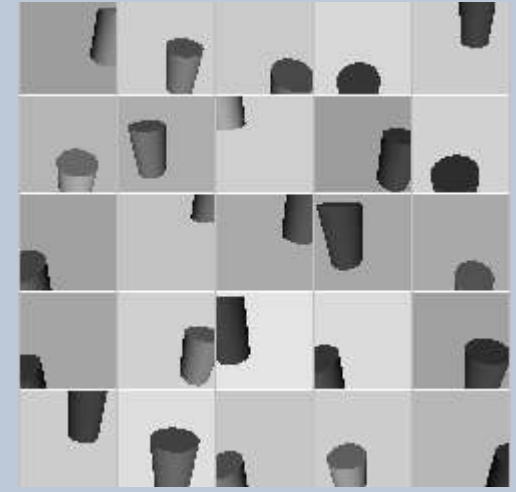
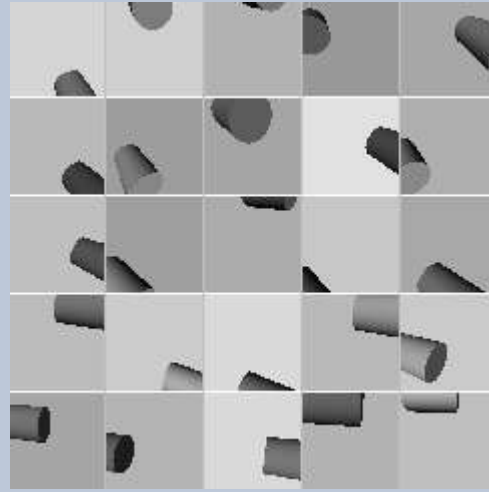
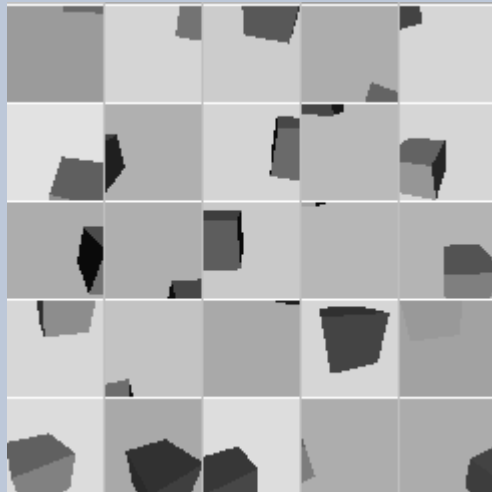


# Image segmentation

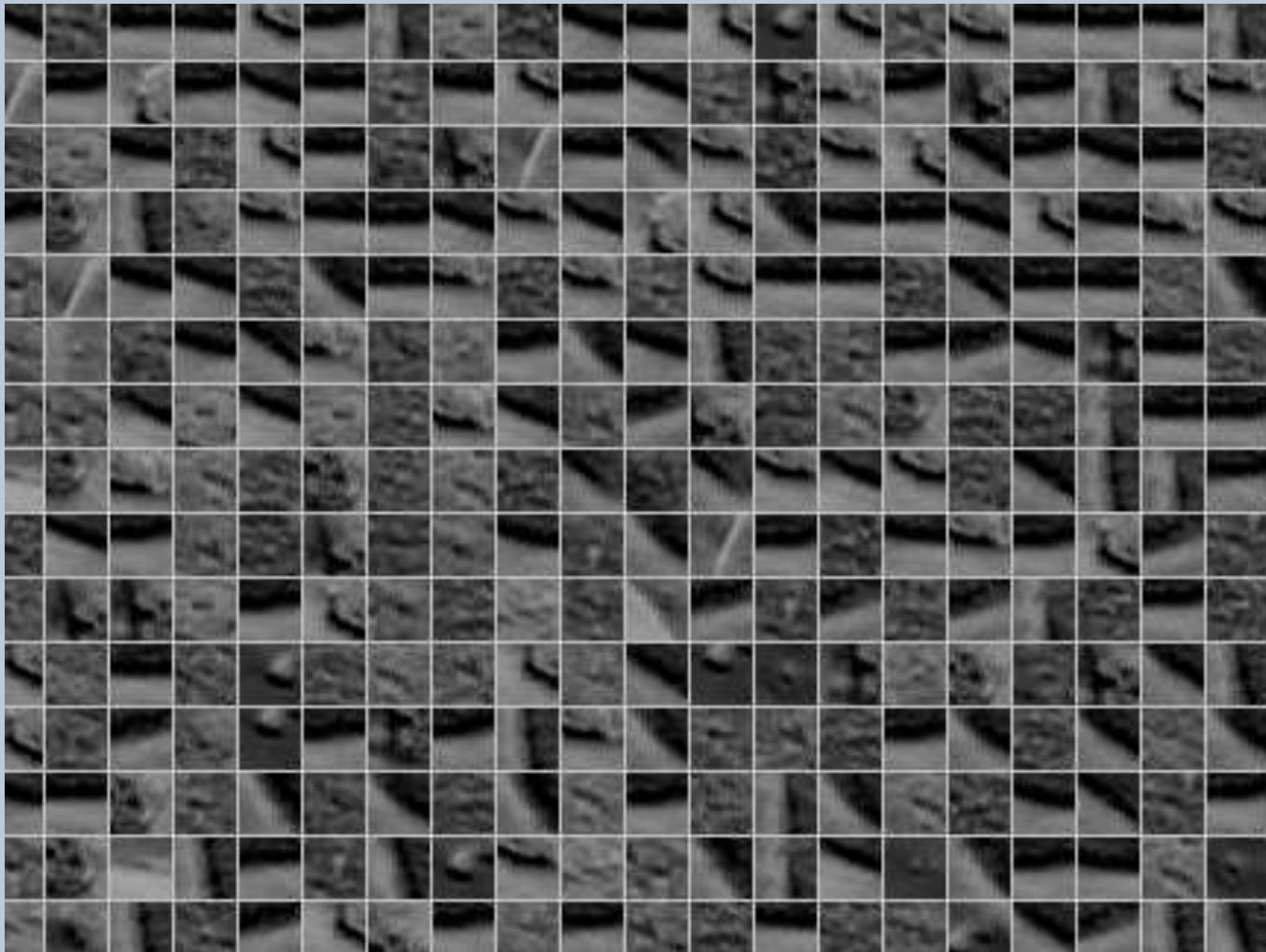


Red – cubes  
Green – cylinders  
White – background

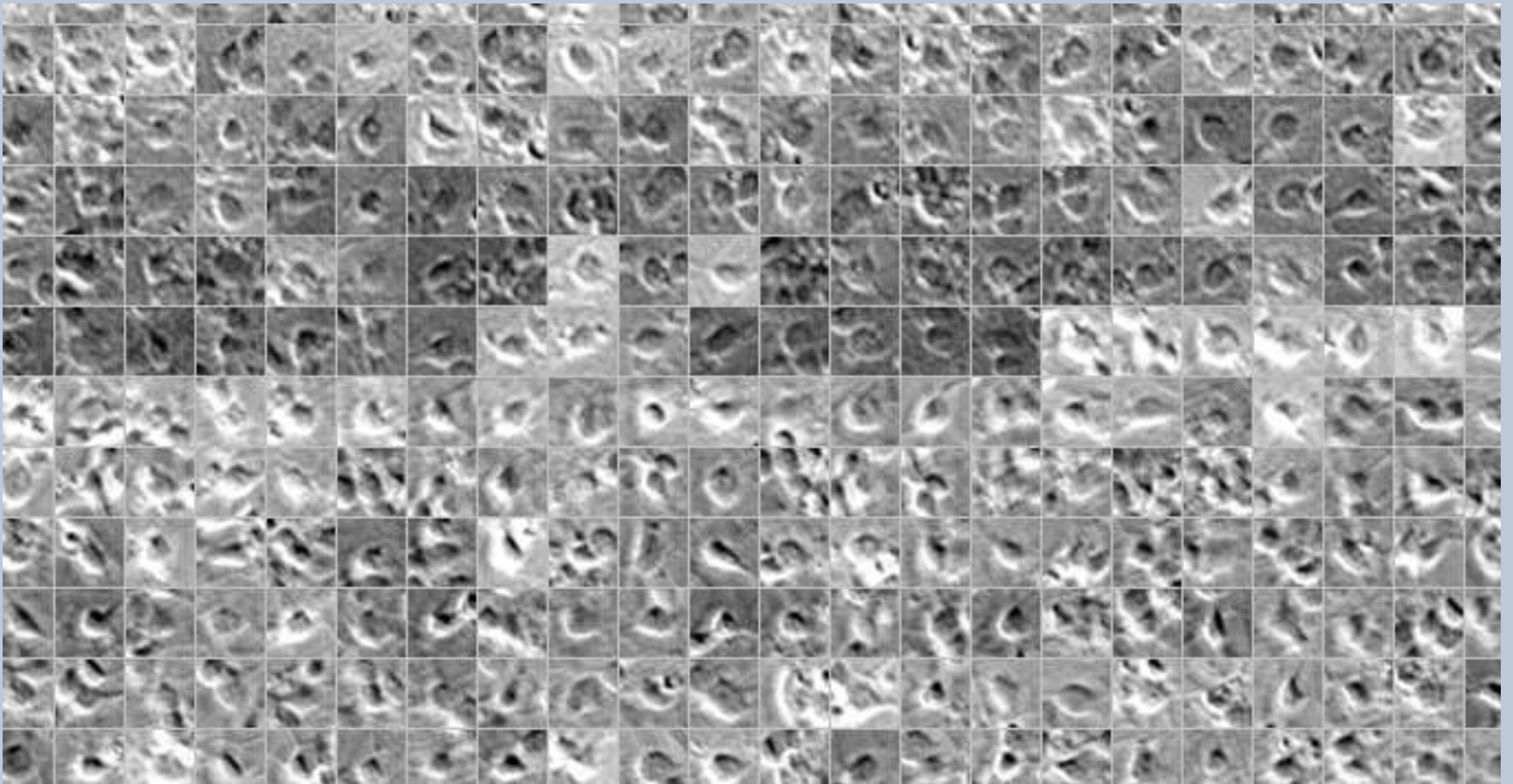
# Searching for objects – adding a negative class with partial cubes and cylinders



# Detecting cell contours



# Detecting cells



# Conclusions

- 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 application.



# What is next?

- Test the CNNs on different datasets:
  - 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 human image classification performance on more datasets.
- Add unsupervised pre-training.
- More at [www.idsia.ch/~ciresan](http://www.idsia.ch/~ciresan)



# Publications

## Conference papers:

- Flexible, High Performance Convolutional Neural Networks for Image Classification (D. Ciresan, U. Meier, J. Masci, L. M. Gambardella, J. Schmidhuber, IJCAI 2011)
- A Committee of Neural Networks for Traffic Sign Classification (D. Ciresan, U. Meier, J. Masci, J. Schmidhuber, IJCNN 2011)
- Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction (J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, ICANN 2011)
- On Fast Deep Nets for AGI Vision (J. Schmidhuber, D. Ciresan, U. Meier, J. Masci, Alex Graves, AGI 2011)
- Convolutional Neural Network Committees For Handwritten Character Classification (D. Ciresan, U. Meier, L. M. Gambardella, J. Schmidhuber, ICDAR 2011)
- Better digit recognition with a committee of simple Neural Nets (U. Meier, D. Ciresan, L. M. Gambardella, J. Schmidhuber, ICDAR 2011)

## Journal papers:

- Deep, Big, Simple Neural Nets for Handwritten Digit Recognition (D. Ciresan, U. Meier, L. M. Gambardella, J. Schmidhuber, Neural Computation, December 2010)

## Under review:

- Narrowing the Gap to Human Image Classification Performance, (D. Ciresan, U. Meier, J. Schmidhuber, NIPS 2011)
- Democratic Committee of Fast Deep MLPs Improves MNIST Recognition Rate (D. Ciresan, U. Meier, L. M. Gambardella, J. Schmidhuber, Neural Computation)

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

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

	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

