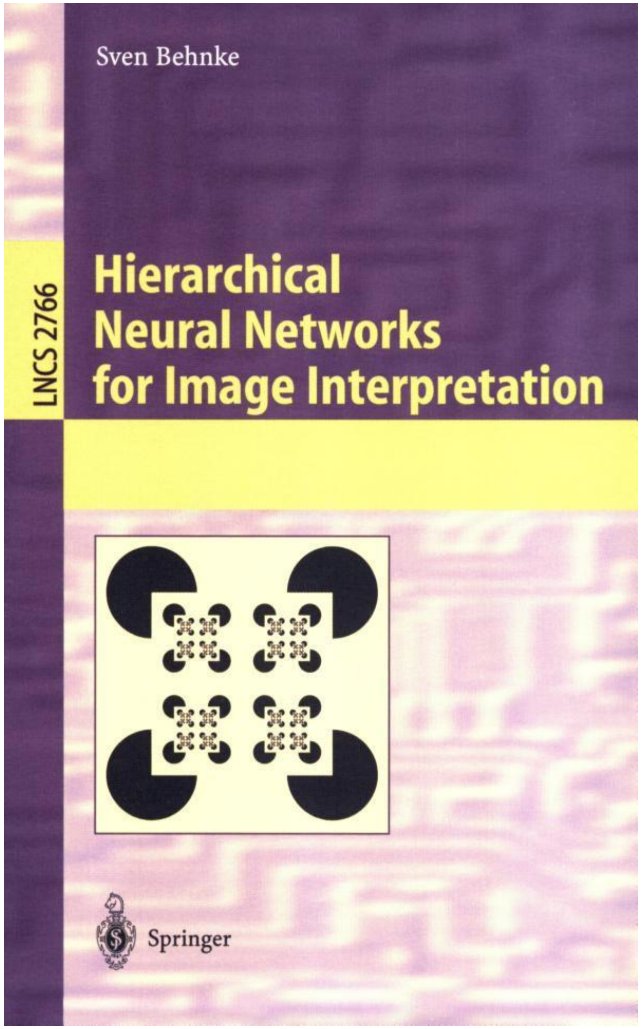
Deep Learning of Semantic Perception for Robots

Sven Behnke

University of Bonn
Computer Science Institute VI
Autonomous Intelligent Systems

Sven Behnke

- Investigates neural networks since 1991
- Programmed early neural hardware (Siemens Synapse, Adaptive Solutions CNAPS)
- Diploma thesis 1997 with Siemens: Recognition of handwritten ZIP codes
- Deep learning research since 1997
- PhD 2002, FU Berlin: proposed Neural Abstaction Pyramid



Communication Robot

[Nieuwenhuisen and Behnke, SORO 2013]

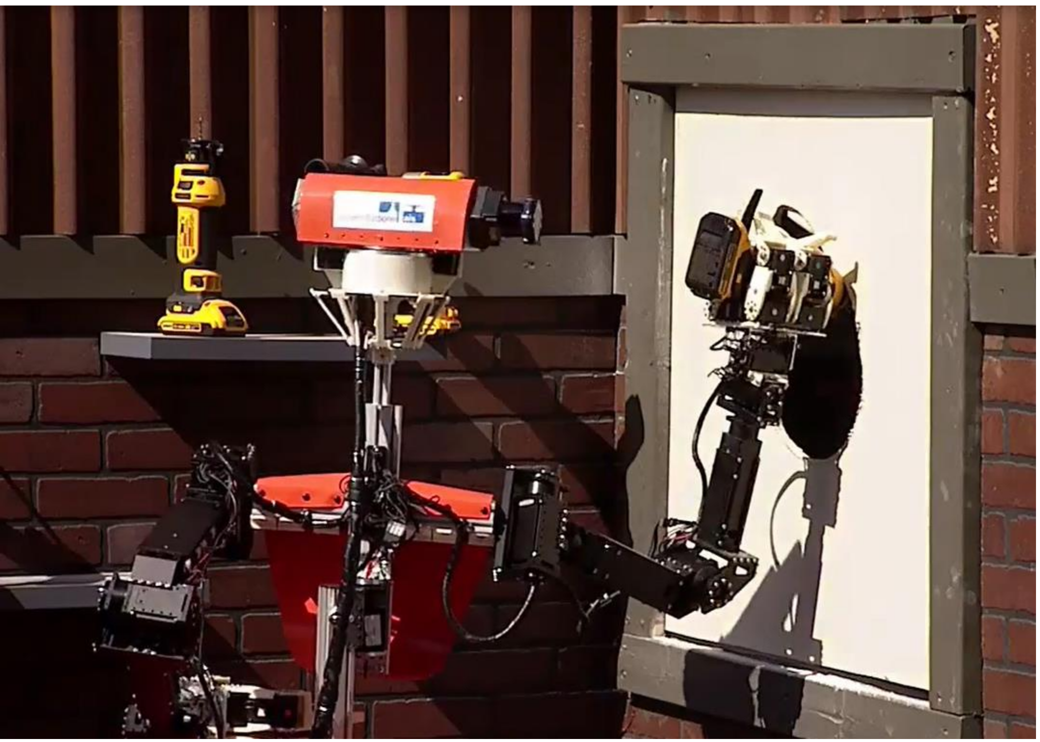
Domestic Service Robots

Dynamaid

Cosero

[Stückler et al.: Frontiers in AI and Robotics 2016]

Search and Rescue, Space Exploration Robots

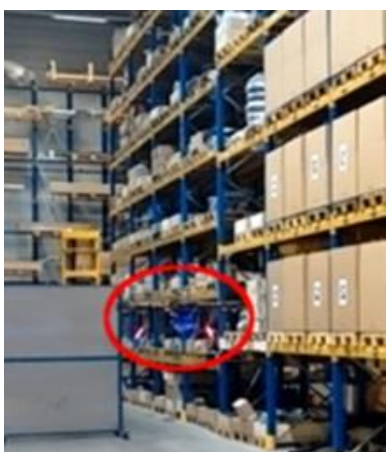


[Schwarz et al.: Frontiers in Robotics and AI 2016, JFR 2017]

Soccer Robots

[Allgeuer et al.: Humanoids 2015, 2016]

Micro Aerial Vehicles

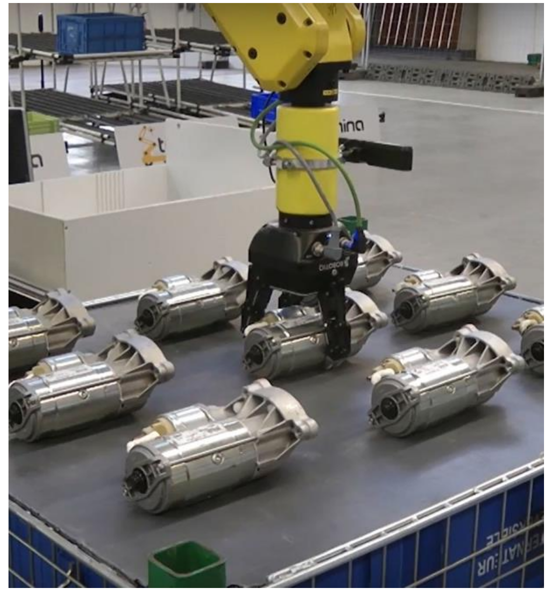


[Nieuwenhuisen et al.: JINT 2015, Droeschel et al: JFR 2016]

Bin Picking Robots

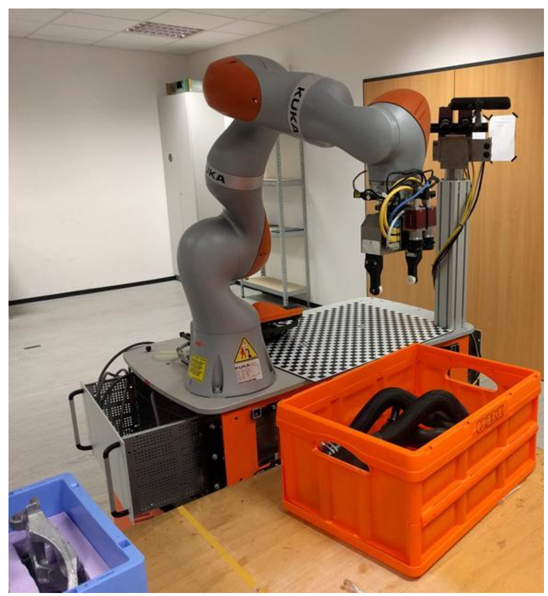
ActReMa

STAMINA



Amazon Picking Challenge

EuRoC C1



EuRoC C2

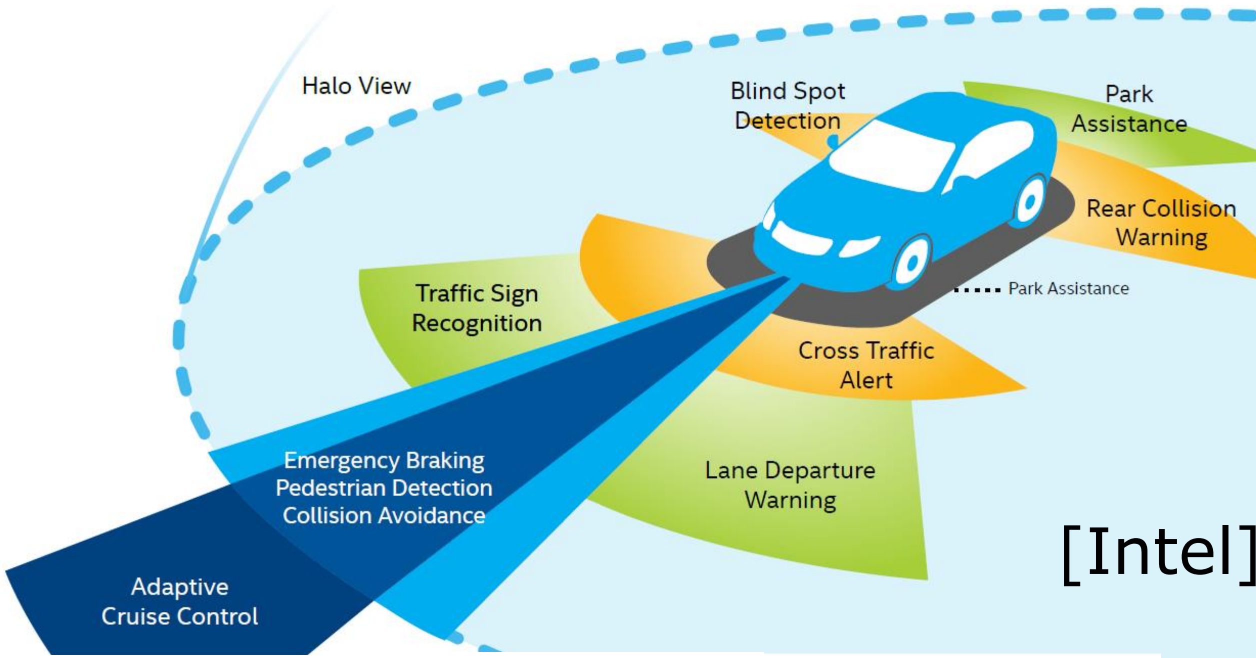
Self-driving Car

Team Berlin at DARPA Urban Challenge

Sensors for Autonomy

Radar

Laser scanner



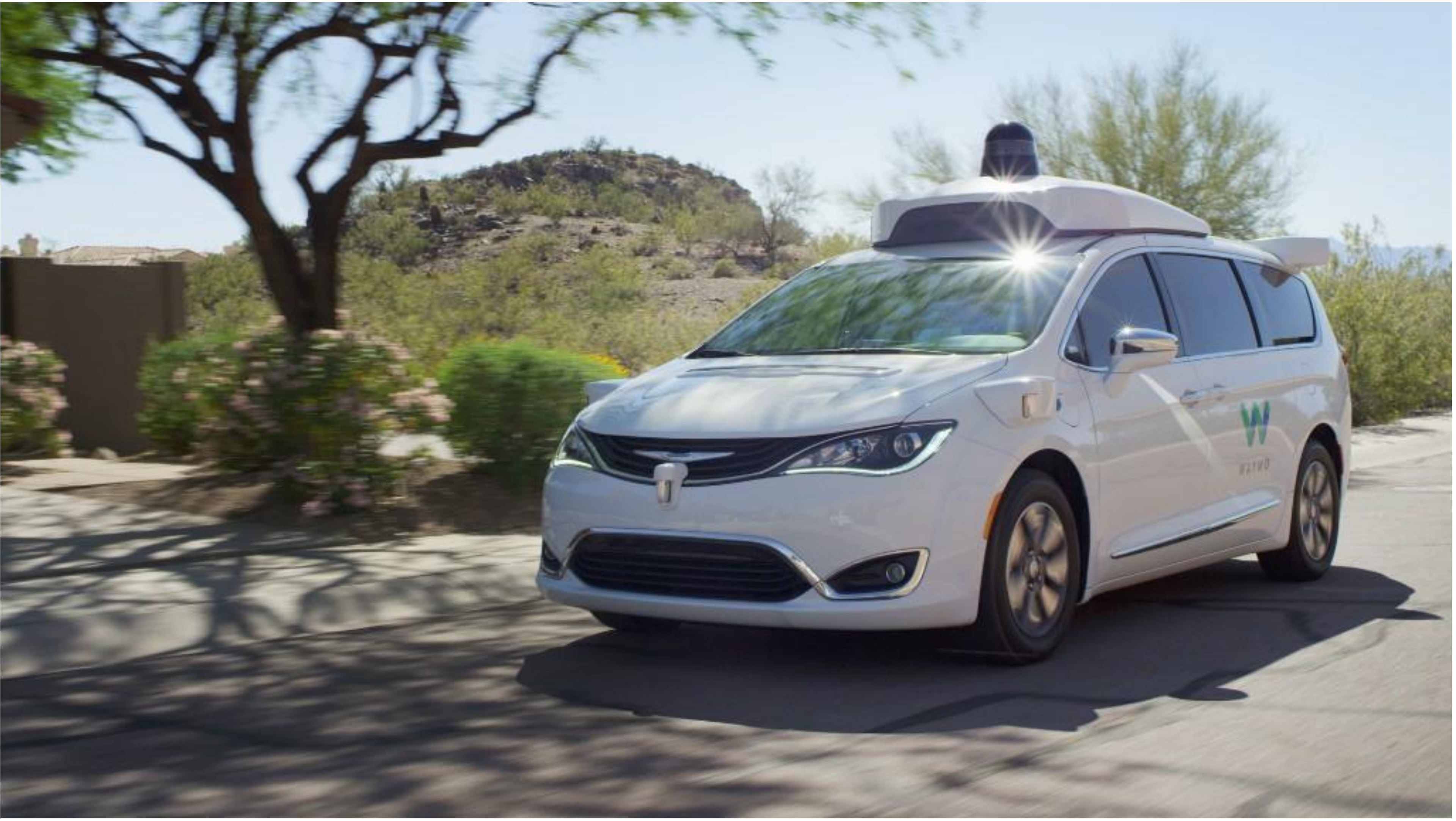
[Intel]

Cameras

Short dist. radar

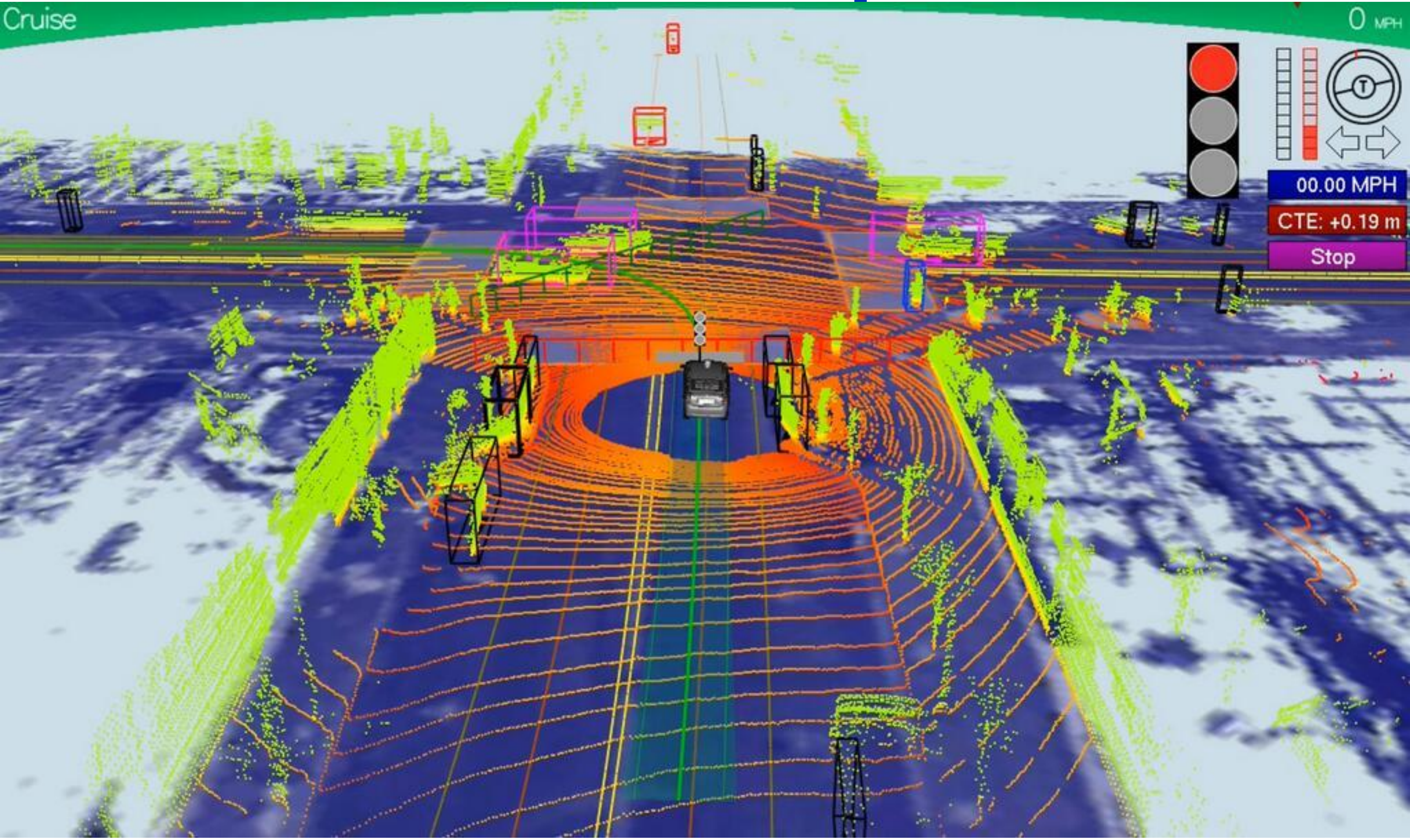
Ultrassonic

Google Self-driving Car



[Waymo]

Environment Perception [Google]



An Image Says More than a Thousand Words

[Vinyals et al. 2014]

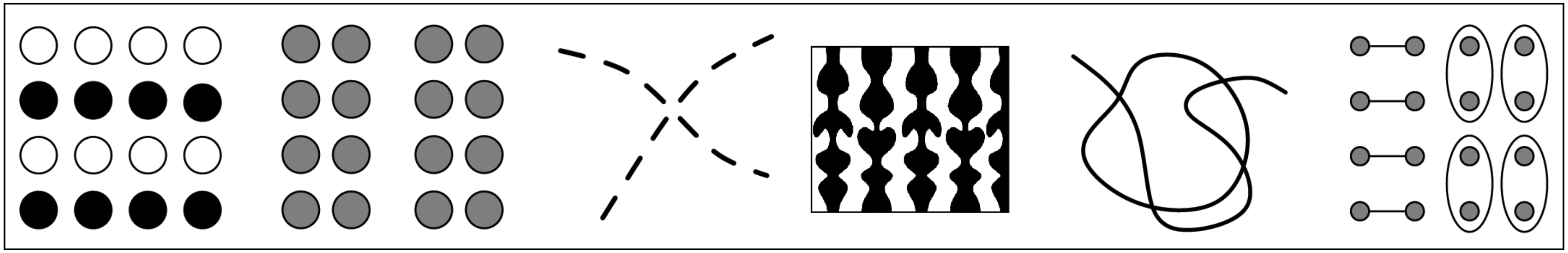
Motivation from Visual Perception

- Visual perception important for humans and computers
- Image interpretation is non-trivial
  - Occlusions
  - 3D reconstruction
  - Ambiguities
- Impressive performance of the human visual system
  - Fast
  - Robust

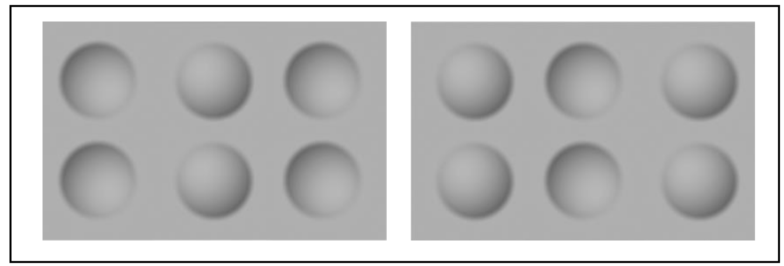
Performance of the Human Visual System

Psychophysics

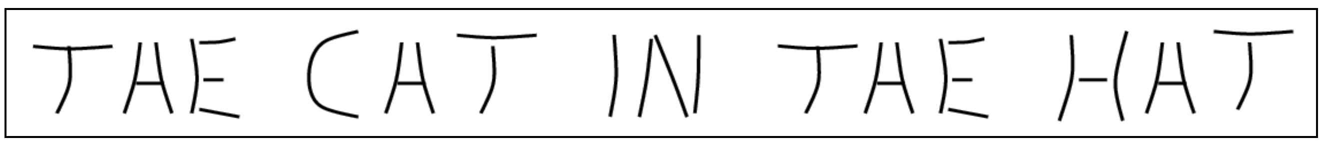
- Gestalt principles



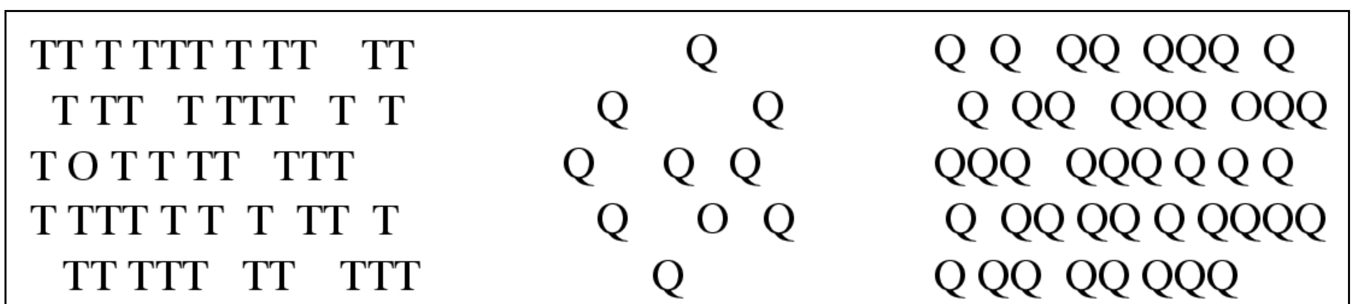
- Heuristics



- Context

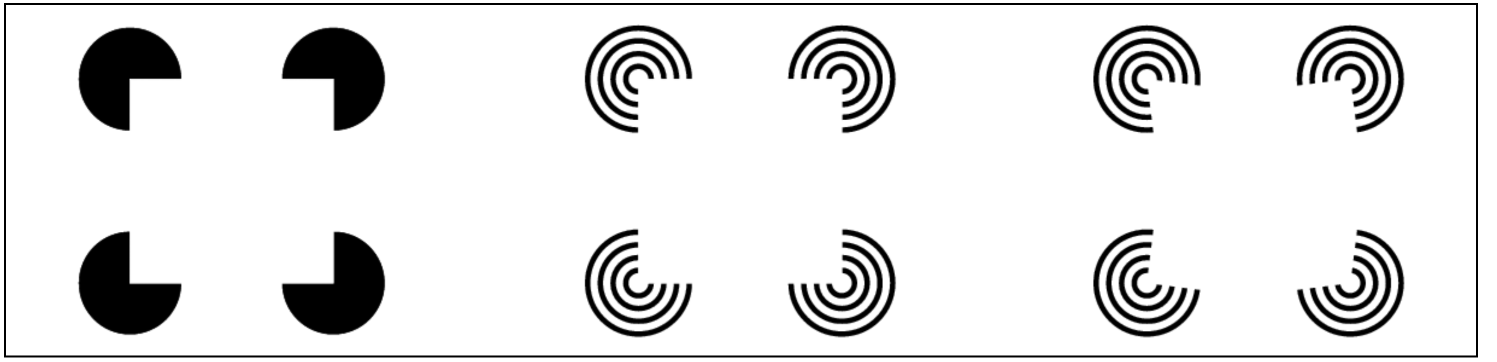


- Attention

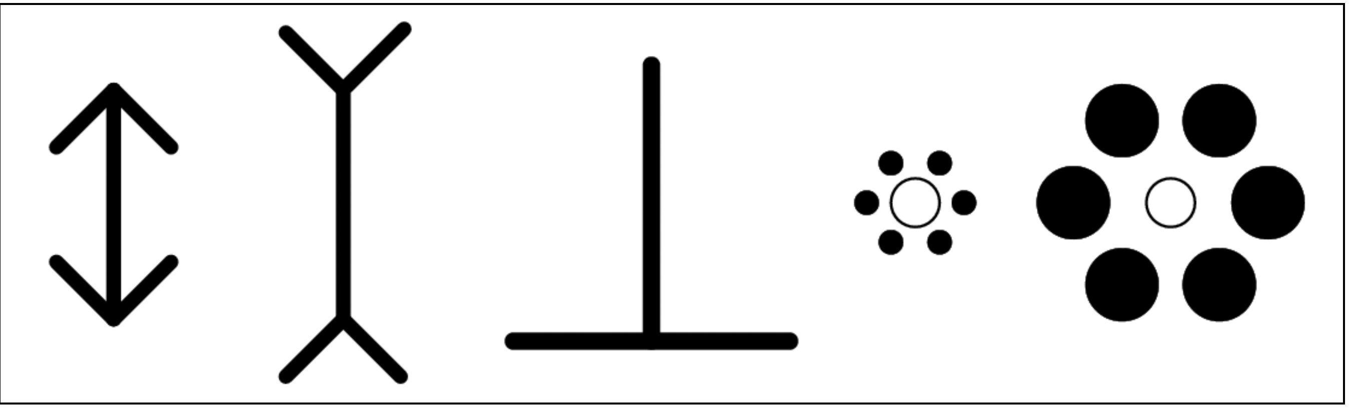


Visual Illusions

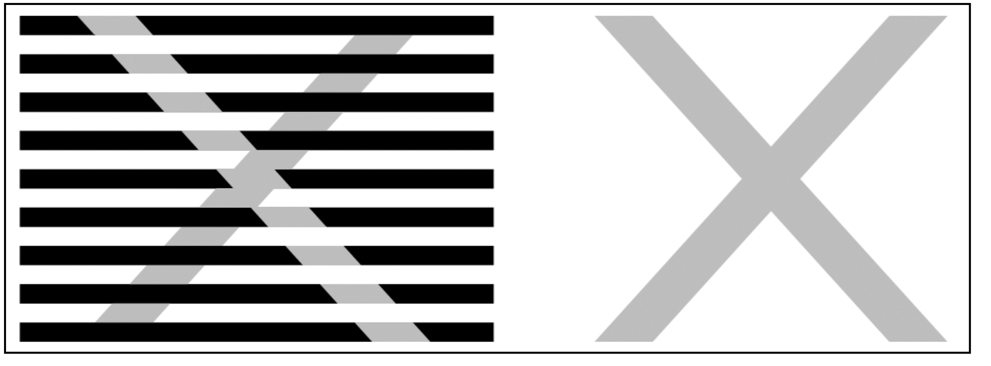
Kanizsa Figures



Müller-Lyer horizontal/ vertical Ebbinghaus-Titchener

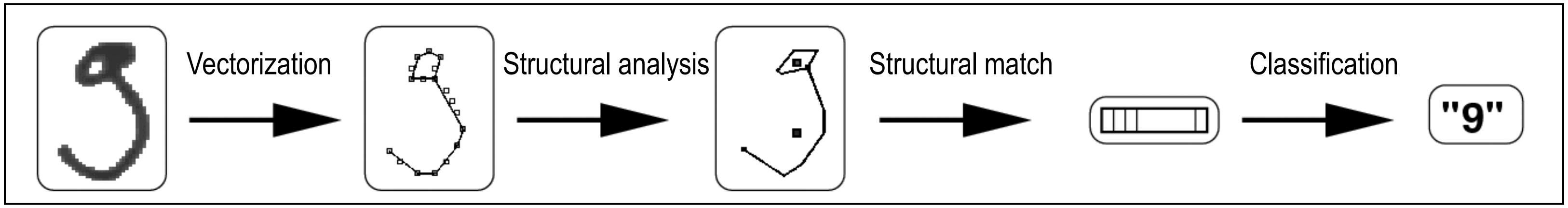


Munker-White

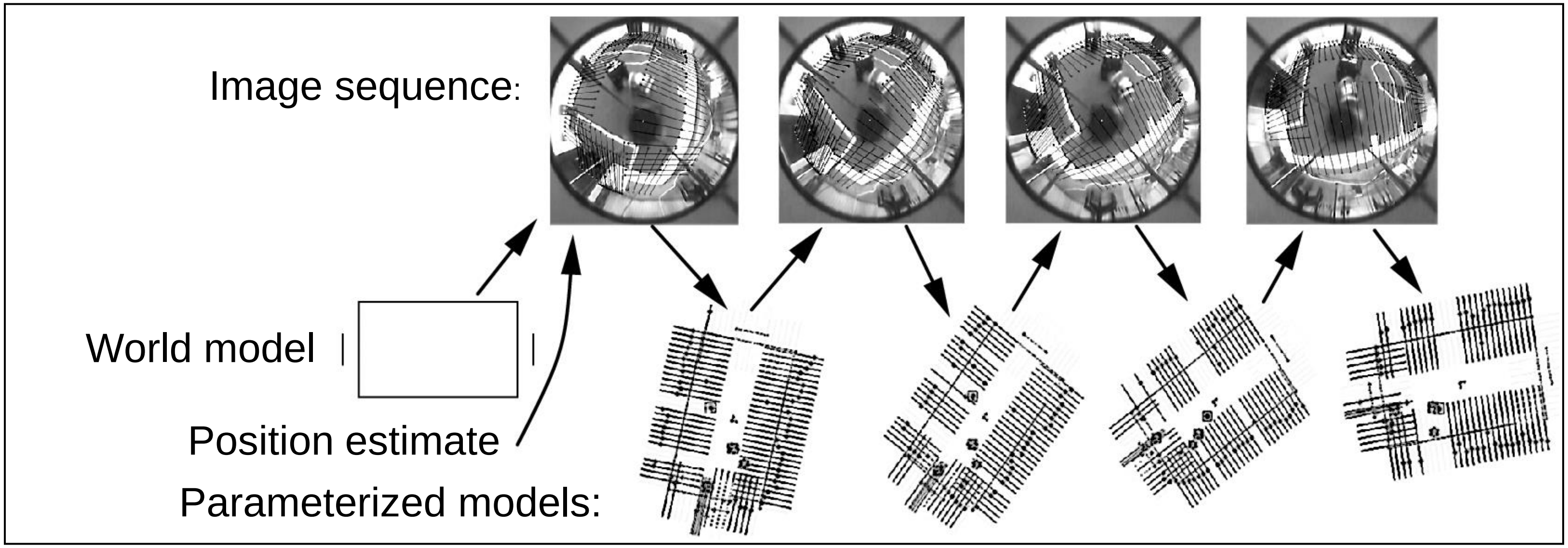


Computer Vision

- Data driven



- Model driven



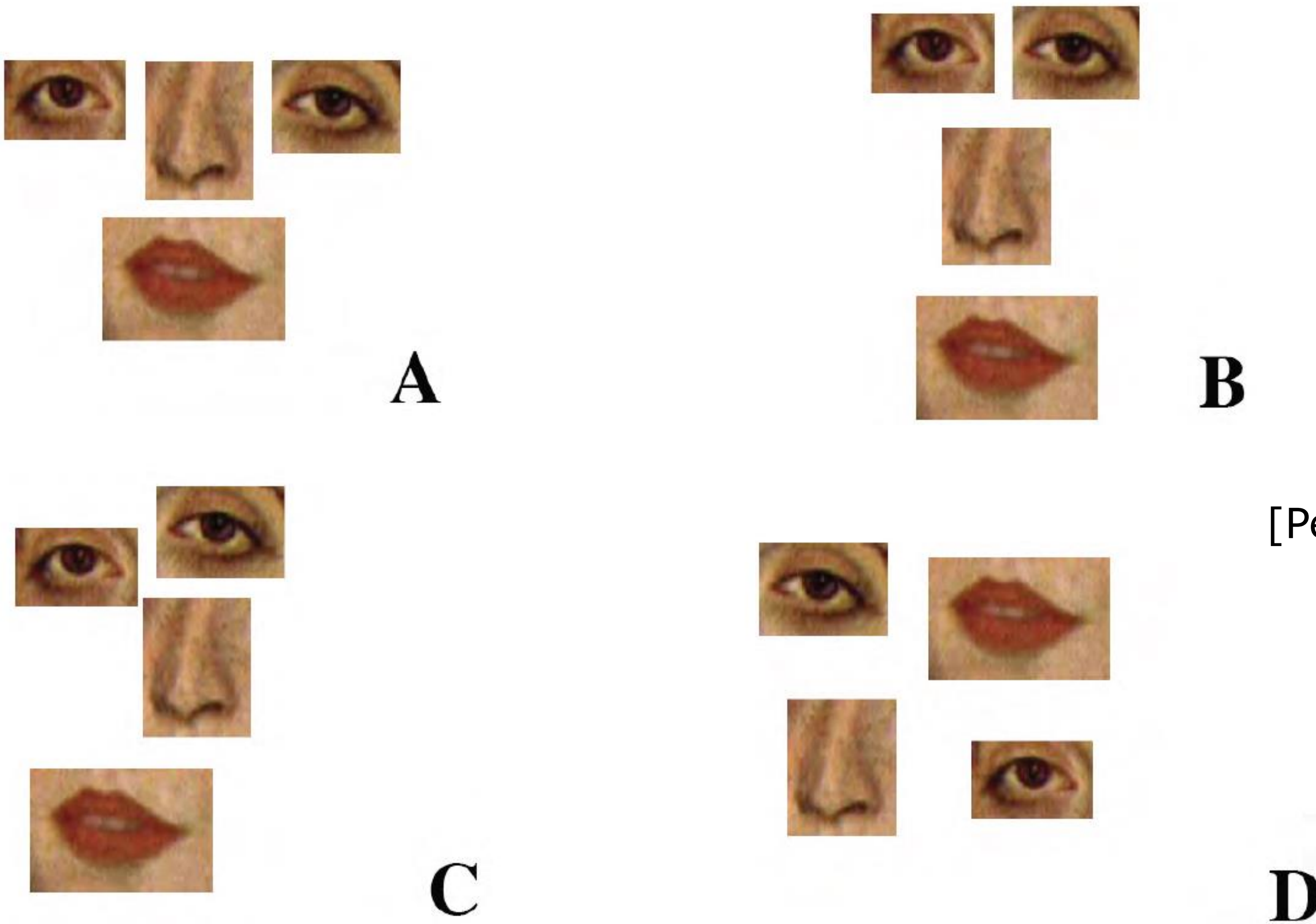
- Interface problem

Observations

In the world around us it mostly holds that:

- Neighboring things have something to do with each other
  - Spatially
  - Temporally
- There is hierarchical structure
  - Objects consist of parts
  - Parts are composed of components, …

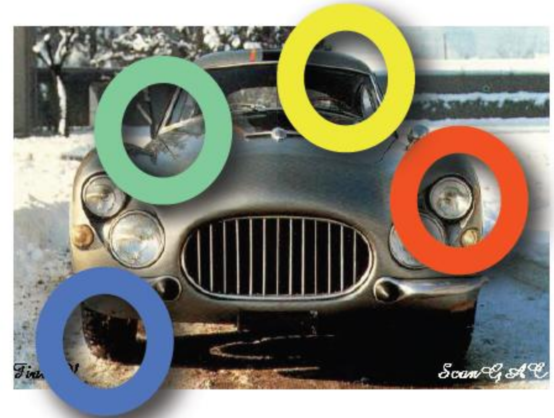
Spatial Arrangement of Facial Parts



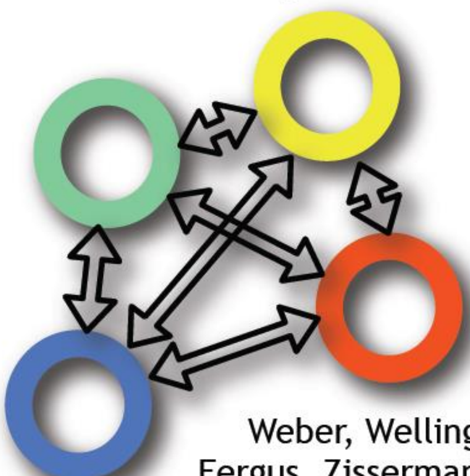
[Perona]

Face Perception

Horizontal and Vertical Dependencies

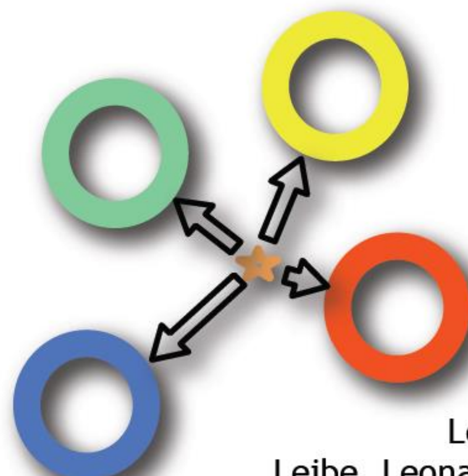


Constellation Model:
Fully connected shape model



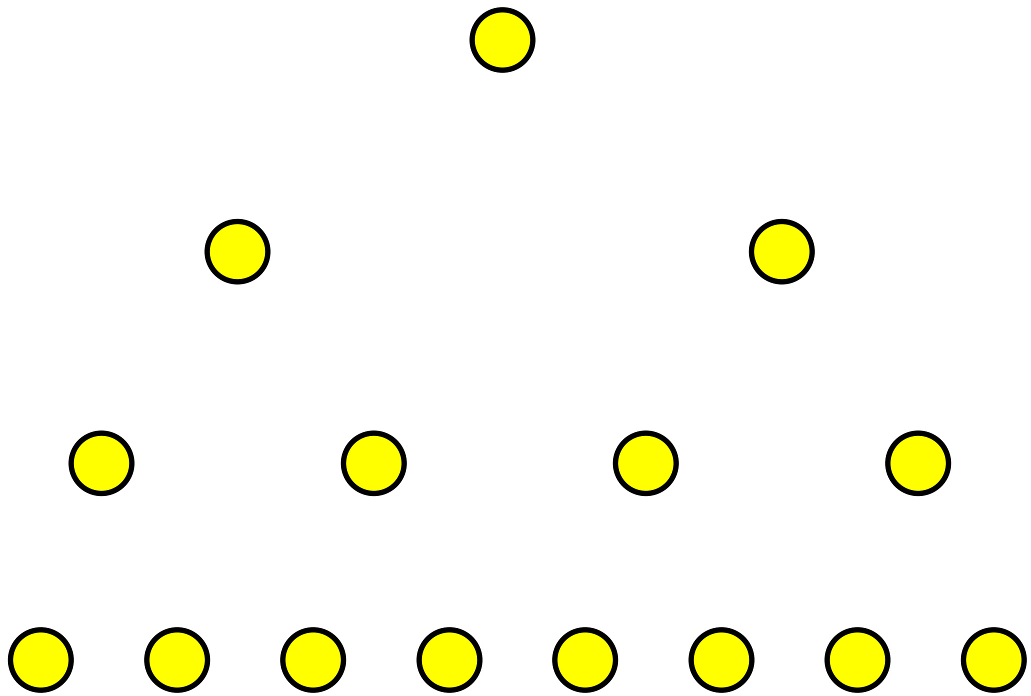
Weber, Welling, Perona ’00
Fergus, Zisserman, Perona ’03

Implicit Shape Model:
Star-Model w.r.t. Reference Point



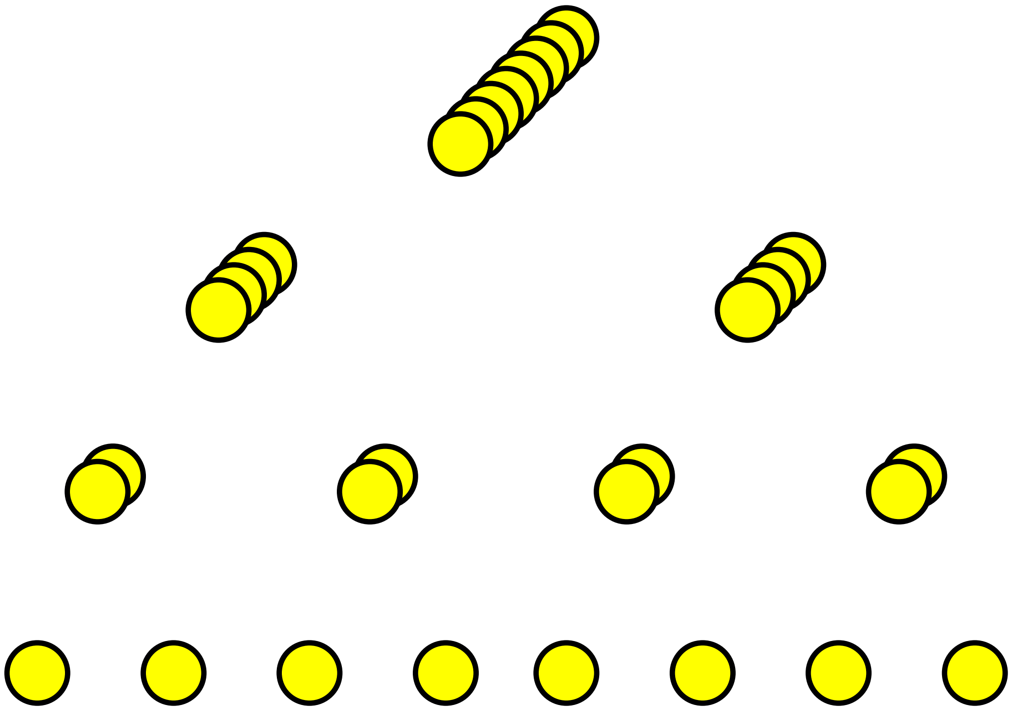
Leibe, Schiele ’03
Leibe, Leonardis, Schiele ’04

Multi-Scale Representation



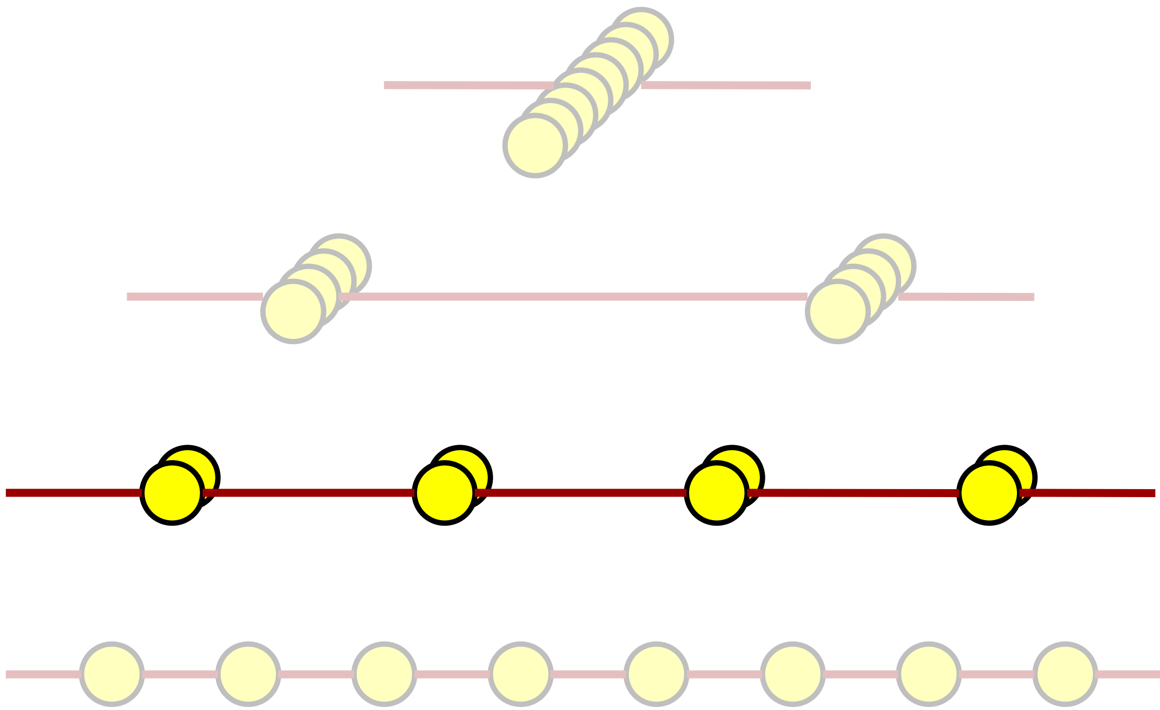
- Image pyramids are not expressive enough

Increasing Number of Features with Decreasing Resolution



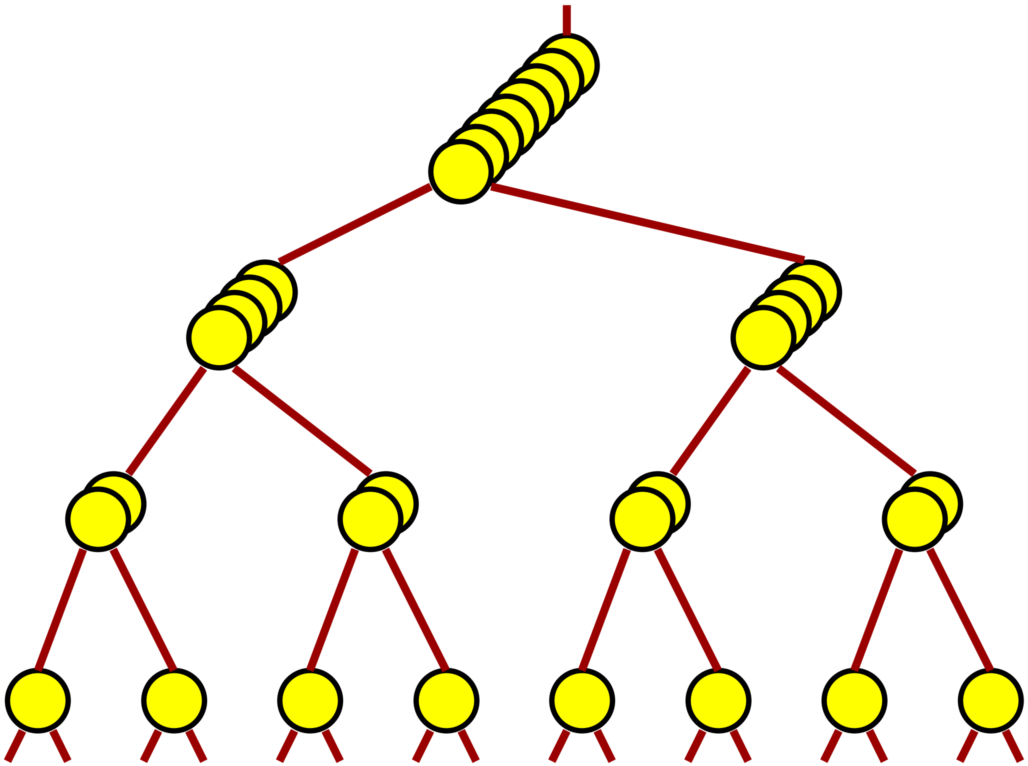
- Rich representations also in the higher layers

Modeling Horizontal Dependencies



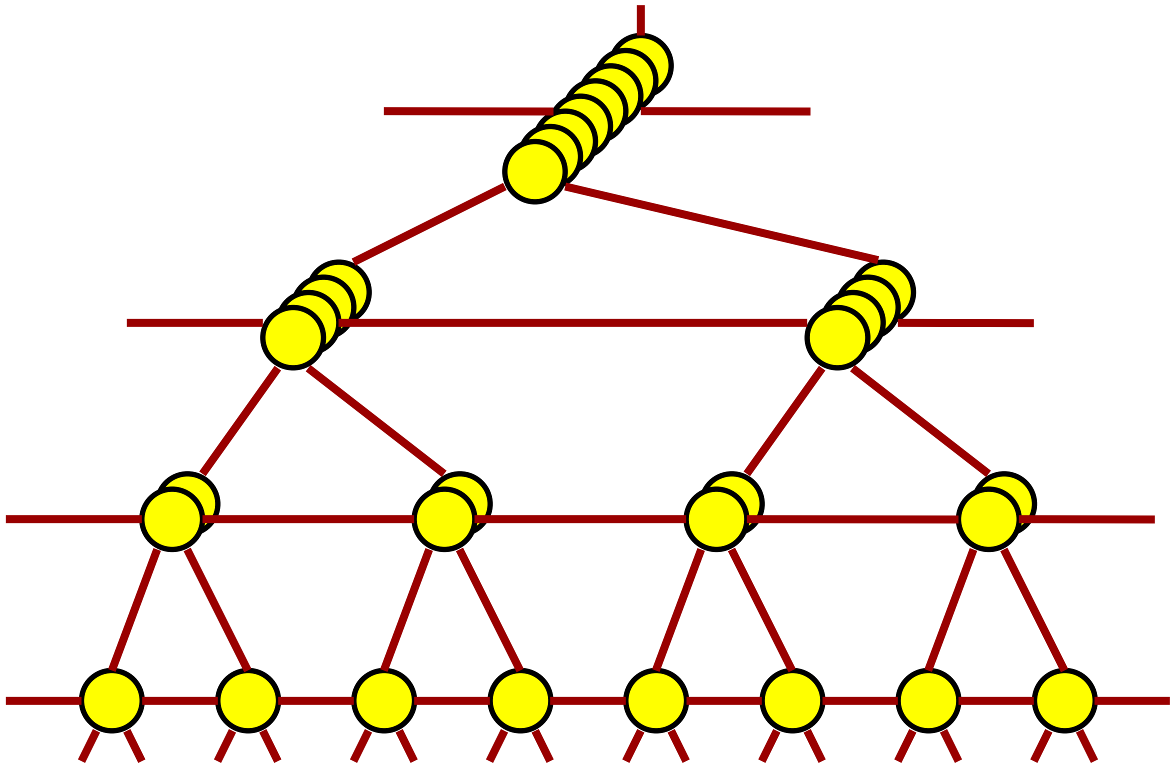
- 1D: HMM, Kalman Filter, Particle Filter
- 2D: Markov Random Fields
- Decision for level of description problematic
- Ignores vertical dependencies, flat models do not scale

Modeling Vertical Dependencies



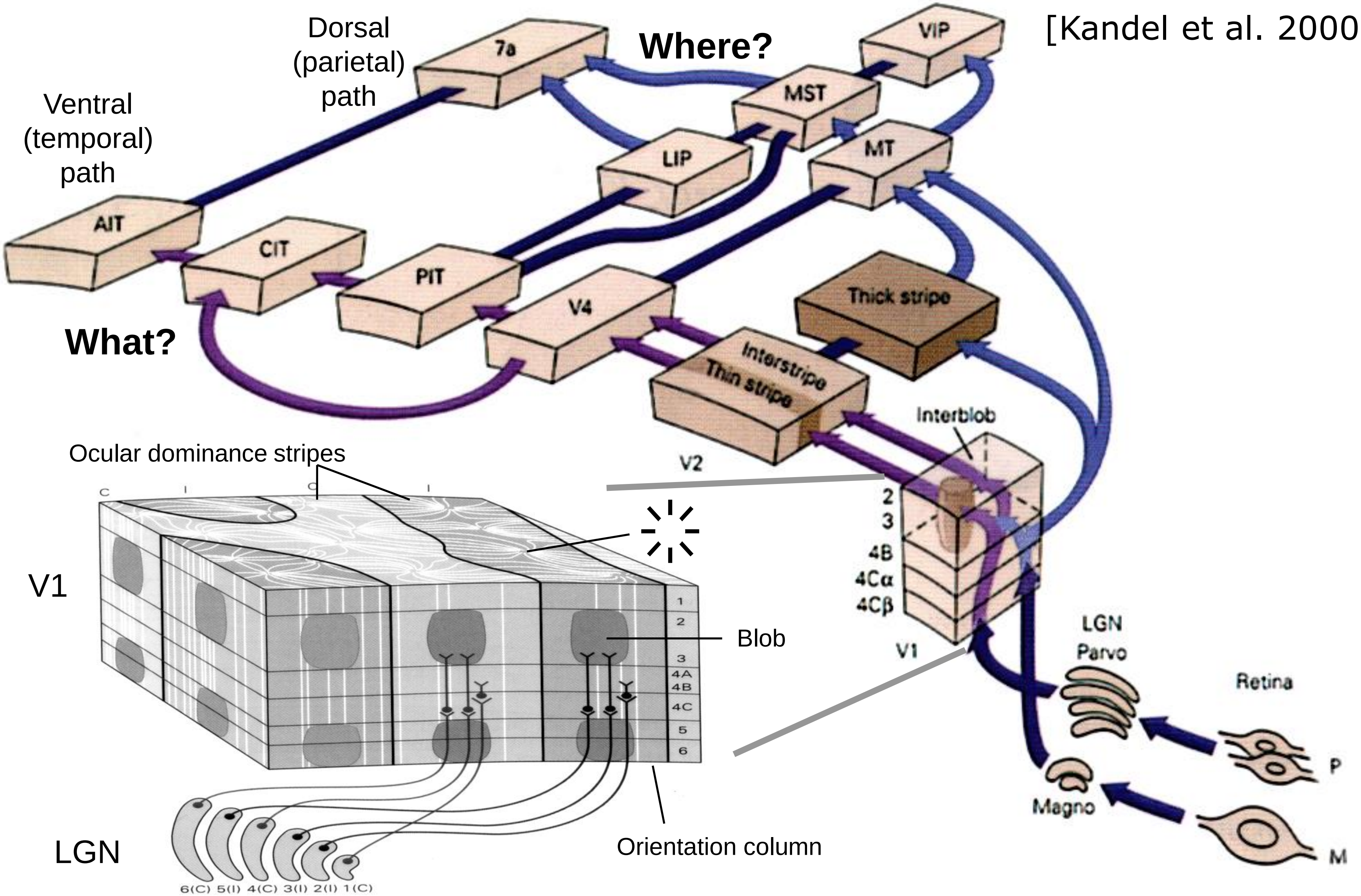
- Structure graphs, etc.
- Ignores horizontal dependencies

Horizontal and vertical Dependencies



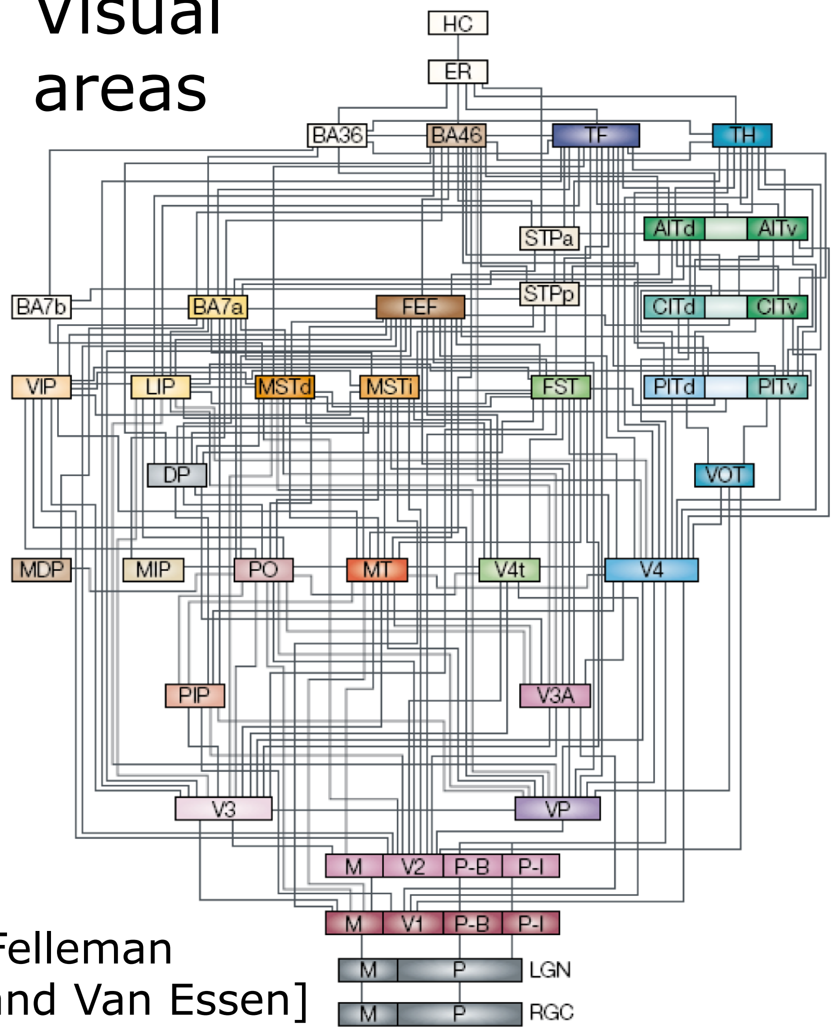
- Problem: Cycles make exact inference impossible
- Idea: Use approximate inference

Human Visual System



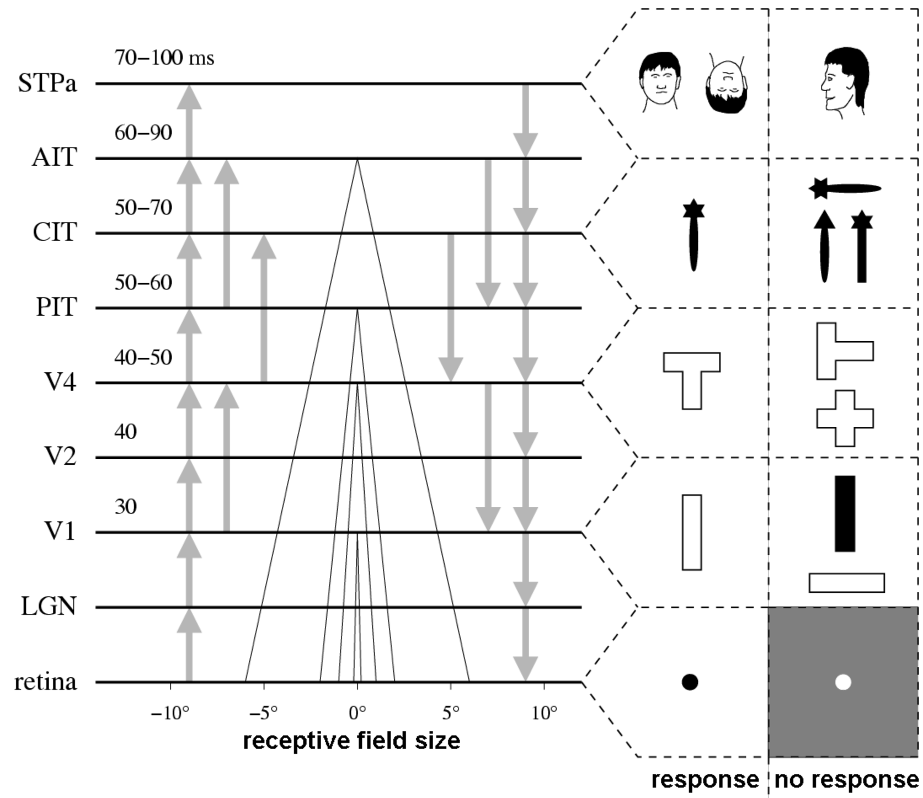
Visual Processing Hierarchy

- Visual areas



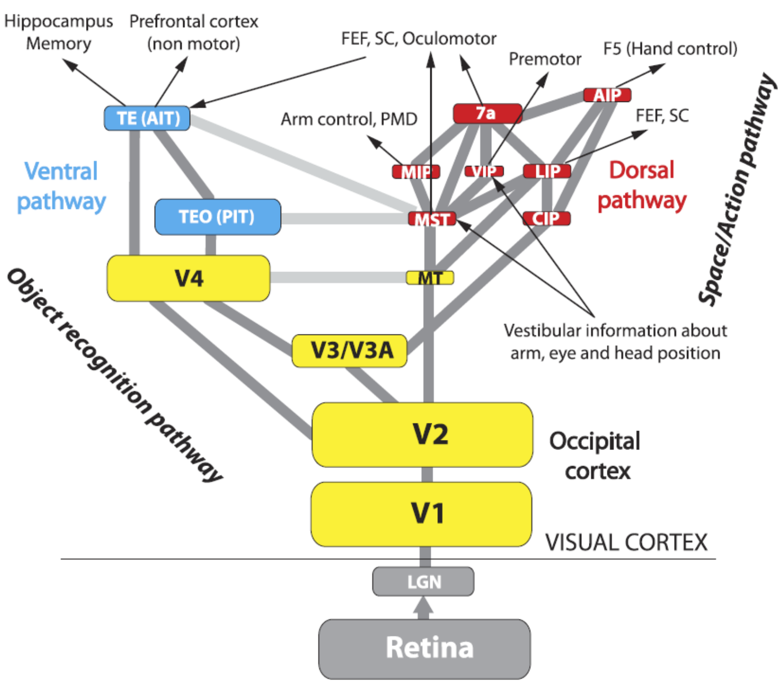
[Felleman and Van Essen]

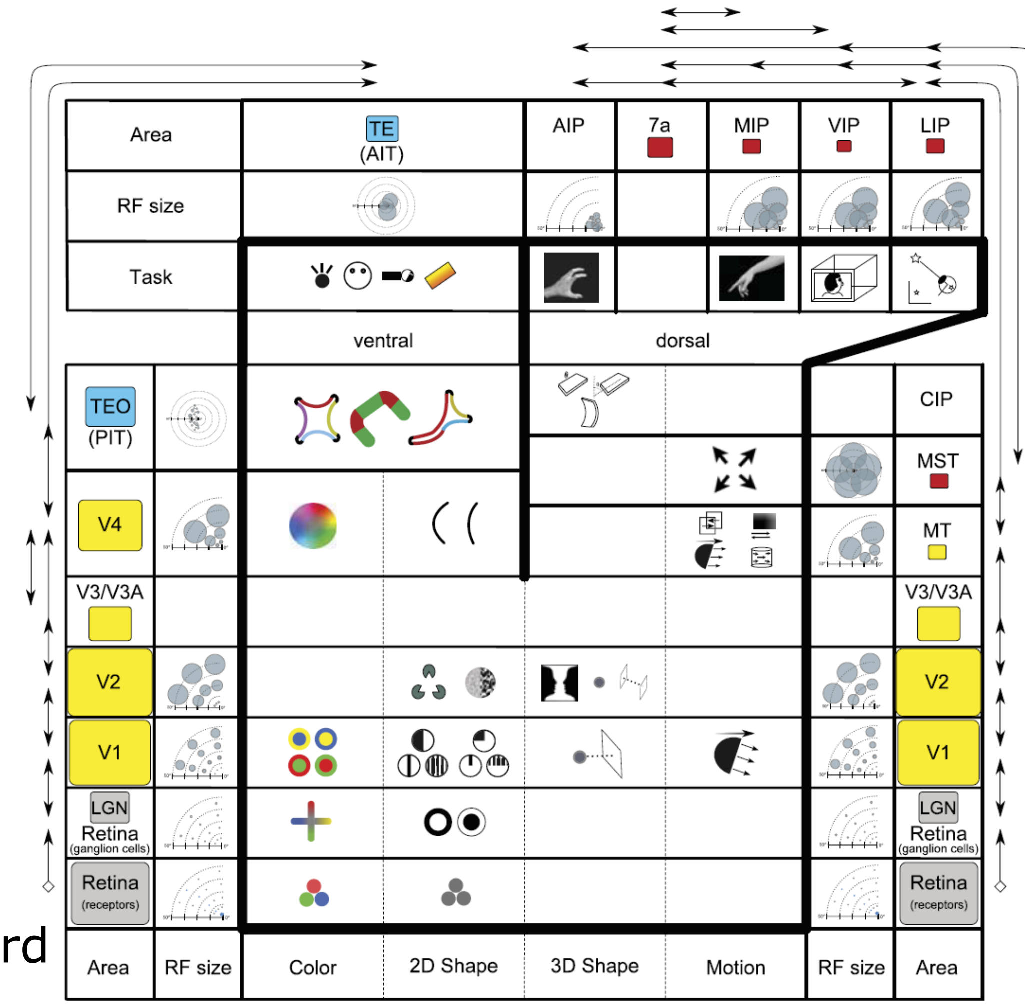
- Represented features



[Wiskott]

Visual Processing Hierarchy

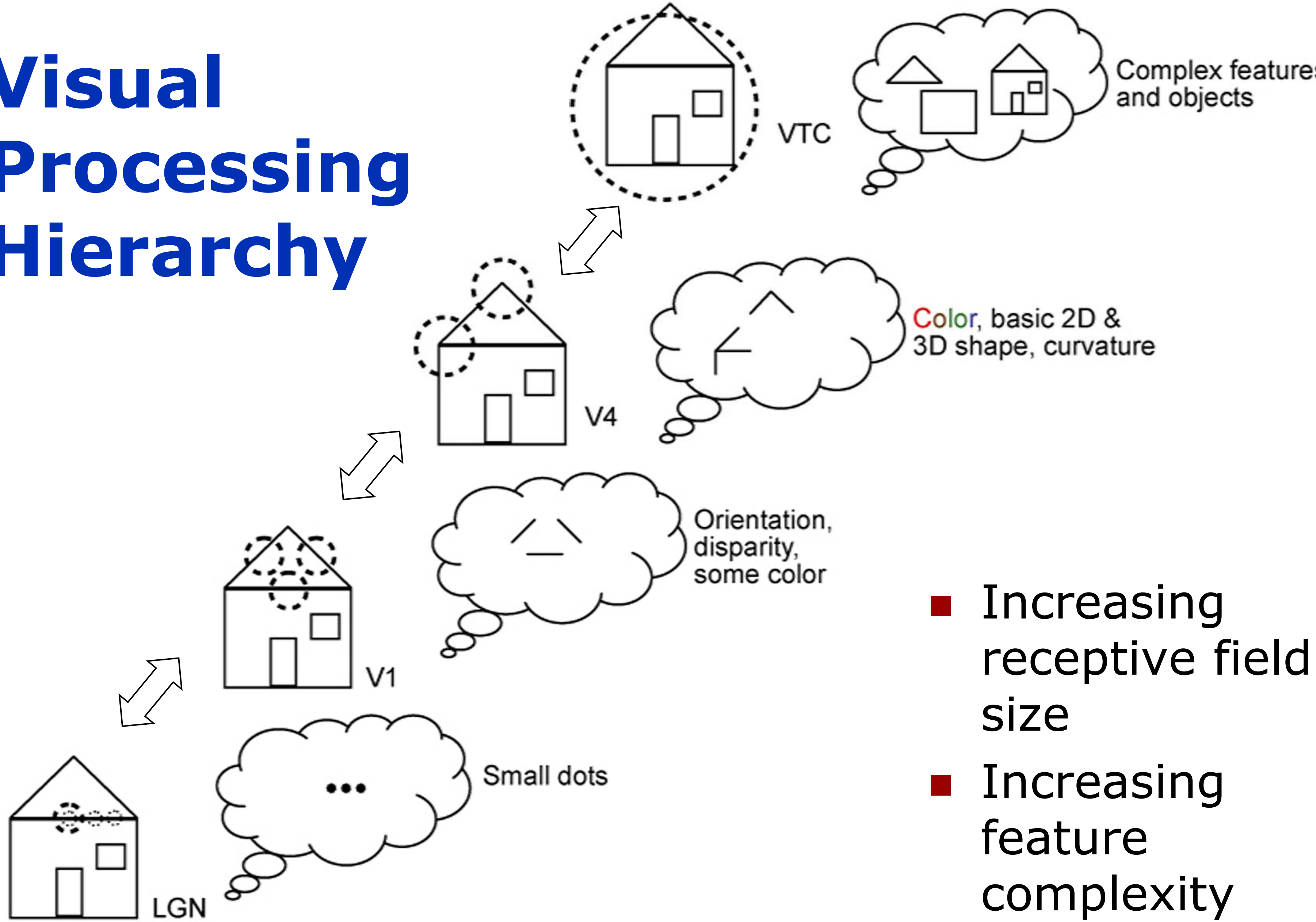




- Increasing complexity
- Increasing invariance
- All connections bidirectional
- More feedback than feed forward
- Lateral connections important

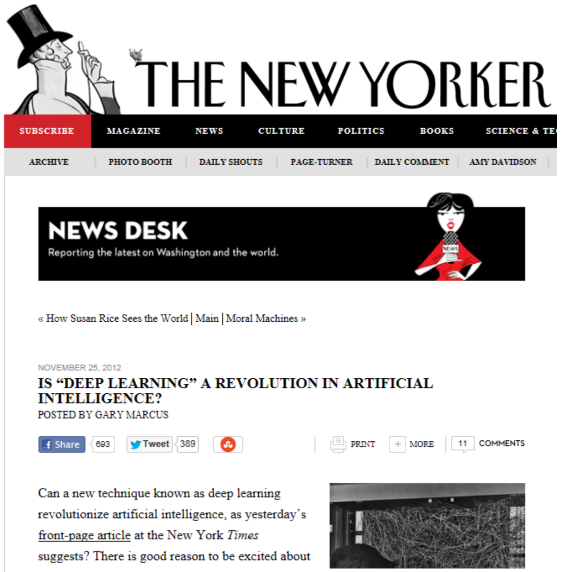
[Krüger et al., TPAMI 2013]

Visual Processing Hierarchy



- Increasing receptive field size
- Increasing feature complexity

Trend since 2006: Deep Learning


THE NEW YORKER

SUBSCRIBE MAGAZINE NEWS CULTURE POLITICS BOOKS SCIENCE & TE

ARCHIVE | PHOTO BOOTH | DAILY SHOUTS | PAGE-TURNER | DAILY COMMENT | AMY DAVIDSON

NEWS DESK
Reporting the latest on Washington and the world.

« How Susan Rice Sees the World | Main | Moral Machines »

NOVEMBER 25, 2012

IS "DEEP LEARNING" A REVOLUTION IN ARTIFICIAL INTELLIGENCE?

POSTED BY GARY MARCUS

Share 693 Tweet 389 PRINT MORE 11 COMMENTS

Can a new technique known as deep learning revolutionize artificial intelligence, as yesterday's front-page article at the New York *Times* suggests? There is good reason to be excited about


The Register®

Data Centre Cloud Software Networks Security Policy Business Jobs Hardware Science Bootno

Financial News Small Biz CIO Media

BUSINESS

Baidu muscles in on Google's turf with Silicon Valley deep learning lab

Chinese search giant beds down next to Apple in Cupertino

By Phil Muncaster, 15th April 2013 Follow 3,371 followers

1

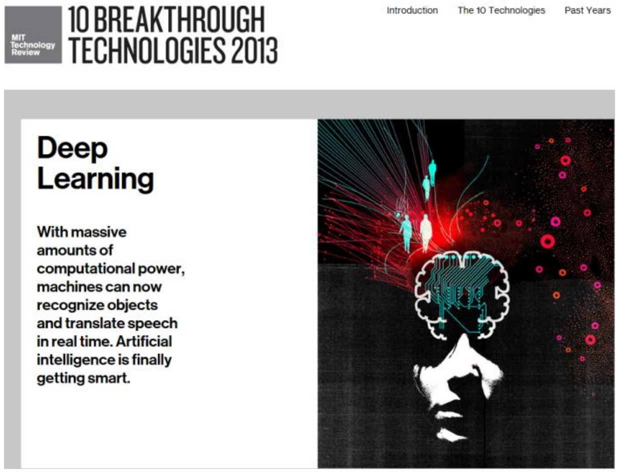
RELATED STORIES

China's Baidu builds new type of App Store

Win a Samsung 40-inch LED HDTV with The Reg and HP!

Chinese search giant Baidu has opened the doors to a new research facility in Google's back yard where it's hoping to tap the local talent to consolidate early mover advantage in the burgeoning field of "deep learning".

The Cupertino-based Institute of Deep Learning (IDL) is the Silicon Valley counterpart of another facility back in China dedicated to accelerating research in the emerging machine learning-related discipline.

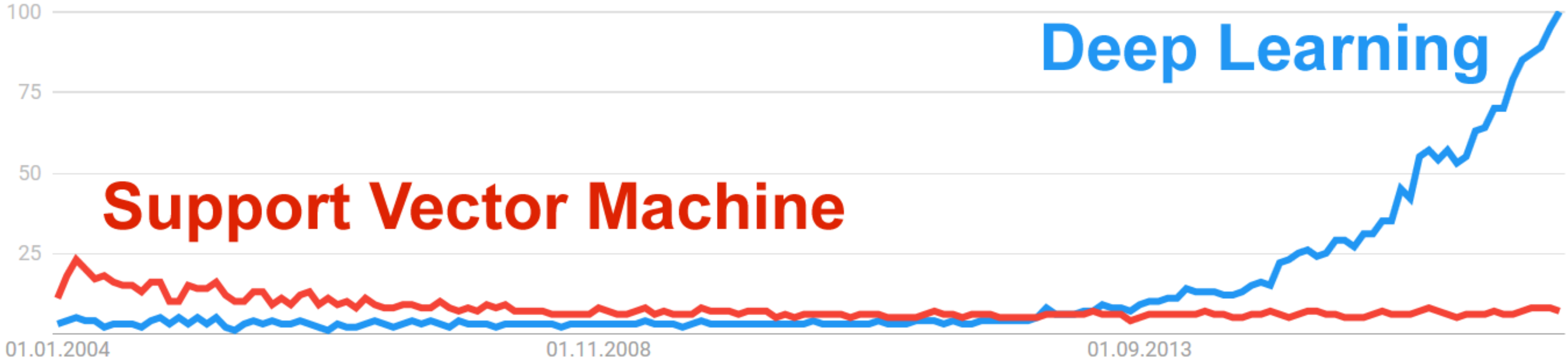

MIT Technology Review

10 BREAKTHROUGH TECHNOLOGIES 2013

Introduction The 10 Technologies Past Years

Deep Learning

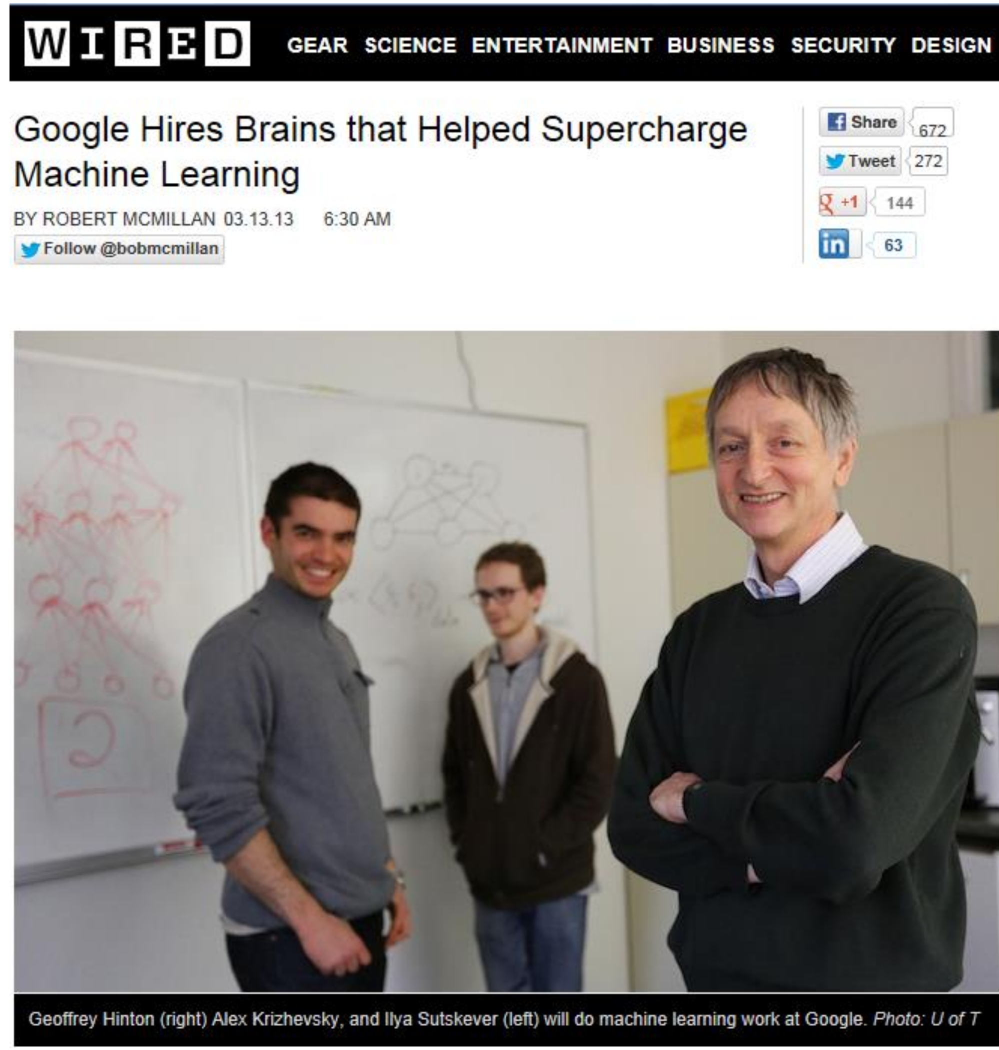
With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.



[Google Trends]

Strong Interest of Industry

- Google
  - DNNresearch (Geoffrey Hinton)
  - DeepMind (Demis Hassabis)
- Baidu
  - Andrew Ng
- Facebook
  - Yann LeCun
- Microsoft
  - Li Deng

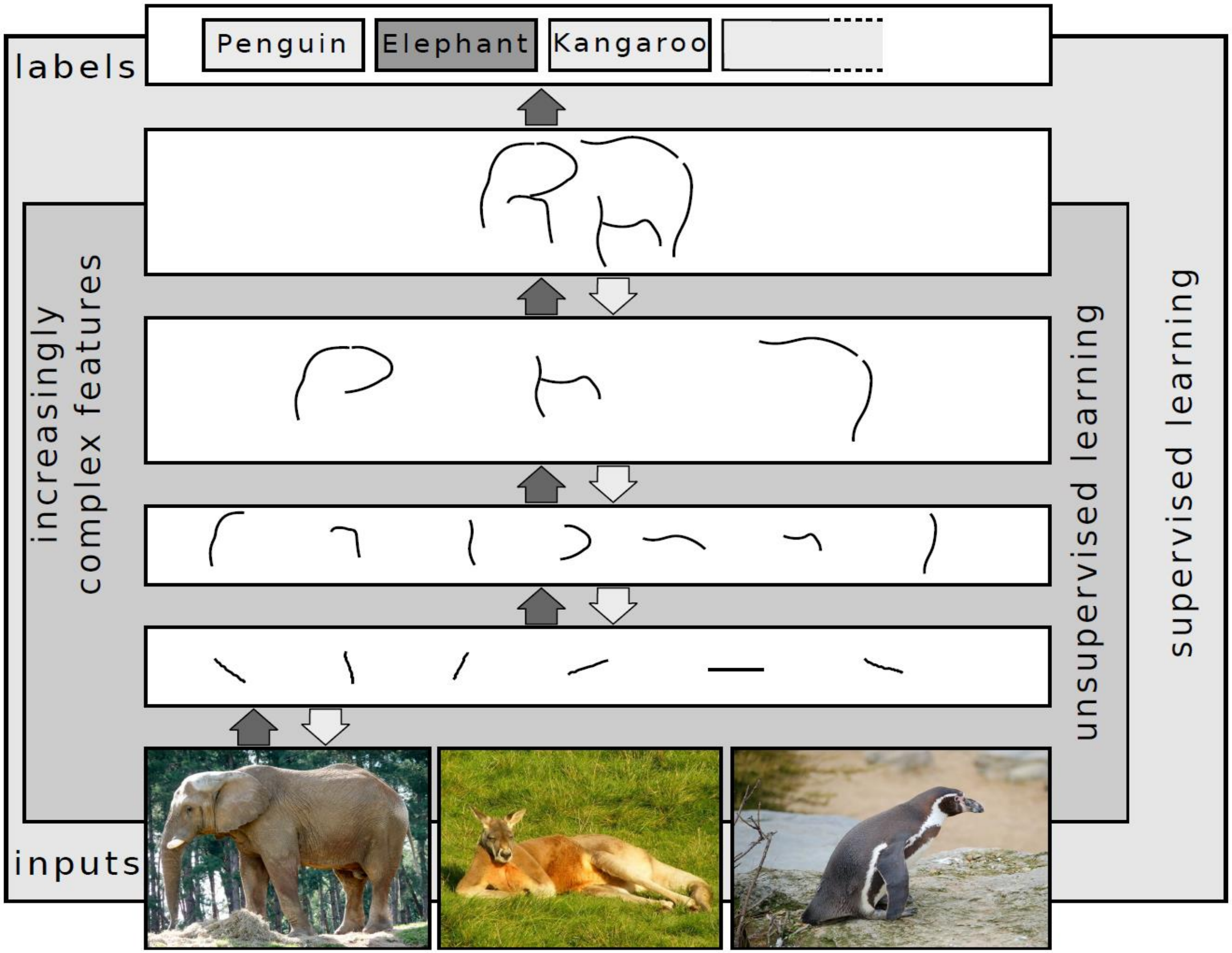


Deep Learning Definition

- Deep learning is a set of algorithms in machine learning that attempt to **learn layered models of inputs**, commonly neural networks.
- The layers in such models correspond to **distinct levels of concepts**, where
  - higher-level concepts are defined from lower-level ones, and
  - the same lower-level concepts can help to define many higher-level concepts.

[Bengio 2009]

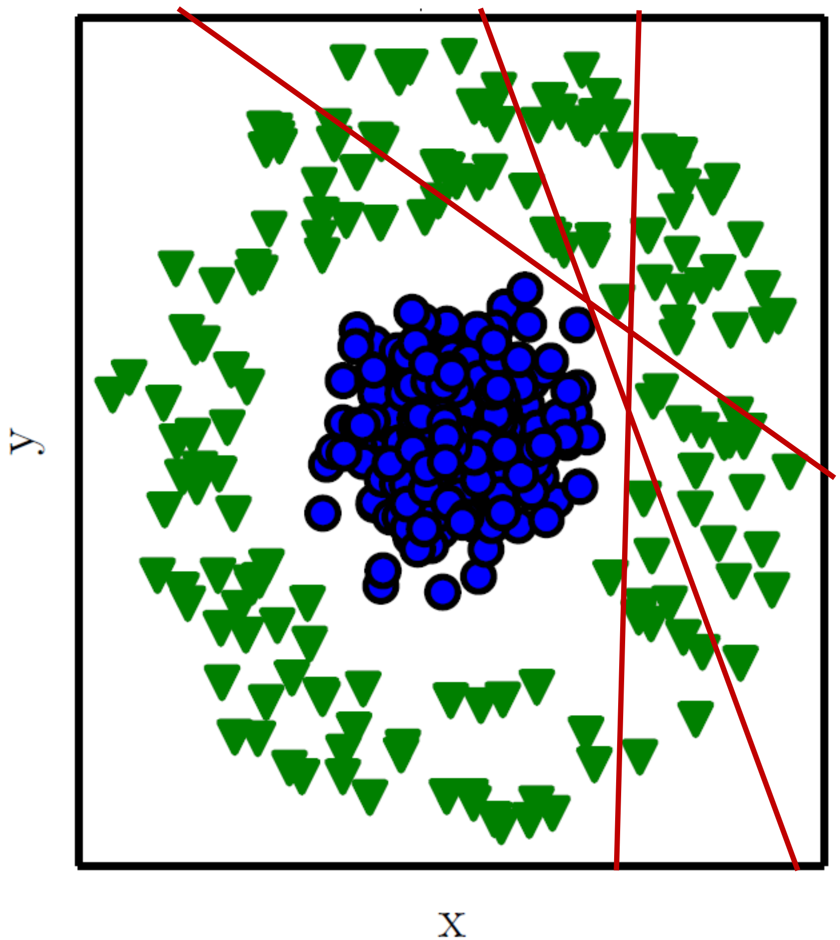
Layered Representations



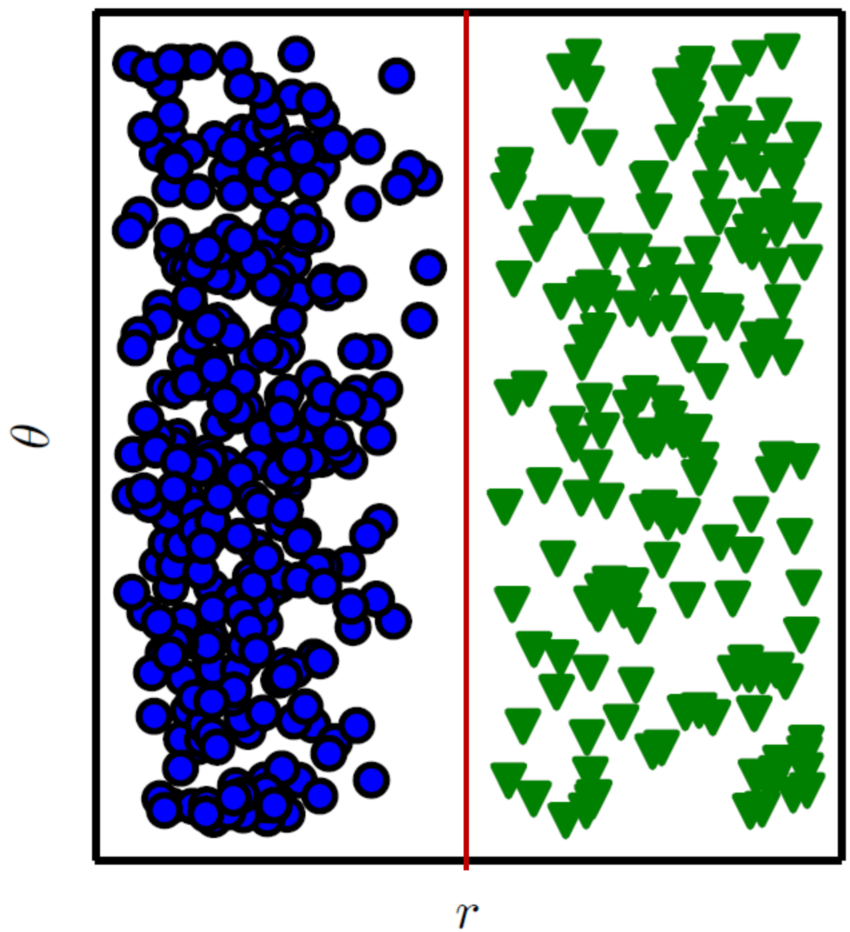
[Schulz and Behnke, KI 2012]

Representations Matter

- Cartesian coordinates



- Polar coordinates

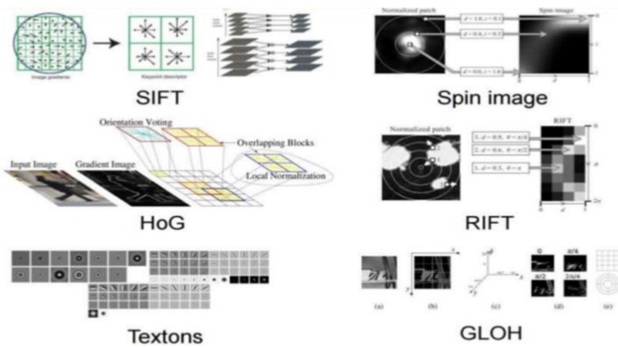


[Goodfellow]

From Hand-crafted Features to Feature Learning

- Traditional computer vision

Hand-crafted features



Simple classifier

„Cat"

- Deep learning

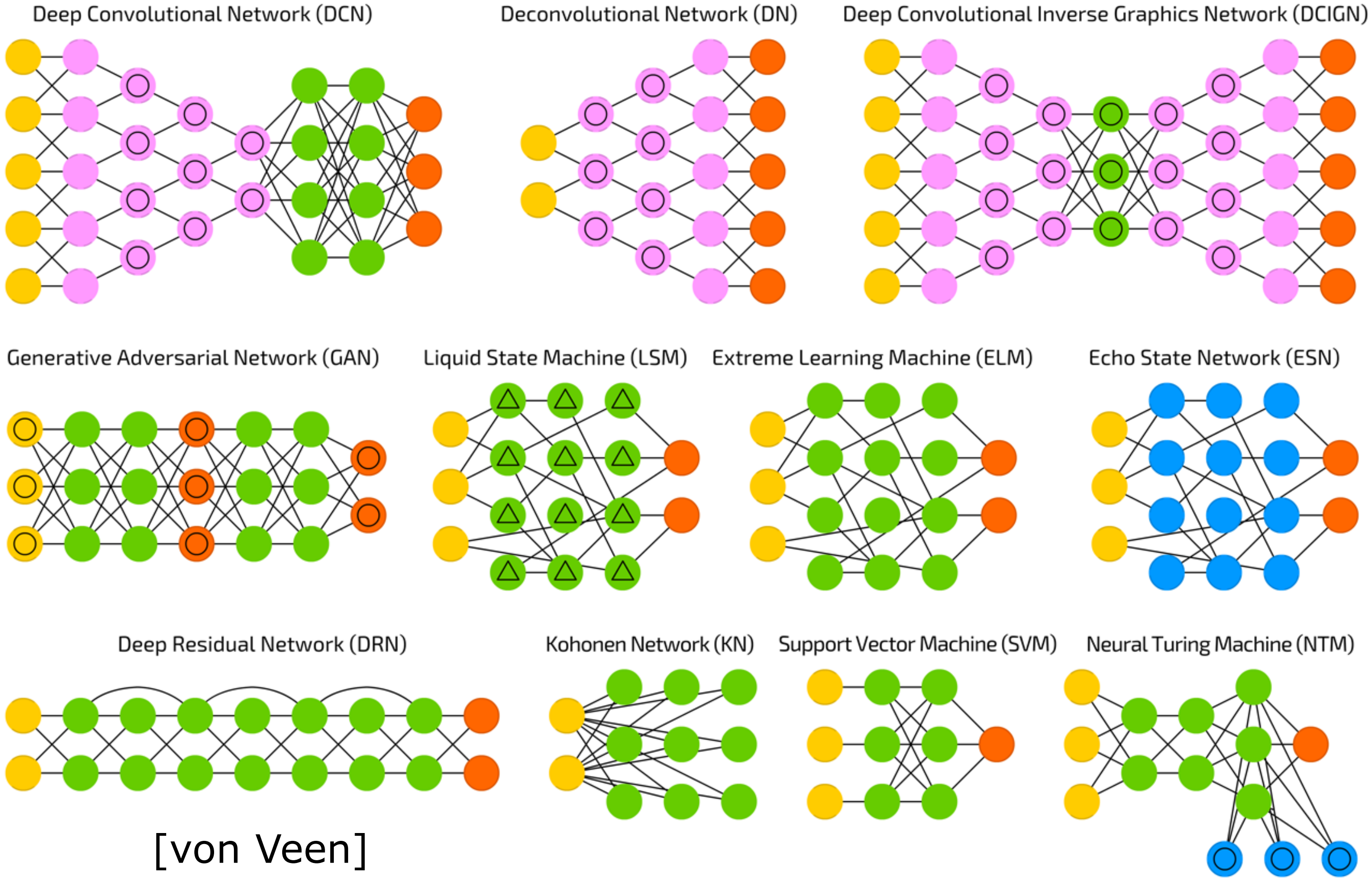
Learned features

Classifier

„Cat"

From Feature Engineering to Model Engineering

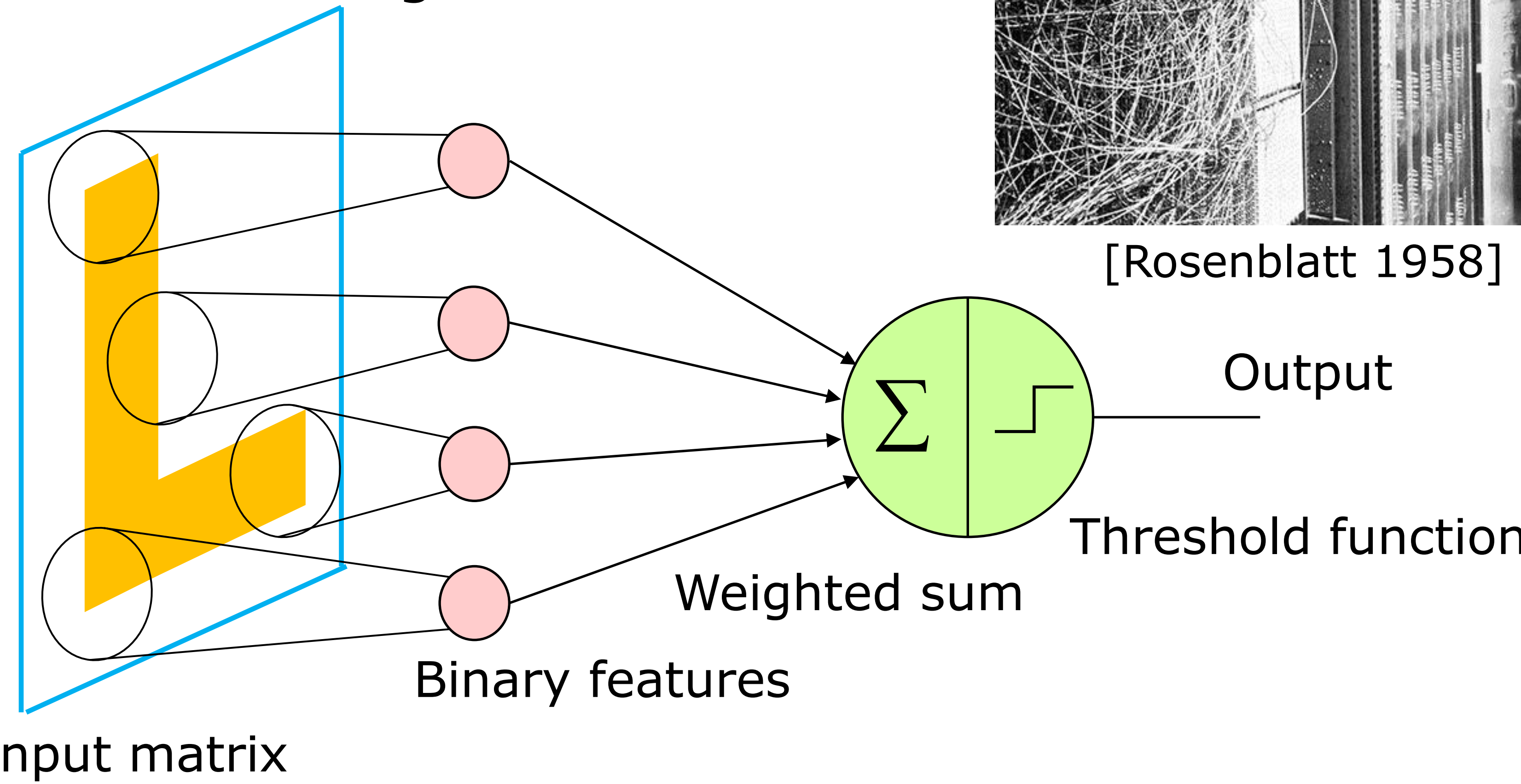
- Structure of the model matters

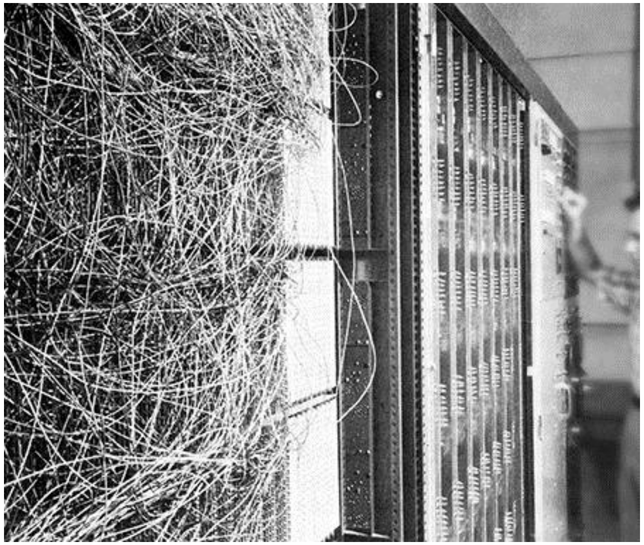


[von Veen]

Perceptron

- Feature extraction
- Pattern recognition





[Rosenblatt 1958]

Supervised Training of Neural Networks

- **Goal:** A function $\mathbf{y}=f(\mathbf{x})$, which is given by examples, shall be approximated by the neural network. Choose the weights w_{ij} to minimize a loss function which measures the approximation error.

- Set of training examples $\{(\mathbf{x}_1,\mathbf{t}_1), \ldots, (\mathbf{x}_p,\mathbf{t}_p)\}$
- The networks maps input $\mathbf{x}_i$ to output $\mathbf{y}_i$
- Example loss: Quadratic error

$$E(w) = 1/2 \sum_{i=1}^{p} (y_i - t_i)^2$$

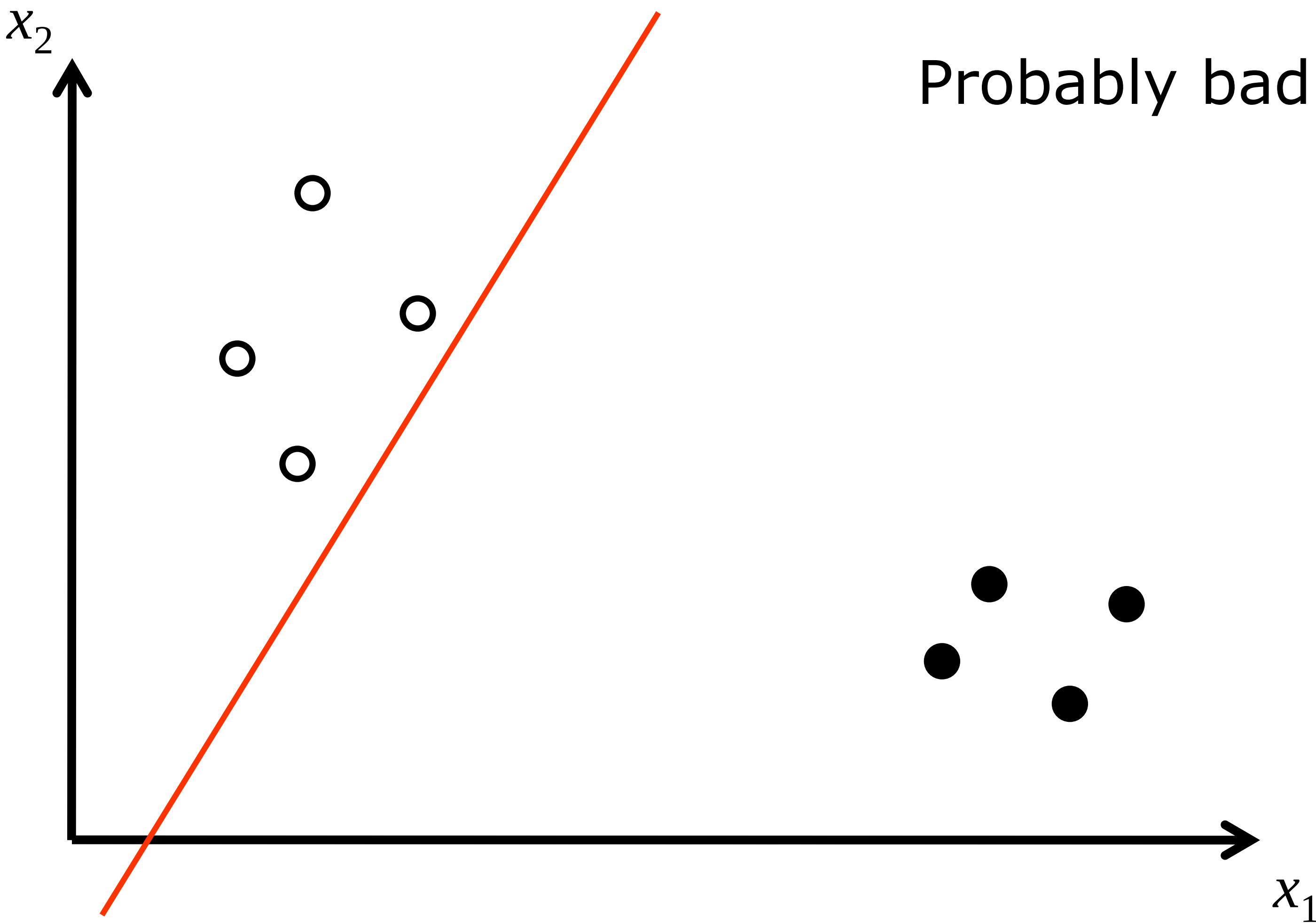
Learning = Generalization

H. Simon -

"Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the task or tasks drawn from the same population more efficiently and more effectively the next time."

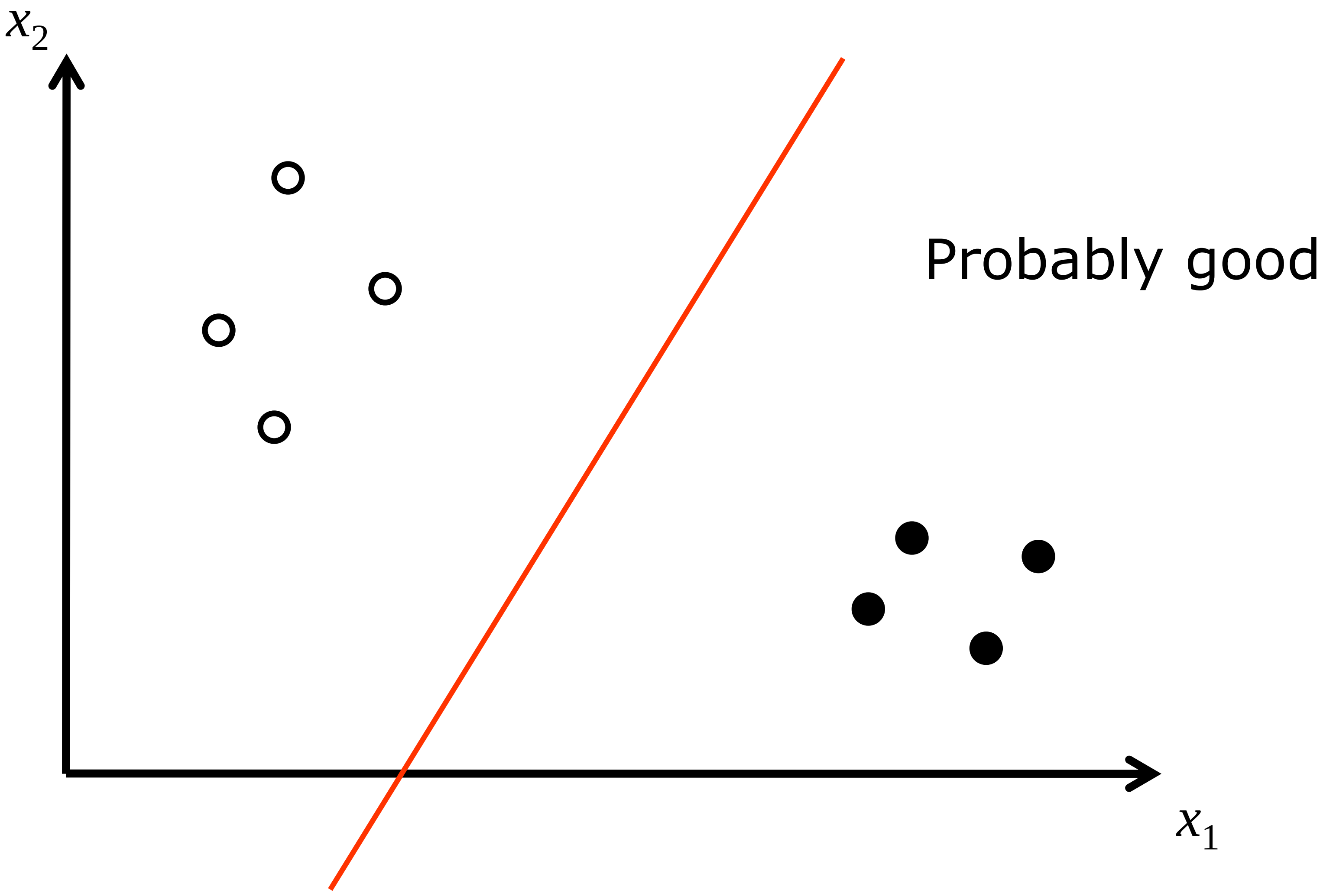
The ability to perform a task in a situation which has never been encountered before

Generalization



Probably bad

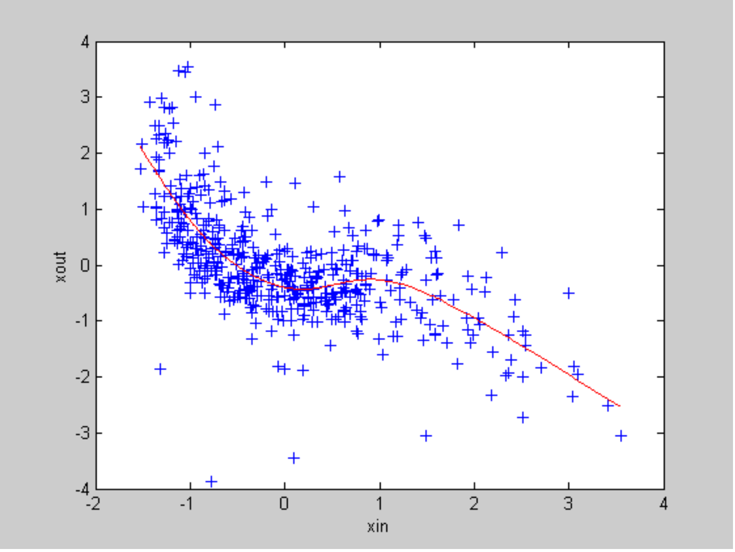
Generalization



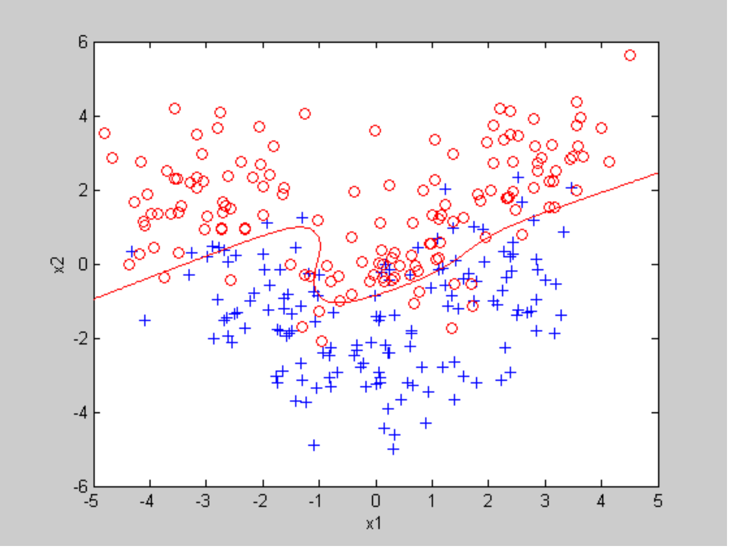
Stochastic View of Supervised Learning

- Only noisy examples of function available
- Two types of learning problems:

Regression



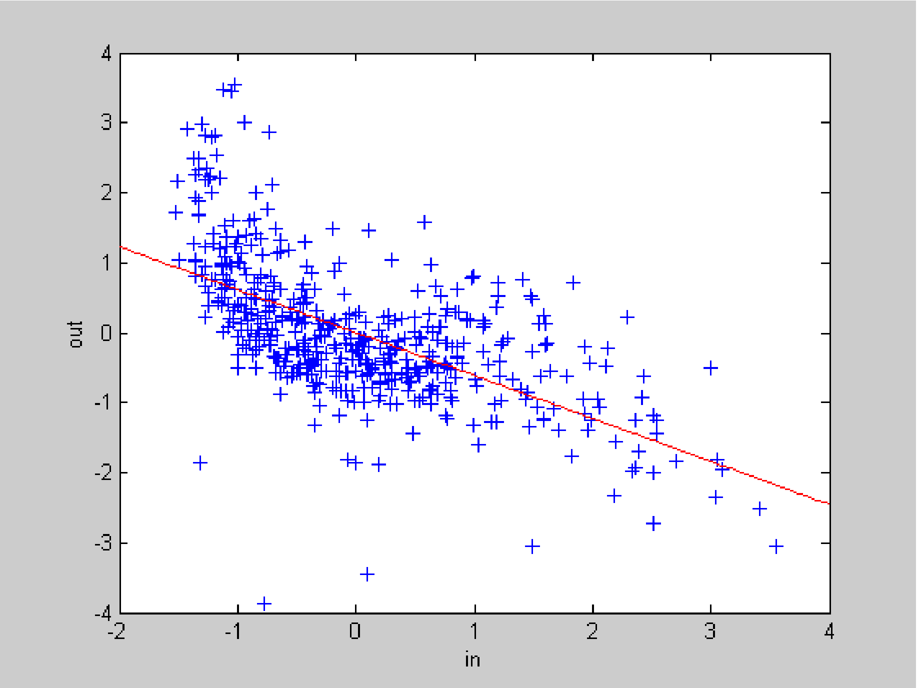
Classification



- Learner must find a mathematical model

Example: Linear Regression

- Training linear neural networks with quadratic loss is linear regression



- 1D case:

$$t = ax^{\text{in}} + b + \varepsilon$$

- General:

Noise

$$\mathbf{t} = \mathbf{W}\mathbf{x}^{\mathbf{in}} + w_0 + \varepsilon$$

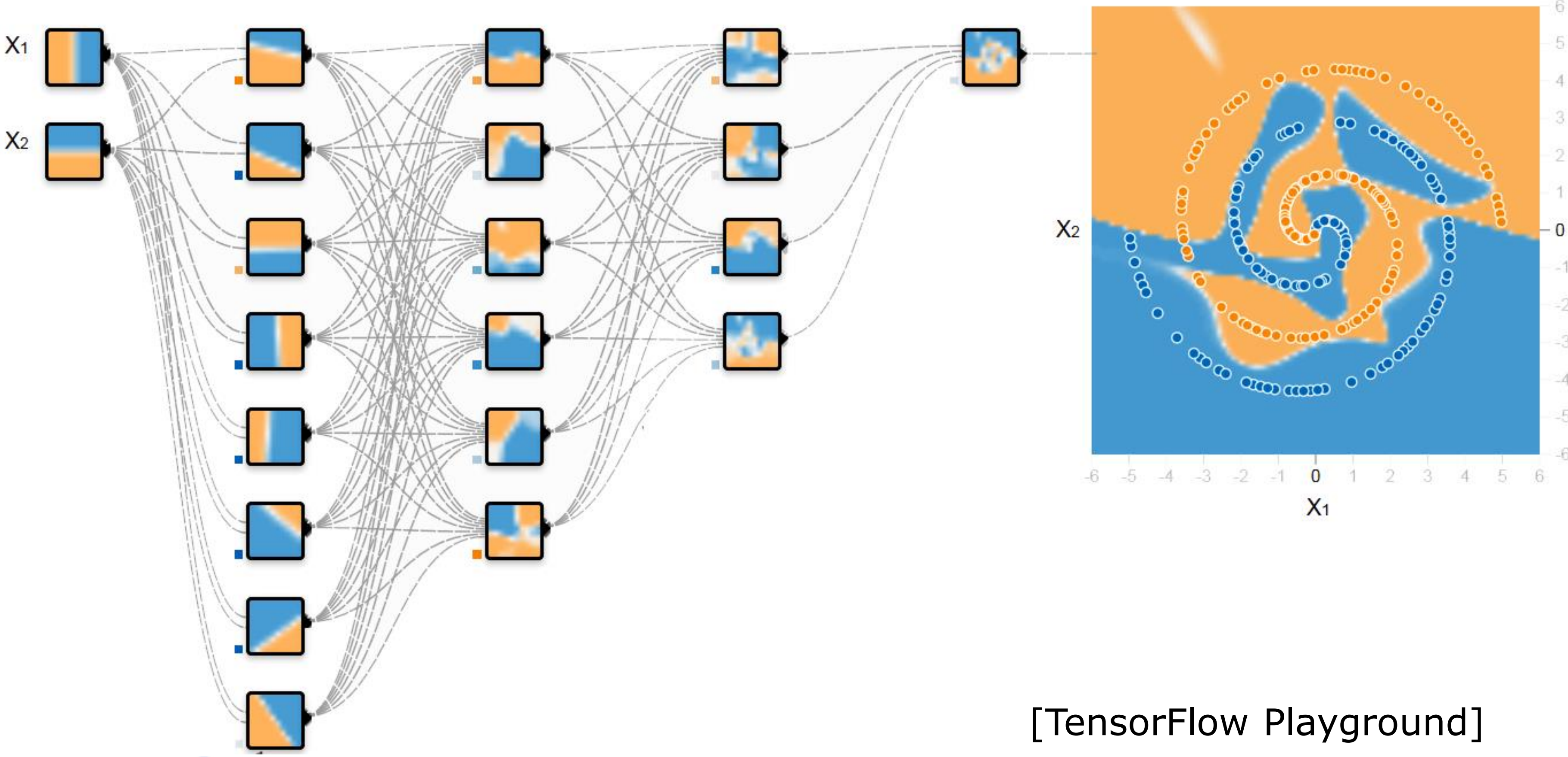
Output variables („target“)

Input variables

- How to find weights and biases that make the error minimal?

Multi-Layer Perzeptron

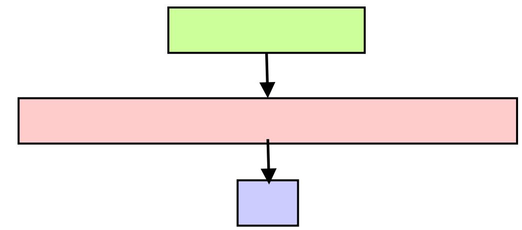
- Non-linear separation of input space
- Backpropagation algorithm [Rumelhart et al. 1986]



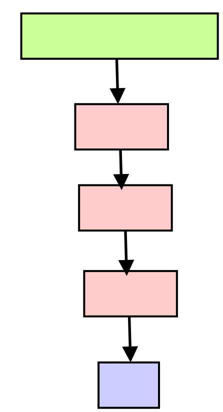
[TensorFlow Playground]

Flat vs. Deep Networks

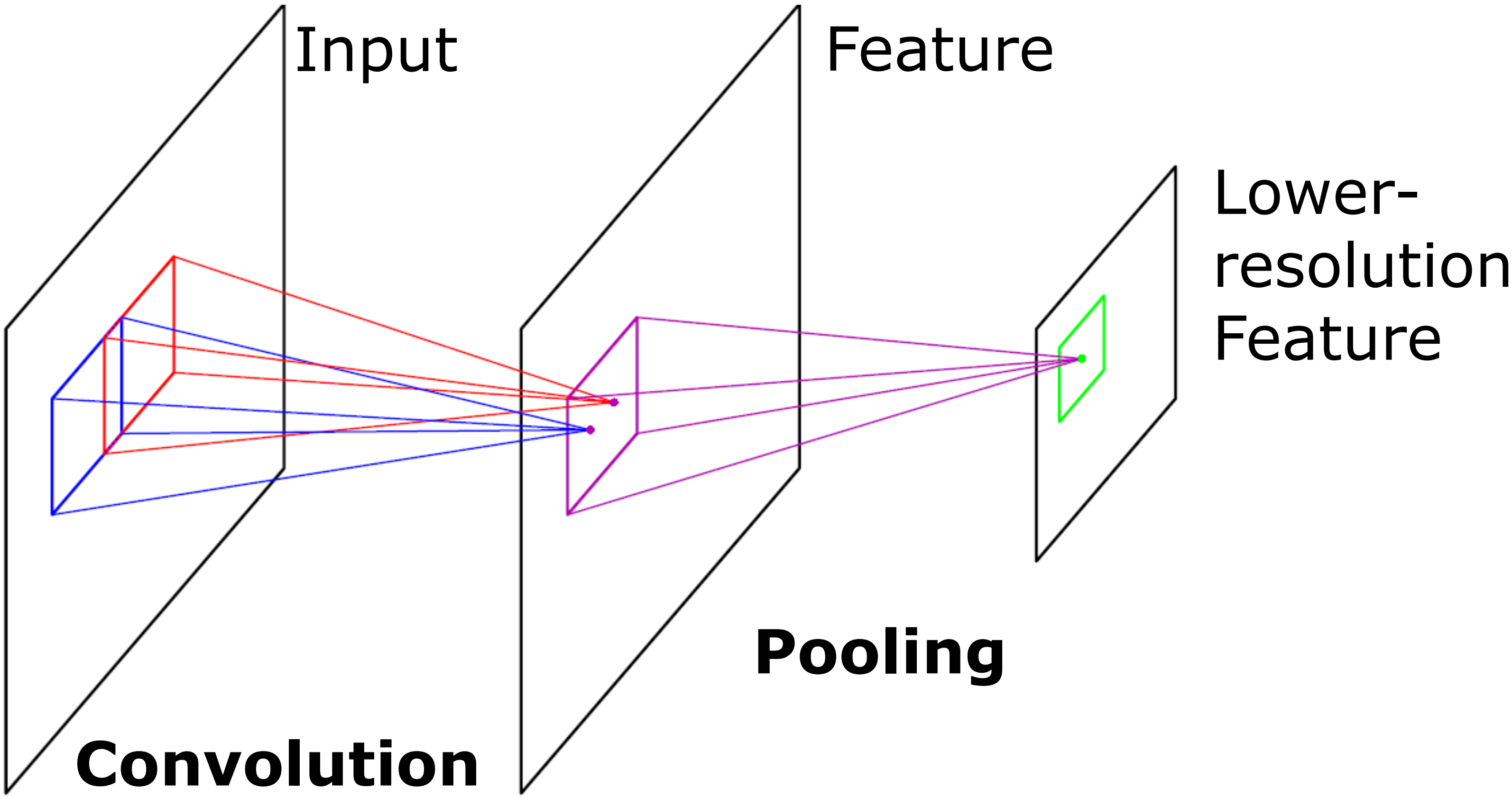
- A neural network with a **single hidden layer** that is wide enough can compute any function (Cybenko, 1989)
  - Certain functions, like parity, may require exponentially many hidden units (in the number of inputs)
  - Compare to conjunctive / disjunctive normal form of Boolean function



- **Deep networks** (with multiple hidden layers) may be exponentially more efficient
  - Parity example:
    - As many hidden layers as inputs
    - Compute carry bit sequentially

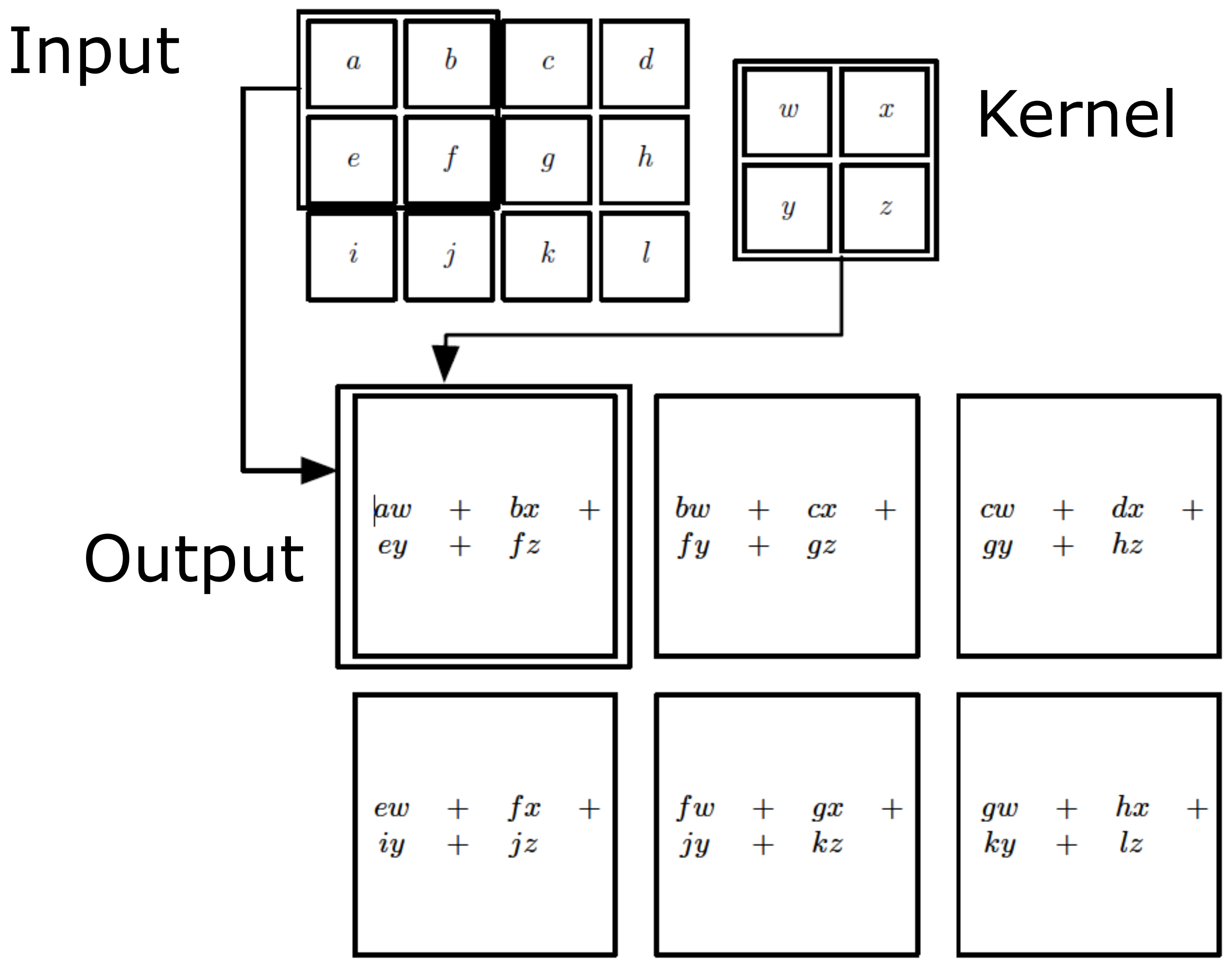


Convolutional Neural Networks



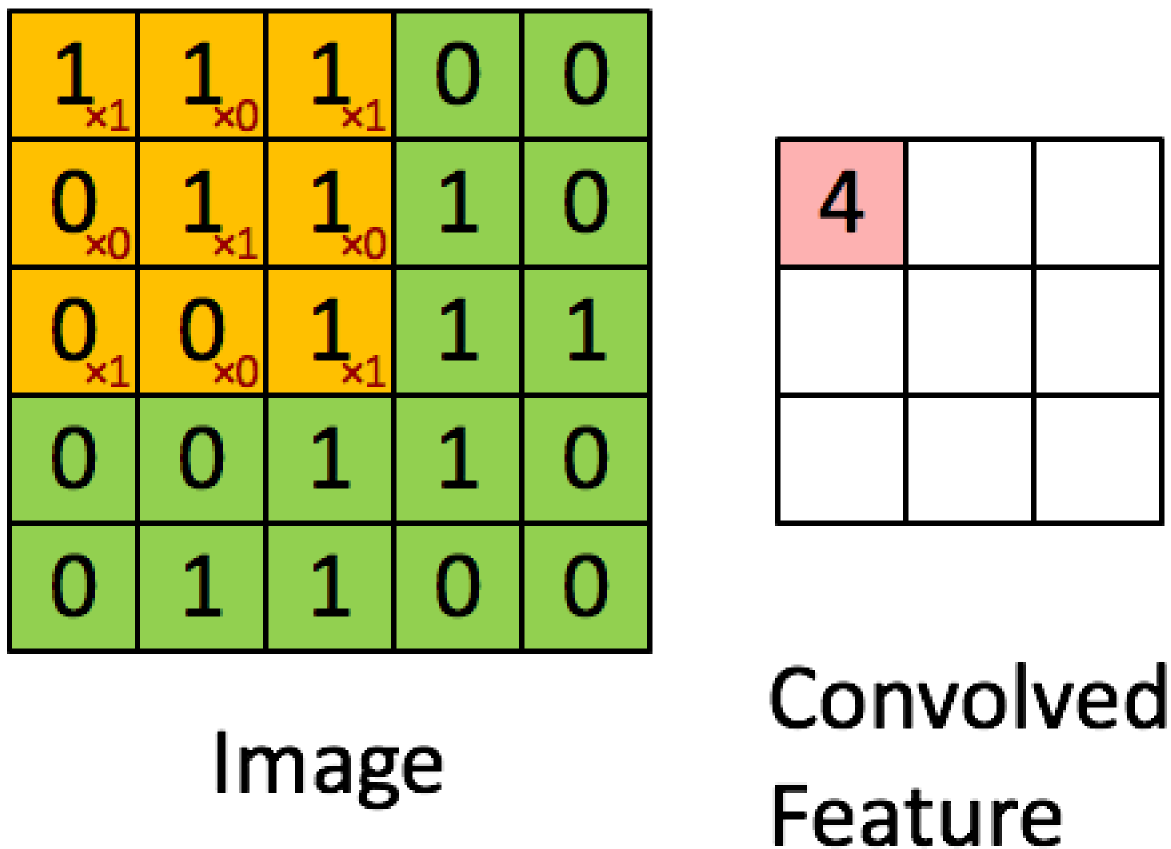
[Bishop]

2D Convolution



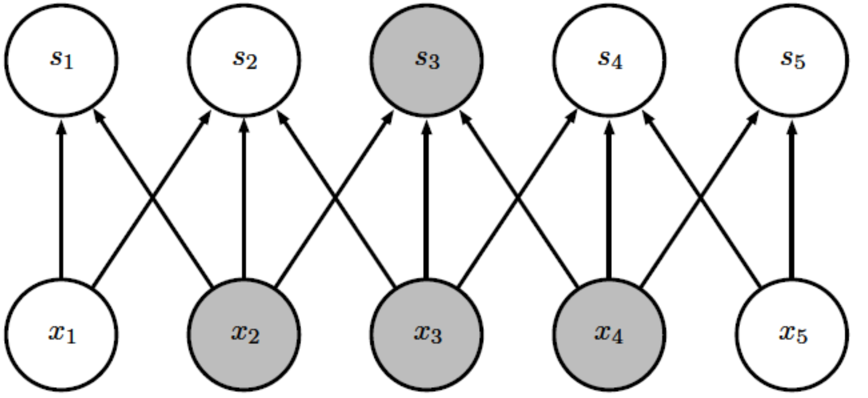
[Goodfellow]

Convolution Example

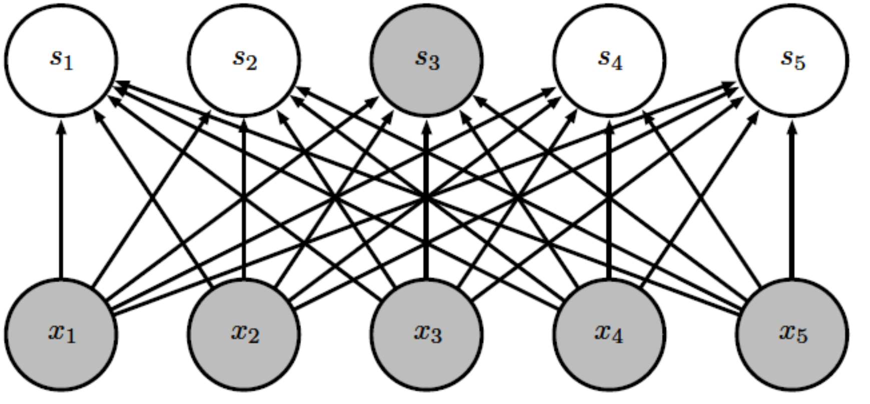


Sparse Local Connectivity

- 1D convolution



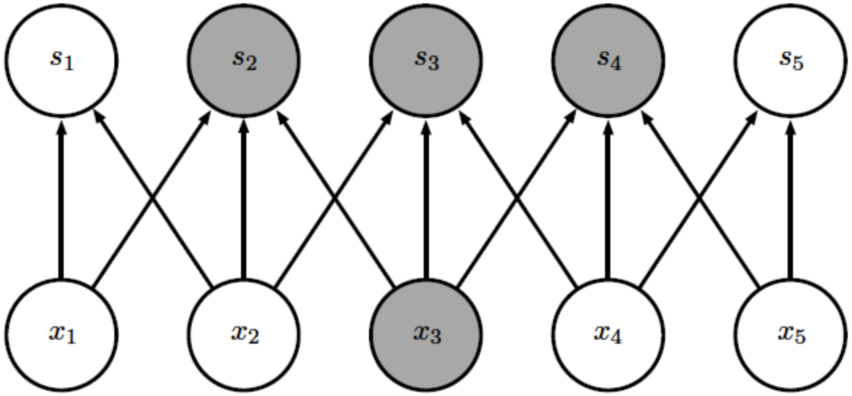
- Fully connected



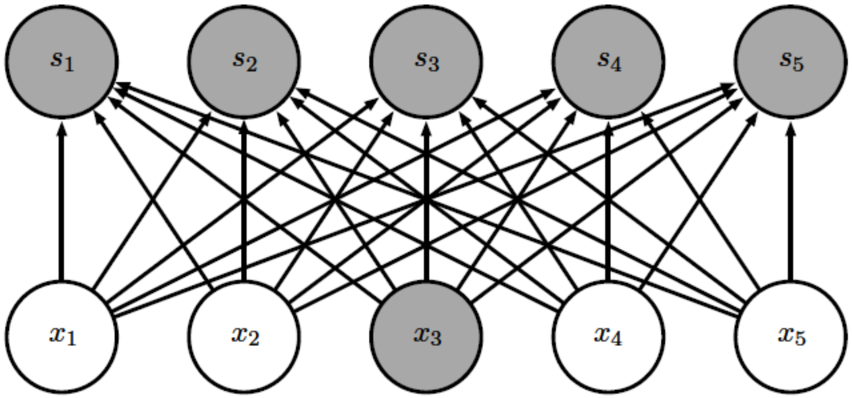
[Goodfellow]

Sparse Local Connectivity

- 1D convolution

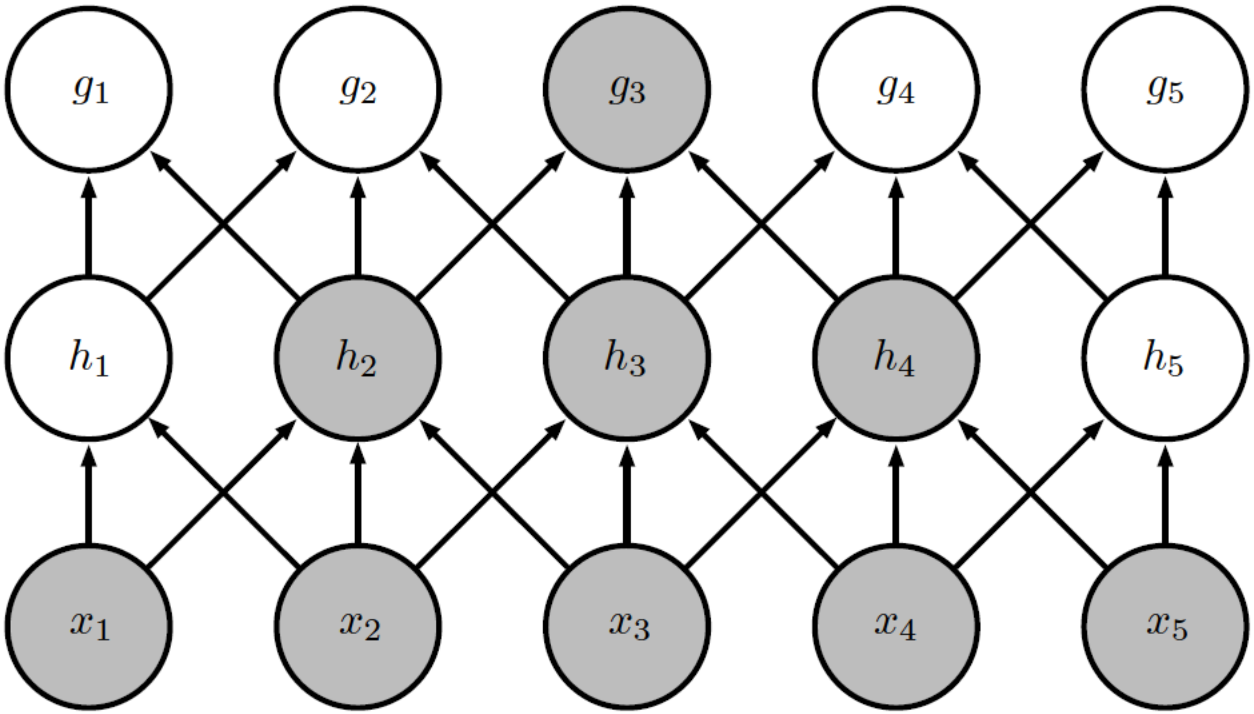


- Fully connected



[Goodfellow]

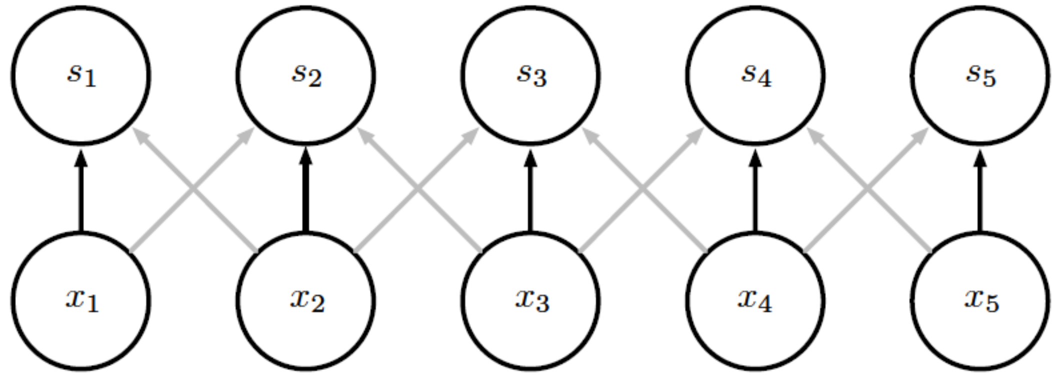
Growing Receptive Fields



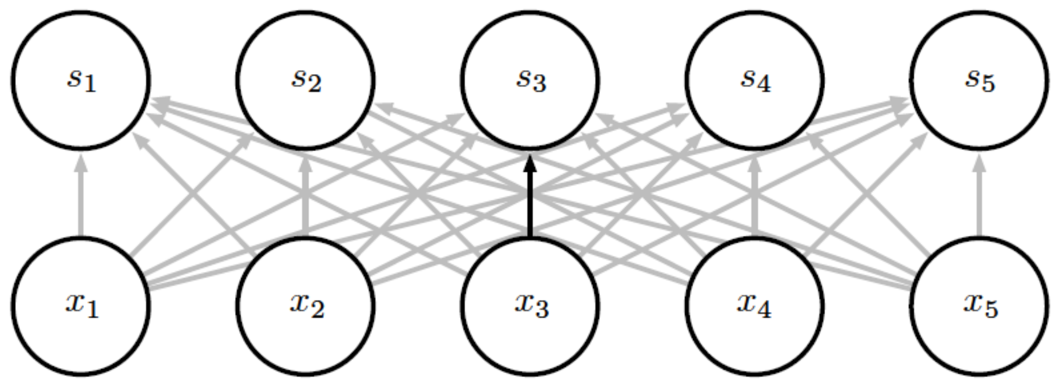
[Goodfellow]

Parameter Sharing

- Same weight used at all spatial locations



- No weight sharing

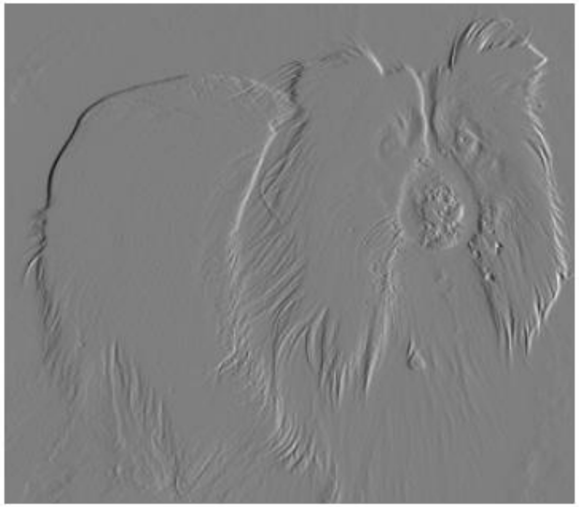


[Goodfellow]

Edge Detection by Convolution

Input

Output



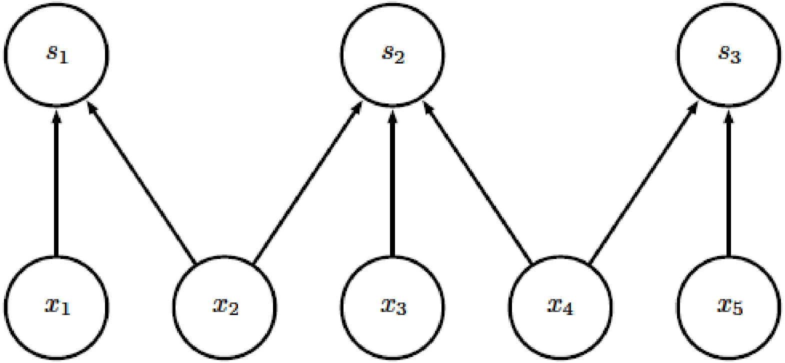
Kernel

1	-1

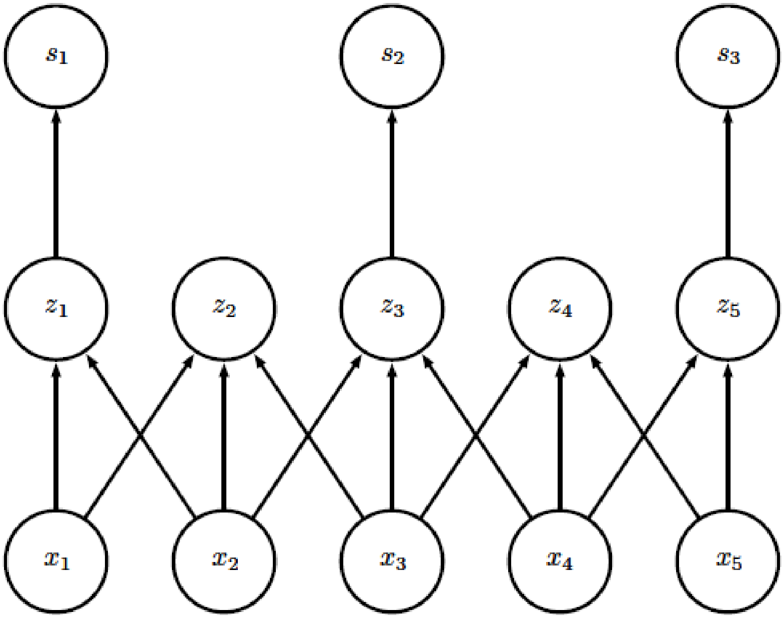
[Goodfellow]

Stride

- Strided convolution



- Convolution followed by subsampling

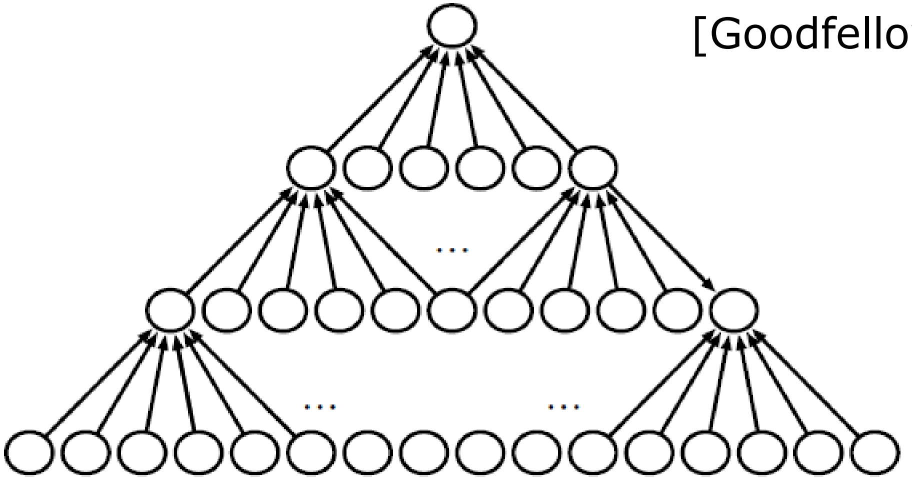


[Goodfellow]

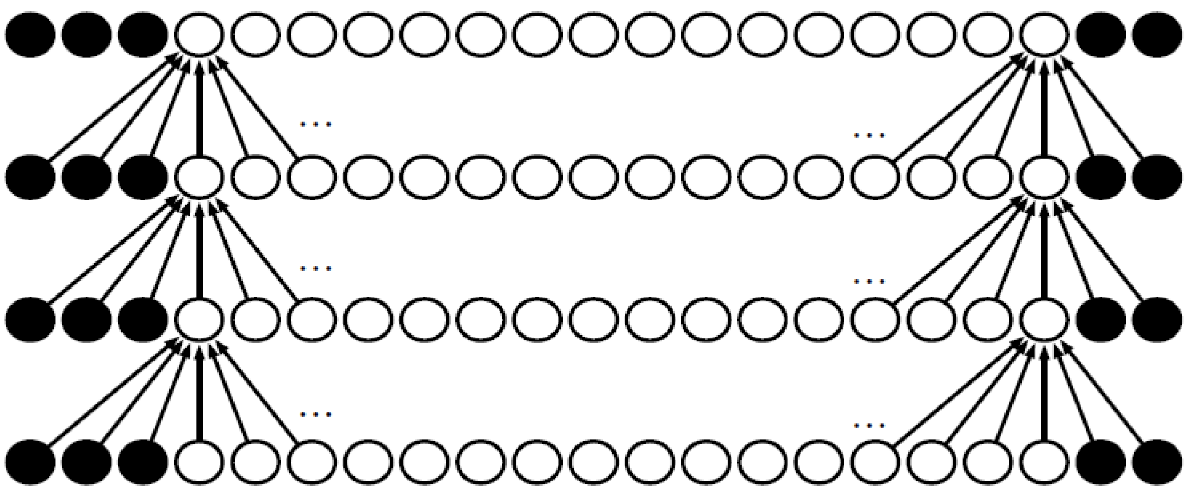
Border Padding

[Goodfellow]

- Valid convolutions reduce image size

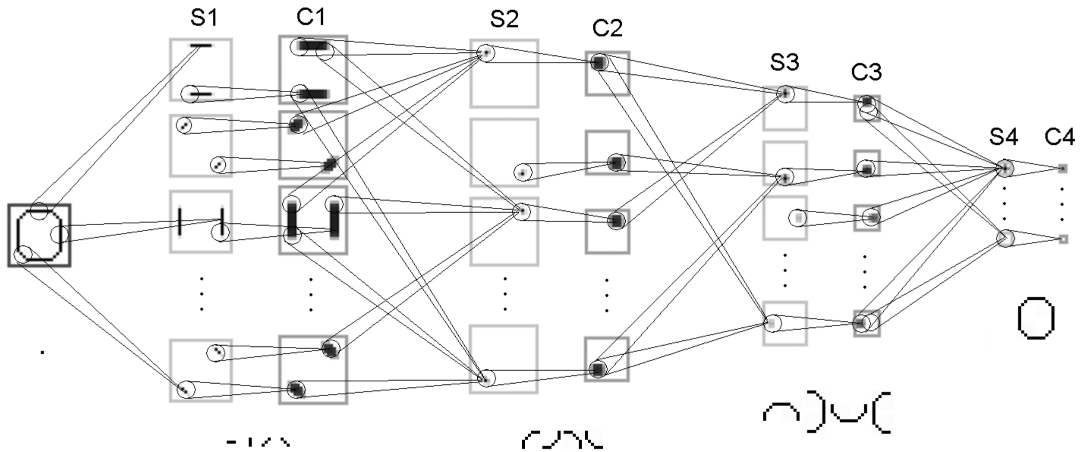


- Border padding maintains image size
  - Zero padding
  - Mirroring
  - Copying

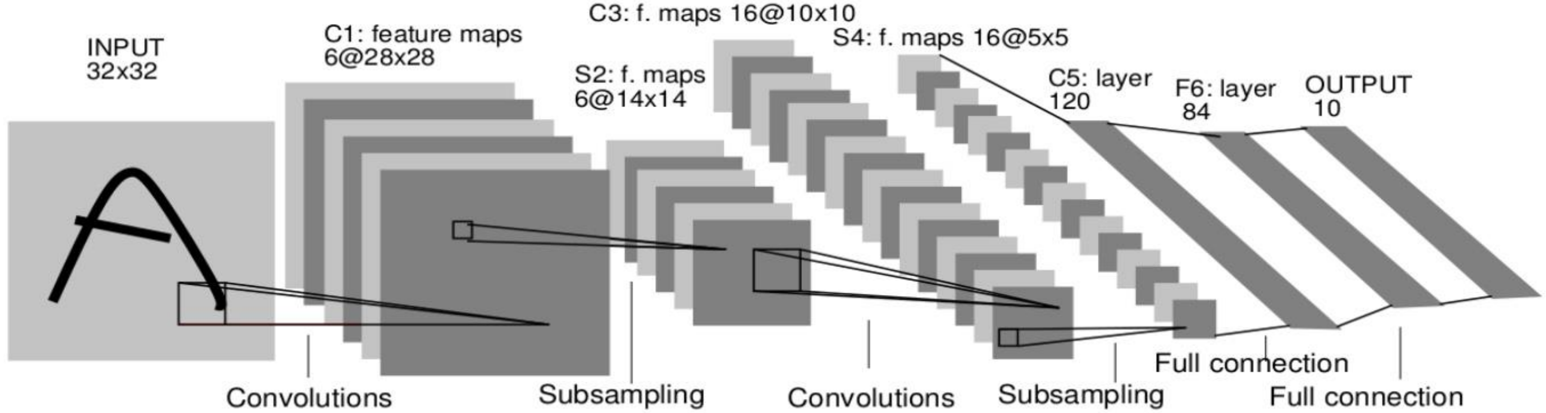


Convolutional Models

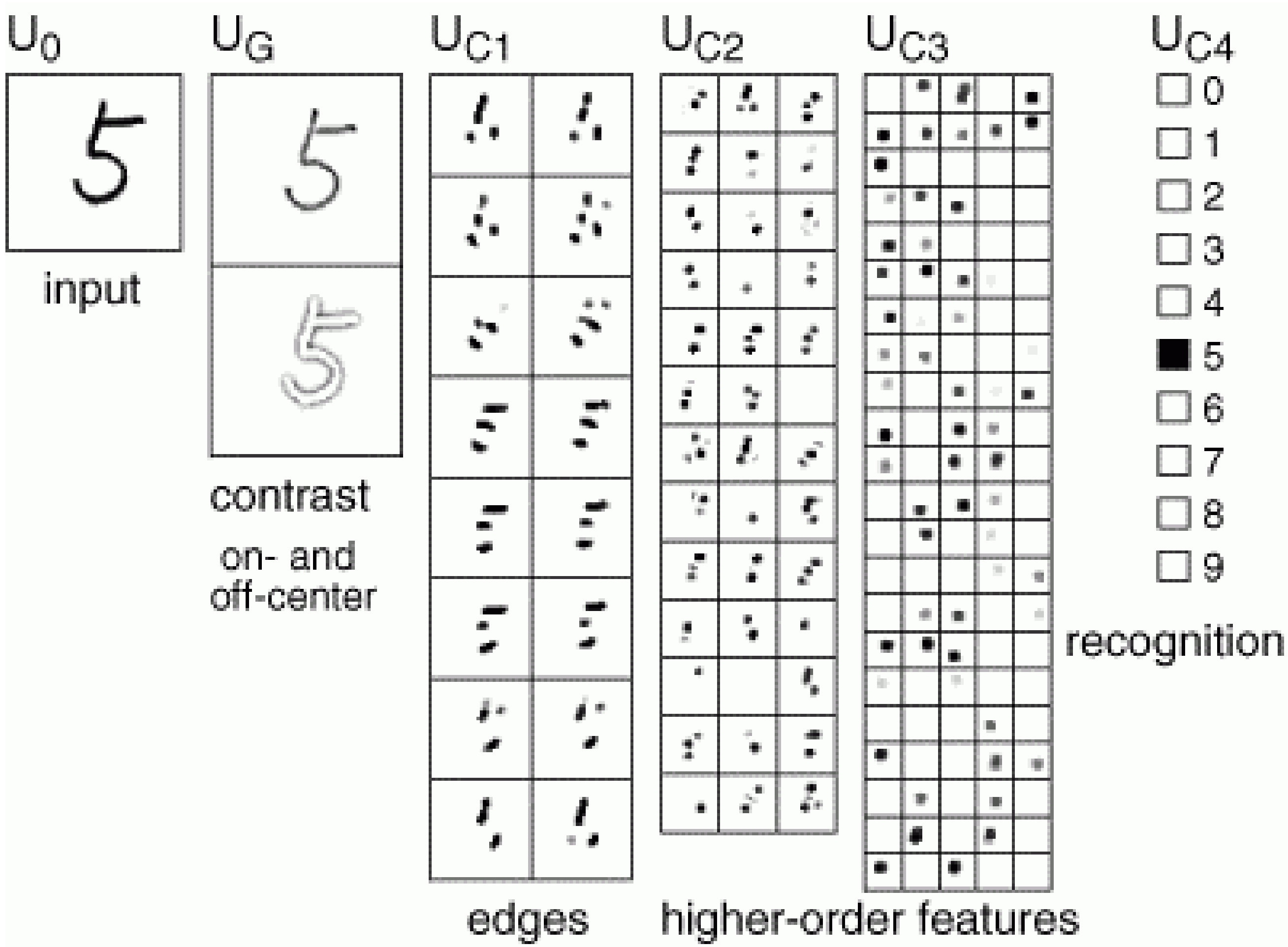
- Neocognitron: Fukushima 1980



- Supervised training of convolutional networks: LeCun 1989



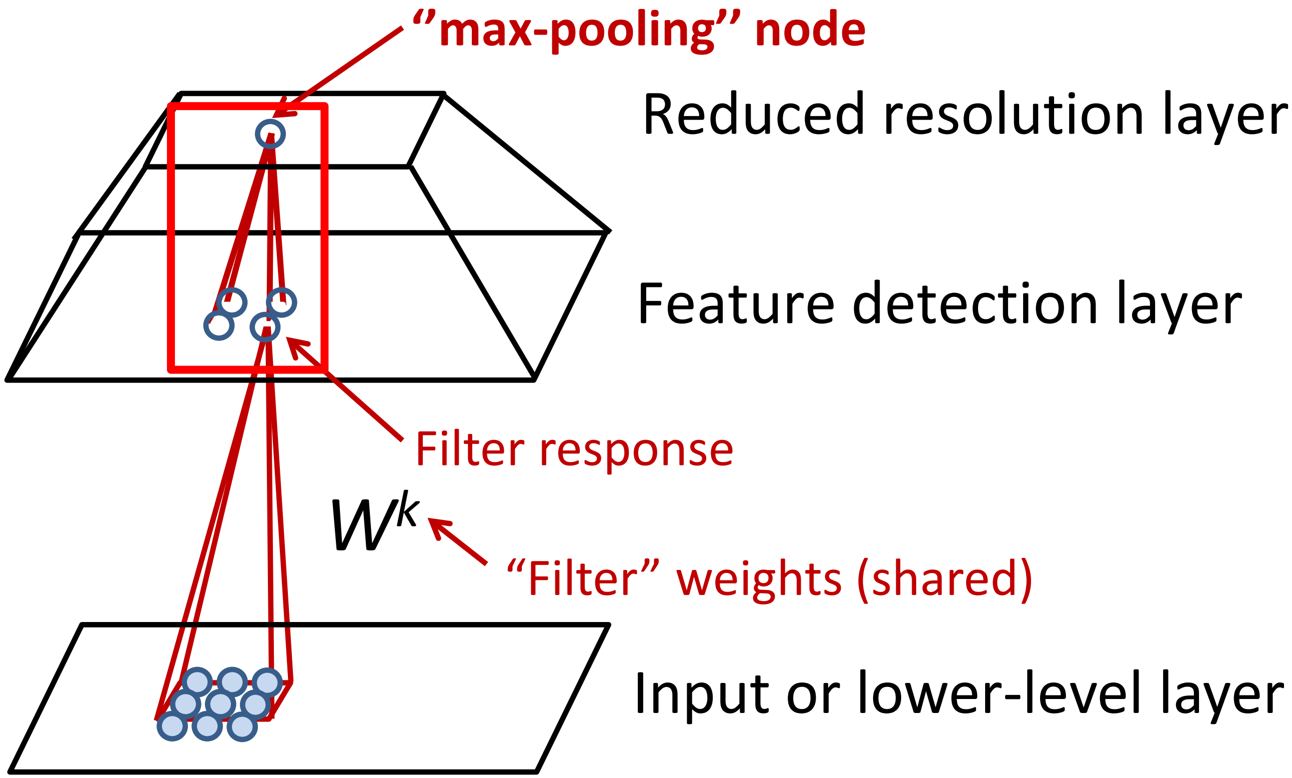
LeNet Character Recognition



[LeCun]

Max Pooling

- $o'_{i,j} = \max(o_{2i,2j}, o_{2i+1,2j}, o_{2i,2j+1}, o_{2i+1,2j+1})$



- Creates invariance to local shifts

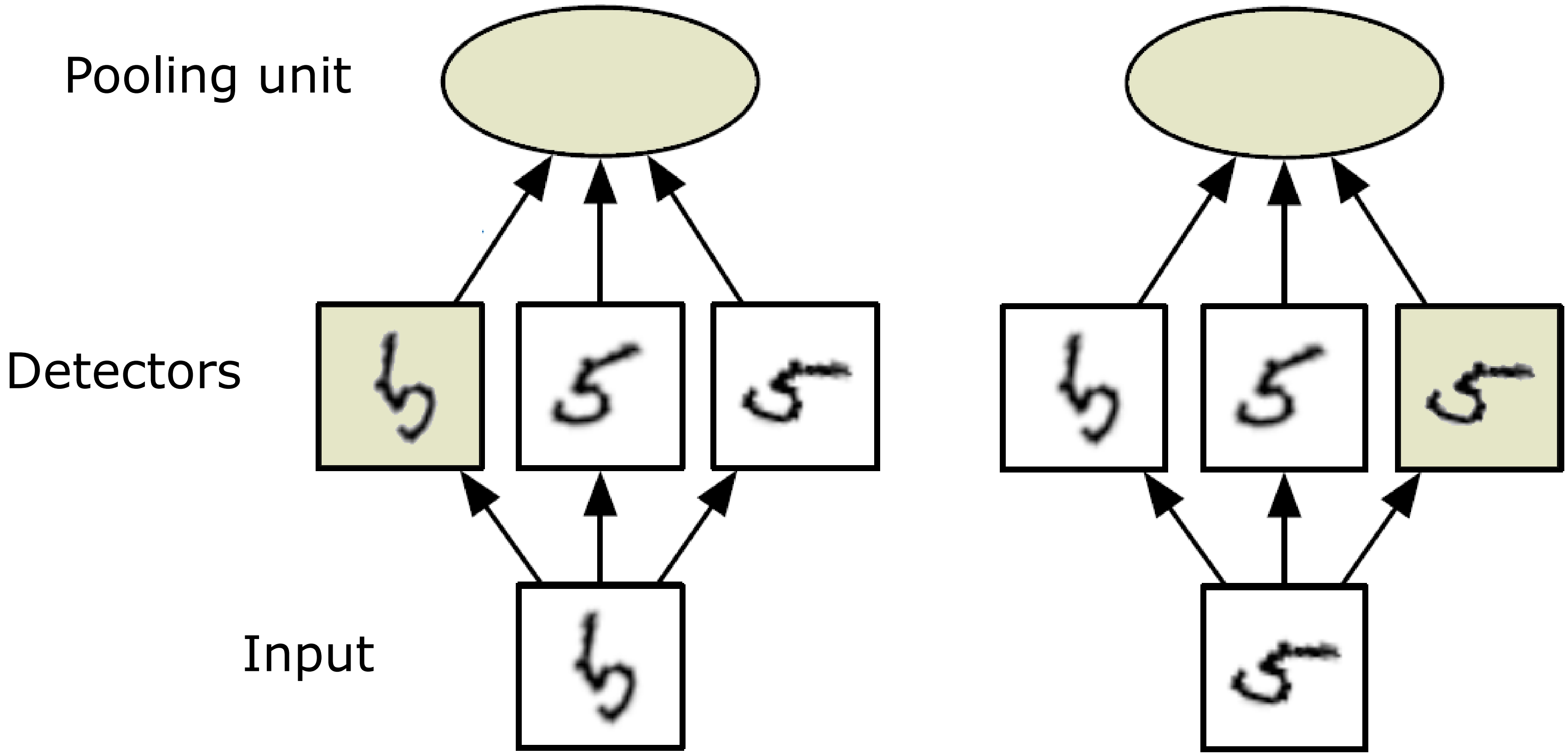
1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

max pool with 2x2 filters and stride 2 →

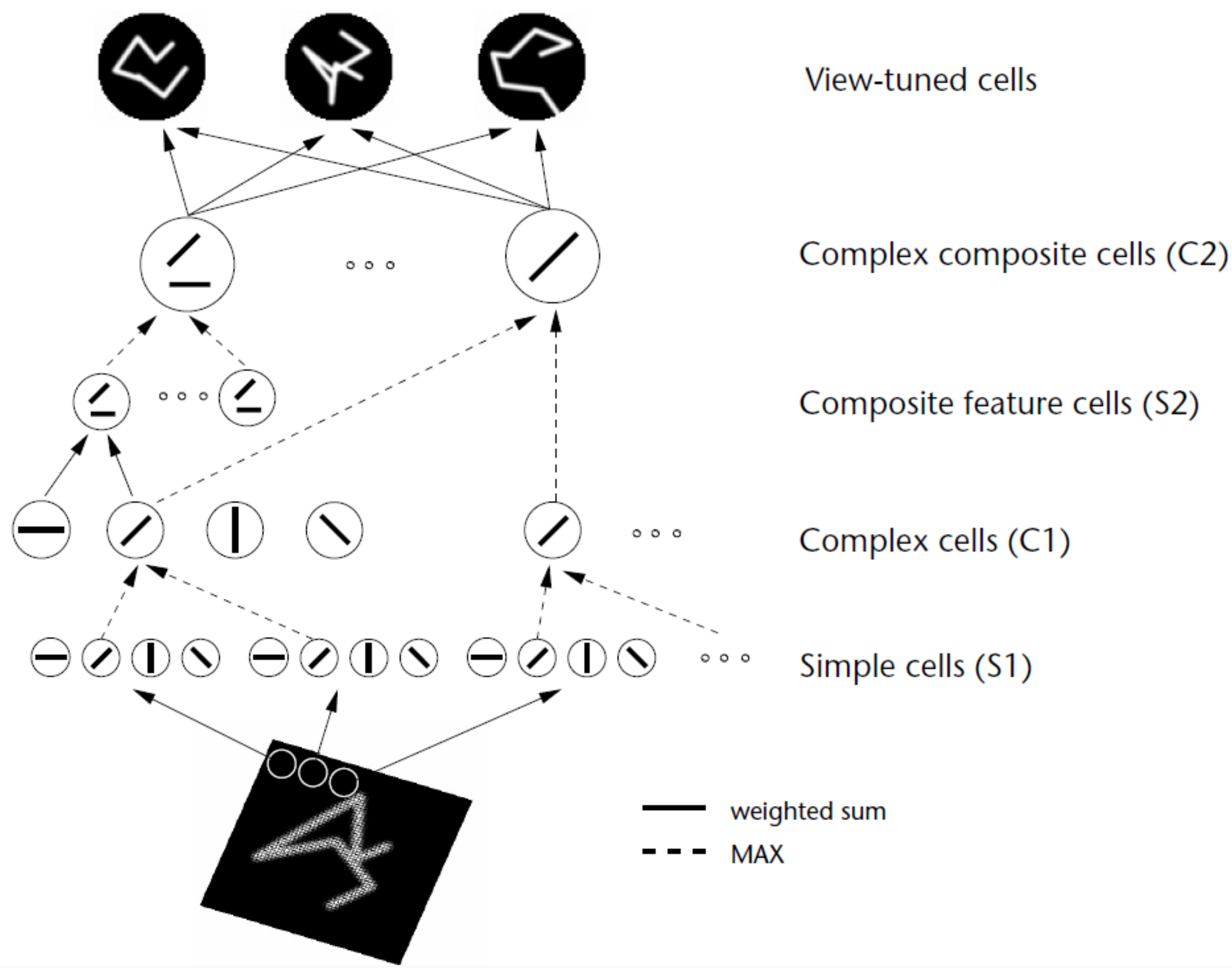
6	8
3	4

Cross-Channel Pooling

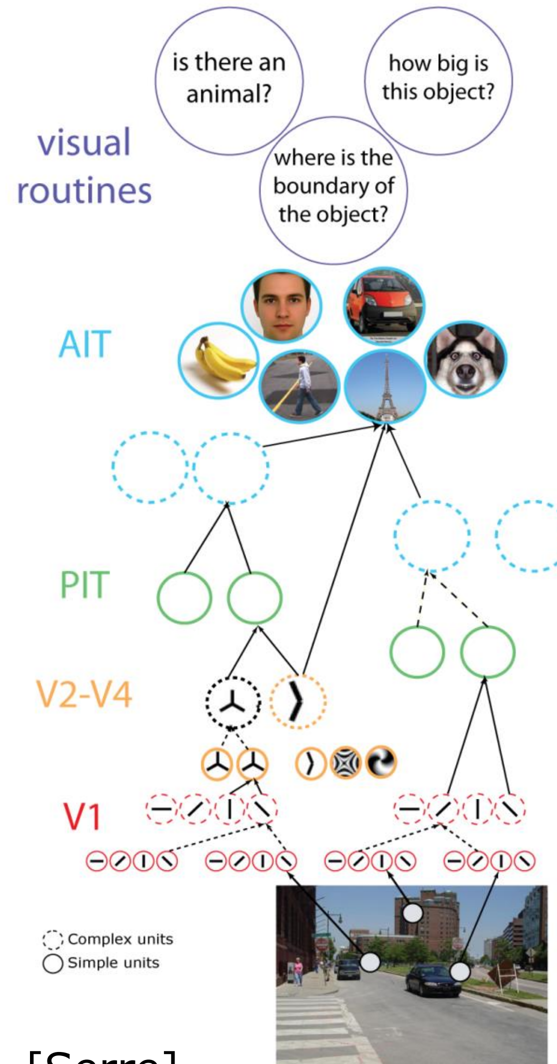
- Creates invariance to learned transformations



HMAX Model



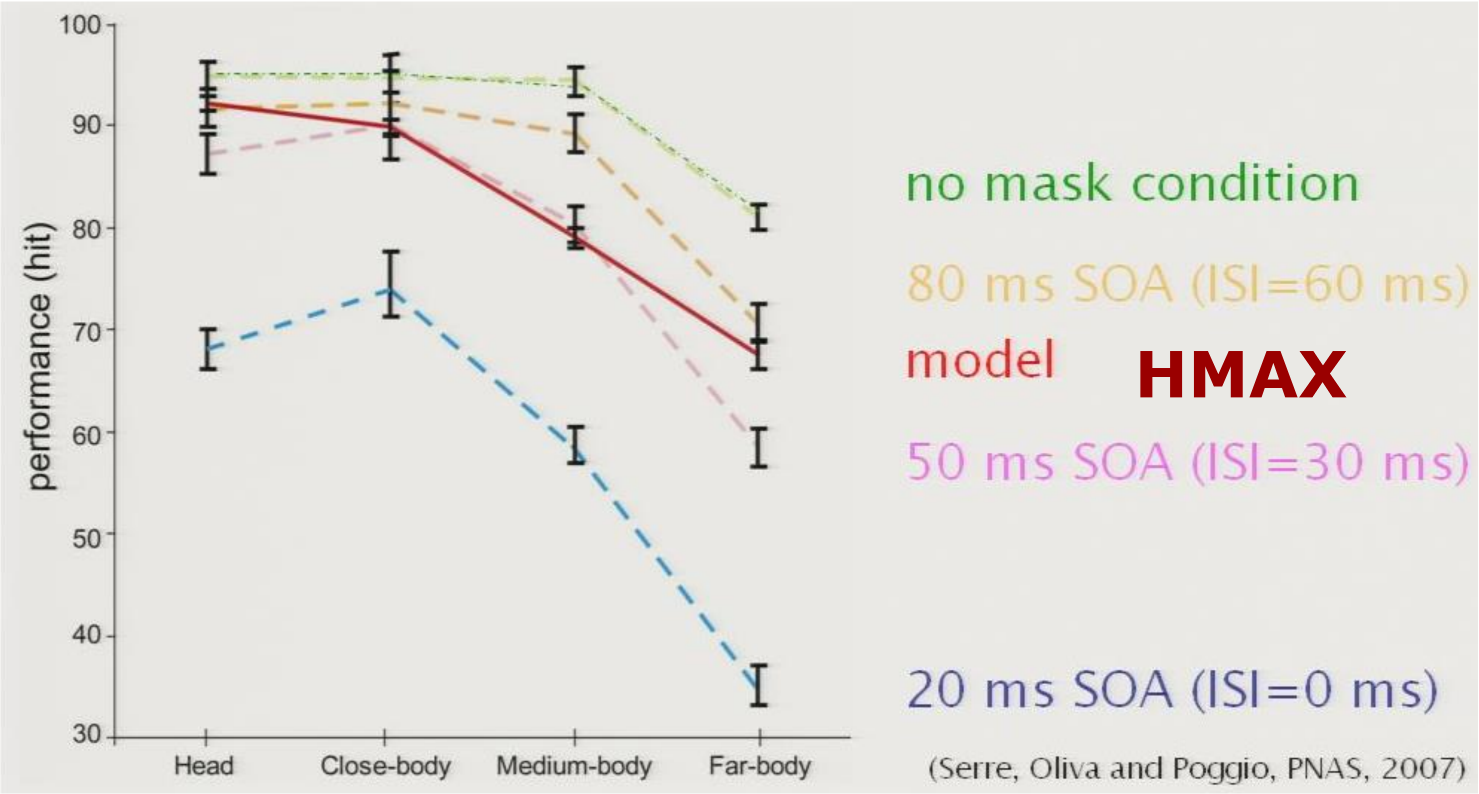
[Riesenhuber and Poggio 1999]



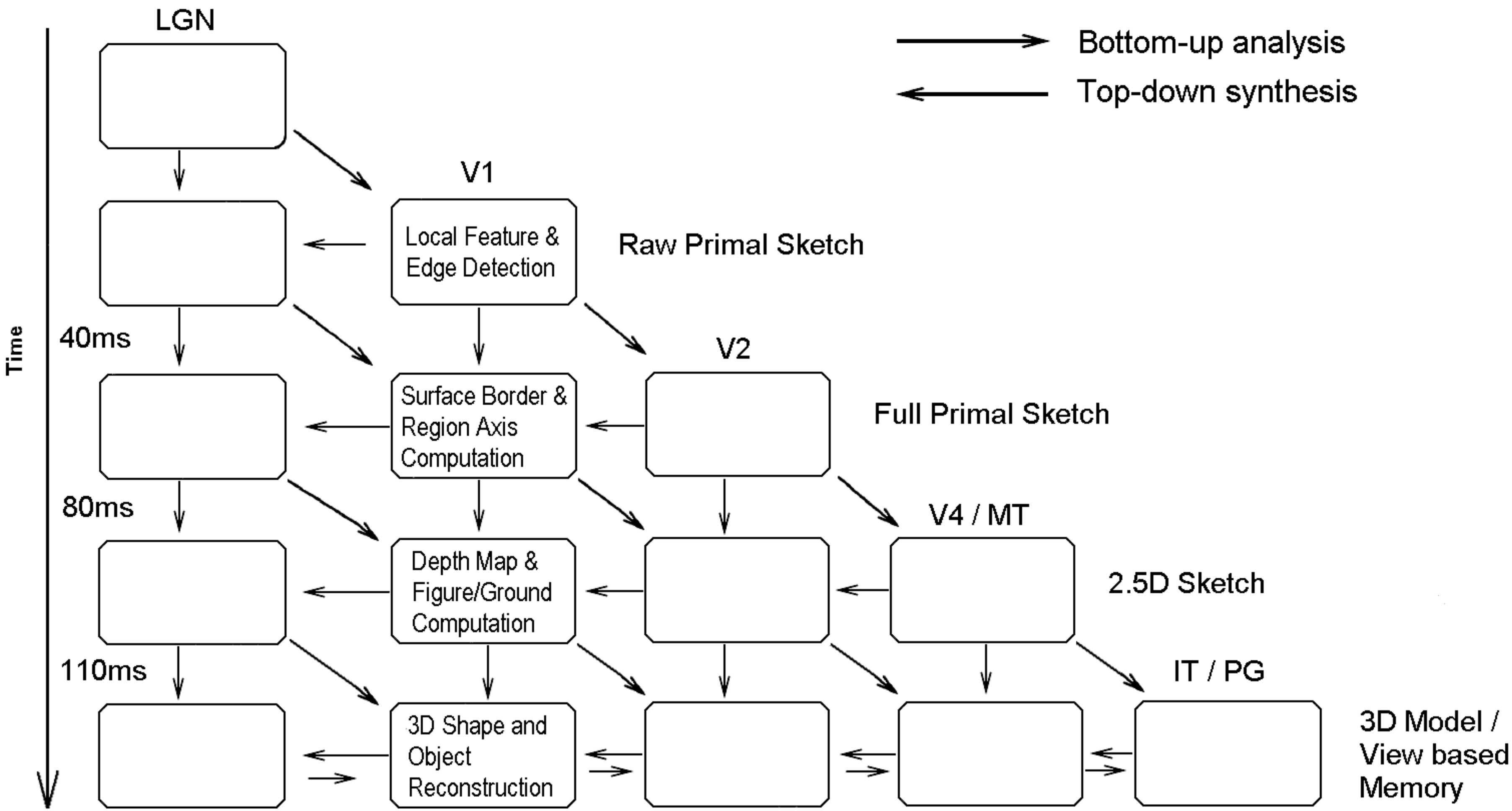
[Serre]

Feed-forward Models Cannot Explain Human Performance

- Performance increases with observation time

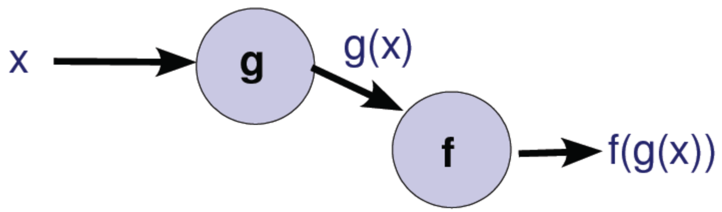


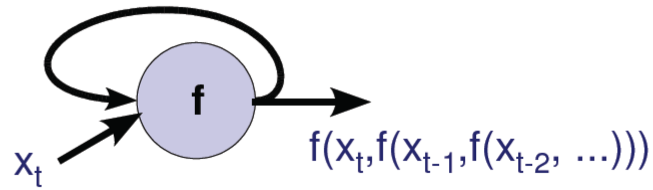
Bottom-up, Lateral, and Top-down Processing



[Lee et al. 1998]

Feed Forward vs. Recurrent



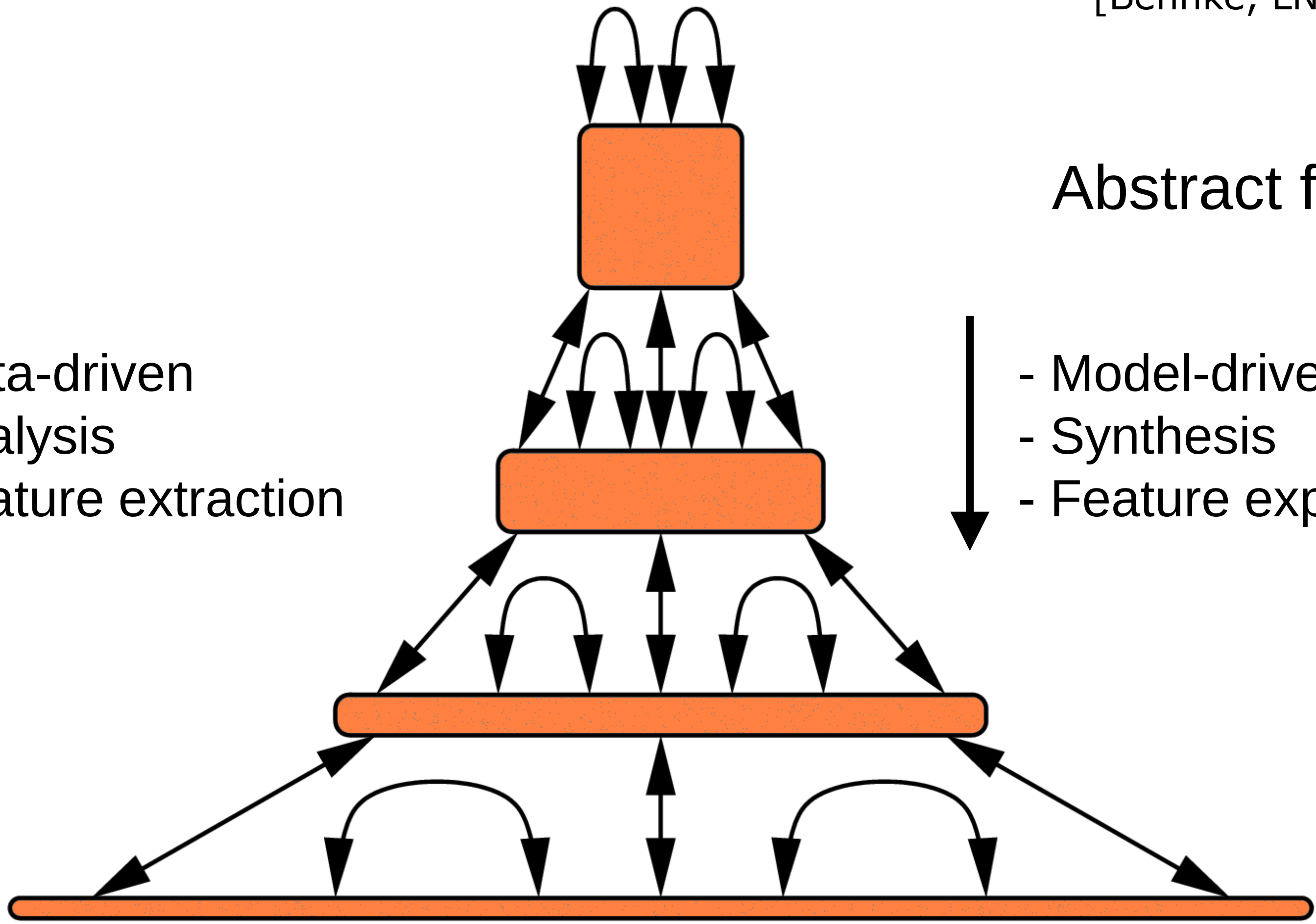


- Connectivity without cycles
- Composition of simple functions
- A node can only by computed if its inputs are available
- Reuse of partial results
- Order of computation determined by directed connectivity

- Connectivity with cycles
- Explicit modeling of computation time necessary
- Computation needs one unit of time
- Input at time t yields output at time t+1
- Order of computation not any longer determined by connectivity

Neural Abstraction Pyramid

[Behnke, LNCS 2766, 2003]



Abstract features

- Data-driven
- Analysis
- Feature extraction

- Model-driven
- Synthesis
- Feature expansion

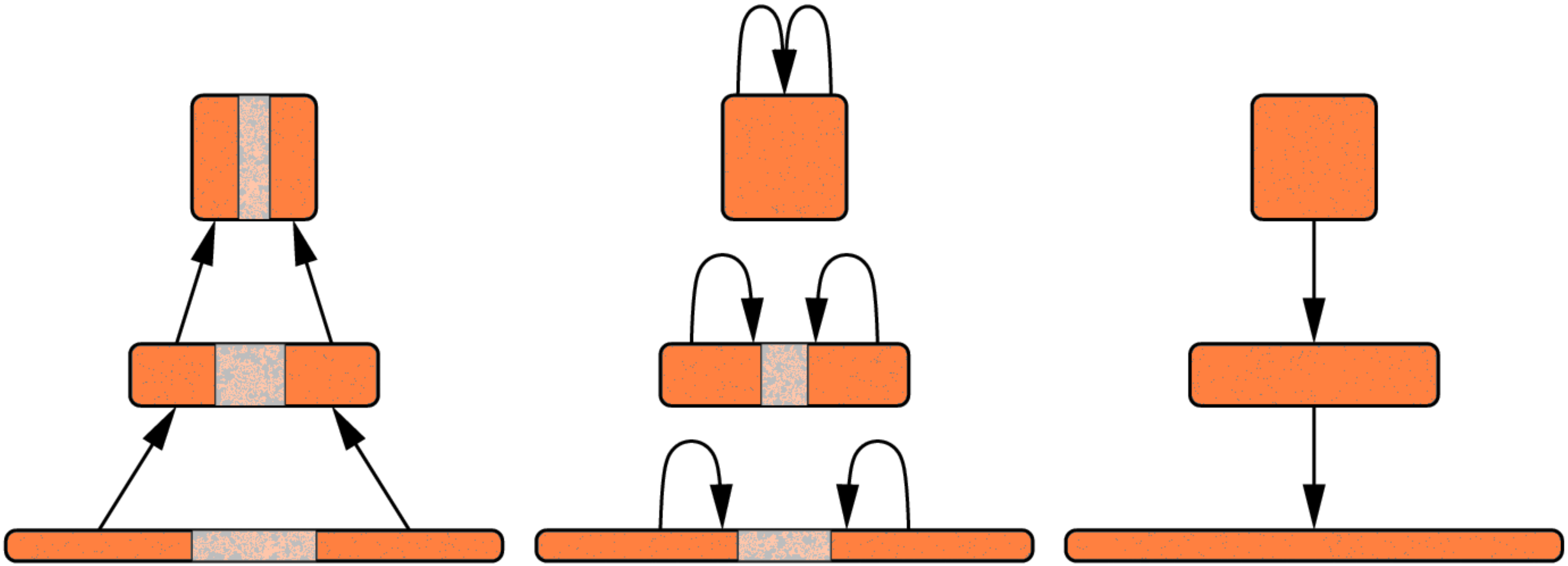
Signals

- Grouping - Competition - Completion

Iterative Interpretation

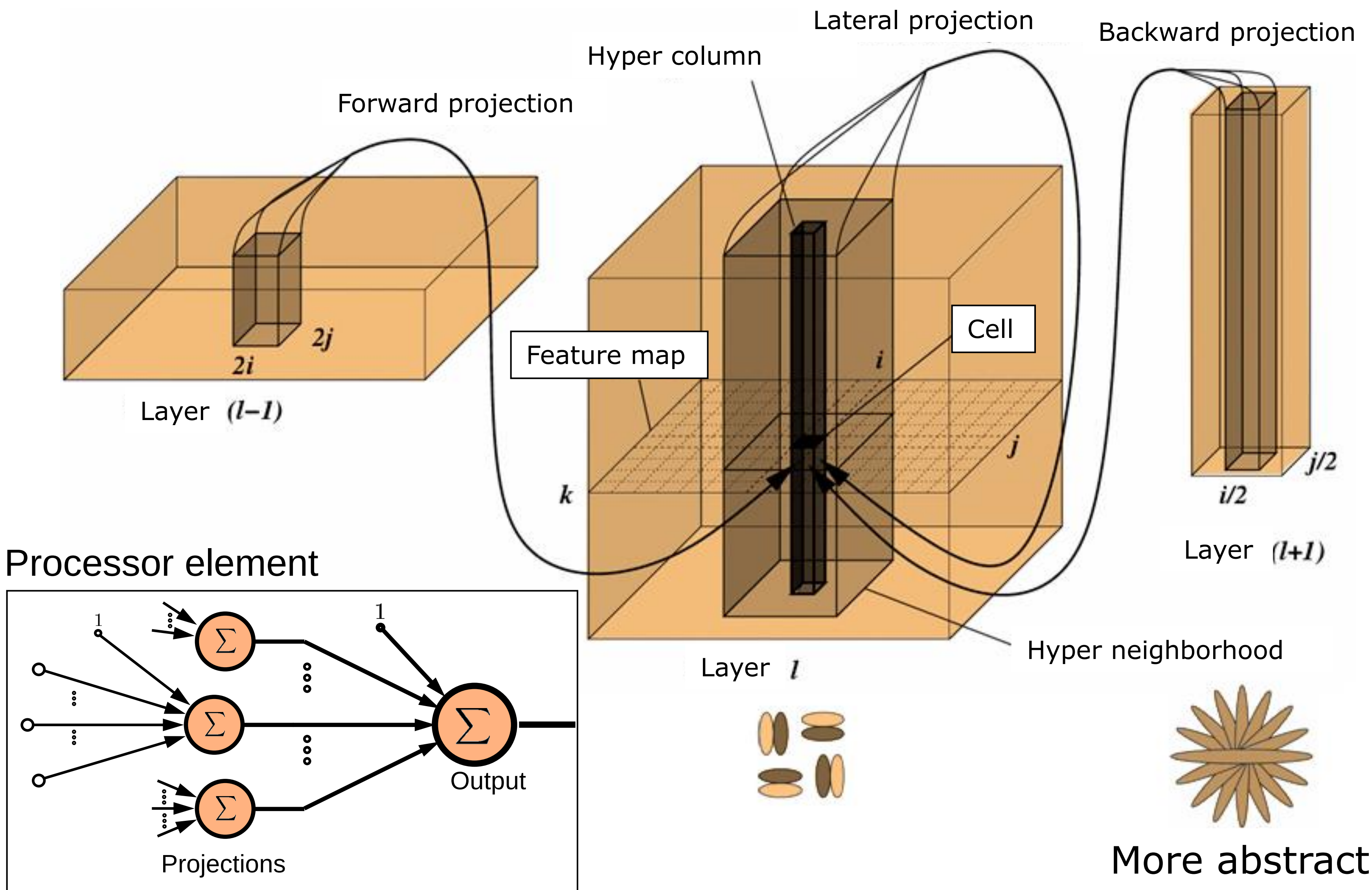
[Behnke, LNCS 2766, 2003]

- Interpret most obvious parts first



- Use partial interpretation as context to resolve local ambiguities

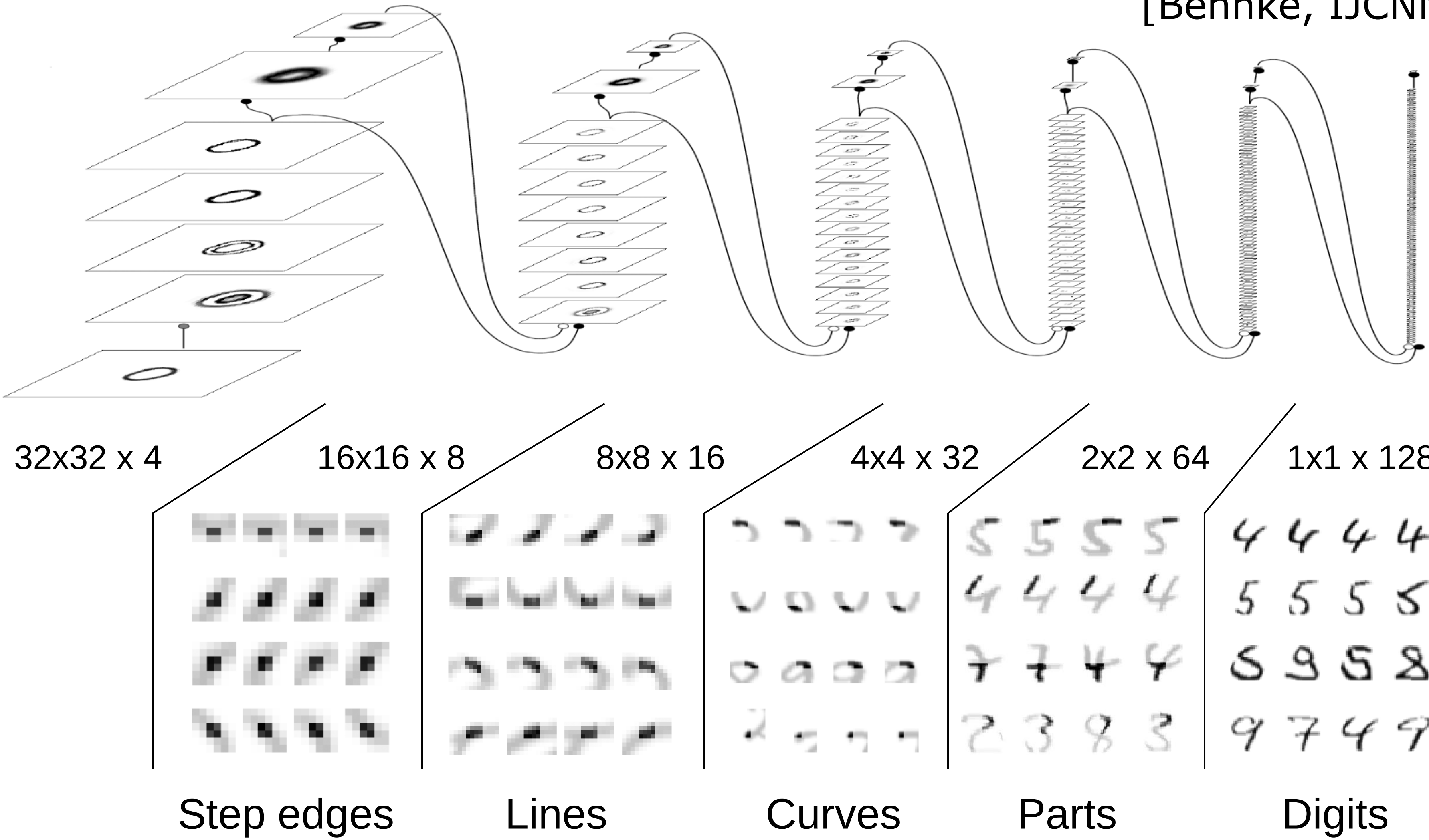
Local Recurrent Connectivity



[Behnke, LNCS 2766, 2003]

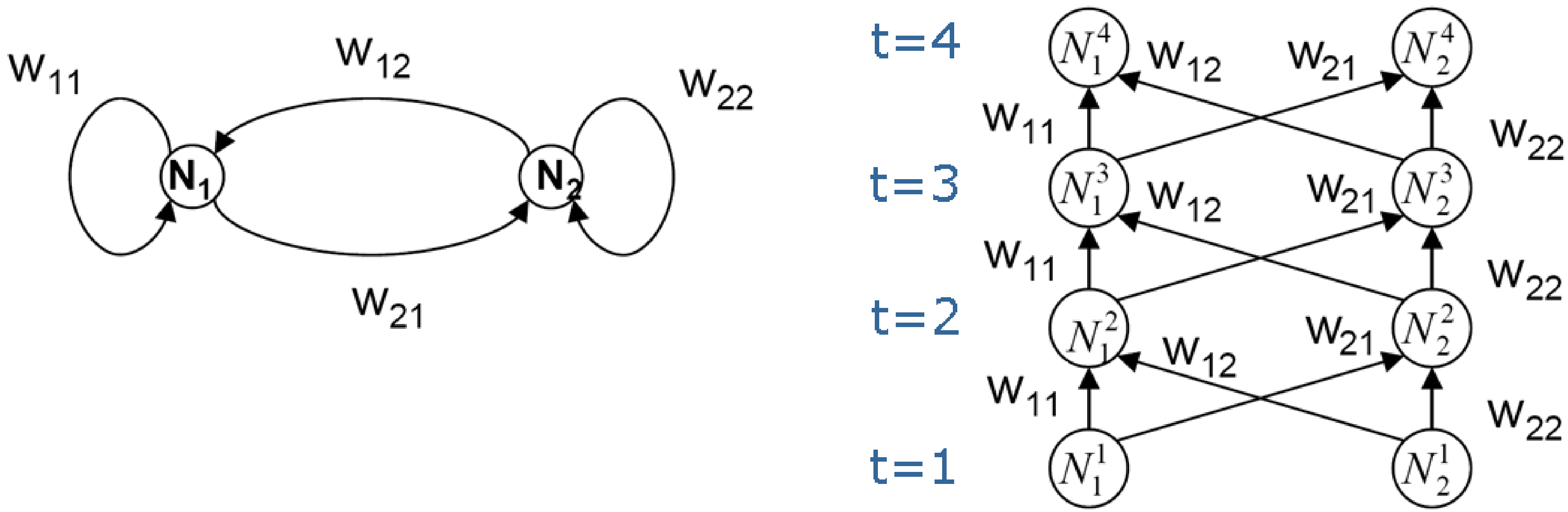
Unsupervised Learning of a Feature Hierarchy

[Behnke, IJCNN'99]



Backpropagation Through Time (BPTT)

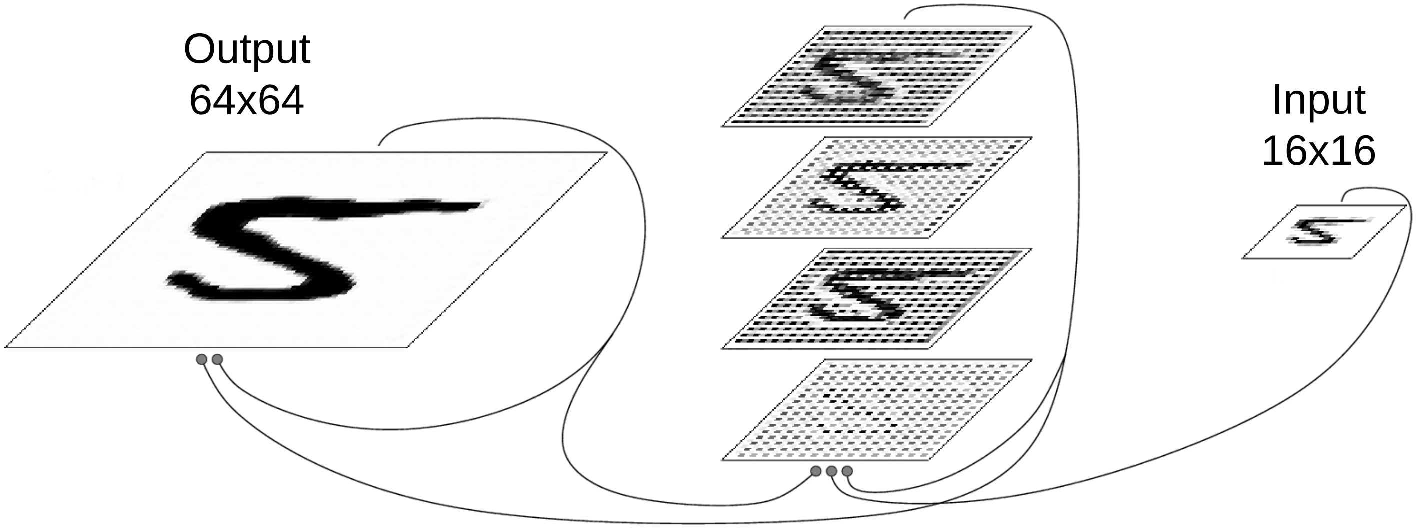
- Unfolding along time axis -> deep network
- Weight-sharing -> Average updates

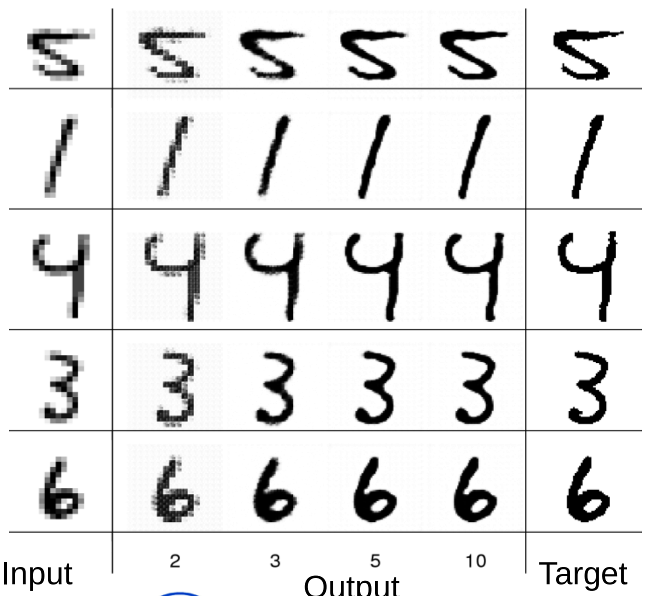


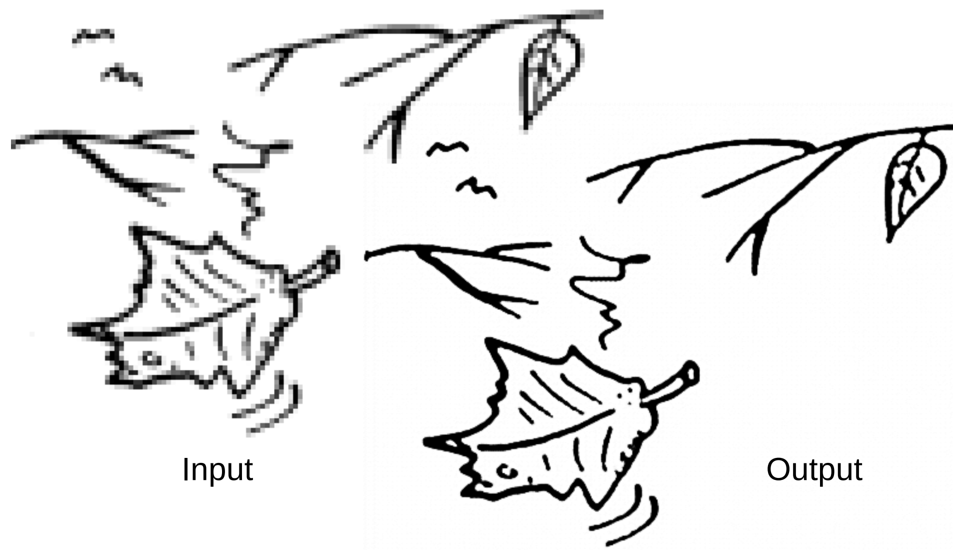
$$o_i\,(t+1) = f\,(net_i(t)) = f\Big(\ \sum_j w_{ij} o_j(t) + x_i(t)\ \Big)$$

Superresolution

[Behnke, IJCAI'01]

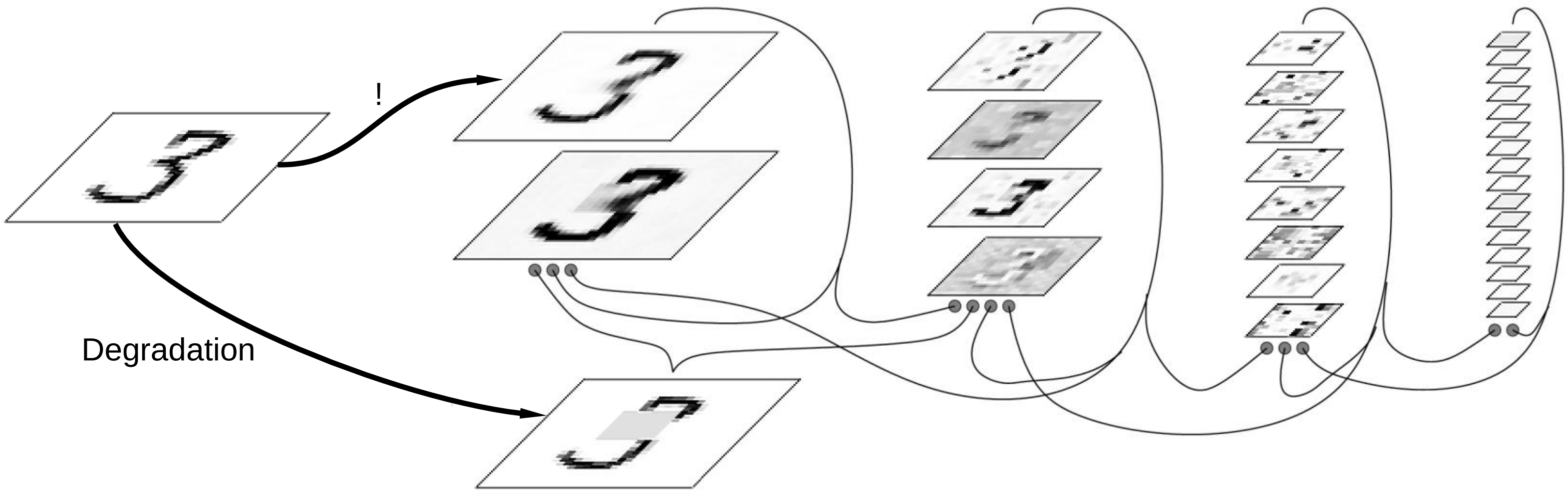


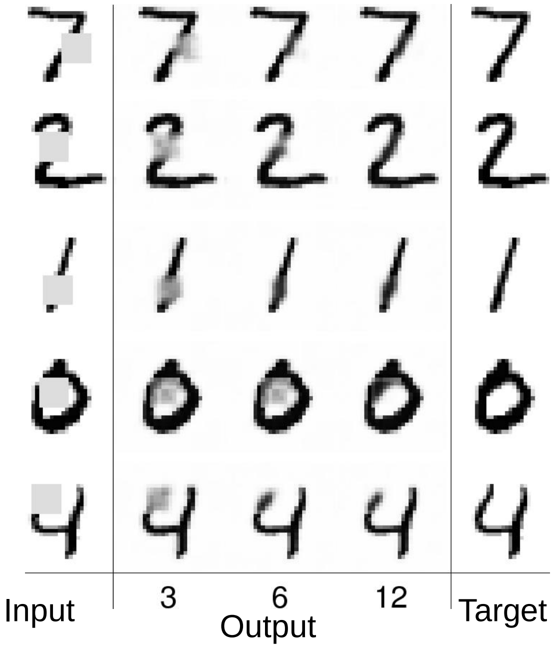


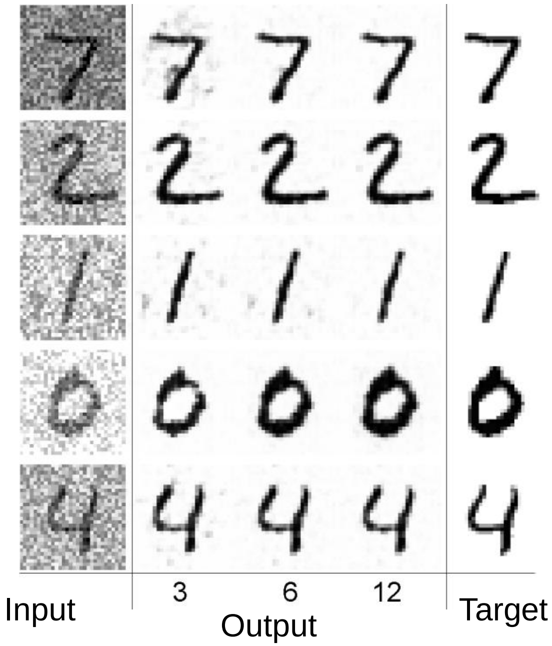


Digit Reconstruction

[Behnke, IJCAI'01]

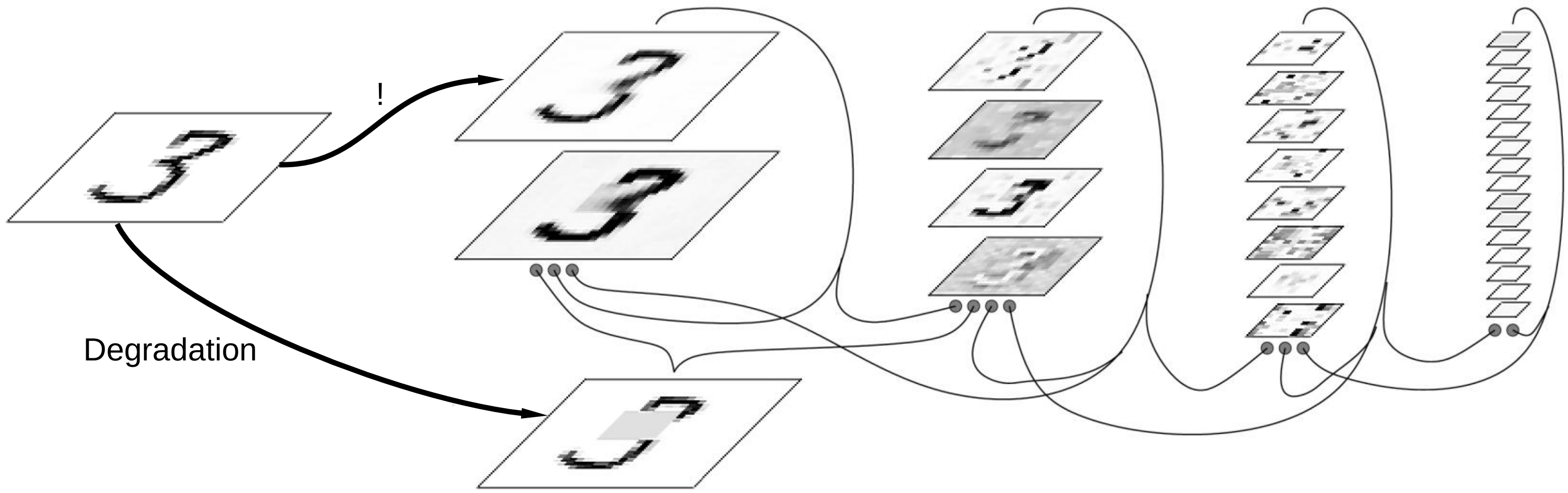


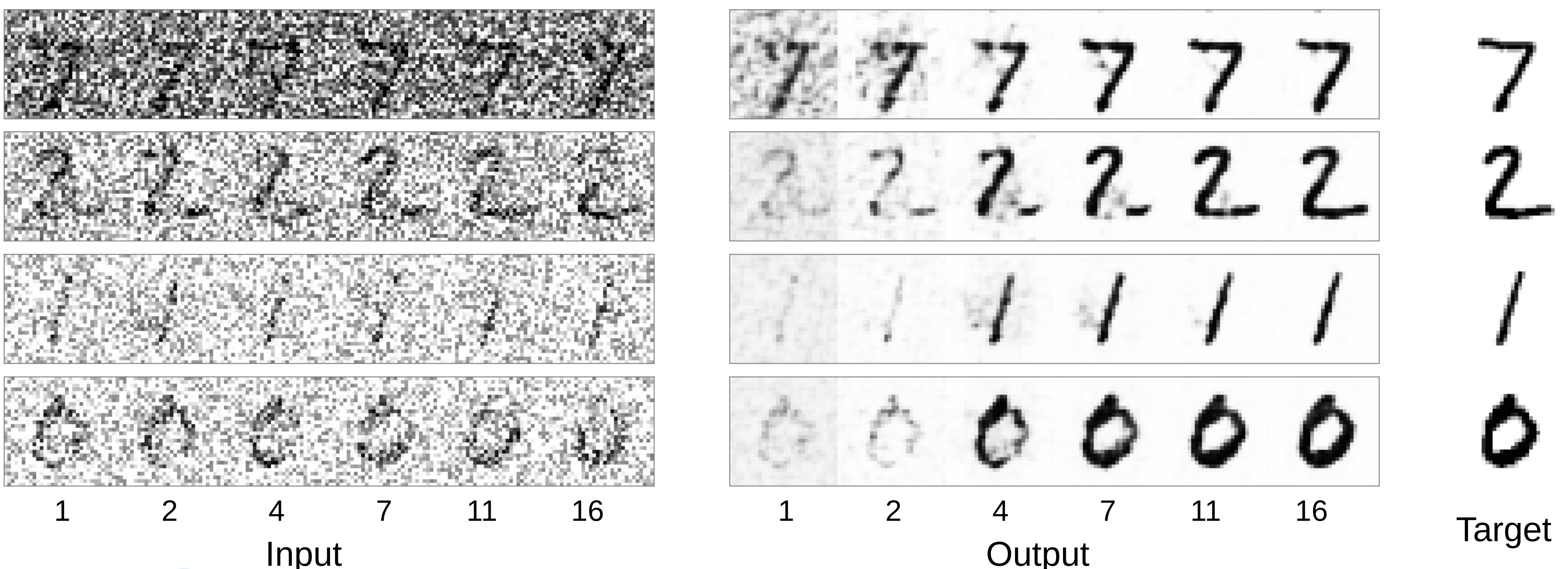




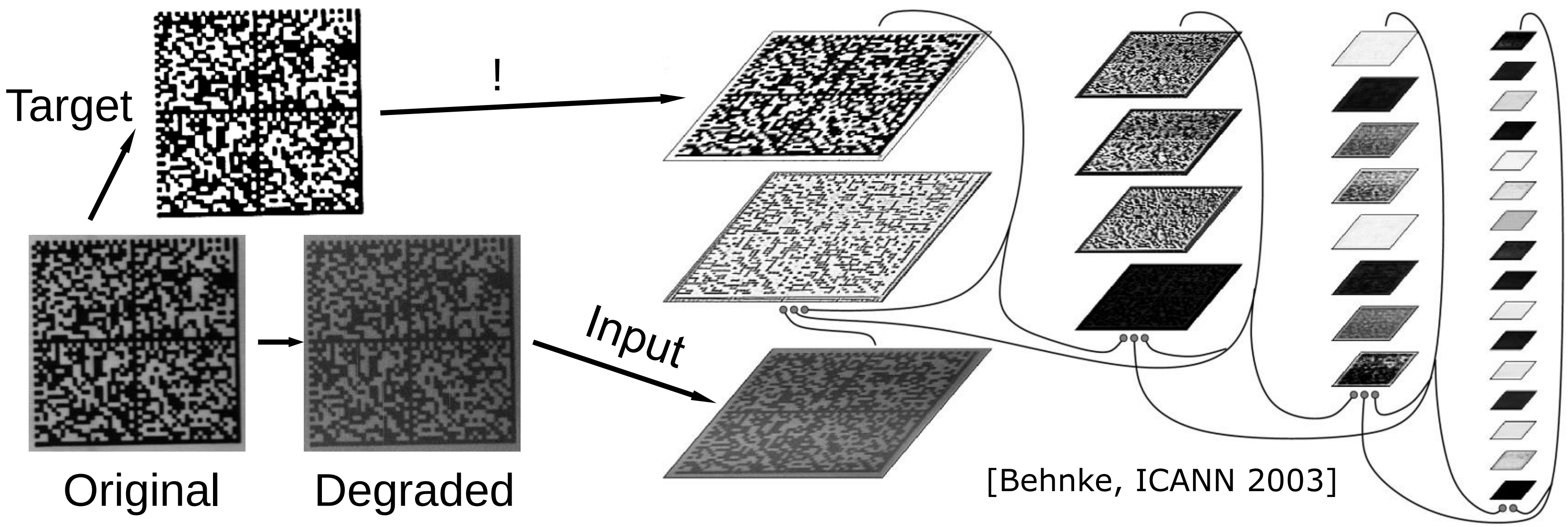
Digit Reconstruction

[Behnke, IJCAI'01]





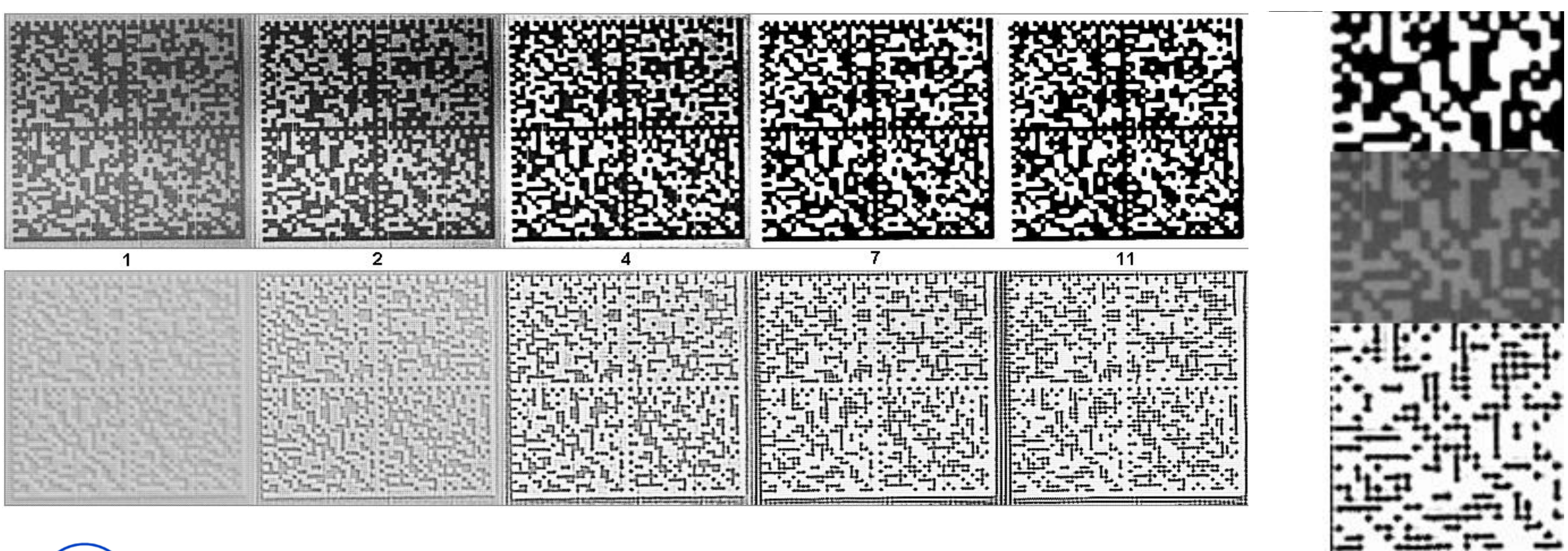
Binarization of Matrix Codes



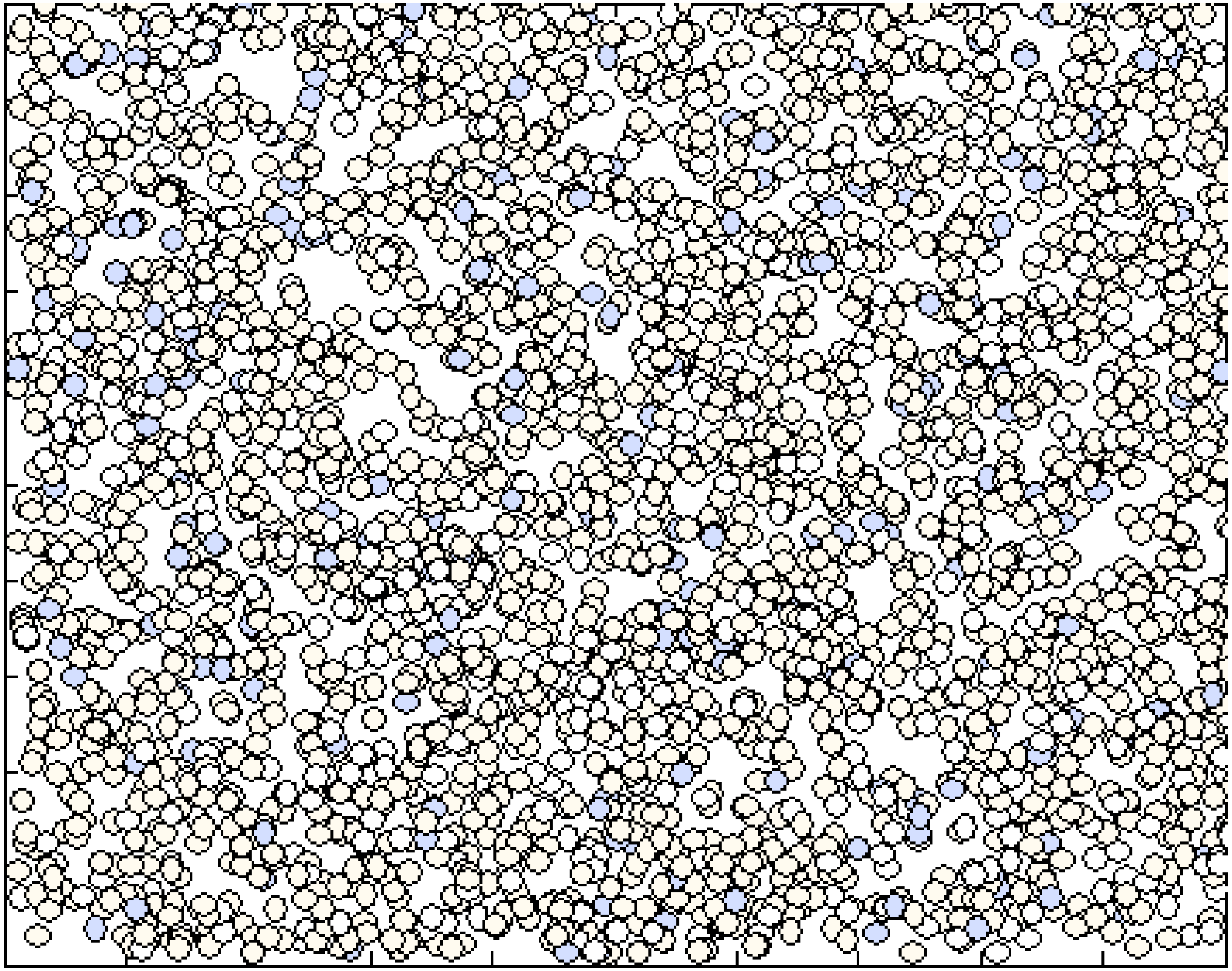
[Behnke, ICANN 2003]

Output

Hidden



Continous Attractor



- Local excitation and global inhibition
- Stable activity blobs can be shifted

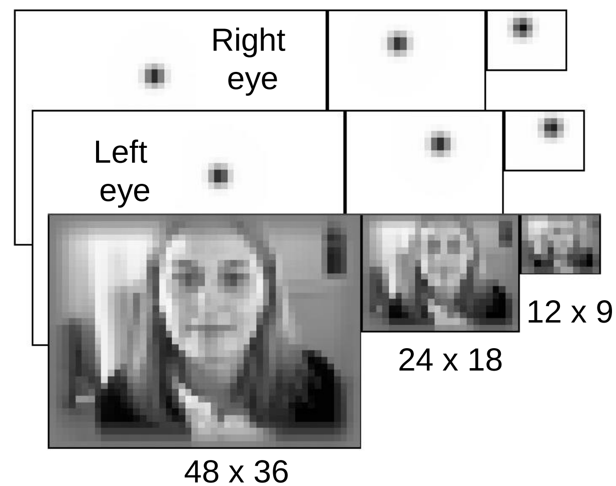
Face Localization

[Behnke, KES'03]

- BioID data set:
  - 1521 images
  - 23 persons

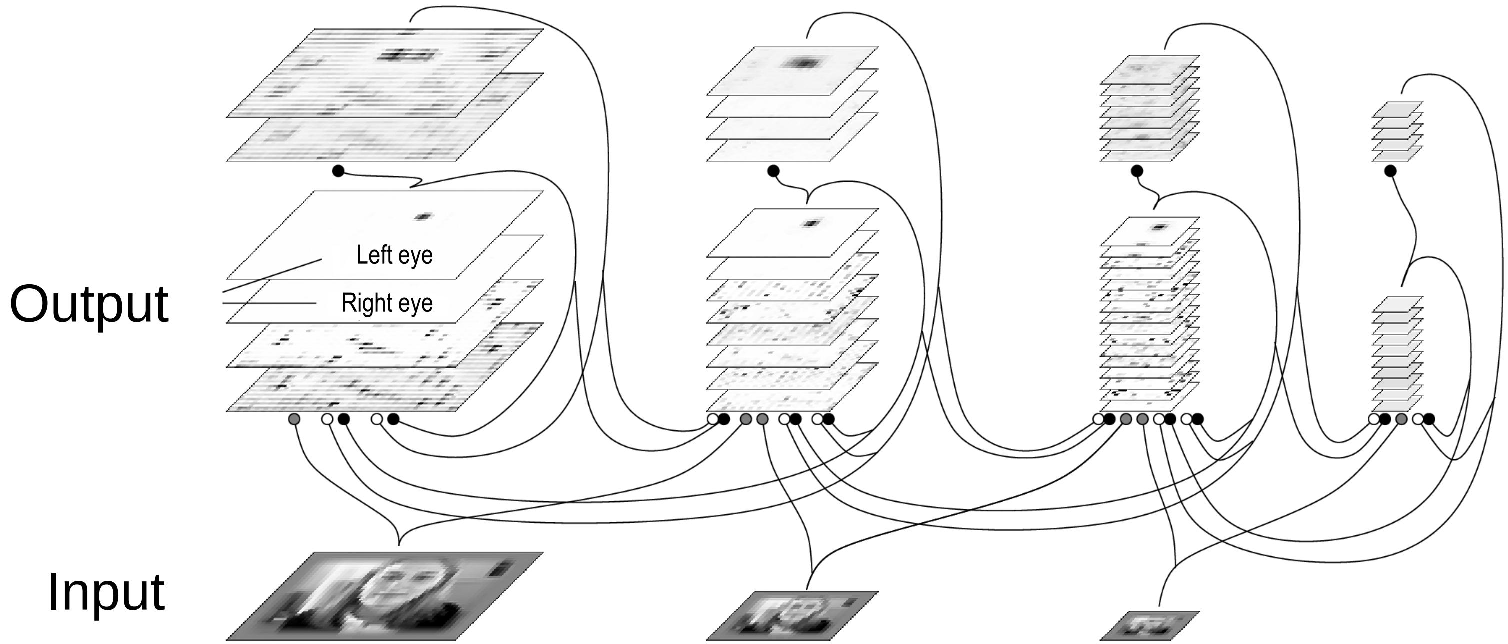
- Encode eye positions with blobs

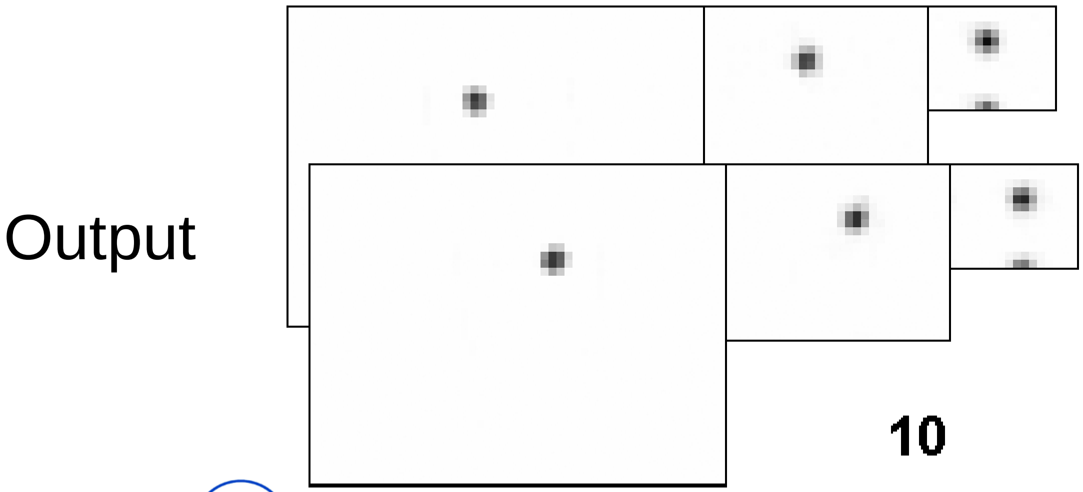
384 x 288



Face Localization

[Behnke, KES'03]



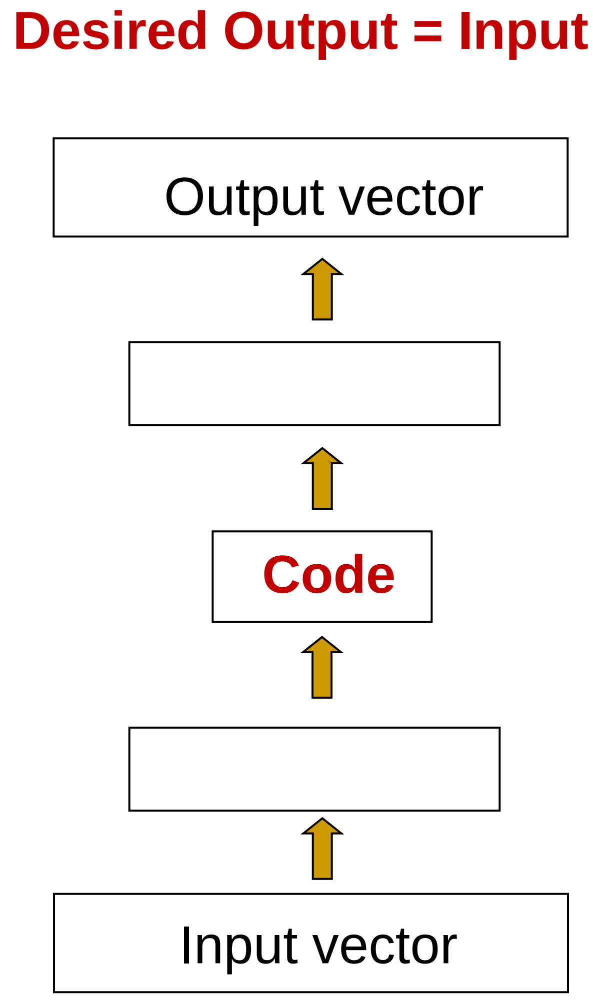


Auto-Encoder

- Try to push input through a bottleneck

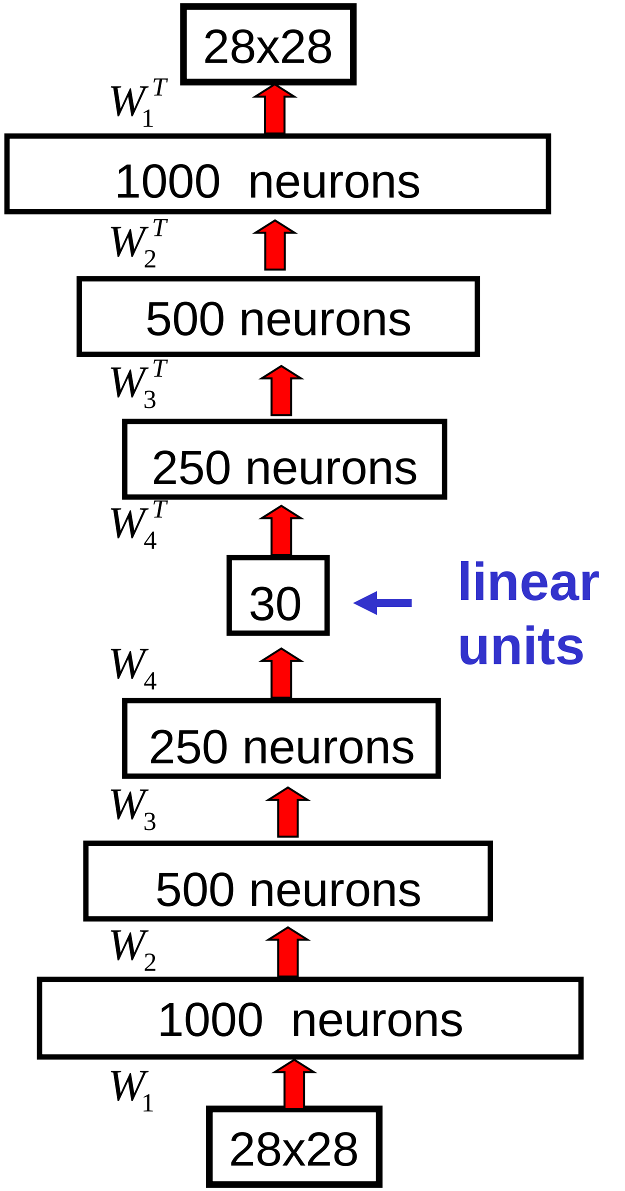
- Activities of hidden units form an efficient code
  - There is no space for redundancy in the bottleneck

- Extracts frequently independent features (factorial code)



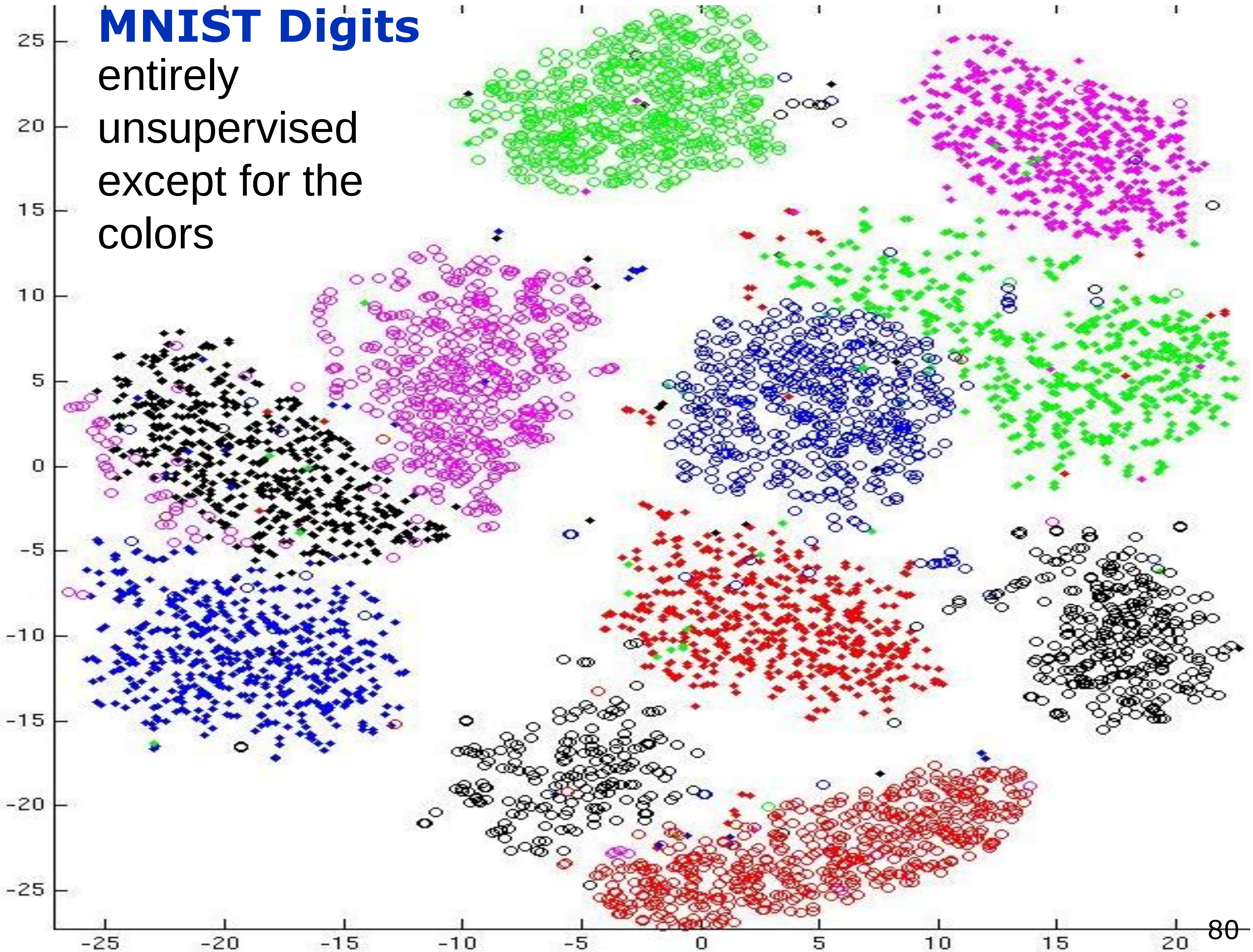
Deep Autoencoders
(Hinton & Salakhutdinov, 2006)

- Multi-layer autoencoders for non-linear dimensionality reduction
- Difficult to optimize deep autoencoders using backpropagation
- Greedy, layer wise training
- Unrolling
- Supervised fine-tuning



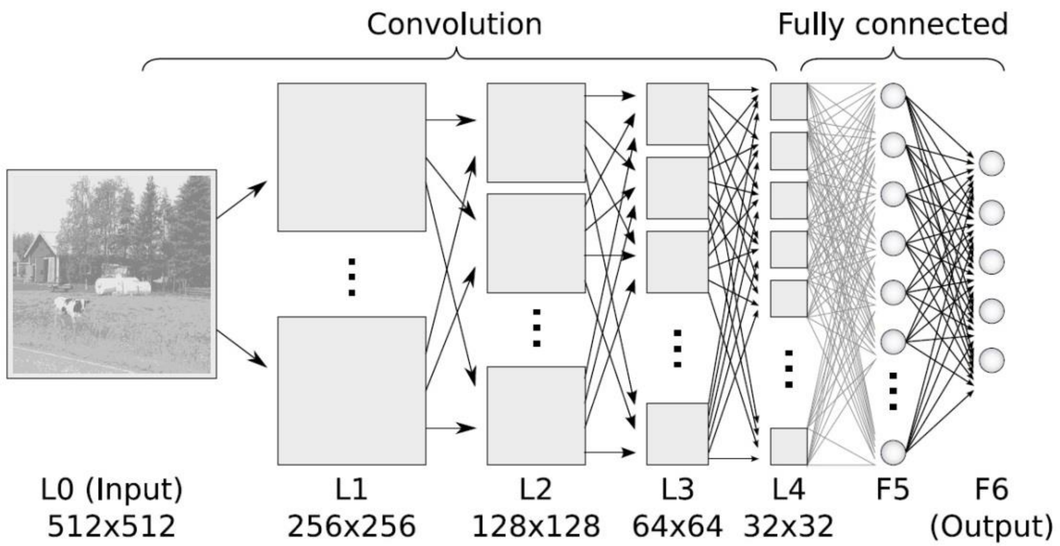
MNIST Digits

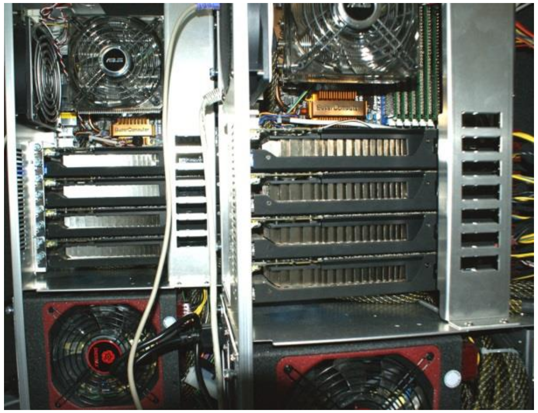
entirely
unsupervised
except for the
colors

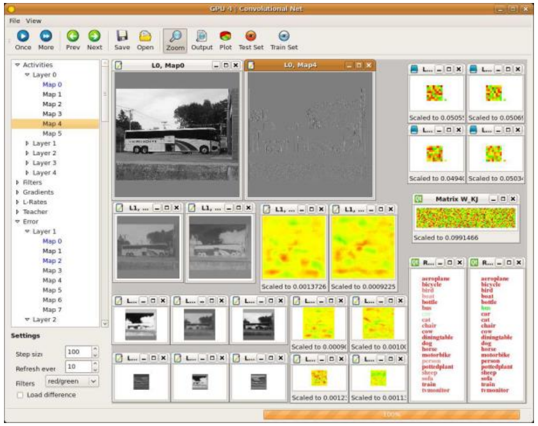


GPU Implementations (CUDA)

- Affordable parallel computers
- General-purpose programming
- Convolutional [Scherer & Behnke, 2009]



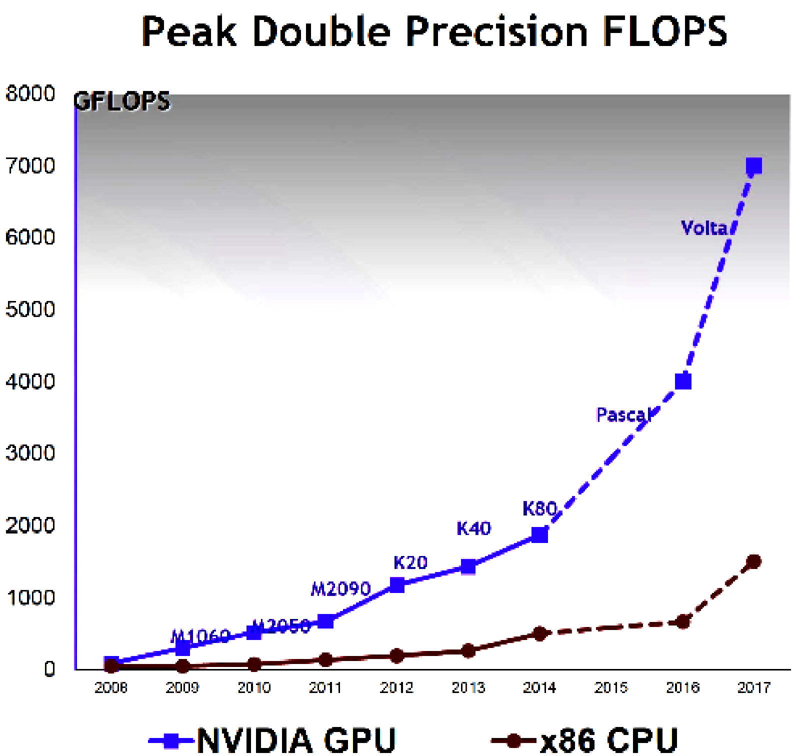


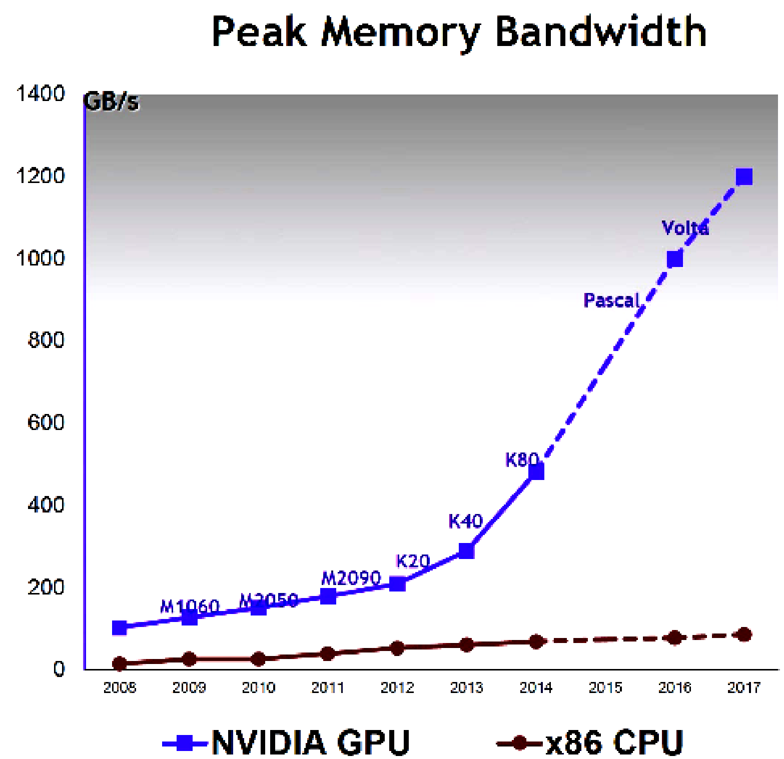


- Local connectivity [Uetz & Behnke, 2009]

GPU vs. CPU Performance

- GPUs are one order of magnitude faster





Tesla Volta V100

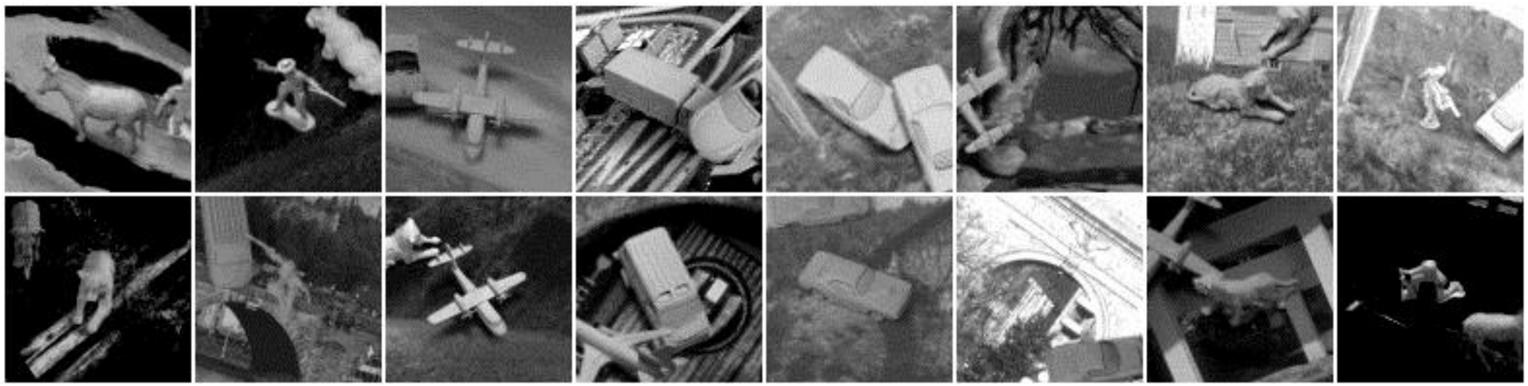
- 7.5 TFLOP/s of double precision (FP64)
- 15 TFLOP/s of single precision (FP32)
- 120 Tensor TFLOP/s for deep learning

$$\underbrace{D}_{\text{FP16 or FP32}} = \underbrace{\begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix}}_{\text{FP16}} \underbrace{\begin{pmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{pmatrix}}_{\text{FP16}} + \underbrace{\begin{pmatrix} C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\ C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,0} & C_{3,1} & C_{3,2} & C_{3,3} \end{pmatrix}}_{\text{FP16 or FP32}}$$

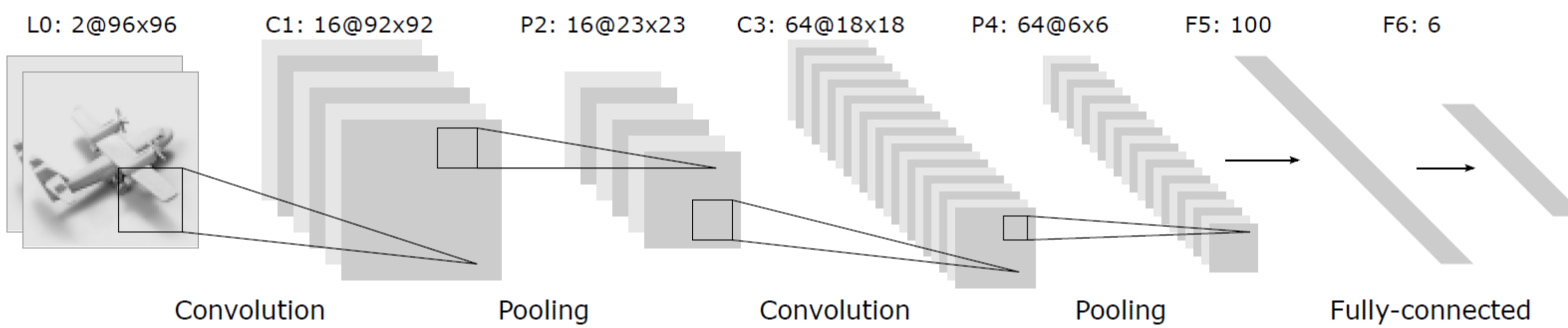
- HBM2 memory with up to 900 GB/sec bandwidth

Image Categorization: NORB

- 10 categories, jittered-cluttered



- **Max-Pooling**, cross-entropy training

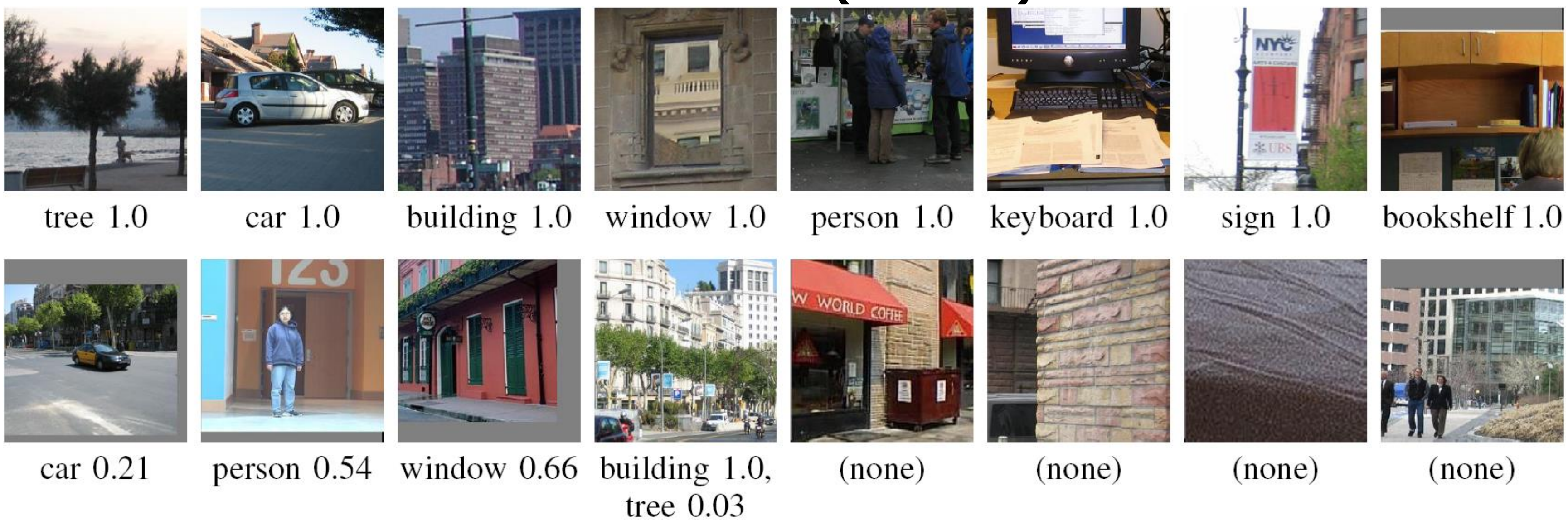


- Test error: 5,6% (LeNet7: 7.8%)

[Scherer, Müller, Behnke, ICANN'10]

Image Categorization: LabelMe

- 50,000 color images (256x256)
- 12 classes + clutter (50%)

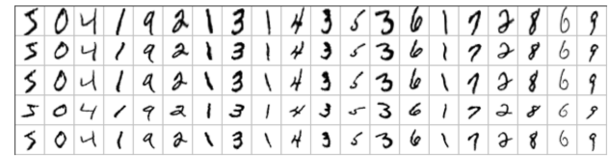


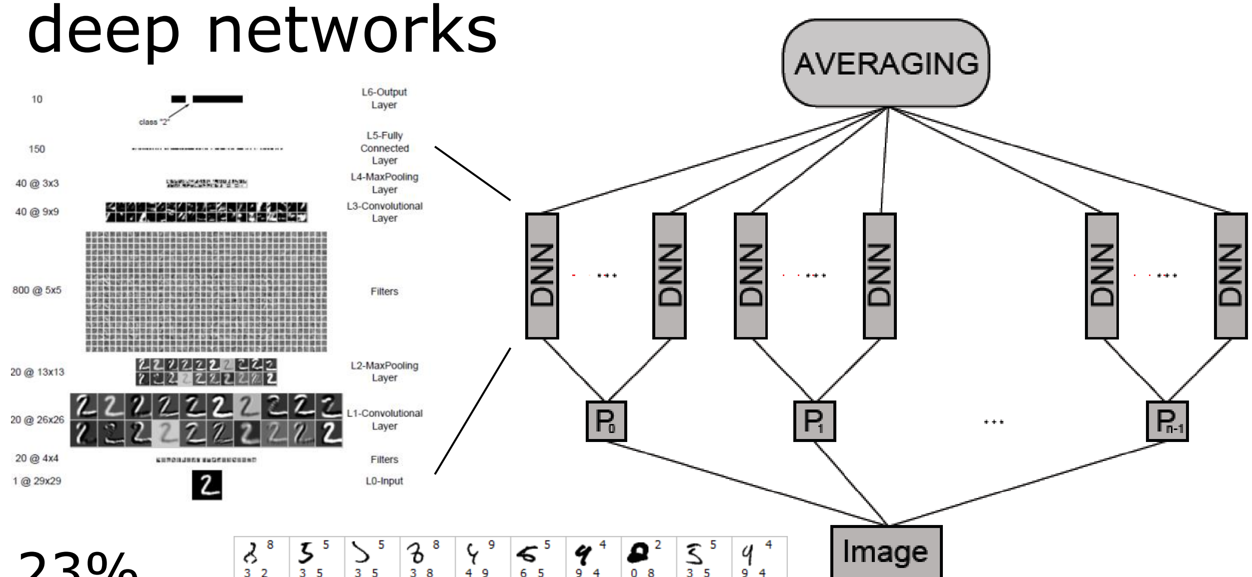
- Error TRN: 3.77%; TST: 16.27%
- Recall: 1,356 images/s

[Uetz, Behnke, ICIS2009]

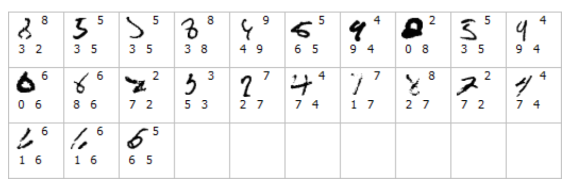
Multi-Column Deep Convolutional Networks

- Different preprocessings
- Trained with distortions
- Bagging deep networks





- MNIST: 0.23%
- NORB: 2.7%
- CIFAR10: 11.2%
- Traffic signs: 0.54% test error



[Ciresan et al. CVPR 2012]

ImageNet Challenge

- 1.2 million images
- 1000 categories, no overlap
- Subset of 11 million images from 15.000+ categories
- Hierarchical category structure (WordNet)

Golf cart (motor vehicle, self-propelled vehicle, wheeled vehicle, …

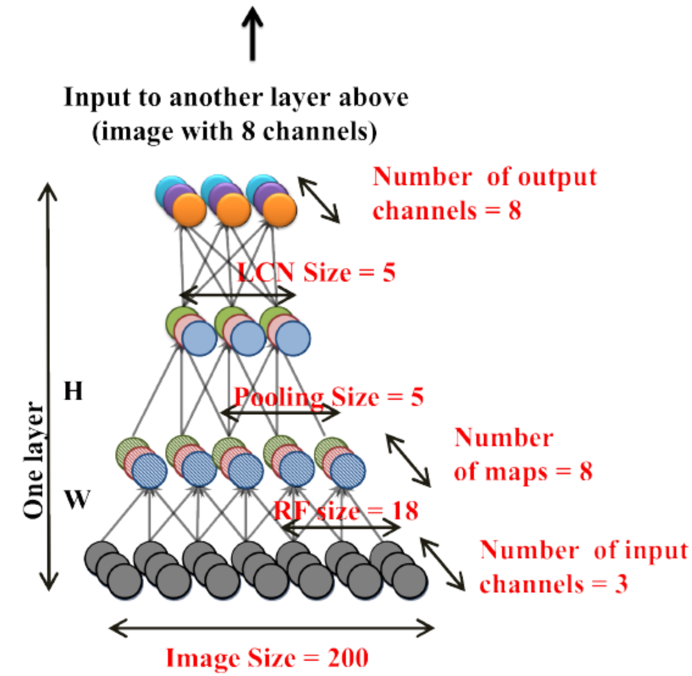
Egyptian cat (domestic cat, domestic animal, animal)

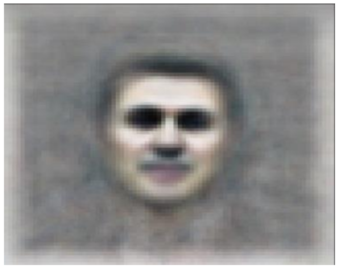
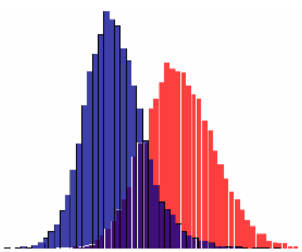
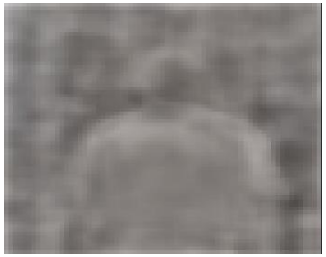
- Task: recognize object category
- Low penalty for extra detections
- Hierarchical error computation

Large Unsupervised Feature Learning

- 9 layer model
- Locally connected
- Sparse auto-encoder
- L2 pooling
- Local contrast normalization
- 1 billion connections
- Trained on 10 million images
- Unsupervised learned detectors

3x



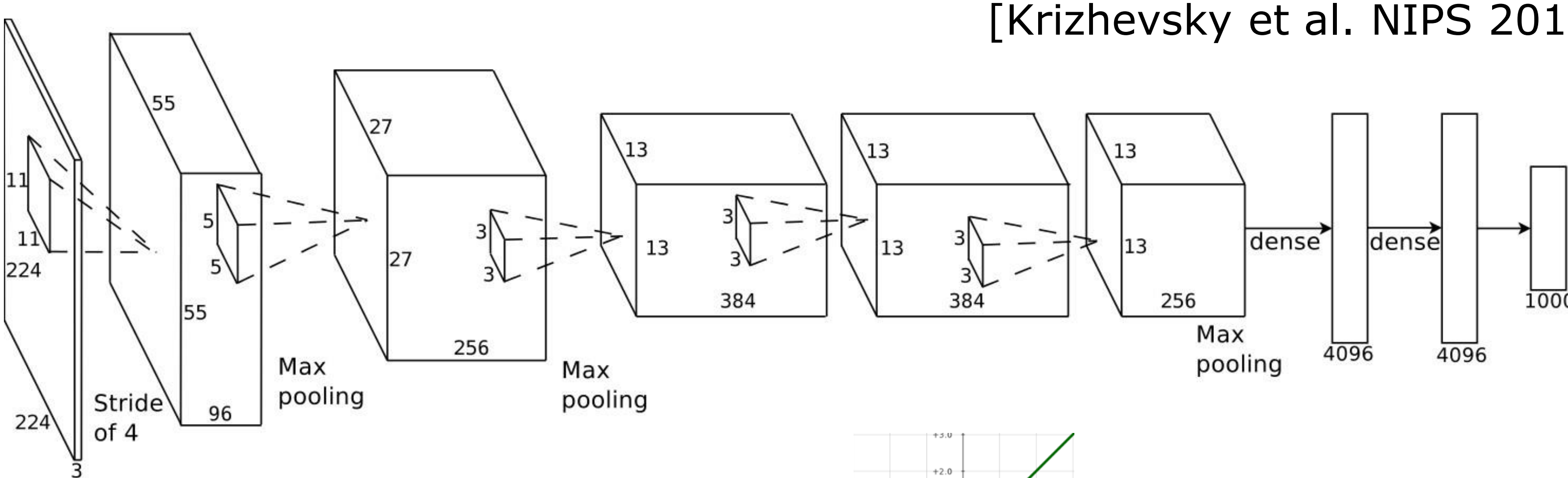




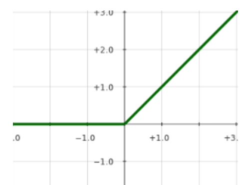

- Supervised ImageNet 2011 results (14M images, 22K categories): 15.8%

[Le et al. 2012]

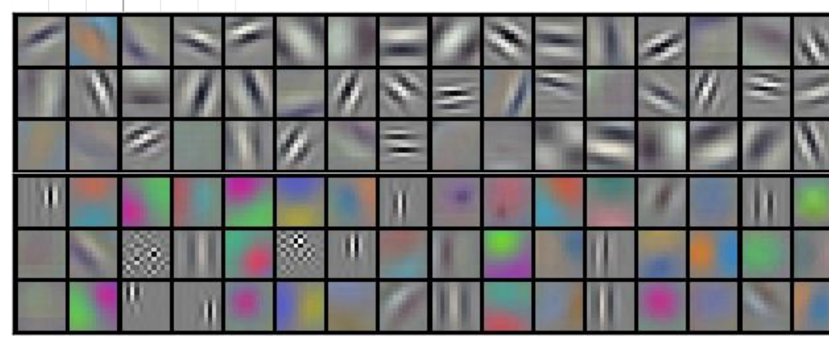
Large Convolutional Network

[Krizhevsky et al. NIPS 2012]



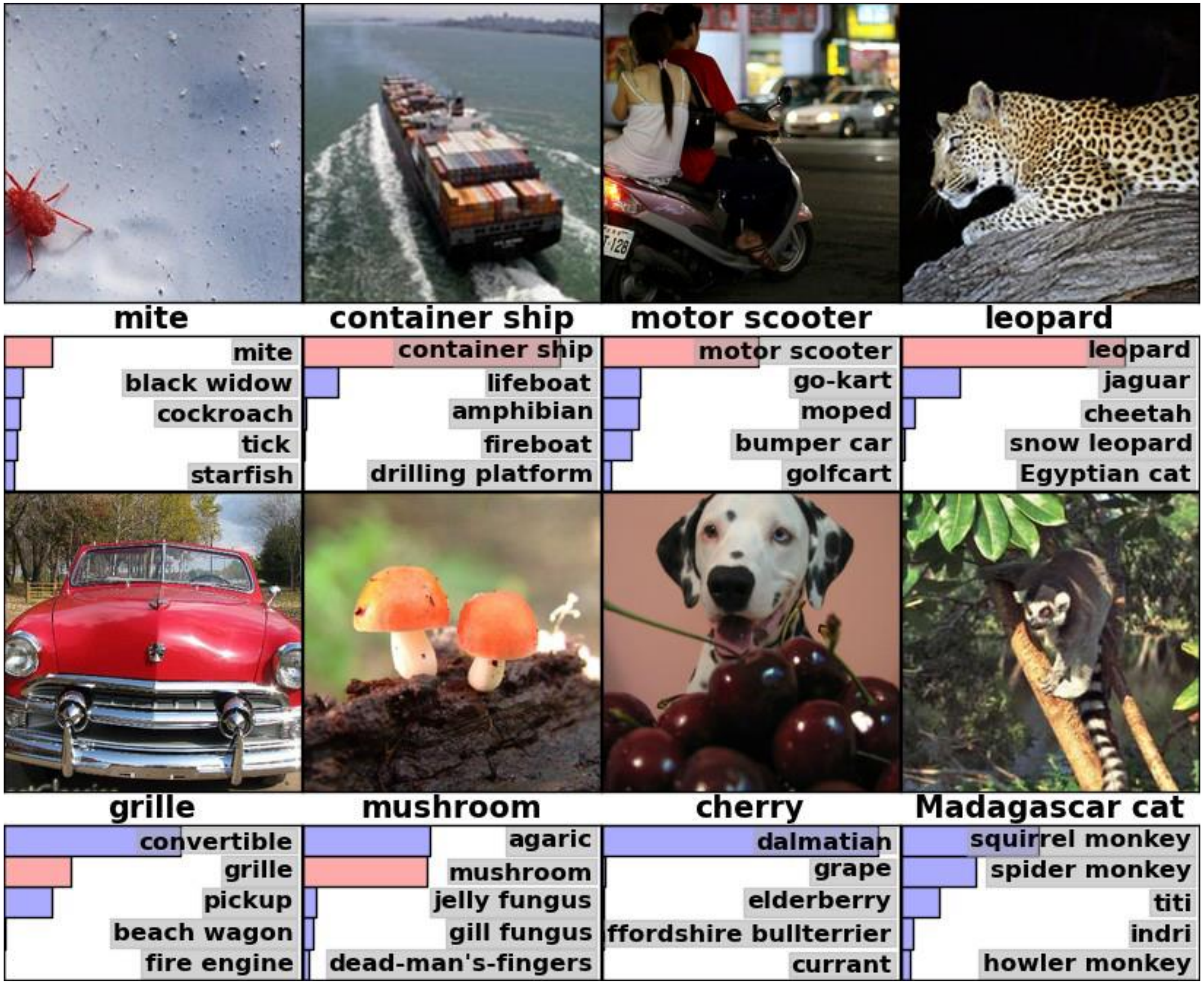


- Rectifying transfer functions
- 650,000 neurons
- 60,000,000 parameters
- 630,000,000 connections
- Trained using dropout and data augmentation
- Testing 10 sub-images
- ILSVRC-2012: top-5 error 15.3%



96 learned low-level filters

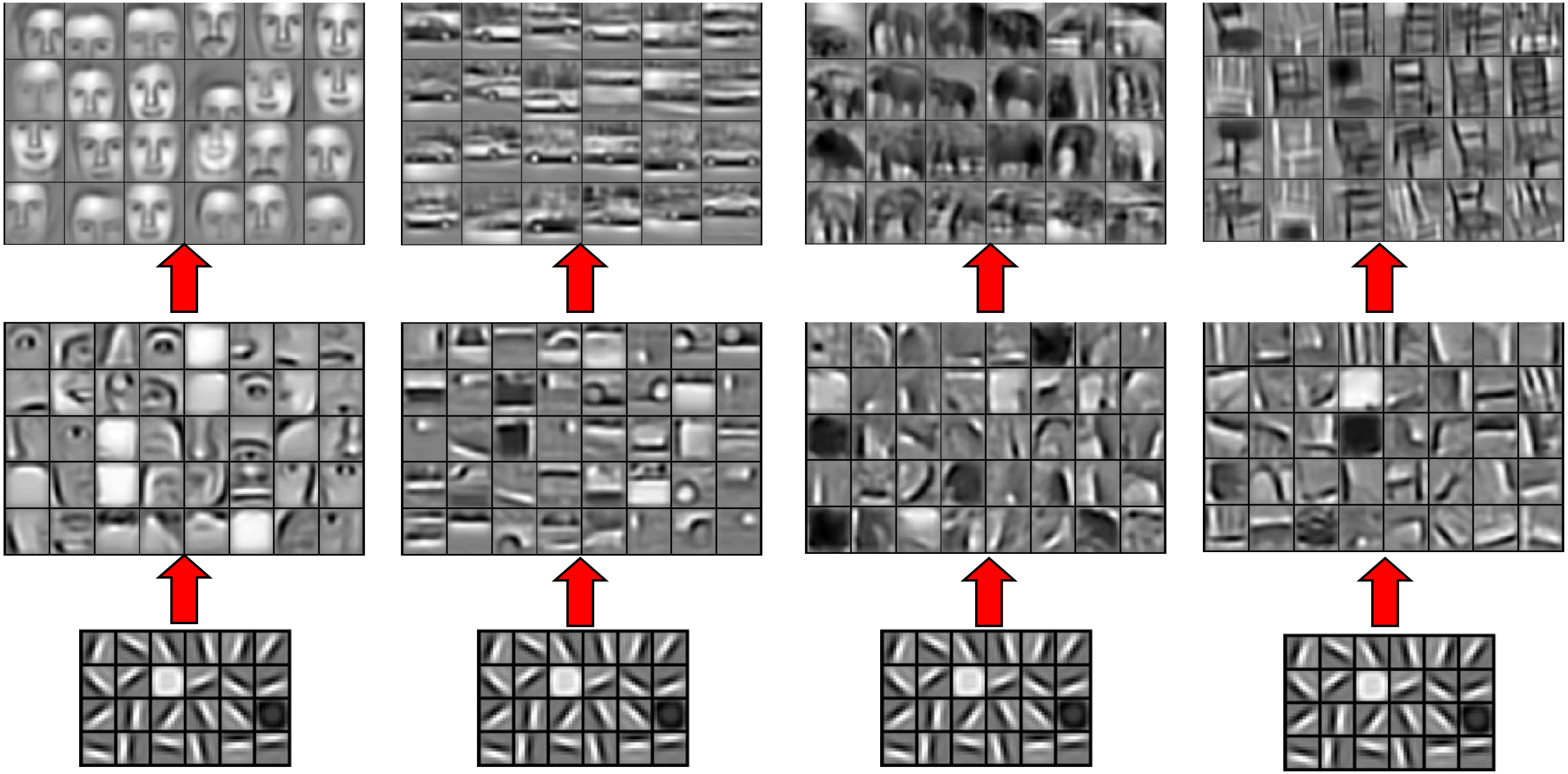
Validation Classification



[Krizhevsky et al. NIPS 2012]

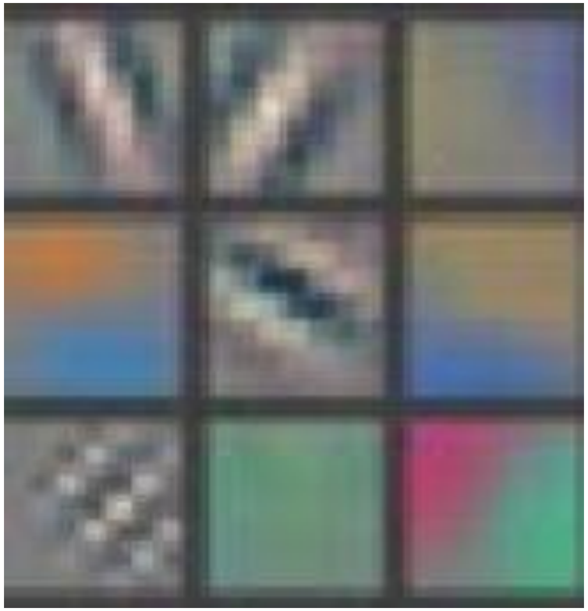
Learning of Object Parts

- Examples of learned object parts from object categories



Learned Visual Features

- Weights with strong contribution to activity

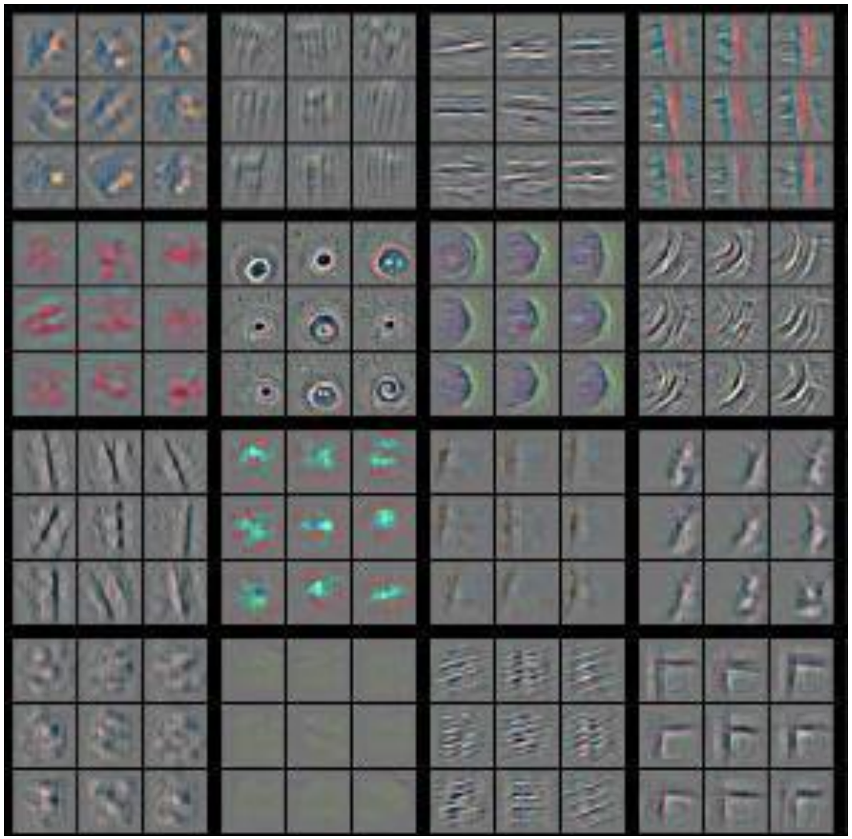


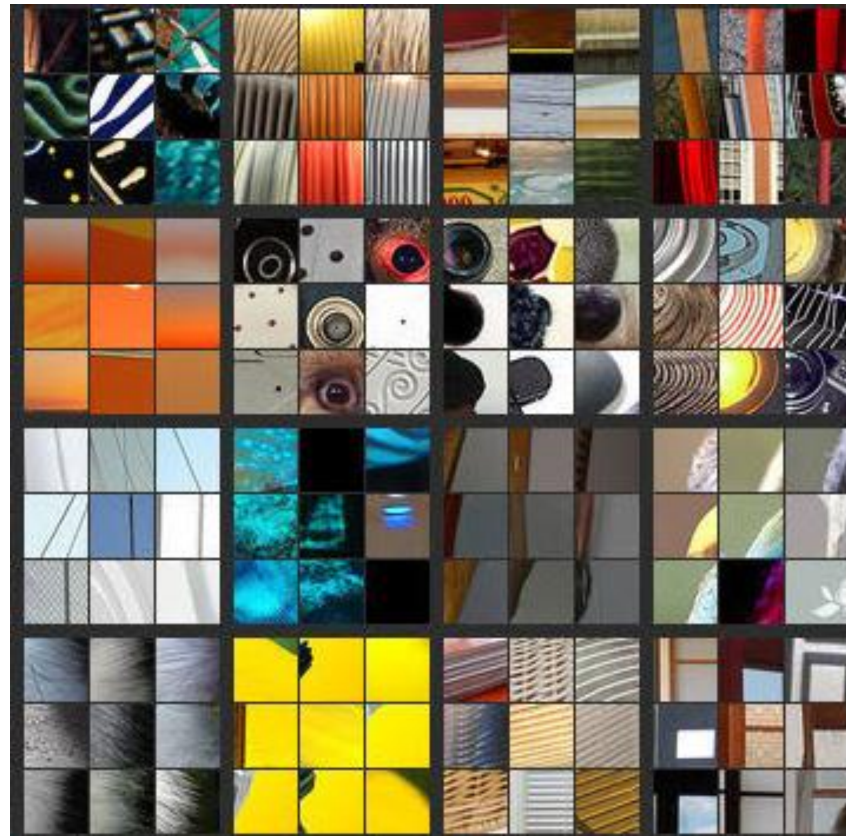
- Strongly activating stimuli

[Zeiler and Fergus 2014]

Learned Visual Features

- Deconvolved features
- Strongly activating stimuli

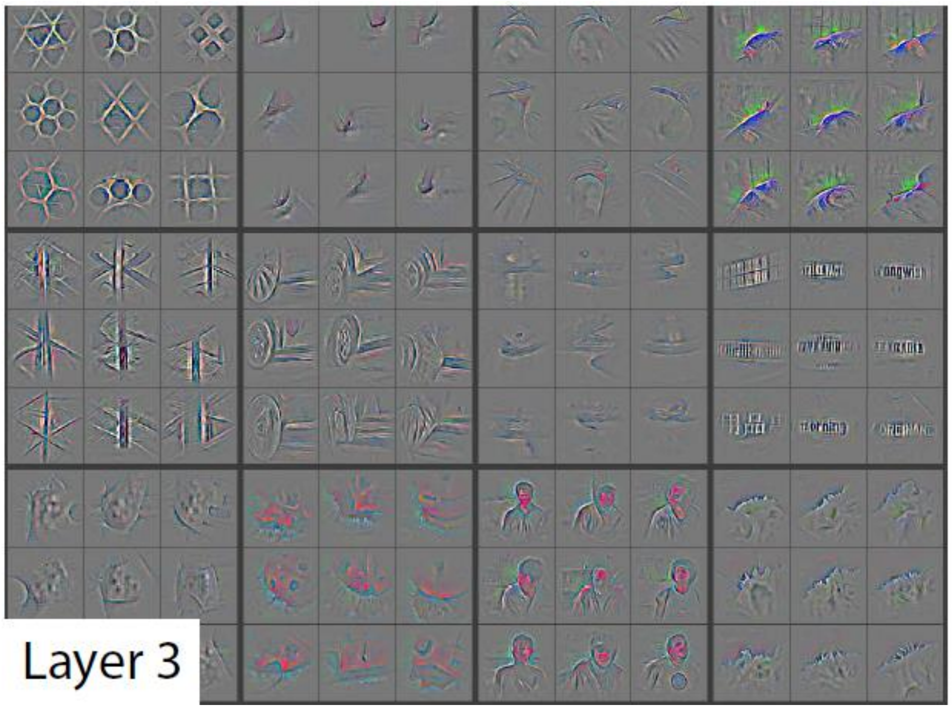


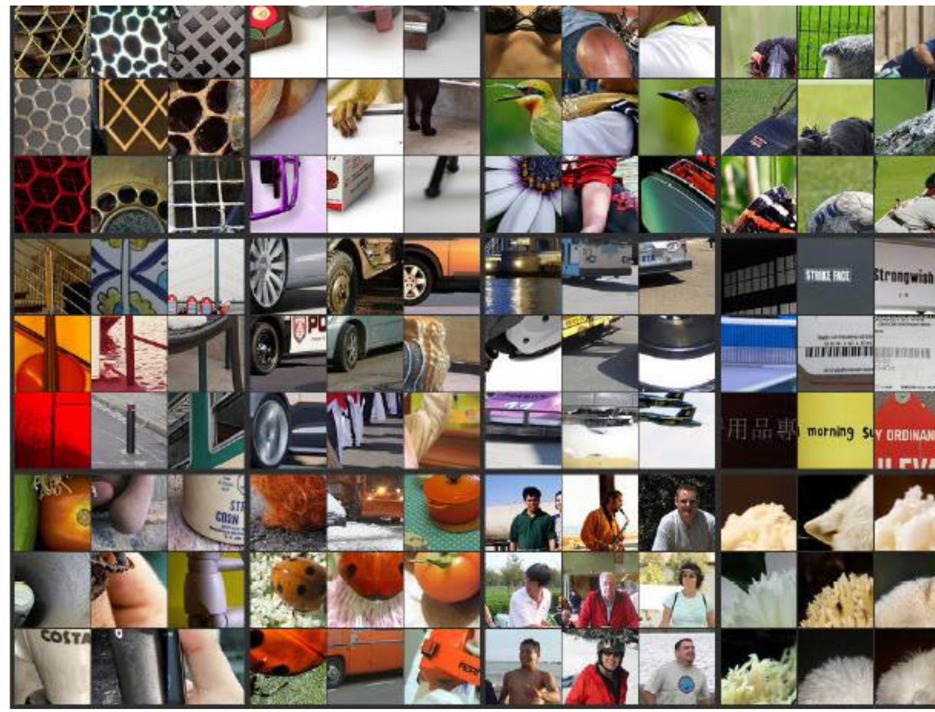


[Zeiler and Fergus 2014]

Learned Visual Features

- Deconvolved features
- Strongly activating stimuli

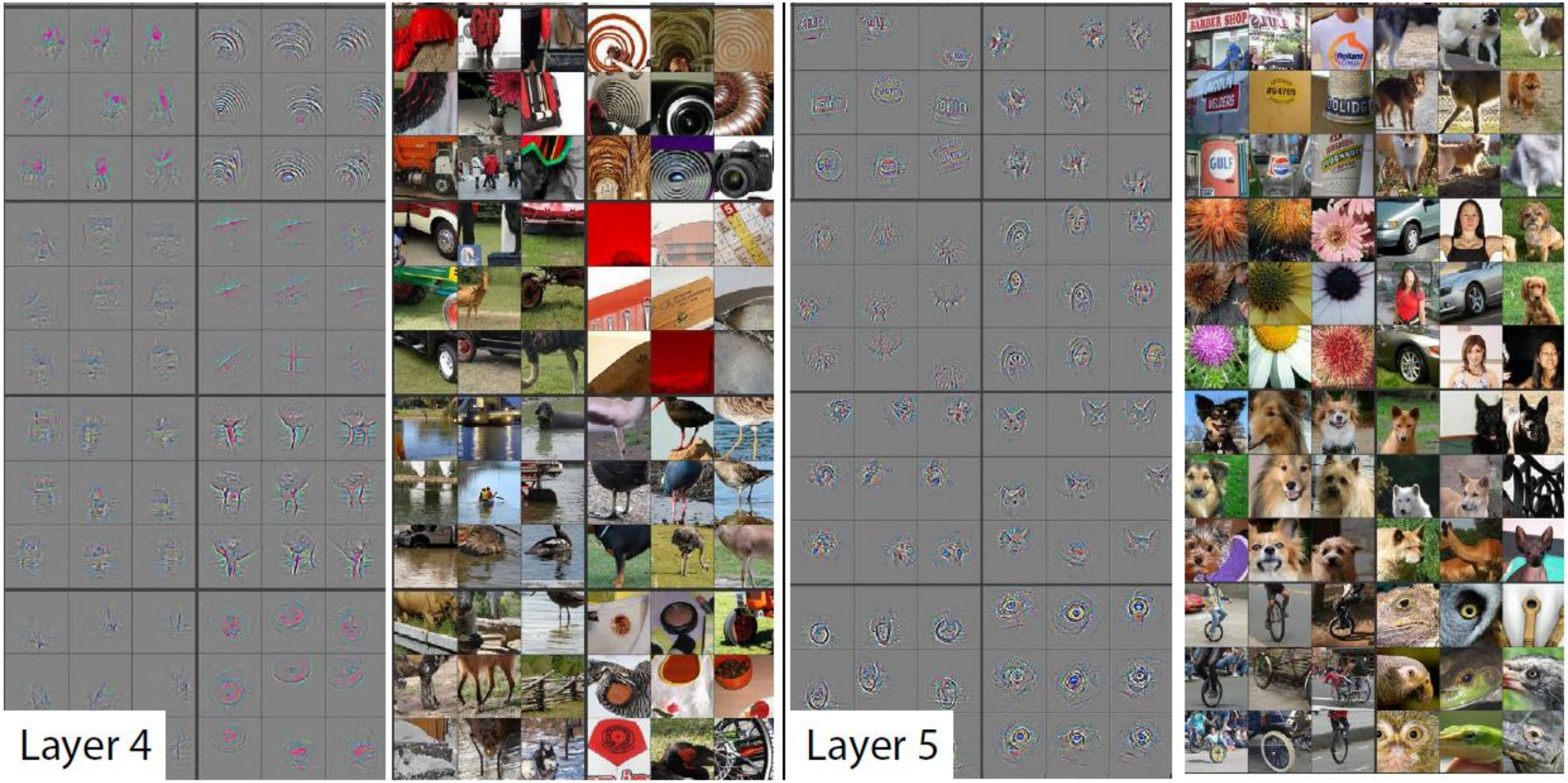




[Zeiler and Fergus 2014]

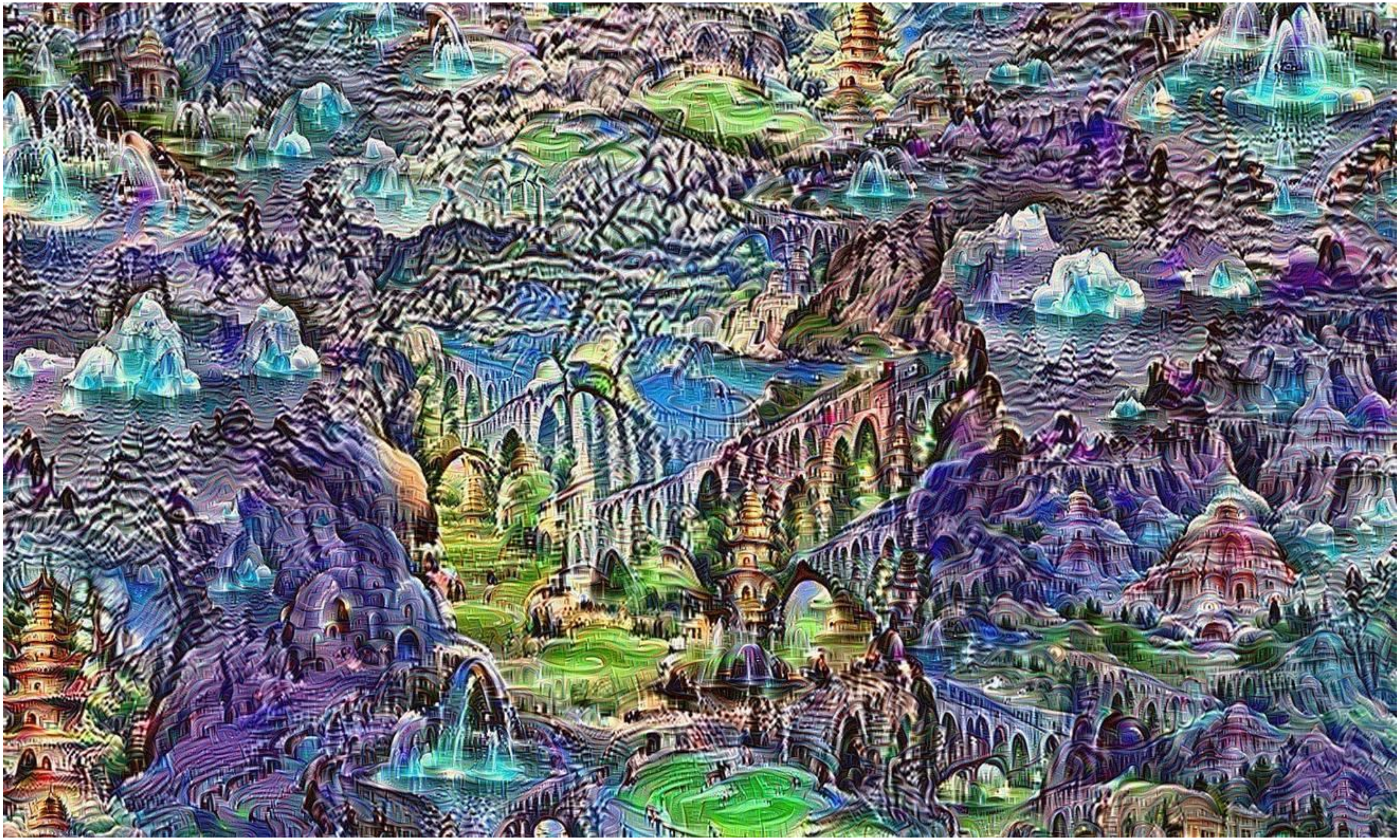
Learned Visual Features

- Deconvolved features and activating stimuli



[Zeiler and Fergus 2014]

Dreaming Deep Networks

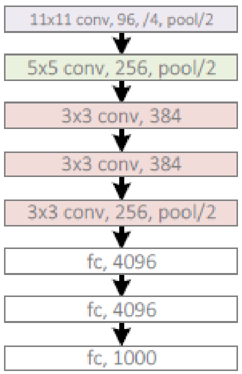


[Mordvintsev et al. 2015]

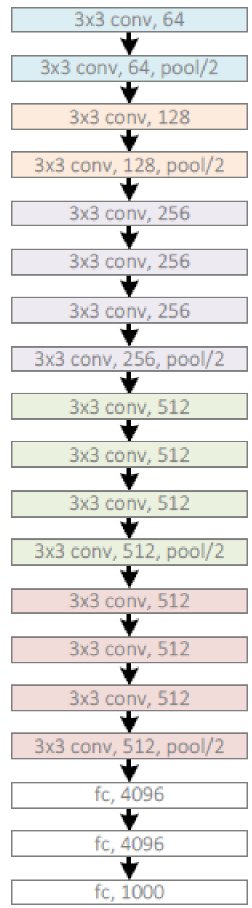
Example CNNs Structures of ILSVRC winners

- Revolution of depth

AlexNet, 8 layers
(ILSVRC 2012)



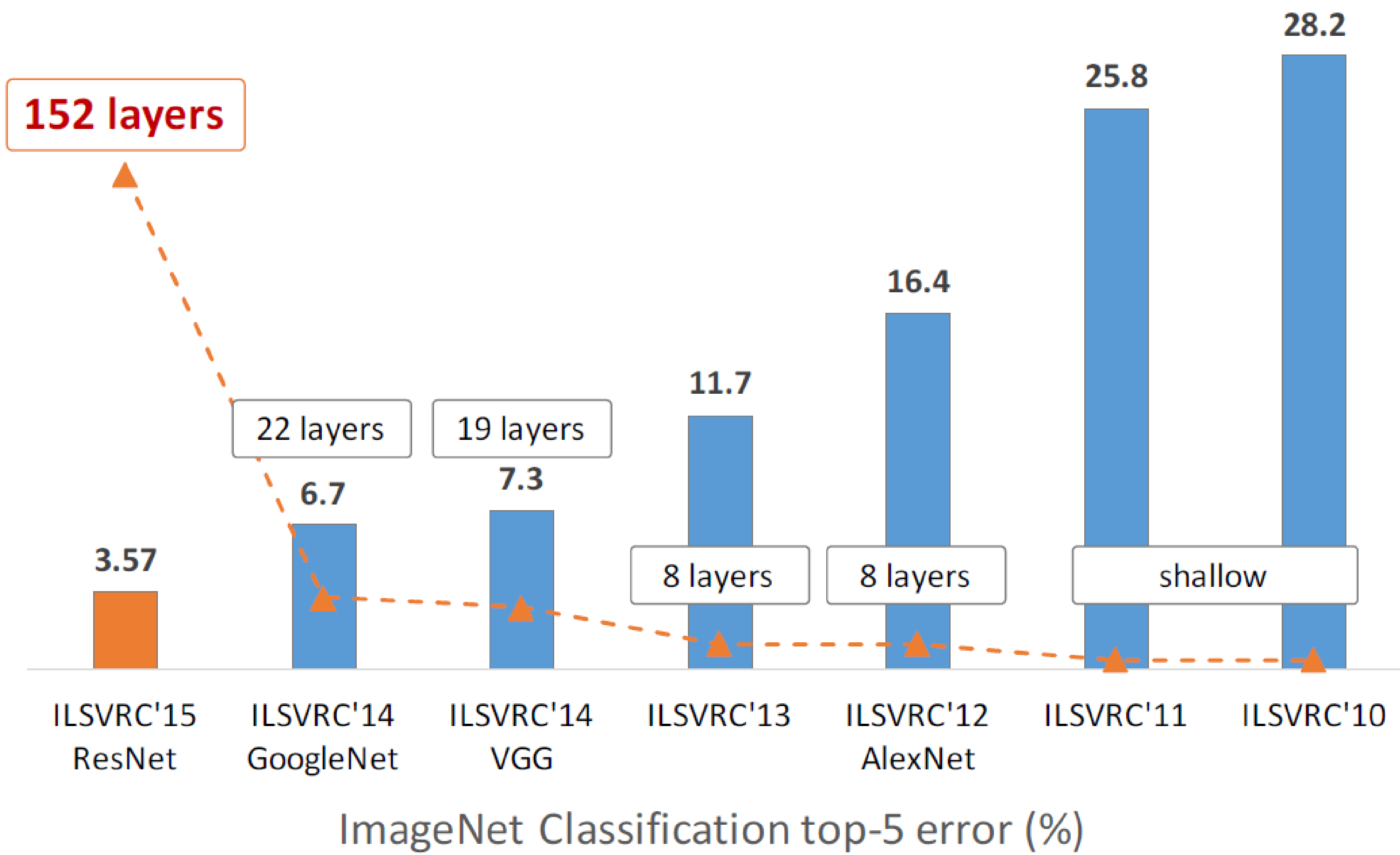
VGG, 19 layers
(ILSVRC 2014)



GoogleNet, 22 layers
(ILSVRC 2014)

[He CVPR 2016]

Object Recognition Performance on ImageNet



[He et al. 2015]

Surpassing Human Performance

GT: horse cart
1: horse cart
2: minibus
3: oxcart
4: stretcher
5: half track

GT: birdhouse
1: birdhouse
2: sliding door
3: window screen
4: mailbox
5: pot

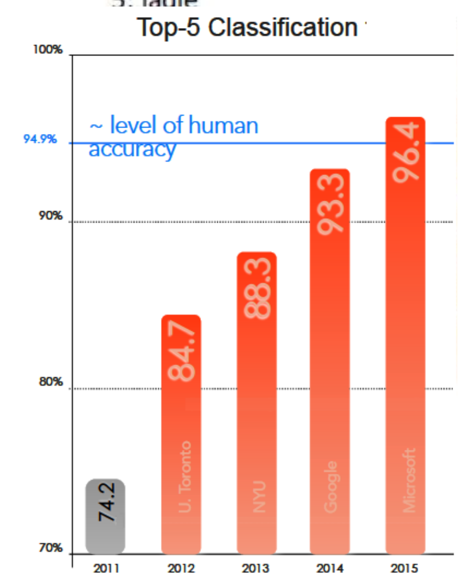
GT: forklift
1: forklift
2: garbage truck
3: tow truck
4: trailer truck
5: go-kart

GT: letter opener
1: drumstick
2: candle
3: wooden spoon
4: spatula
5: ladle

GT: coucal
1: coucal
2: indigo bunting
3: lorikeet
4: walking stick
5: custard apple

GT: komondor
1: komondor
2: patio
3: llama
4: mobile home
5: Old English sheepdog

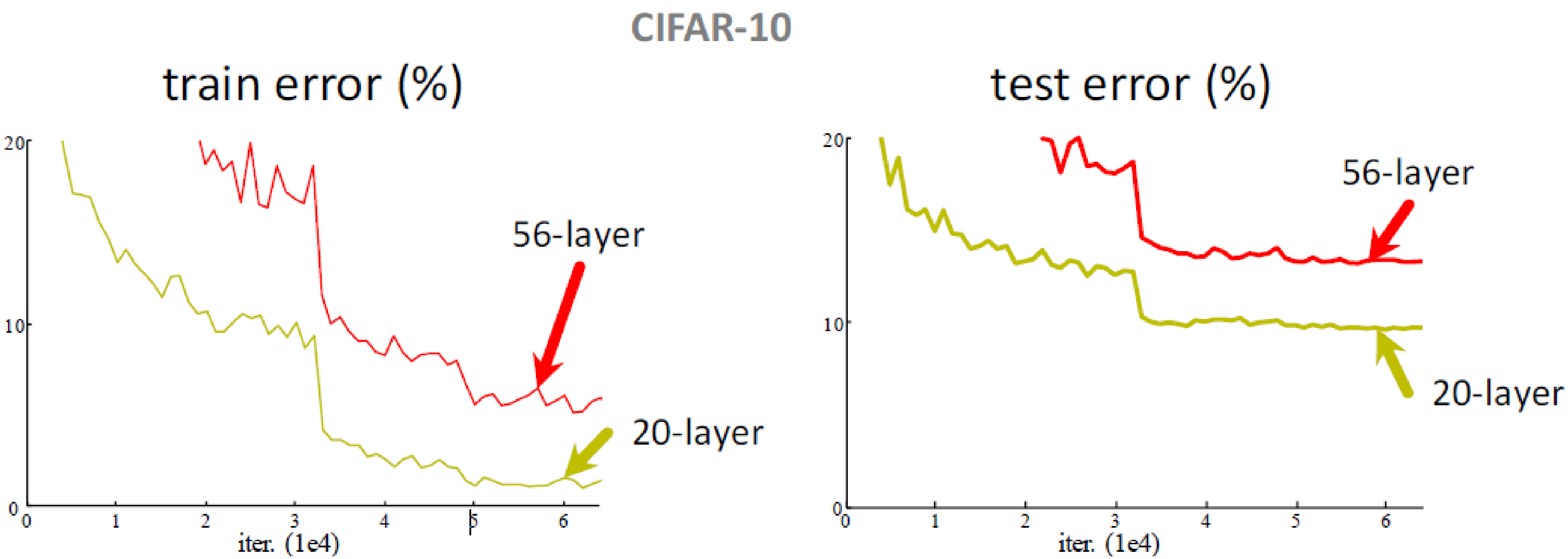
GT: yellow lady's slipper
1: yellow lady's slipper
2: slug
3: hen-of-the-woods
4: stinkhorn
5: coral fungus



[He et al. 2015]

Are Deeper Networks Always Better?

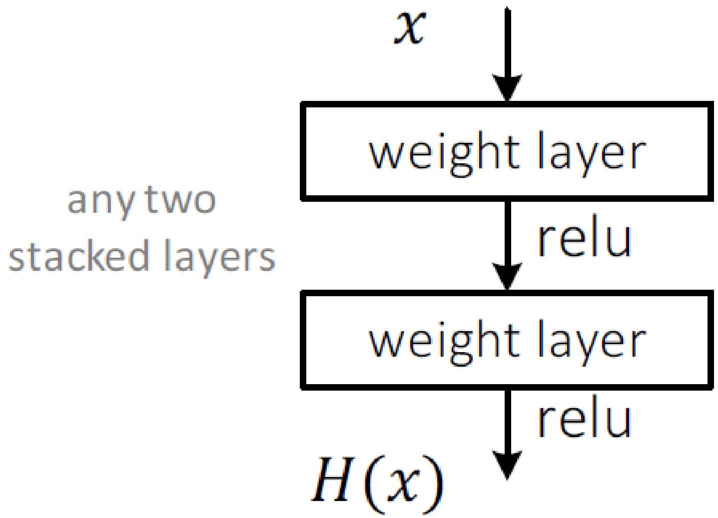
- Plain nets: Stacking 3x3 convolutional layers
- 56-layer network has **higher training error** and test error than 20-layer network



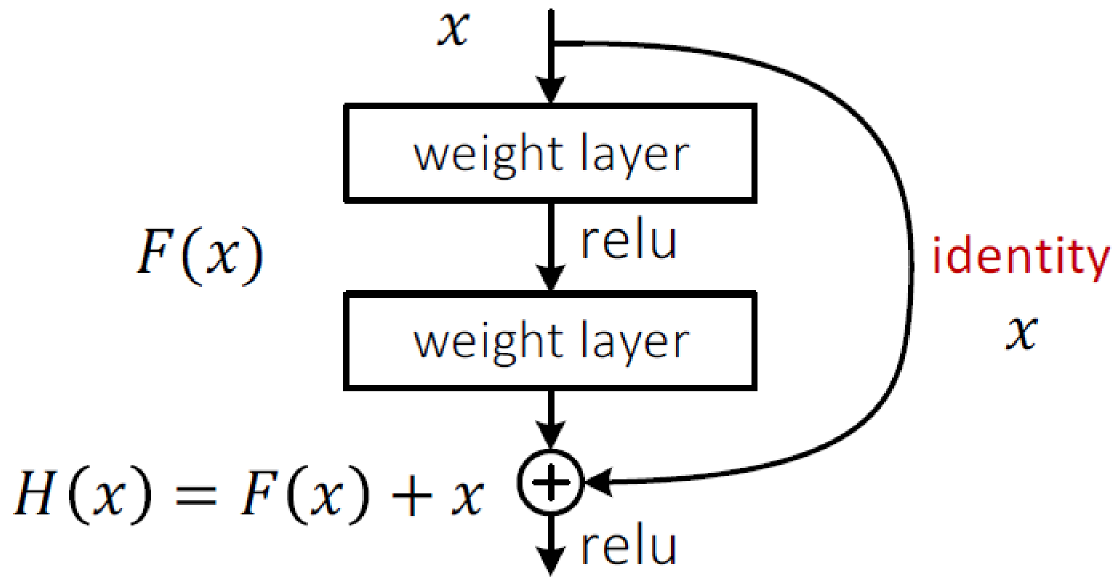
[He et al. CVPR 2016]

Deep Residual Learning

- Plain network

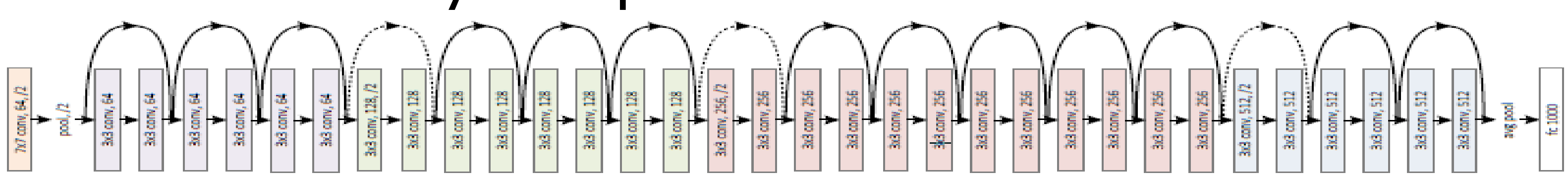


- Residual network



[He et al. CVPR 2016]

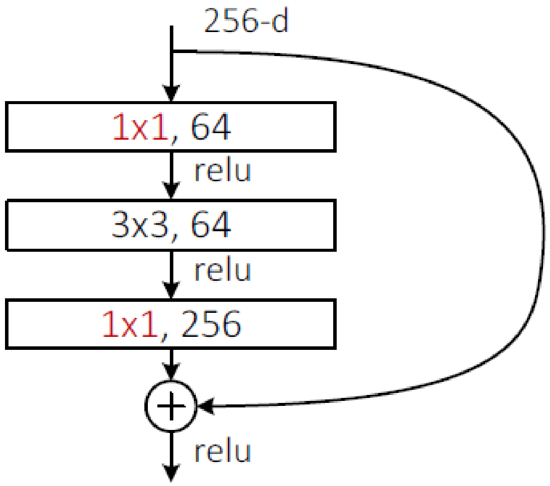
- ResNet: Very deep network

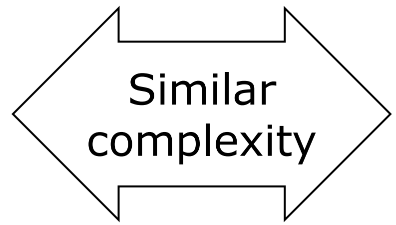


- Iteratively refining representations [Greff et al. ICLR 2017]

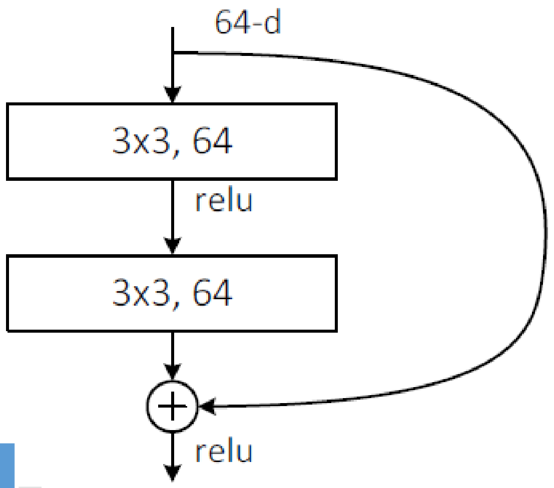
Local Bottlenecks to make Networks Deeper

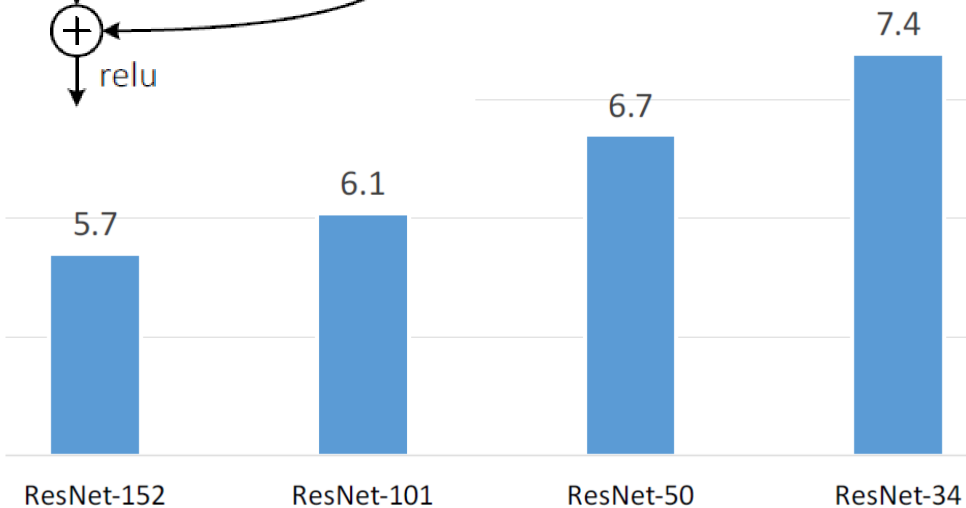
- Bottleneck





- All 3×3 conv.





ImageNet **10-crop** testing, top-5 val error (%)

[He et al. CVPR 2016]

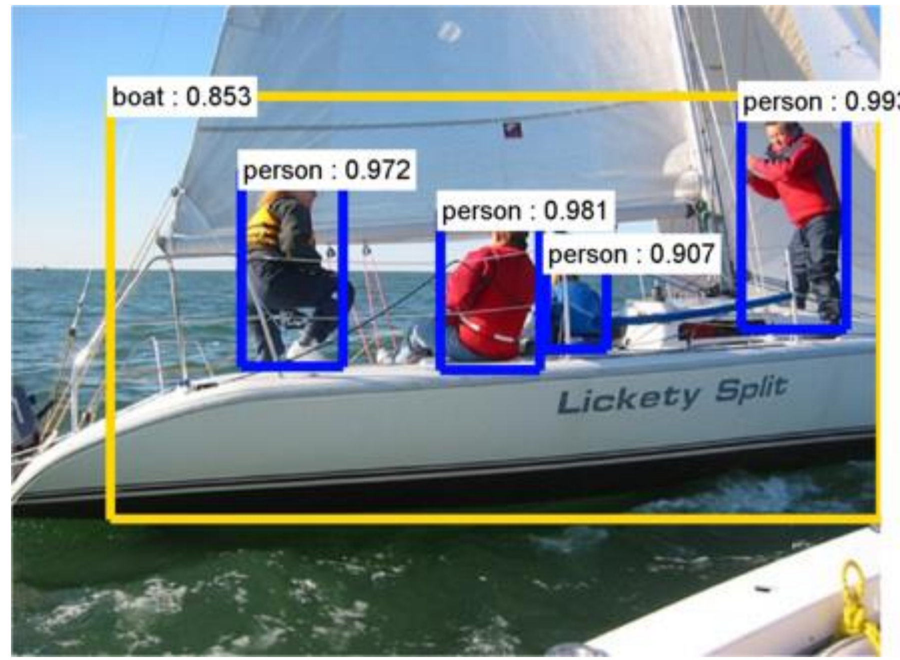
Limitations of Convolutional Processing

- All image positions processed in the same way
- No scale invariance
- No focus of attention

Object Detection

- Image categorization
  What?

- Object detection
  What + where?



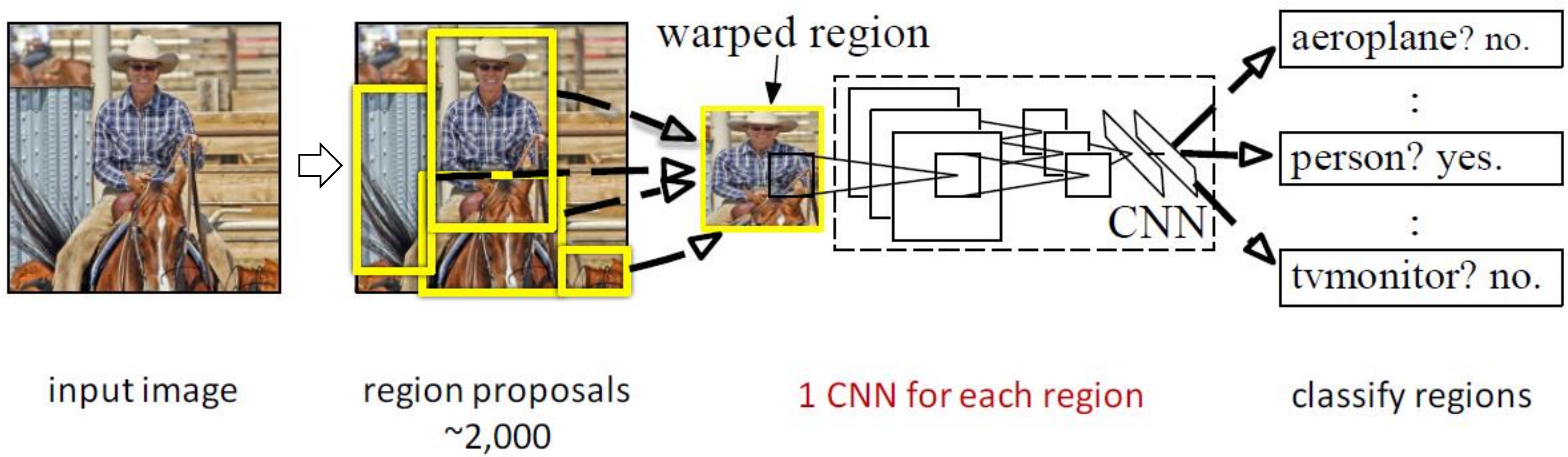
Object Detection in Images

- Bounding box annotation
- Structured loss that directly maximizes overlap of the prediction with ground truth bounding boxes
- Evaluated on two of the Pascal VOC 2007 classes: cows and horses

[Schulz, Behnke, ICANN 2014]

Region-based CNN Pipeline (R-CNN)

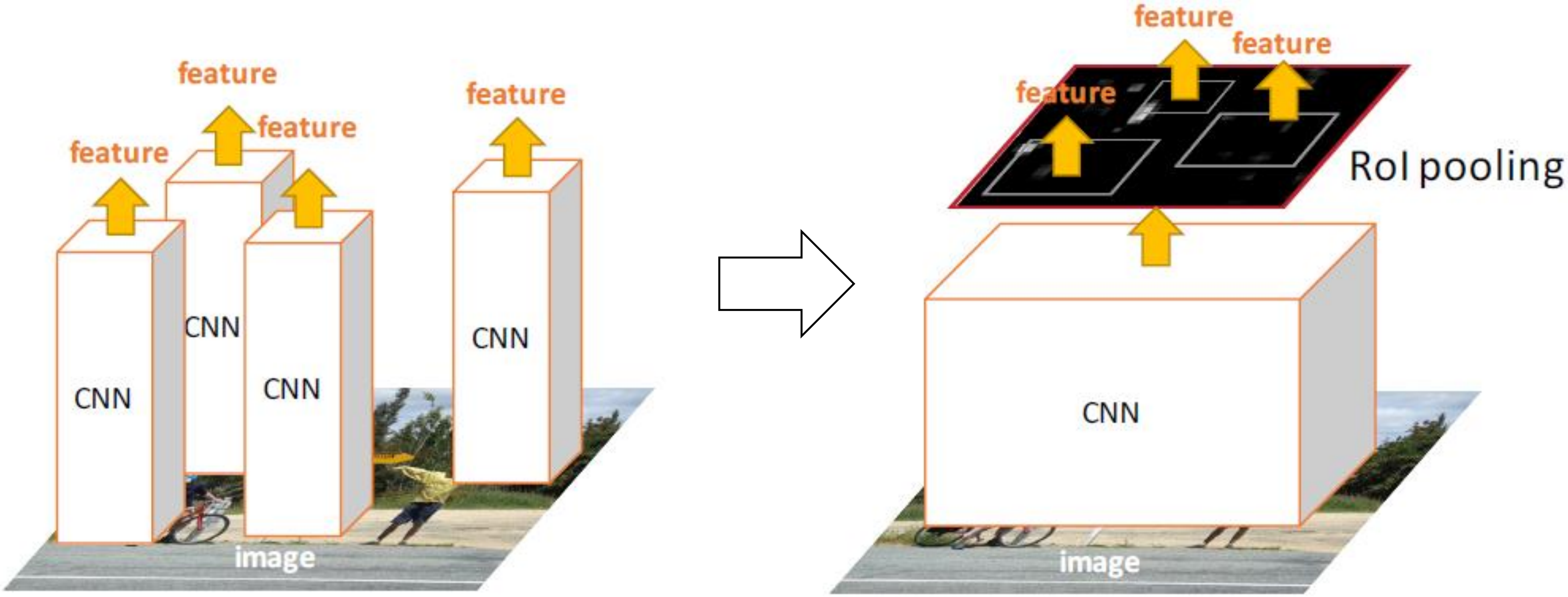
- Generate region proposals
- Cut out and warp them to constant size
- Classify warped regions with CNN



[Girshick et al. CVPR 2014]

Fast R-CNN

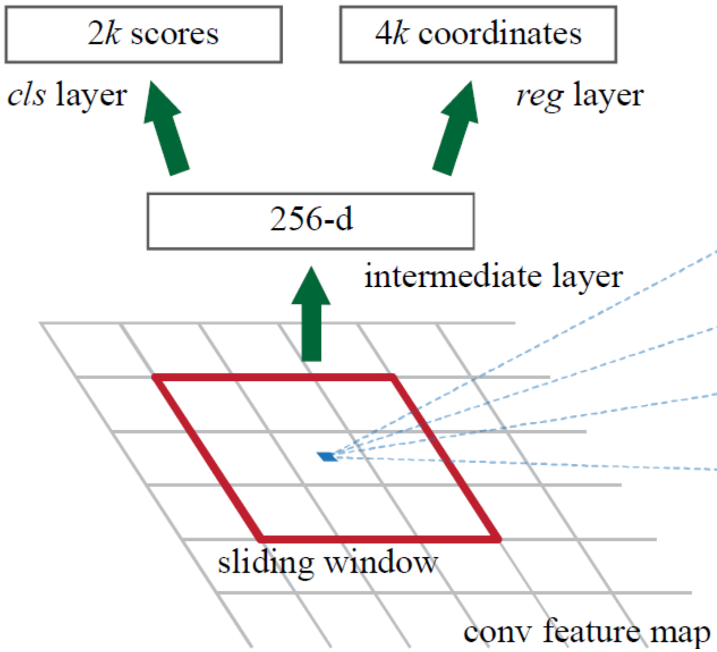
- Convolutional processing at many overlapping regions inefficient
- Share convolutional layers and cut out features (Region of interest pooling)

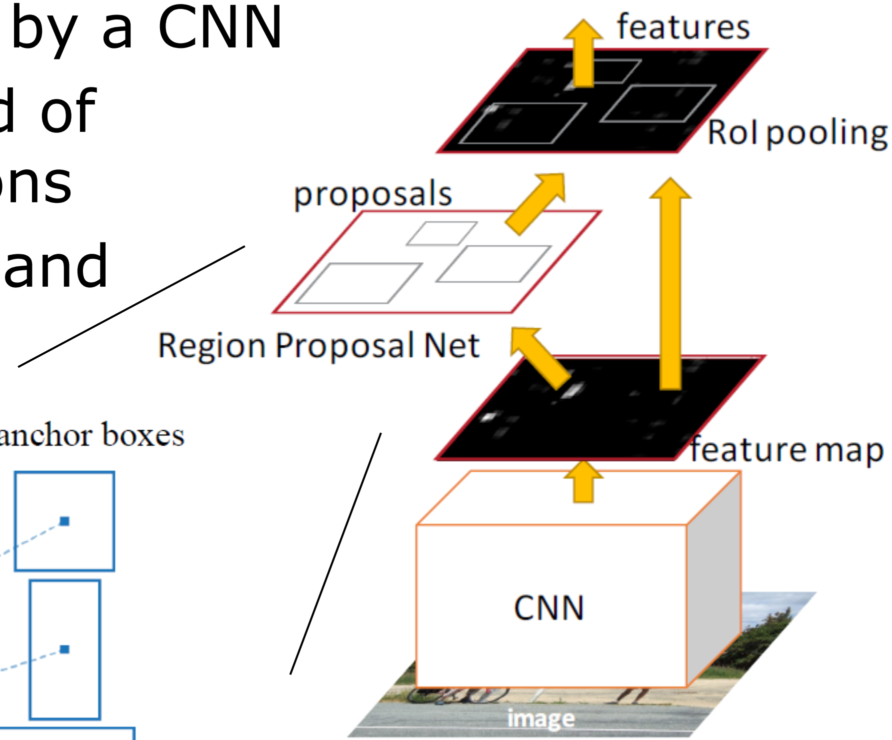


[Girschik ICCV 2015]

Faster R-CNN

- Region proposals computed by a CNN
- Anchor boxes placed for grid of locations, sizes, aspect rations
- For each anchor objectness and box coordinates predicted

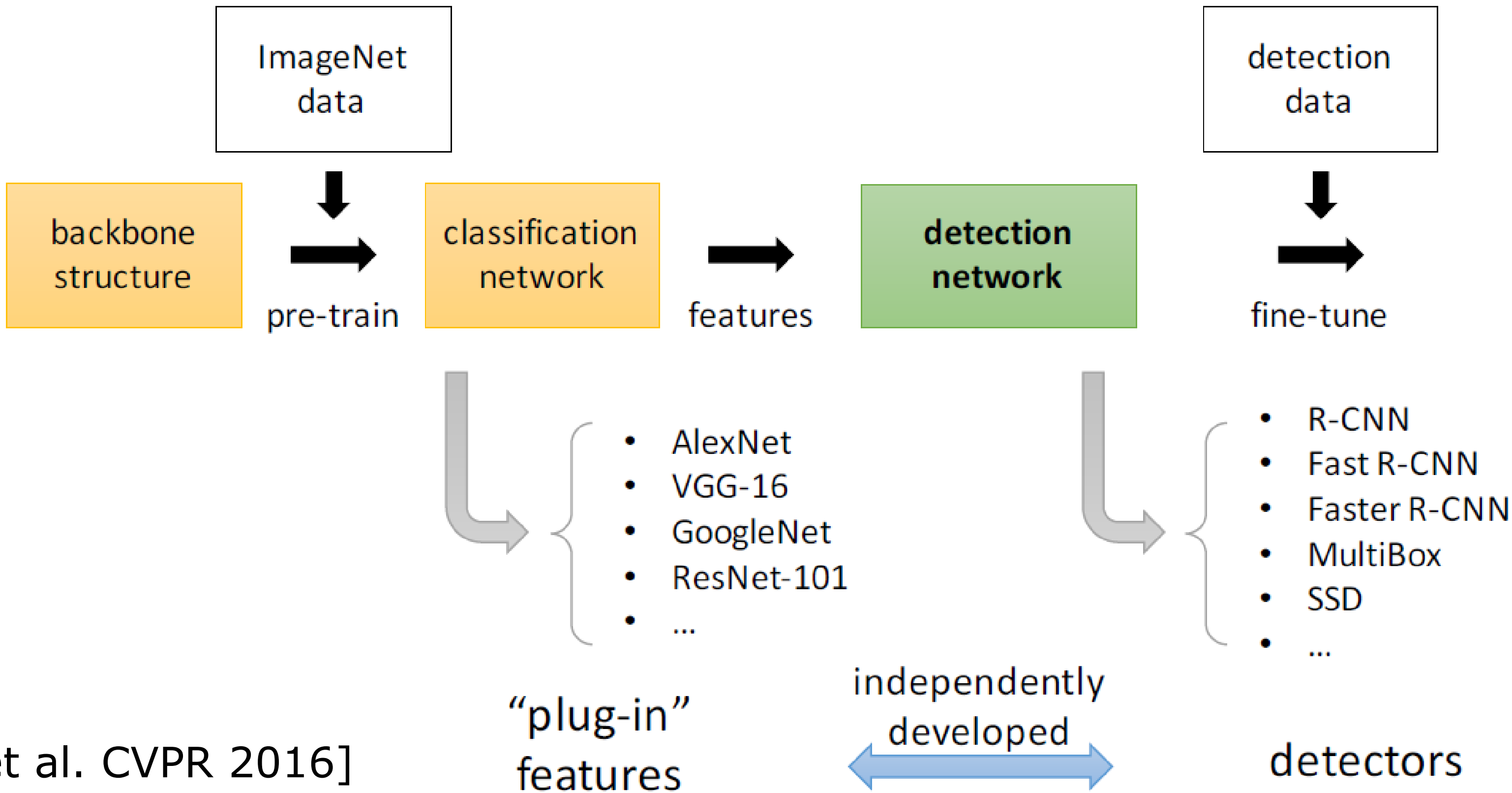




[Ren et al. NIPS 2015]

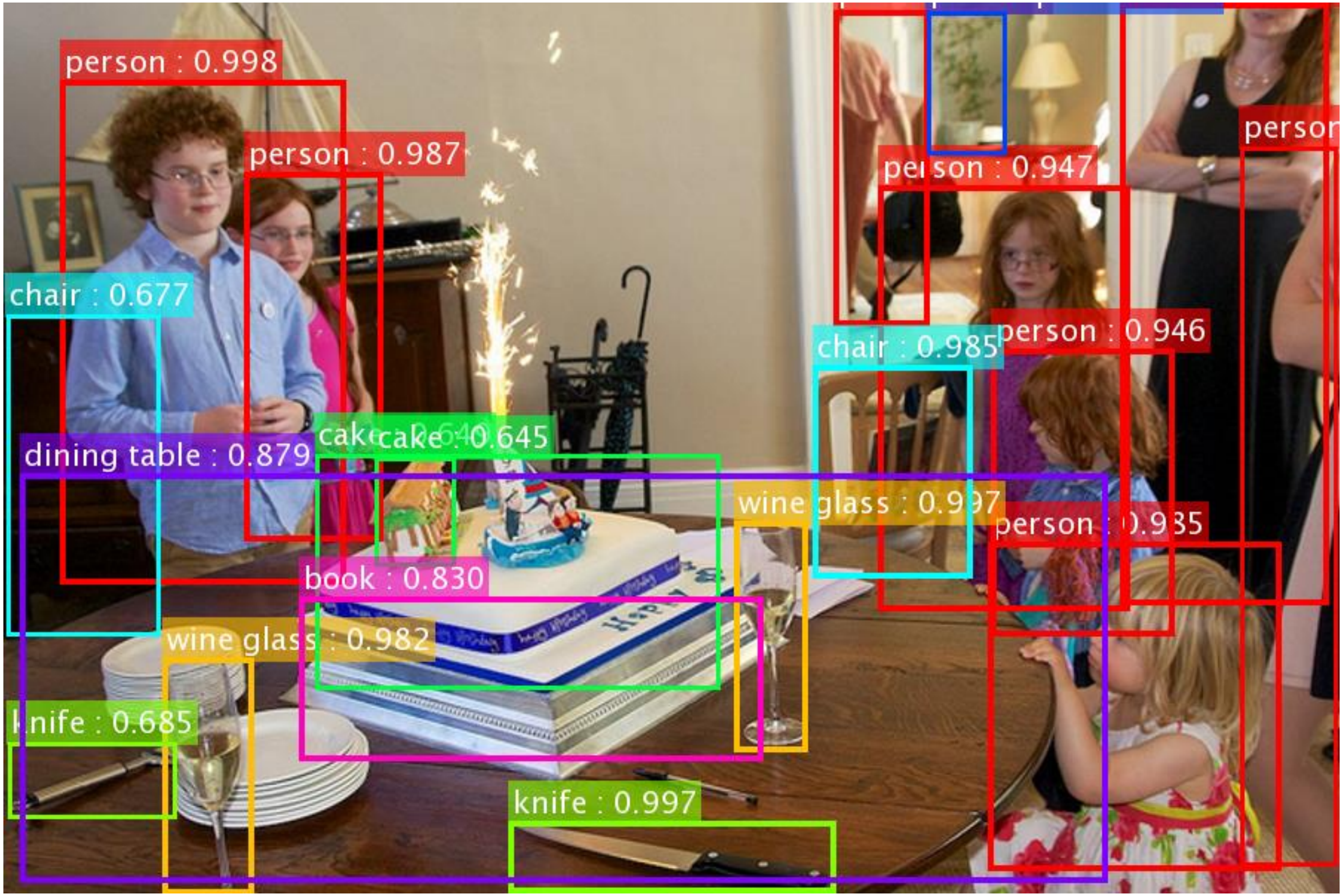
Object Detection Pipeline

- Combine classification and detection models
- Use pre-trained features



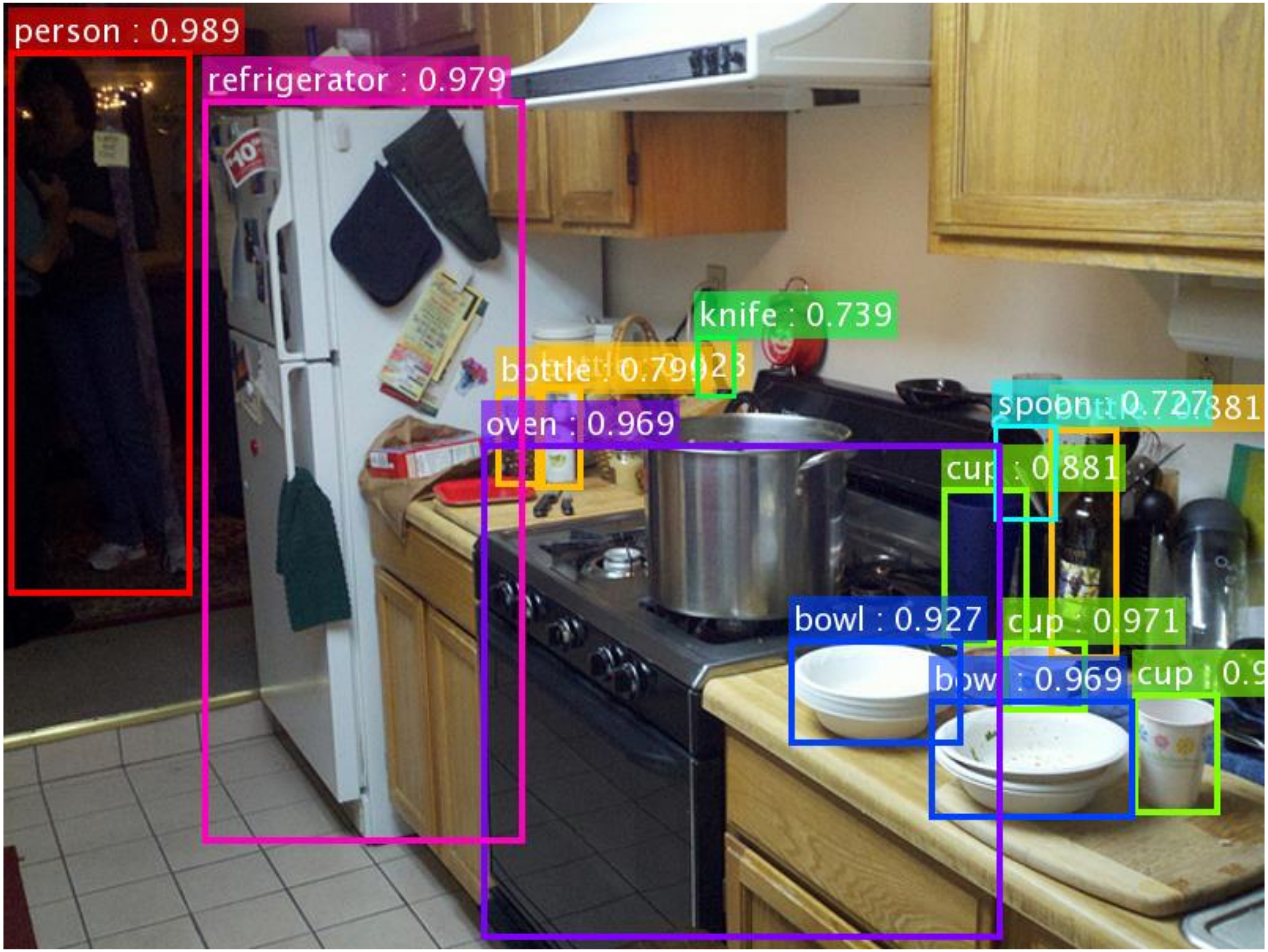
[He et al. CVPR 2016]

Faster R-CNN + ResNet Object Detection Result

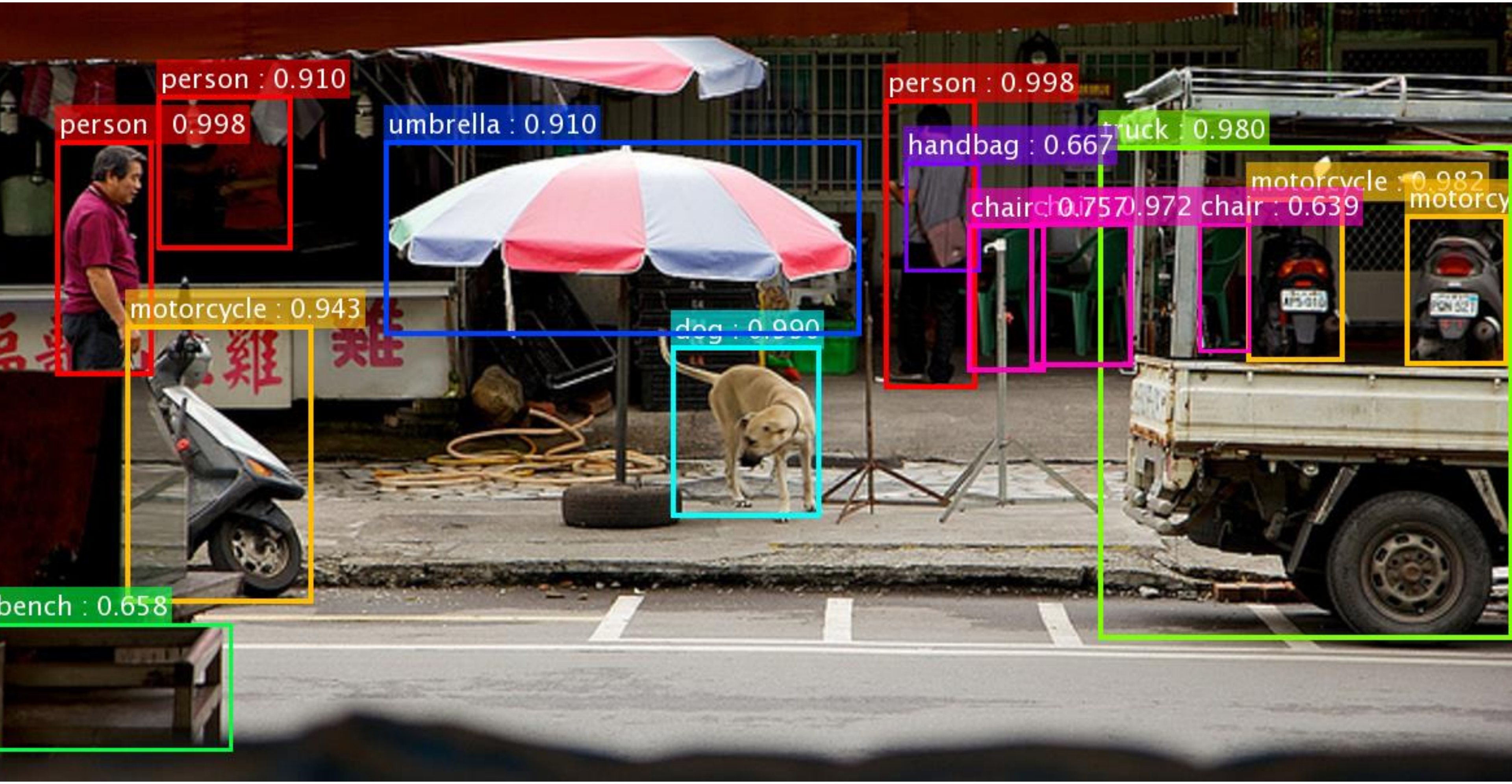


[He et al. CVPR 2016]

Faster R-CNN + ResNet Object Detection Result

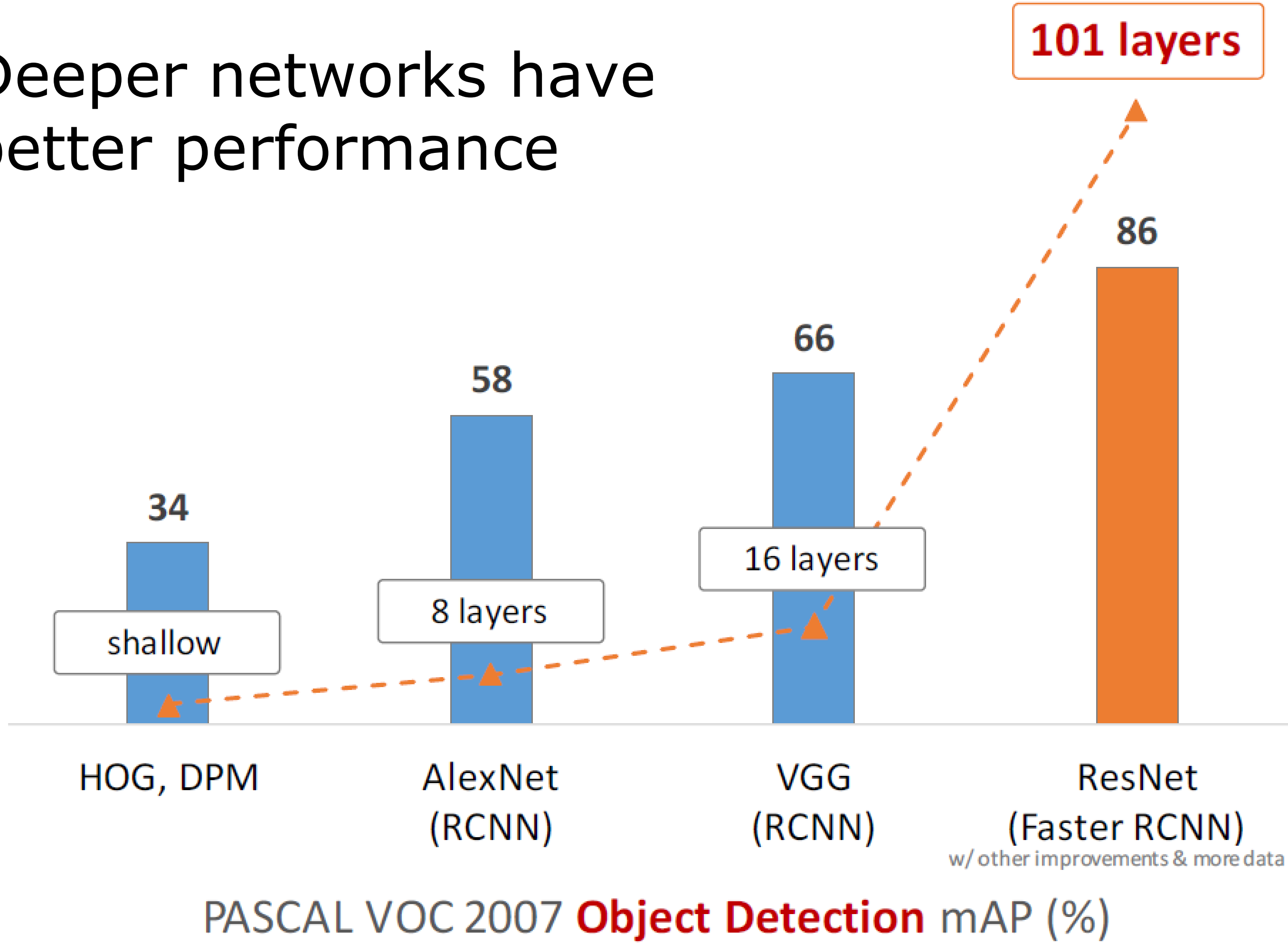


Faster R-CNN + ResNet Object Detection Result



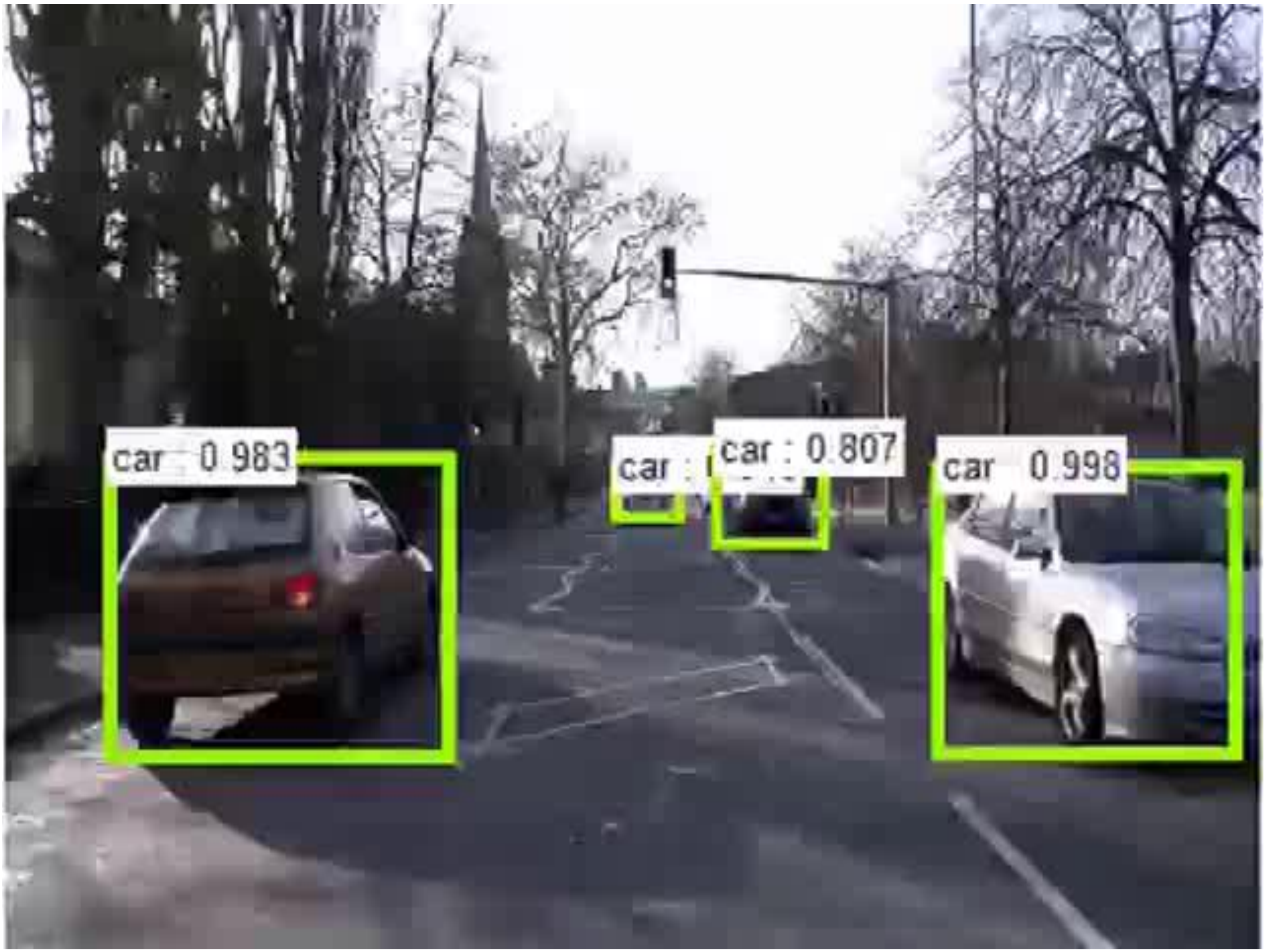
Object Detection Performance

- Deeper networks have better performance



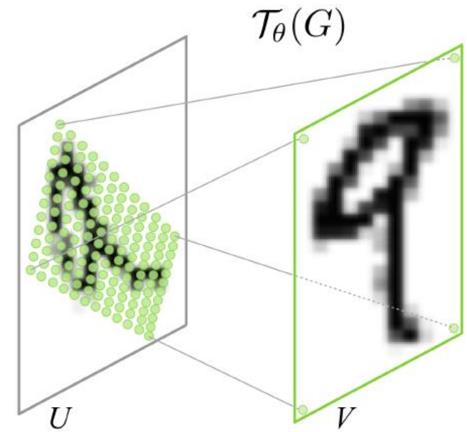
[He et al. CVPR 2016]

Faster R-CNN + ResNet Object Detection in Video

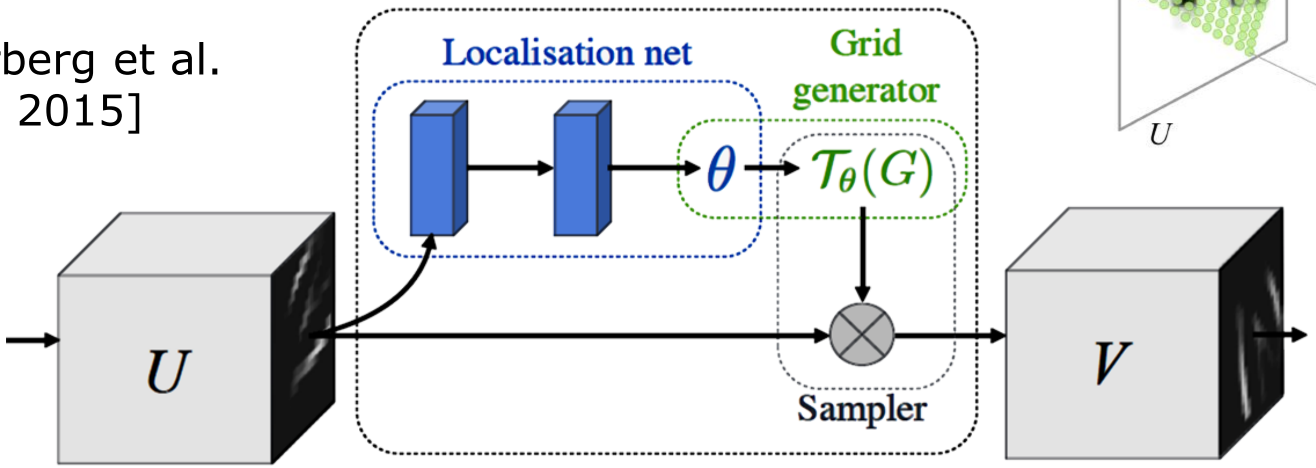


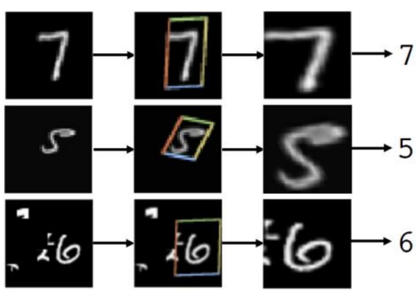
Spatial Transformer Networks

- Localization network estimates transformation parameters θ
- Grid generator computes sampling locations
- Sampler cuts out image parts

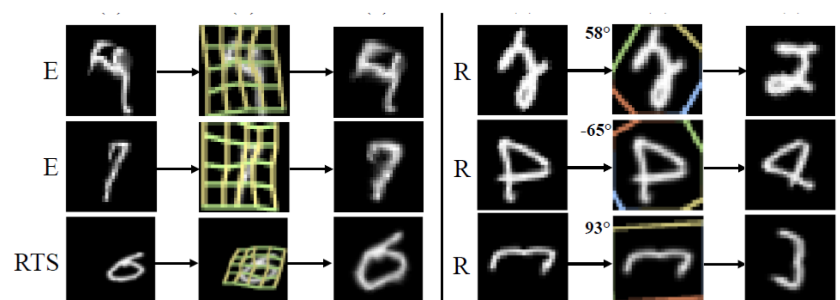


[Jaderberg et al. NIPS 2015]

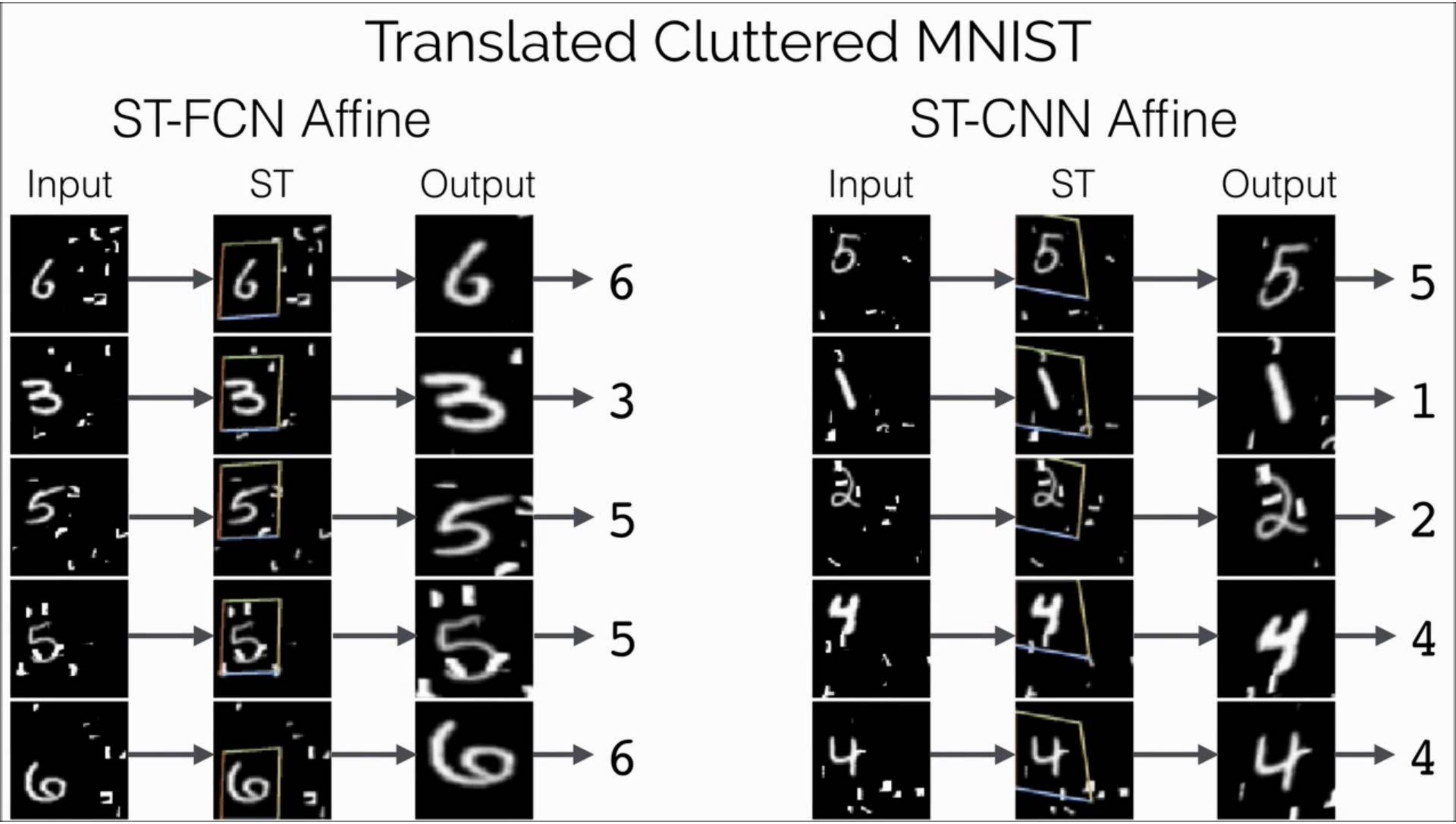




Model		R	RTS	P	E
		MNIST Distortion			
FCN		2.1	5.2	3.1	3.2
CNN		1.2	0.8	1.5	1.4
ST-FCN	Aff	1.2	0.8	1.5	2.7
	Proj	1.3	0.9	1.4	2.6
	TPS	1.1	0.8	1.4	2.4
ST-CNN	Aff	0.7	0.5	0.8	1.2
	Proj	0.8	0.6	0.8	1.3
	TPS	0.7	0.5	0.8	1.1



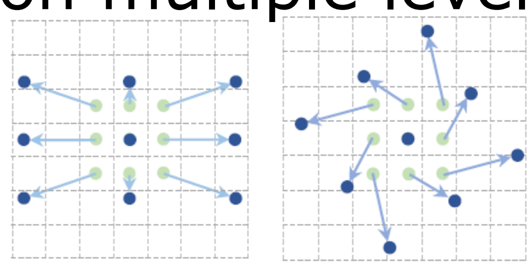
Spatial Transformer Networks



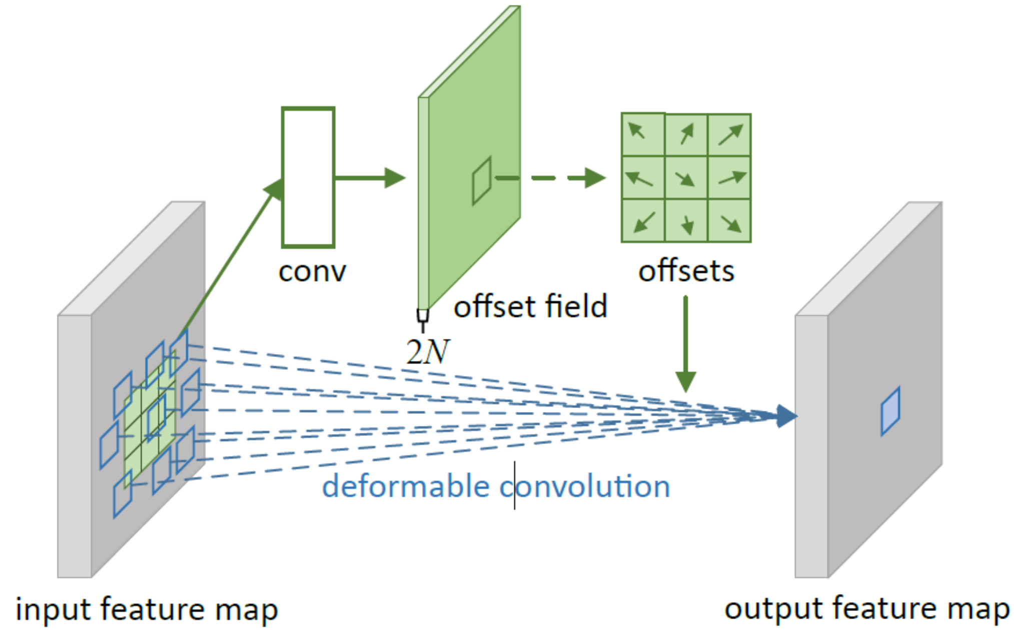
[Jaderberg et al. NIPS 2015]

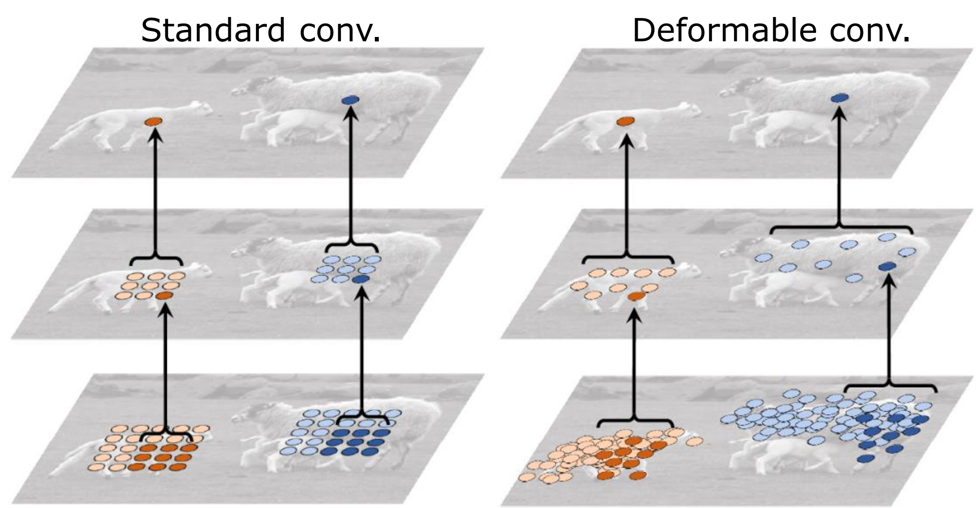
Deformable Convolutional Networks

- Similar to spatial transformer networks, but locally within a CNN
- Local distortions on multiple levels



- Flexible receptive field





[Dai et al. 2017]

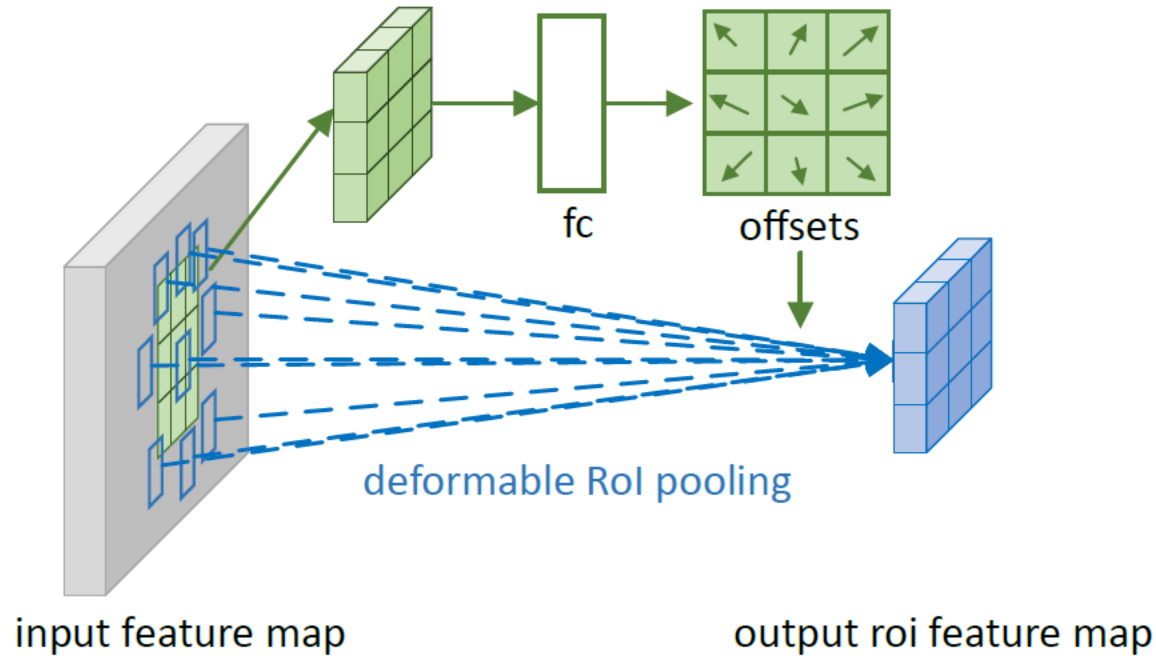
Deformable Convolutional Network

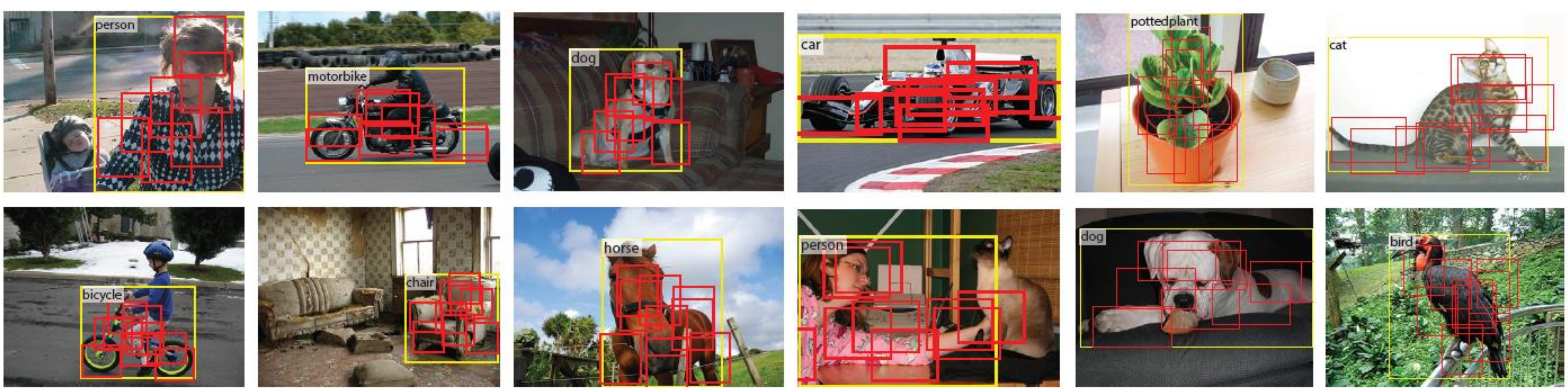
[Dai et al. 2017]

Deformable Convolutional Networks

- After convolutions to cut out an object
- Part placement is adapted to input

[Dai et al. 2017]

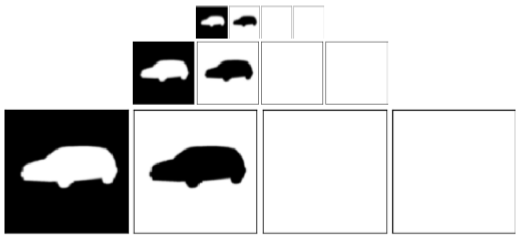




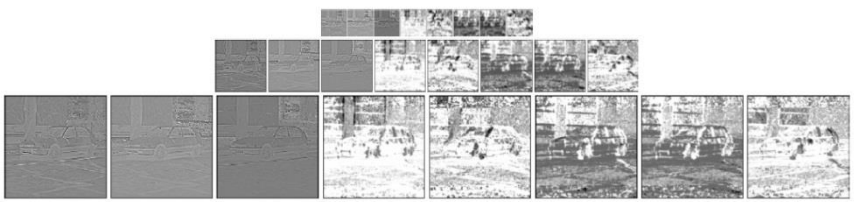
Object-class Segmentation

[Schulz, Behnke 2012]

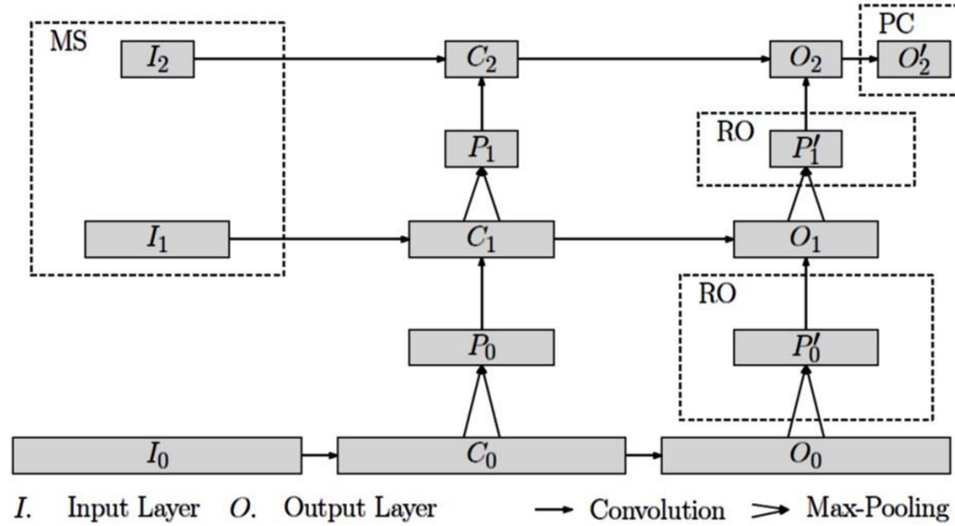
- Class annotation per pixel

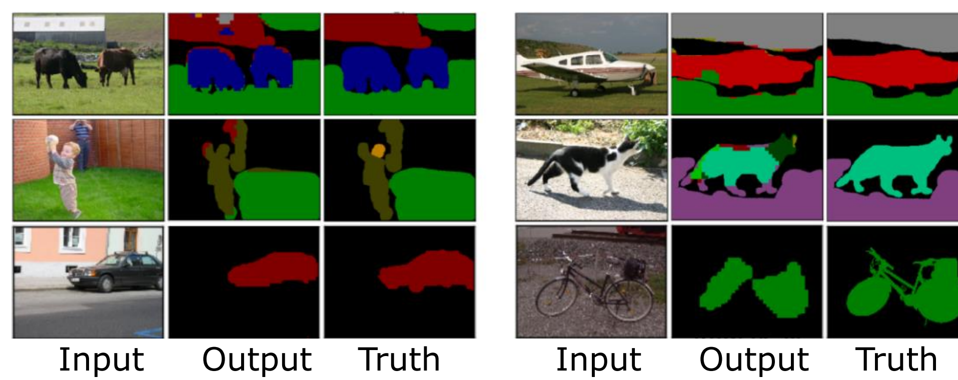


- Multi-scale input channels



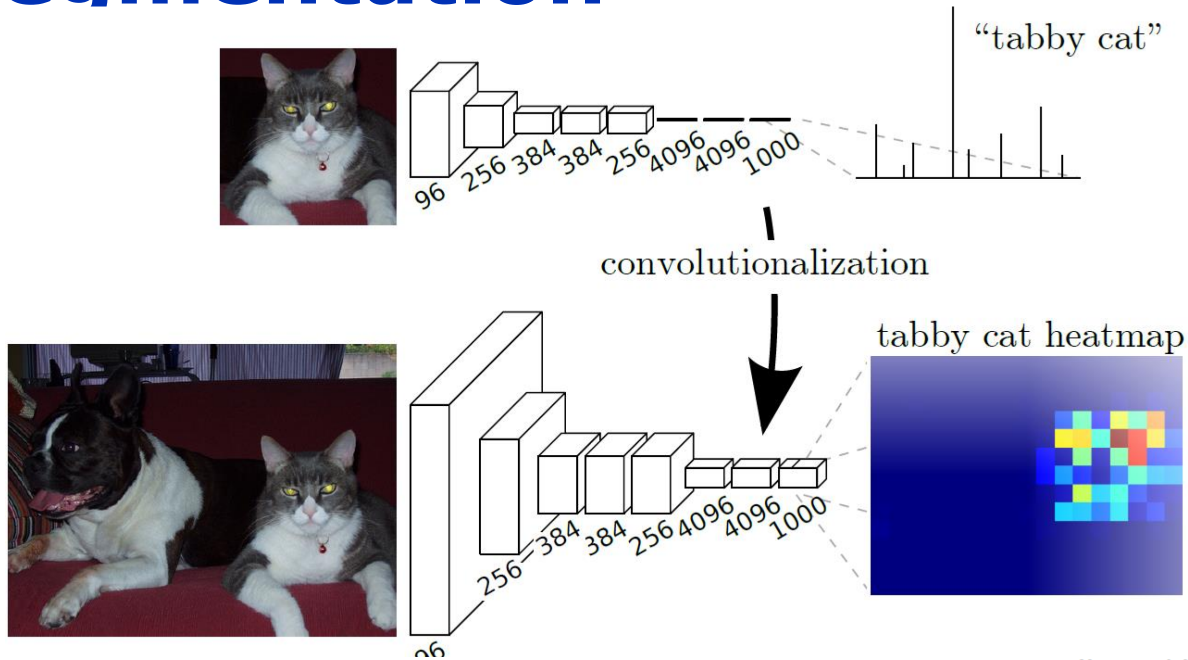
- Evaluated on MSRC-9/21 and INRIA Graz-02 data sets

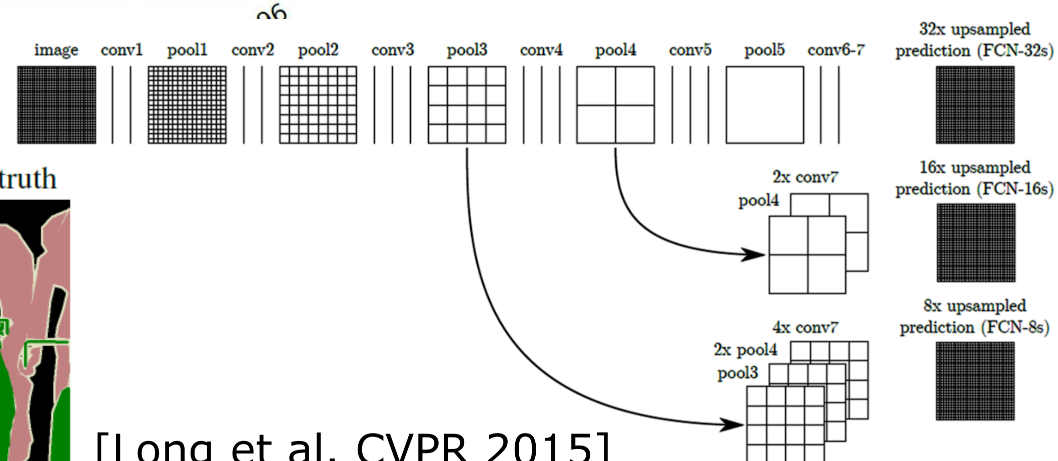


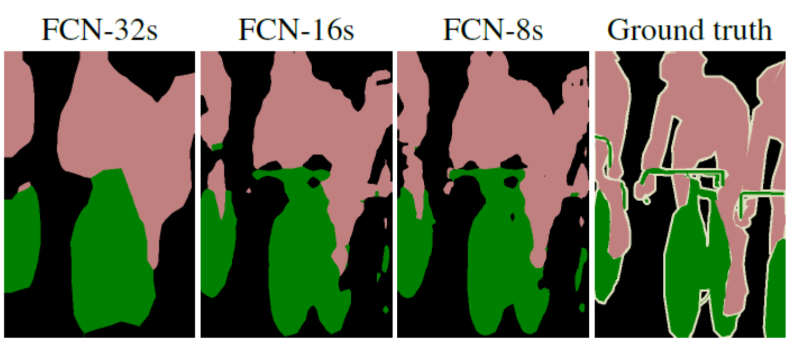


Fully Convolutional Networks for Semantic Segmentation

- Apply classification network at all image locations
- Problem: coarse output resolution
- Idea: Upsampling, use features from finer resolutions



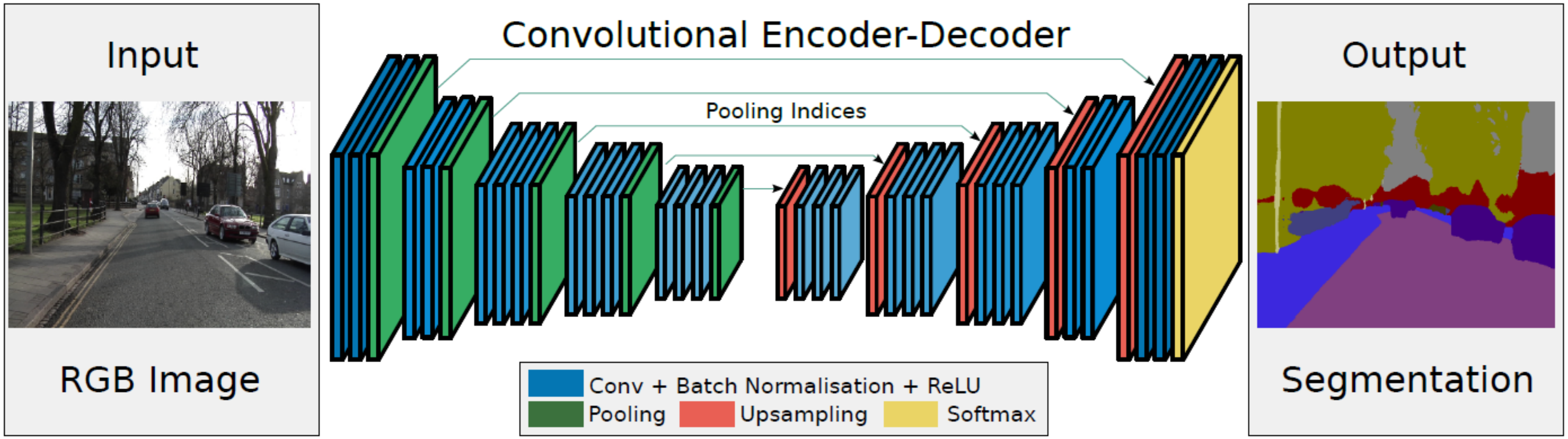


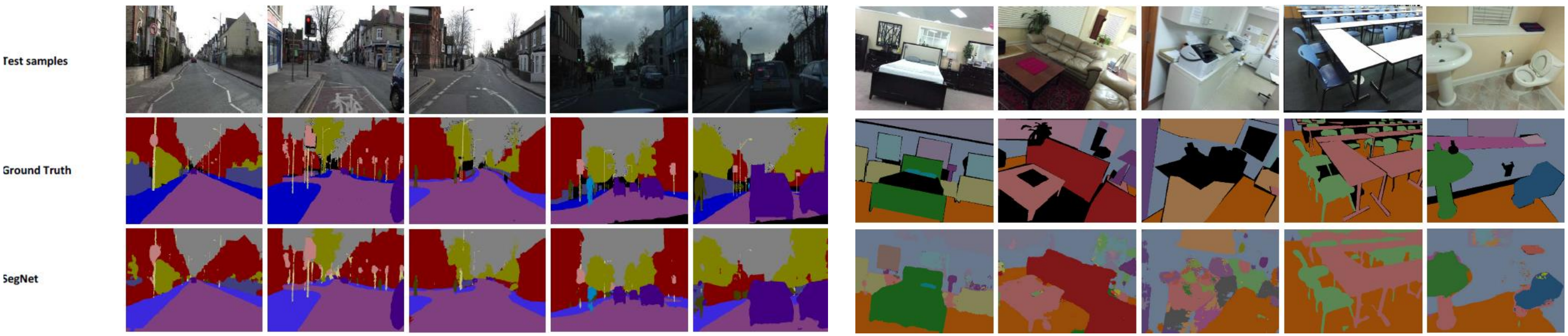


[Long et al. CVPR 2015]

SegNet: Encoder-Decoder

- Use pooling indices for upsampling



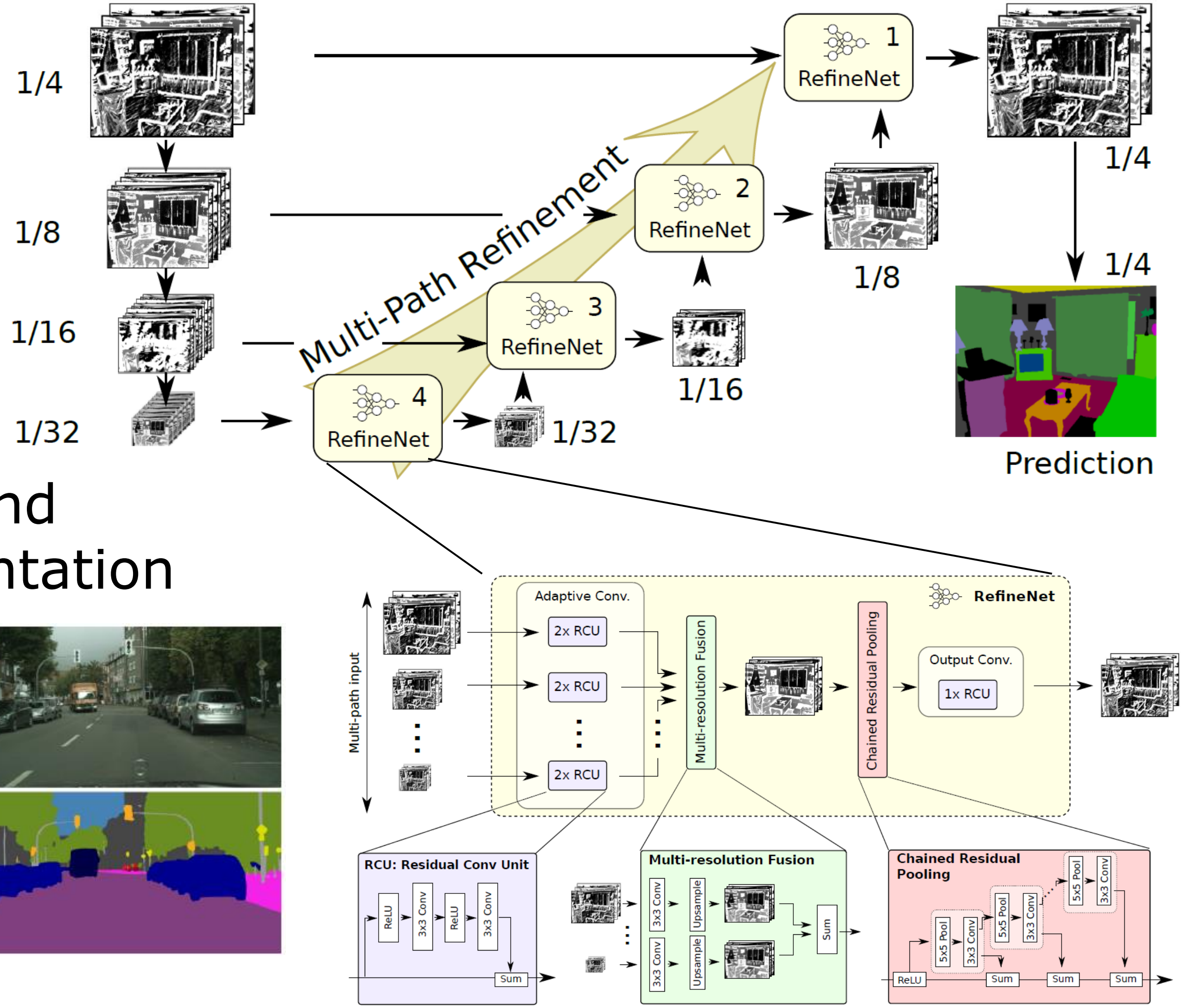


[Badrinarayanan et al. PAMI 2017]

RefineNet

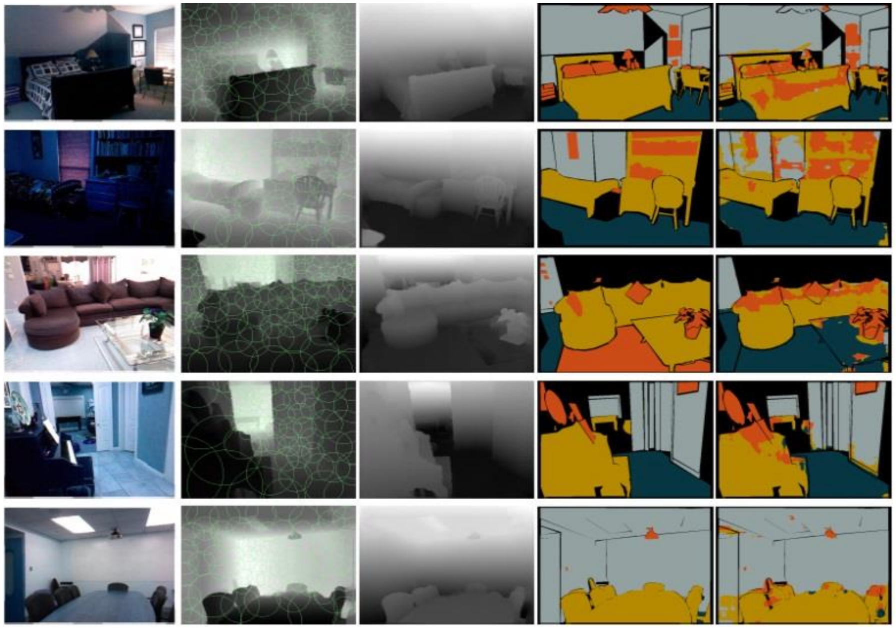
[Lin et al. CVPR 2017]

- Increase resolution by using features from the higher resolution
- Corse-to-fine
- Object parsing and semantic segmentation



RGB-D Object-Class Segmentation

- Kinect-like sensors provide dense depth
- Scale input according to depth, compute pixel height above floor



RGB Depth Height Truth Output

NYU Depth V2

Method	floor	struct	furnit	prop	Class Avg.	Pixel Acc.
CW	84.6	70.3	58.7	52.9	66.6	65.4
CW+DN	87.7	70.8	57.0	53.6	67.3	65.5
CW+H	78.4	74.5	55.6	62.7	67.8	66.5
CW+DN+H	93.7	72.5	61.7	55.5	70.9	70.5
CW+DN+H+SP	91.8	74.1	59.4	63.4	72.2	71.9
CW+DN+H+CRF	93.5	80.2	66.4	54.9	**73.7**	**73.4**
Müller et al.[8]	94.9	78.9	71.1	42.7	71.9	72.3
Random Forest [8]	90.8	81.6	67.9	19.9	65.1	68.3
Couprie et al.[9]	87.3	86.1	45.3	35.5	63.6	64.5
Höft et al.[10]	77.9	65.4	55.9	49.9	62.3	62.0
Silberman [12]	68	59	70	42	59.7	58.6

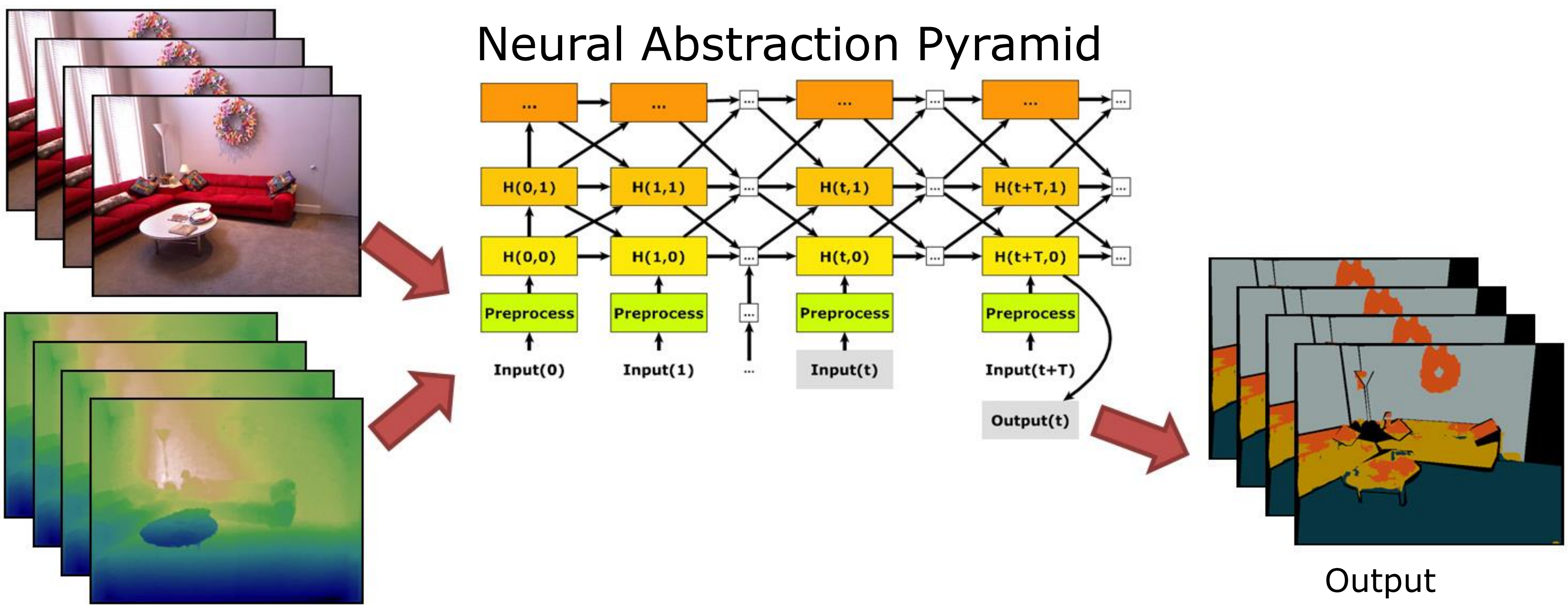
CW is covering windows, H is height above ground, DN is depth normalized patch sizes. SP is averaged within superpixels and SVM-reweighted. CRF is a conditional random field over superpixels [8]. Structure class numbers are optimized for class accuracy.

[Schulz, Höft, Behnke, ESANN 2015]

Neural Abstraction Pyramid for RGB-D Video Object-class Segmentation

- Input: RGB-D-Video (NYU Depth V2)

RGB

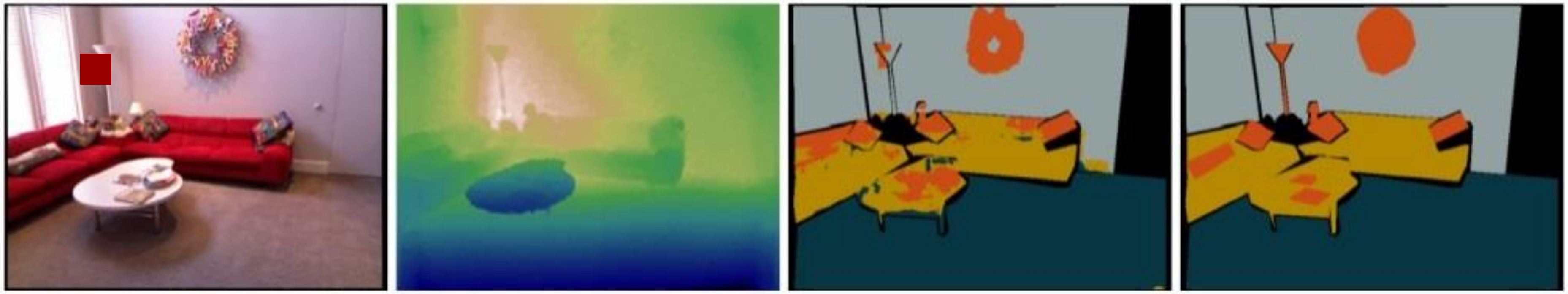


Depth

- Recursive computation is efficient for temporal integration

[Pavel, Schulz, Behnke, Neural Networks 2017]

Neural Abstraction Pyramid for RGB-D Video Object-class Segmentation



RGB Depth Output Truth

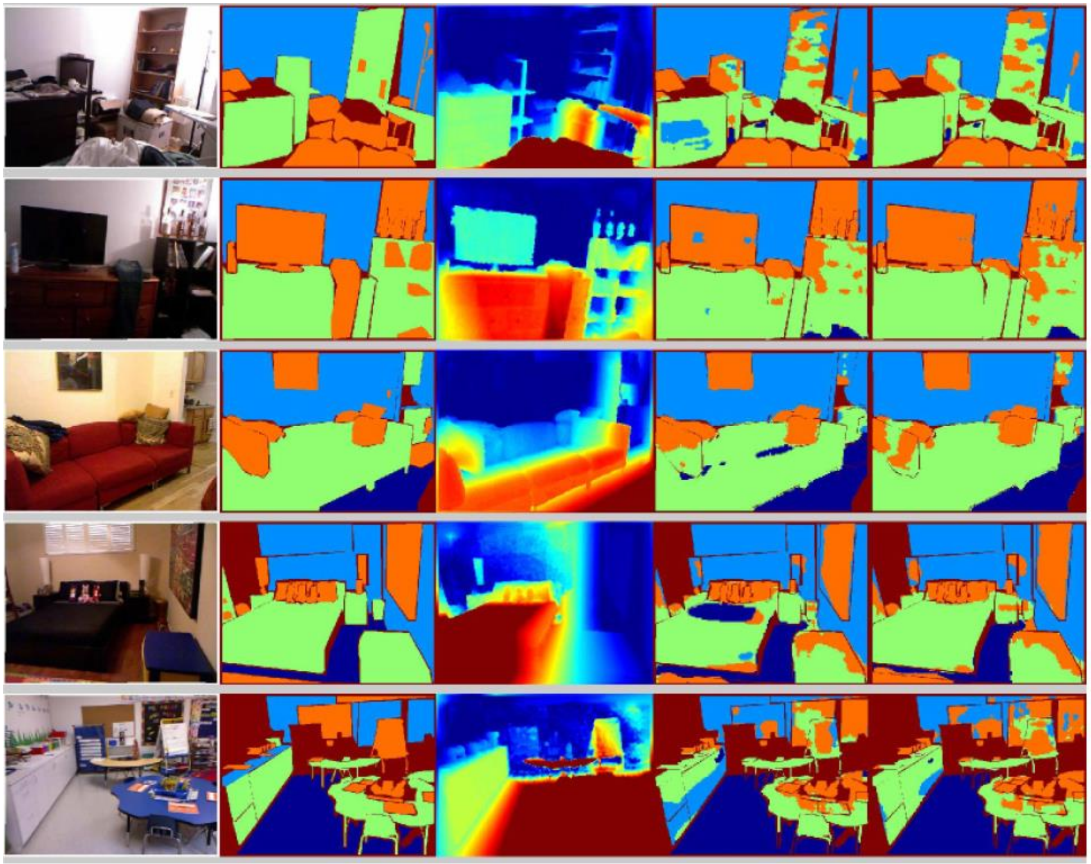
Method	Class Accuracies (%)				Average (%)	
	ground	struct	furnit	prop	Class	Pixel
ours (CW, RGB-D only)	78.6	49.2	48.7	48.3	56.2	52.0
ours (CW)	95.8	74.6	54.2	**64.0**	72.1	68.6
ours (WI+CW)	94.9	76.8	65.5	60.8	74.5	73.1
ours (WI)	94.3	83.7	72.0	54.9	76.2	76.4
ours (WI+CW+CRF)	95.4	78.9	67.3	60.8	75.6	74.6
ours (WI+CRF)	94.2	83.9	**72.0**	56.3	**76.6**	**77.2**
all-frames	**97.2**	70.0	51.1	56.0	68.6	64.6
Schulz et al. (2015a) (CNN+CRF)	93.6	80.2	66.4	54.9	73.7	73.4
Müller and Behnke (2014) (RF+CRF)	94.9	78.9	**71.1**	42.7	71.9	72.3
Stückler et al. (2013) (RF+SLAM)	90.8	81.6	67.9	19.9	65.0	68.3
Couprie et al. (2013) (CNN)	87.3	**86.1**	45.3	35.5	63.5	64.5
Silberman et al. (2012) (RF)	68.0	59.0	70.0	42.0	59.6	58.6

[Pavel, Schulz, Behnke, Neural Networks 2017]

Geometric and Semantic Features for RGB-D Object-class Segmentation

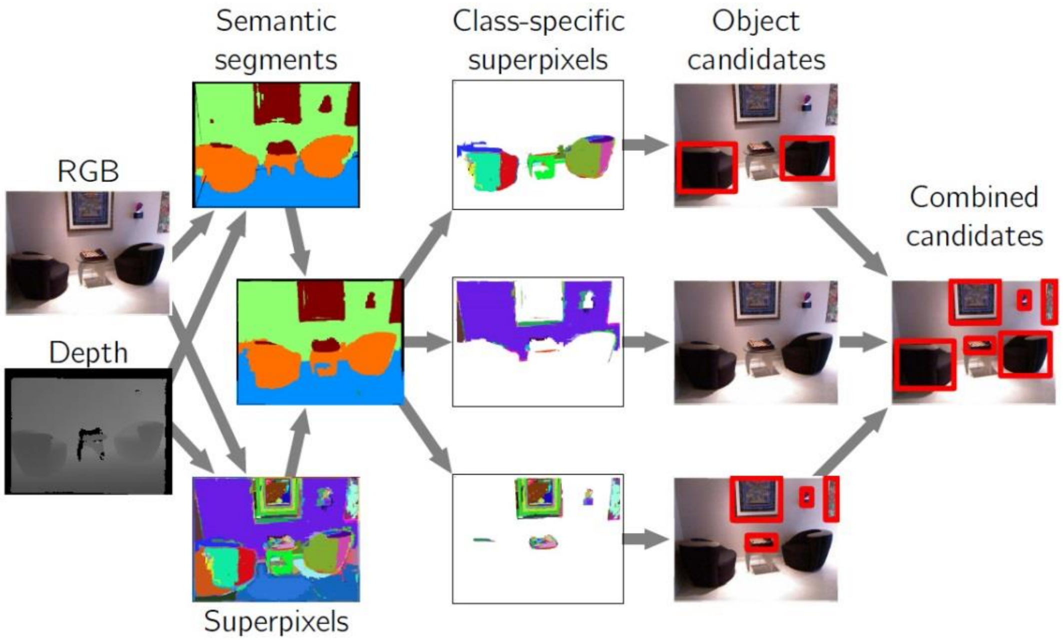
- New **geometric** feature: distance from wall
- **Semantic** features pretrained from ImageNet
- Both help significantly

[Husain et al. RA-L 2016]



Semantic Segmentation Priors for Object Discovery

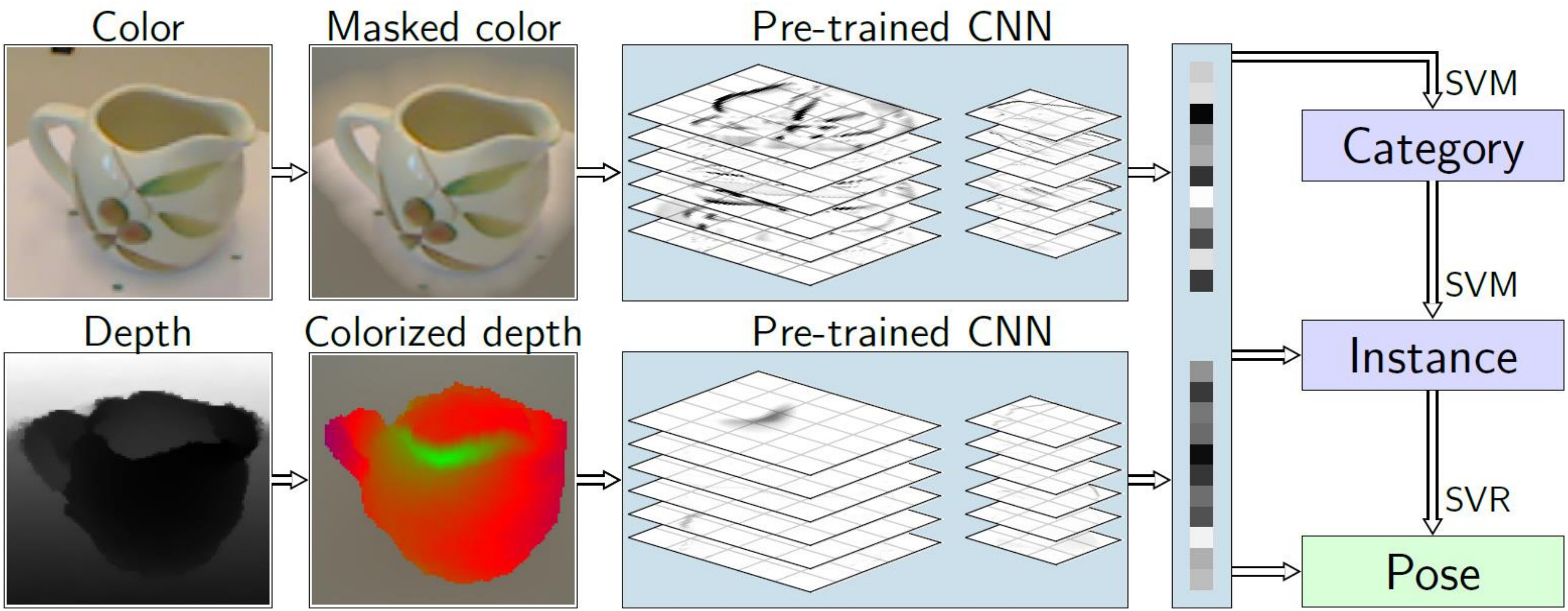
- Combine bottom-up object discovery and semantic priors
- Semantic segmentation used to classify color and depth superpixels
- Higher recall, more precise object borders



[Garcia et al. ICPR 2016]

RGB-D Object Recognition and Pose Estimation

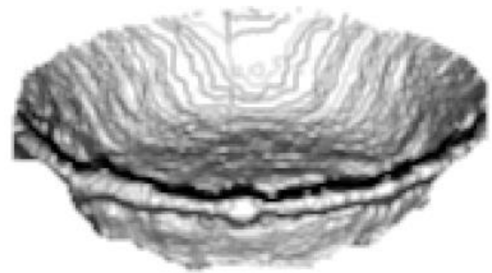
- Use pretrained features from ImageNet

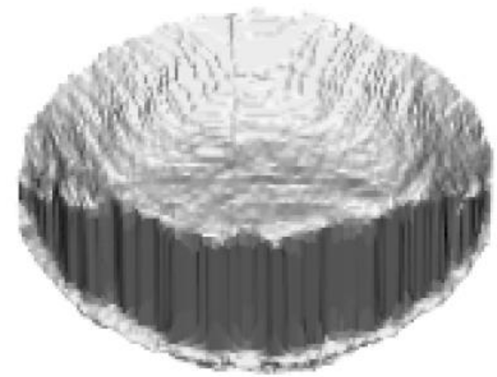


[Schwarz, Schulz, Behnke, ICRA2015]

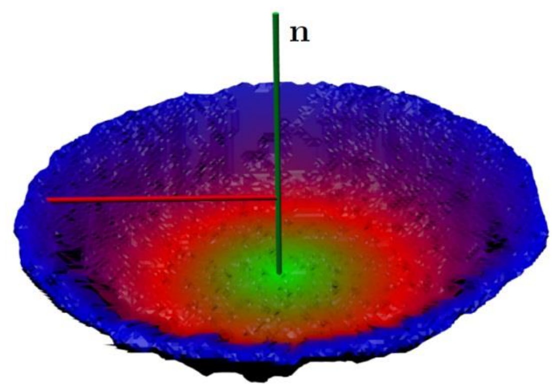
Canonical View, Colorization

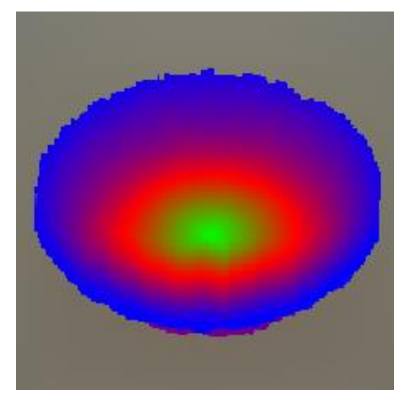
- Objects viewed from different elevation
- Render canonical view





- Colorization based on distance from center vertical

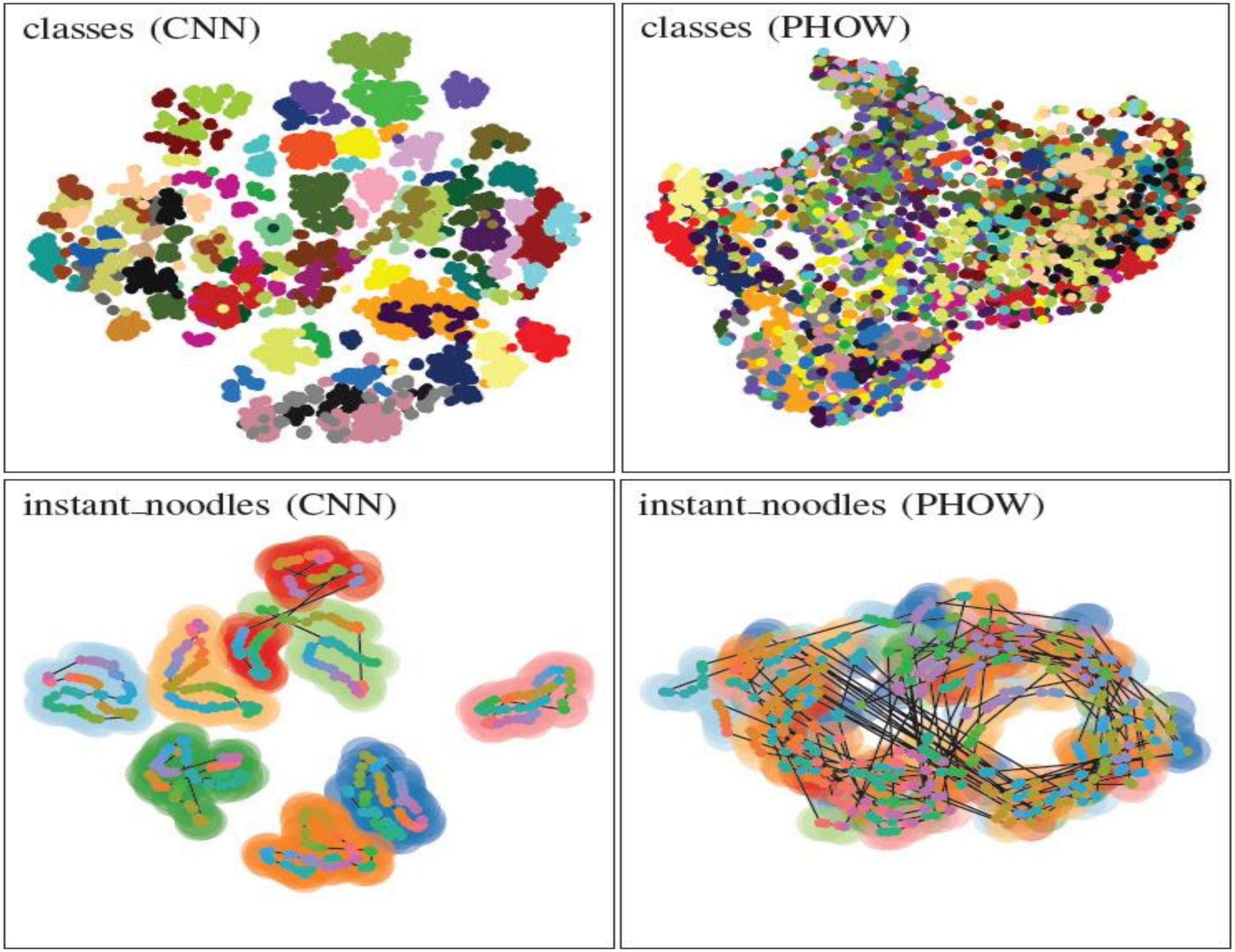




[Schwarz, Schulz, Behnke, ICRA2015]

Features Disentangle Data

- t-SNE embedding



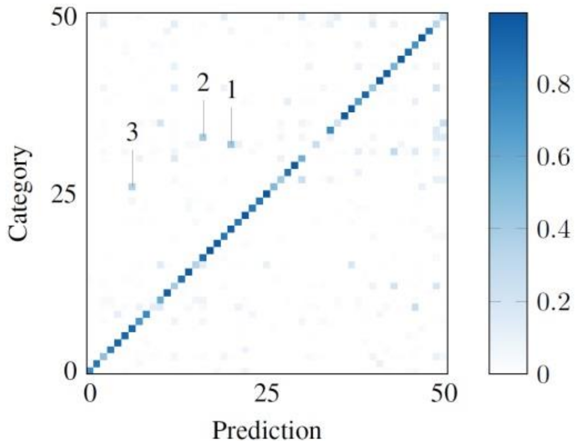
[Schwarz, Schulz, Behnke ICRA2015]

Recognition Accuracy

- Improved both category and instance recognition

	Category Accuracy (%)		Instance Accuracy (%)	
Method	RGB	RGB-D	RGB	RGB-D
Lai *et al.* [1]	74.3 ± 3.3	81.9 ± 2.8	59.3	73.9
Bo *et al.* [2]	82.4 ± 3.1	87.5 ± 2.9	**92.1**	92.8
PHOW[3]	80.2 ± 1.8	—	62.8	—
Ours	**83.1 ± 2.0**	88.3 ± 1.5	92.0	**94.1**
Ours	**83.1 ± 2.0**	**89.4 ± 1.3**	92.0	**94.1**

- Confusion

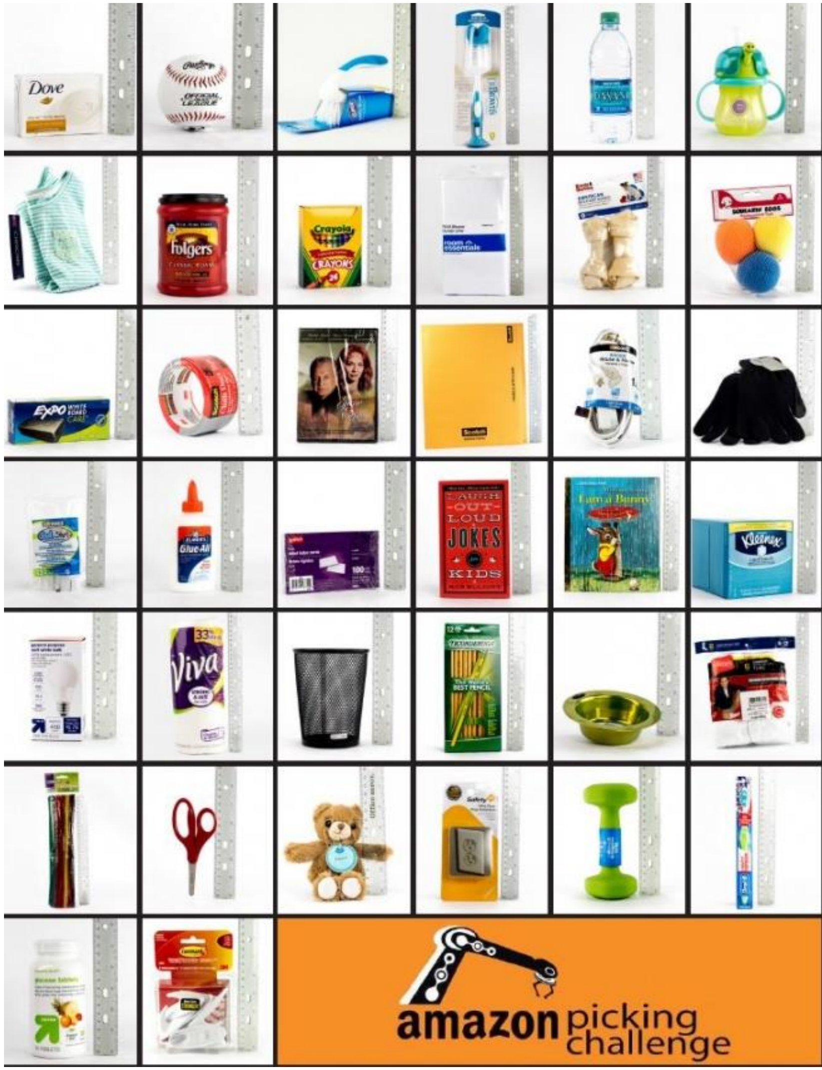


1: pitcher / coffe mug 2: peach / sponge

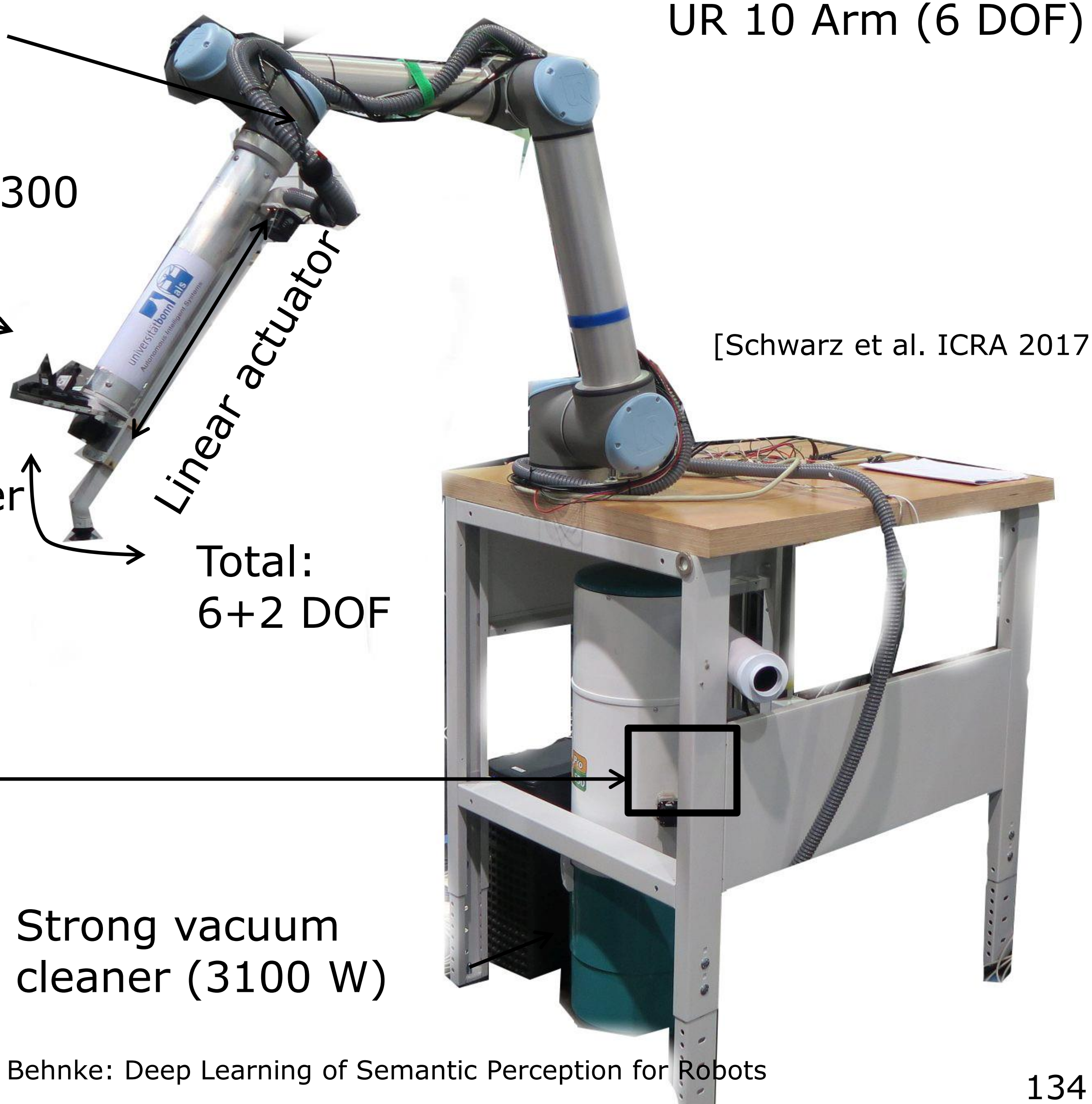
[Schwarz, Schulz, Behnke, ICRA2015]

Amazon Picking Challenge 2016

- Large variety of objects
- Different properties
  - Transparent
  - Shiny
  - Deformable
  - Heavy
- Stowing task
- Picking task



System



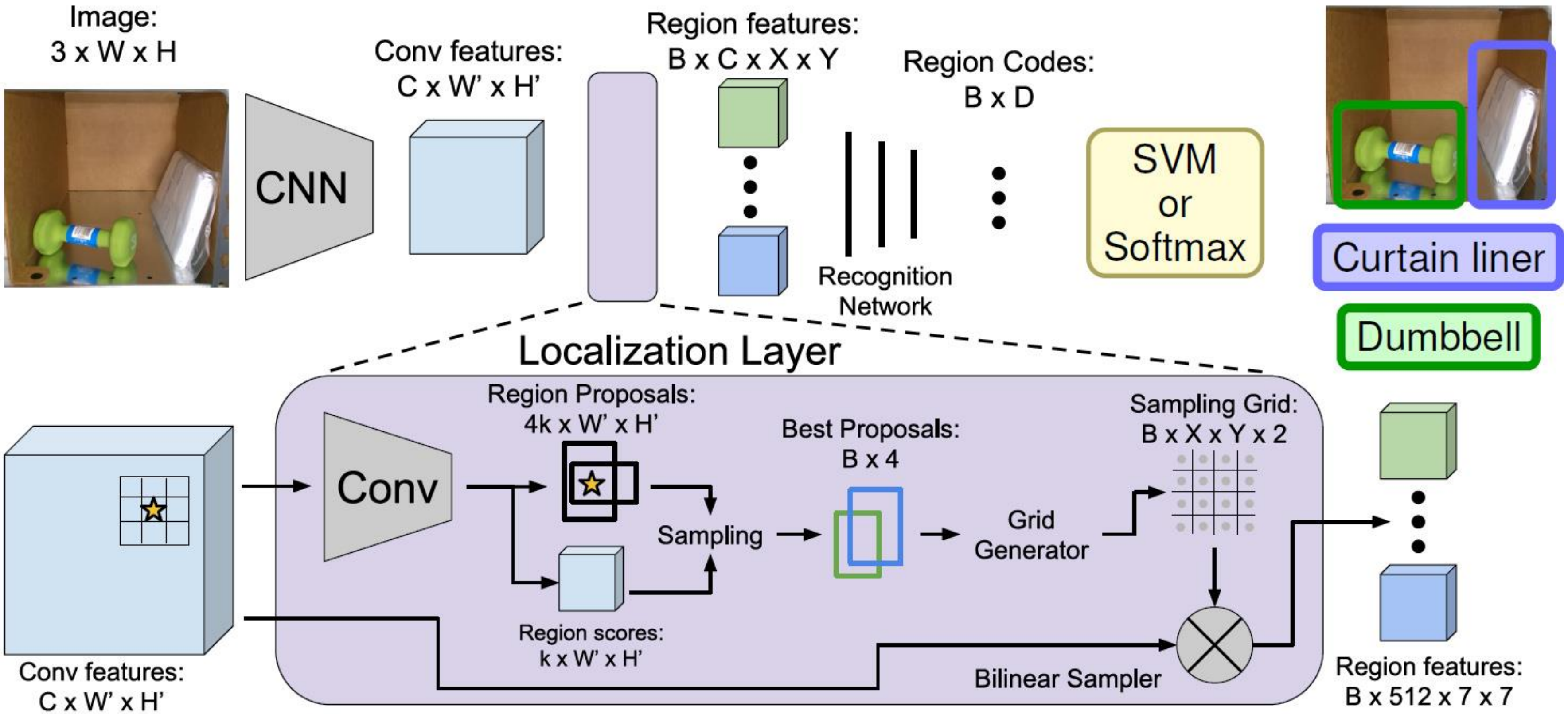
Suction strength control

RGB-D Cameras

- 2x Intel RealSense SR300
- Fusion of three depth estimates per pixel (including RGB stereo)

[Schwarz et al. ICRA 2017]

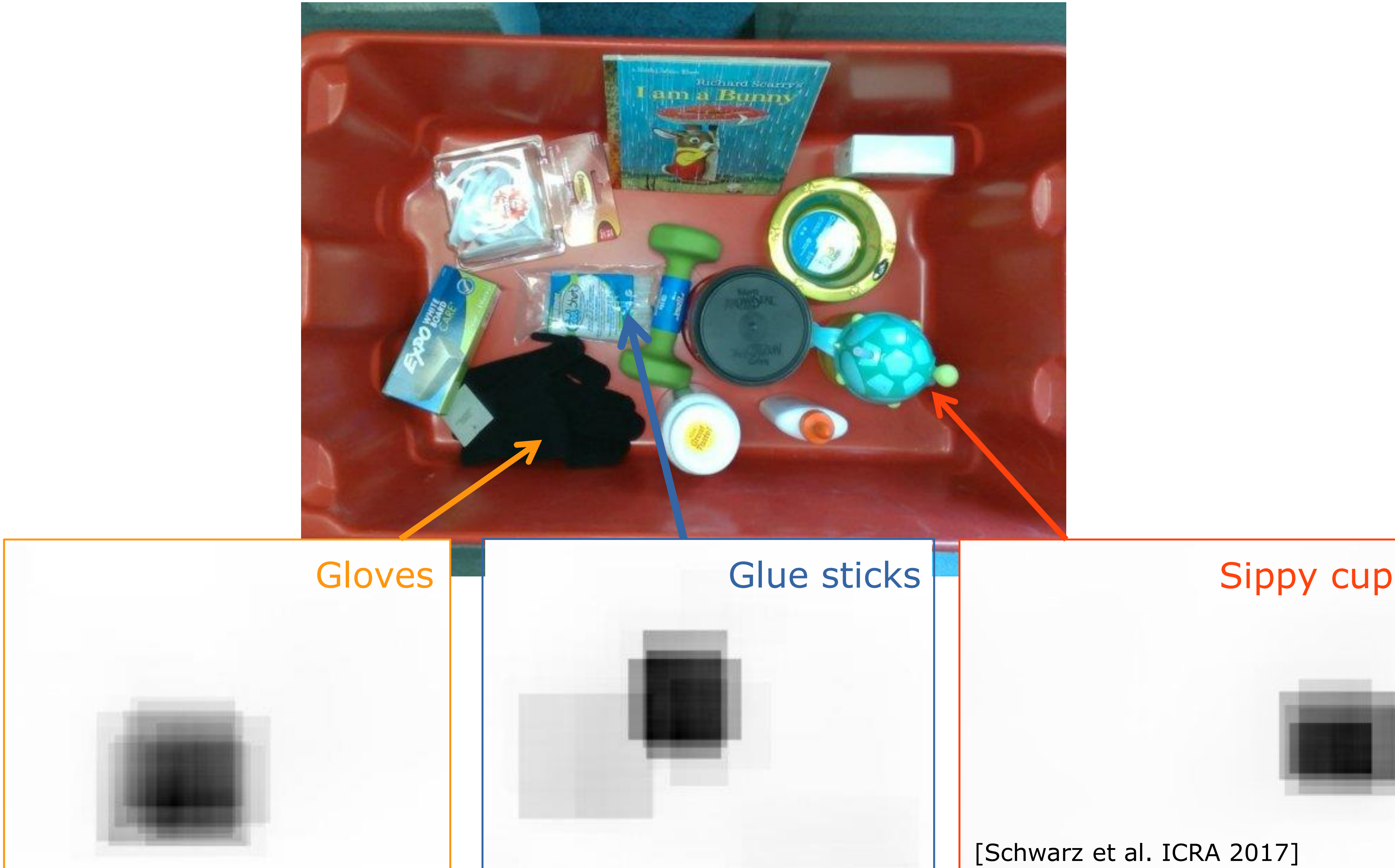
Object Detection



[Adapted from Johnson et al. CVPR 2016]

[Schwarz et al. ICRA 2017]

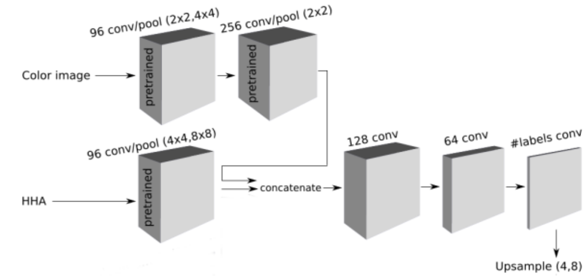
Example Detections



[Schwarz et al. ICRA 2017]

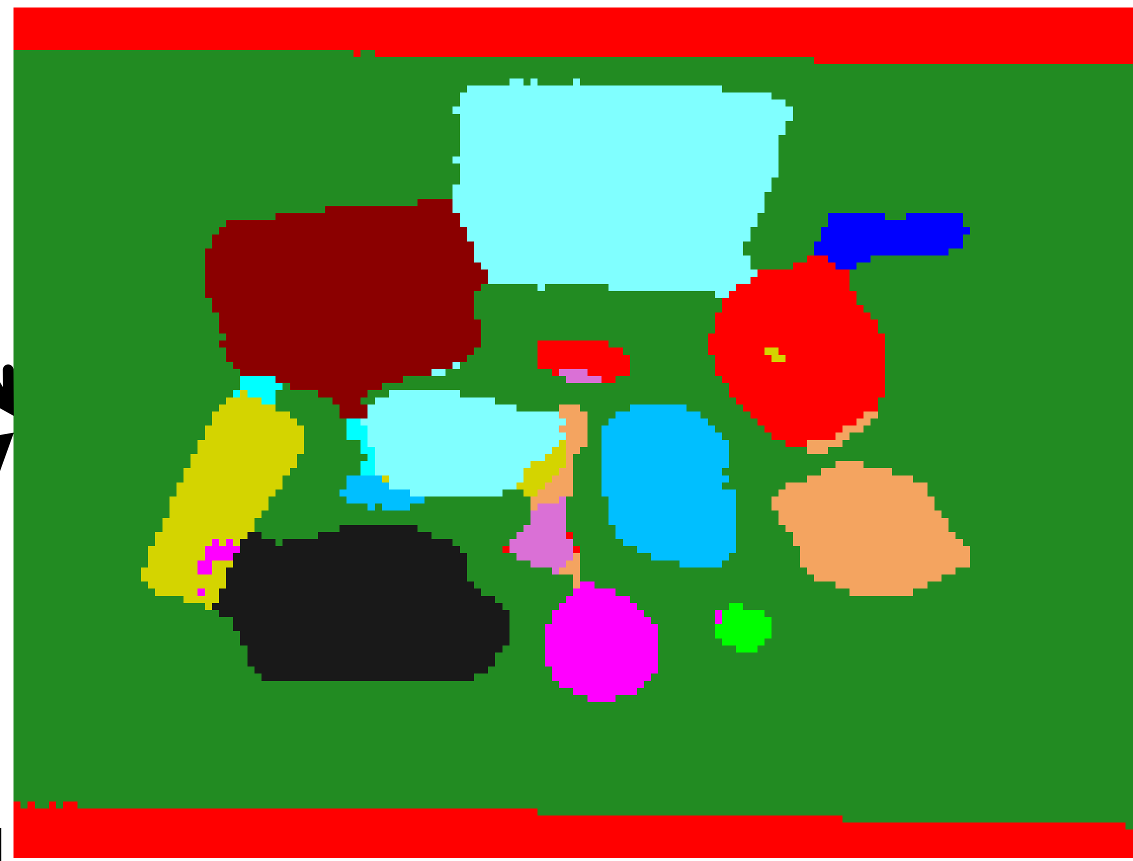
Semantic Segmentation

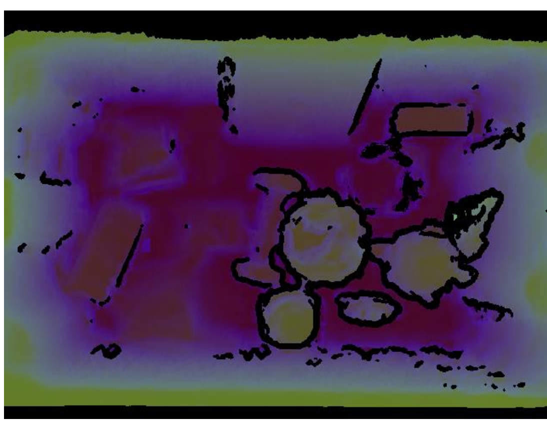
- Deep Convolutional Network



RGB

Result



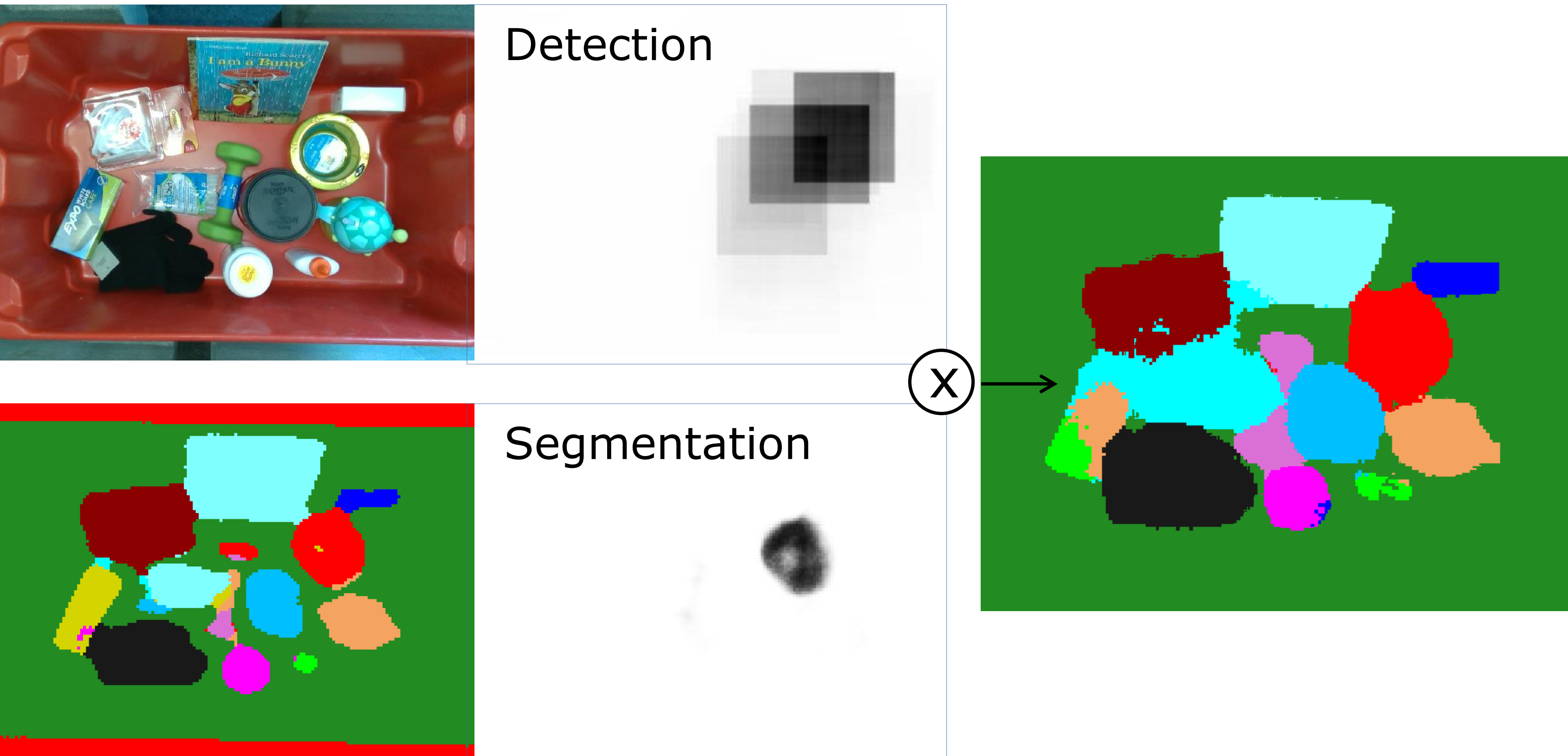


HHA

[Husain et al. RA-L 2016]

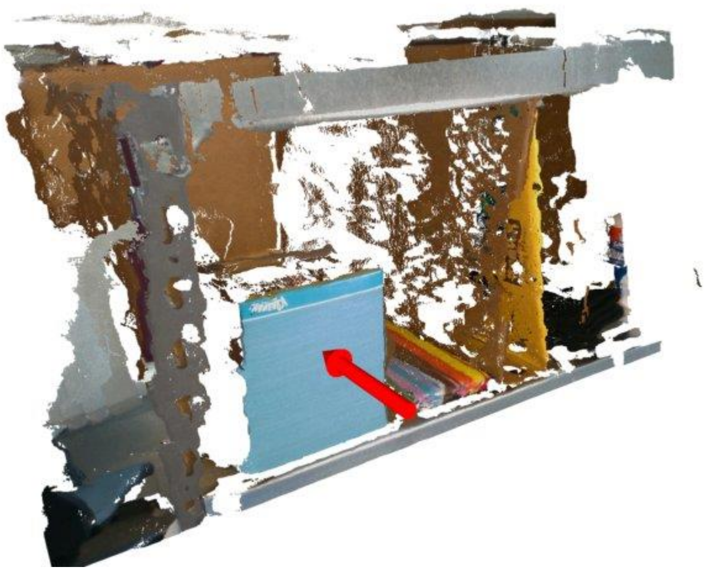
Combined Detection and Segmentation

- Pixel-wise multiplication

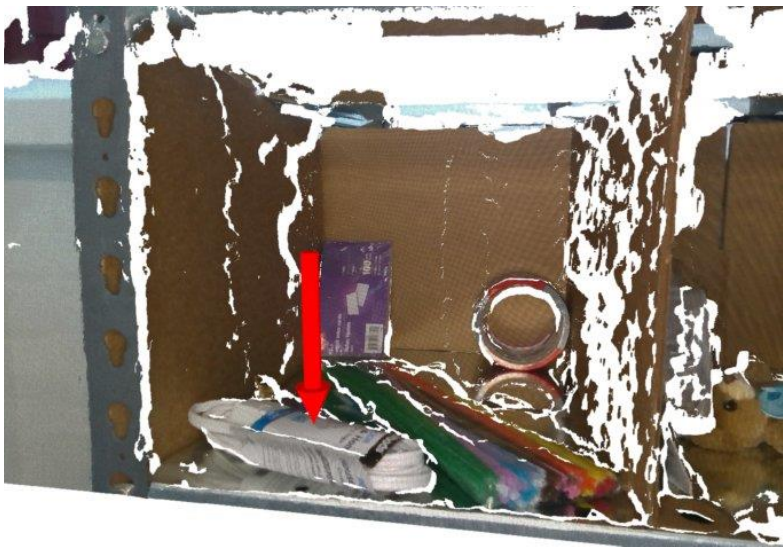


Grasp Pose Selection

- Center grasp for "standing" objects:
  - Find support area for suction close to bounding box center

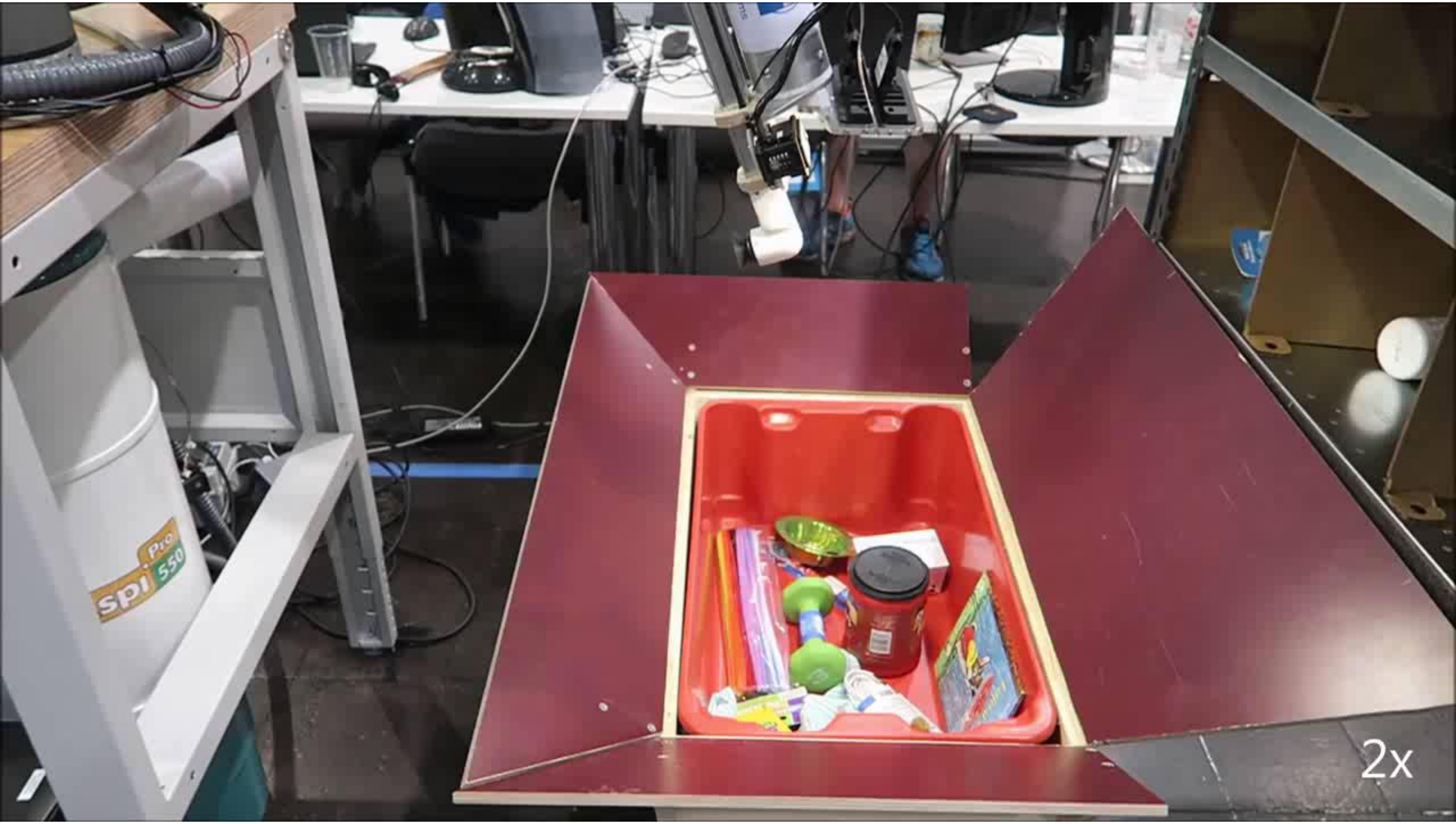


- Top grasp for "lying" objects:
  - Find support area for suction close to horizontal bounding box center



[Schwarz et al. ICRA 2017]

Example Stowing Top Grasp

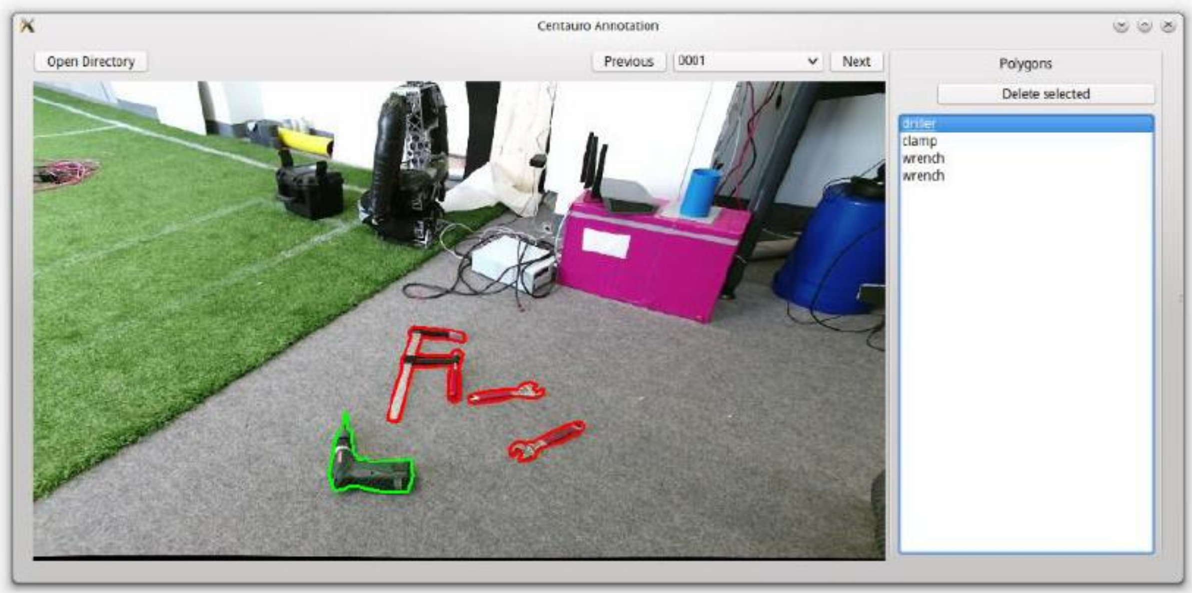


[Schwarz et al. ICRA 2017]

Example Picking Grasps

[Schwarz et al. ICRA 2017]

Workspace Perception Data Set

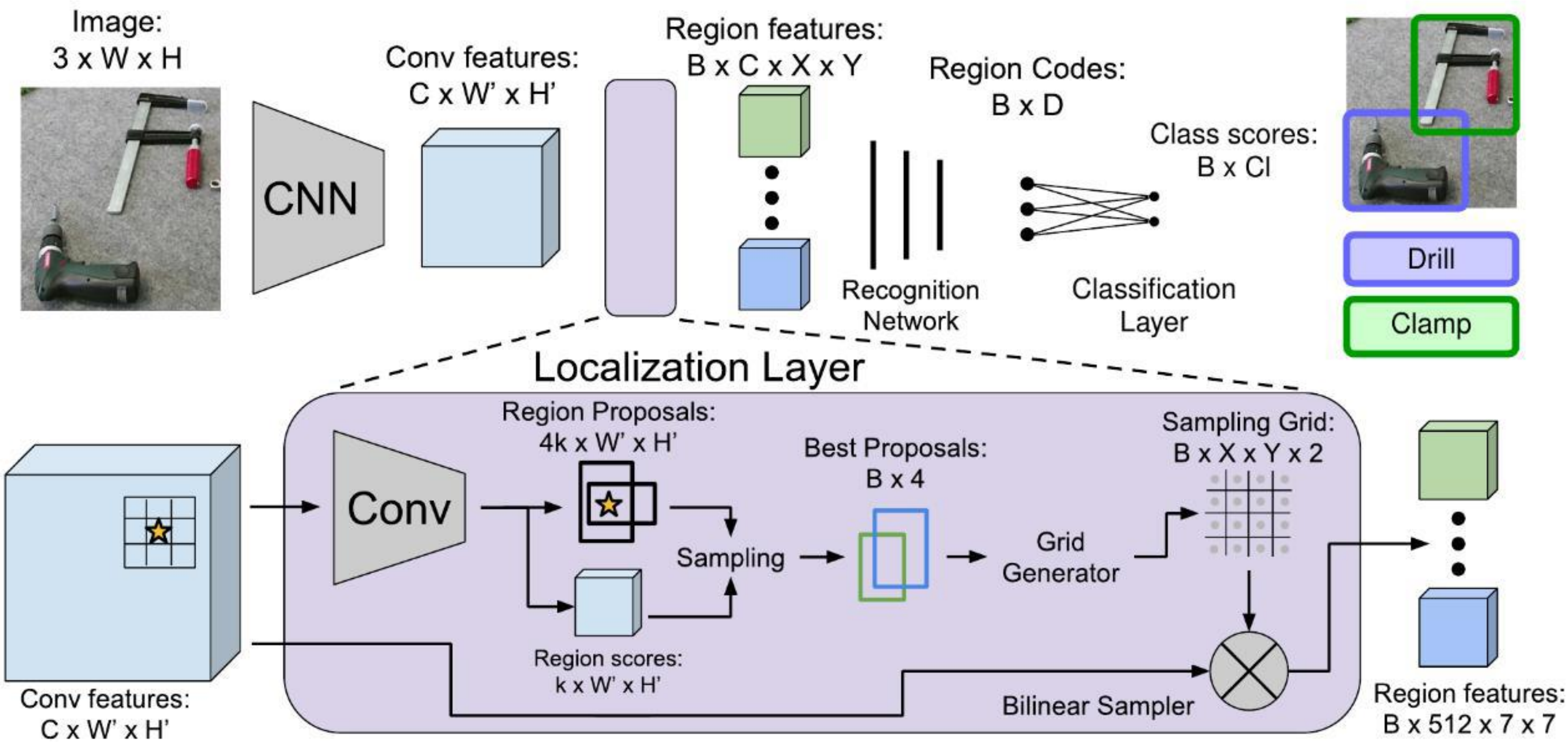


129 frames, 6 object classes

https://www.centauro-project.eu/data_multimedia/tools_data

Deep Learning Object Detection

- Adapted DenseCap [Johnson et al. 2015] pipeline



- Transfer learning needs only few annotated images

[Schwarz et al. IJRR 2017]

Tool Detection Results

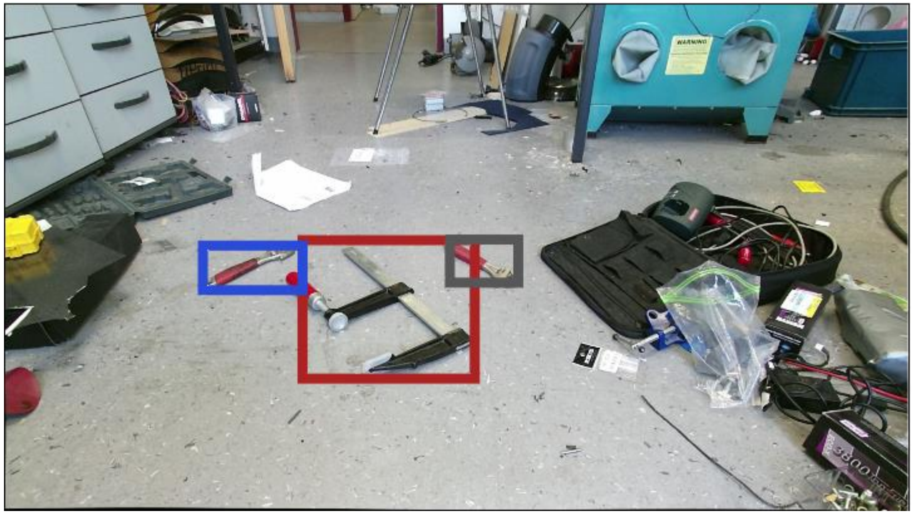
extension_box stapler driller clamp [background]

Resolution	Clamp	Door handle	Driller	Extension	Stapler	Wrench	Mean
	AP / F1	AP / F1	AP / F1	AP / F1	AP / F1	AP / F1	AP / F1
720×507	0.881/0.783	0.522/0.554	0.986/0.875	1.000/0.938	0.960/0.814	0.656/0.661	0.834/0.771
1080×760	0.926/0.829	0.867/0.632	0.972/0.893	1.000/0.950	0.992/0.892	0.927/0.848	0.947/0.841
1470×1035	0.913/0.814	0.974/0.745	1.000/0.915	1.000/0.952	0.999/0.909	0.949/0.860	0.973/0.866

[Schwarz et al. IJRR 2017]

Tool Detection Examples

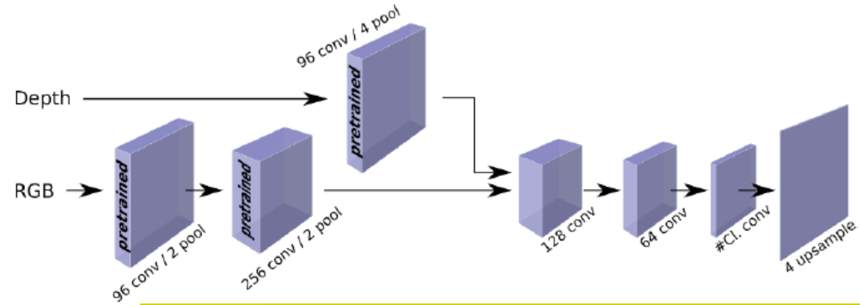


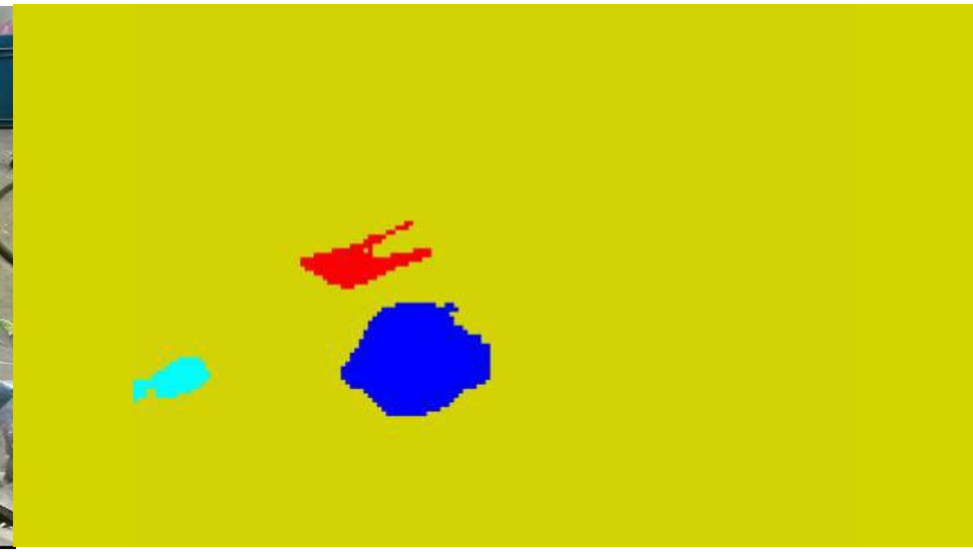



Semantic Segmentation

- Deep CNN

  [Husain et al. RA-L 2016]





Pixel-wise accuracy:

Clamp	Door handle	Driller	Extension	Stapler	Wrench	Background	Mean
0.727	0.751	0.769	0.889	0.775	0.734	0.992	0.805

MBZIRC Challenge 2

Wrench Perception

- DenseCap Object detection of mouth and ring
- Training set: 100 Stereo image pairs

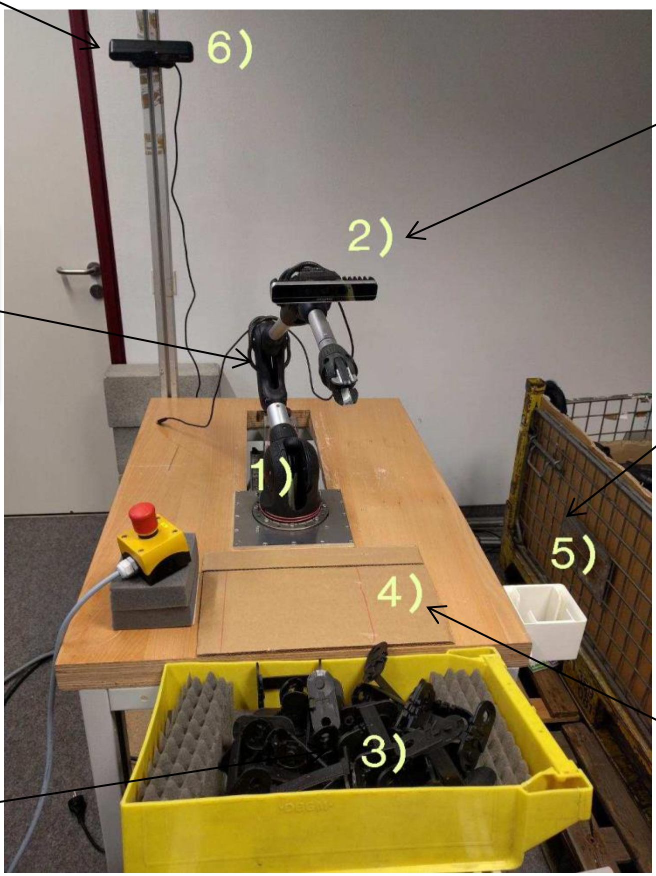
EuRoC Challenge 1: Robolink Feeder

ASUS Xtion RGB-D workspace camera

SR300 RGB-D wrist camera

Cable-driven 6DOF igus-robolink® manipulator

Energy chain part feeder



Place for regrasping

Pile of the chain parts

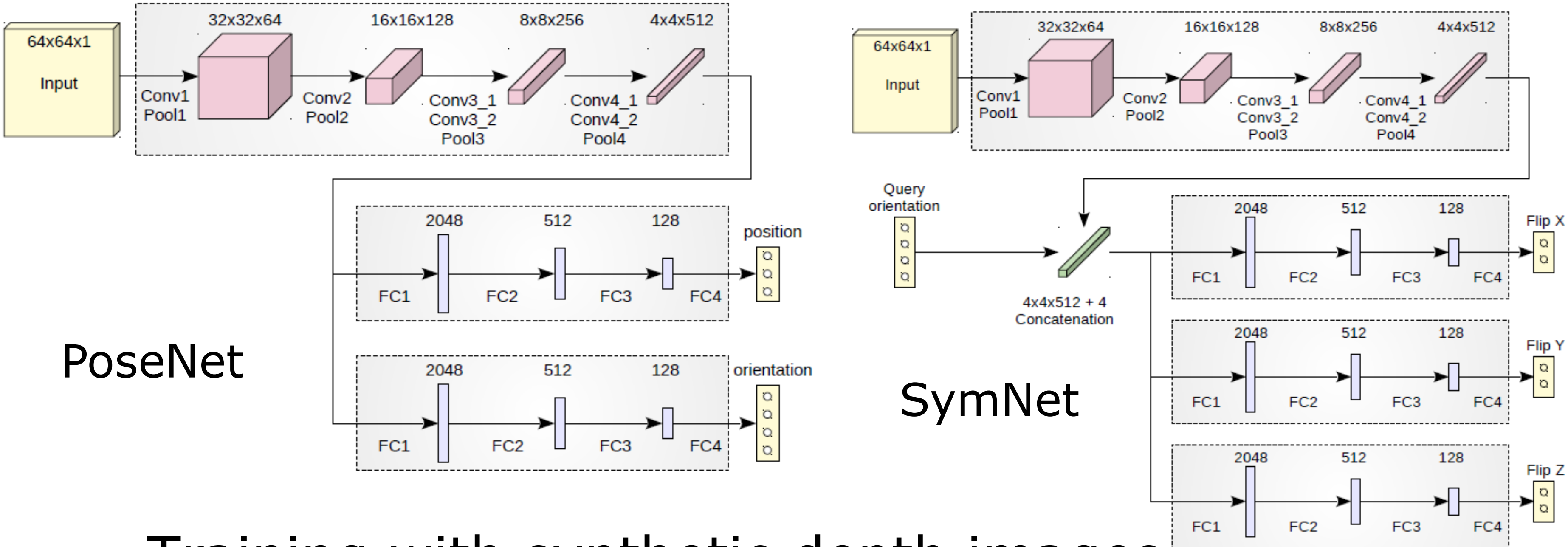
[Koo et al. CASE 2017]

Robolink Feeder: Bin Picking

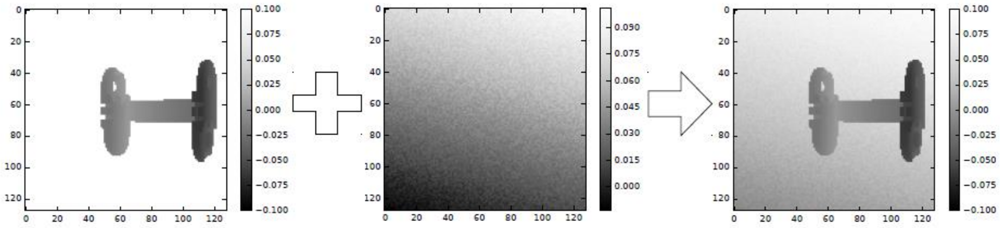
[Koo et al. CASE 2017]

Part Pose Estimation

- Two convolutional neural networks



- Training with synthetic depth images



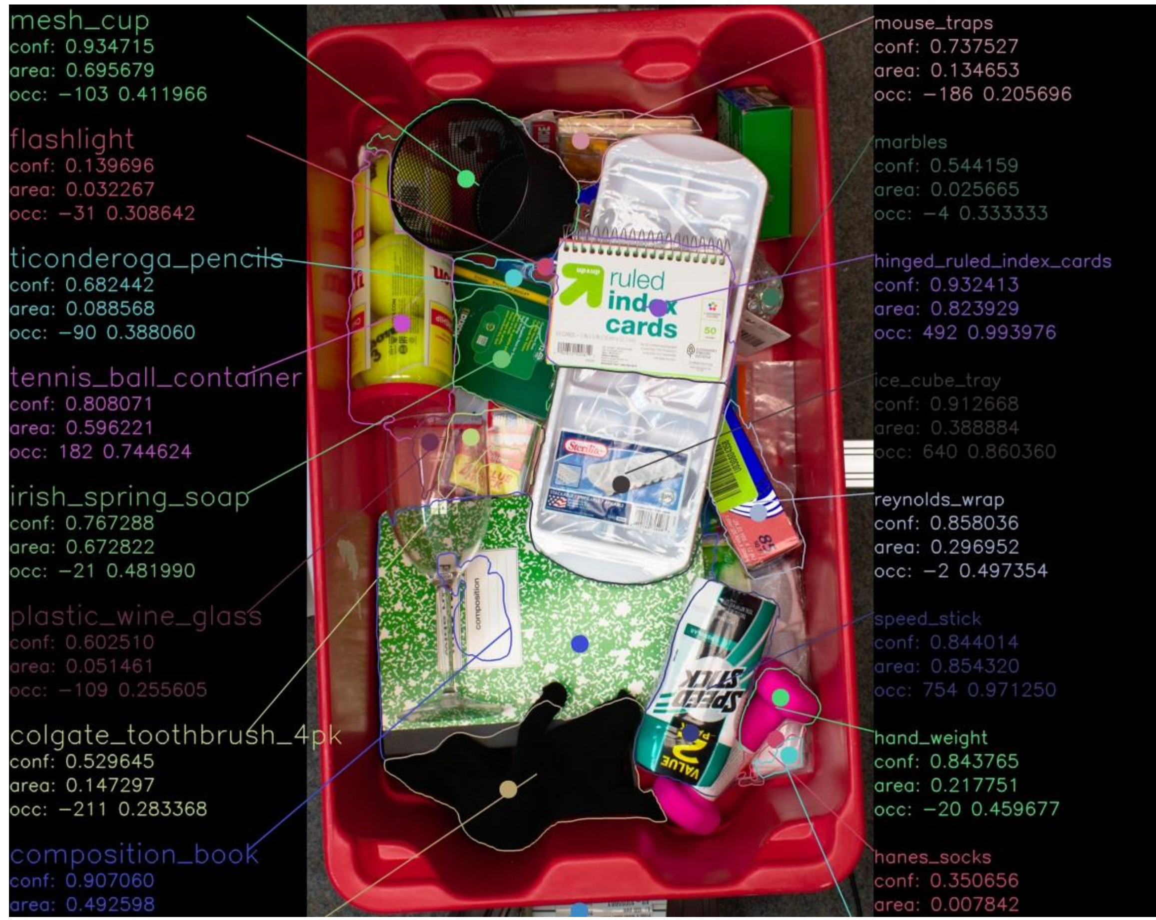
[Koo et al. CASE 2017]

Robolink Feeder: Regrasping and Placing

[Koo et al. CASE 2017]

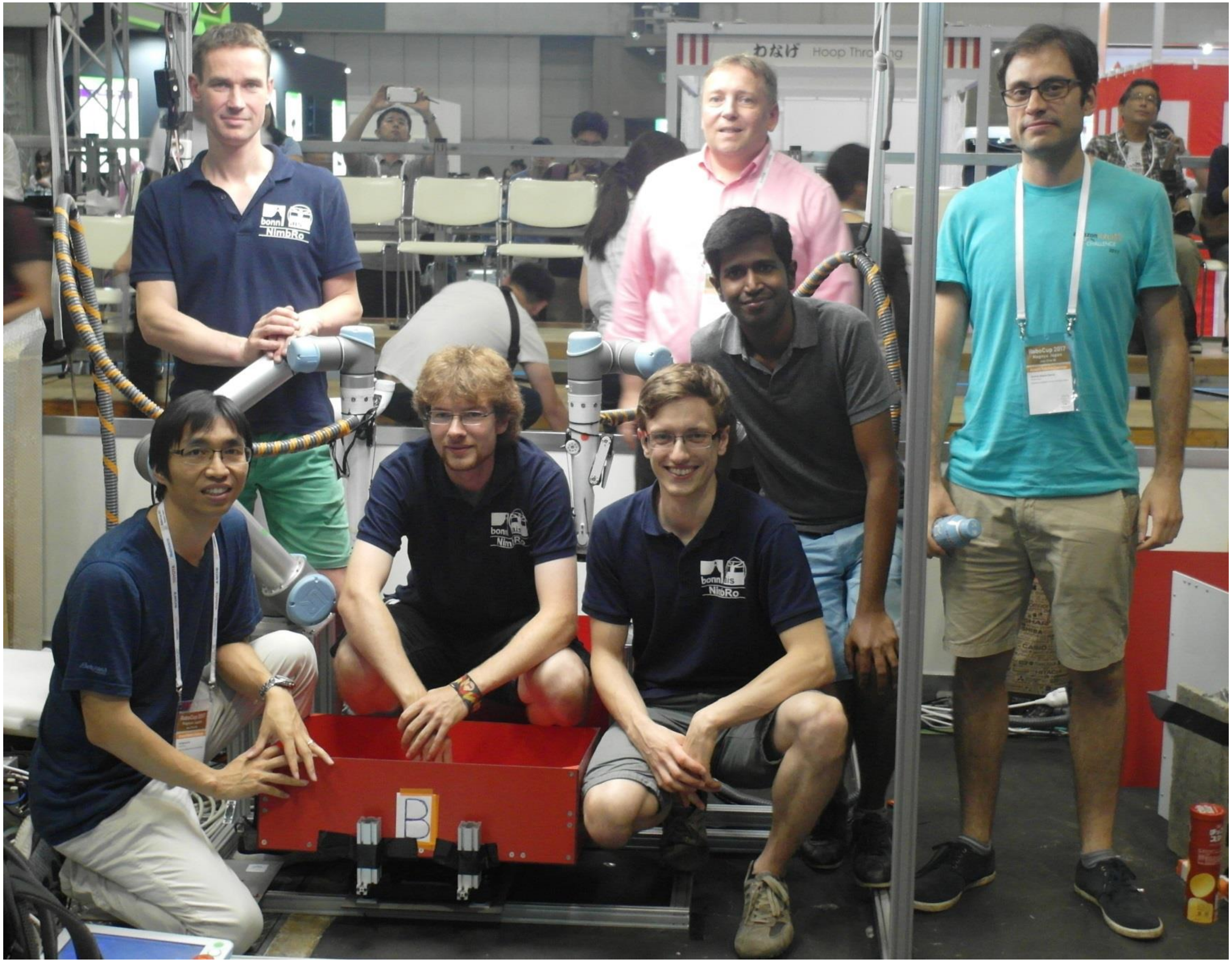
Amazon Robotics Challenge 2017

- Quick learning of novel objects
- Training with rendered scenes



Amazon Robotics Challenge 2017 Final

NimbRo Picking 2017 Team



Conclusion

- Flat models do not suffice
  - Jump from signal to symbols too large
- Deep learning helps here:
  - **Hierarchical, locally connected** models
  - **Non-linear** feature extraction
- **Structure** of learning machine does matter
- Proposed architectures map well to **GPUs**
- **Iterative interpretation** uses partial results as context to resolve ambiguities
- Many open questions, e.g.
  - Object-centered representations
  - Full scene parsing / vision as inverse graphics