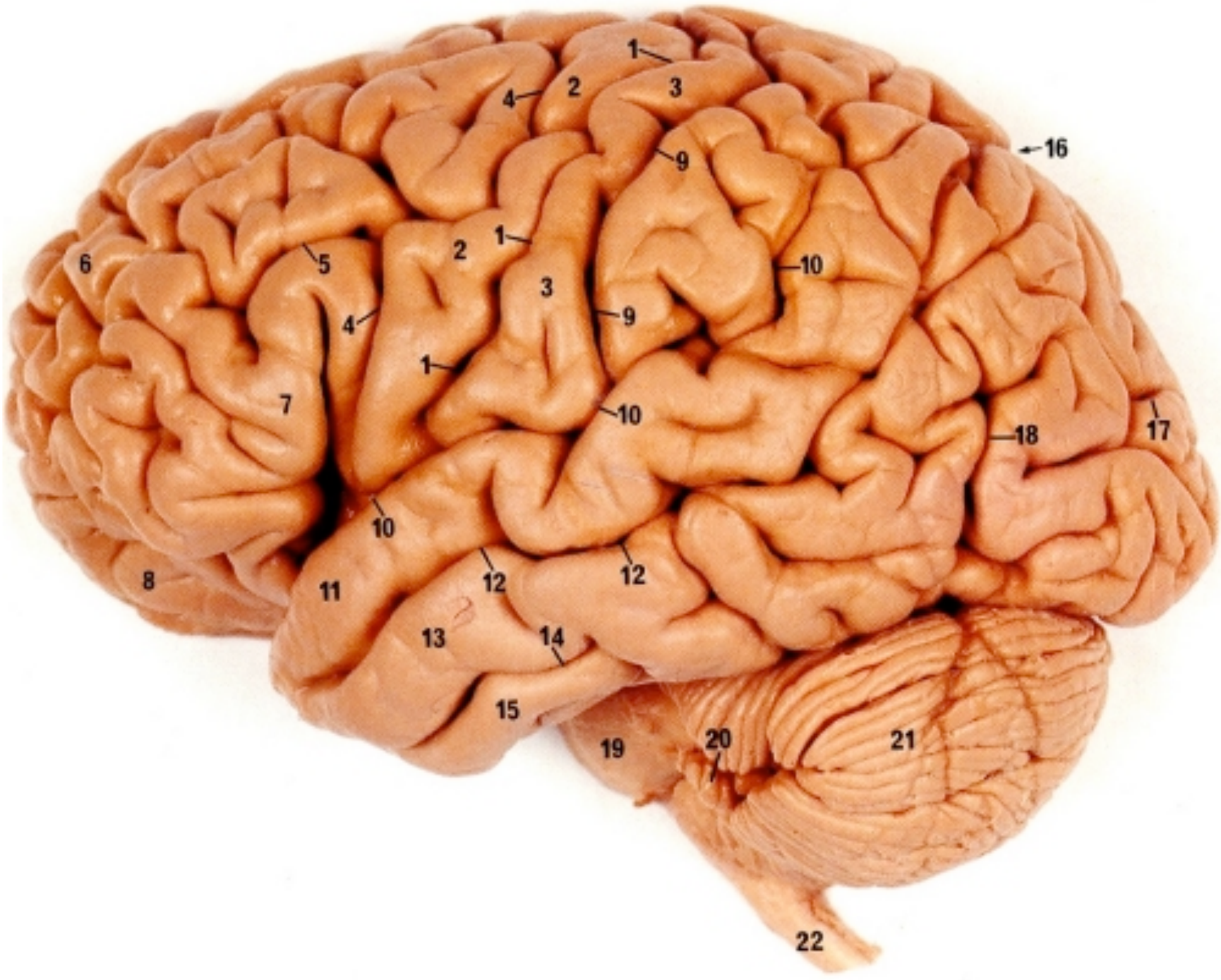
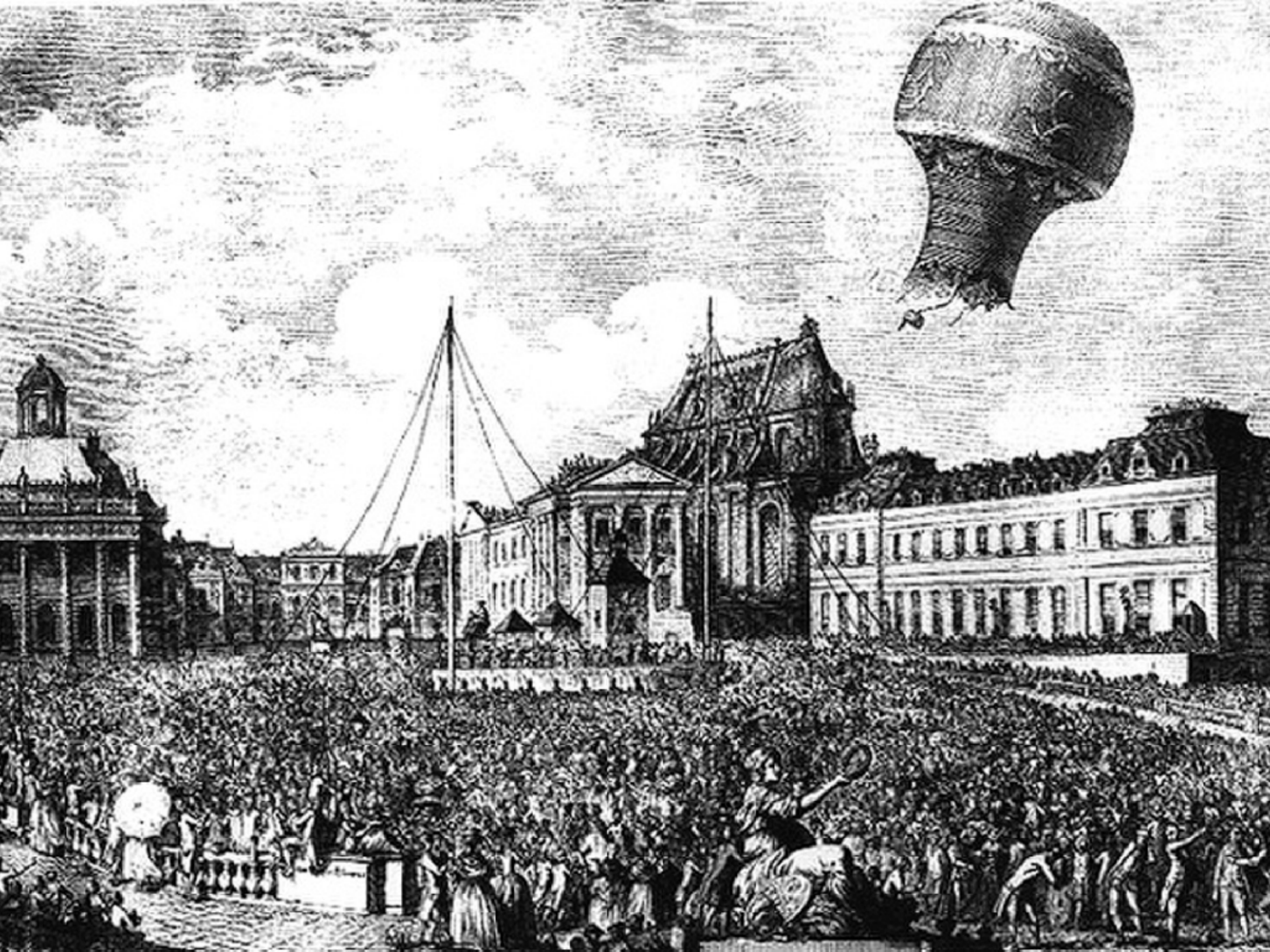
Is AI is still at the balloon stage?





Limitations of this approach

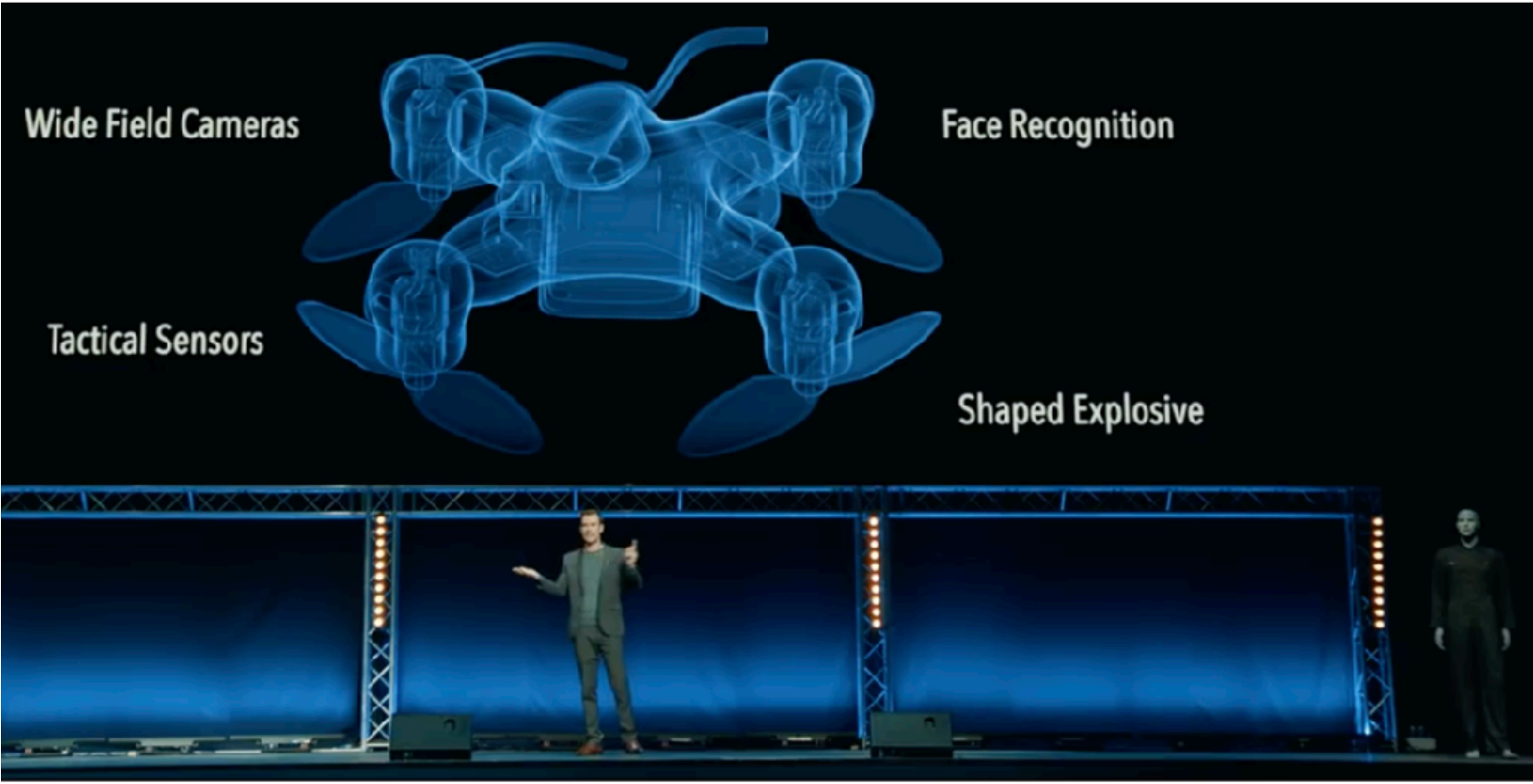
AI research focuses on underlying principles. Russel & Norvig:

Aeronautical engineering texts do not define the goal of their field as making "machines that fly so exactly like pigeons that they can fool even other pigeons."

AI: Threat or promise?

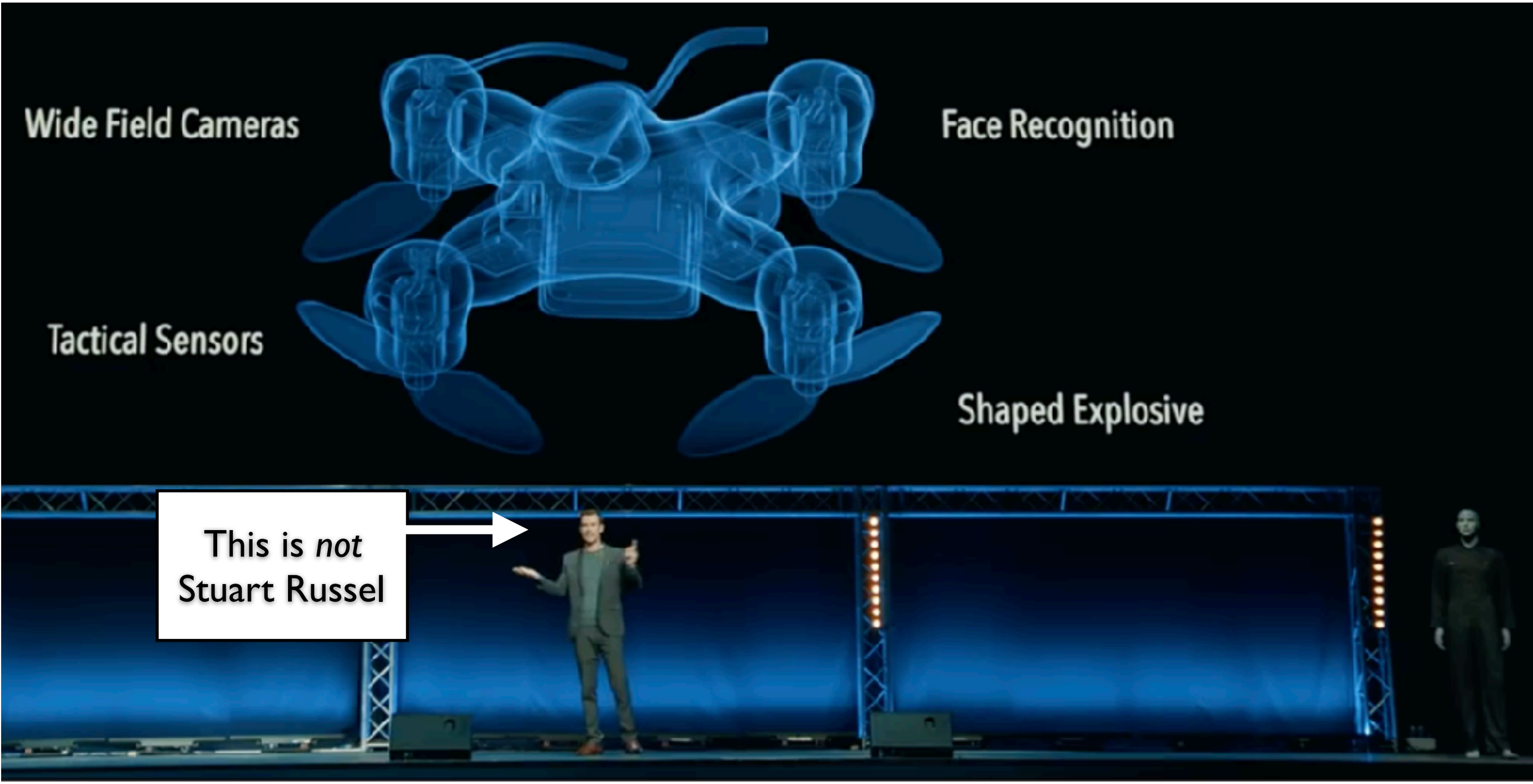
- AI is the "*biggest risk we face as a civilization*" — Elon Musk
- "*The development of full artificial intelligence could spell the end of the human race*" — Stephan Hawking
- "*I visualize a time when we will be to robots what dogs are to humans, and I'm rooting for the machines.*" — Claude Shannon
- "*Artificial intelligence is growing up fast, as are robots whose facial expressions can elicit empathy and make your mirror neurons quiver.*" — Diane Ackerman
- *"Some people worry that artificial intelligence will make us feel inferior, but then, anybody in his right mind should have an inferiority complex every time he looks at a flower.* — Alan Kay
- "*Before we work on artificial intelligence why don't we do something about natural stupidity?*" —Steve Polyak

Slaughterbots



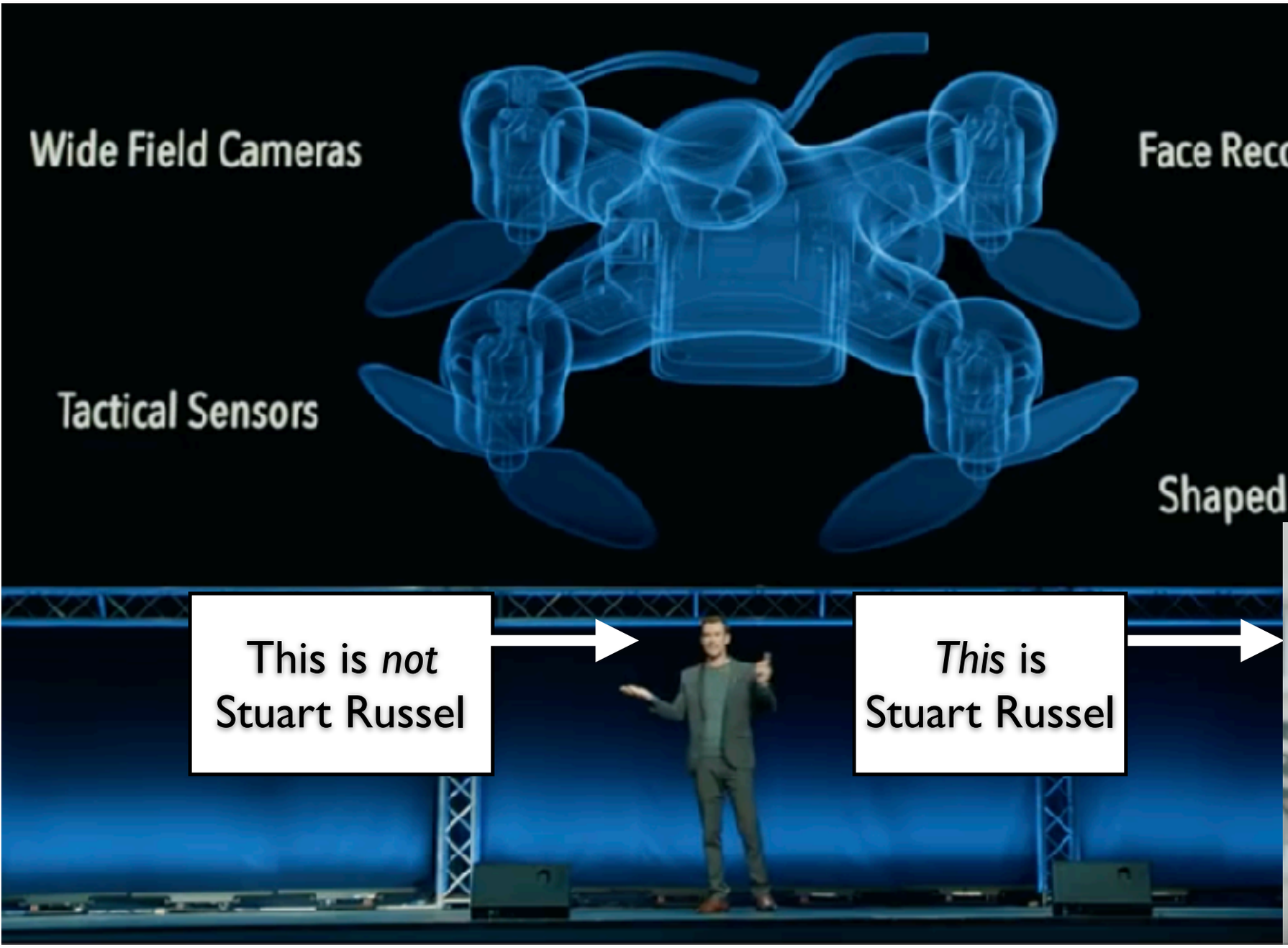
youtube “slaughterbots”
https://www.youtube.com/watch?v=9CO6M2HsoIA

Slaughterbots



youtube “slaughterbots”
https://www.youtube.com/watch?v=9CO6M2HsoIA

Slaughterbots



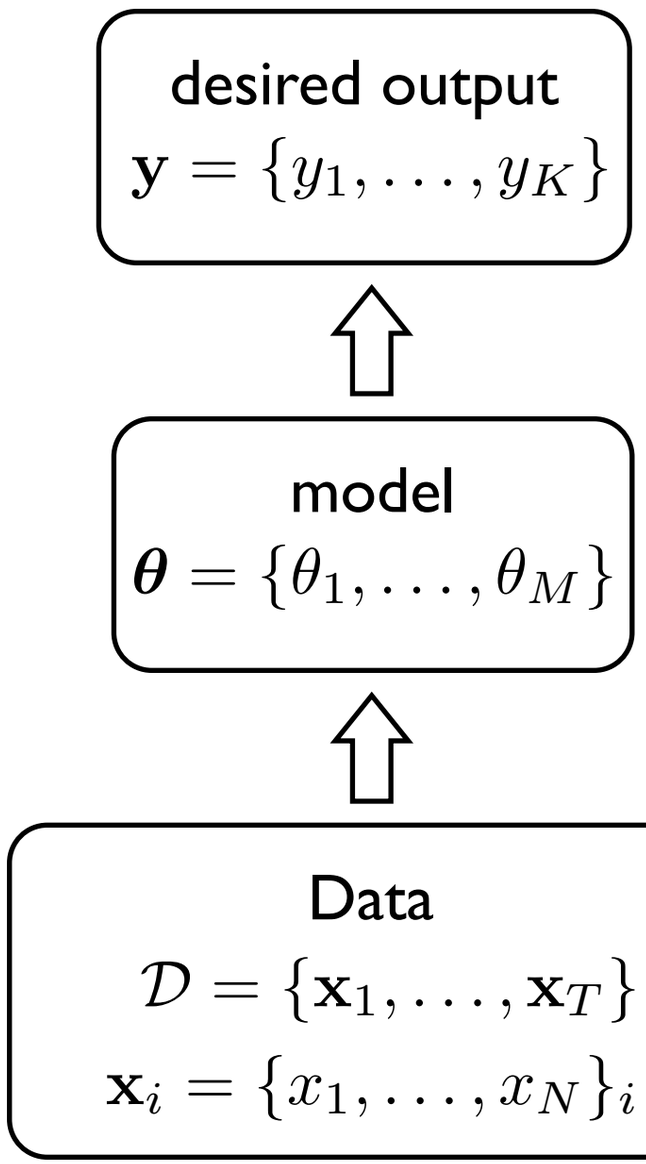
See Ban lethal autonomous weapons site:
http://autonomousweapons.org

EECS 391
Intro to AI

Applications of Neural Networks

L19 Tue Nov 28

The general classification/regression problem



for classification:

$$y_i = \begin{cases} 1 & \text{if } \mathbf{x}_i \in C_i \equiv \text{class } i, \\ 0 & \text{otherwise} \end{cases}$$

regression for arbitrary y.

model (e.g. a decision tree) is defined by M parameters, **e.g. a multi-layer neural network.**

input is a set of T observations, each an N-dimensional vector (binary, discrete, or continuous)

Given data, we want to learn a model that can correctly classify novel observations **or map the inputs to the outputs**

A general multi-layer neural network

- Error function is defined as before, where we use the target vector t_n to define the desired output for network output y_n.

$$E = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{y}_n(\mathbf{x_n}, \mathbf{W_{1:L}}) - \mathbf{t}_n)^2$$

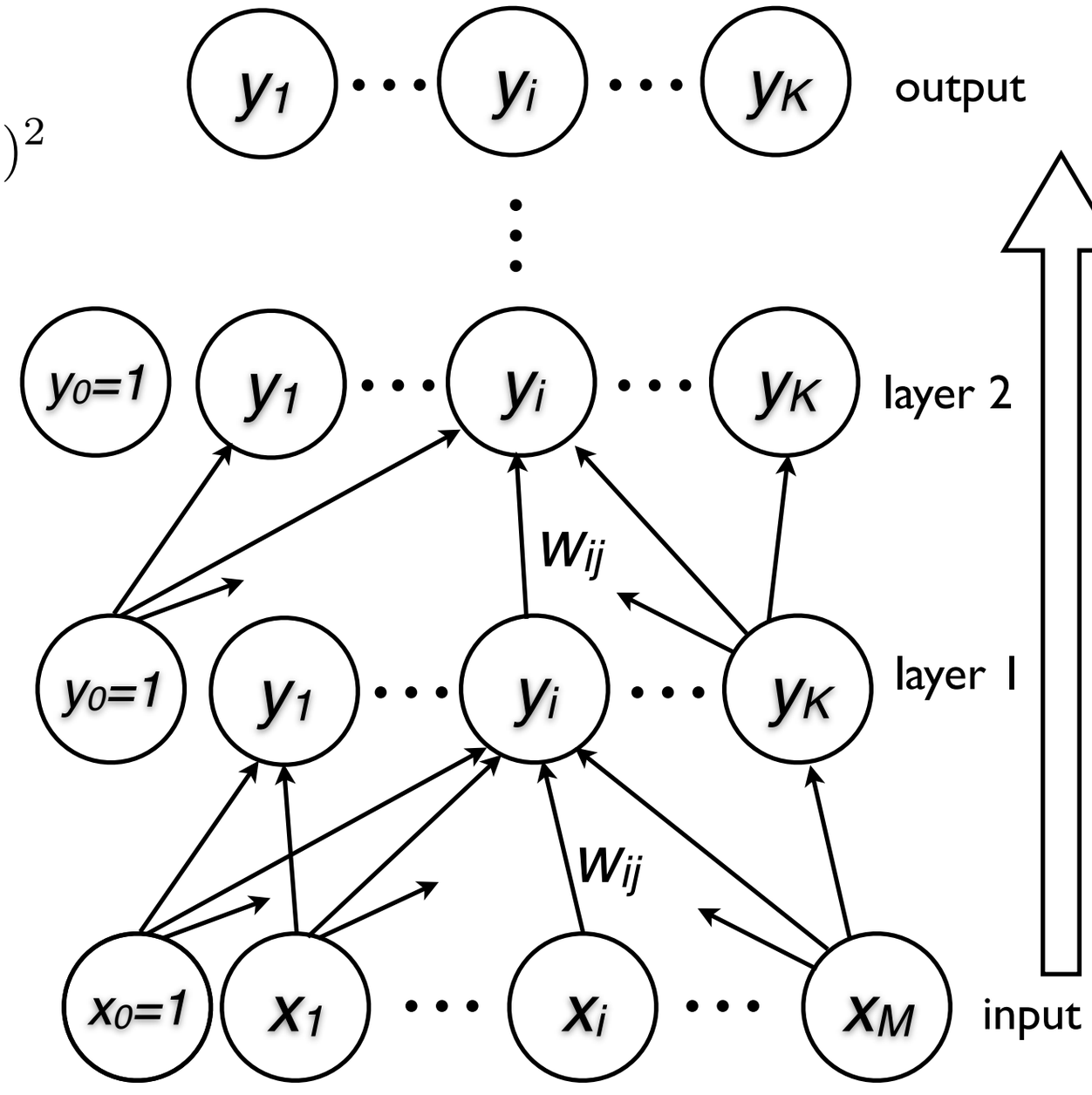
- The "forward pass" computes the outputs at each layer:

$$y_j^l = f(\sum_i w_{i,j}^l \, y_j^{l-1})$$

$$l = \{1, \ldots, L\}$$

$$\mathbf{x} \equiv \mathbf{y}^0$$

$$\text{output} = \mathbf{y}^L$$

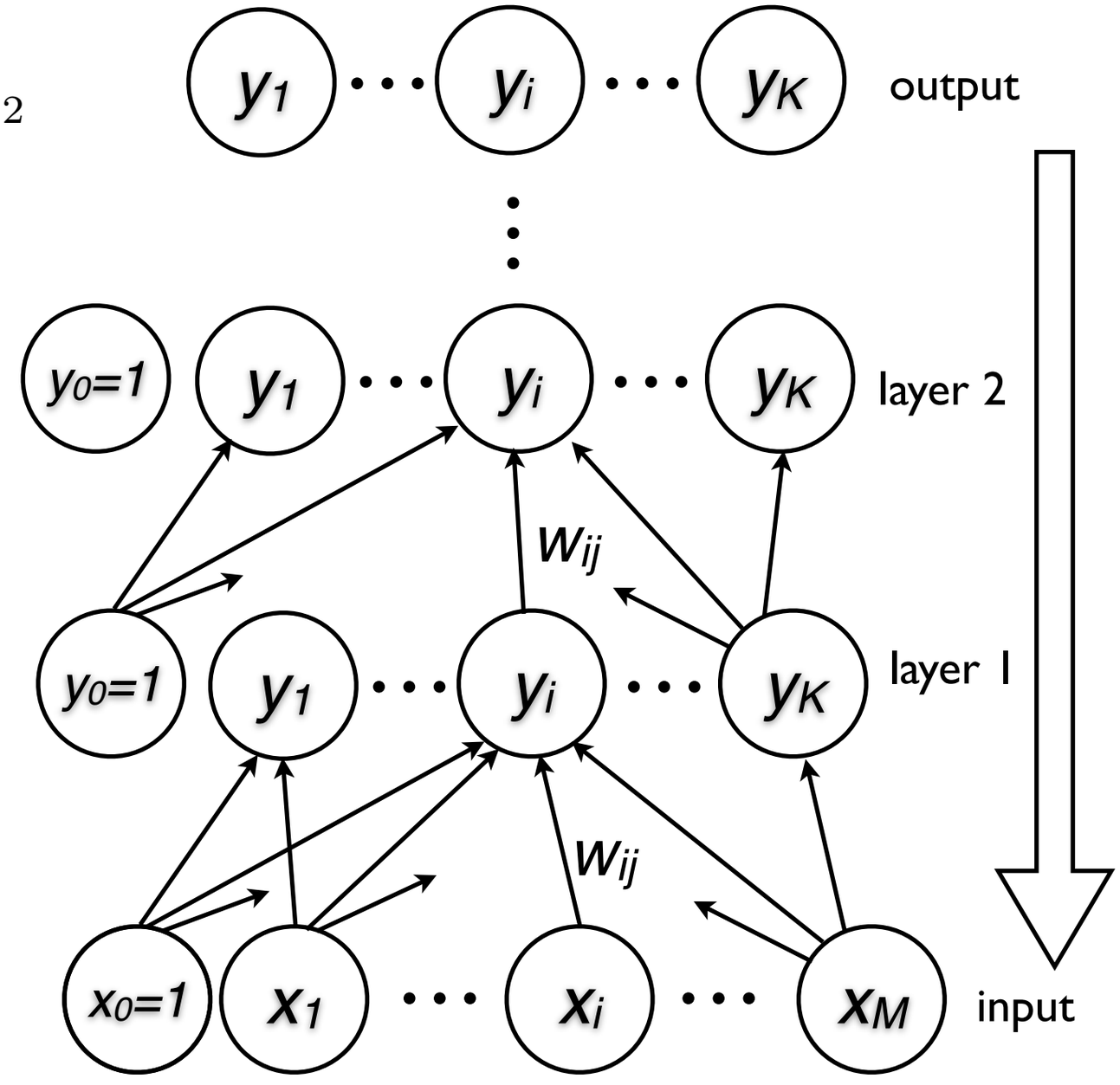


Deriving the gradient for a sigmoid neural network

- Mathematical procedure for train is gradient descient: same as before, except the gradients are more complex to derive.

New problem: local minima

$$E = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{y}_n(\mathbf{x_n}, \mathbf{W_{1:L}}) - \mathbf{t}_n)^2$$



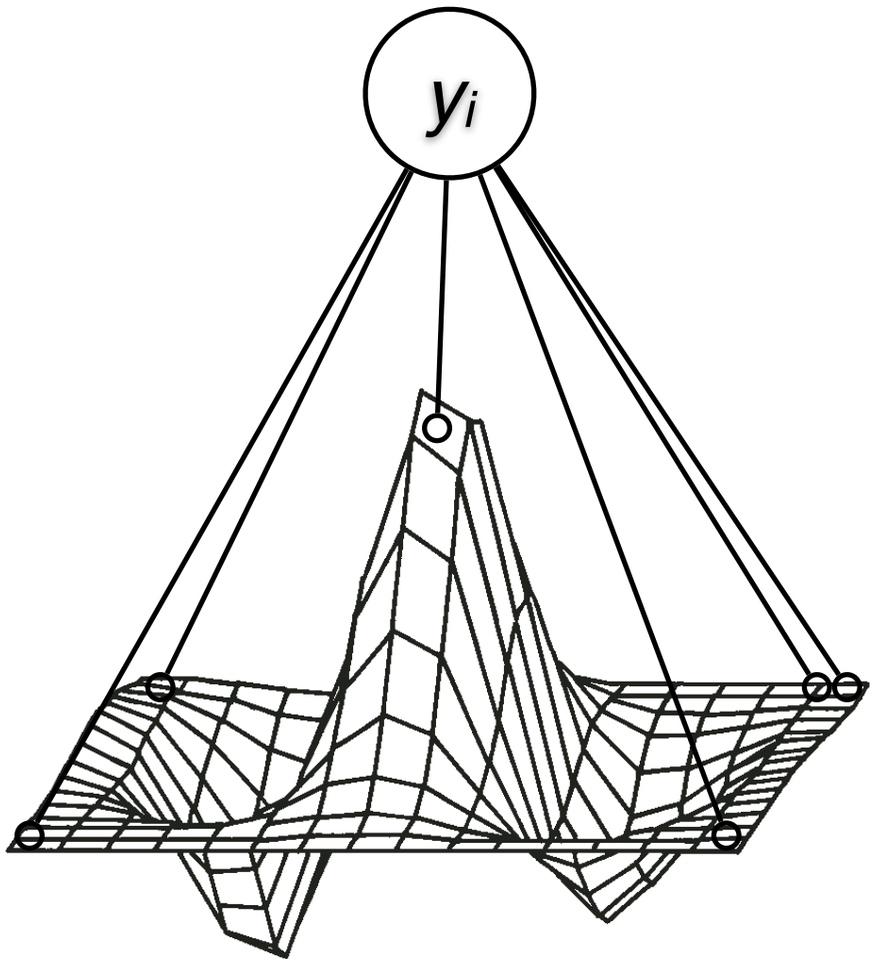
- Convenient fact for the sigmoid non-linearity:

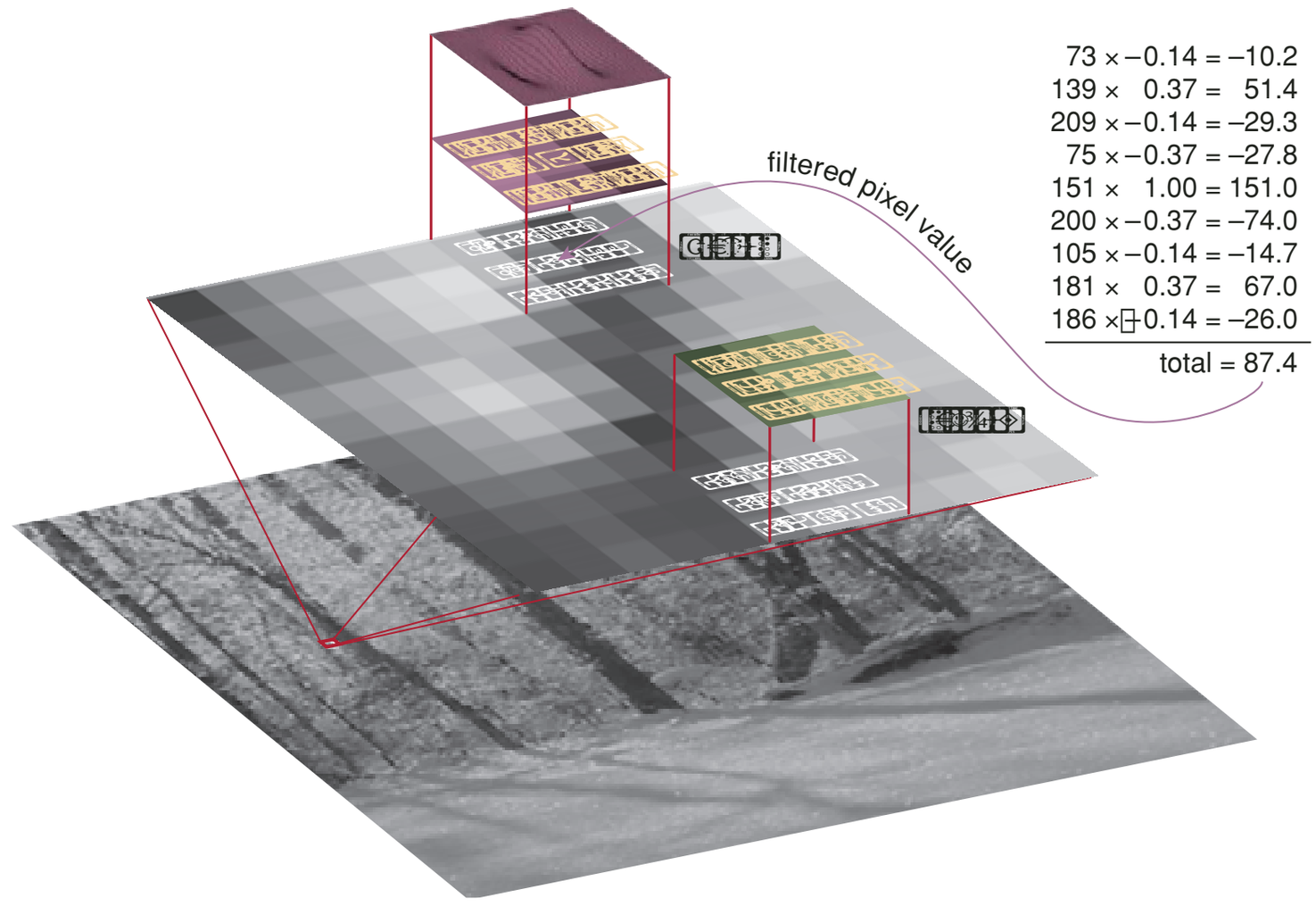
$$\frac{d\sigma(x)}{dx} = \frac{d}{dx}\frac{1}{1+\exp(-x)}$$
$$= \sigma(x)(1-\sigma(x))$$

- backward pass computes the gradients: *back-propagation*

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \epsilon\frac{\partial E}{\mathbf{W}}$$

Linear NNs perform the same operation as linear filters





from Olshausen and Field (2000)

$$y_j = \sum_{i=0}^{M} w_{i,j}\, x_i$$
$$= \mathbf{w}_j^T \mathbf{x}$$

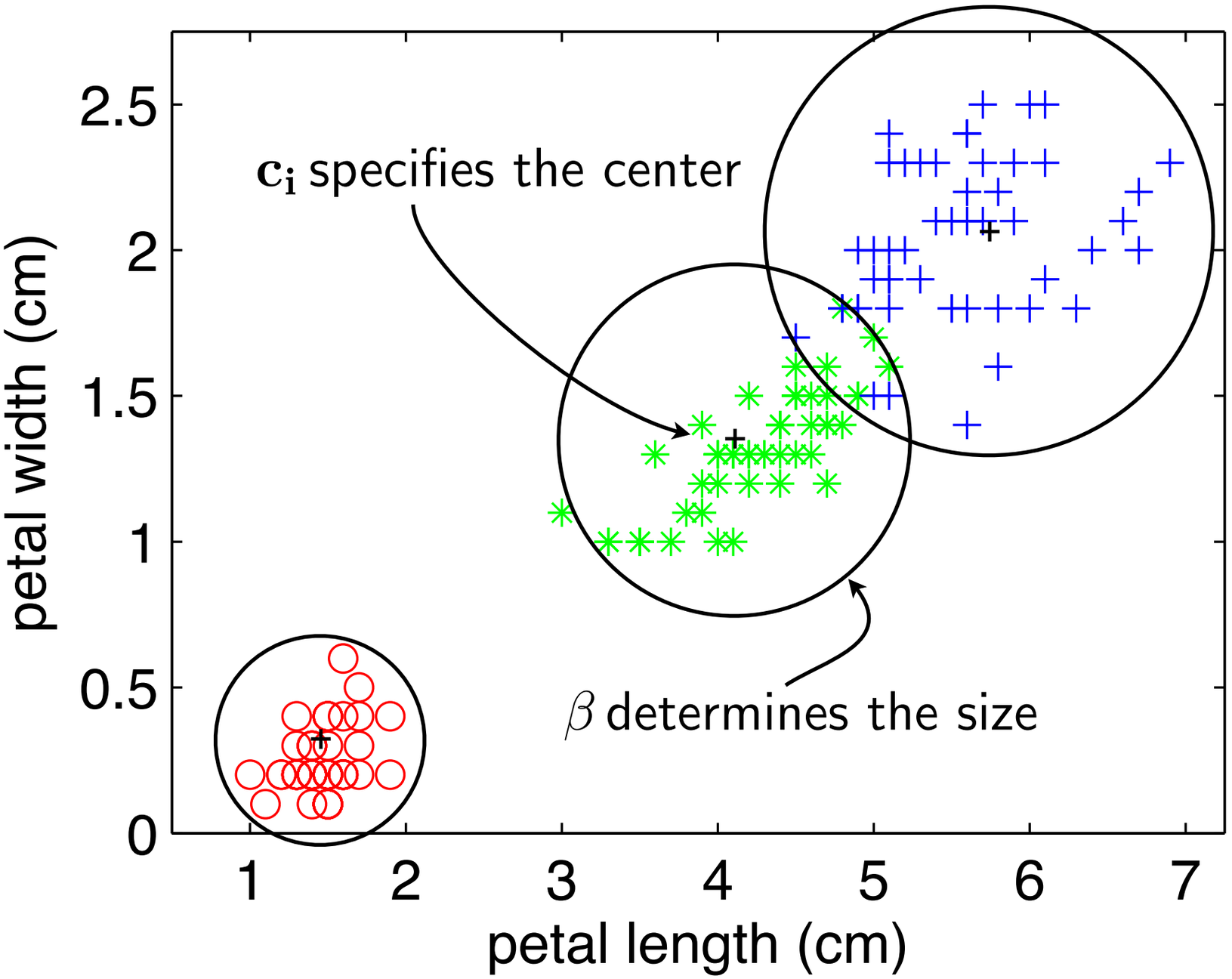
What would the weights look like for a NN trained to detect faces?

Other types of non-linearities

- a radial basis NN

$$y_i = \exp[-\beta||\mathbf{x} - \mathbf{c_i}||^2]$$

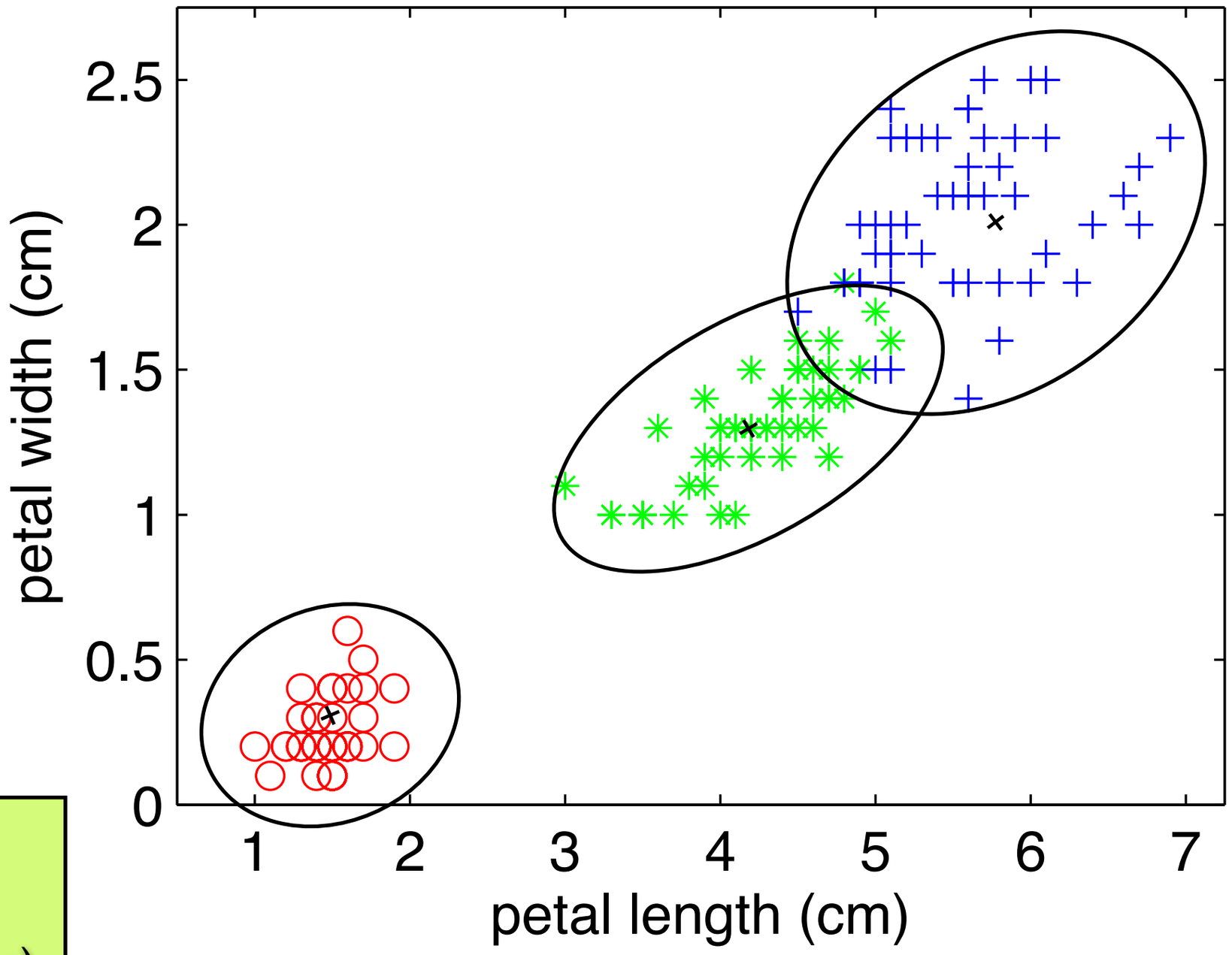
- this is clustering
- NN "weights" learn the cluster centers
- this NN needs only a single layer



Other types of non-linearities

- can generalize to non-spherical Gaussians:

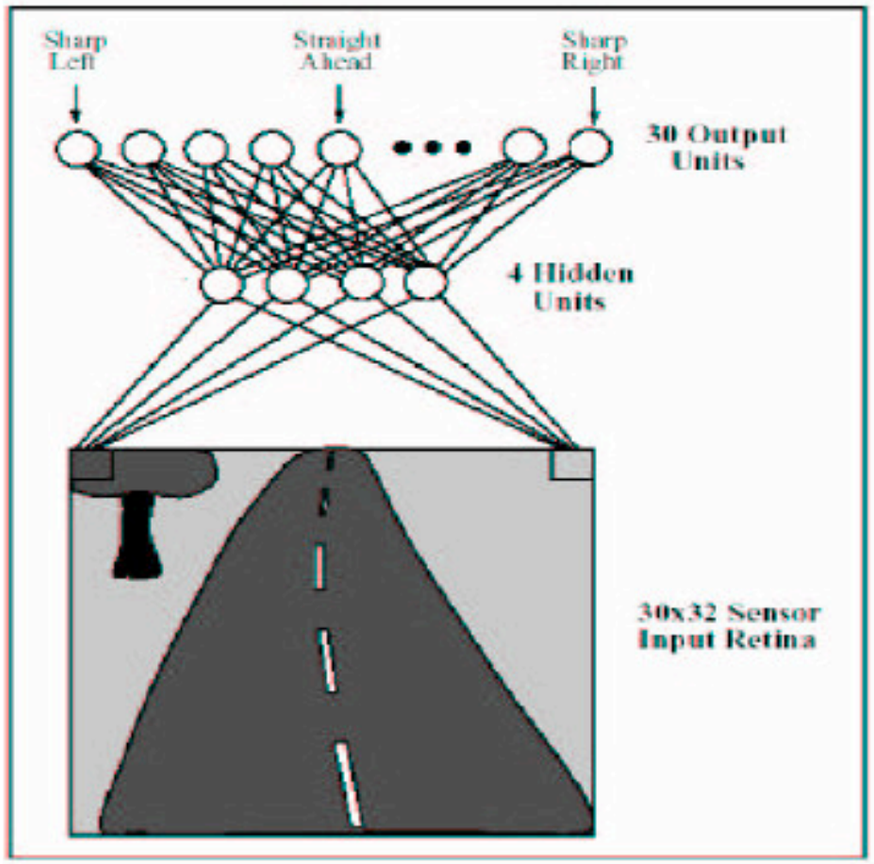
$$y_i = \frac{1}{Z(\Sigma_i)} \exp\left[-\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_i)\right]$$

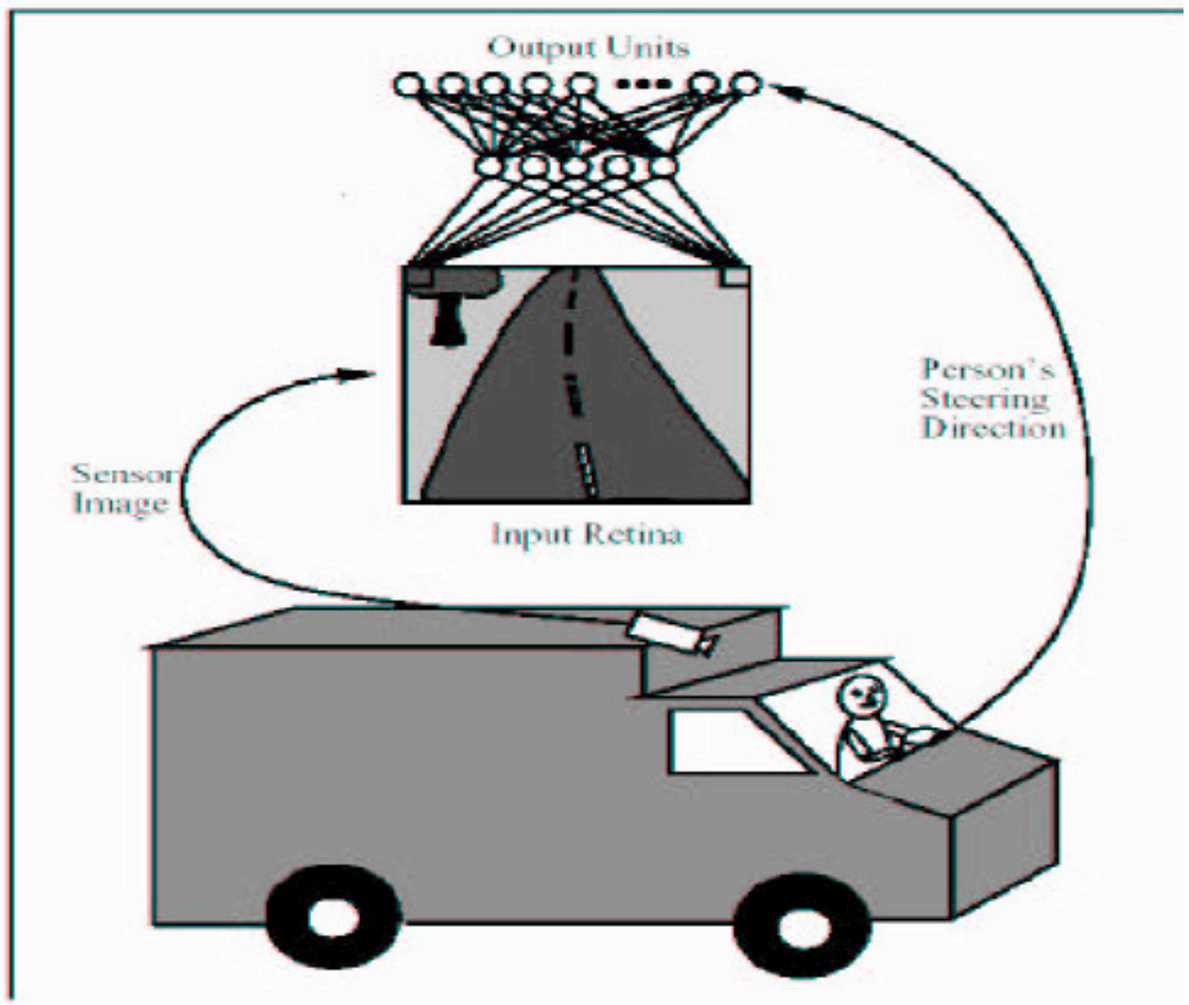


Many other possible ML algorithms, e.g. SVMs (support vector machines)

Applications: Driving (output is analog: steering direction)

network with 1 layer
(4 hidden units)





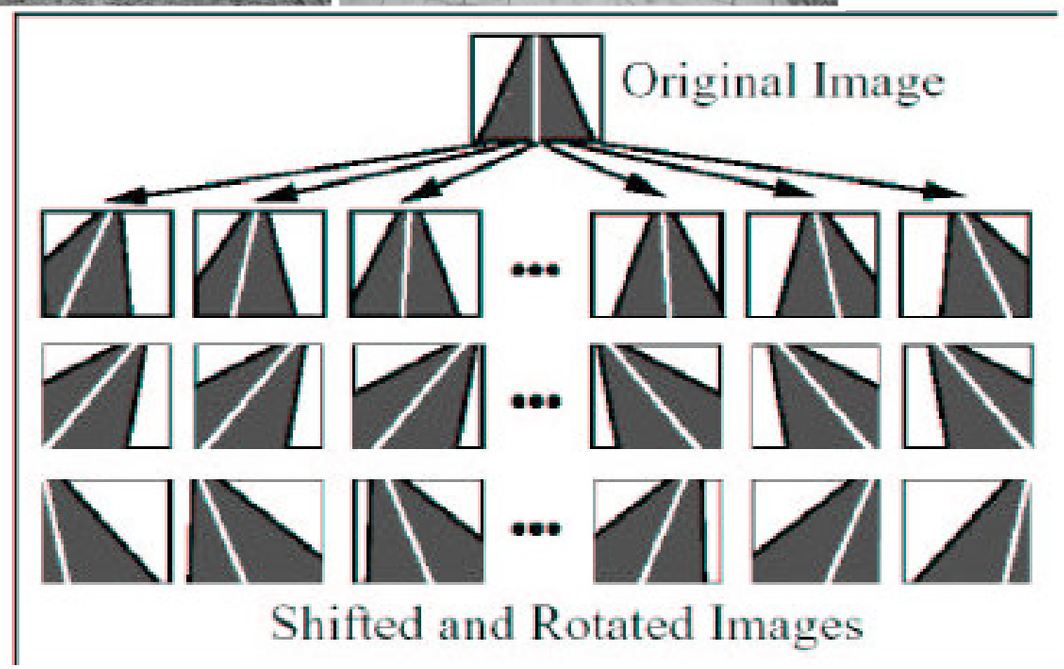
- Learns to drive on roads
- Demonstrated at highway speeds over 100s of miles

D. Pomerleau. *Neural network perception for mobile robot guidance.* Kluwer Academic Publishing, 1993.

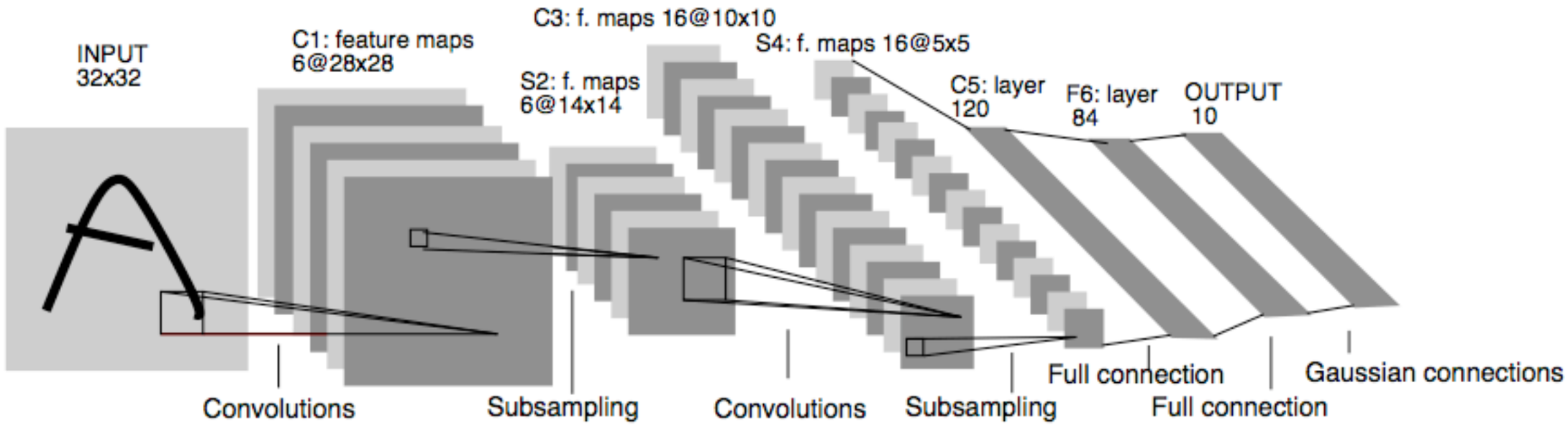
Real image input is augmented to avoid overfitting

Training data:
Images +
corresponding
steering angle

Important:
Conditioning of
training data to
generate new
examples → avoids
overfitting



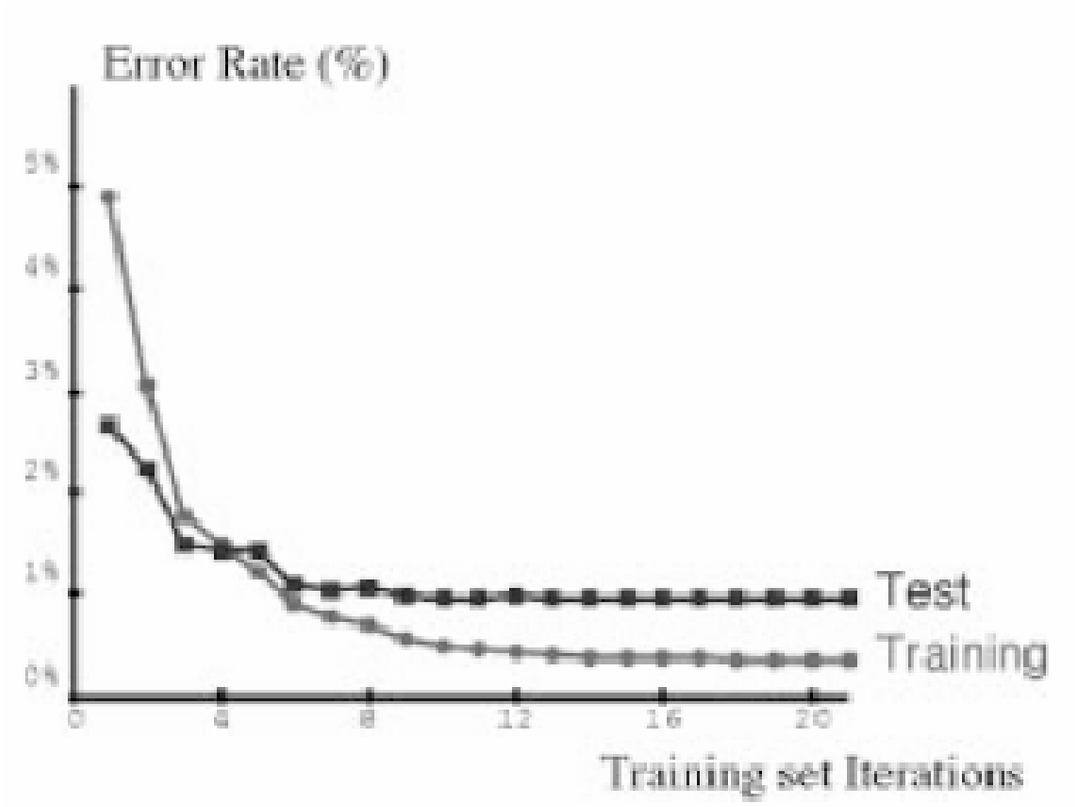
Hand-written digits: LeNet



- Takes as input image of handwritten digit
- Each pixel is an input unit
- Complex network with many layers
- Output is digit class

- Tested on large (50,000+) database of handwritten samples
- Real-time
- Used commercially

LeNet

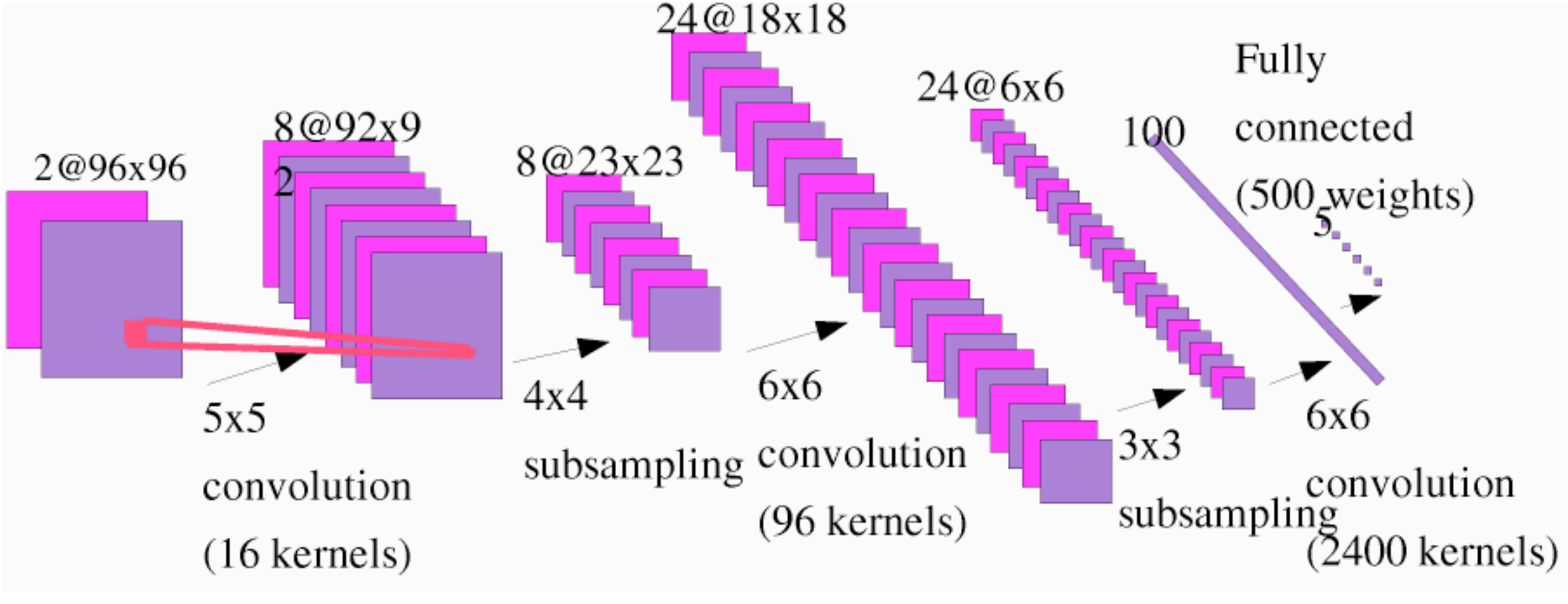


Very low error rate (<< 1%

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, november 1998.

http://yann.lecun.com/exdb/lenet/

Object recognition



- LeCun, Huang, Bottou (2004). Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting. Proceedings of CVPR 2004.
- http://www.cs.nyu.edu/~yann/research/norb/

Form of most object recognition models

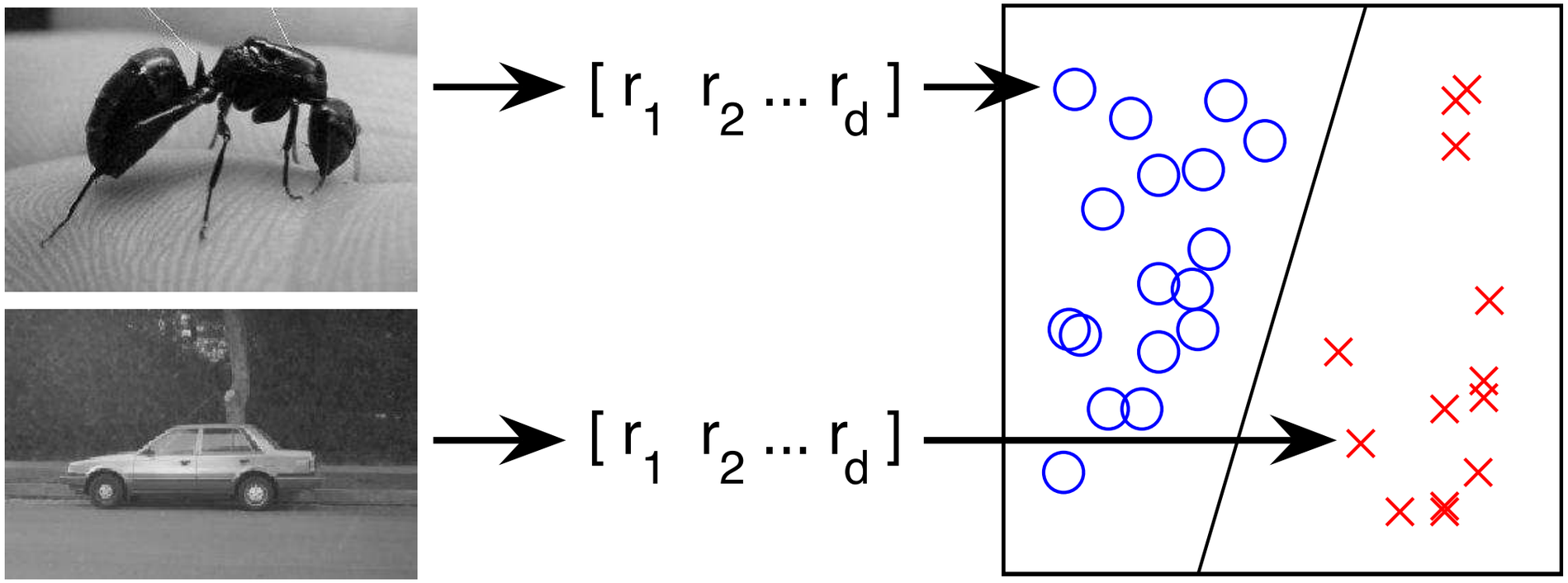


Figure 1. Overall form of our model. Images are reduced to feature vectors which are then classified by an SVM.

from (Mutch and Lowe, 2008)

arXiv.org > cs > arXiv:1710.08864

Search or Article ID inside arXiv | All papers | Broaden your search using Semantic Scholar

(Help | Advanced search)

Computer Science > Learning

One pixel attack for fooling deep neural networks

Jiawei Su, Danilo Vasconcellos Vargas, Sakurai Kouichi

(Submitted on 24 Oct 2017 (v1), last revised 16 Nov 2017 (this version, v2))

Recent research has revealed that the output of Deep Neural Networks (DNN) can be easily altered by adding relatively small perturbations to the input vector. In this paper, we analyze an attack in an extremely limited scenario where only one pixel can be modified. For that we propose a novel method for generating one-pixel adversarial perturbations based on differential evolution. It requires less adversarial information and can fool more types of networks. The results show that 70.97% of the natural images can be perturbed to at least one target class by modifying just one pixel with 97.47% confidence on average. Thus, the proposed attack explores a different take on adversarial machine learning in an extreme limited scenario, showing that current DNNs are also vulnerable to such low dimension attacks.

Subjects: **Learning (cs.LG)**; Computer Vision and Pattern Recognition (cs.CV); Machine Learning (stat.ML)
Cite as: arXiv:1710.08864 **[cs.LG]**
(or arXiv:1710.08864v2 **[cs.LG]** for this version)

Submission history

From: Jiawei Su [view email]
[v1] Tue, 24 Oct 2017 16:02:19 GMT (815kb,D)
[v2] Thu, 16 Nov 2017 07:58:35 GMT (958kb,D)

Download:

- PDF
- Other formats

(license)

Current browse context:
cs.LG
< prev | next >
new | recent | 1710

Change to browse by:
cs
cs.CV
stat
stat.ML

References & Citations
- NASA ADS

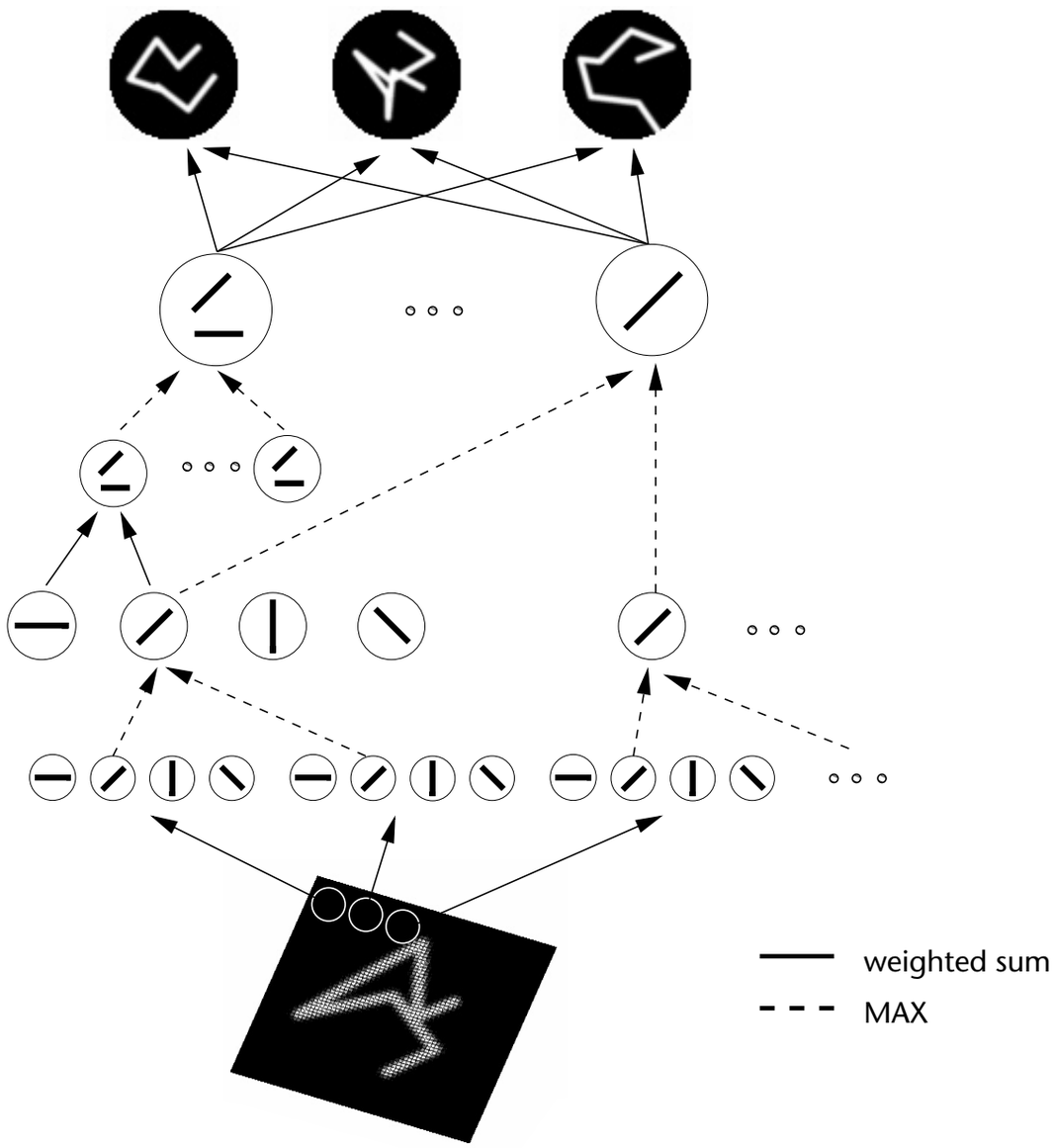
1 blog link (what is this?)

Bookmark (what is this?)

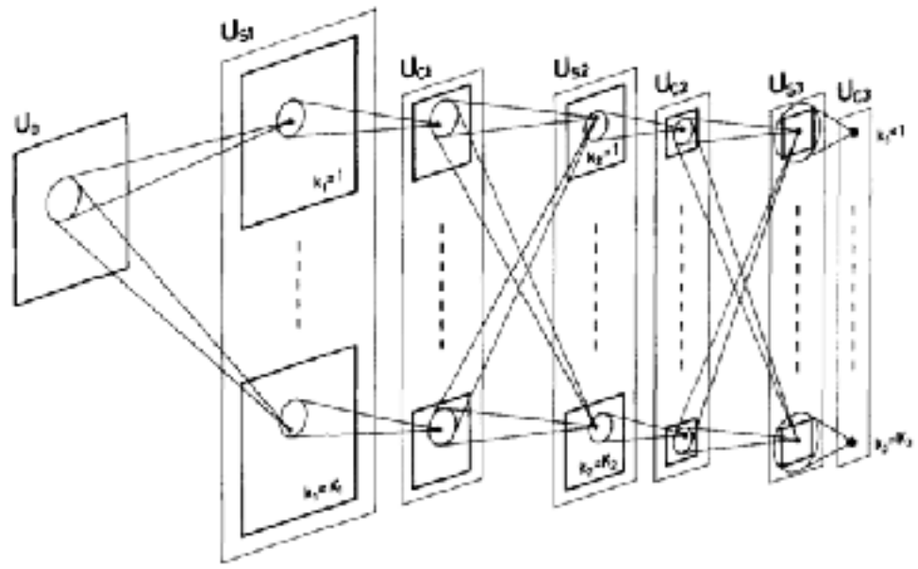
Figure 1. One-pixel attacks created with the proposed algorithm that successfully fooled a target DNN. The original class labels are written below each image with the target class label written inside brackets.

Feedforward models

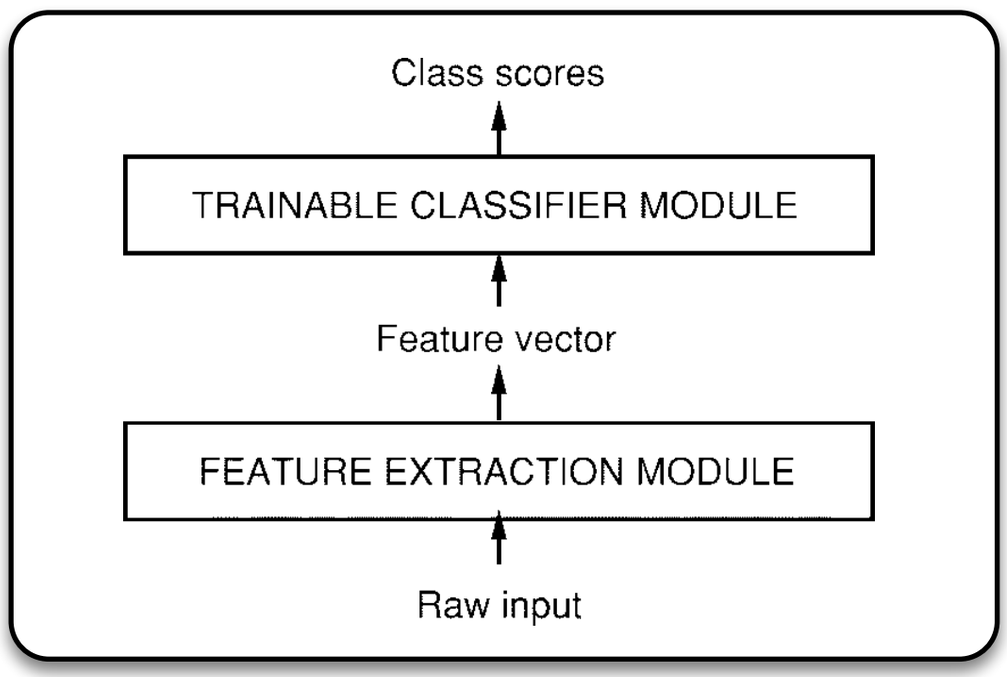
HMAX



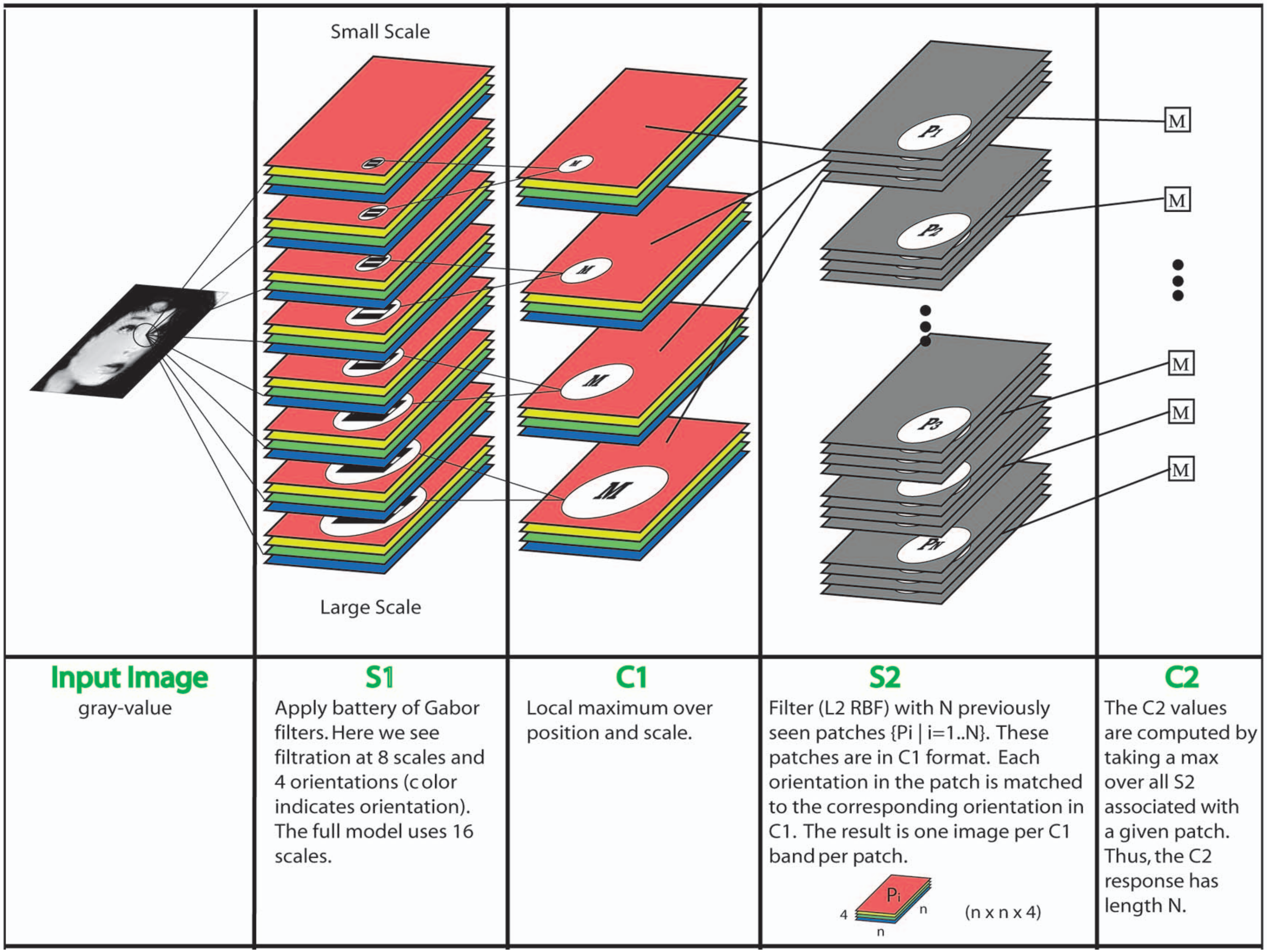
Neocognitron

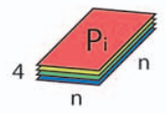


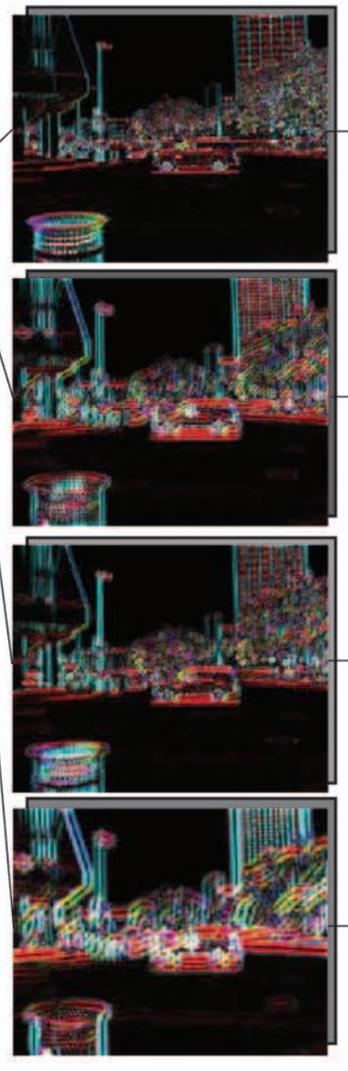
Generic Feed-forward System

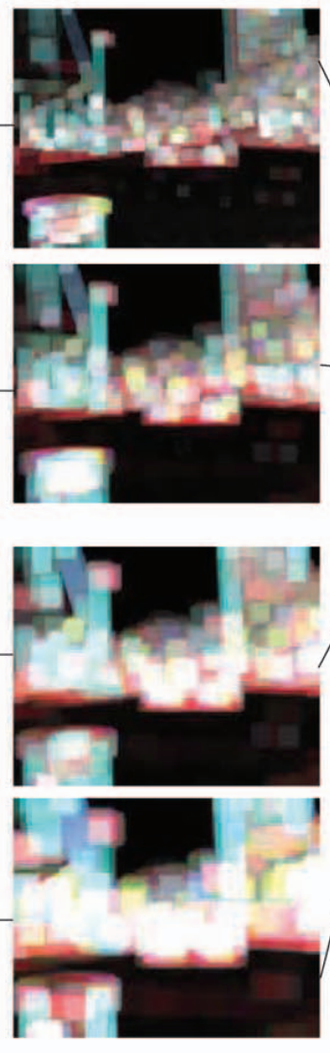


Extensions of HMAX (Serre et al, 2007)



Input Image	S1	C1	S2	C2
gray-value	Apply battery of Gabor filters. Here we see filtration at 8 scales and 4 orientations (color indicates orientation). The full model uses 16 scales.	Local maximum over position and scale.	Filter (L2 RBF) with N previously seen patches {Pi | i=1..N}. These patches are in C1 format. Each orientation in the patch is matched to the corresponding orientation in C1. The result is one image per C1 band per patch.  (n x n x 4)	The C2 values are computed by taking a max over all S2 associated with a given patch. Thus, the C2 response has length N.





.11

.09

.19

.78

Four S1 scale bands are shown here. The levels in these images have been scaled for illustrative purposes. Color indicates orientation, as above.

Only one S2 scale is shown for each patch.

Applying the model to object recognition in complex scenes

TABLE 1
Summary of the S_1 and C_1 *SMFs* Parameters

C_1 layer			S_1 layer		
Scale band $\mathcal{S}$	Spatial pooling grid ($N_S \times N_S$)	Overlap $\Delta_{\mathcal{S}}$	filter size s	Gabor σ	Gabor λ
Band 1	8×8	4	7×7 9×9	2.8 3.6	3.5 4.6
Band 2	10×10	5	11×11 13×13	4.5 5.4	5.6 6.8
Band 3	12×12	6	15×15 17×17	6.3 7.3	7.9 9.1
Band 4	14×14	7	19×19 21×21	8.2 9.2	10.3 11.5
Band 5	16×16	8	23×23 25×25	10.2 11.3	12.7 14.1
Band 6	18×18	9	27×27 29×29	12.3 13.4	15.4 16.8
Band 7	20×20	10	31×31 33×33	14.6 15.8	18.2 19.7
Band 8	22×22	11	35×35 37×37	17.0 18.2	21.2 22.8

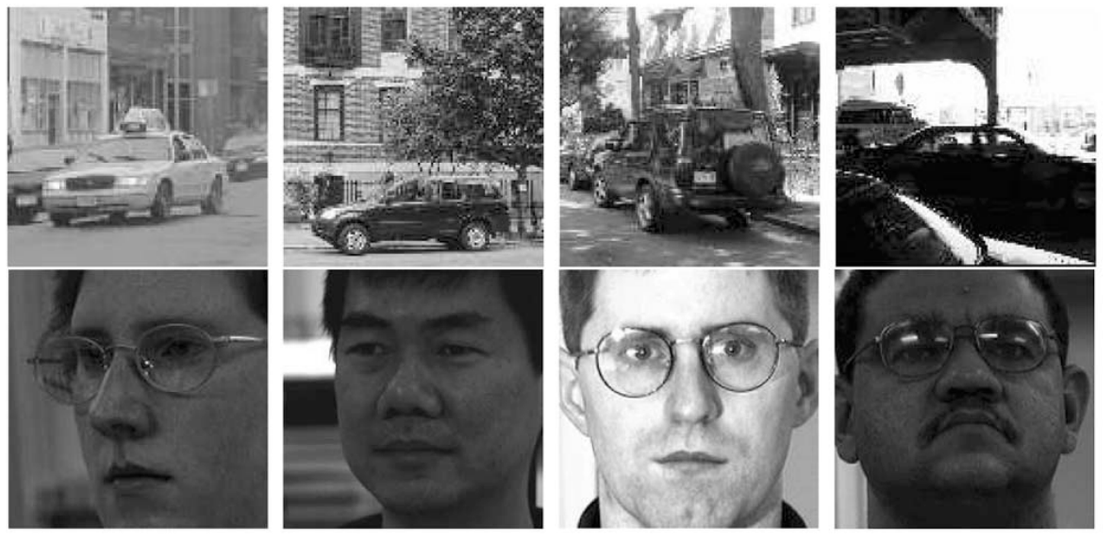


Fig. 2. Sample images from the MIT-CBCL multiview car [18] and face [17] data sets.

Caltech 101 Database: 101 categories, 9144 images

Google Background

Comparison between C2 and SIFT features

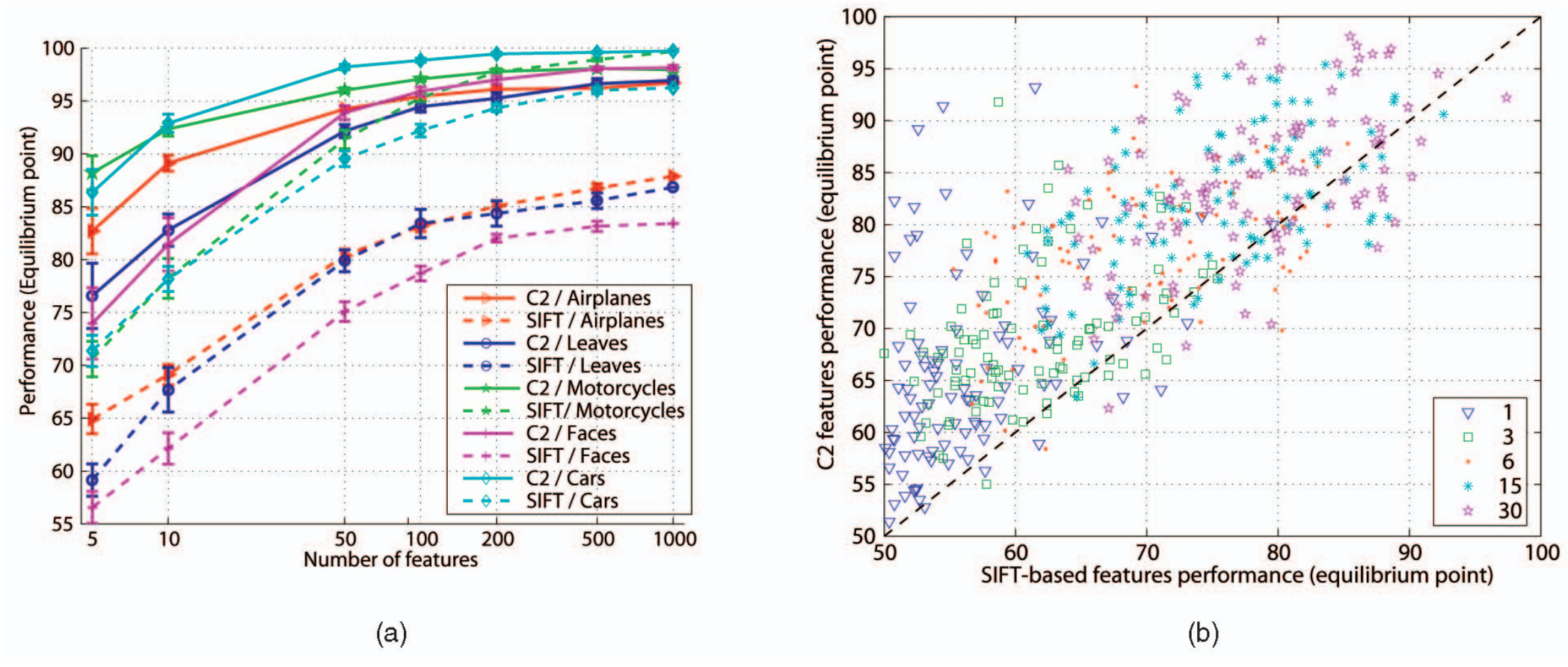


Fig. 3. Comparison between the SIFT and the C_2 features on the *CalTech5* for (a) different numbers of features and on the (b) *CalTech101* for a different number of training examples.

Result on different object categories

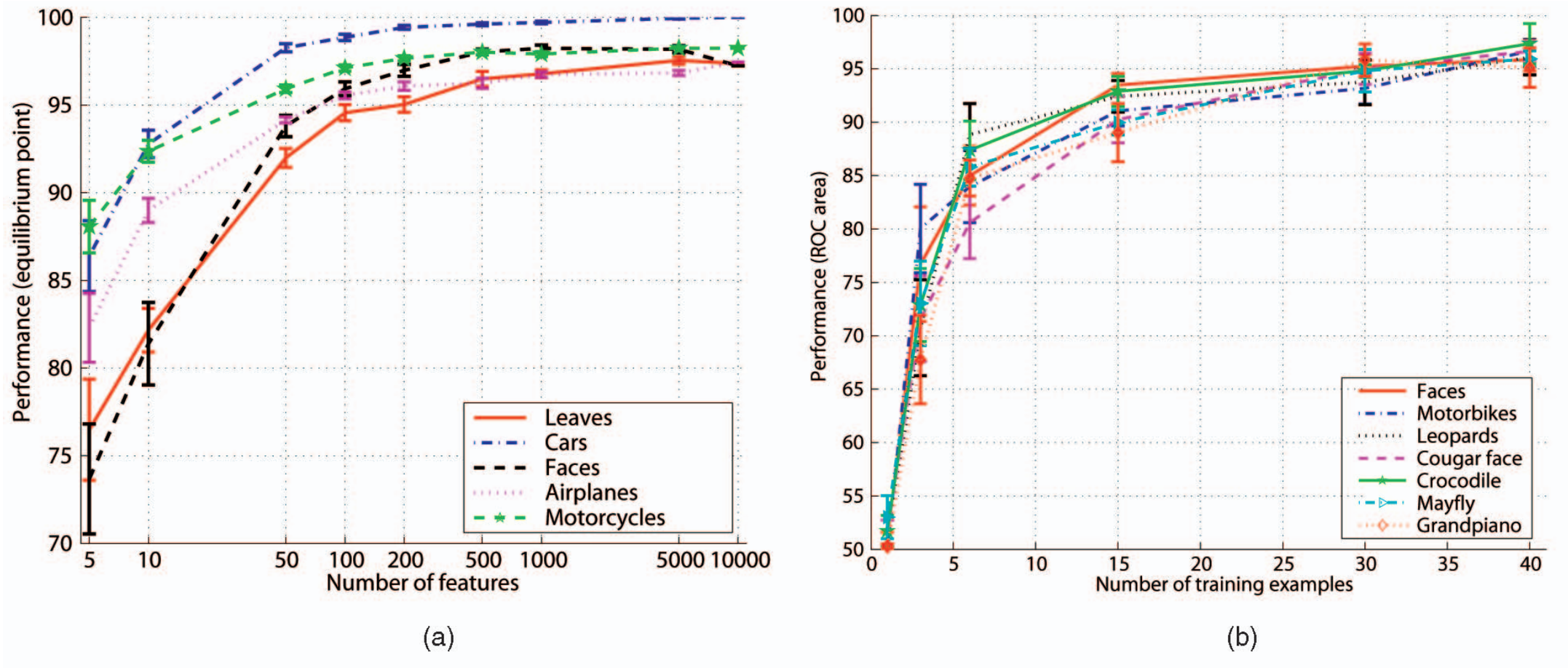
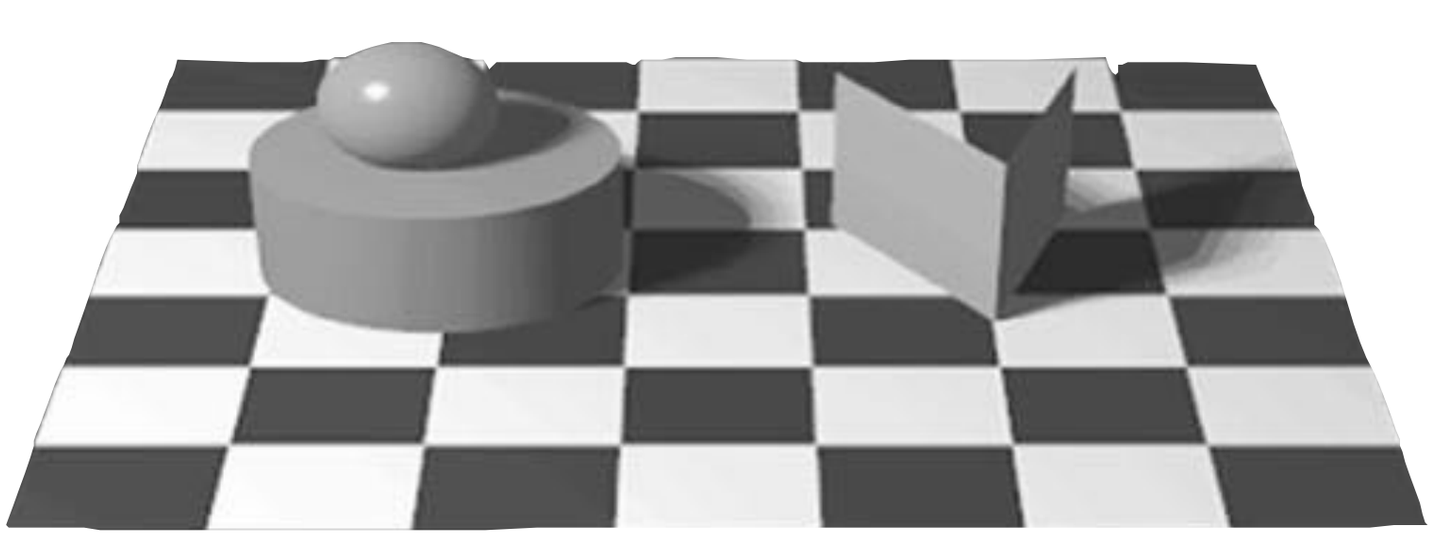


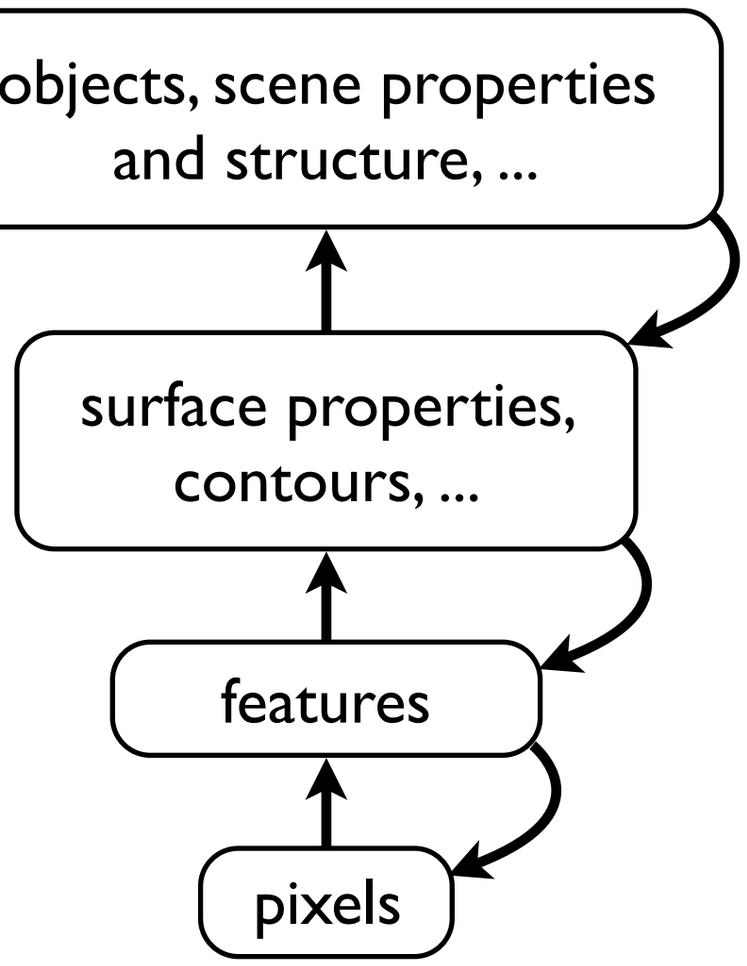
Fig. 4. Performance obtained with gentleBoost and different numbers of C_2 features on the (a) *CalTech5* and on sample categories from the (b) *CalTech101* for a different number of training examples.

Comments?

Do feed-forward models incorporate context?

- many real-world patterns are hierarchical in structure
- interpretation of patterns depends on context
- essential for complex recognition tasks

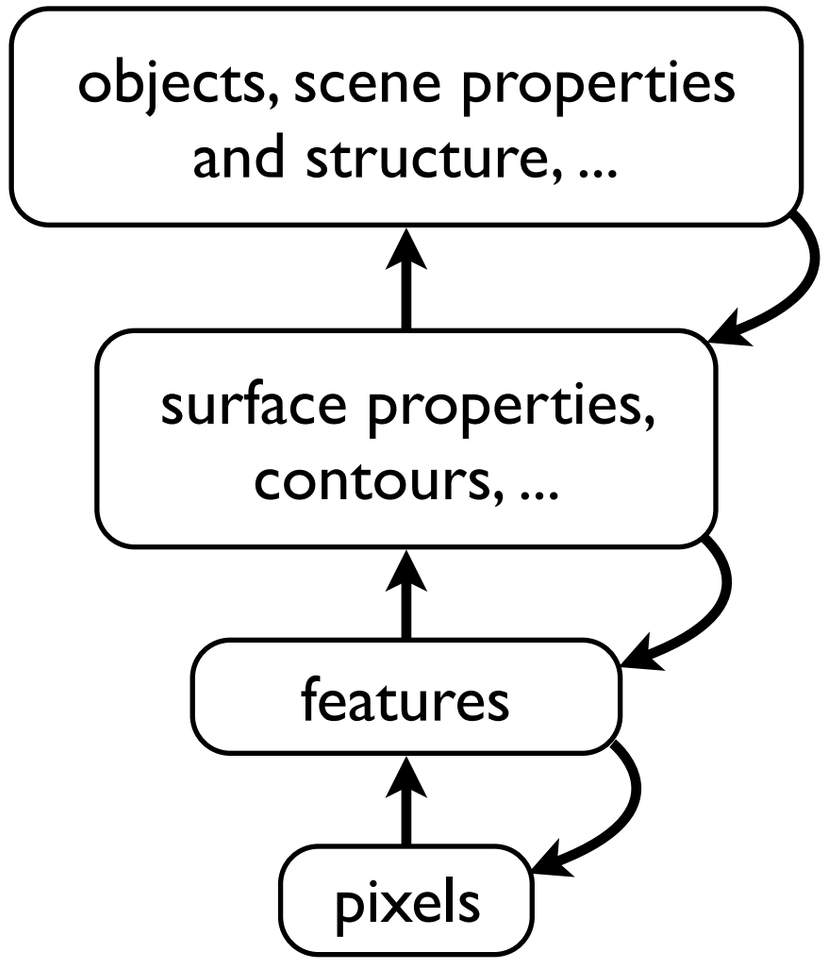




Do feed-forward models incorporate context?

- many real-world patterns are hierarchical in structure
- interpretation of patterns depends on context
- essential for complex recognition tasks

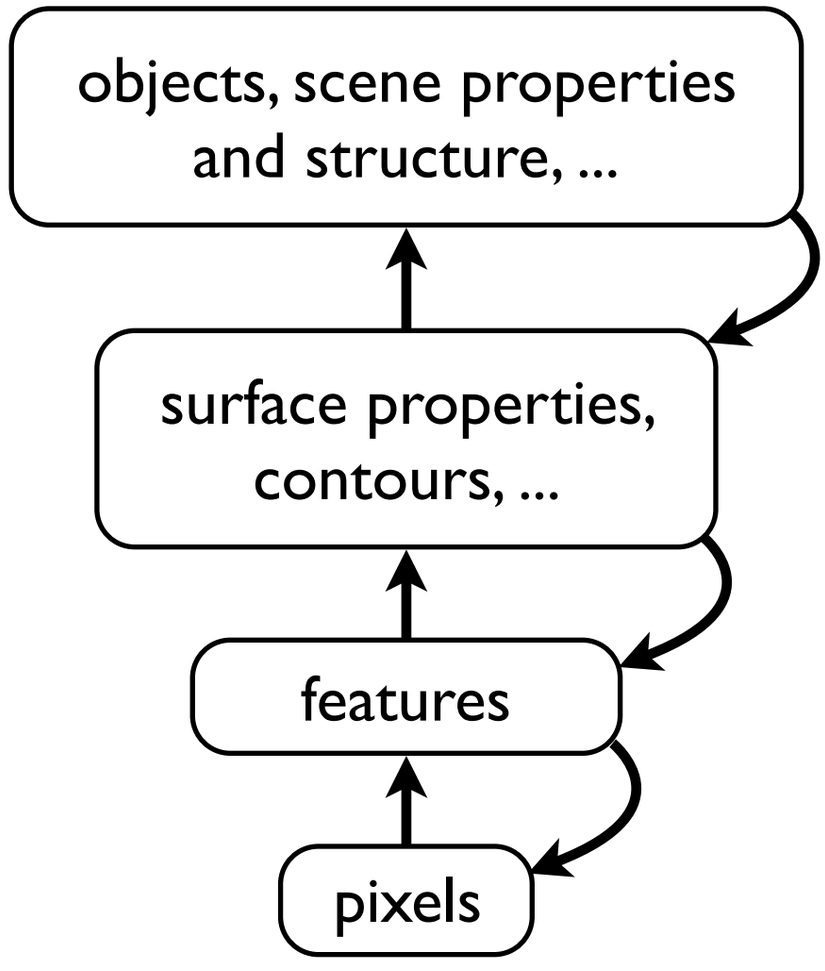
from Torralba, 2003



Do feed-forward models incorporate context?

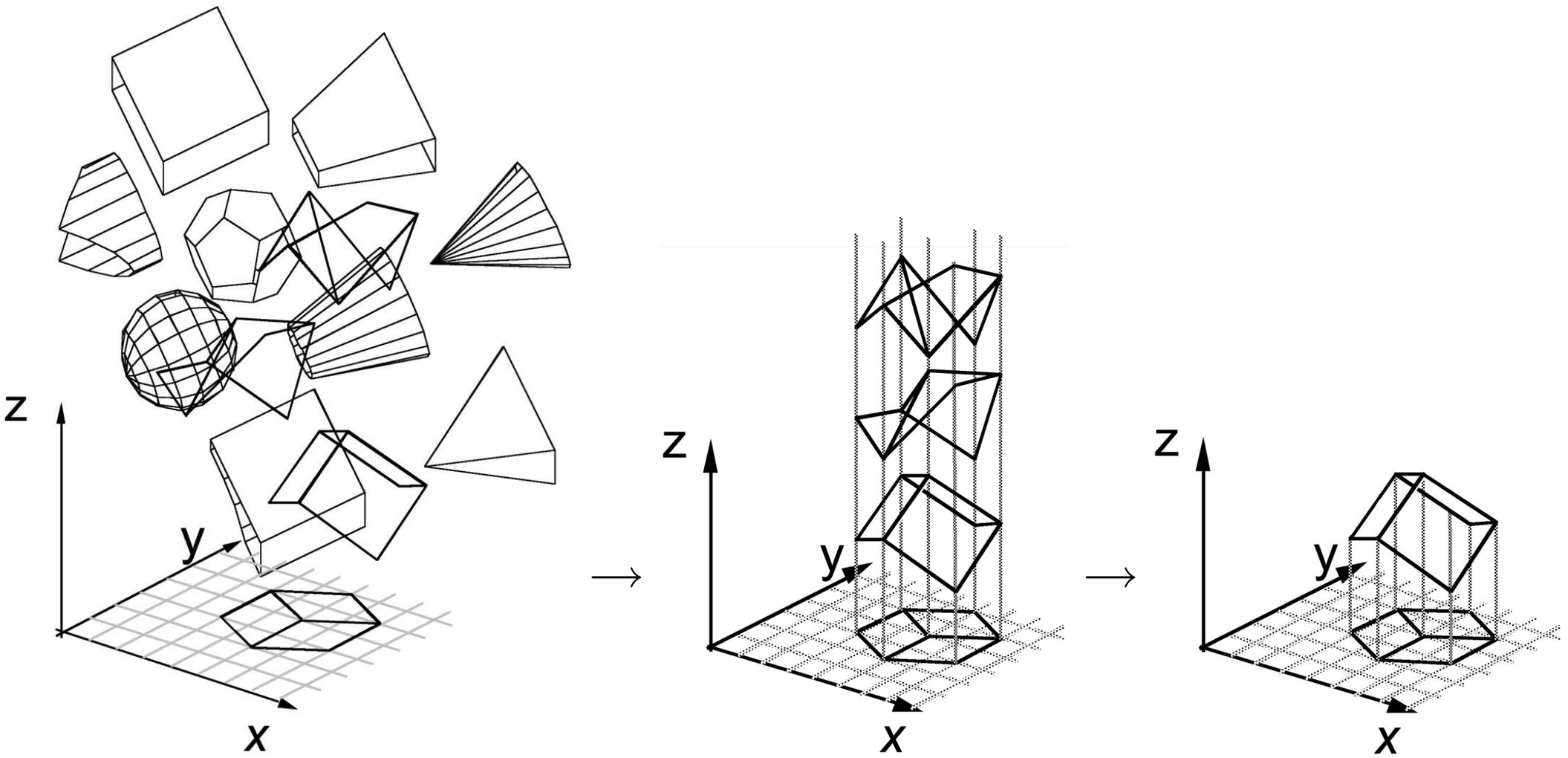
- many real-world patterns are hierarchical in structure
- interpretation of patterns depends on context
- essential for complex recognition tasks

from Torralba, 2003

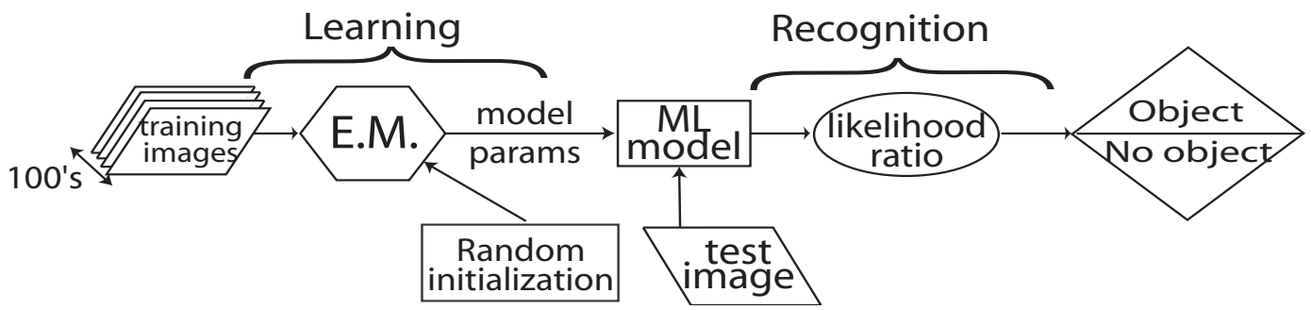


Recognition should include prior knowledge

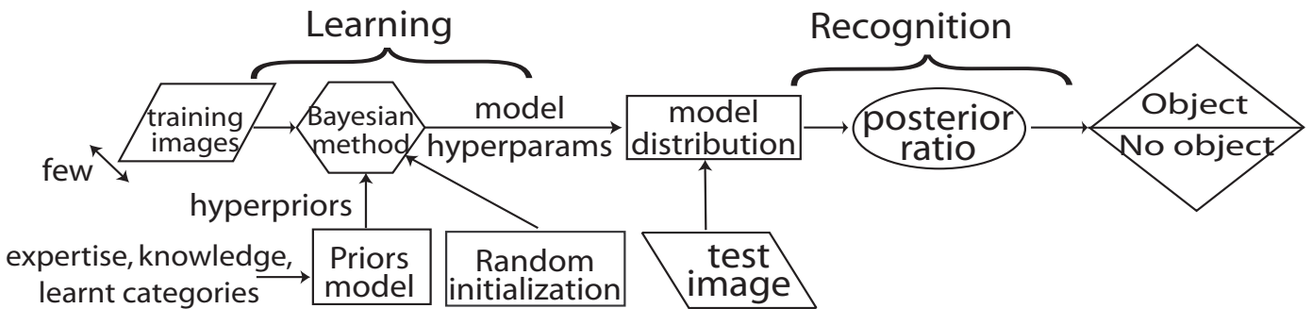
$$p(\hat{S}|I,C) = \arg\max_{S} \frac{p(I|S,C)p(S|C)}{p(I|C)}$$



Using prior knowledge in feature models (Fei-Fei et al, 2007)



(a) Maximum Likelihood (ML)



(b) Bayesian Algorithm

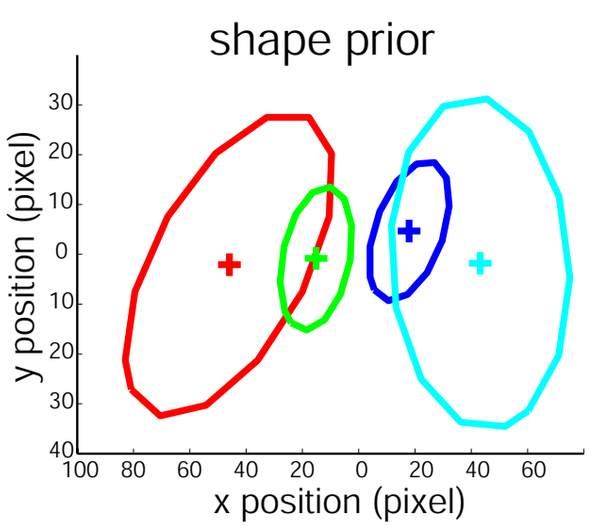
$$p(\mathcal{X}, \mathcal{A}|\,\boldsymbol{\theta}) = \sum_{\mathbf{h}\in H} p(\mathcal{X}, \mathcal{A}, \mathbf{h}|\,\boldsymbol{\theta}) = \sum_{\mathbf{h}\in H} \underbrace{p(\mathcal{A}|\mathbf{h}, \boldsymbol{\theta})}_{Appearance}\ \underbrace{p(\mathcal{X}|\mathbf{h}, \boldsymbol{\theta})}_{Shape}$$

compute image probability by summing over image hypotheses

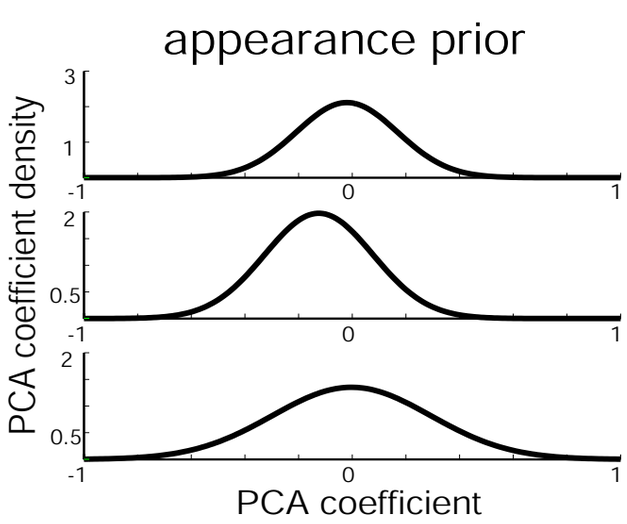
Caltech 101 Database: 101 categories, 9144 images

Google Background

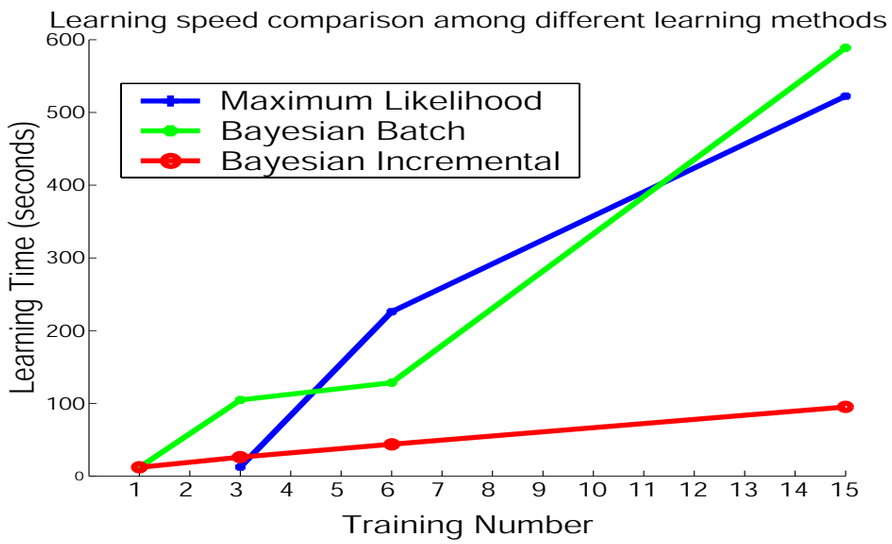
Learn shape and appearance priors



(a) shape prior



(b) appearance prior.



(c) Learning time comparison.

Fig. 3: **(a)-(b)** Prior distribution for shape mean ($\boldsymbol{\mu}^{\mathcal{X}}$) and appearance mean ($\boldsymbol{\mu}^{\mathcal{A}}$) for all the categories to be learned. Each prior's hyperparameters are estimated from models learned with maximum likelihood methods, using "the other" datasets [8]. Only the first three PCA dimensions of the appearance priors are displayed. All four parts of the appearance begin with the same prior distribution for each PCA dimension. **(c)** Average learning time for ML, Bayesian Batch and Bayesian Incremental methods over all 101 categories.

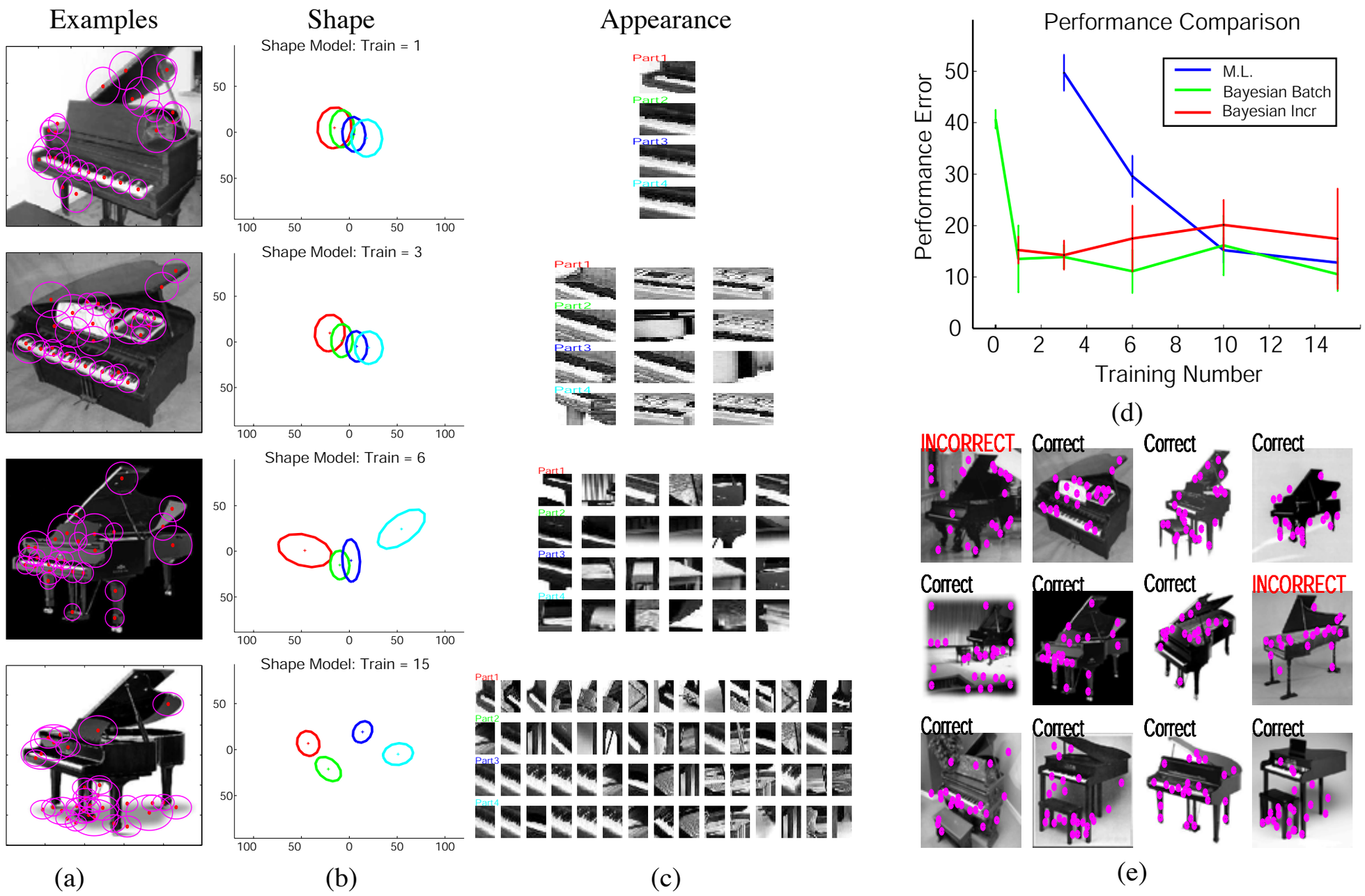
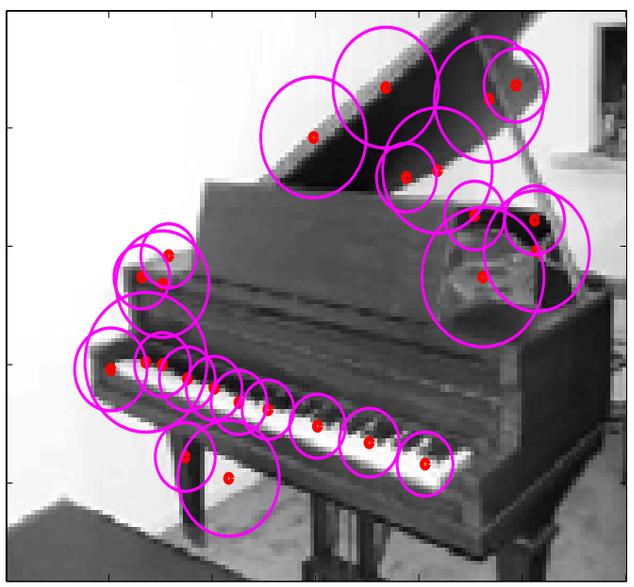
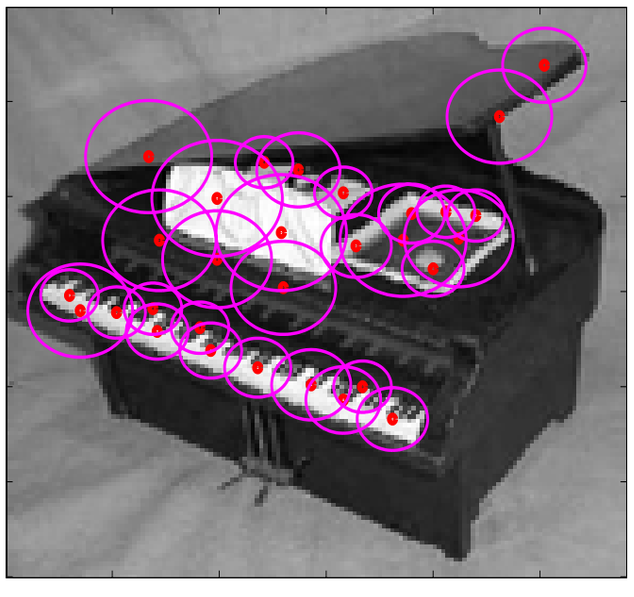


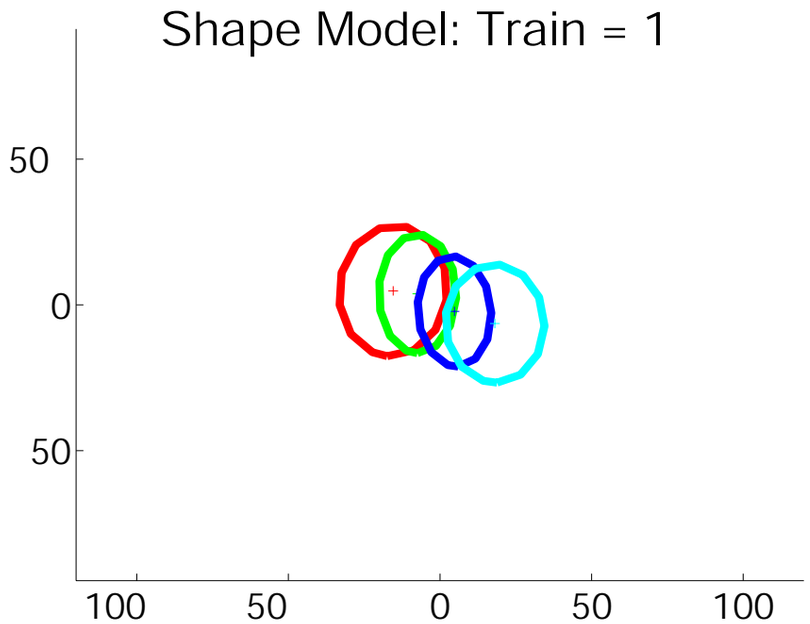
Fig. 4: Results for the "grand-piano" category. Panel (a) shows examples of feature detection. Panel (b) shows the shape models learned at Training Number $= (1, 3, 6, 15)$. Similarly to Fig.3(a), the x-axis represents the x position, measured by pixels, and the y-axis represents the y position, measured by pixels. Panel (c) shows the appearance patches for the model learned at Training Number $= (1, 3, 6, 15)$. Panel (d) shows the comparative results between ML, Bayesian Batch and Bayesian Incremental methods (the error bars show the variation over the 10 runs). Panel (e) shows recognition result for the incremental method at Training Number $= 1$. Pink dots indicate the center of detected interest points.

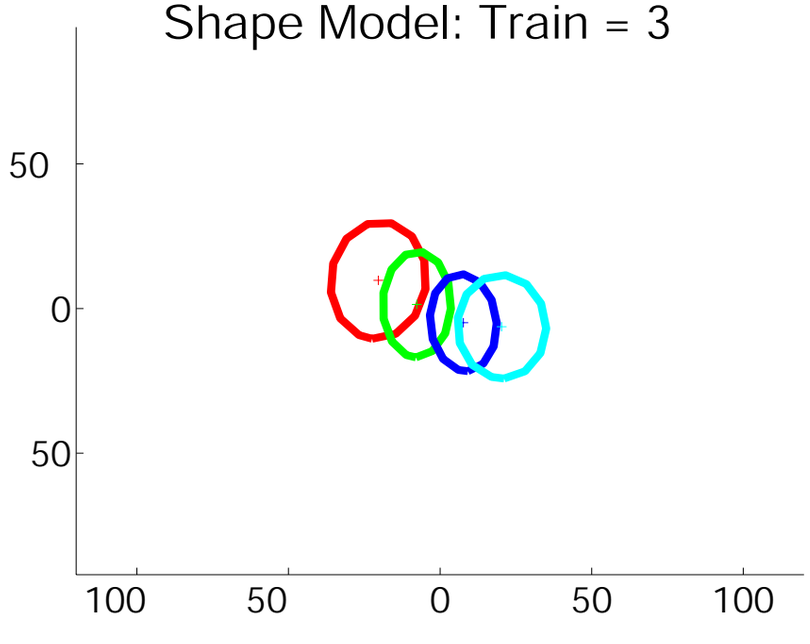
Examples



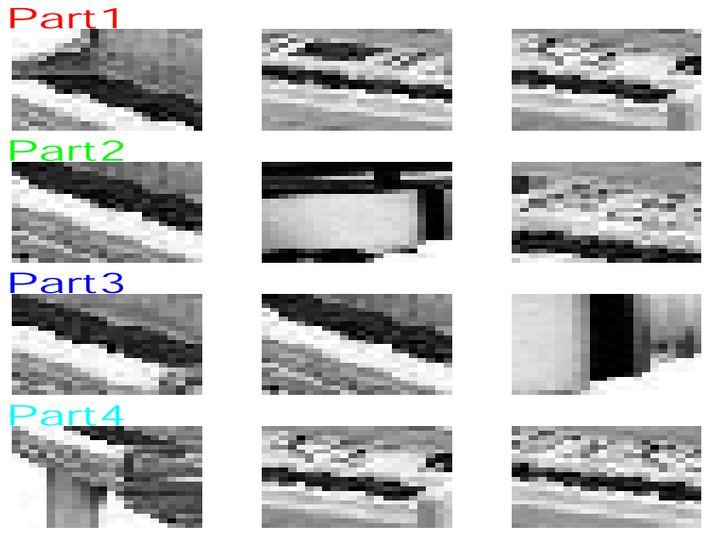


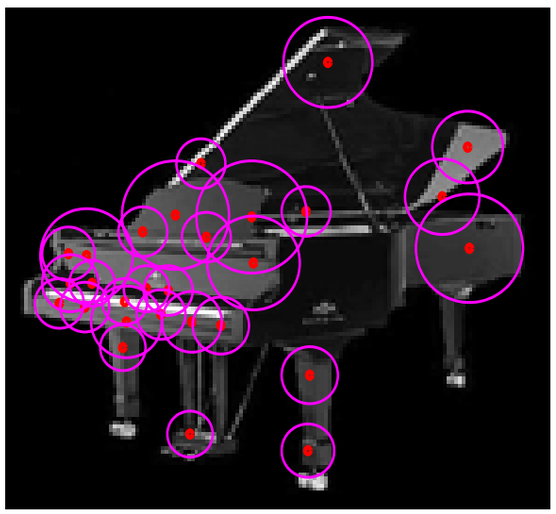
Shape

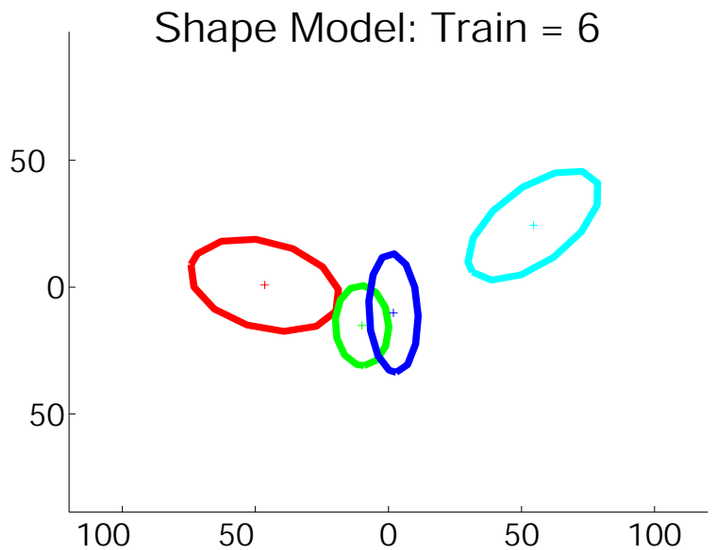


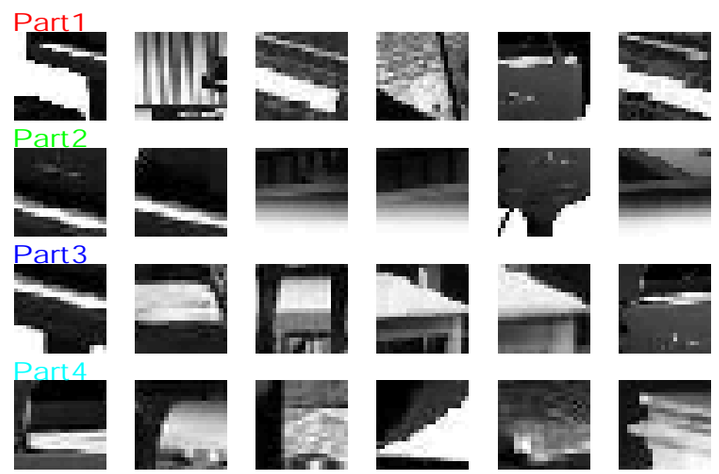


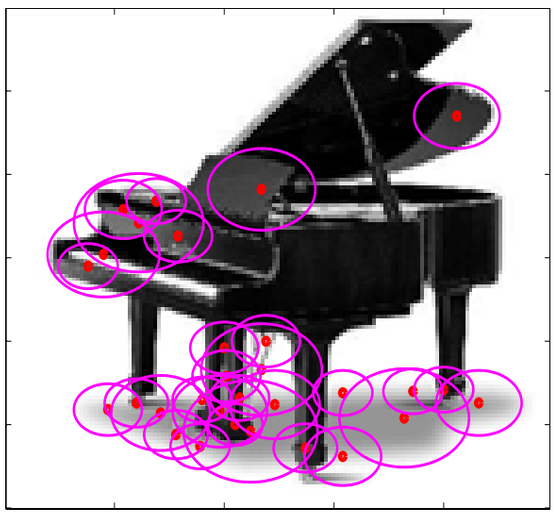
Appearance

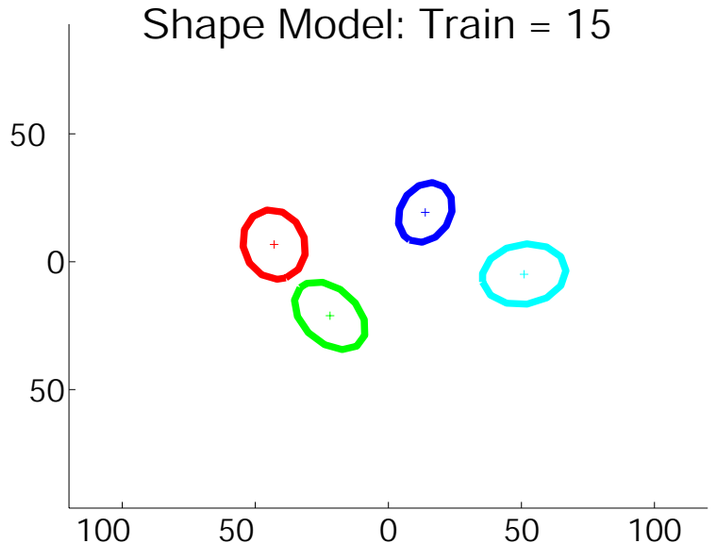


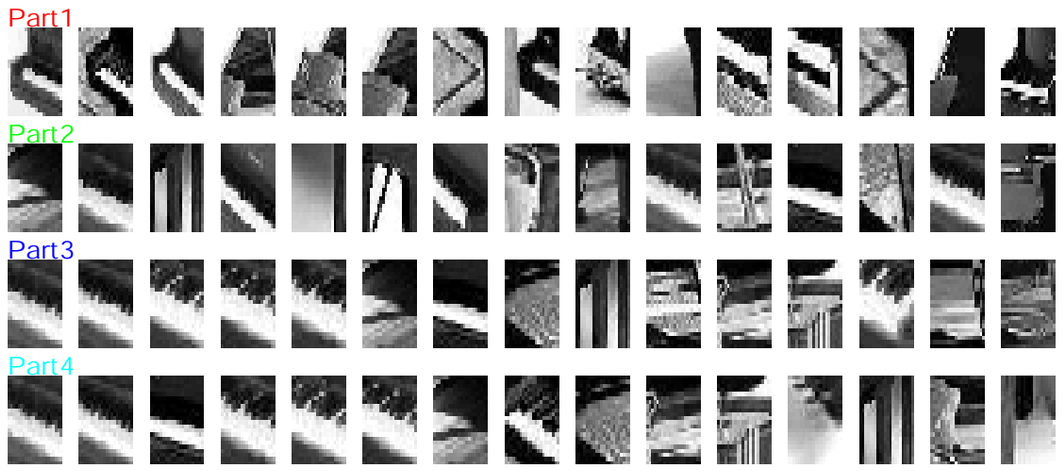


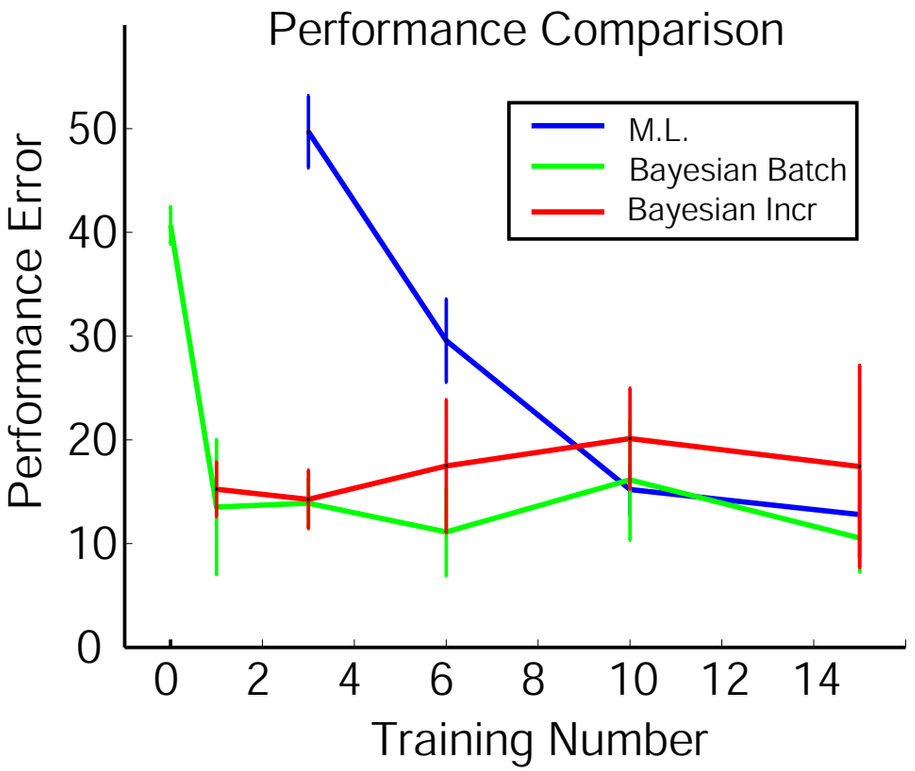

Shape Model: Train = 6
50
0
50
100
50
0
50
100

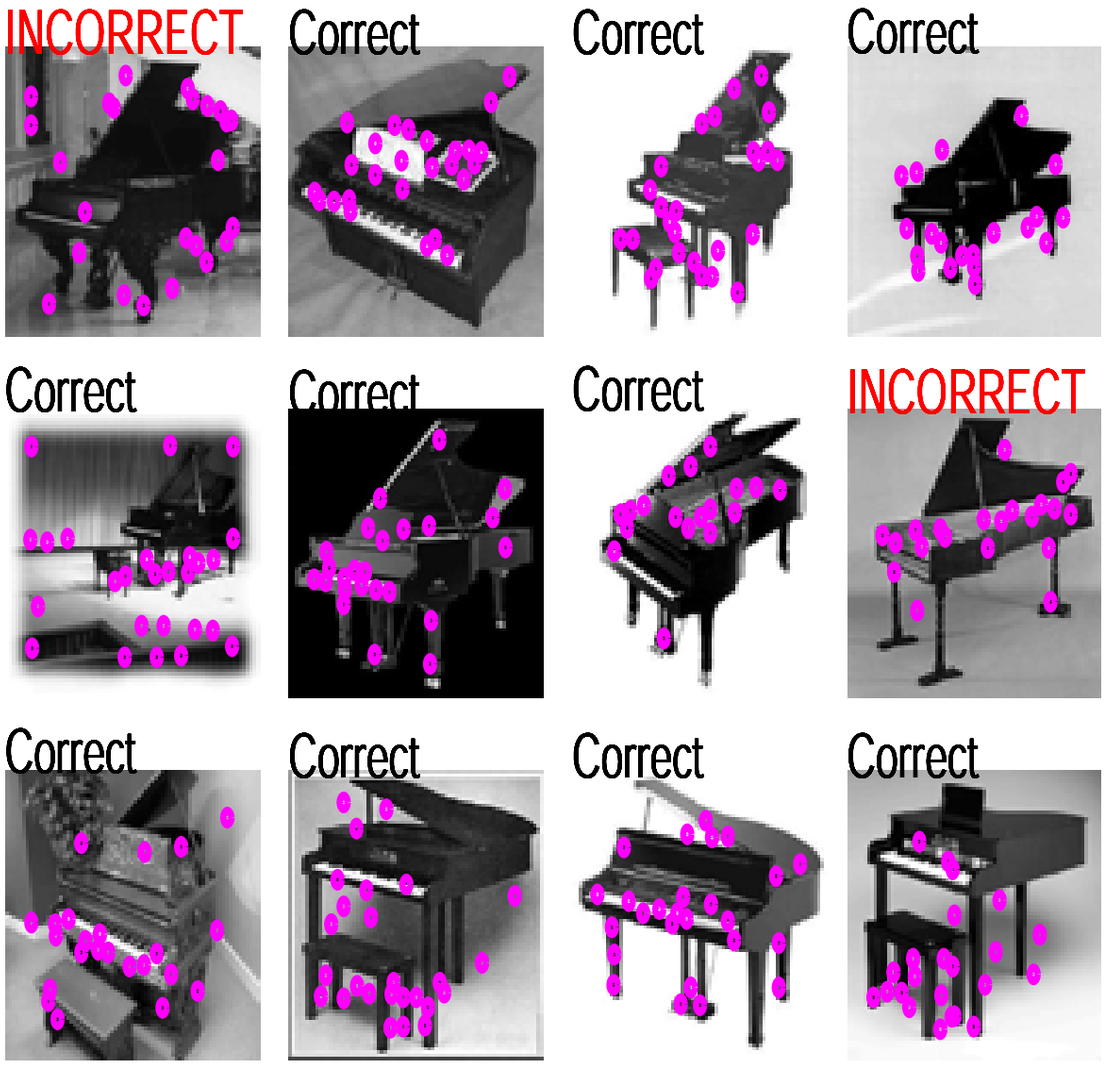

Part 1
Part 2
Part 3
Part 4



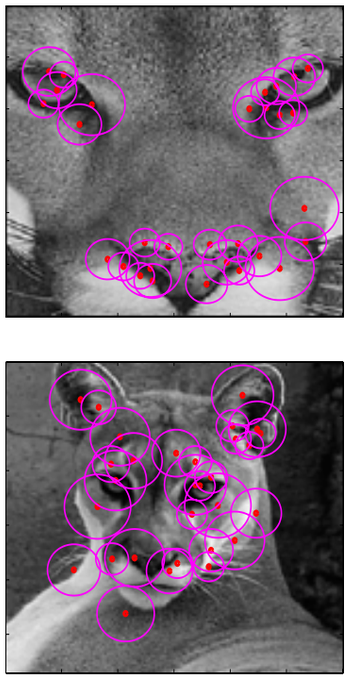

Shape Model: Train = 15
50
0
50
100
50
0
50
100


Part 1
Part 2
Part 3
Part 4

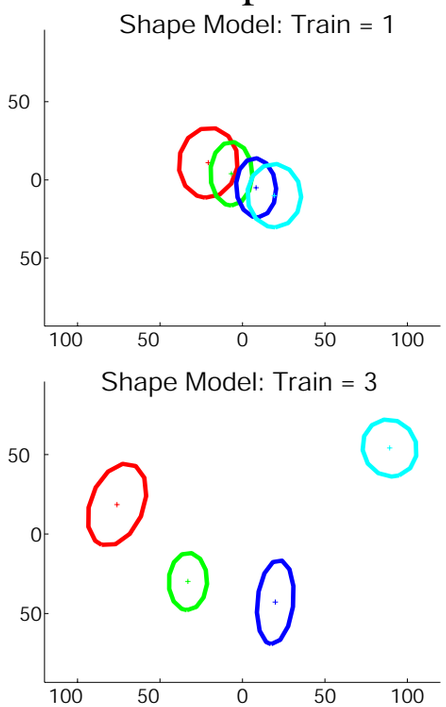

Performance Comparison
M.L.
Bayesian Batch
Bayesian Incr
Performance Error
Training Number
0
10
20
30
40
50
0
2
4
6
8
10
12
14

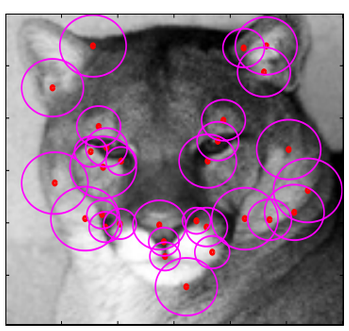

INCORRECT
Correct
Correct
Correct
Correct
Correct
Correct
Correct
INCORRECT
Correct
Correct
Correct
Correct

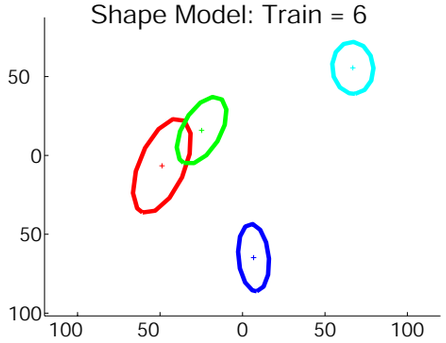
Examples

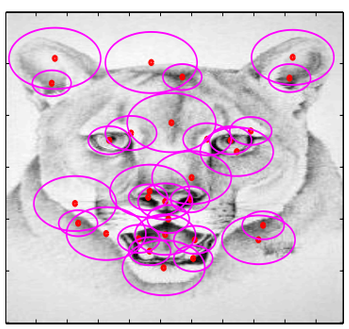


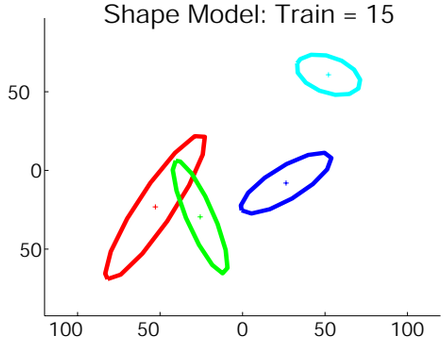
Shape



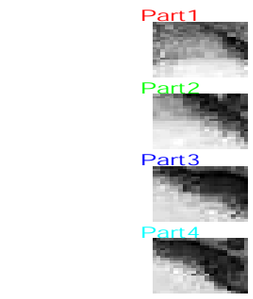


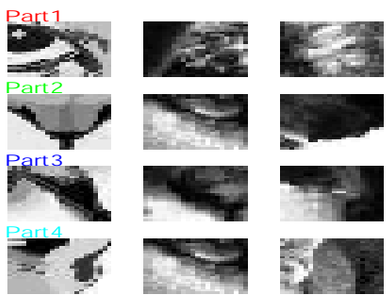


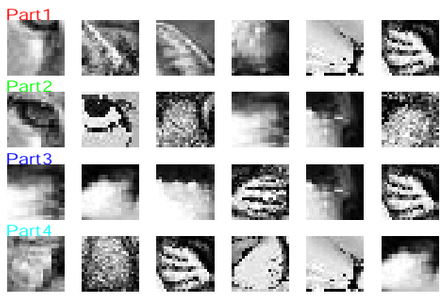


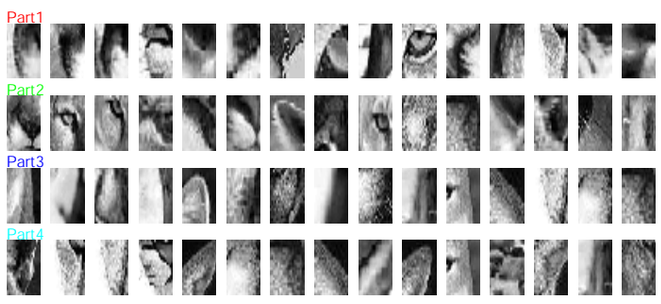


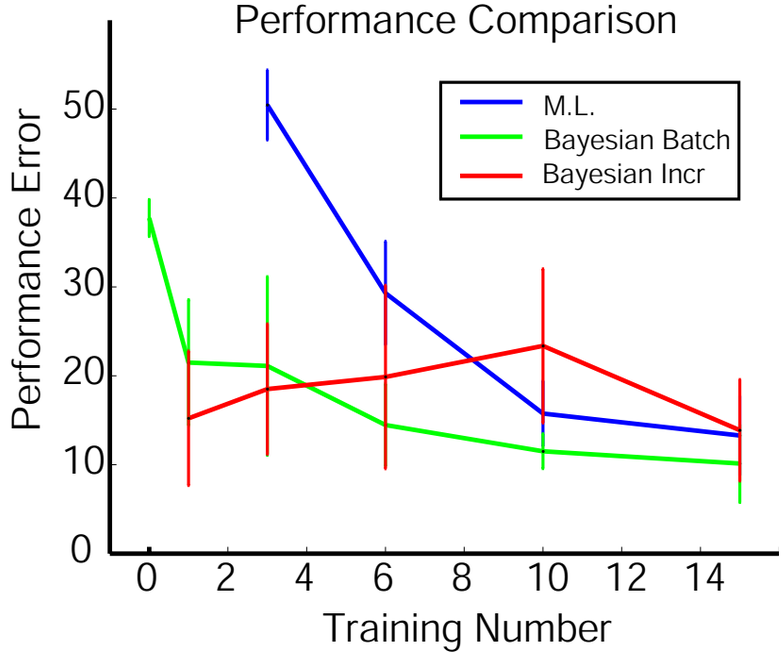
Appearance



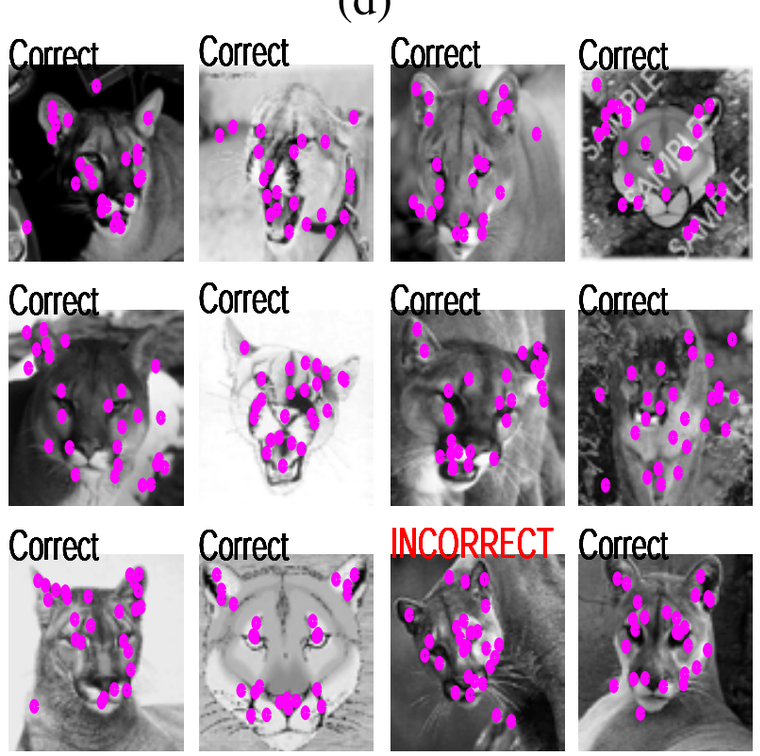


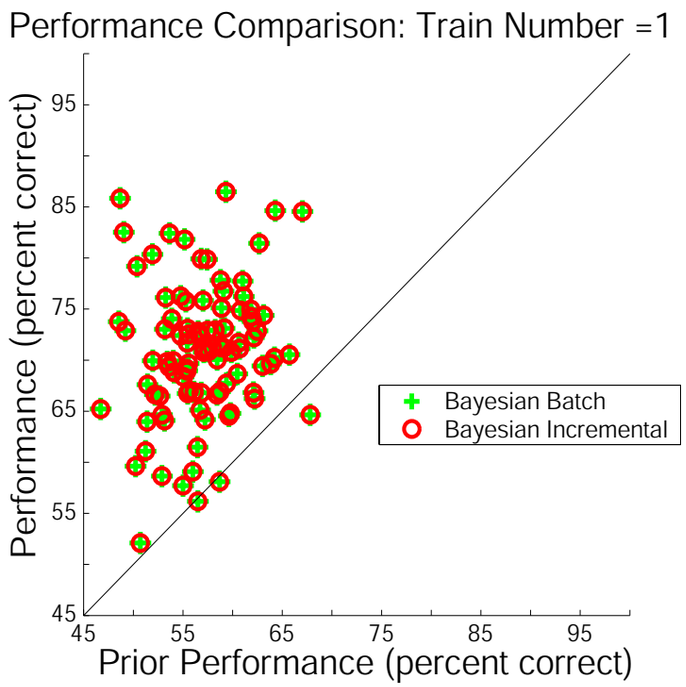




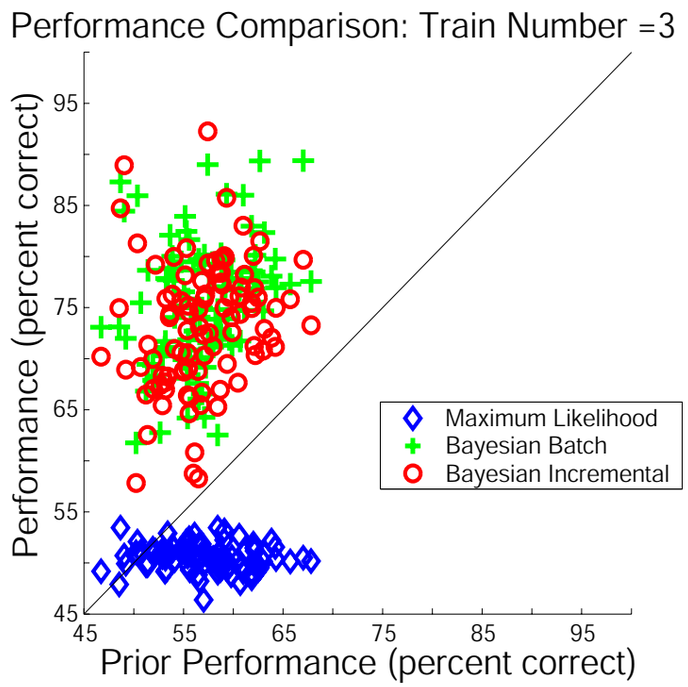


(d)

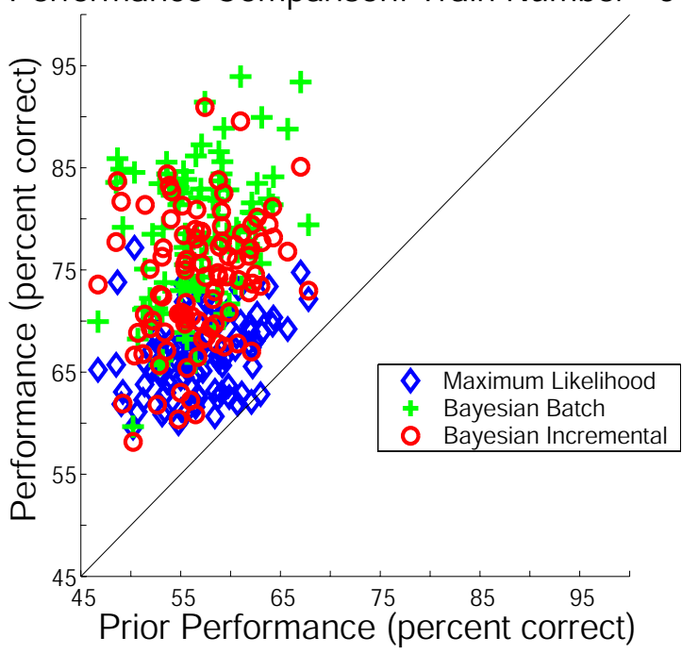




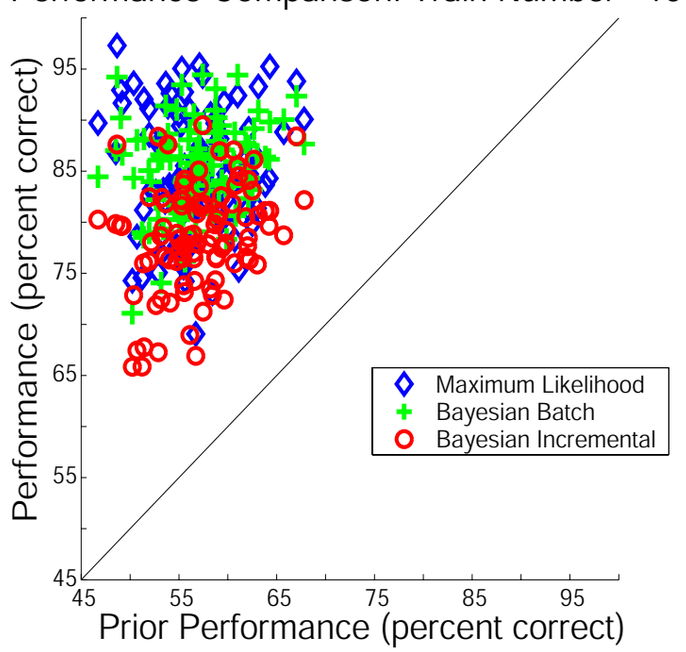
(a) Training = 1



(b) Training = 3



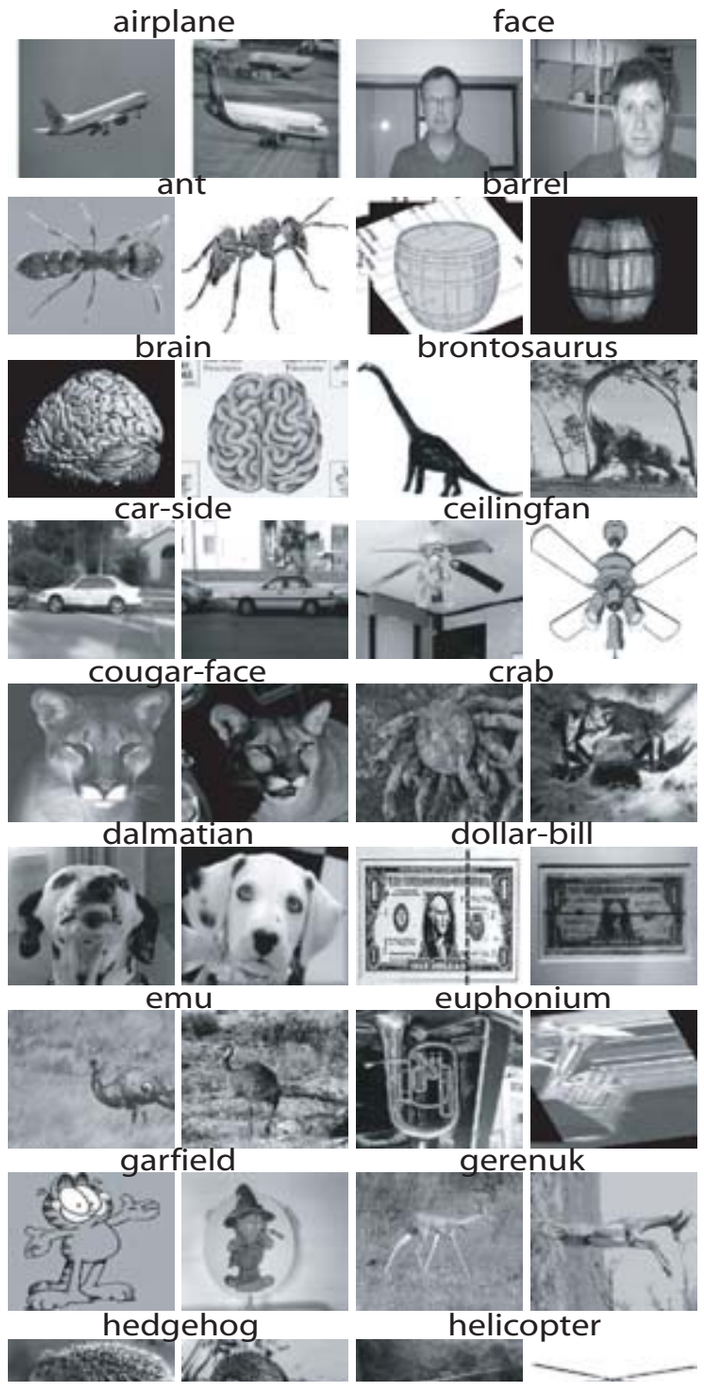
(c) Training = 6



(d) Training = 15

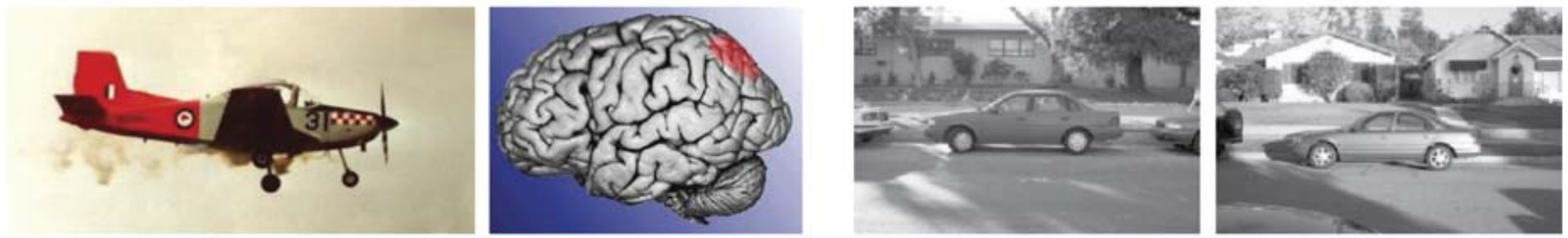
Why is object recognition hard? (Pinto, Cox, and DiCarlo, 2008)

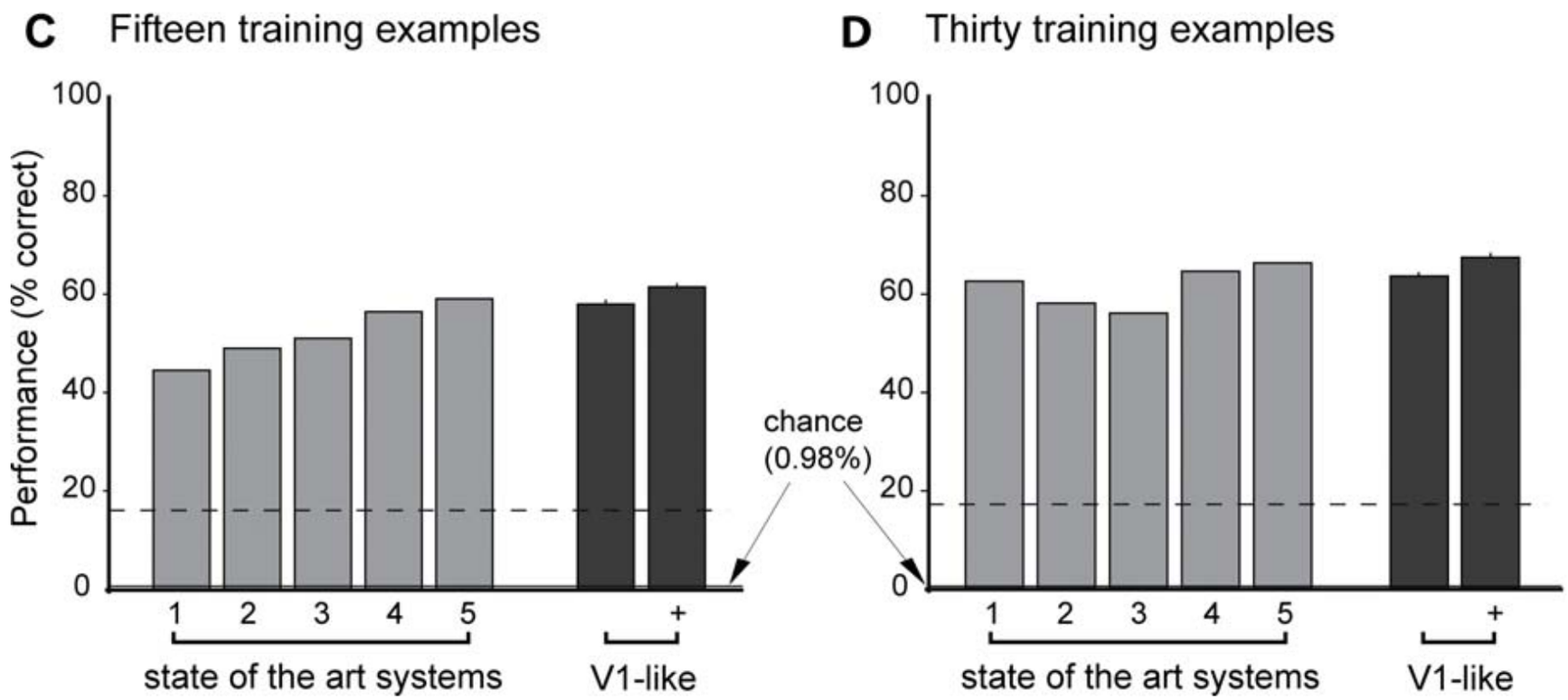
- Data object recognition databases like Caltech 101 seem hard: 101 categories, 9000 images
- systems seem to do well, 60+% performance, chance < 1%
- But there are biases:
  - poorly controlled variation
  - object placement not random
- How well do these really test core problems?
- How well would the simplest model do?



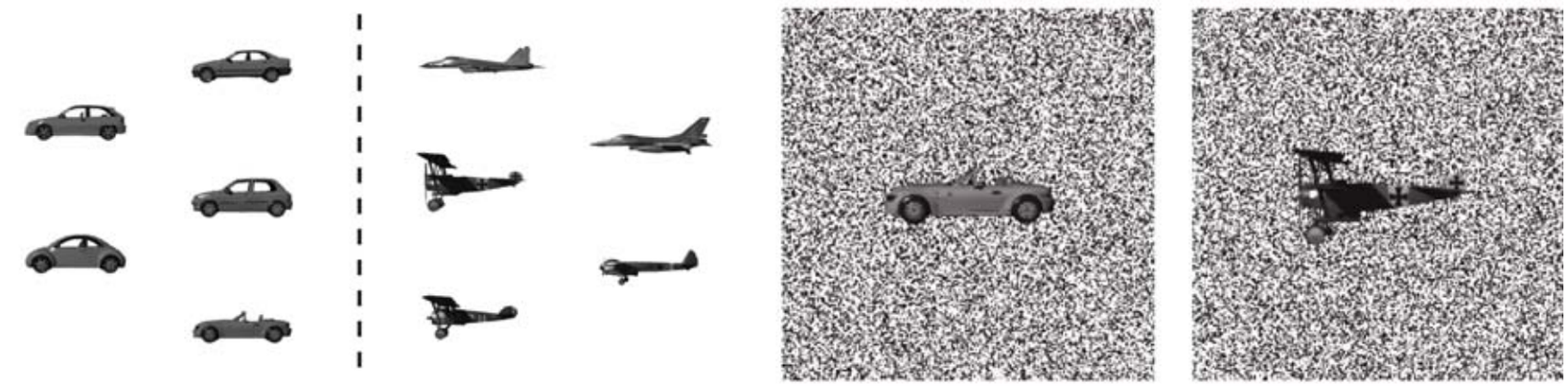
Why is object recognition hard? (Pinto, Cox, and DiCarlo, 2008)

- Compare state of the art systems with controlled variability





Introduce explicit, controlled variation

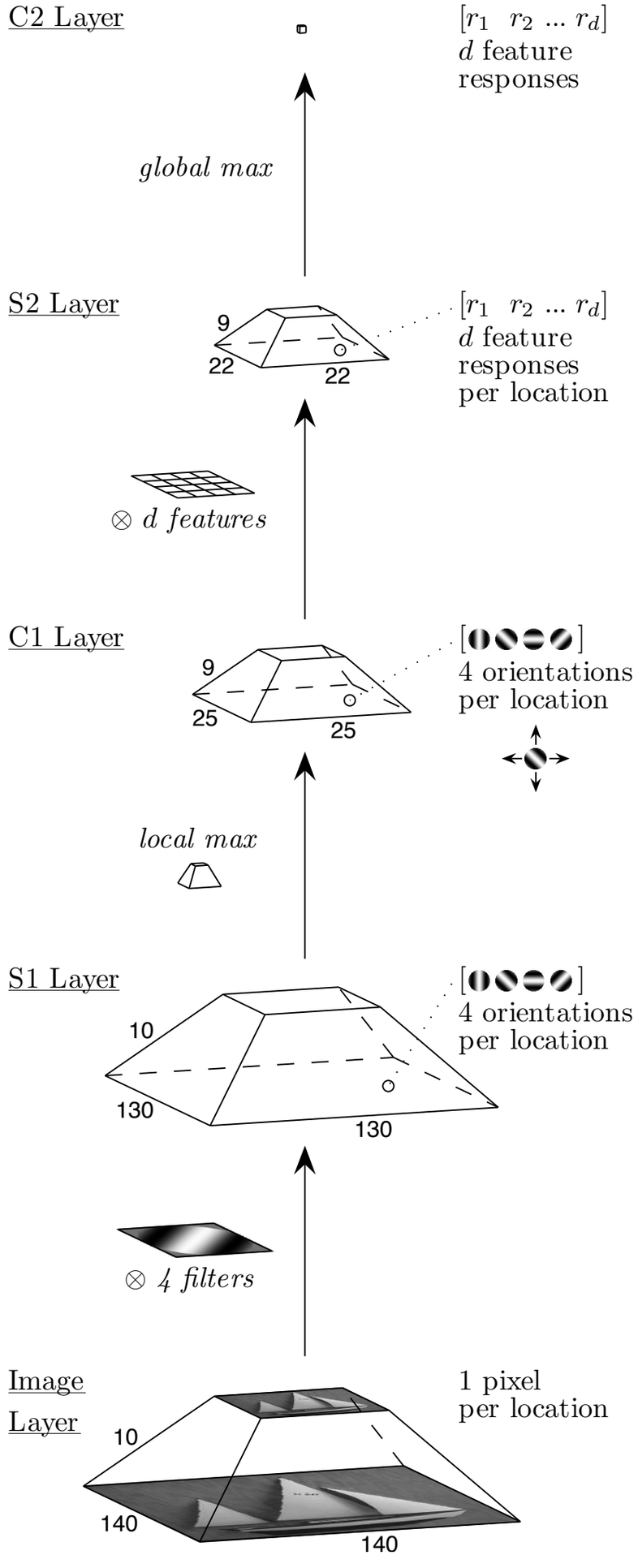


Comparing state-of-the-art visual features (Pinto et al, 2011)

- How well do generic models do?
- performance on Caltech 101: avg accuracy for 15 training and 15 test examples
- 5 state of the art algorithms and 2 baselines
  - baseline: just classify a pixel representation
  - SIFT features
  - SLF: sparse localized features
  - PHOG: pyramid histogram of gradients
  - PHOW: pyramid histogram of visual words (words are SIFT features, quantized using K-means clustering)
  - Geometric blur: apply spatially varying blur to edge points
  - “V1 like” known properties of simple cells (normalized Gabor functions)

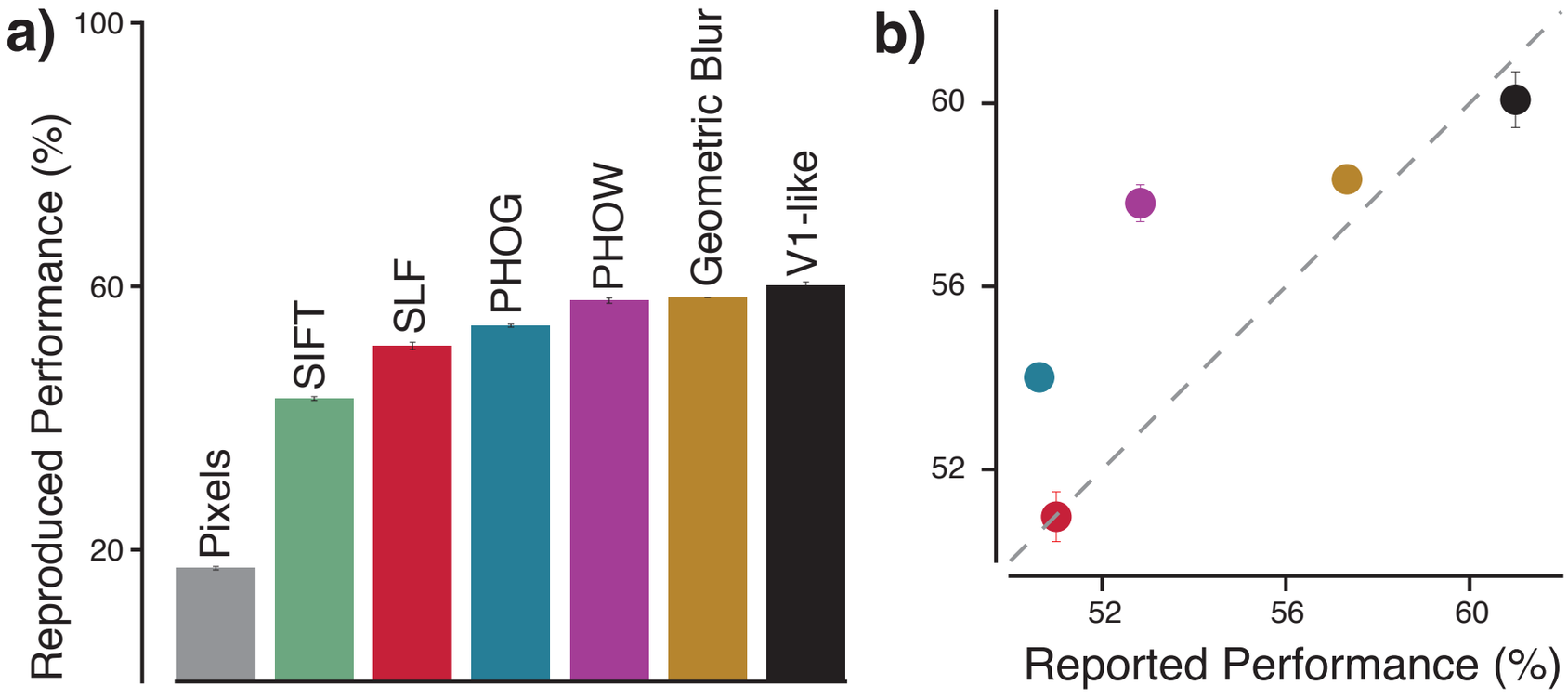
Sparse Localized Features (SLF)

- extension of C2 features of HMAX model
- apply Gabor filters at all positions and scales
- each layer has units covering x, y position and scale
- feature complexity and position/ scale invariance achieved by alternating template matching and max pooling operations
- increase sparsity of features by constraining number of feature inputs, lateral inhibition, and feature selection



Comparing state-of-the-art visual features (Pinto et al, 2011)

- performance on Caltech 101: avg accuracy for 15 training and 15 test examples
- 5 state of the art algorithms and 2 baselines
  - baseline: just classify a pixel representation
  - SIFT features
  - SLF: sparse localized features
  - PHOG: pyramid histogram of gradients
  - PHOW: pyramid histogram of visual words (words are clustered SIFT features)
  - Geometric blur: apply spatially varying blur to edge points
  - "V1 like" known properties of simple cells (normalized Gabor functions)



Achieves “state-of-the-art” performance

- Classification results:
  - 80.5% for bikes
  - 70.1% for cars
  - 81.7% for people

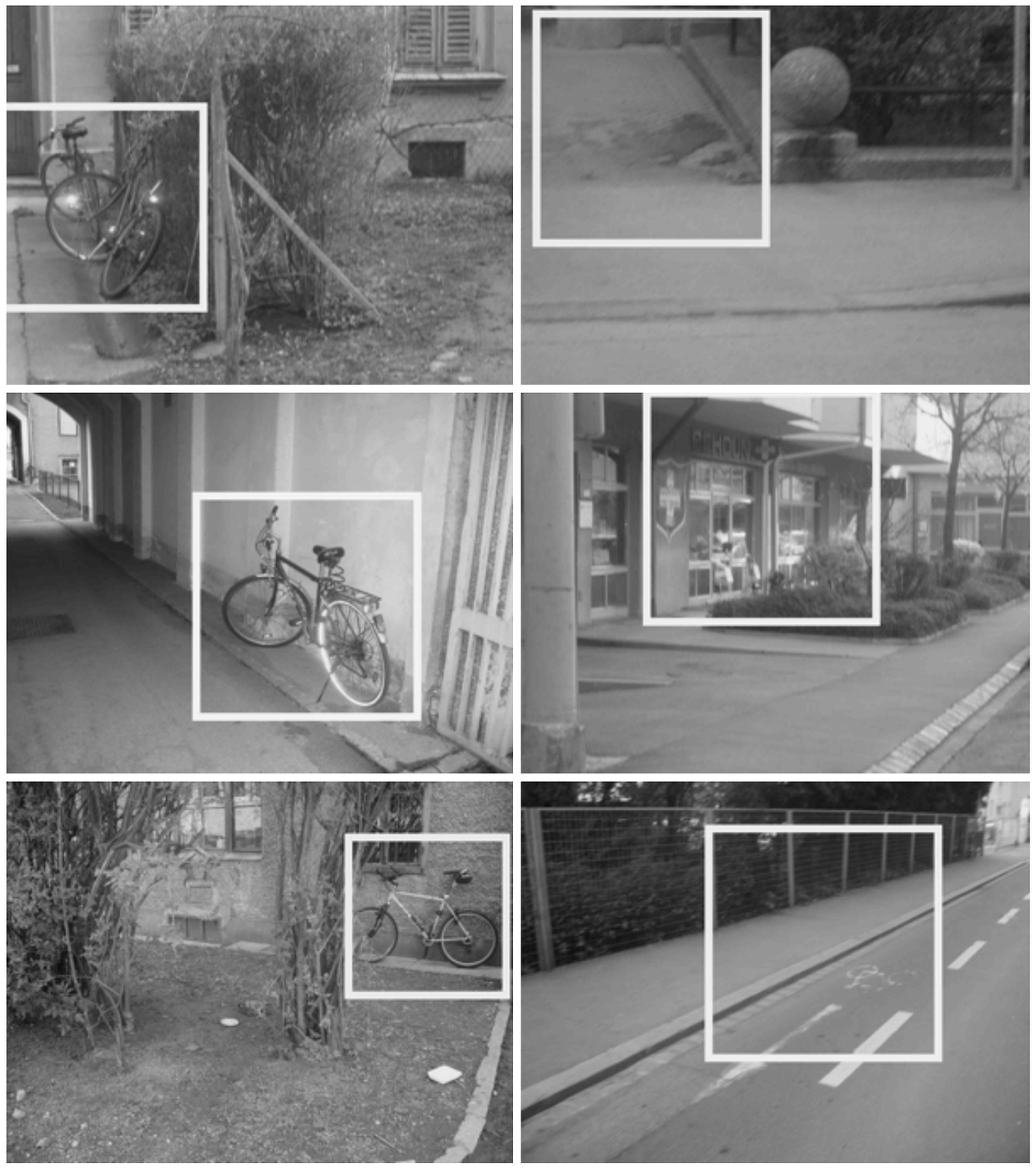
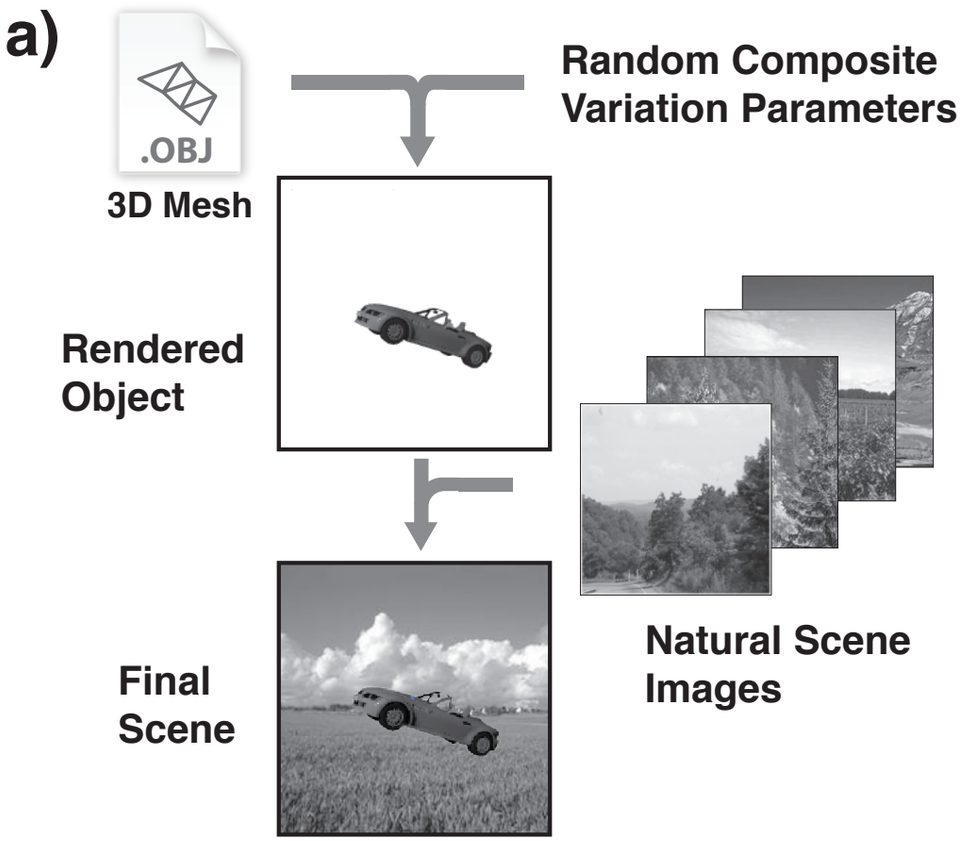
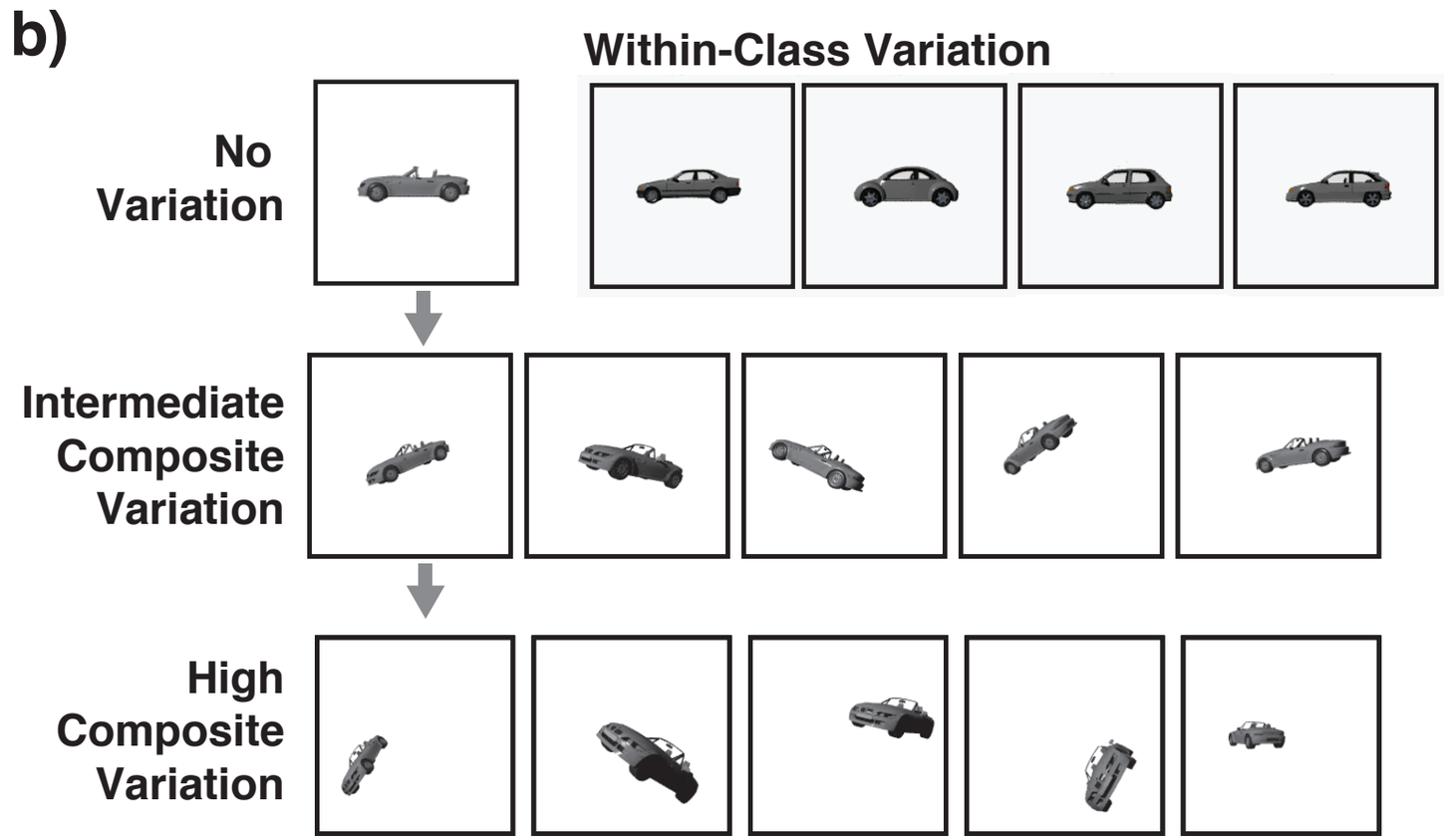


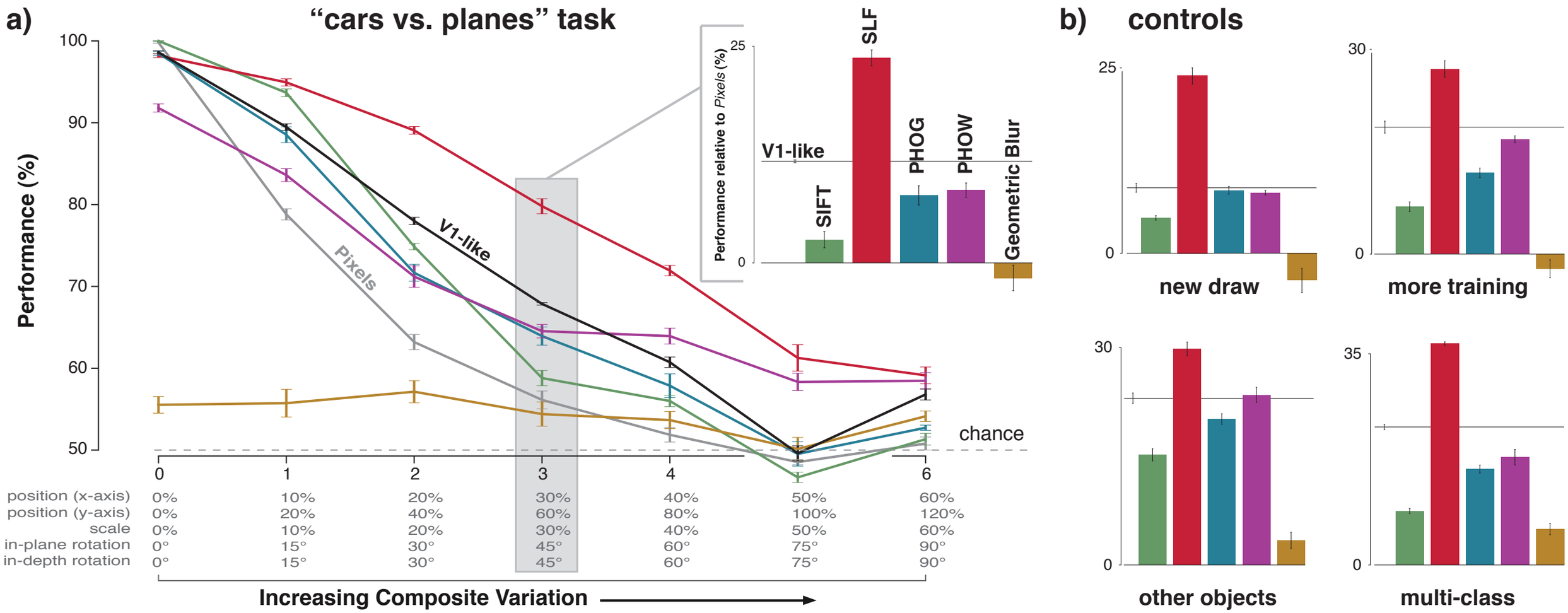
Figure 15. Some subimages used to train our Graz-02 “bikes” classifier.

Generating object data with controlled variation



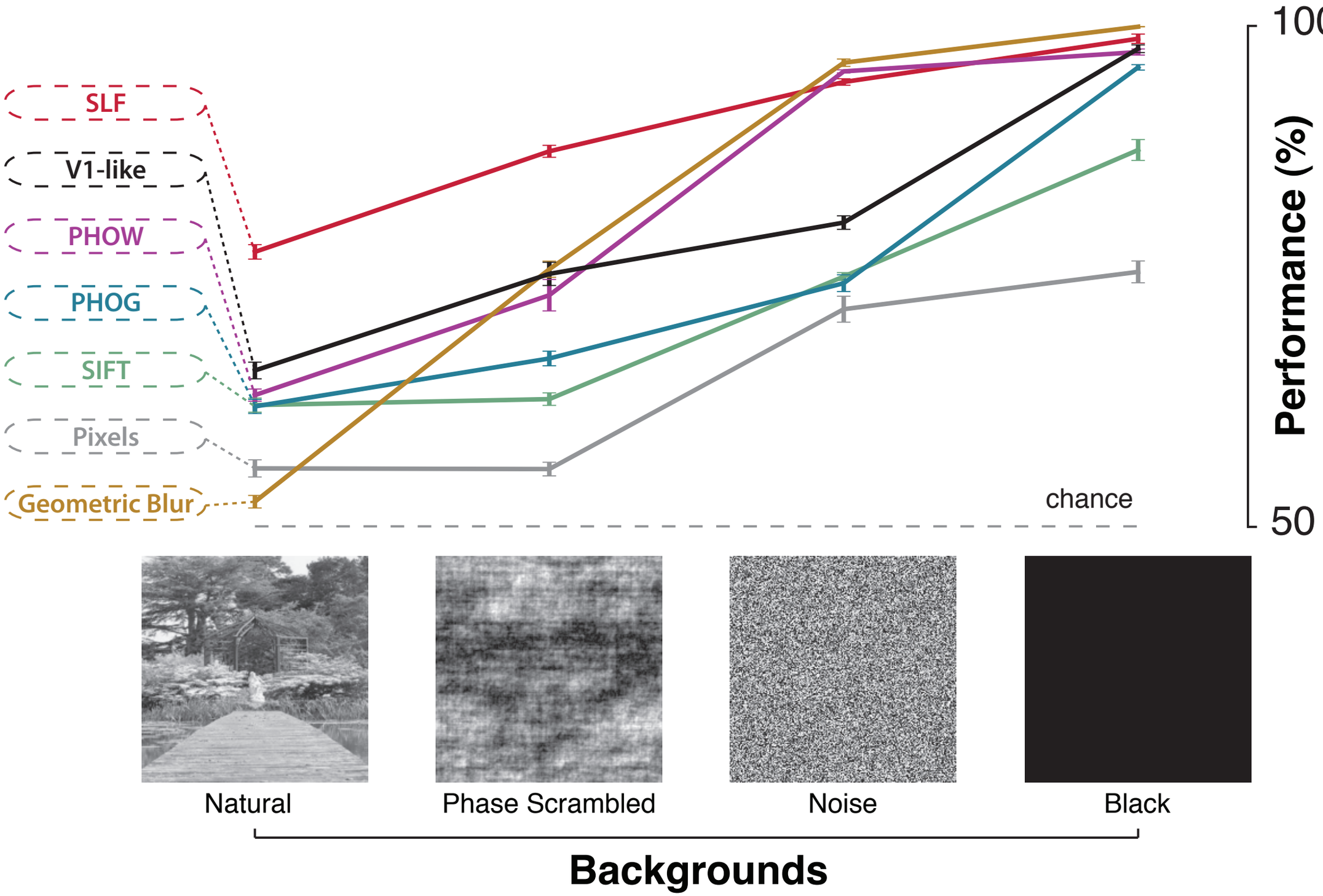


- render objects using 3D mesh
- control view parameters
- composite on randomly selected natural backgrounds
- test algorithm performance again

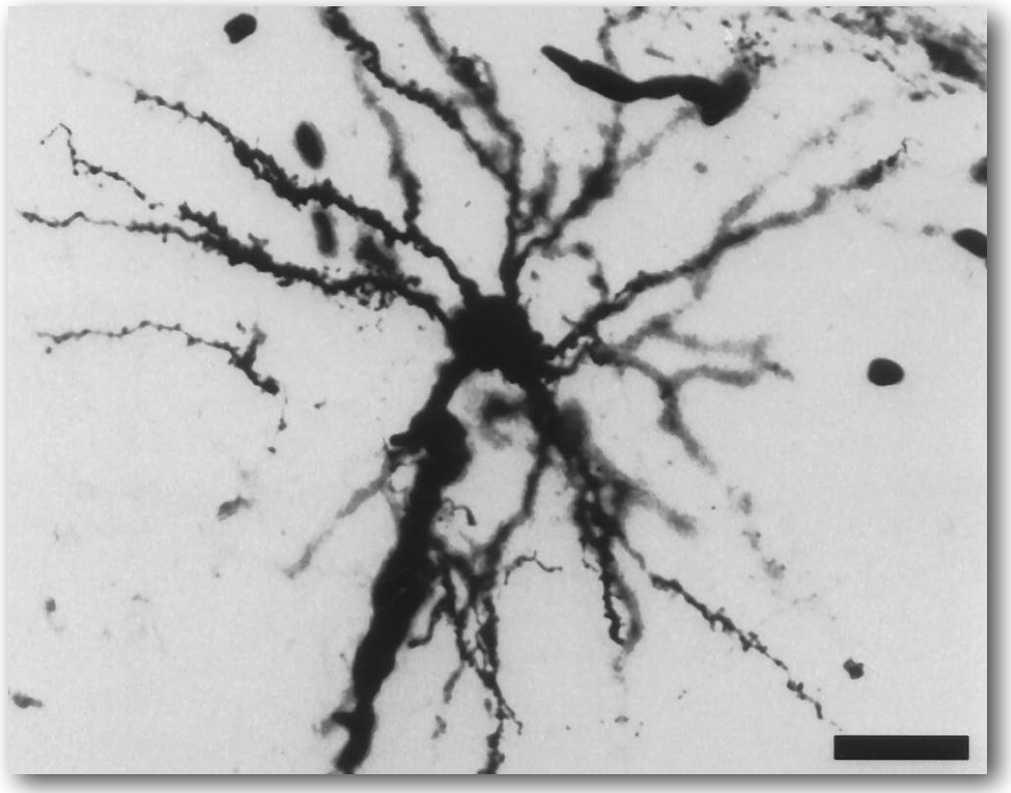


- decide “car” or “plane”
- increasing variation systematically reduces performance to chance for all models
- only the SLF model does significantly better (but still not great) than the baseline Gabor filter model

Effect of background type



My own neural recordings from my PhD studies



A neuron in an auditory brain area

Neural response

Sound waveform

Brains vs computers

Brains (adult cortex)

- surface area: 2500 cm^2
- squishy

- neurons: 20 billion
- synapses: 240,000 billion

- neuron size: 15,000 nm
- synapse size: 1,000 nm

- synaptic OPS: 30,000 billion

Intel quad core (Nehalem)

- surface area: 107 mm^2
- crystalline

- transistors: 0.82 billion

- transistor size: 45 nm

- FLOPS: 45 billion

Deep Blue: 512 processors, 1 TFLOP

Brains vs computers: energy efficiency

Brains (adult cortex)

- surface area: 2500 cm^2
- squishy

- neurons: 20 billion
- synapses: 240,000 billion

- neuron size: 15,000 nm
- synapse size: 1,000 nm

- synaptic OPS: 30,000 billion

- power usage: ~12 W
- **2500 GFLOPS/W**

Intel quad core (Nehalem, 2008)

- surface area: 107 mm^2
- crystalline

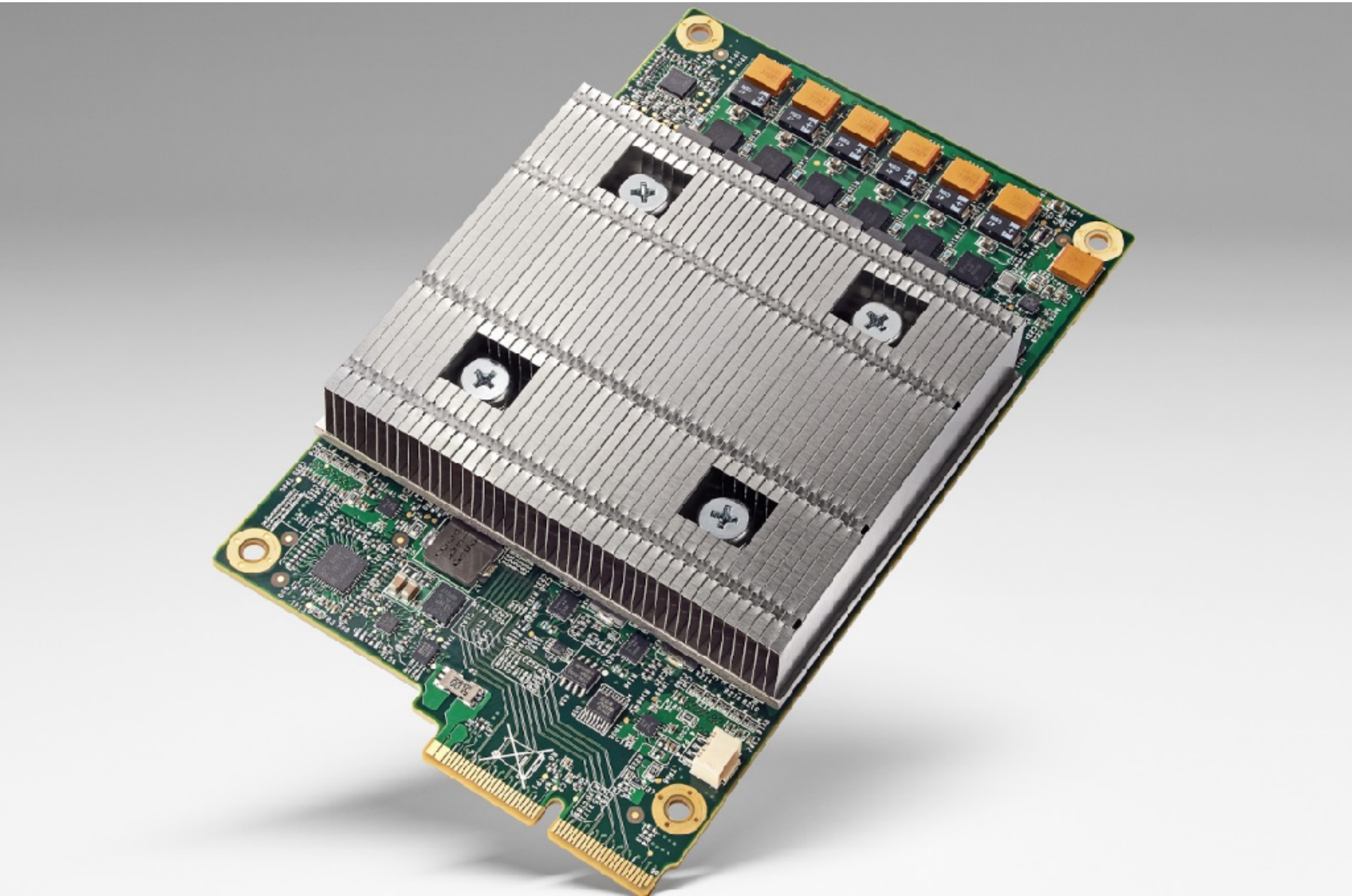
- transistors: 0.82 billion

- transistor size: 45 nm

- FLOPS: 45 billion

- power usage: 60 W
- **0.75 GFLOPS/W**

Google's Deep Learning Chip: The Tensor Processing Unit



Brains vs computers: energy efficiency

Brains (adult cortex)

- surface area: 2500 cm^2
- squishy

- neurons: 20 billion
- synapses: 240,000 billion

- neuron size: 15,000 nm
- synapse size: 1,000 nm

- synaptic Terra OPS: ~30

- power usage: ~12 W
- **2.5 TOPS/W**

Google TPU 2.0

- surface area: ~331 mm^2
- crystalline

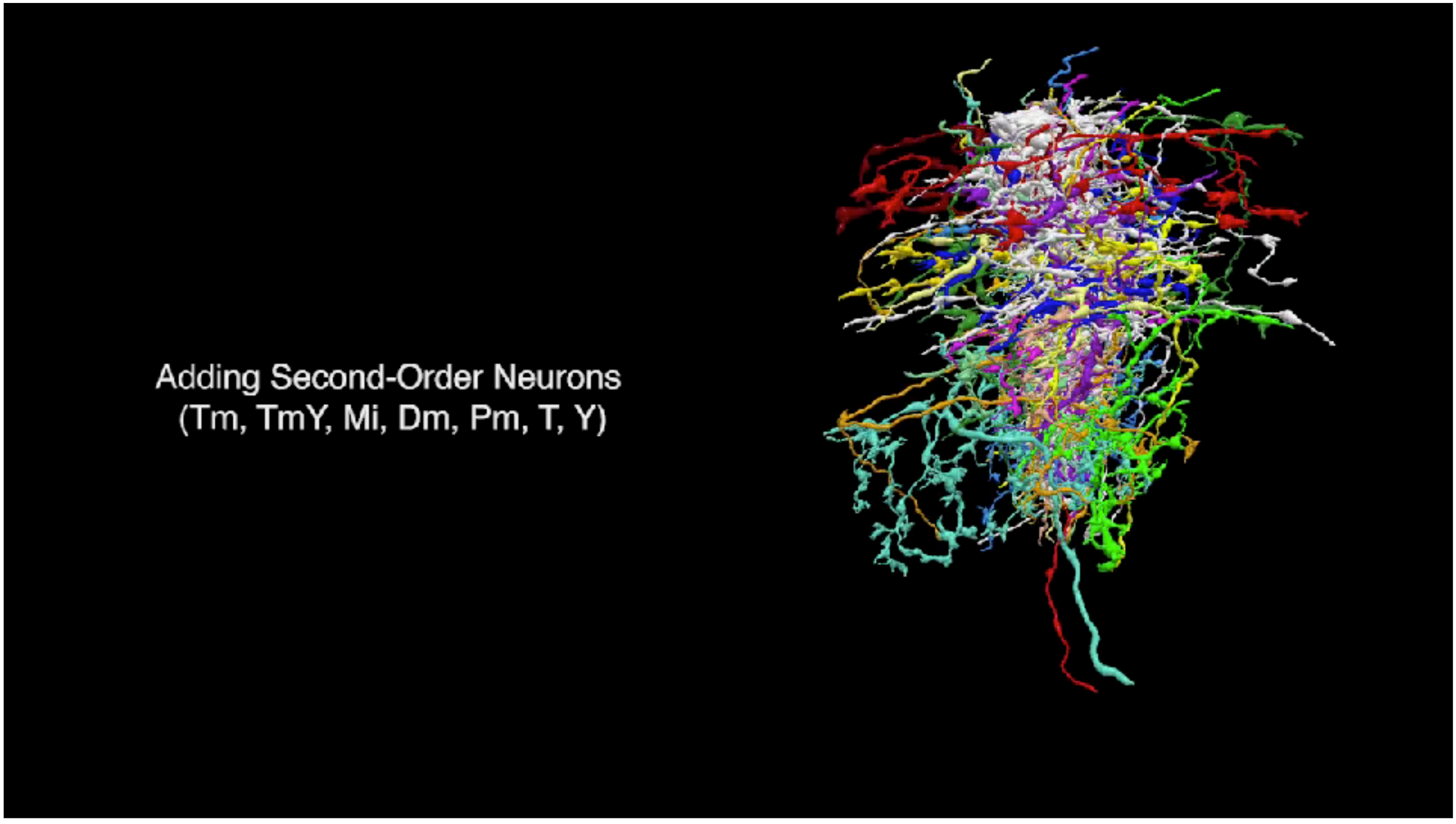
- transistors: 2-2.5 billion

- transistor size: 28 nm

- Terra OPS (8 bit): 92
  (TPU 2.0 is 180 teraops)
- power usage: 40 W
- **2.3 TOPS/W**

  (4.5 for TPU 2.0 assuming 40W)

Reconstruction of neural motion processing circuits in a fly



A visual motion detection circuit suggested by *Drosophila* connectomics

379 neurons; 8,637 chemical synaptic connections