9.54
class 6

Supervised learning

Gradient descent, Stochastic gradient descent, applications

Shimon Ullman + Tomaso Poggio

Danny Harari + Daneil Zysman + Darren Seibert

Center for Brains,
Minds & Machines

“Linear” learning

Suppose

$$x_i \in \mathbb{R}^n \; and \; y_i \in \mathbb{R}^m, \; i = 1, \cdots, N$$

Define

$$(x_1, \cdots, x_N) = X \; and \; (y_1, \cdots, y_N) = Y$$

Find linear operator (eg a matrix) such that

$$MX = Y$$

“Linear” learning

If X^{-1} exists, then

$$MX = Y \quad \Longrightarrow \quad M = YX^{-1}$$

If X^{-1} does not exists, then

$$MX = Y \quad \Longrightarrow \quad M = YX^{\dagger}$$

where the pseudo inverse is the solution of

$$min \; ||MX - Y||_F \qquad \text{with} \qquad ||A||_F = \sqrt{(\sum_{i,j} |a_{i,j}|^2)}$$

Thus $$Y - MX = 0$$

More in general look for M such that

$$min||Y - MX||^2$$

The solution is given by putting the gradient to zero

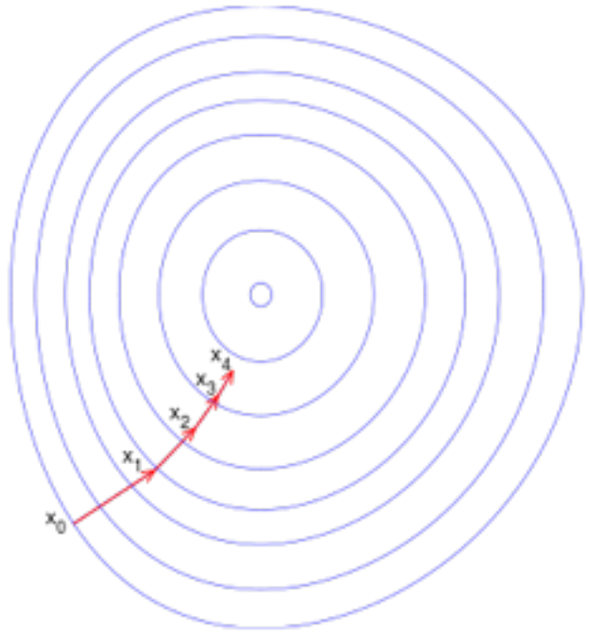
$$\nabla V(M) = 2(Y - MX)X^T = 0 \text{ yielding } YX^T = MXX^T \text{ that is } M = YX^T(XX^T)^{-1}$$

which is the same we had derived earlier…

How could minimization done in general, in practice, by the brain?
Probably not by analytic solution….

The gradient offers a general way to compute a solution to a minimization problem

$$\frac{dM}{dt} = -\gamma \nabla V(M)$$



finds the elements of M which correspond to $minV(M)$

As an example let us look again at

$$min||Y - MX||^2$$

Using $$\nabla V(M) = 2(Y - MX)X^T$$

Let us make the example more specific. Assume that y_i are scalar

Then $M = w^T$ and

$min_{m_{i,j}} ||MX - Y||^2$ becomes $min_{w \in R^d} \frac{1}{n} \sum_{1=1}^{n} (y_i - w^T x_i)^2$

yielding

$$\nabla V(M) = \nabla V(w_i^T) = \frac{2}{n} \sum_{i=1}^{n} (y_i - w^T x_i) x_i^T$$

and thus

$$\frac{dw^T}{dt} = -\gamma_t \sum_{i=1}^{n} (y_i - w_t^T x_i) x_i^T$$

Discretizing time in

$$\frac{dw^T}{dt} = -\gamma_t \sum_{i=1}^{n} (y_i - w_t^T x_i) x_i^T$$

we obtain

$$w_{t+1}^T = w_t^T - \gamma_t \sum_{i=1}^{n} (y_i - w_t^T x_i) x_i^T$$

Gradient descent has several nice properties but it is still not “biological”…

$$w_{t+1}^T = w_t^T - \gamma_t \sum_{i=1}^{n} (y_i - w_t^T x_i) x_i^T$$

can be written as

$$\frac{dw^T}{dt} = -\gamma_t \sum_{i=1}^{n} \nabla V_i(w)$$

Stochastic gradient descent is…

$$\frac{dw^T}{dt} = -\gamma_t \nabla V_i(w) \ , \ i = 1, \cdots , n$$

Learning is an ill-posed problem

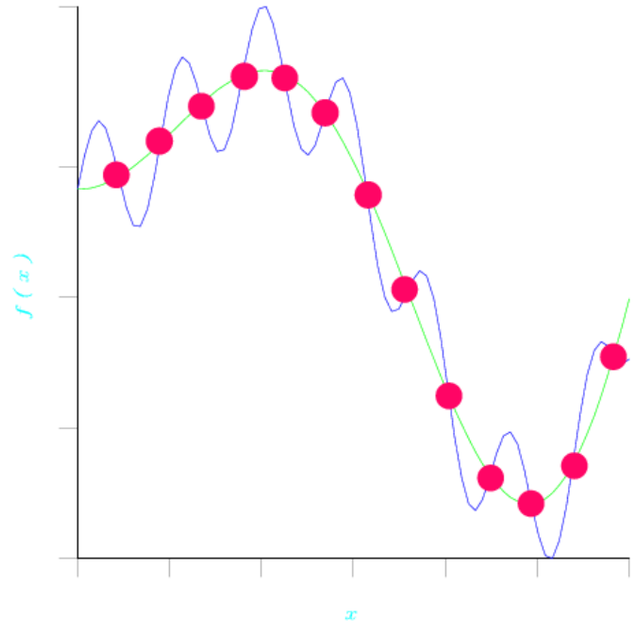
Jacques Hadamard

Ill posed problems often arise if one tries to infer general laws from few data

- the hypothesis space is too large
- there are not enough data

In general ERM leads to ill-posed solutions because

- the solution may be too complex
- it may be not unique
- it may change radically when leaving one sample out



Regularization theory provides results and techniques to restore well-posedness, that is stability (and generalization)

ERM finds the function in $(\mathcal{H})$ which minimizes

$$\frac{1}{n}\sum_{i=1}^{n} V(f(x_i), y_i)$$

which in general – for arbitrary hypothesis space $\mathcal{H}$ – is *ill-posed*.

- Ivanov regularizes by finding the function that minimizes

$$\frac{1}{n}\sum_{i=1}^{n} V(f(x_i), y_i)$$

while satisfying $\mathcal{R}(f) \leq A$.

- Tikhonov regularization minimizes over the hypothesis space $\mathcal{H}$, for a fixed positive parameter γ, the regularized functional

$$\frac{1}{n}\sum_{i=1}^{n} V(f(x_i), y_i) + \gamma\mathcal{R}(f). \tag{2}$$

$\mathcal{R}(f)$ is the regulirizer, a penalization on f. In this course we will mainly discuss the case $\mathcal{R}(f) = \|f\|_K^2$ where $\|f\|_K^2$ is the norm in the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$, defined by the kernel K.

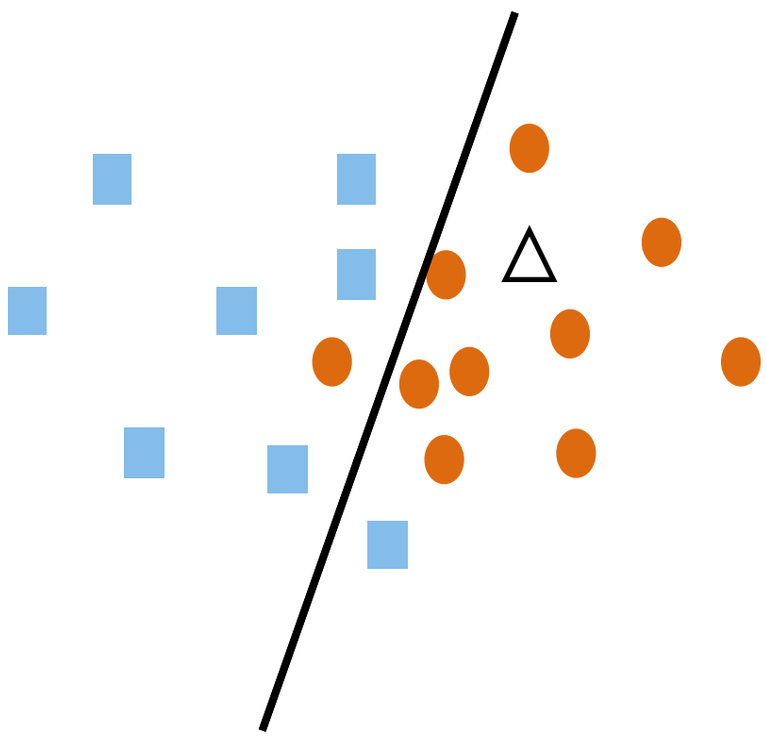
Of all the principles which can be proposed for that purpose, I think there is none more general, more exact, and more easy of application, that of which we made use in the preceding researches, and which consists of rendering the ***sum of squares of the errors*** *a minimum.*

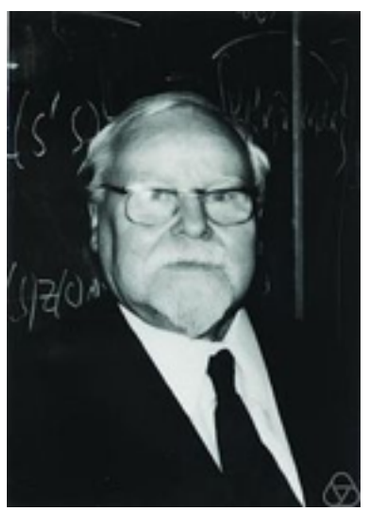
(Legendre 1805)

We consider the following algorithm

$$\min_{w \in \mathbb{R}^D} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^T x_i))^2 + \lambda w^T w, \quad \lambda \geq 0.$$

$$f(x) = w^T x = 0$$





?

?

Tikhonov ‘62 Phillips ‘62 Hoerl et al. ‘62

$$\min_{w \in \mathbb{R}^D} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^T x_i))^2 + \lambda w^T w, \quad \lambda \geq 0.$$

Computations?

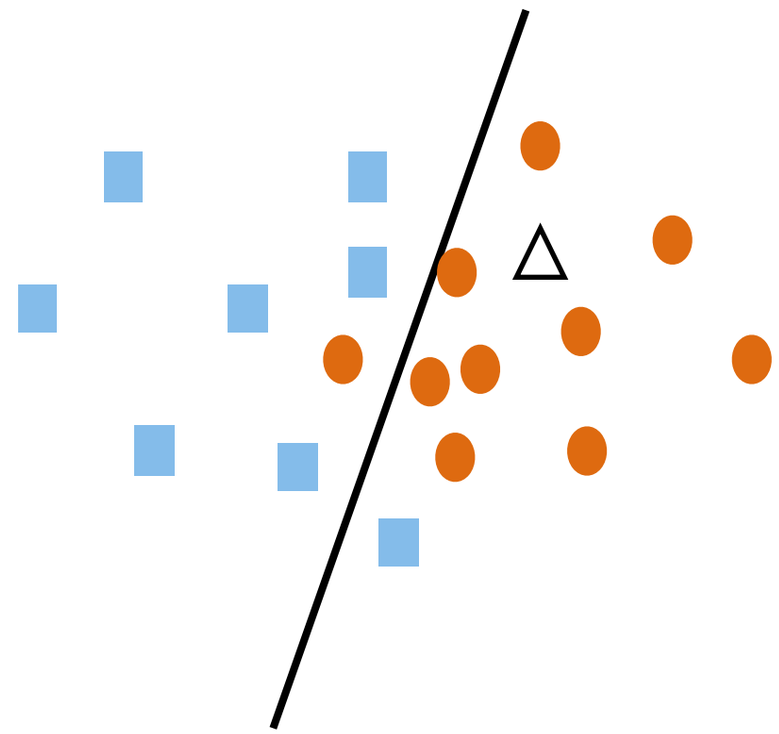
Notation $\frac{1}{n} \sum_{i=1}^{n} (y_i - w^T x_i))^2 = \frac{1}{n} \|Y_n - X_n w\|^2$

$-\frac{2}{n} X_n^T (Y_n - X_n w)$, and, $2w$ **Setting gradients...**

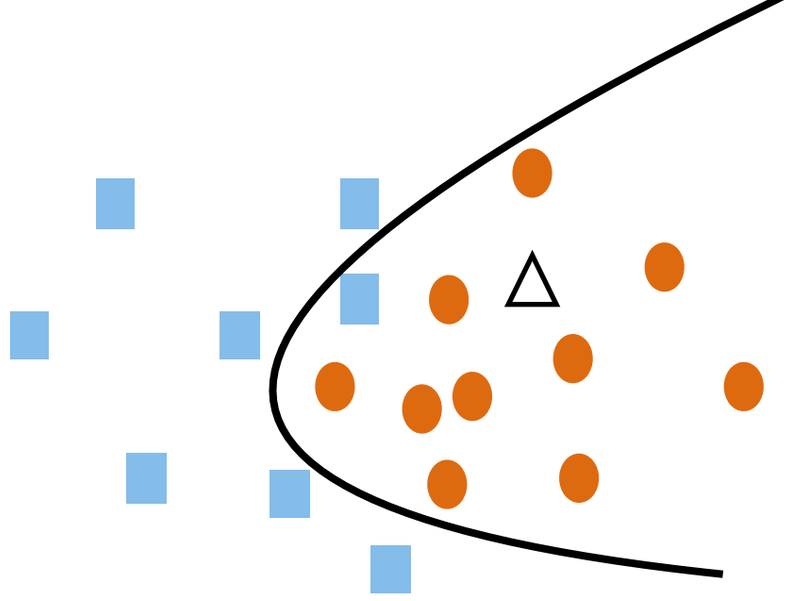
...to zero $(X_n^T X_n + \lambda n I) w = X_n^T Y_n$

$$(X_n^T X_n + \lambda n I)w = X_n^T Y_n$$

$$(X_n^T X_n + \lambda n I)w = X_n^T Y_n$$



Why a linear decision rule?



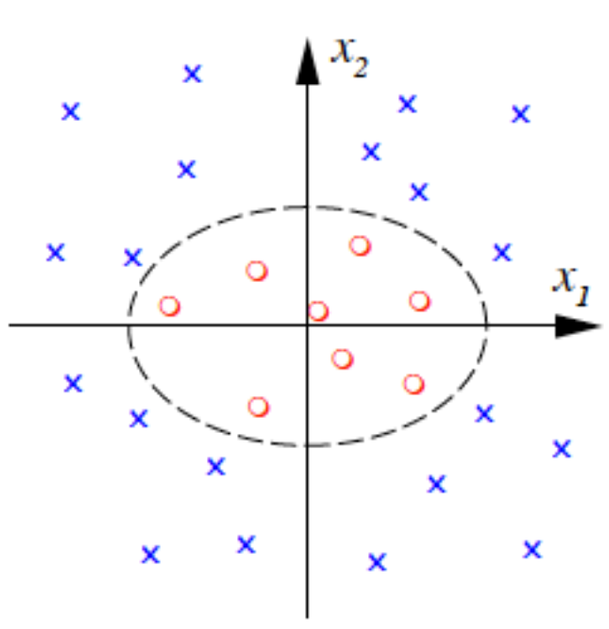
Dictionaries

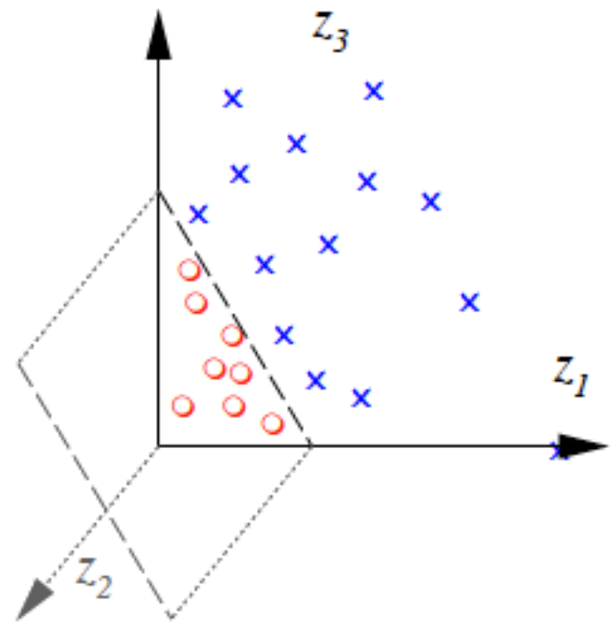
$$x \mapsto \tilde{x} = (\phi_1(x), \ldots, \phi_p(x)) \in \mathbb{R}^p$$

$$f(x) = w^T \tilde{x} = \sum_{j=1}^{p} \phi_j(x) w^j$$

$$\Phi : R^2 \to R^3$$

$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2} x_1 x_2, x_2^2)$$





Supervised learning
applications

Learning from Examples:

Computer Vision

- Face detection
- Pedestrian detection
- Scene understanding
- Video categorization

Decoding the Neural Code

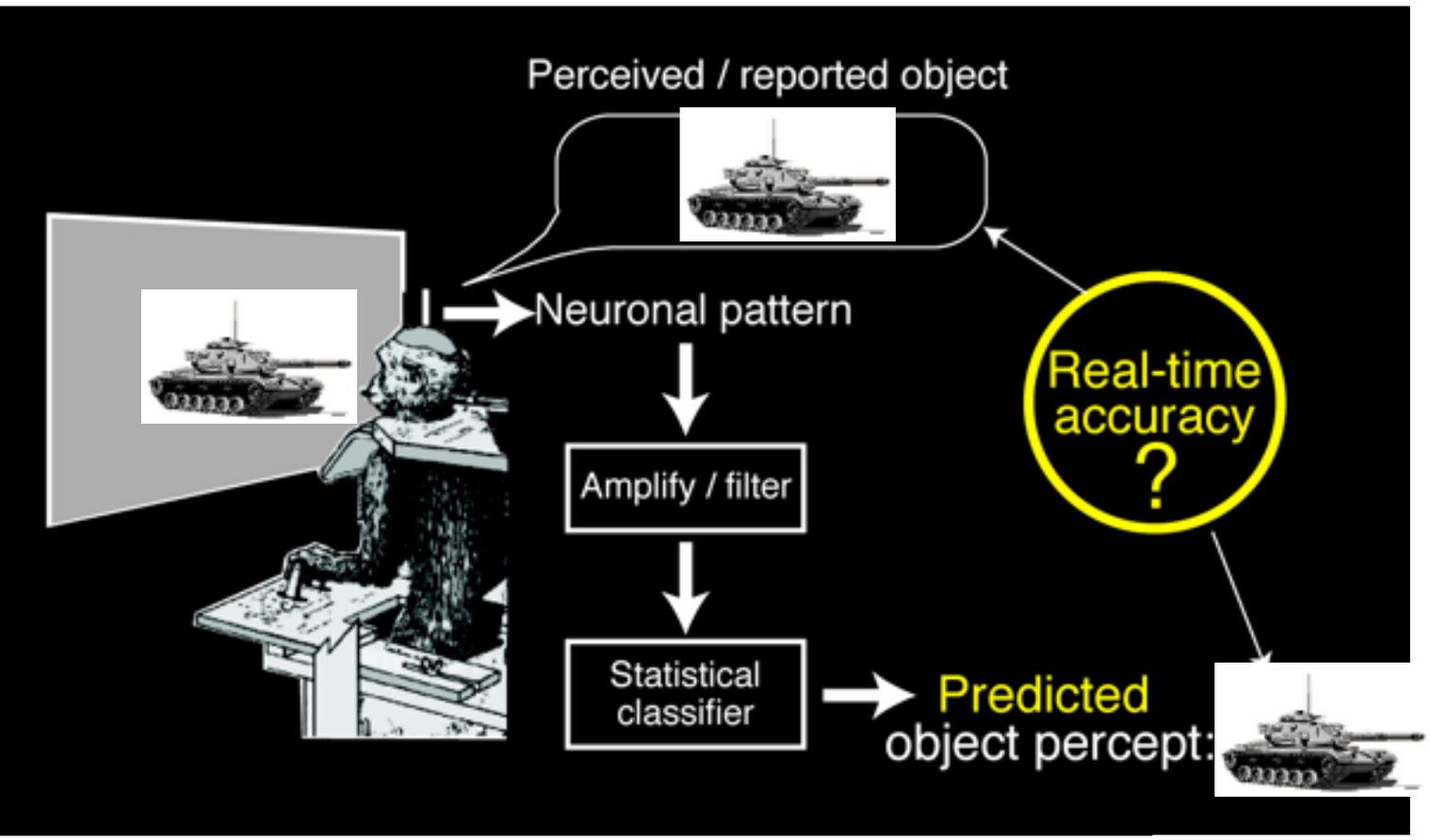
Bioinformatics

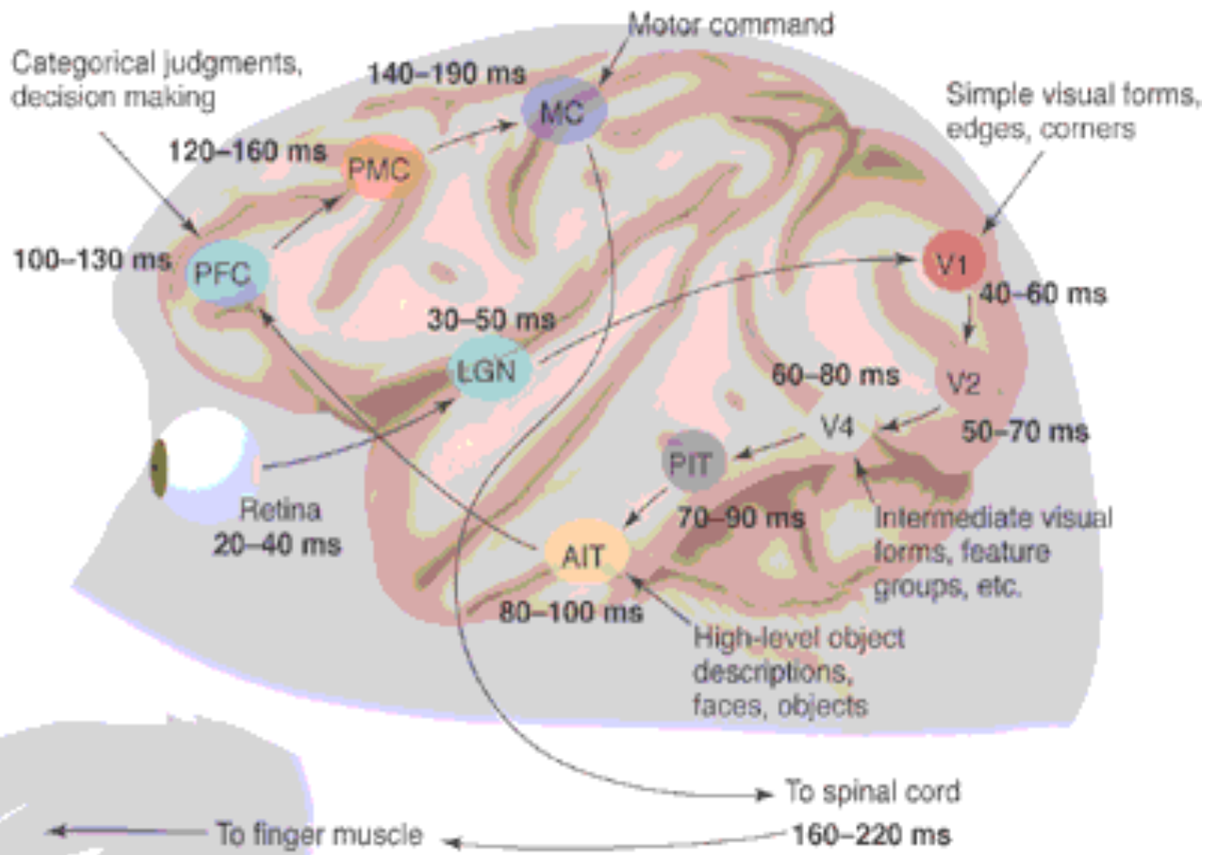
Graphics

Text Classification

Artificial Markets

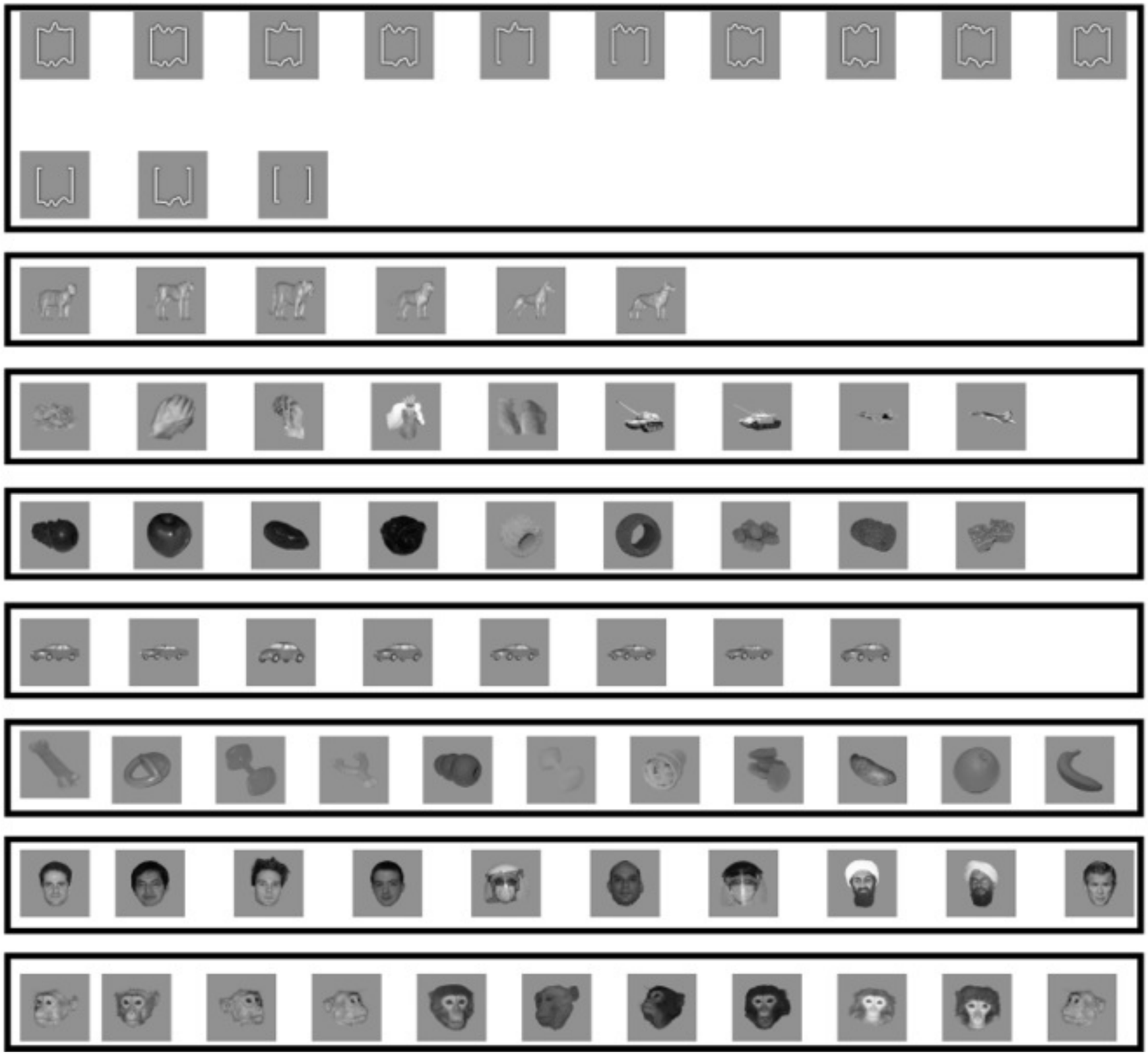
.....


Perceived / reported object
Neuronal pattern
Amplify / filter
Statistical classifier
Predicted object percept:
Real-time accuracy ?


Motor command
Categorical judgments, decision making
140–190 ms
MC
Simple visual forms, edges, corners
120–160 ms
PMC
100–130 ms
PFC
V1
40–60 ms
30–50 ms
LGN
60–80 ms
V2
V4
50–70 ms
PIT
Retina
20–40 ms
70–90 ms
Intermediate visual forms, feature groups, etc.
AIT
80–100 ms
High-level object descriptions, faces, objects
To spinal cord
To finger muscle
160–220 ms

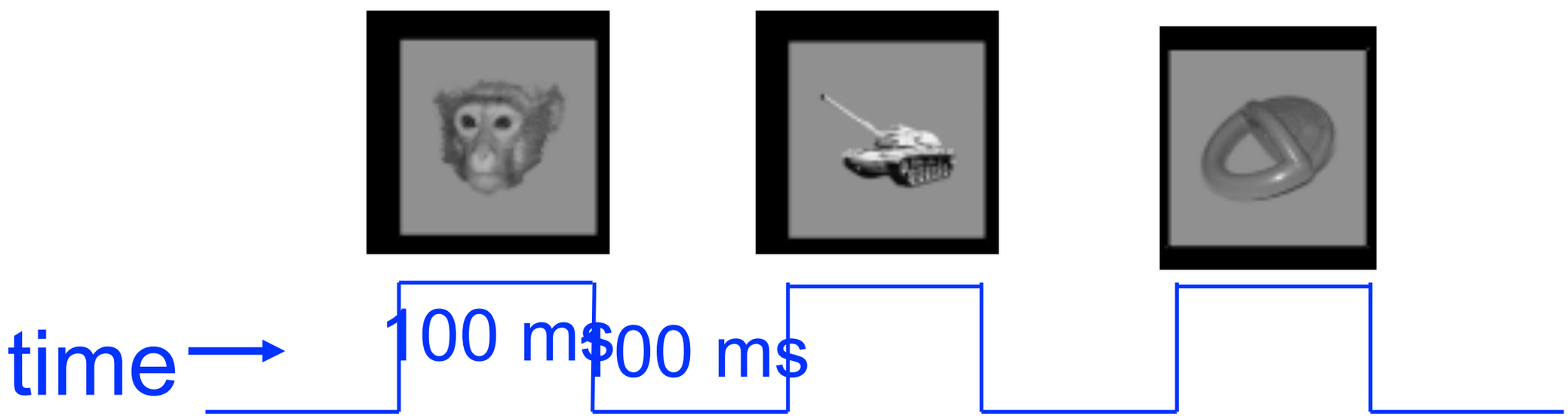
Reading-out the neural code in AIT

77 objects, 8 classes



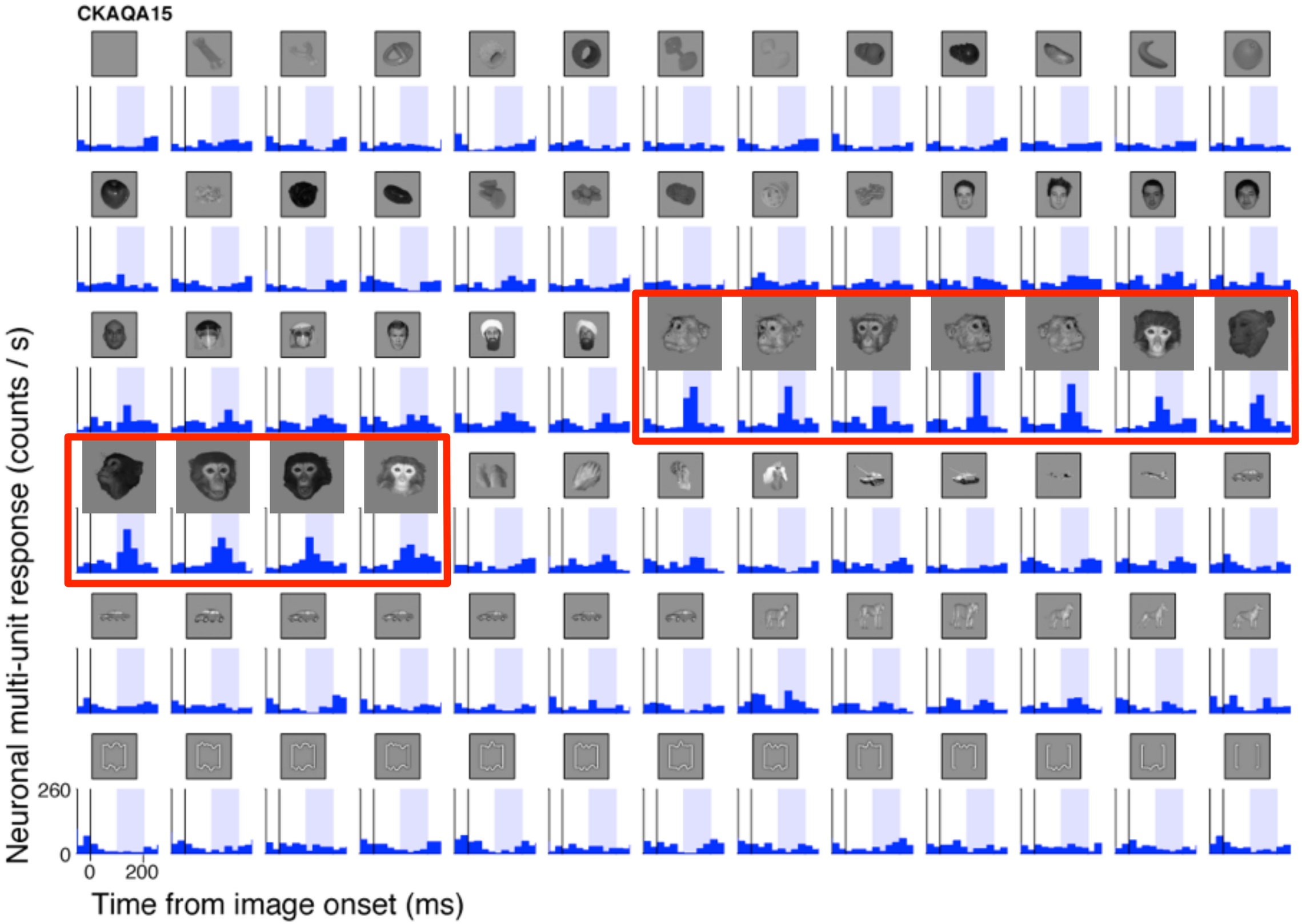
Chou Hung, Gabriel Kreiman, James DiCarlo, Tomaso Poggio, *Science, Nov 4, 2005*

Recording at each recording site during passive viewing

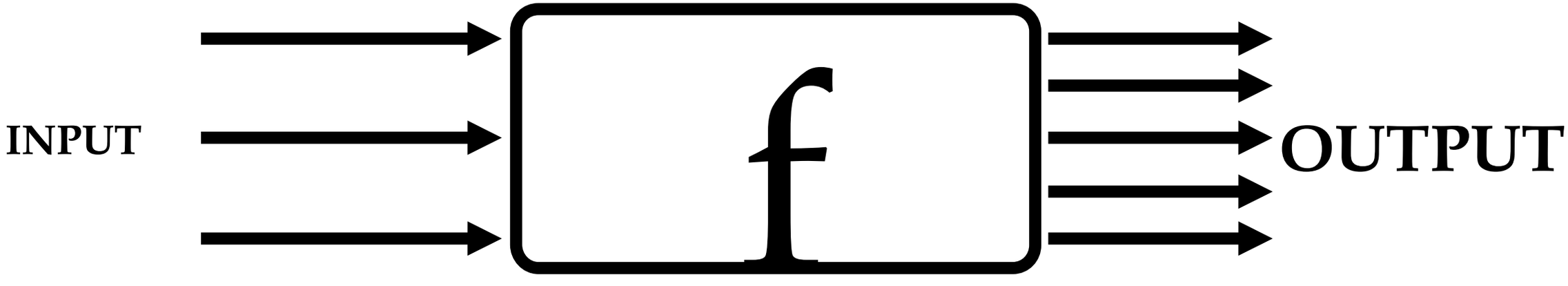


- 77 visual objects
- 10 presentation repetitions per object
- presentation order randomized and counter-balanced

Example of one AIT cell



Training a classifier on neuronal activity.



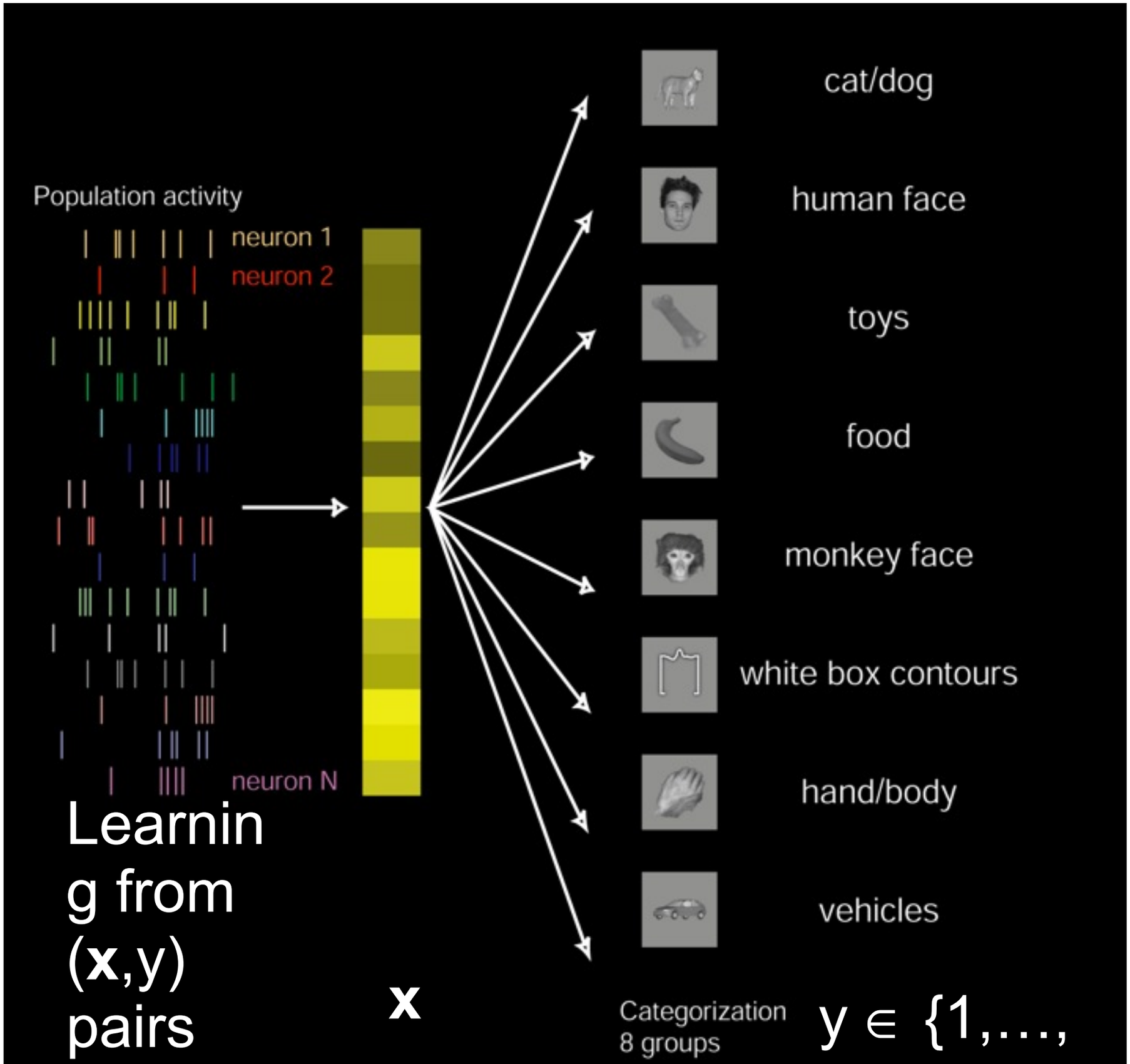
From a set of data (vectors of activity of n neurons (x) and object label (y)

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_\ell, y_\ell)\}$$

$$f(x) = \hat{y}$$

Find (by training) a classifier eg a function f such that

Decoding the neural code … using a classifier

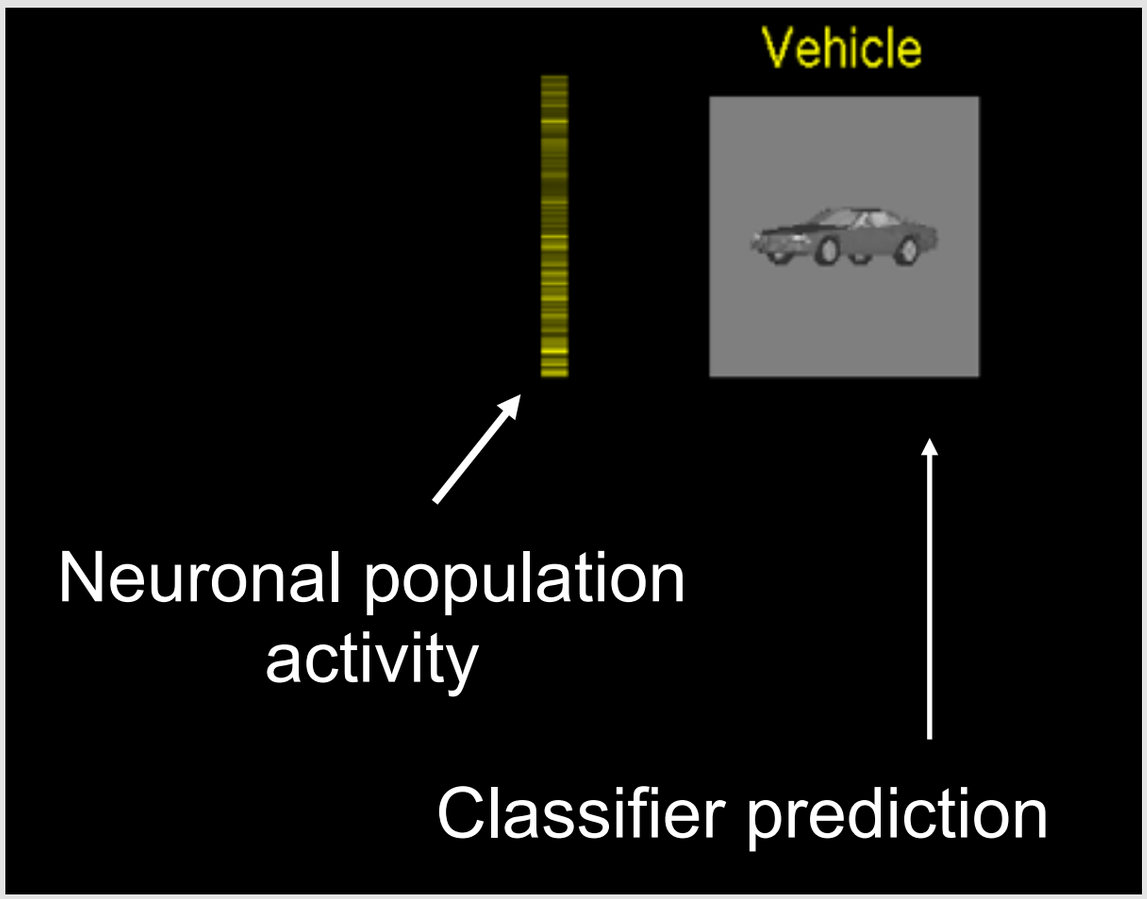


We can decode the brain's code and read-out from neuronal populations:

reliable object categorization (>90% correct) using ~200 <u>arbitrary</u> AIT "neurons"

Video speed: 1 frame/sec

Actual presentation rate: 5 objects/sec



<u>Categorization</u>

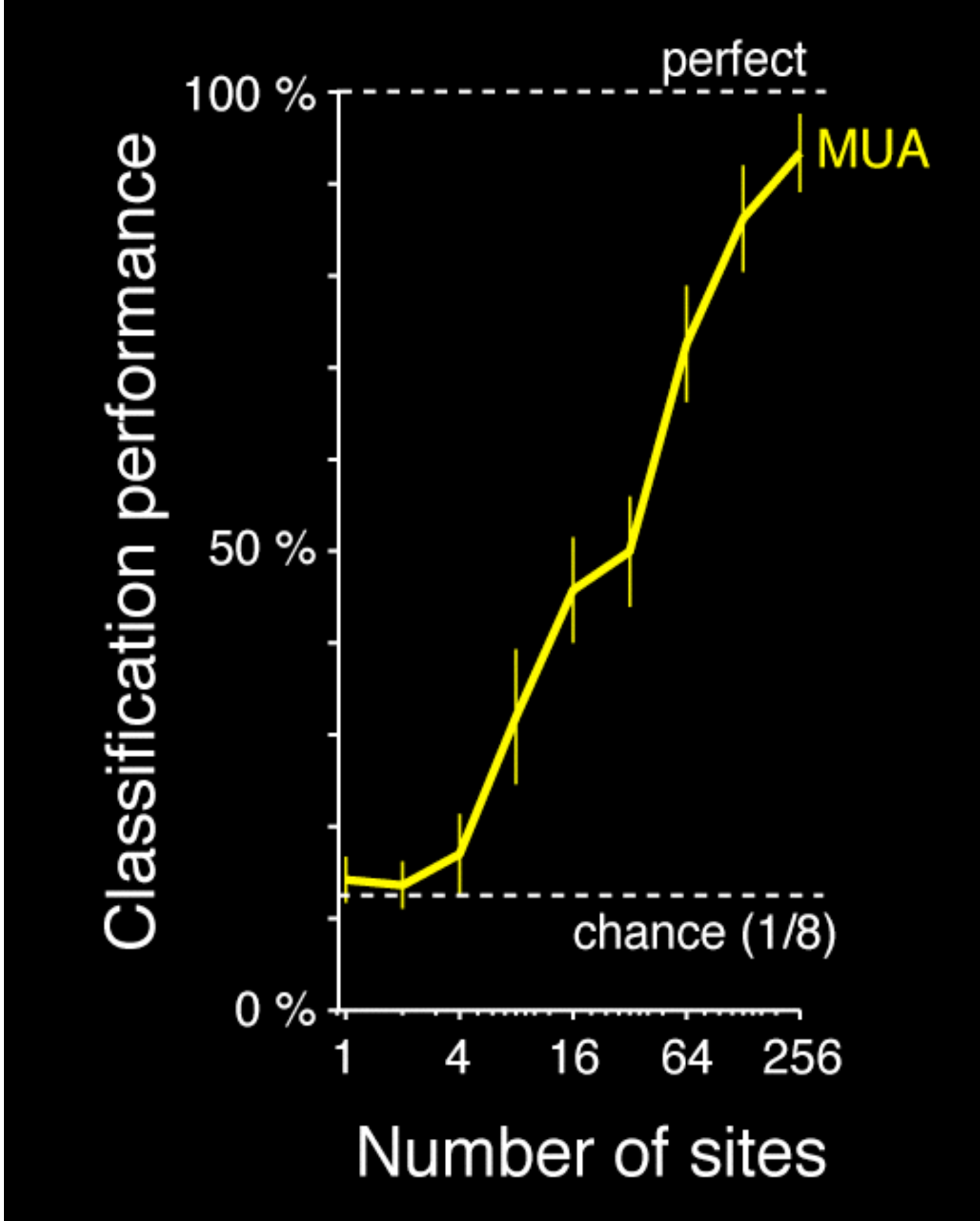
- Toy
- Body
- Human Face
- Monkey Face
- Vehicle
- Food
- Box
- Cat/Dog

Hung, Kreiman, Poggio, DiCarlo. *Science 2005*

We can decode the brain's code and read-out from neuronal populations:

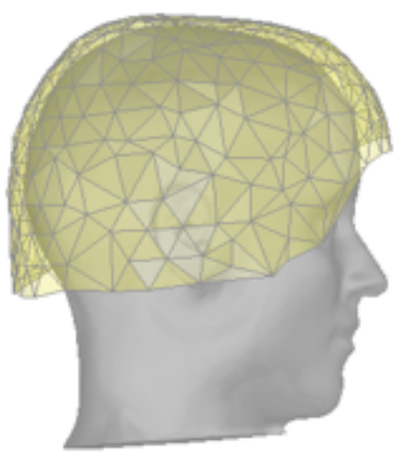
reliable object categorization using ~100 arbitrary AIT sites

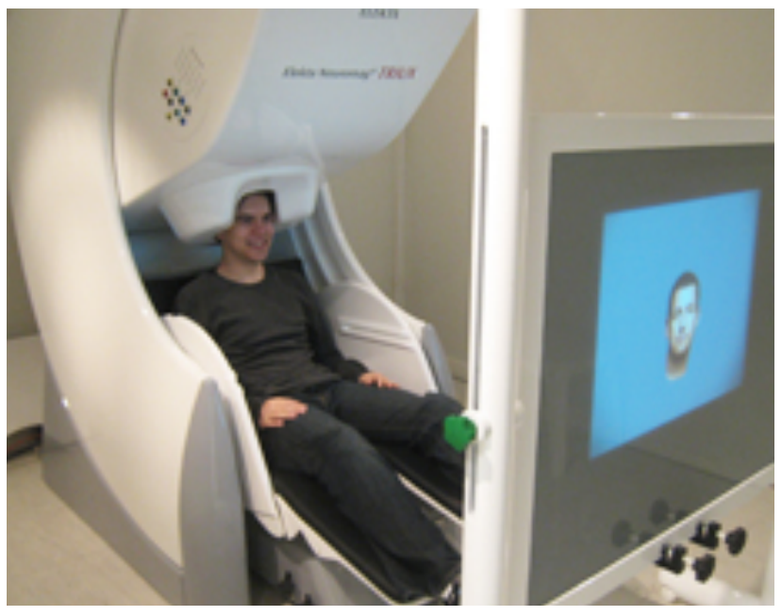
- *[100-300 ms] interval*
- *50 ms bin size*



Magnetoencephalography

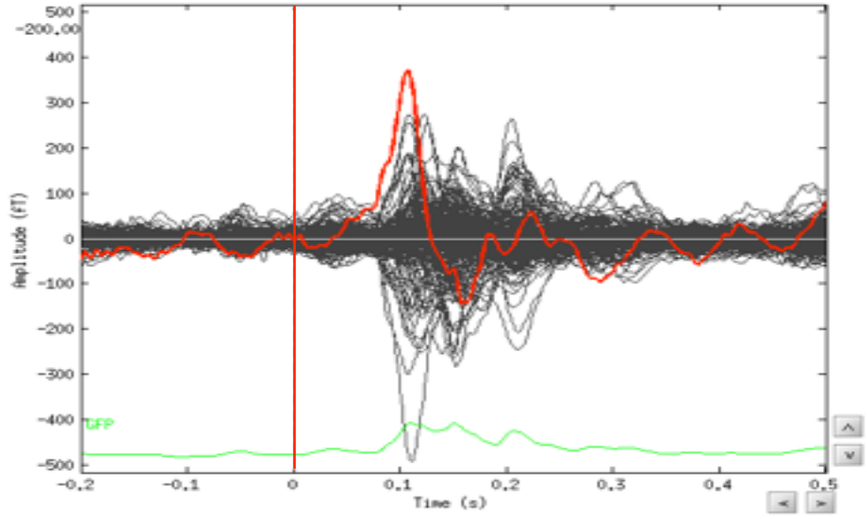
- Detects magnetic fields induced by synchronous neural currents
- Brain: 10^{-12} Tesla vs. Earth: 10^{-5} Tesla
- Shielded room and super-conducting quantum interference devices (SQUIDs)
- Millisecond temporal resolution

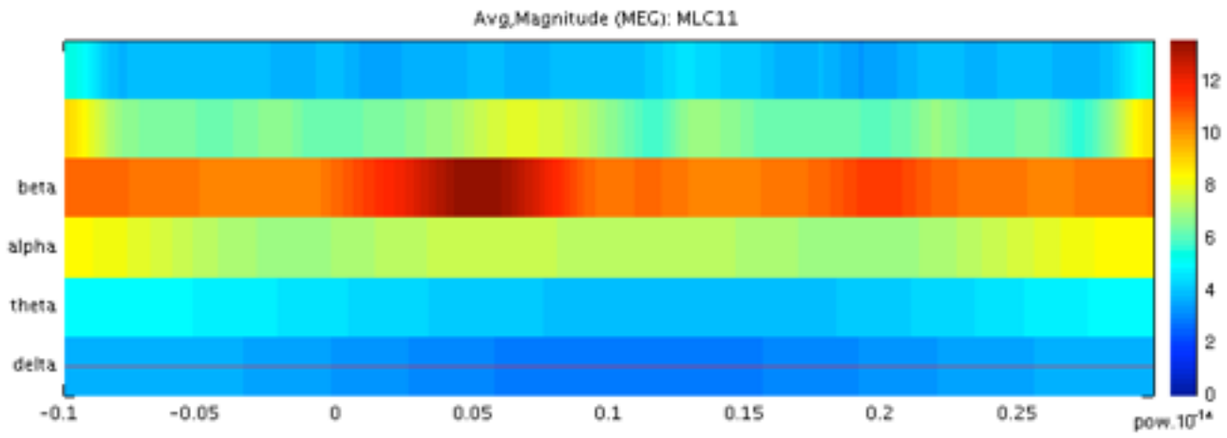


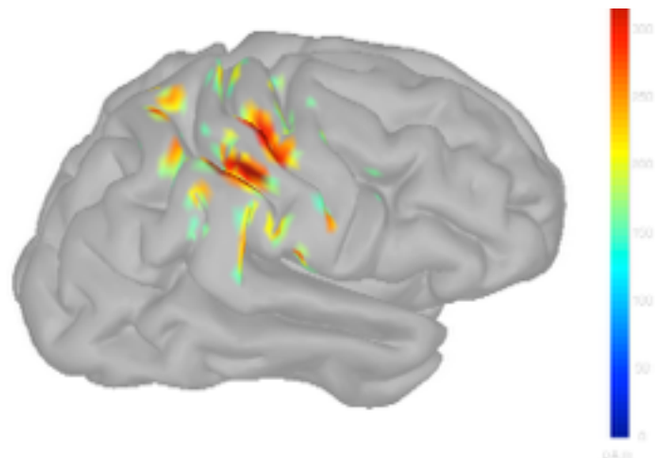


MEG data analysis

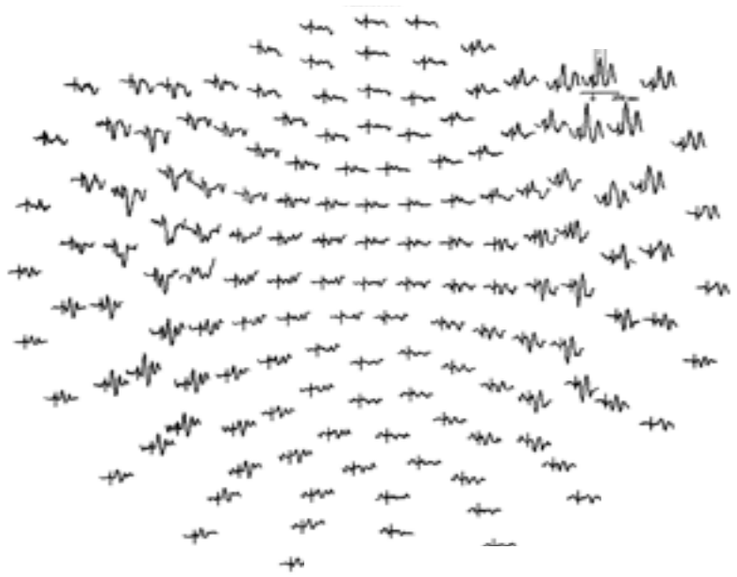
- Event-related response
  - Average 100+ trials
- Source localization
  - Inverse problem, ill-posed
  - Constraint with MRI/fMRI
- Time-frequency analysis

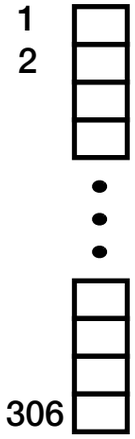






MEG decoding

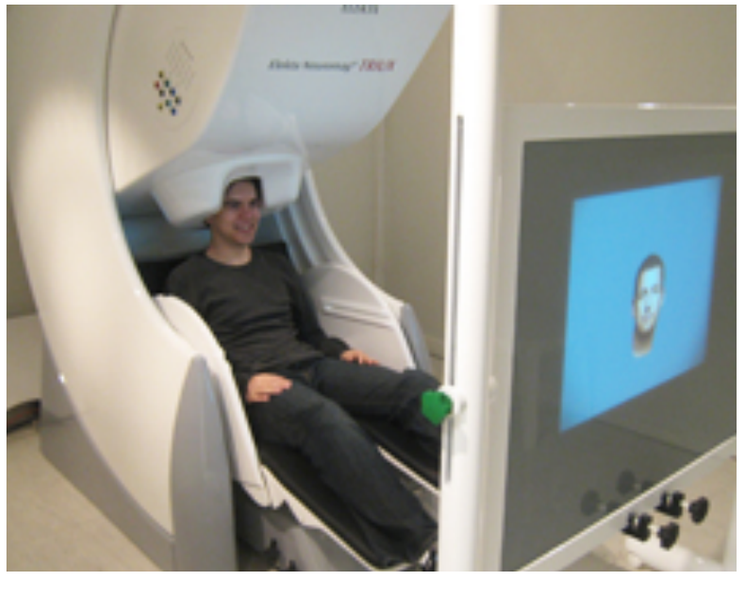




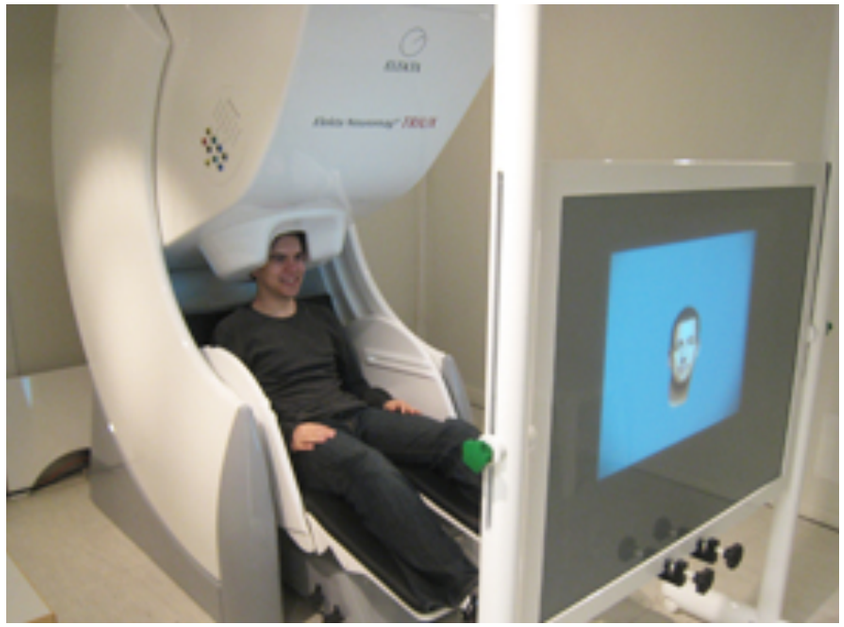
Pattern Classifier

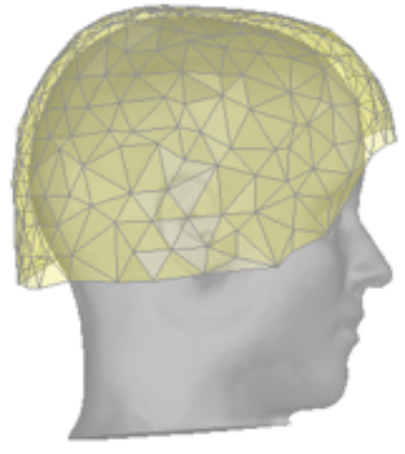
Predicted label

“face”



MEG decoding analysis

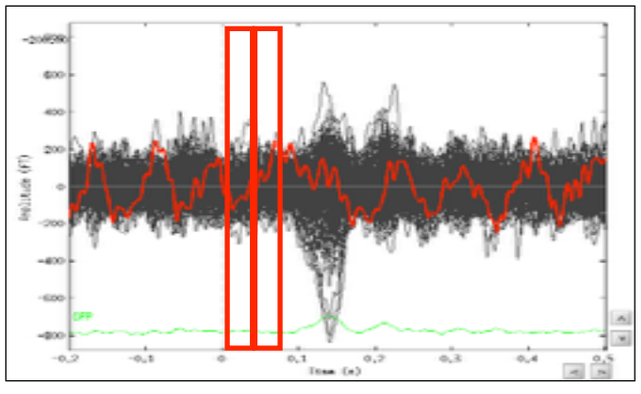




t = 0 ms

t = 50 ms

...



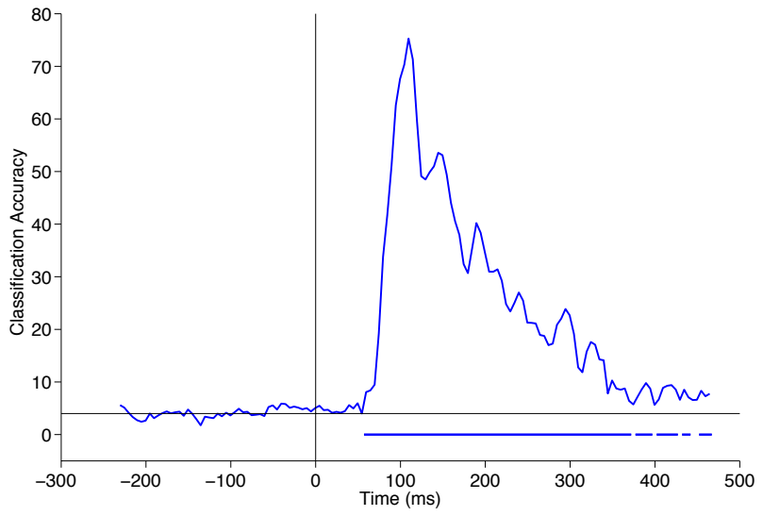
1
2

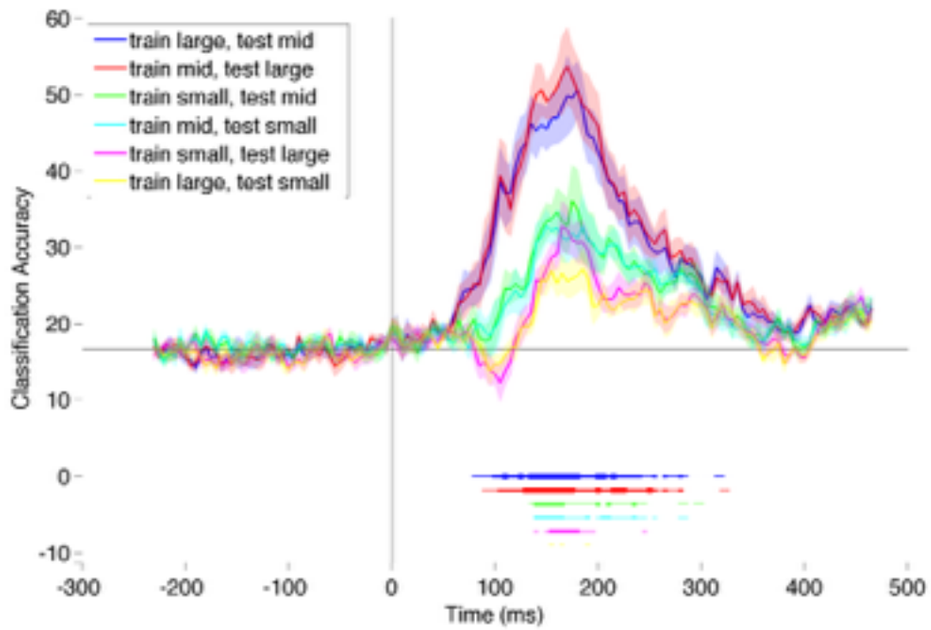
306

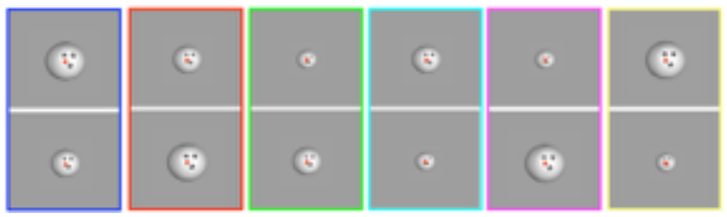
Pattern classifier

Predicted label

- MEG decoding





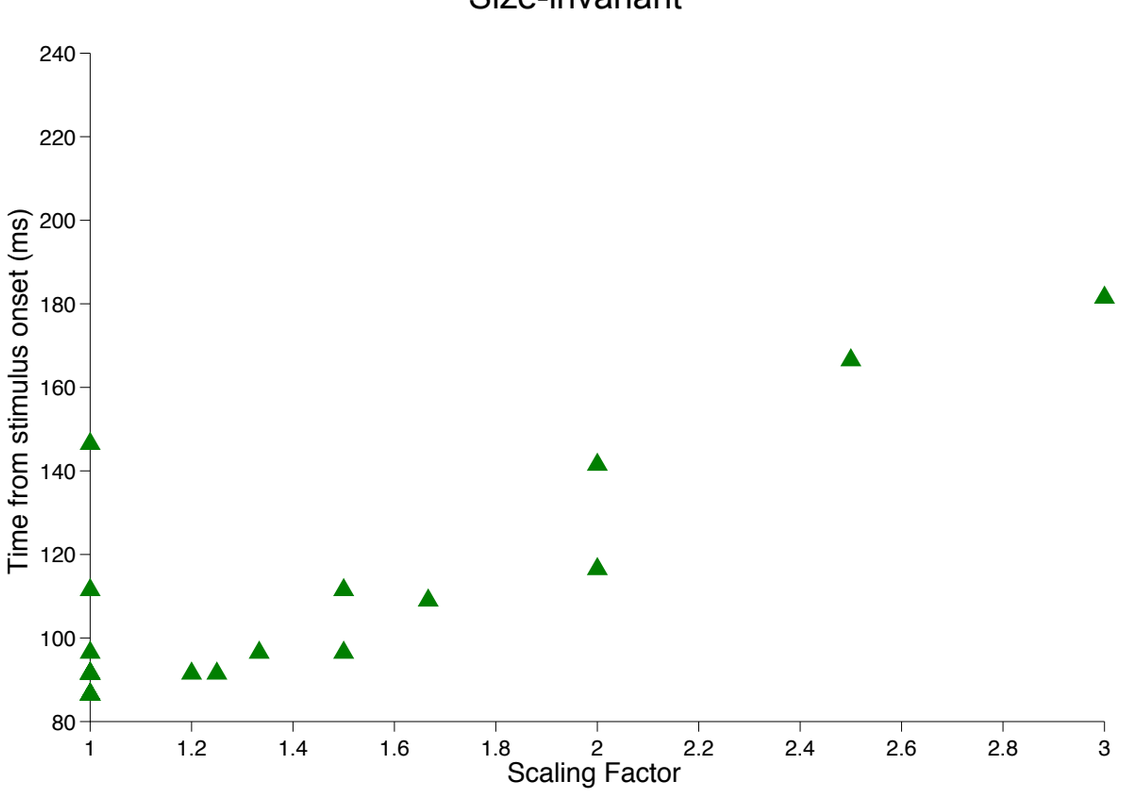


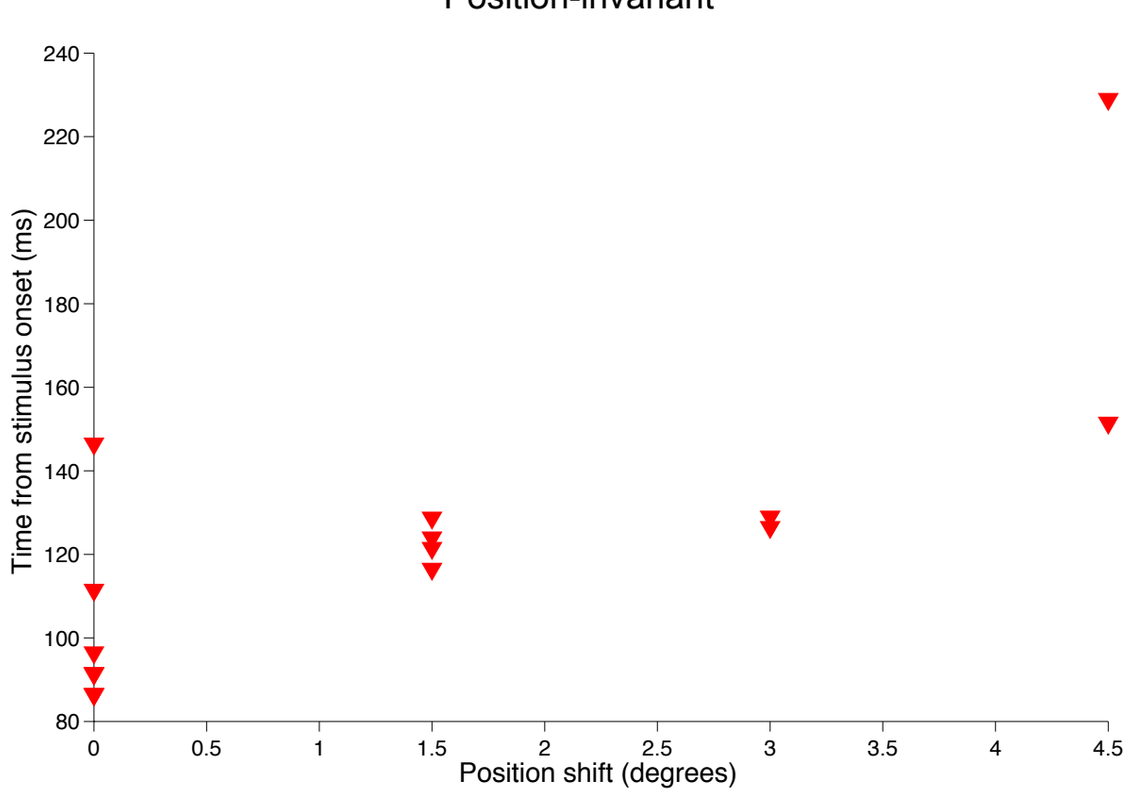
- Timing of size- and position-invariance
- Increase number of conditions in experiments (decrease number of repetitions)
  - Post-processing methods
  - Combining data from across subject/recording sessions

Isik,2013

Timing of size- and position-invariance

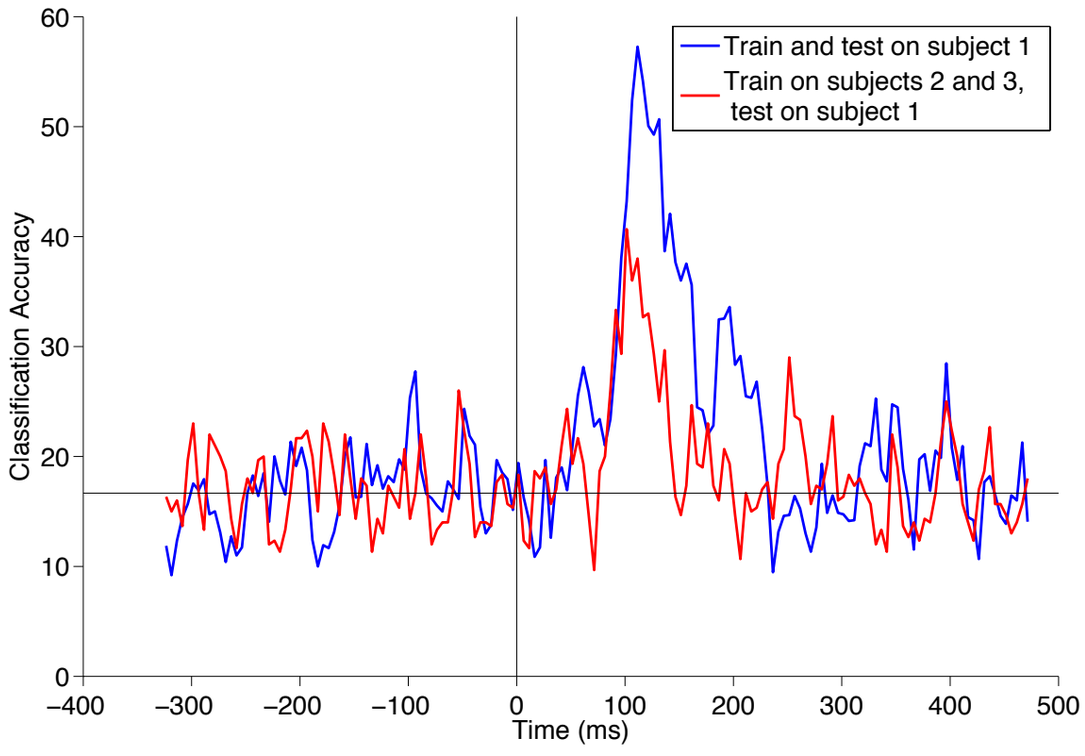
- When does decoding first rise significantly above chance?
  - Size: 6x6, 5x5, 4x4, 3x3, 2x2; Pos: 0, +/- 1.5, +/- 3

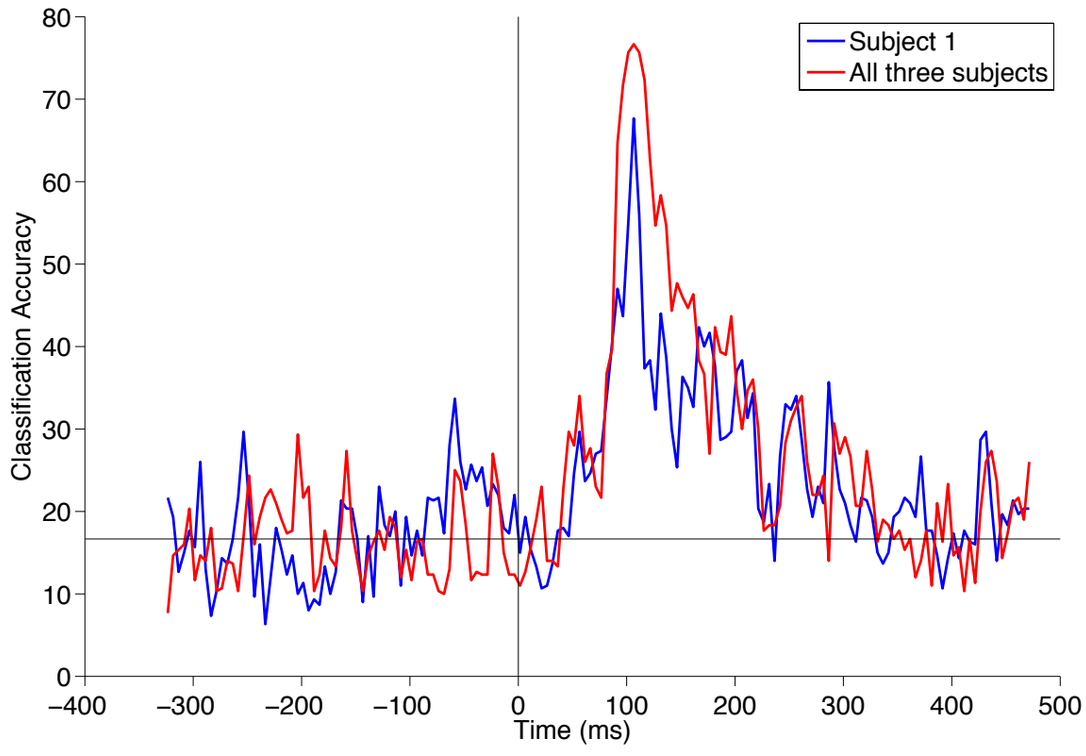




Decoding across recording sessions/subjects

- Transform sensors to common coordinate space
- Decode across subjects



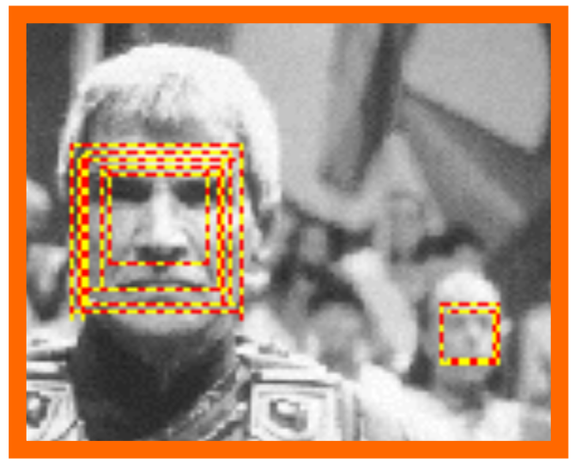


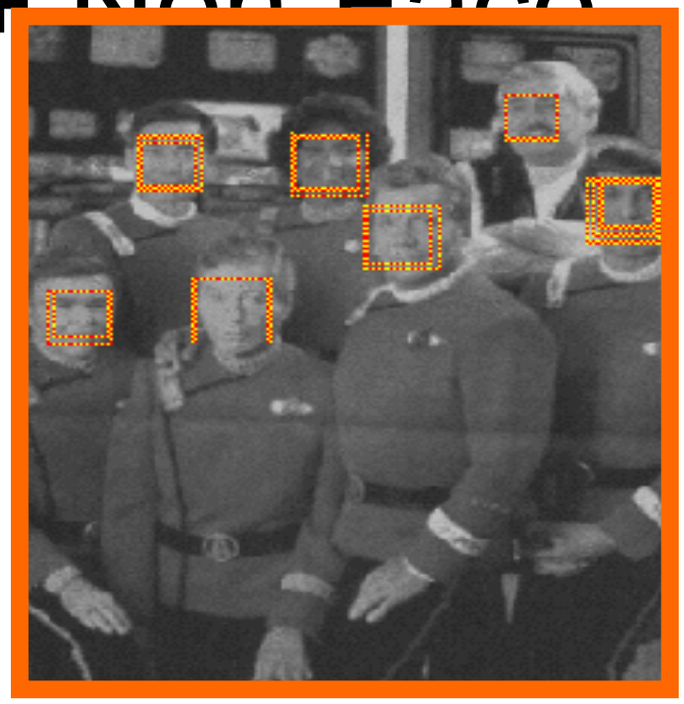
~15 year old CBCL computer vision research:

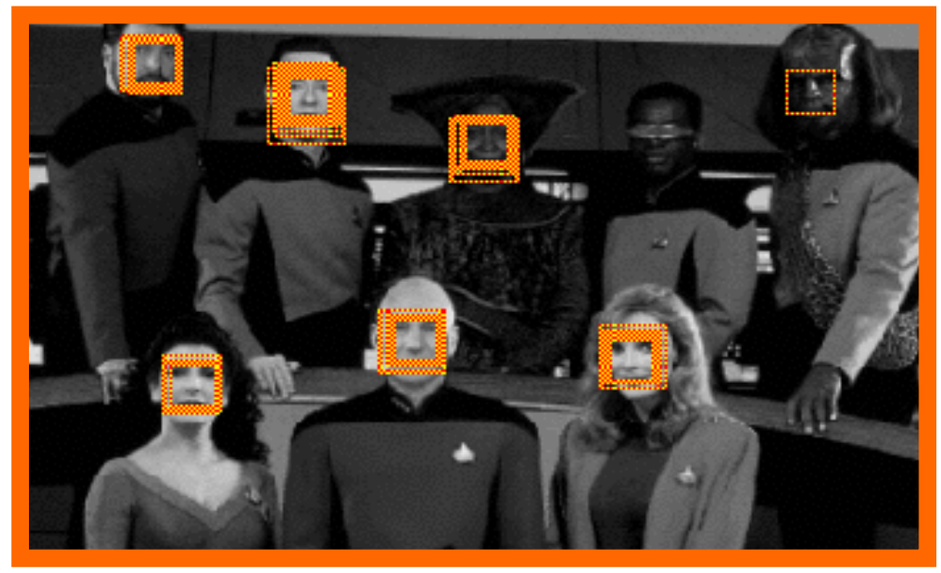
face detection;

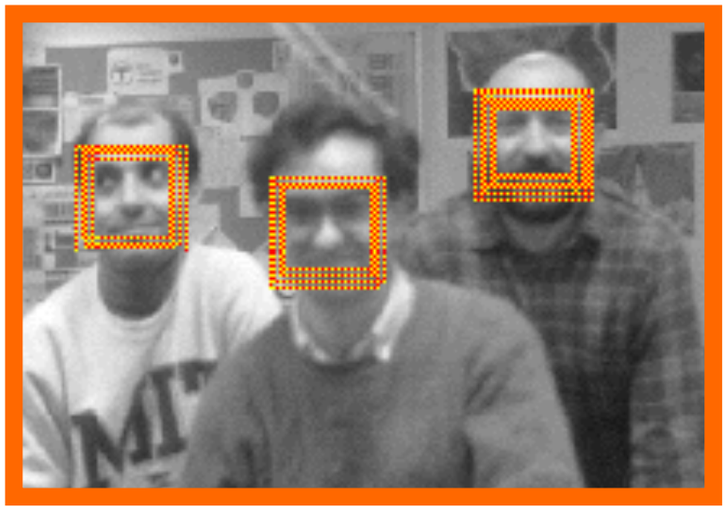
since 2006 on the market (digital cameras…)

- Training Database
- 1000+ Real, 3000+ VIRTUAL
- 50,0000+ Non-Face Pattern







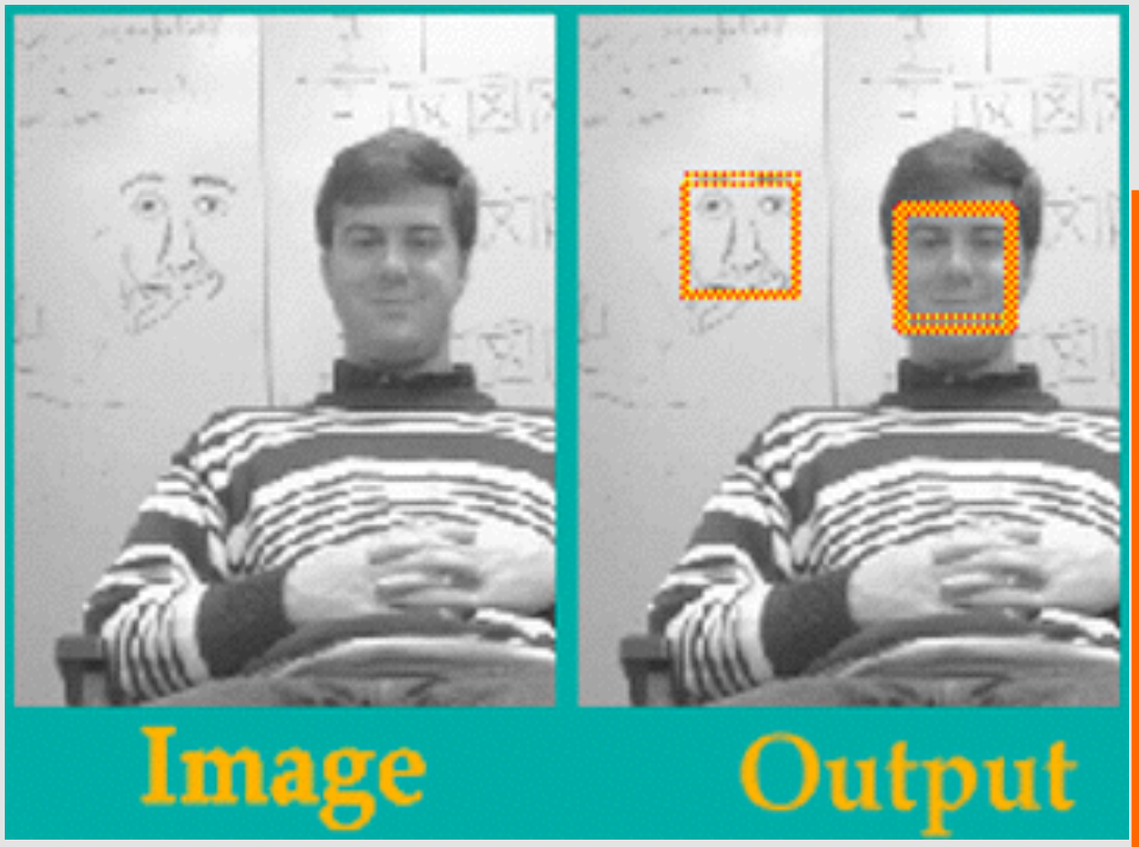


Sung & Poggio

~15 year old CBCL computer vision research:

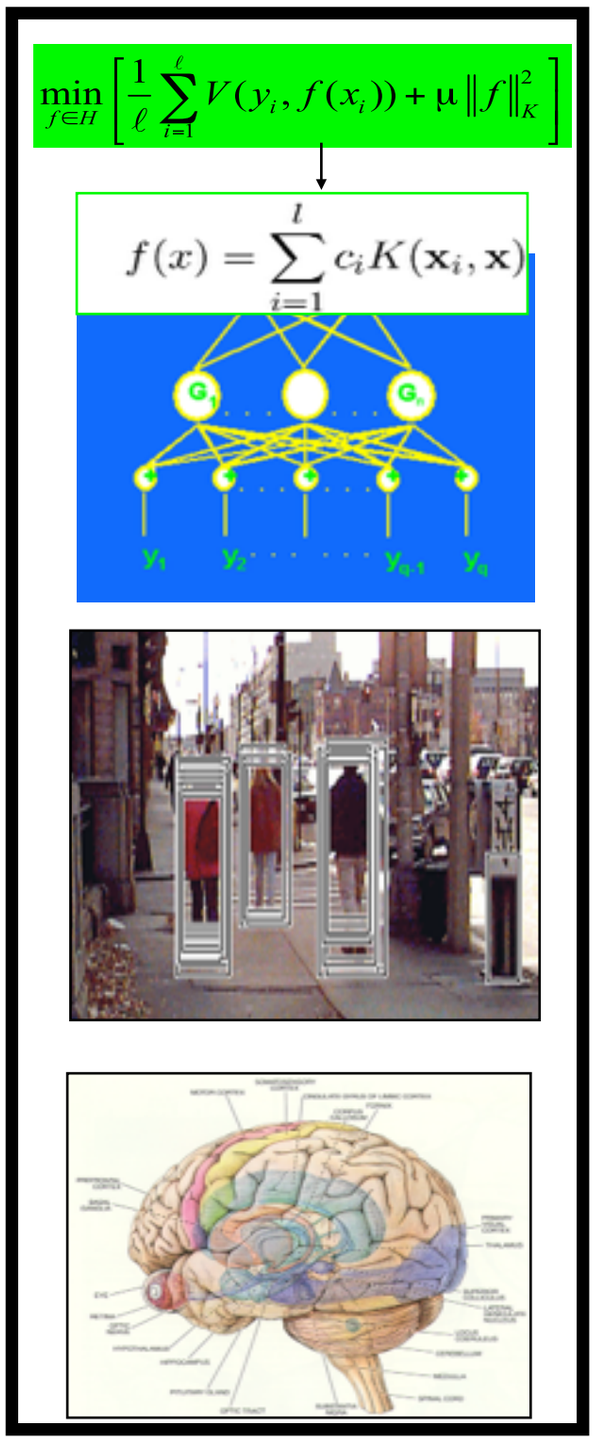
face detection;

since 2006 on the market (digital cameras...)



Sung & Poggio 1995

Learning



LEARNING THEORY + ALGORITHMS

Theorems on foundations of learning

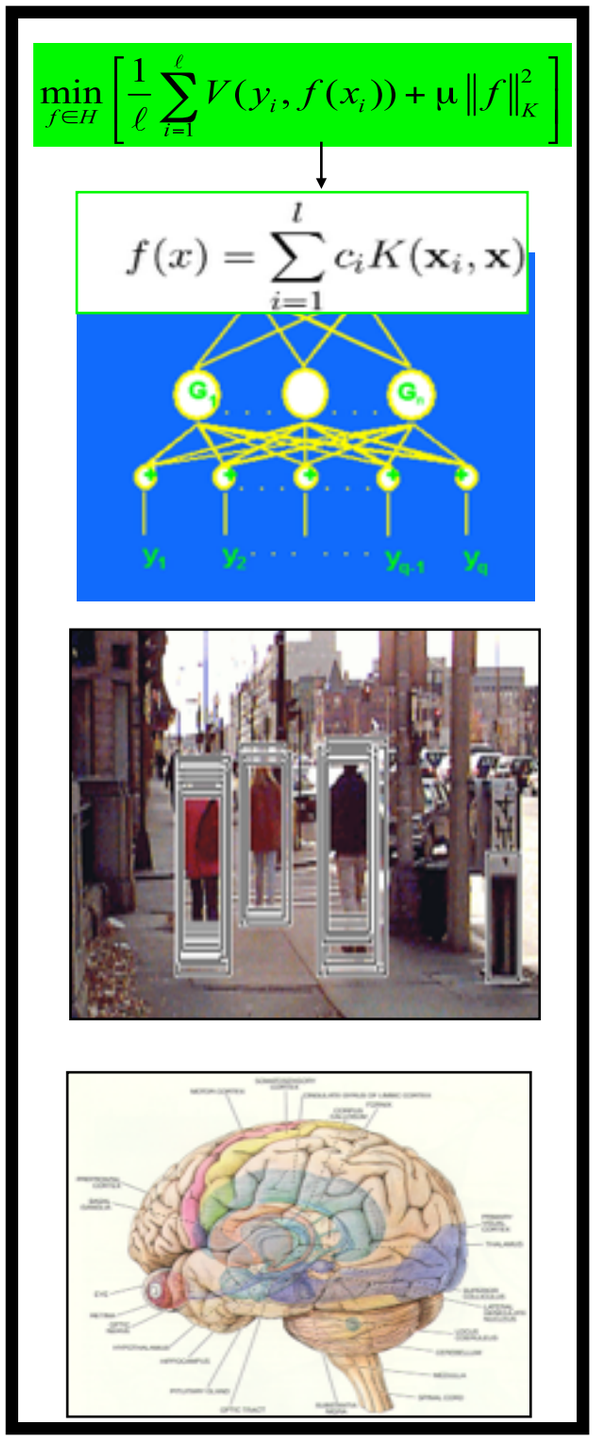
Predictive algorithms

Papageorgiou&Poggio, 1997, 2000
also Kanade&Scheiderman

COMPUTATIONAL NEUROSCIENCE: models+experiments

How visual cortex works

Learning



LEARNING THEORY + ALGORITHMS

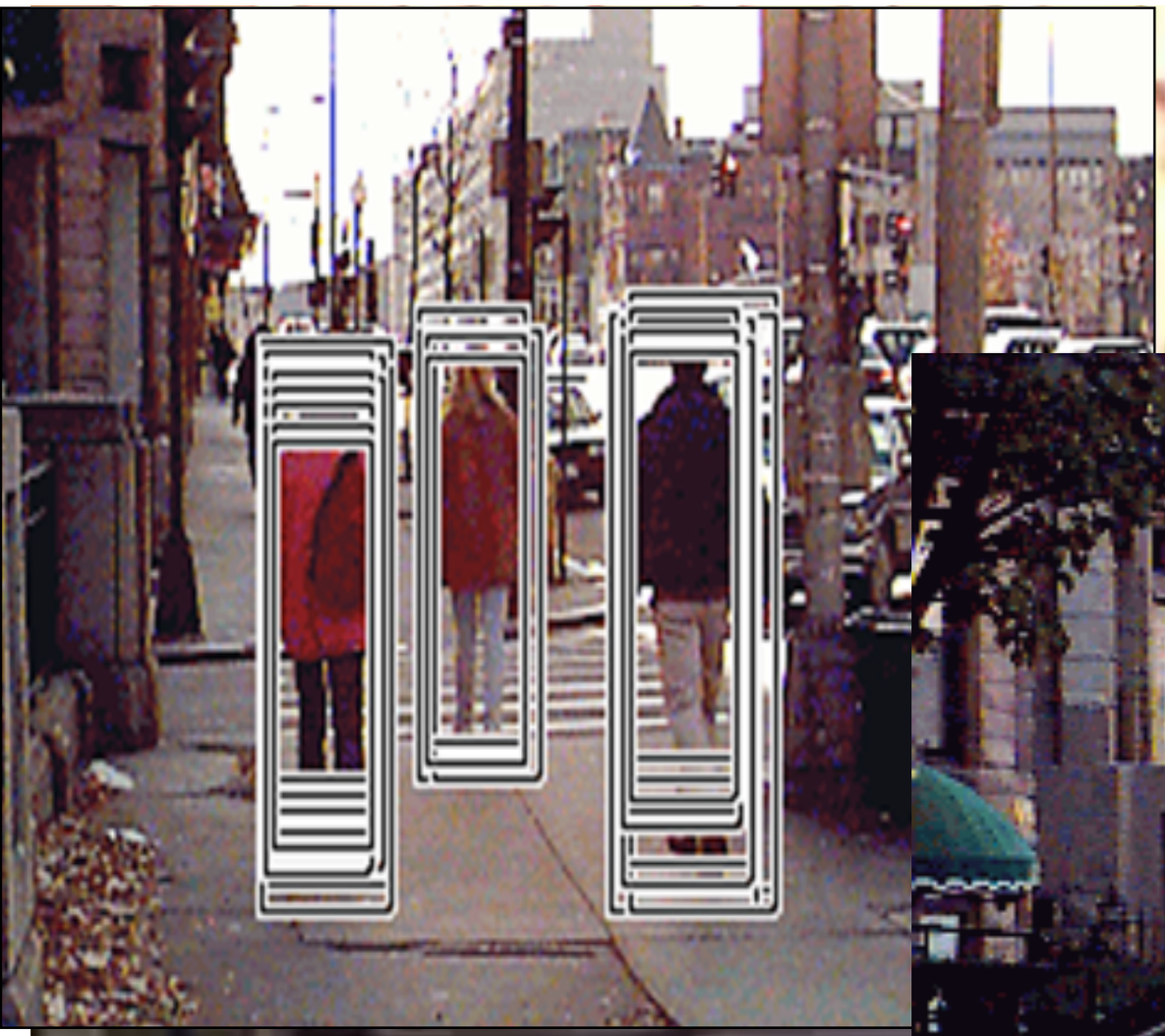
Theorems on foundations of learning

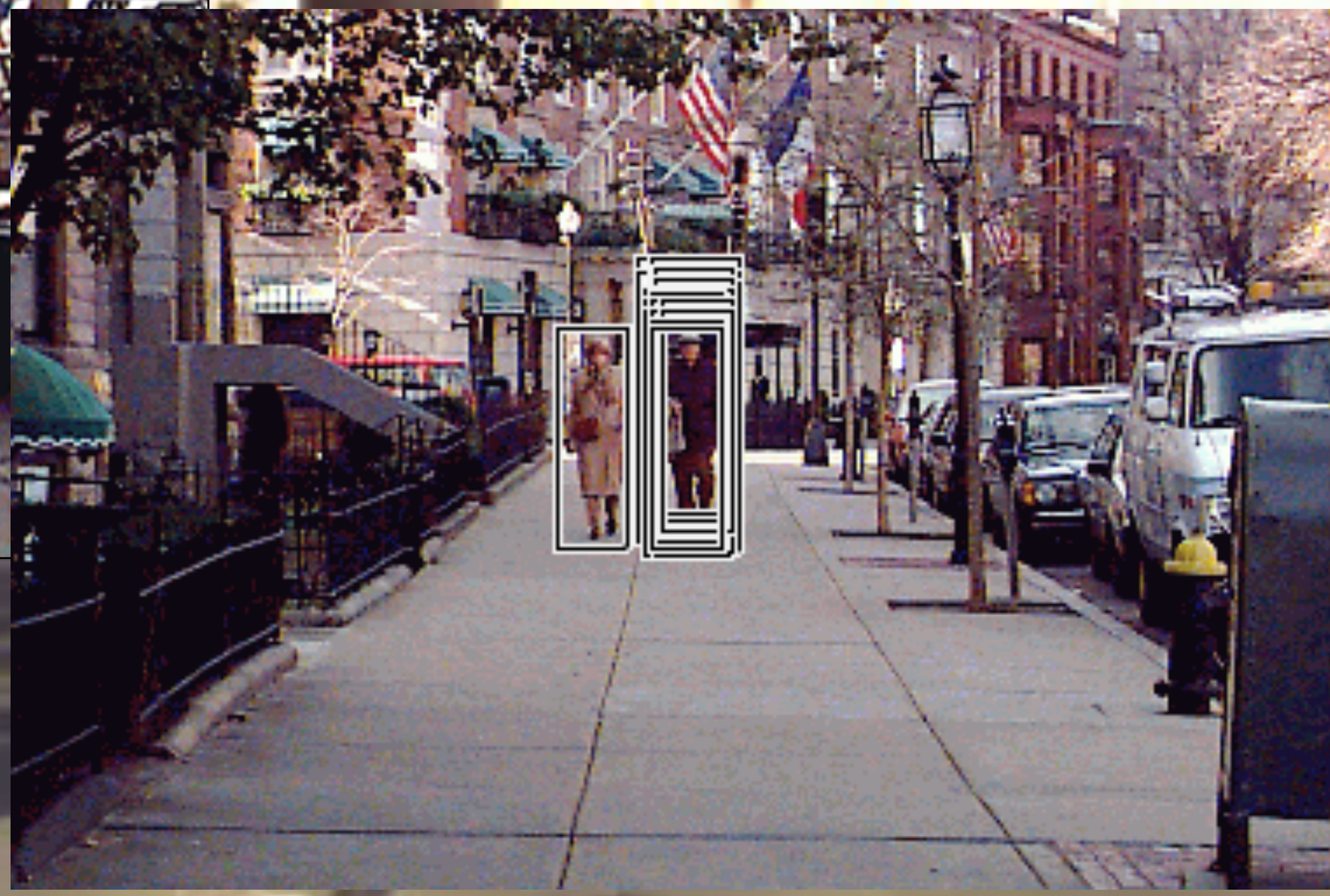
Predictive algorithms

Papageorgiou&Poggio, 1997, 2000
also Kanade&Scheiderman

COMPUTATIONAL NEUROSCIENCE: models+experiments

How visual cortex works



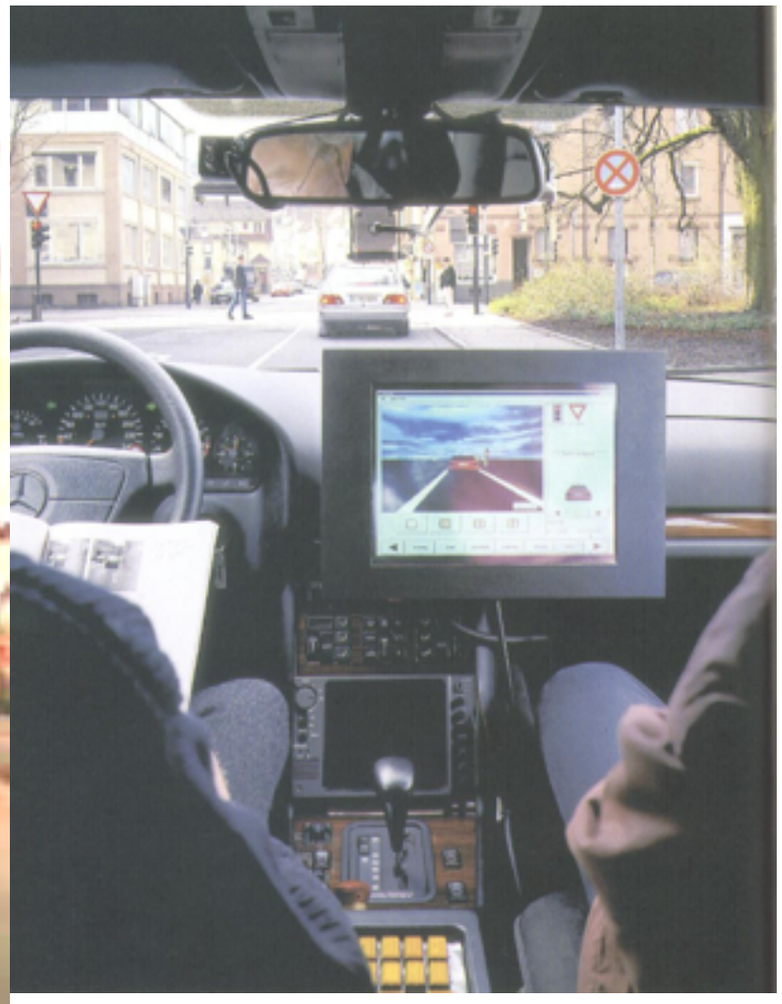


~10 year old CBCL computer vision research:

Pedestrian detection System

in Mercedes test car;

now (2010) there is a product (MobilEye)

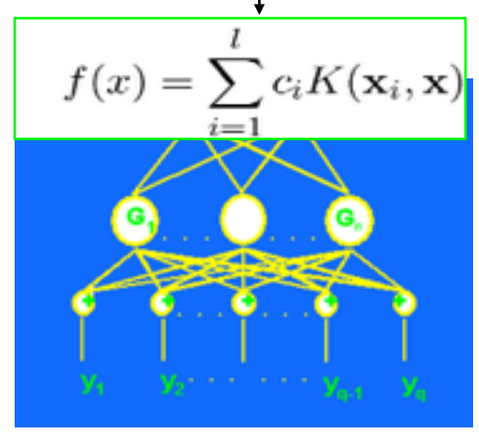


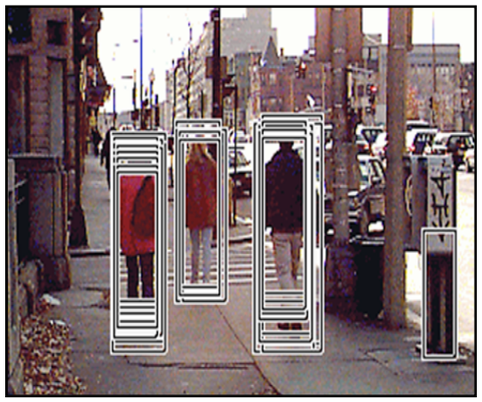
Papageorgiou, C. and T. Poggio, 1997, 2000

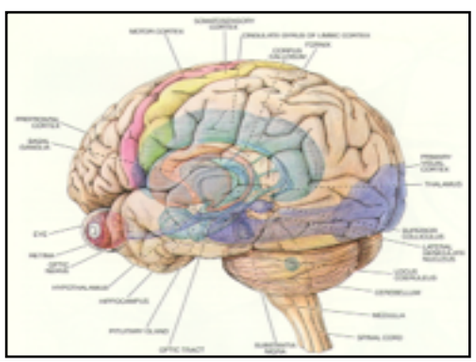
Learning

$$\min_{f \in H} \left[\frac{1}{\ell} \sum_{i=1}^{\ell} V(y_i, f(x_i)) + \mu \|f\|_K^2 \right]$$

$$f(x) = \sum_{i=1}^{l} c_i K(\mathbf{x}_i, \mathbf{x})$$







LEARNING THEORY + ALGORITHMS

Theorems on foundations of learning

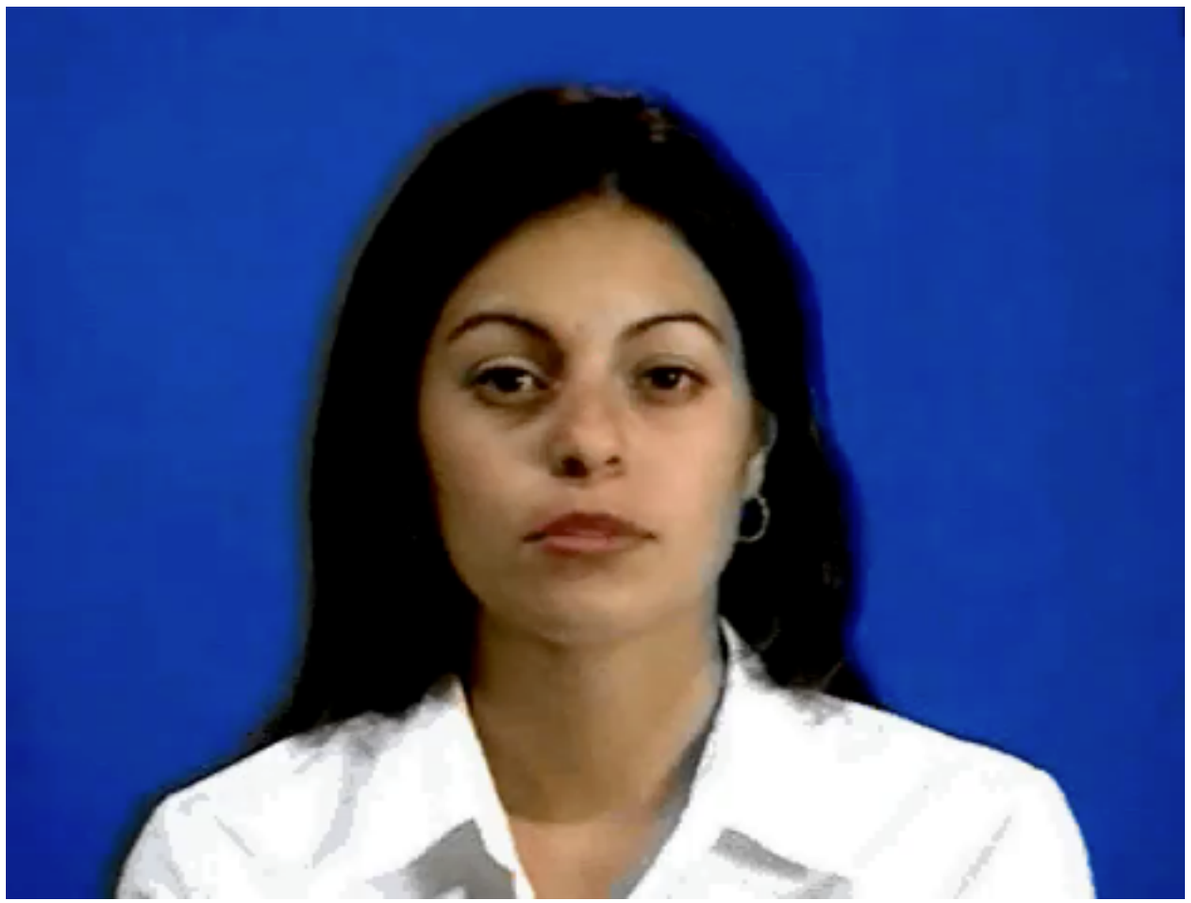
Predictive algorithms

Pedestrian and car detection are also “solved” (commercial systems, *MobilEye*)

COMPUTATIONAL NEUROSCIENCE: models+experiments

How visual cortex works

http://www.volvocars.com/us/all-cars/volvo-s60/pages/5-things.aspx?p=5



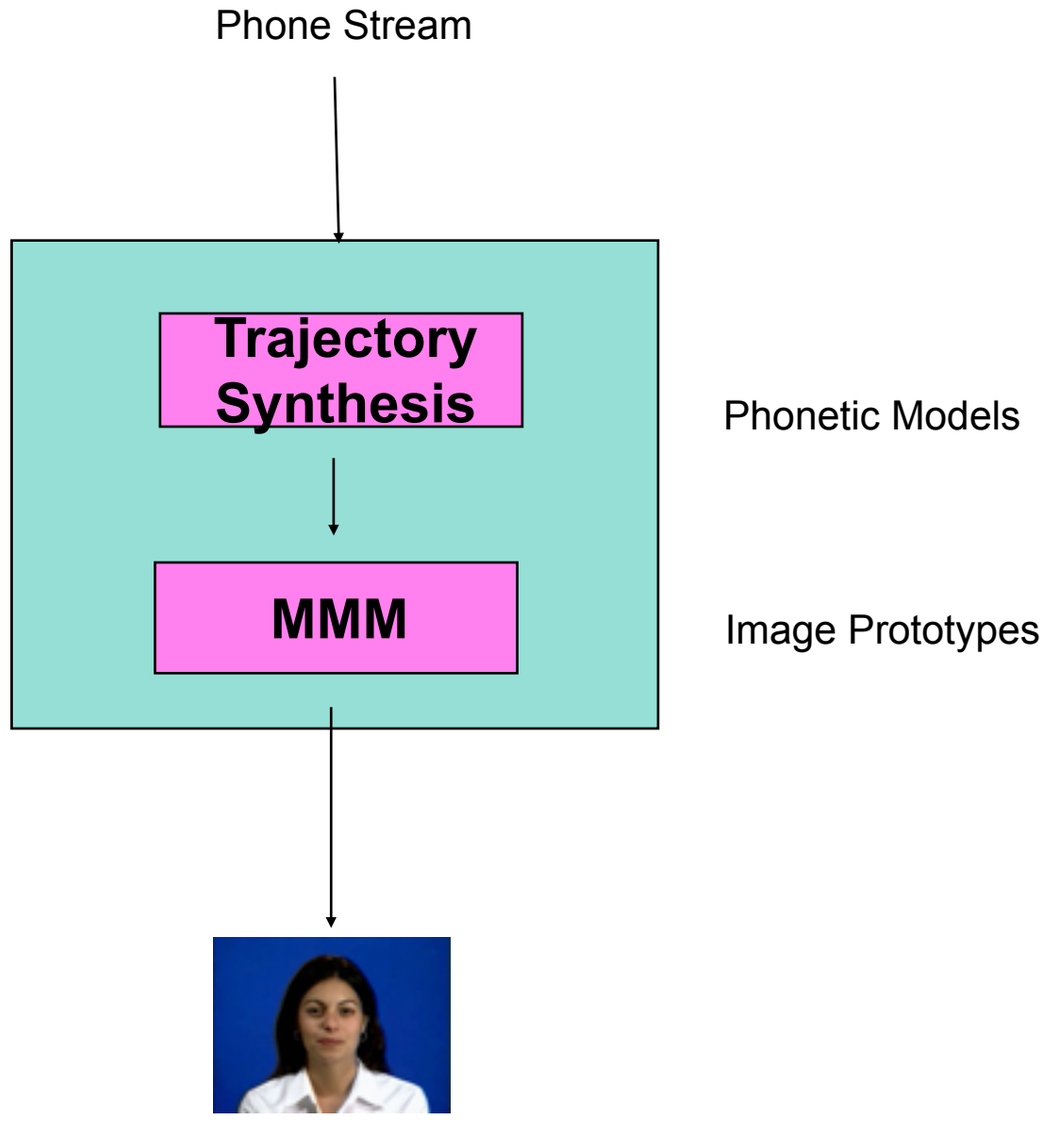
A- more in a moment

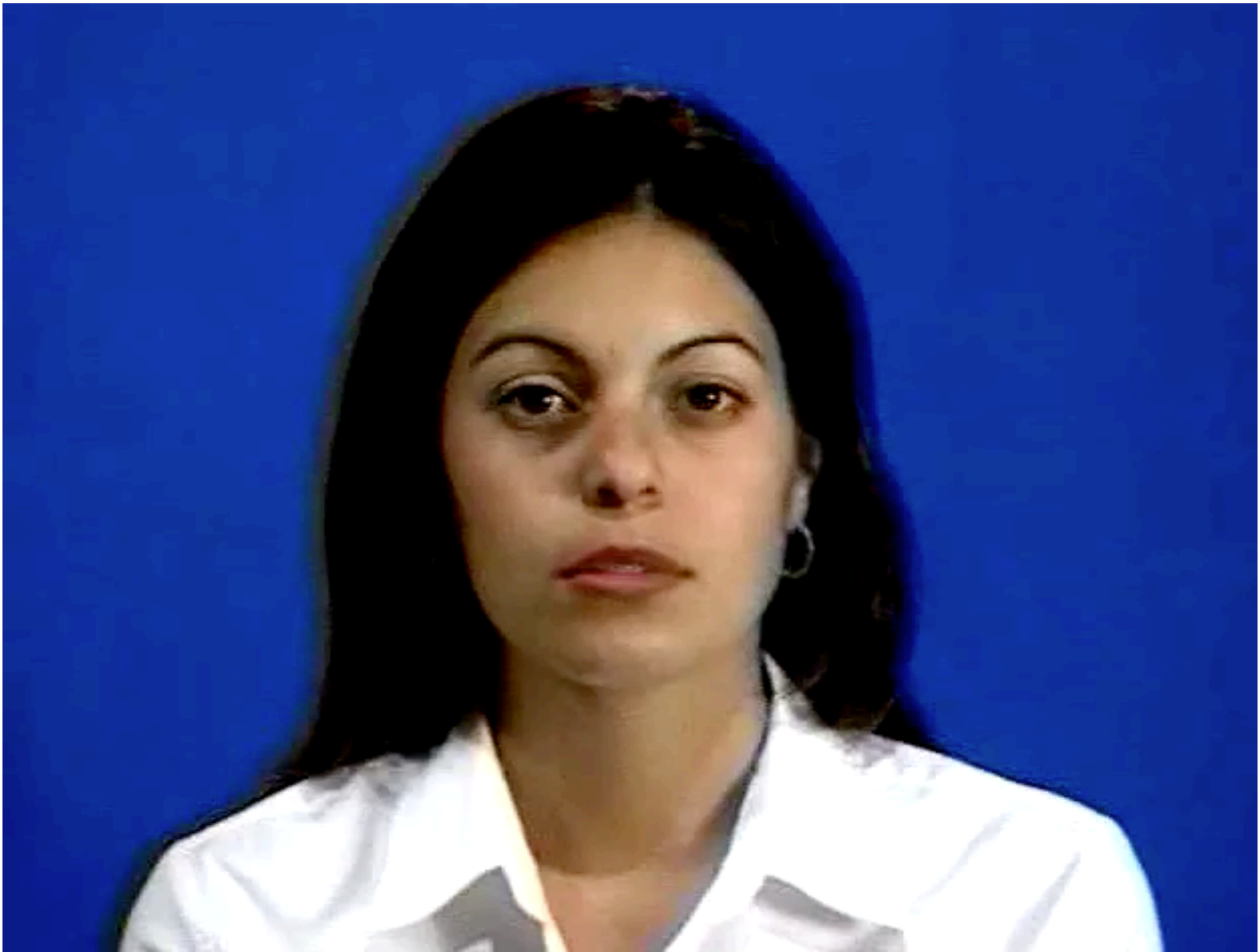
1. Learning

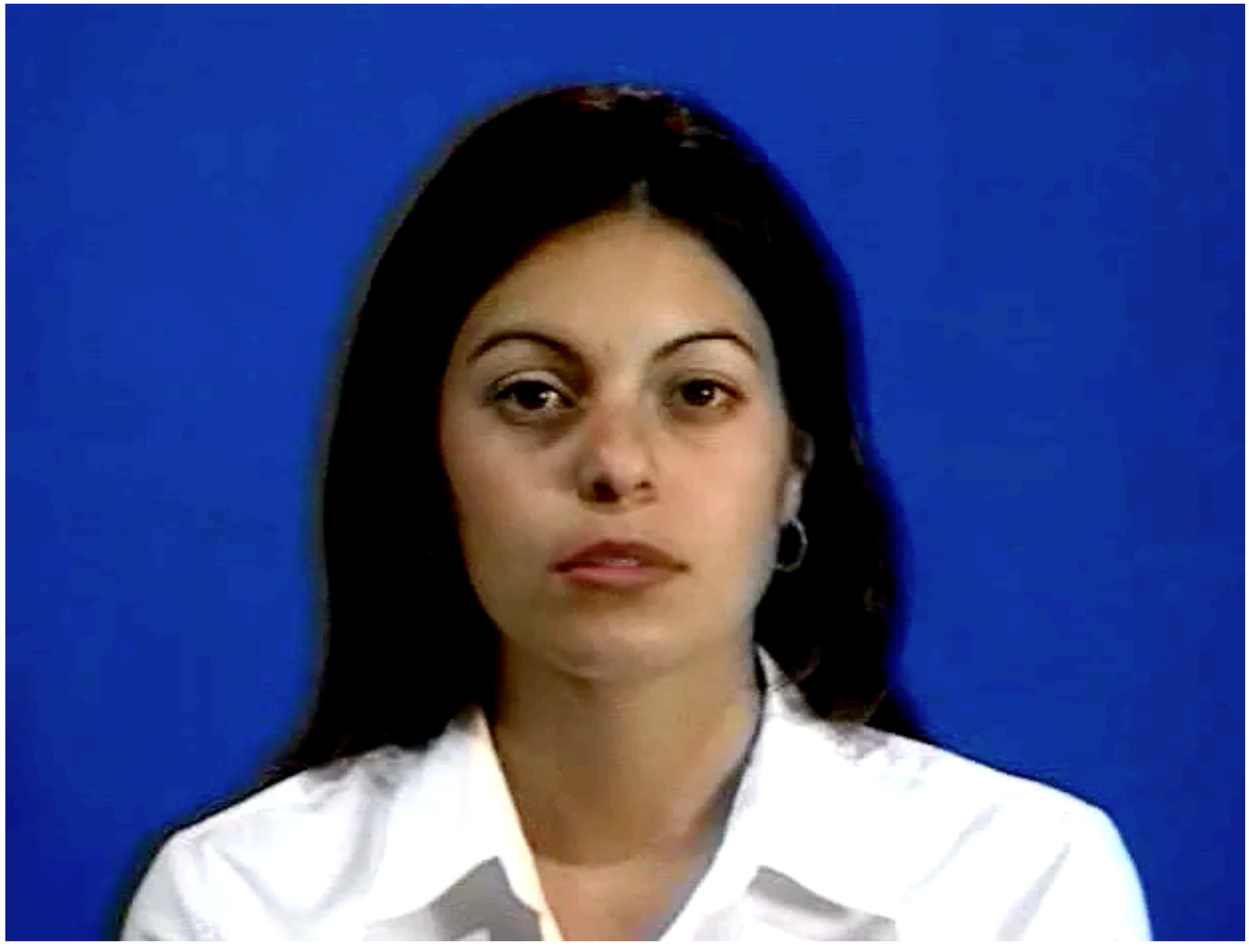
System learns from 4 mins of video face appearance (Morphable Model) and speech dynamics of the person

2. Run Time

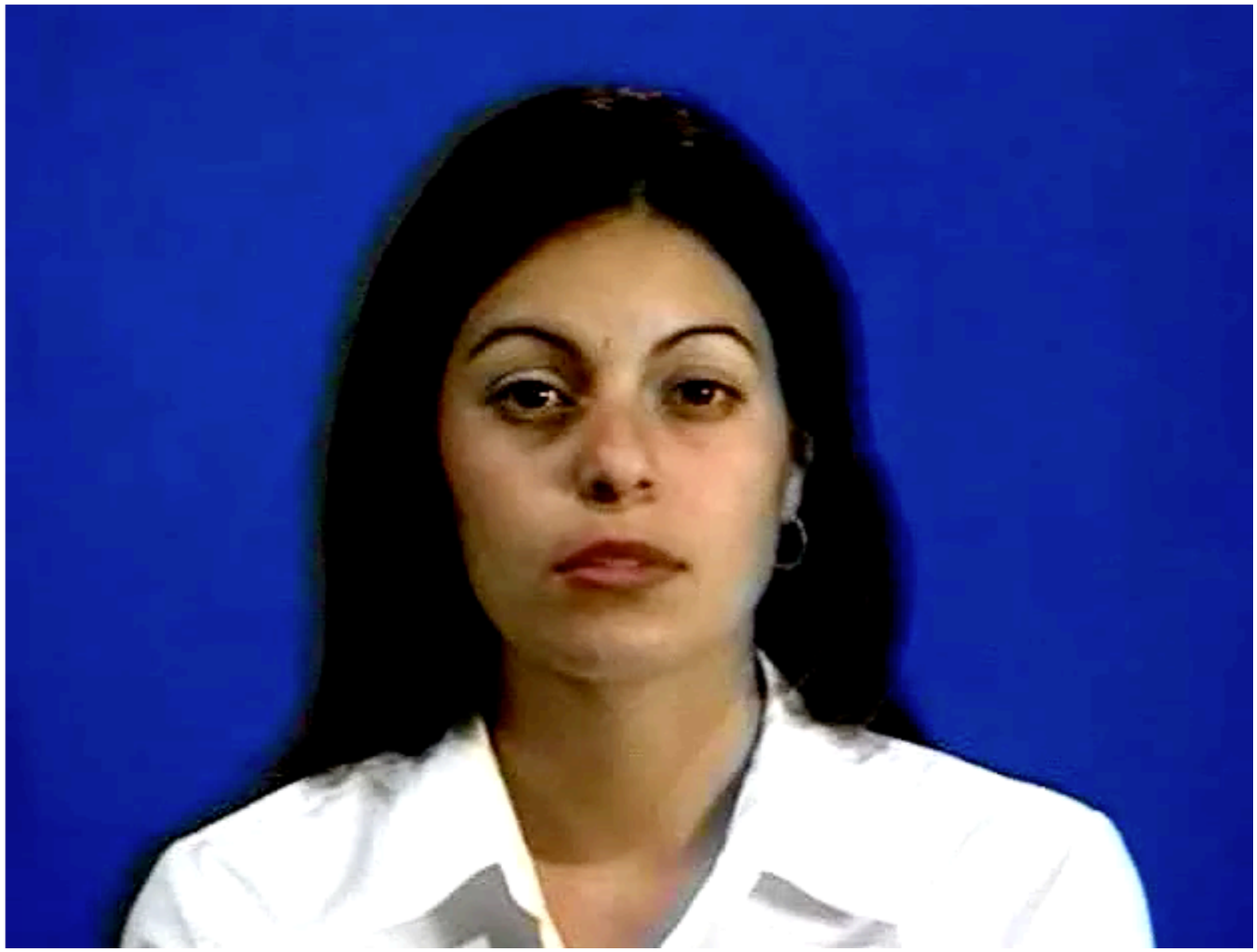
For any speech input the system provides as output a synthetic video stream



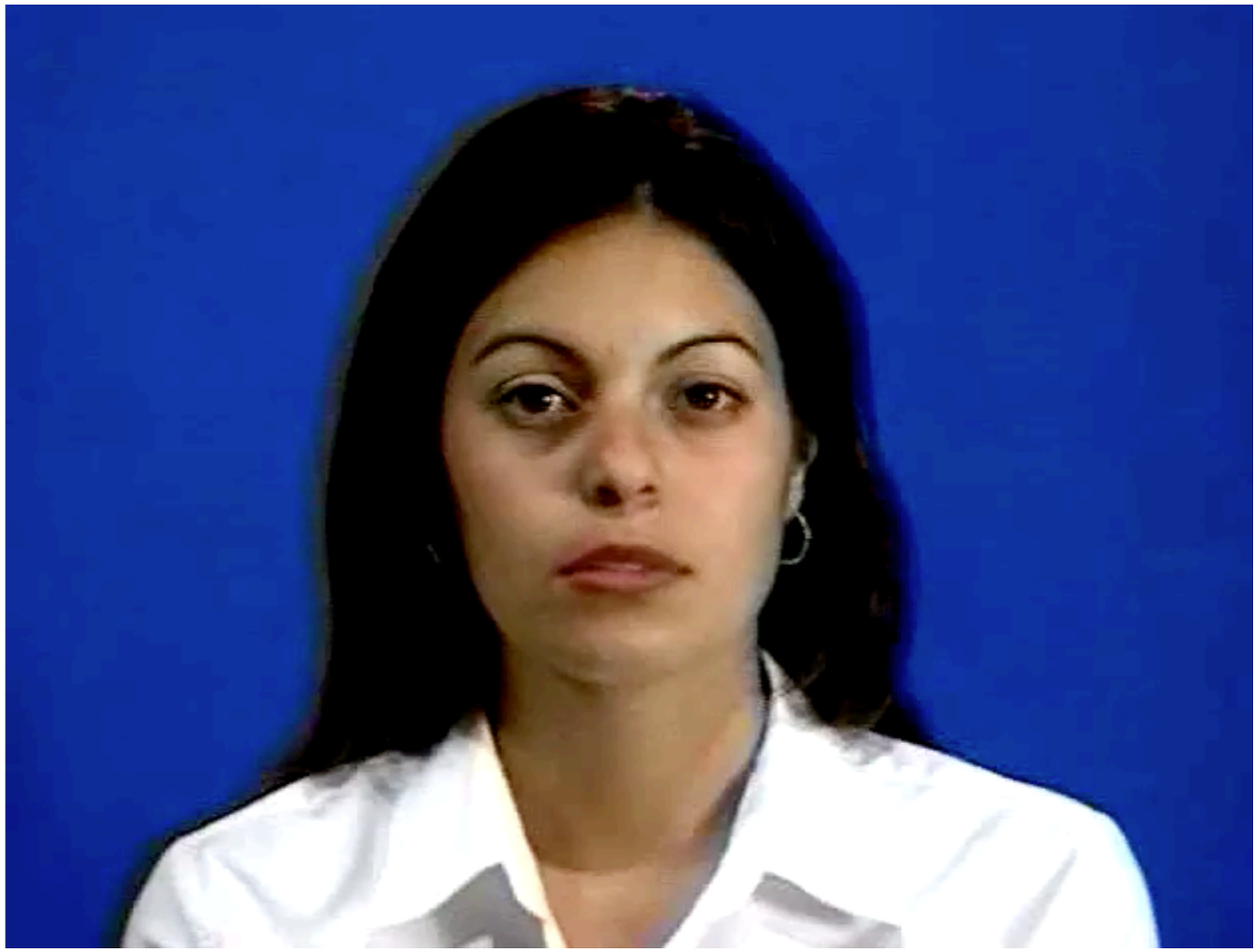




B-Dido



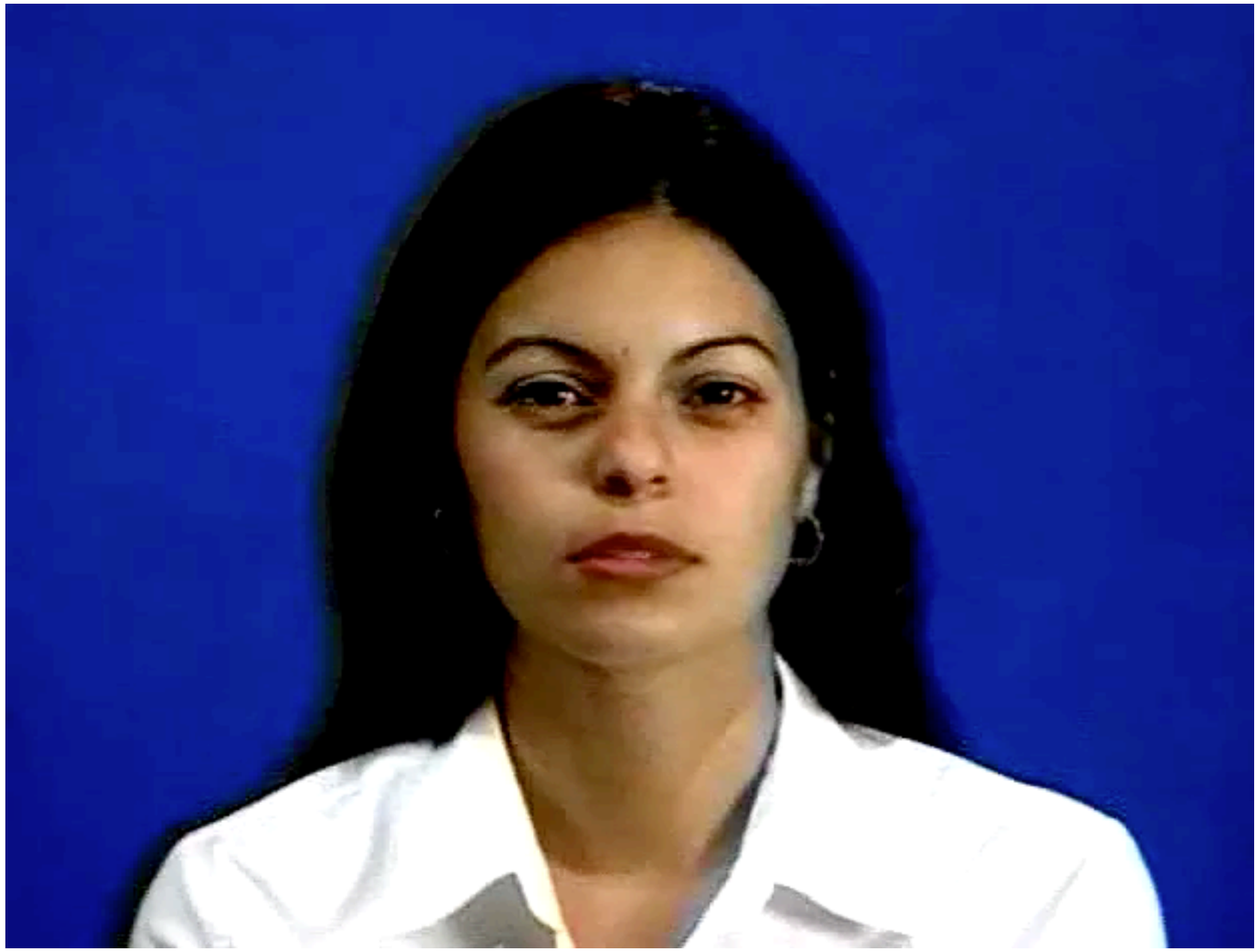
C-Hikaru



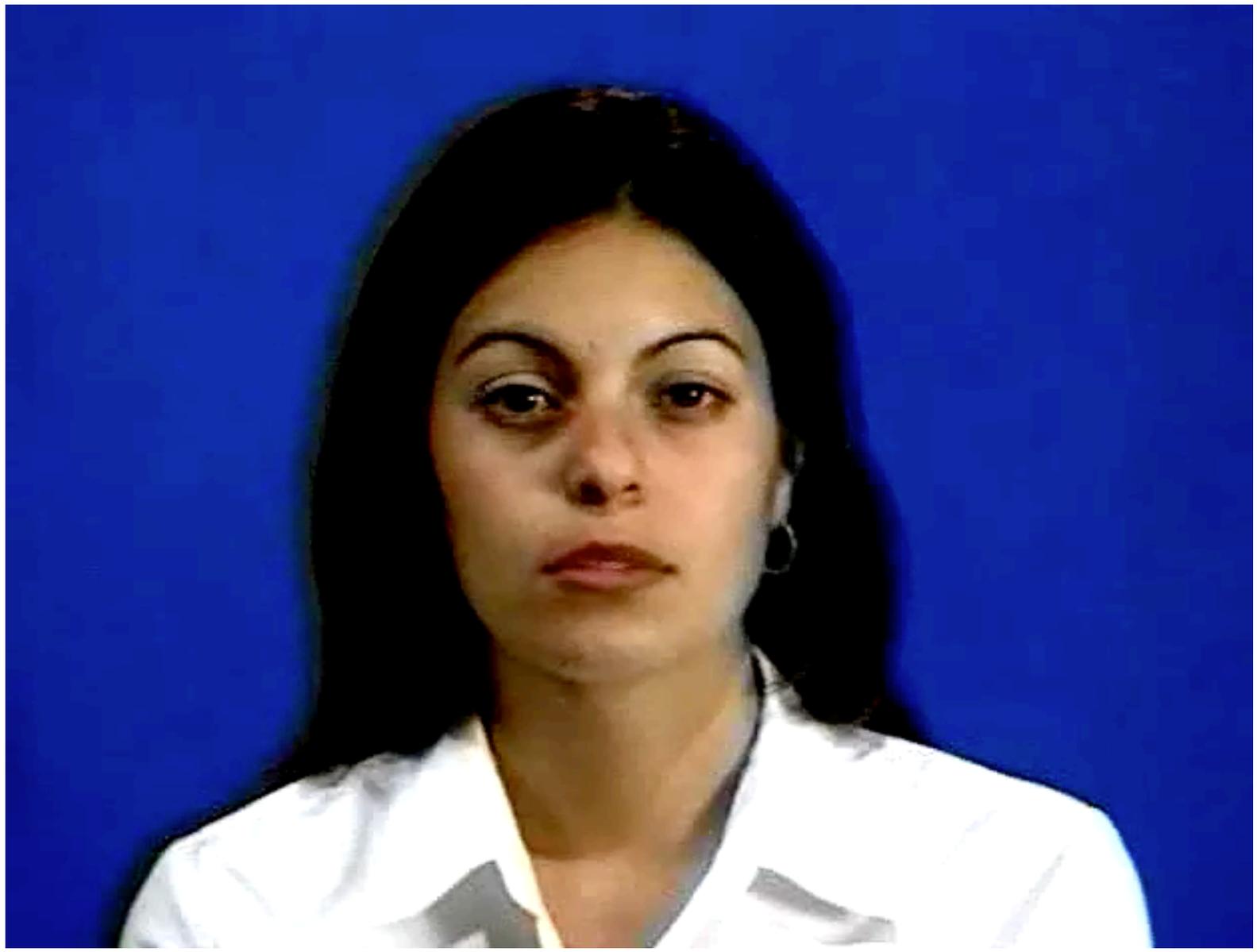
D-Denglijun

E-Marylin

F-Katie Couric



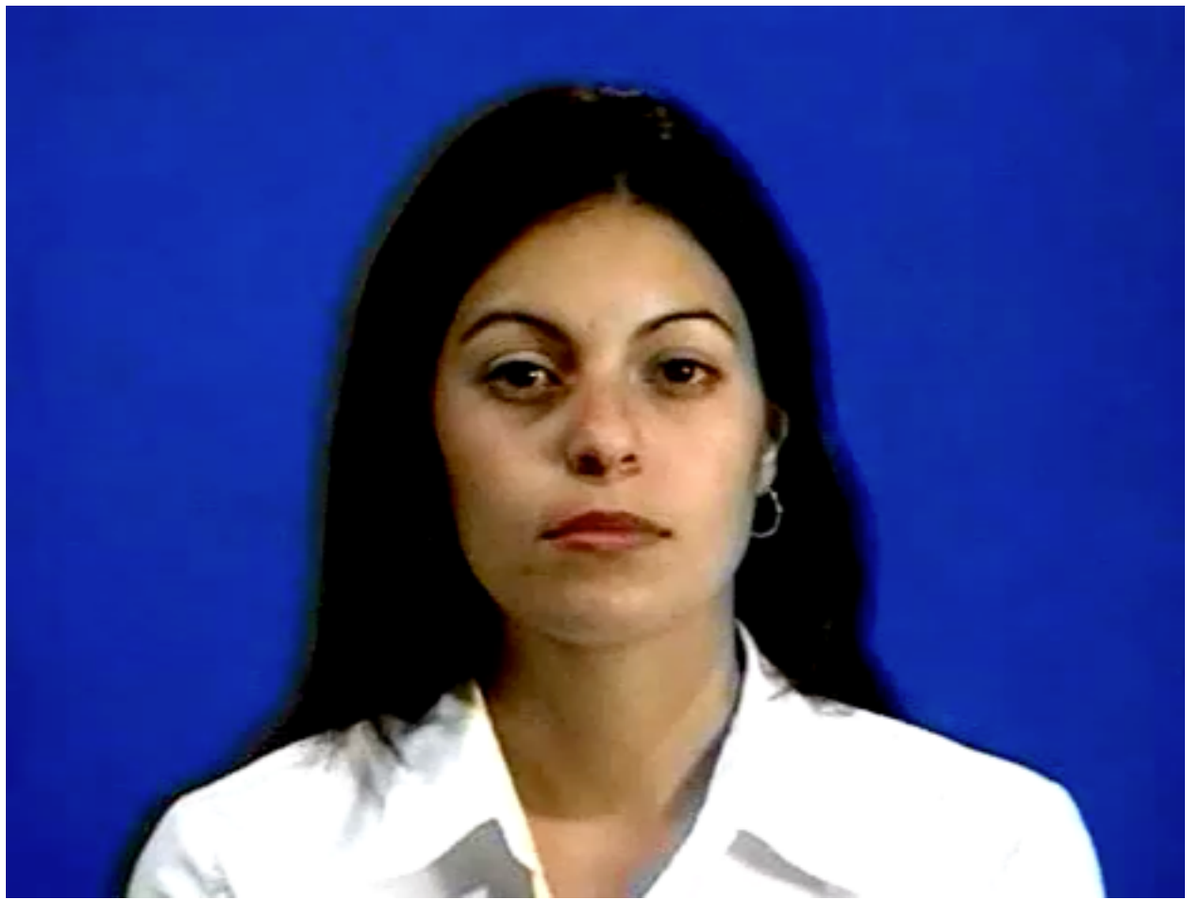
G-Katie



H-Rehema

I-Rehemax

A Turing test: what is real and what is synthetic?



L-real-synth

A Turing test: what is real and what is synthetic?

Experiment	# subjects	% correct	t	p<
Single pres.	22	54.3%	1.243	0.3
Fast single pres.	21	52.1%	0.619	0.5
Double pres.	22	46.6%	-0.75	0.5

Table 1: Levels of correct identification of real and synthetic sequences. t represents the value from a standard t-test with significance level of p<.

Tony Ezzat,Geiger, Poggio, **SigGraph 2002**