Aka Ankel

Introduction to Computer Vision

CSE 327 Spring 2012

Lecture 6

Prof. Alex Berg

Today

- Course perspective
  - Estimation -- of physical & semantic properties
- Recognition
  - Introduction
  - Classification
- Read
  - Chapter 14 (Come to class with 2 questions)
- New assignment out on Thursday
  - face detection

Computer Vision

Alexander C. Berg

Research Scientist
Columbia University

PhD U.C. Berkeley

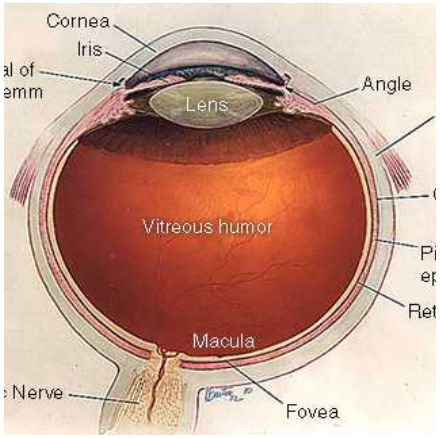
Computer vision

- Estimation (extracting information)
  - Physical properties
  - Semantic properties

Why Vision?

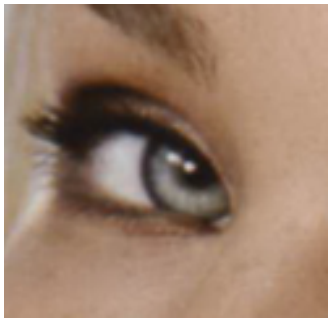
Why Vision? Light!

It is how we see other people, navigate our environment, communicate ideas, entertain, and **measure** the world around us.

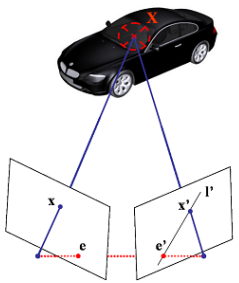


Why is light good for measurement?

Microscopy



Surveillance



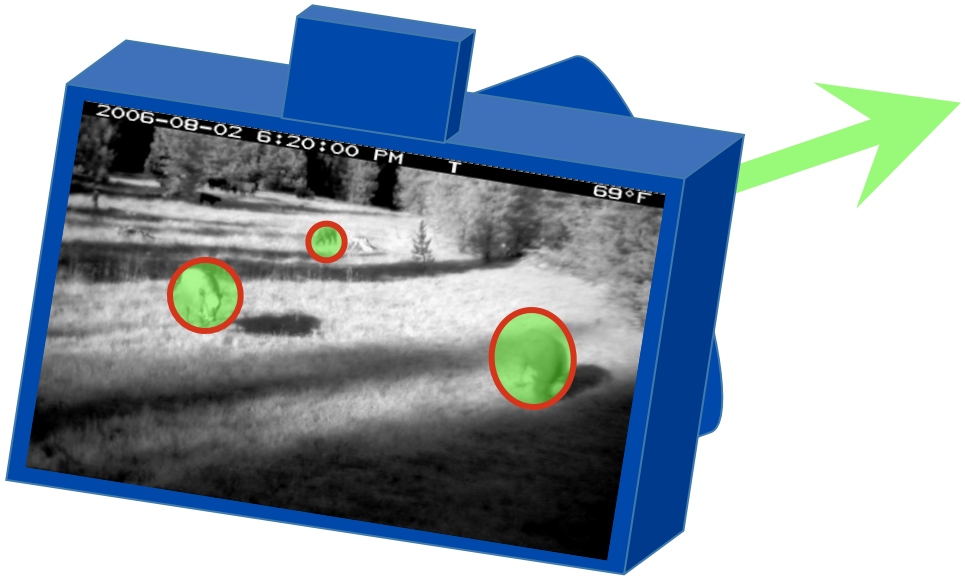
3D Analysis / Navigation

Remote Sensing

- Plentiful, sometimes free
- Interacts with many things, but not too many
- Goes generally straight over distance
- Very small → high spatial resolution
- Fast, but not too fast → time of flight sensors
- Easy to detect → cameras work, are cheap
- Comes in flavors (wavelengths)

Why Computational Visual Recognition?

Because we need to know which bits to measure!

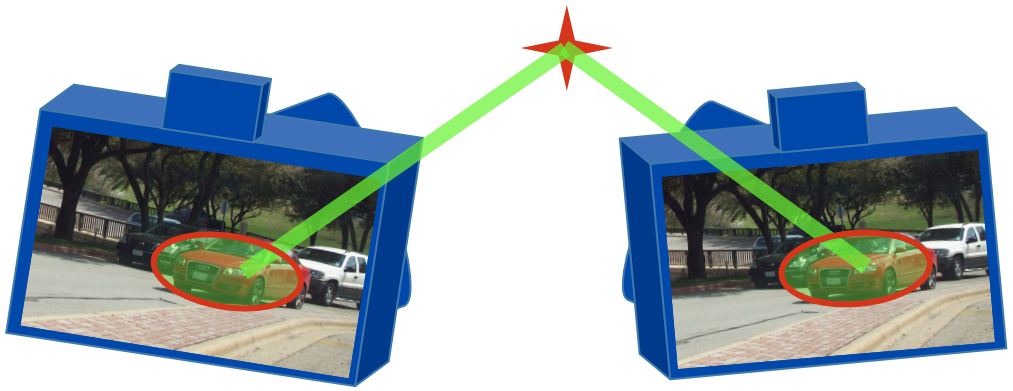


3 Cows at 6:20 PM

Need to recognize which parts of the image are cows, deer, humans, grass, shadows, etc.

Why Computational Visual Recognition?

Because we need to know which bits to measure!



Need to recognize which parts of the two images show the car.

Why Computational Visual Recognition?

Why Computational Visual Recognition?

Why Computational Visual Recognition?

Why Computational Visual Recognition?

Why Computational Visual Recognition?

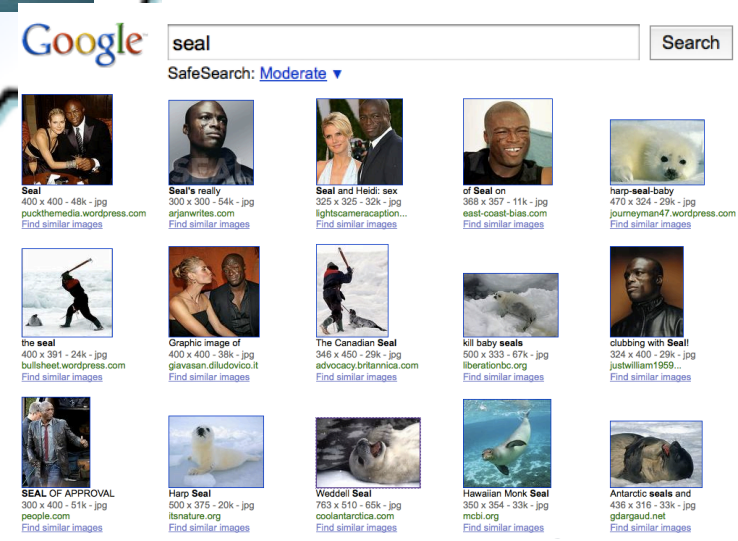
Google images

seal Search Images

Billions of images, many with no text / meta data

Why Computational Visual Recognition?

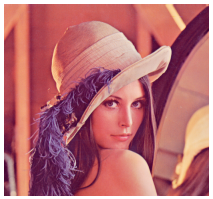
Billions of images, many with no text / meta data



Range of recognition tasks

- Duplicate detection
- Edge detection
- Same (rigid) object
- Face detection
- Face Identification
- General category recognition

Range of recognition tasks

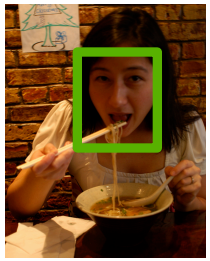


- Duplicate detection
- Edge detection
- Same (rigid) object
- Face detection
- Face Identification
- General category recognition

Range of recognition tasks

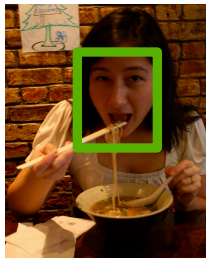

- Duplicate detection
- Edge detection
- Same (rigid) object
- Face detection
- Face Identification
- General category recognition

Range of recognition tasks



- Duplicate detection
- Edge detection
- Same (rigid) object
- Face detection
- Face Identification
- General category recognition

Range of recognition tasks



Tamara

- Duplicate detection
- Edge detection
- Same (rigid) object
- Face detection
- Face Identification
- General category recognition

Range of recognition tasks

Chair

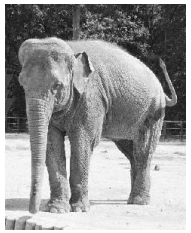
- Duplicate detection
- Edge detection
- Same (rigid) object
- Face detection
- Face Identification
- General category recognition

Outline

- Image Category Recognition
  - warm up
- Additive Models for Classification & Detection
  - efficient algorithms for very large datasets
- Describable Visual attributes for Face Recognition
  - benefitting from very large datasets

Exemplars:
An Approach to Category Recognition

What is this?



Database of Examples

Exemplars:
An Approach to Category Recognition

Pick the most similar exemplar…

Exemplars:
An Approach to Category Recognition

Query

Example

Transferred to Query

Known spatial support in Exemplar

Exemplars:
How to Measure Image Similarity?

- Directly comparing the pixels in the images does not work very well, how can we do better?

Detection

Find pedestrians

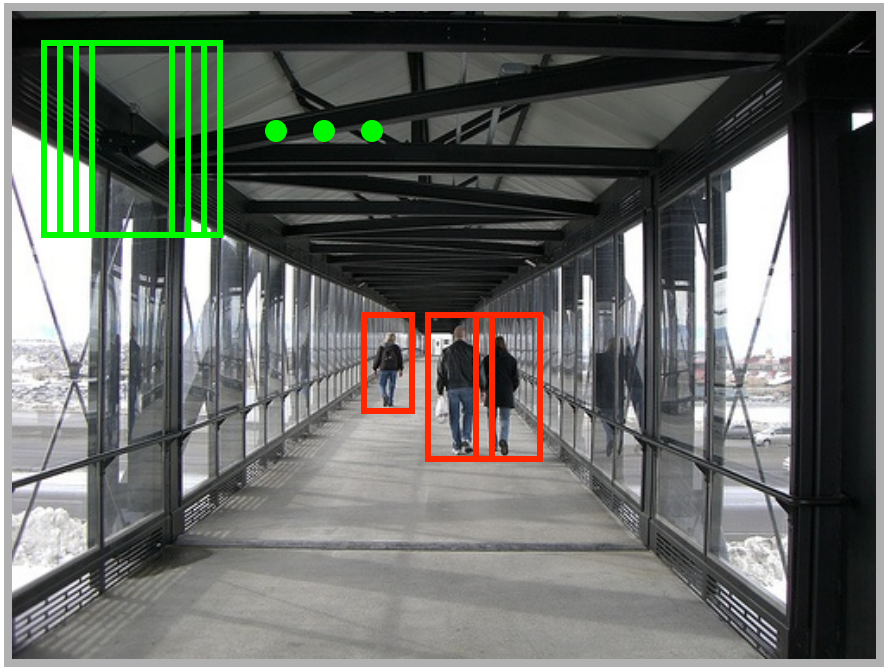
Detection

Find pedestrians

Detection

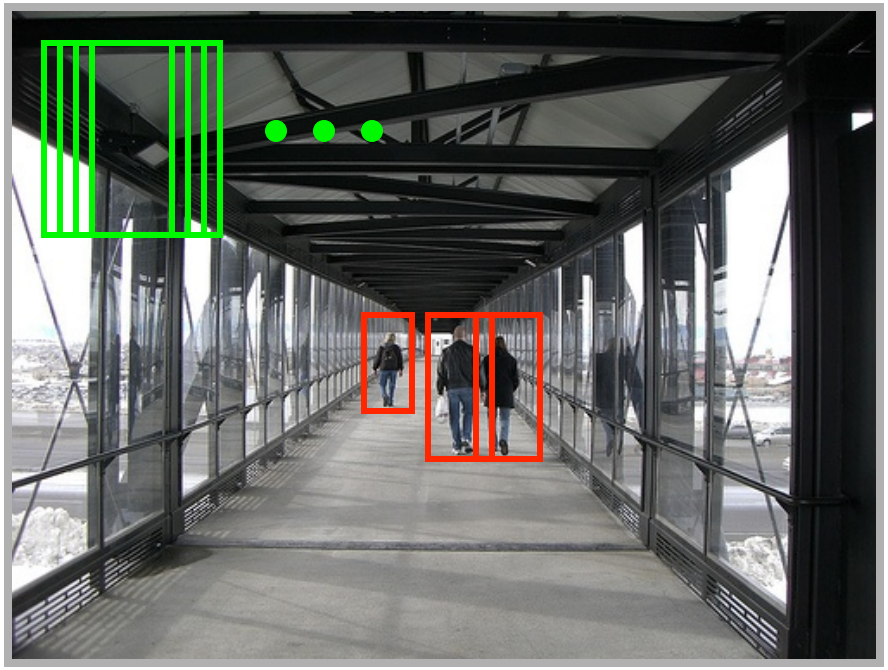
Find pedestrians

Detection



Find pedestrians

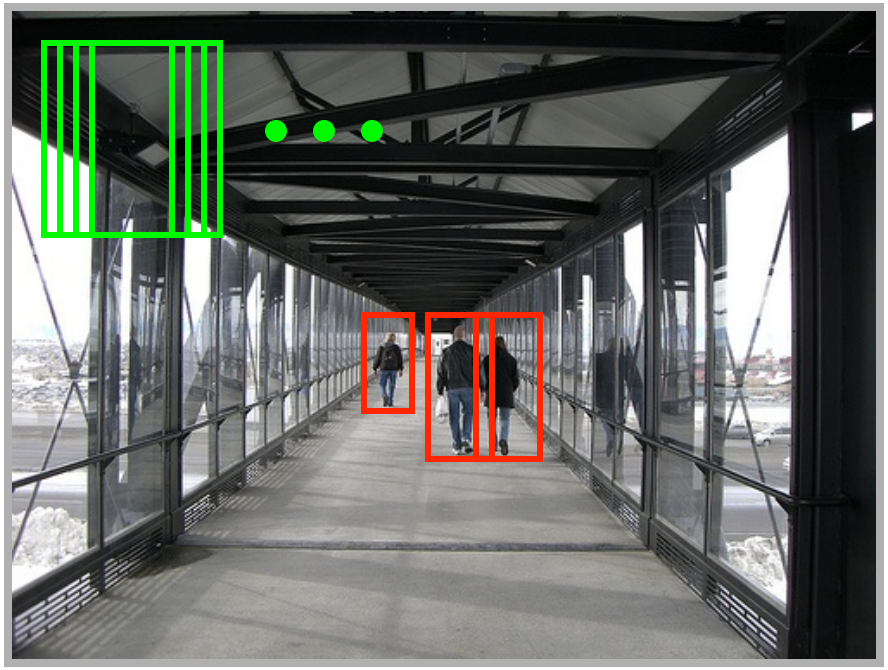
Detection



Find pedestrians

10^4 to 10^6 or more windows per image

Detection



Find pedestrians

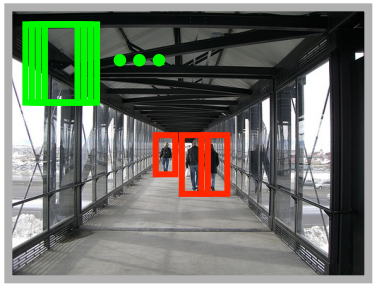
10^4 to 10^6 or more
windows per image

Boosting + Decision Trees
Viola & Jones (faces)

Linear Classifier
Dalal & Triggs (pedestrians)
Felzenszwalb et al

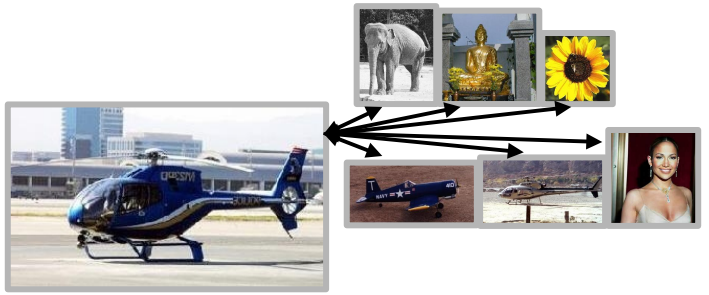
Neural Networks
Rowley et al (faces)

Detection



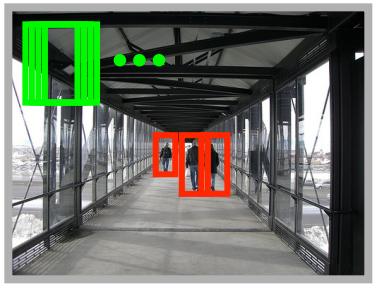
Linear Classifier

Categorization



Kernel + SVM Classifier

Detection

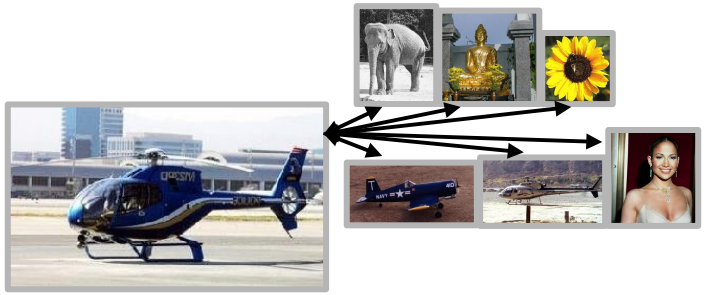


Linear Classifier

$$h(x) = \left(\sum_{i=1}^{\#\text{dimensions}} w_i x_i \right) + b$$

Decision function is *sign(h)*

Categorization

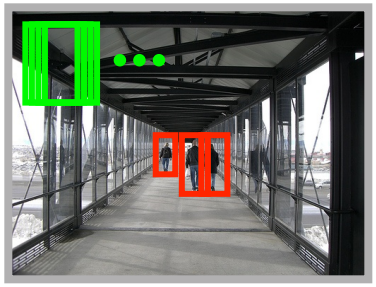


Kernel + SVM Classifier

$$h(x) = \sum_{j=1}^{\#\text{sv}} \alpha^j K(x, x^j) + b$$

Decision function is *sign(h)*

Detection



Linear Classifier

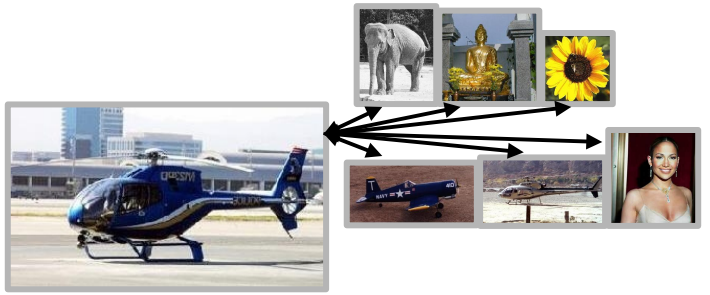
O(#dims)

$$h(x) = \left(\sum_{i=1}^{\#\text{dimensions}} w_i x_i \right) + b$$

Test feature vector

One coordinate of feature vector

Categorization



Kernel + SVM Classifier

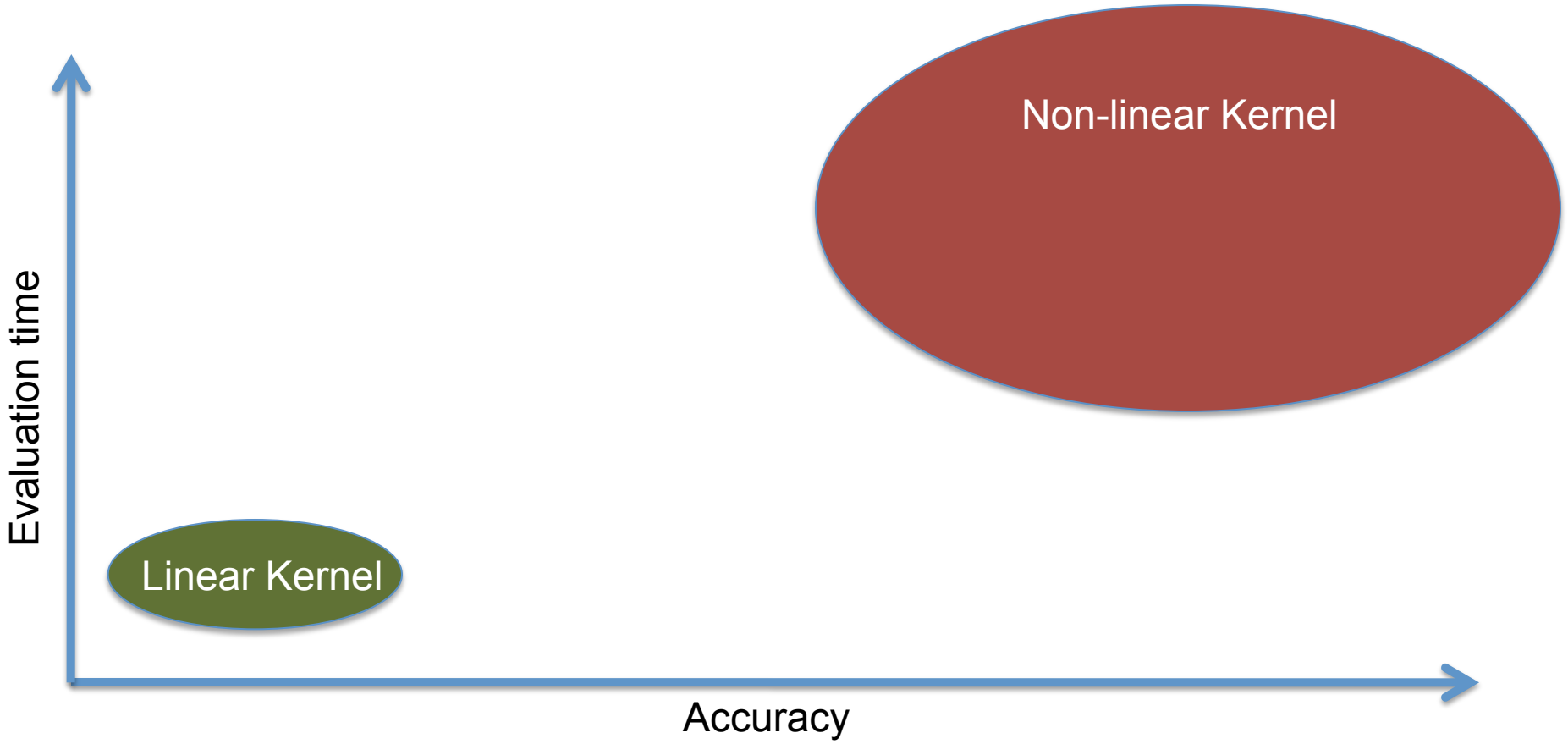
O(#sv x #dims)

$$h(x) = \sum_{j=1}^{\#\text{sv}} \alpha^j K(x, x^j) + b$$

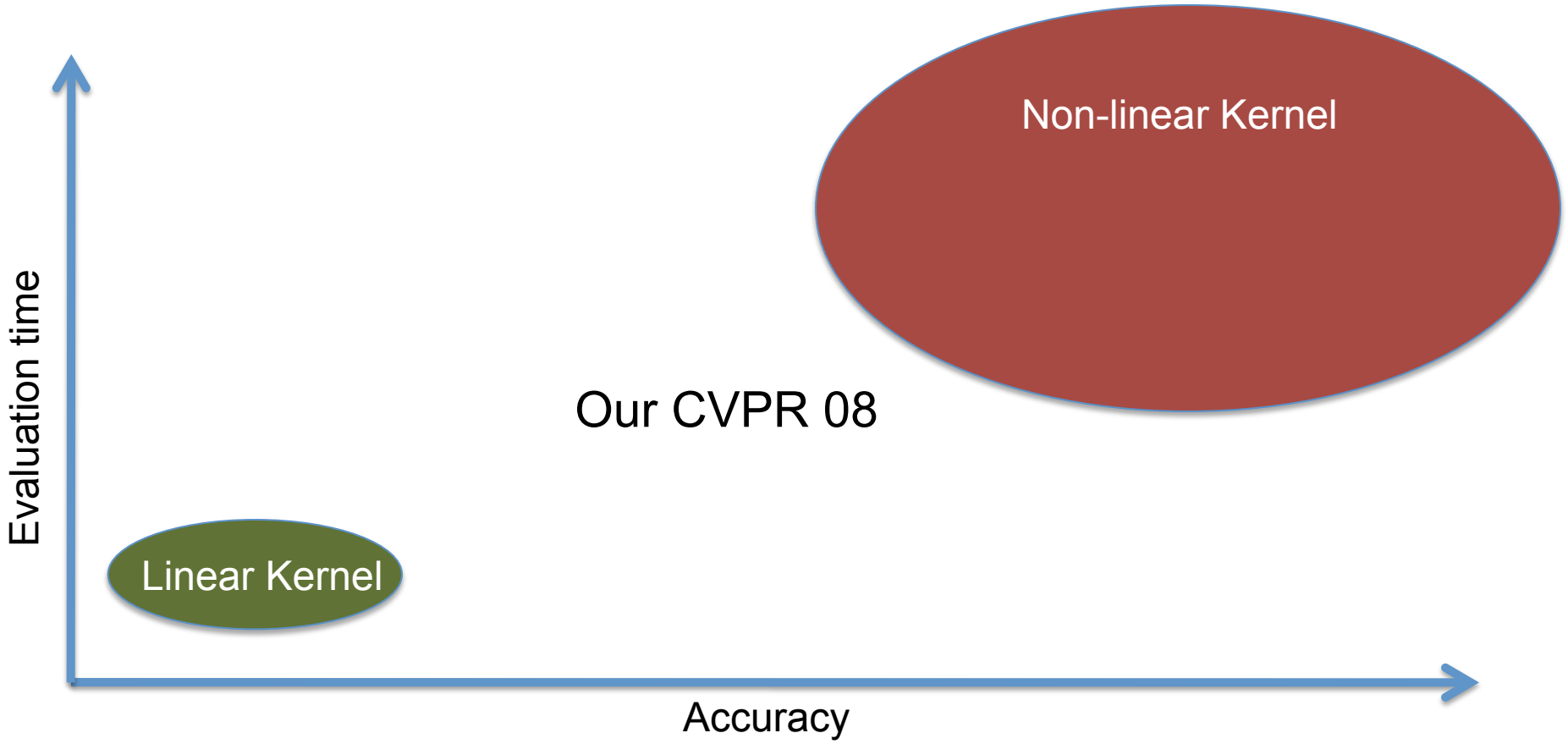
Kernel Function (comparison)

Support Vector (training example)

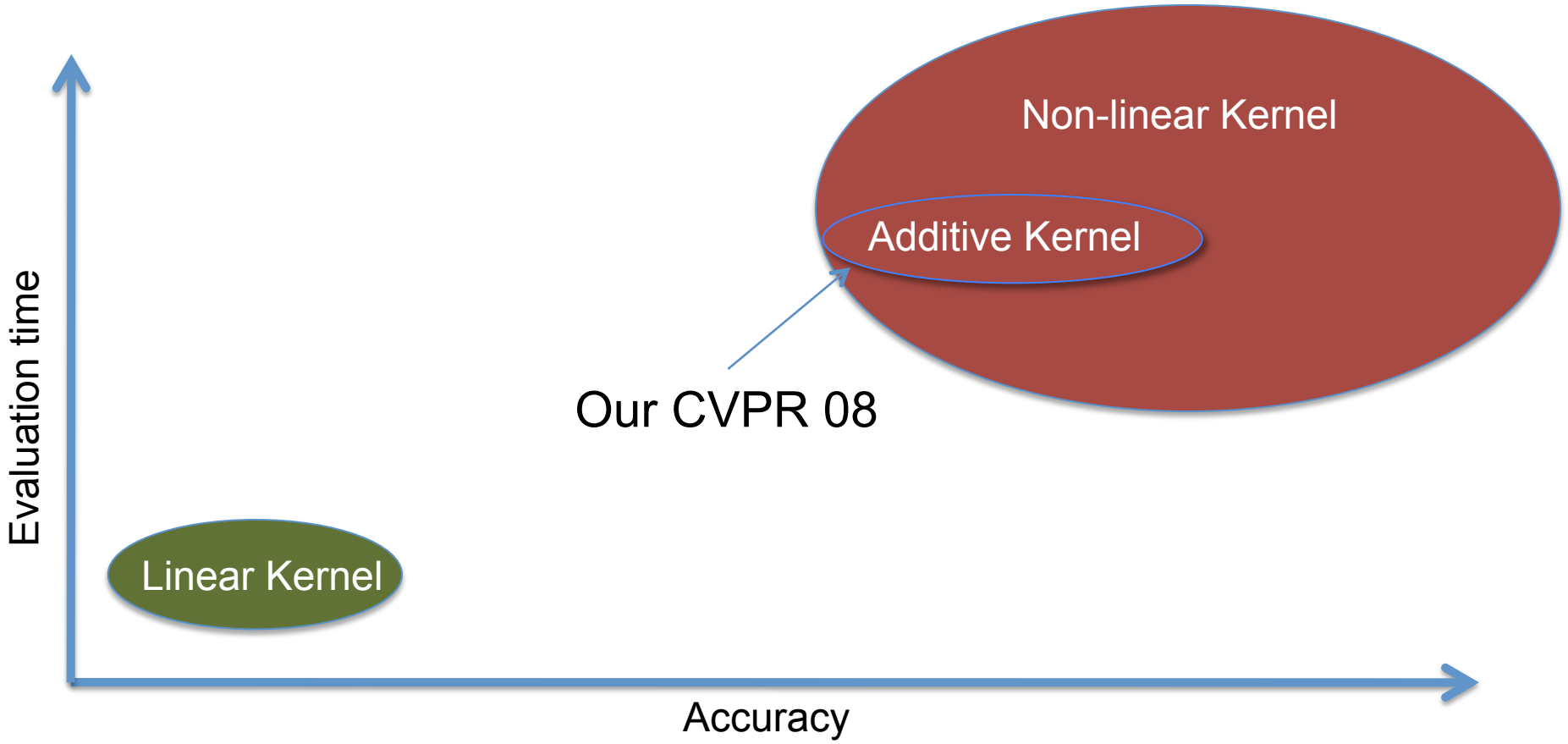
Accuracy vs. Evaluation Time for SVM Classifiers



Accuracy vs. Evaluation Time for SVM Classifiers

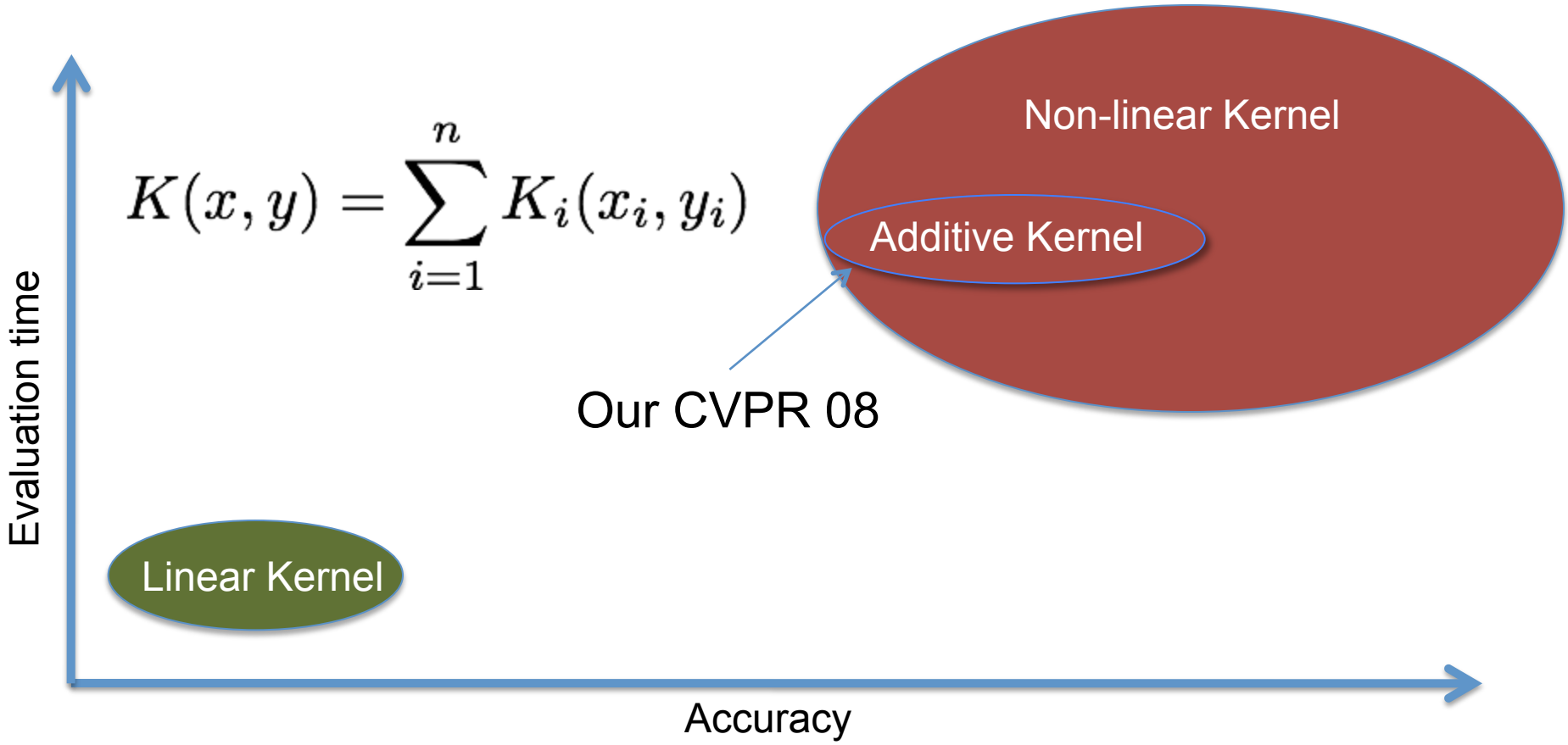


Accuracy vs. Evaluation Time for SVM Classifiers

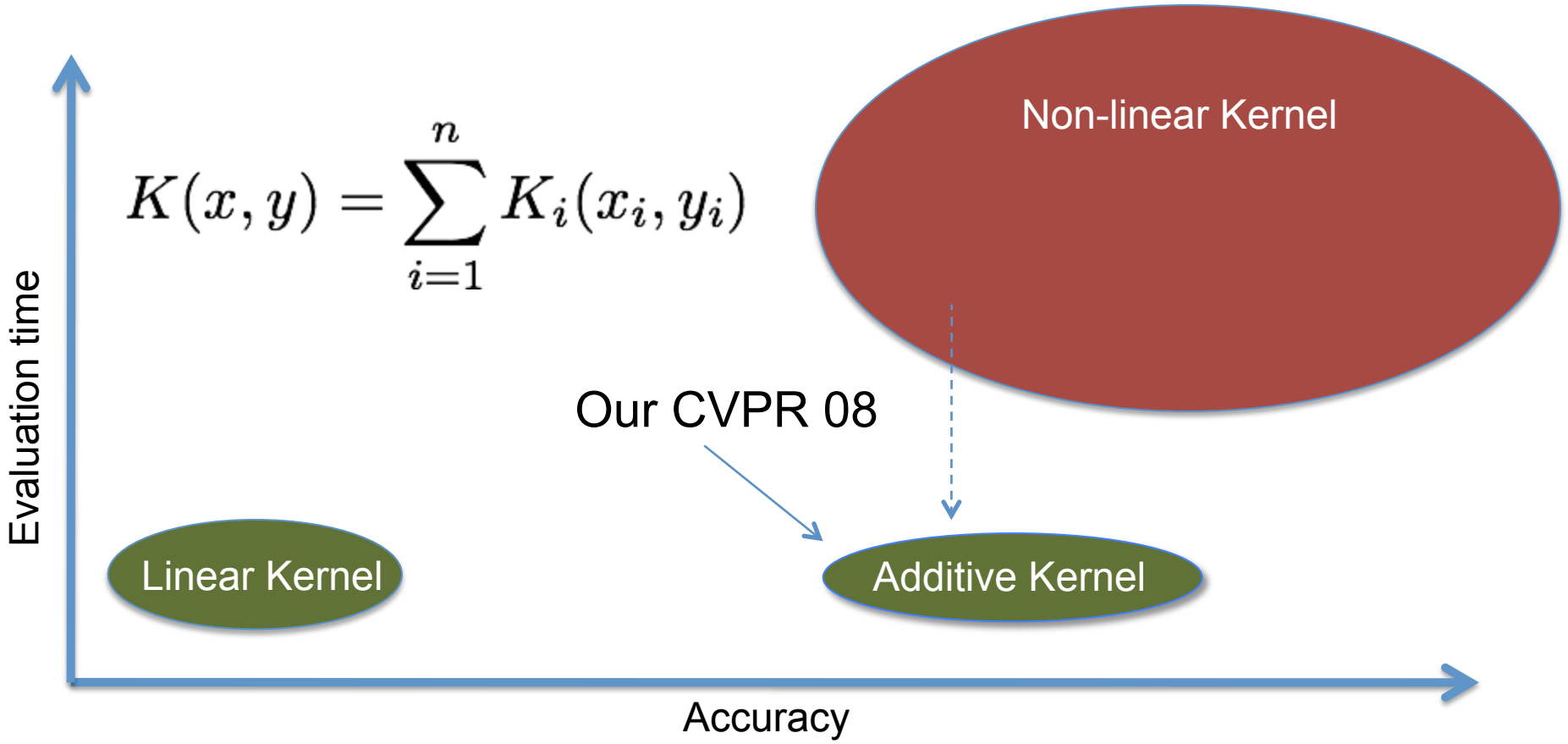


Accuracy vs. Evaluation Time for SVM Classifiers

$$K(x, y) = \sum_{i=1}^{n} K_i(x_i, y_i)$$



Accuracy vs. Evaluation Time for SVM Classifiers



Made it possible to use SVMs with additive kernels for detection and other large problems

Additive Classifiers

$$h(x) = h_1(x_1) + h_2(x_2) + \ldots + h_n(x_n)$$

- Commonly used! Any SVM with an additive kernel is an additive classifier
- Histogram intersection, chi-squared kernel

$$K_{\min}(x, y) = \sum_{i=1}^{n} \min(x_i, y_i) \qquad K_{\chi^2}(x, y) = 1 - \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)}$$

  - Pyramid Match Kernel (Grauman & Darell, ICCV'05)
  - Spatial Pyramid Kernel (Lazebnik et.al., CVPR'06)
  - Work from Oxford (Vedaldi, Chum, Bosch, Zisserman et al '06-'09)
- A boosted decision stump classifier is also additive
  - Any linear combination of additive functions is additive

A SVM with *additive* Kernel can be Evaluated Efficiently

Maji, Berg, Malik CVPR 2008

If $$K(a,b) = \sum_{i=1}^{\#\text{dimensions}} K_i(a_i, b_i)$$

Then
$$h(x) = \sum_{j=1}^{\#\text{sv}} \alpha^j K(x, x^j) + b$$
$$= \sum_{j=1}^{\#\text{sv}} \alpha^j \left(\sum_{i=1}^{\#\text{dimensions}} K_i(x_i, x_i^j) \right) + b$$
$$= \sum_{i=1}^{\#\text{dimensions}} \left(\sum_{j=1}^{\#\text{sv}} \alpha^j K_i(x_i, x_i^j) \right) + b$$
$$= \sum_{i=1}^{\#\text{dimensions}} h_i(x_i)$$

A SVM with *additive* Kernel can be Evaluated Efficiently

Maji, Berg, Malik CVPR 2008

If $K(a,b) = \sum_{i=1}^{\#\text{dimensions}} K_i(a_i, b_i)$

If you have an additive kernel…

Then

$$h(x) = \sum_{j=1}^{\#\text{sv}} \alpha^j K(x, x^j) + b$$

$$= \sum_{j=1}^{\#\text{sv}} \alpha^j \left(\sum_{i=1}^{\#\text{dimensions}} K_i(x_i, x_i^j) \right) + b$$

$$= \sum_{i=1}^{\#\text{dimensions}} \left(\sum_{j=1}^{\#\text{sv}} \alpha^j K_i(x_i, x_i^j) \right) + b$$

$$= \sum_{i=1}^{\#\text{dimensions}} h_i(x_i)$$

A SVM with *additive* Kernel can be Evaluated Efficiently

Maji, Berg, Malik CVPR 2008

If $K(a,b) = \sum_{i=1}^{\#\text{dimensions}} K_i(a_i, b_i)$

If you have an additive kernel…

Then

$$h(x) = \sum_{j=1}^{\#\text{sv}} \alpha^j K(x, x^j) + b$$

$$= \sum_{j=1}^{\#\text{sv}} \alpha^j \left(\sum_{i=1}^{\#\text{dimensions}} K_i(x_i, x_i^j) \right) + b$$

$$= \sum_{i=1}^{\#\text{dimensions}} \left(\sum_{j=1}^{\#\text{sv}} \alpha^j K_i(x_i, x_i^j) \right) + b$$

$$= \sum_{i=1}^{\#\text{dimensions}} h_i(x_i)$$

A SVM with *additive* Kernel can be Evaluated Efficiently

Maji, Berg, Malik CVPR 2008

If $K(a,b) = \sum_{i=1}^{\#\text{dimensions}} K_i(a_i, b_i)$

If you have an additive kernel…

Then $h(x) = \sum_{j=1}^{\#\text{sv}} \alpha^j K(x, x^j) + b$

$= \sum_{j=1}^{\#\text{sv}} \alpha^j \left(\sum_{i=1}^{\#\text{dimensions}} K_i(x_i, x_i^j) \right) + b$

then the SVM decision function is additive.

$= \sum_{i=1}^{\#\text{dimensions}} \left(\sum_{j=1}^{\#\text{sv}} \alpha^j K_i(x_i, x_i^j) \right) + b$

$= \sum_{i=1}^{\#\text{dimensions}} h_i(x_i)$

A SVM with *additive* Kernel can be Evaluated Efficiently

Maji, Berg, Malik CVPR 2008

If $K(a,b) = \sum_{i=1}^{\#\text{dimensions}} K_i(a_i, b_i)$

If you have an additive kernel…

Then $h(x) = \sum_{j=1}^{\#\text{sv}} \alpha^j K(x, x^j) + b$

$= \sum_{j=1}^{\#\text{sv}} \alpha^j \left(\sum_{i=1}^{\#\text{dimensions}} K_i(x_i, x_i^j) \right) + b$

then the SVM decision function is additive.

$= \sum_{i=1}^{\#\text{dimensions}} \left(\sum_{j=1}^{\#\text{sv}} \alpha^j K_i(x_i, x_i^j) \right) + b$

$= \sum_{i=1}^{\#\text{dimensions}} h_i(x_i)$

Evaluate these 1D functions efficiently using a look up table, spline (exact or approx.)

Intersection or Min Kernel

Maji, Berg, Malik CVPR 2008

$$K_{\min}(a,b) = \sum_{i=1}^{\#\text{dimensions}} min(a_i, b_i)$$

The Intersection or Min Kernel

Grauman et al Pyramid Match Kernel

Lazebnik et al spatial pyramids

Much follow on work…

Intersection or Min Kernel

Maji, Berg, Malik CVPR 2008

$$K_{\min}(a,b) = \sum_{i=1}^{\#\text{dimensions}} min(a_i, b_i)$$

The Intersection or Min Kernel

$$\begin{aligned} h(x) &= \sum_{j=1}^{\#\text{sv}} \alpha^j K_{\min}(x, x^j) + b \\ &= \sum_{j=1}^{\#\text{sv}} \alpha^j \left(\sum_{i=1}^{\#\text{dimensions}} \min(x_i, x_i^j) \right) + b \\ &= \sum_{i=1}^{\#\text{dimensions}} h_i(x_i) + b \end{aligned}$$

Where $h_i(x_i) = \sum_{j=1}^{\#sv} \alpha^j \min(x_i, x_i^j)$

Intersection or Min Kernel

Maji, Berg, Malik CVPR 2008

$$K_{\min}(a,b) = \sum_{i=1}^{\#\text{dimensions}} min(a_i, b_i)$$

The Intersection or Min Kernel

$$h(x) = \sum_{j=1}^{\#\text{sv}} \alpha^j K_{\min}(x, x^j) + b$$

$$= \sum_{j=1}^{\#\text{sv}} \alpha^j \left(\sum_{i=1}^{\#\text{dimensions}} \min(x_i, x_i^j) \right) + b$$

$$= \sum_{i=1}^{\#\text{dimensions}} h_i(x_i) + b$$

Support vectors are fixed & min(x_i , *const.*) is piecewise linear, so $h_i(x_i)$ is piecewise linear.

Where $h_i(x_i) = \sum_{j=1}^{\#sv} \alpha^j \min(x_i, x_i^j)$

Intersection or Min Kernel

Maji, Berg, Malik CVPR 2008

$$K_{\min}(a,b) = \sum_{i=1}^{\#\text{dimensions}} min(a_i, b_i)$$

The Intersection or Min Kernel

$$h(x) = \sum_{j=1}^{\#\text{sv}} \alpha^j K_{\min}(x, x^j) + b$$

$$= \sum_{j=1}^{\#\text{sv}} \alpha^j \left(\sum_{i=1}^{\#\text{dimensions}} \min(x_i, x_i^j) \right) + b$$

$$= \sum_{i=1}^{\#\text{dimensions}} h_i(x_i) + b$$

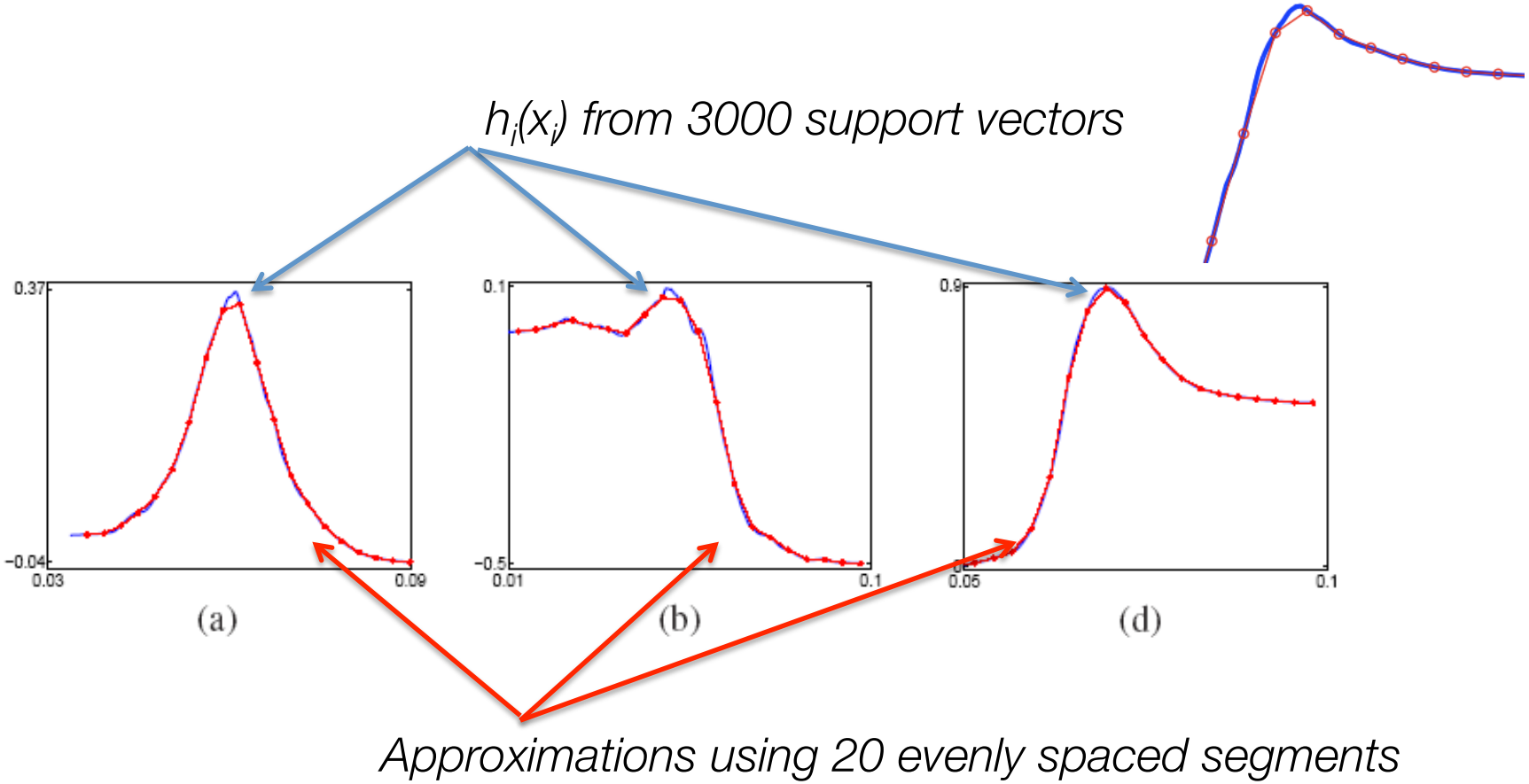
O(#dims x #sv) becomes O(#dims x log(#sv)) exact

or O(#dims) approx.

Support vectors are fixed & min(x_i , *const.*) is piecewise linear, so $h_i(x_i)$ is piecewise linear.

Where $h_i(x_i) = \sum_{j=1}^{\#sv} \alpha^j \min(x_i, x_i^j)$

Example $h_i(x_i)$ and Approximations



Time to Perform Classification

Maji, Berg, Malik CVPR 2008

			Linear	Additive			
	Model parameters		SVM kernel type		fast IKSVMs		
Dataset	#SVs	#features	linear	intersection	binary search	piecewise-const	piecewise-lin
INRIA Ped	3363	1360	0.07±0.00	659.1±1.92	2.57±0.03	0.34±0.01	0.43±0.01
DC Ped	5474±395	656	0.03±0.00	459.1±31.3	1.43±0.02	0.18±0.01	0.22±0.00
Caltech 101	175±46	1360	0.07±0.01	24.77±1.22	1.63±0.12	0.33±0.03	0.46±0.03

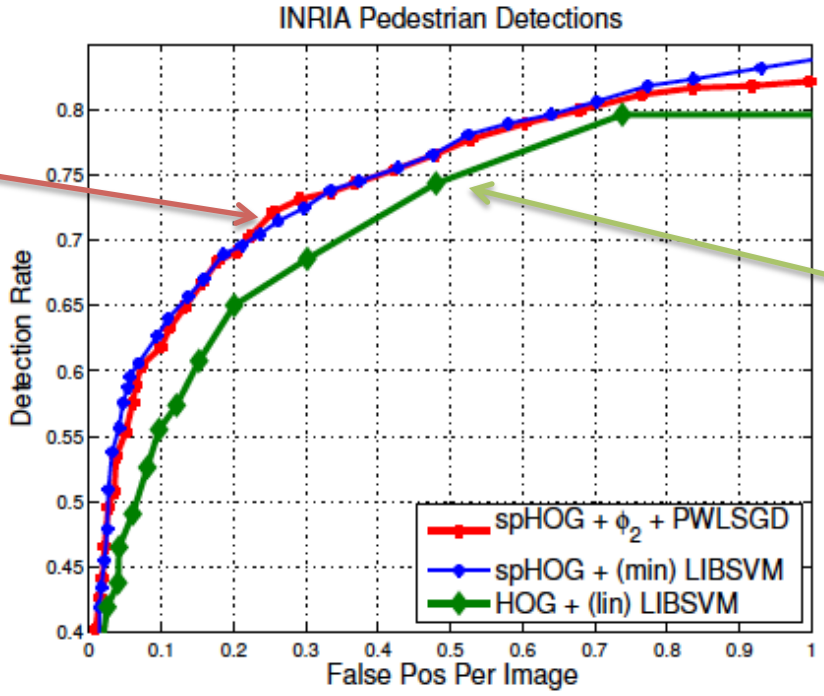
Times in seconds to classify 10,000 test vectors

Exactly the same classifier, more than 200 times faster than the straightforward version

Approximately the same classifier, more than 1000 times faster.

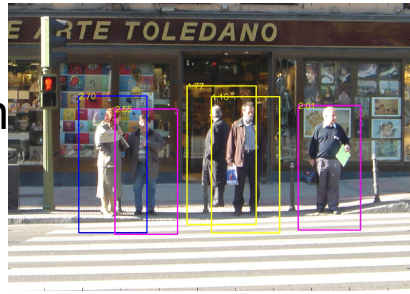
Min Kernel “Better” than Linear for Detection = Classification + Non-Max Suppression

IKSVM / Fast IKSVM Classifier



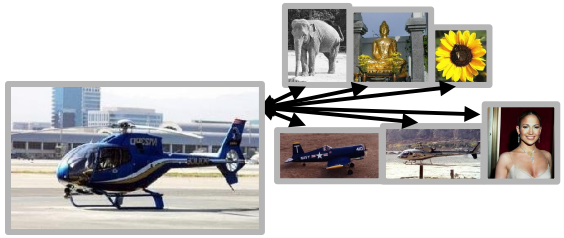
Linear Classifier

Pedestrian Detection



Min Kernel “Better” than Linear (other classification problems)

Caltech 101 **Categorization** with SPM “simple features”
15 training examples per category



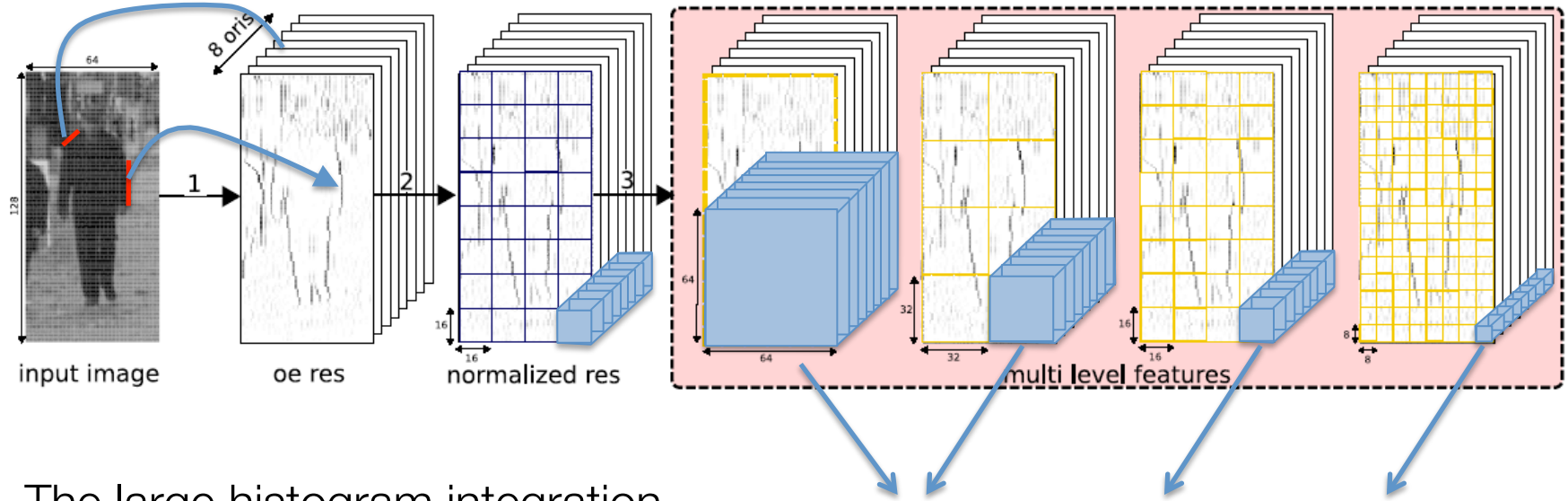
Linear SVM	40% accuracy
Min Kernel (IKSVM)	52% accuracy

Accuracy of Min Kernel vs Linear on **Text classification**

Accuracy Values					
Classification Method	**R8**	**R52**	**20Ng**	**Cade12**	**WebKb**
SVM (Linear Kernel)	0.9666(1)	0.9322(1)	0.8155(0.04)	0.5650(0.05)	0.8796(0.04)
SVM (Intersection Kernel)	0.9693(1)	0.9326(0.8)	0.8115(0.05)	0.5777(0.10)	0.9105(0.04)

Multiscale spHOG features
(Very Similar to Spatial Pyramids)

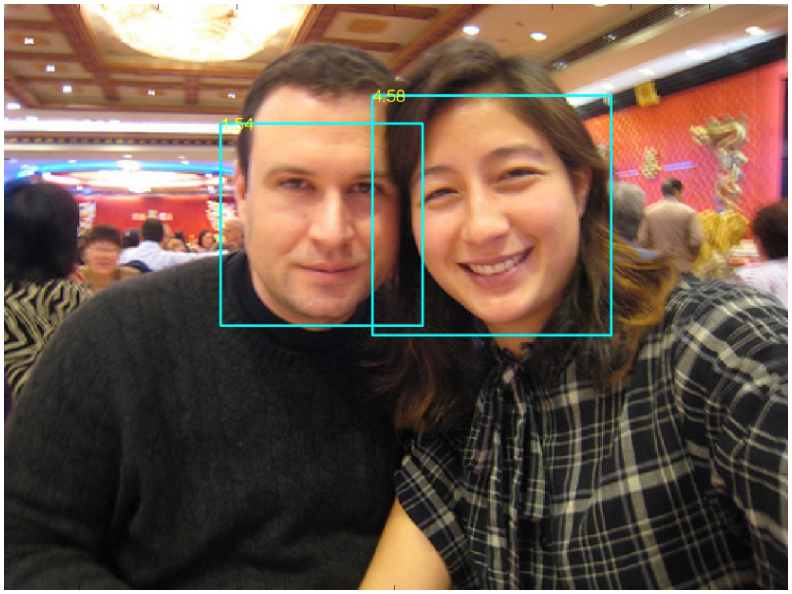
Based on histograms of response to eight orientated edge detections. Non-overlapping windows of integration and fixed size windows for contrast normalization allow efficient computation.

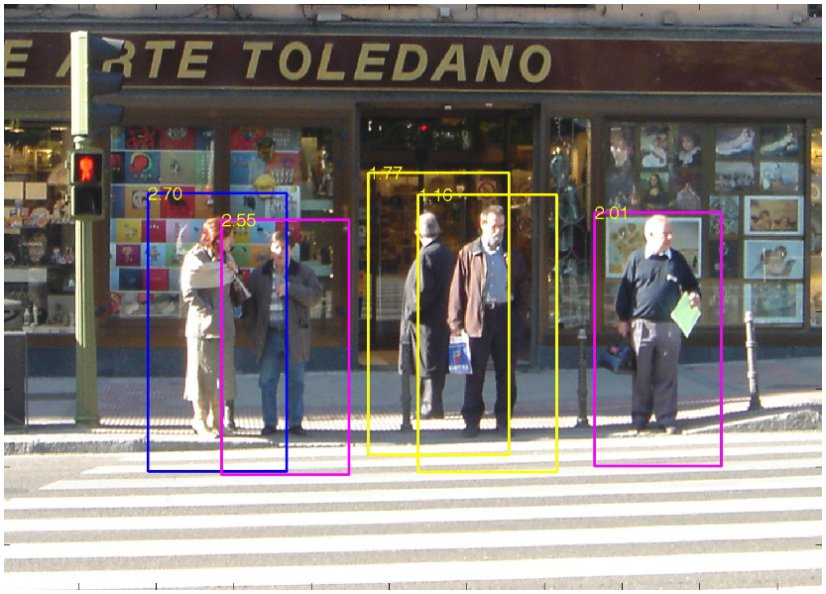


The large histogram integration windows provide variation in individual counts, allowing an additive model to have more flexibility than a linear model

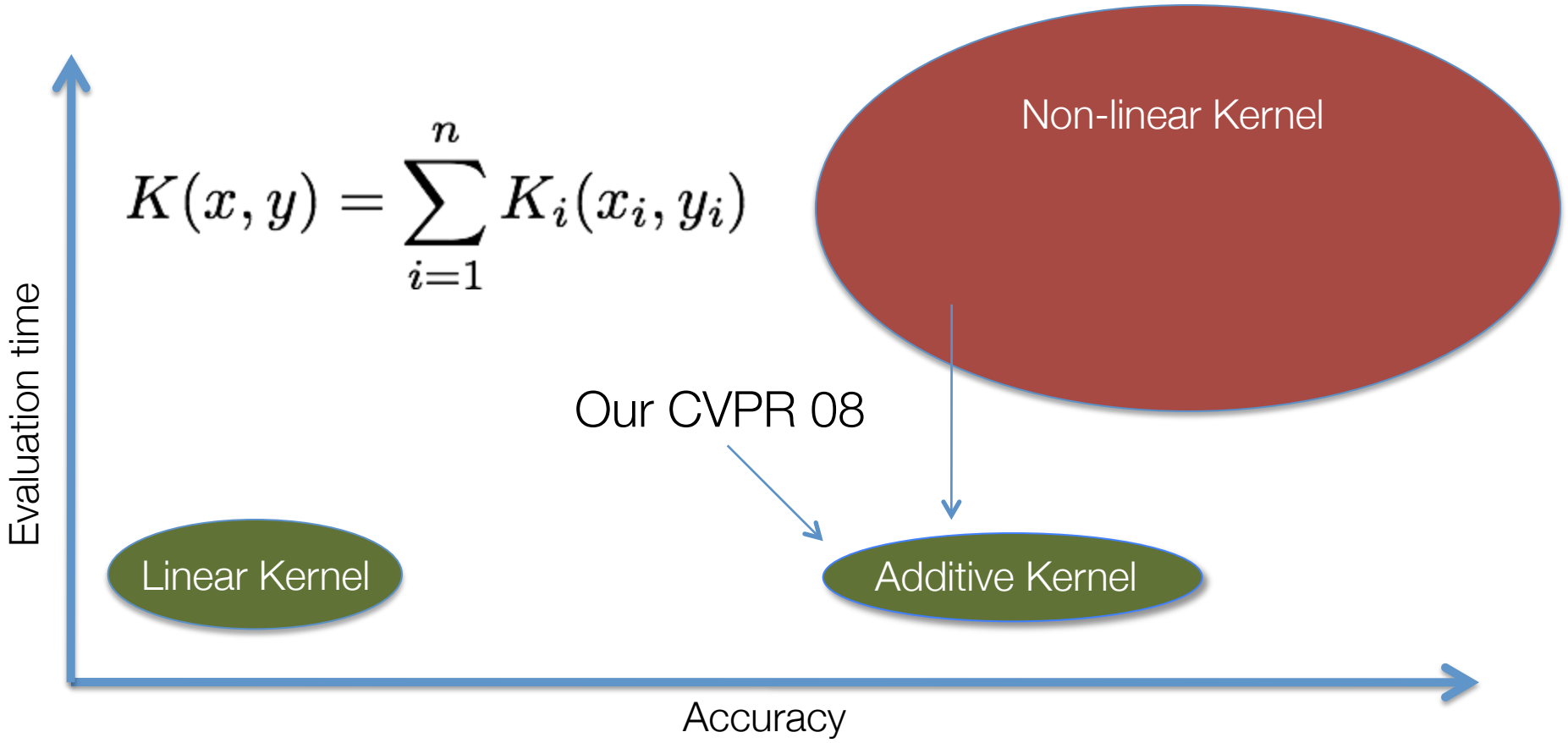
Append histograms counting edges in each region and orientation.

Now we can use Min Kernel for Detection in Seconds Instead of Hours



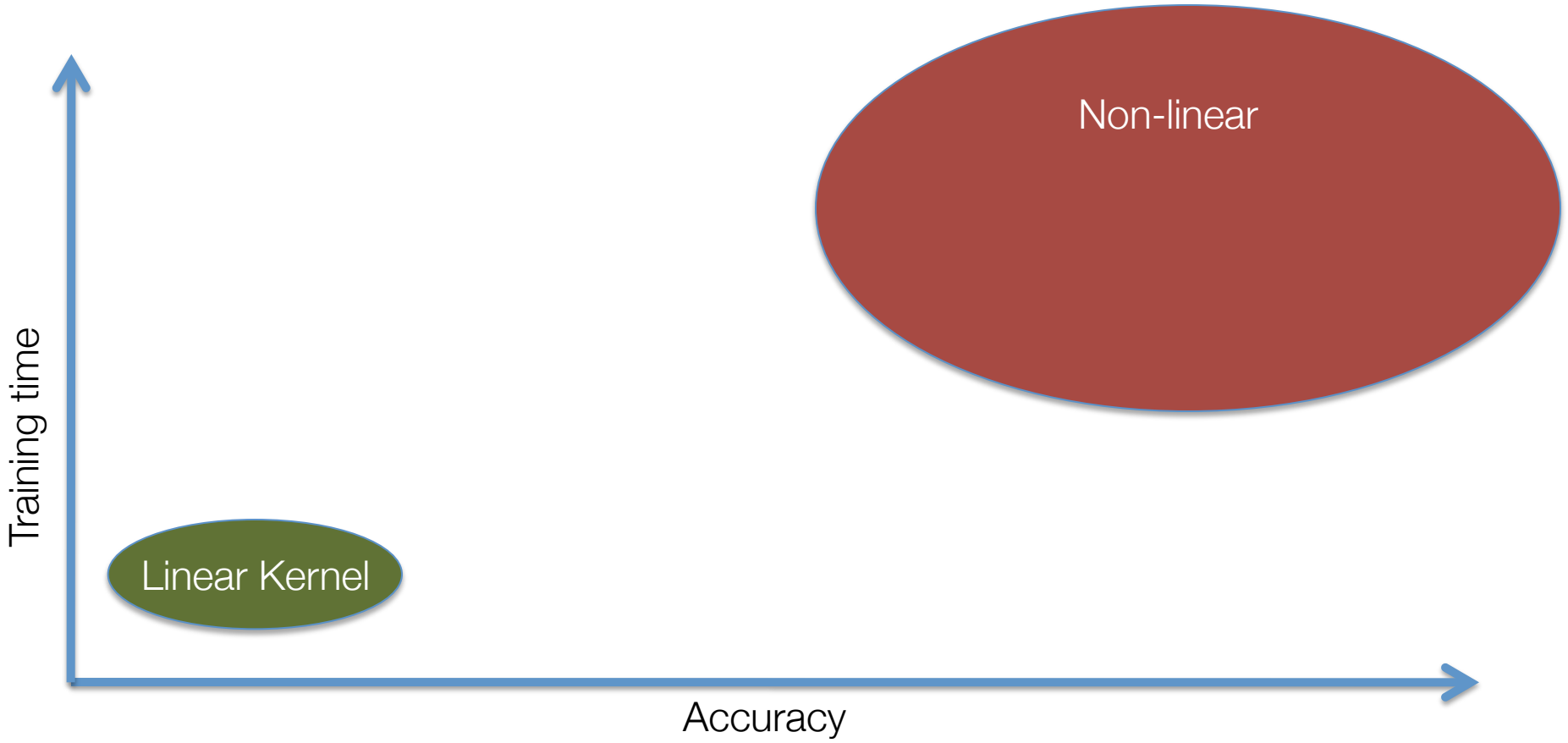


Accuracy vs. Evaluation Time for SVM Classifiers

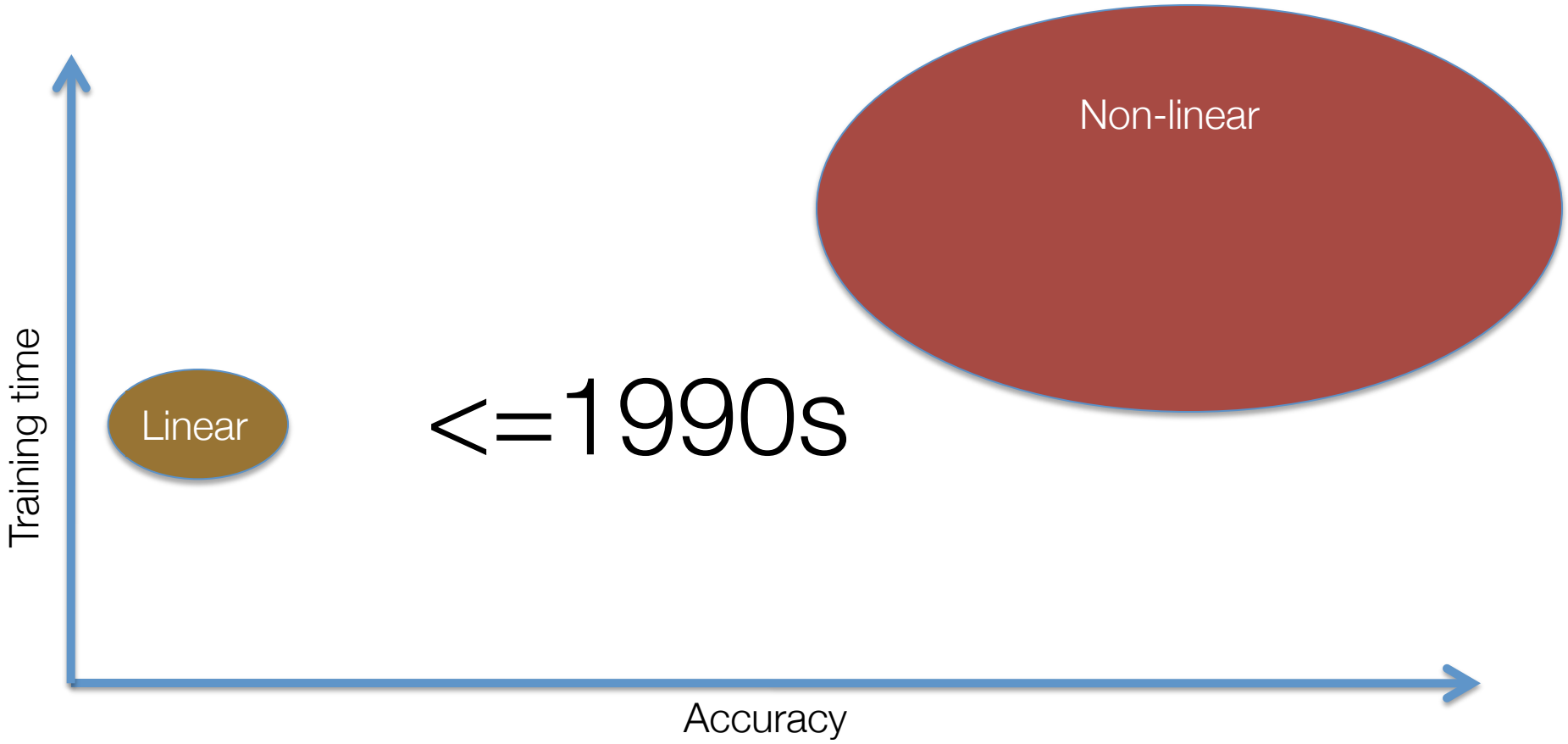


Made it possible to use SVMs with additive kernels for detection.

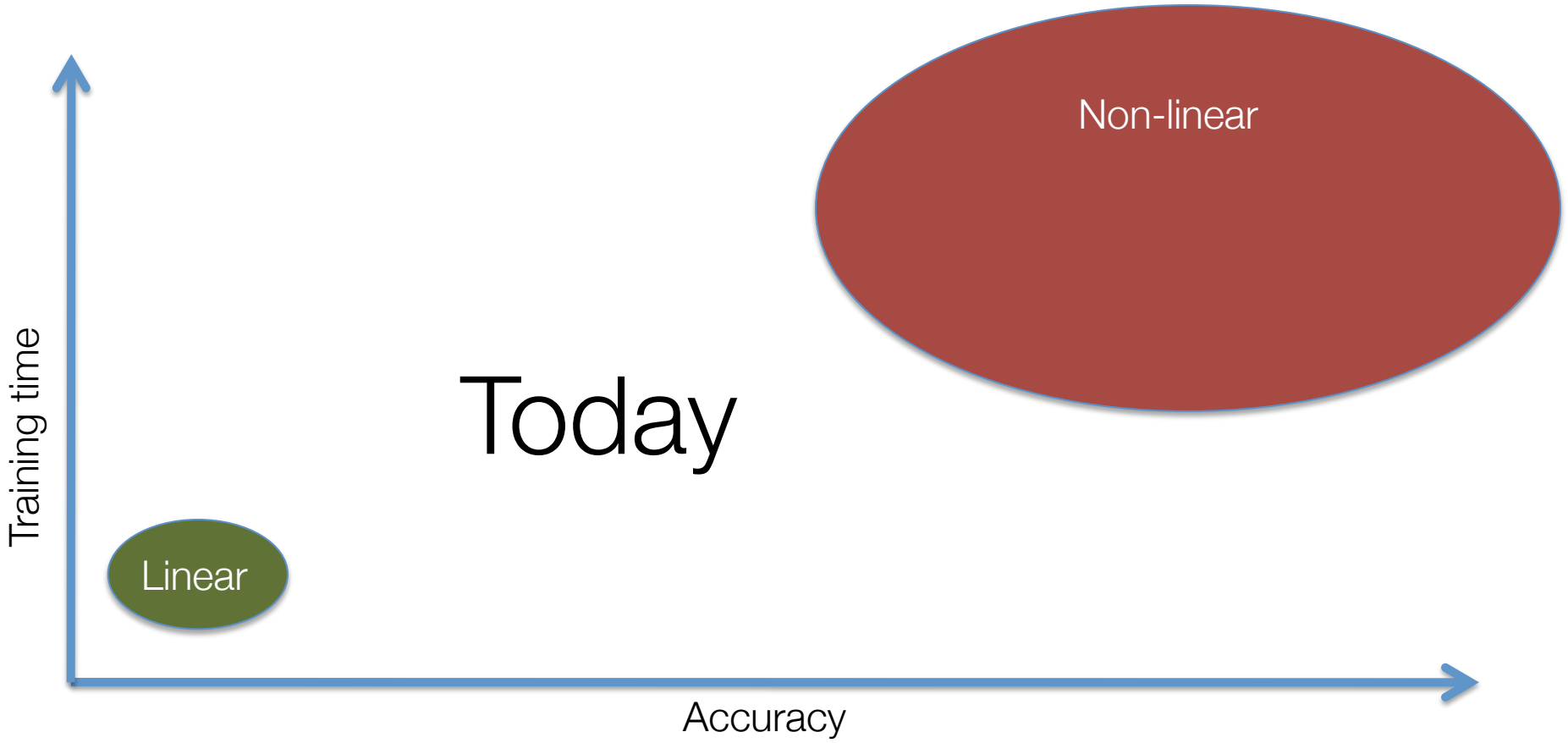
Accuracy vs. Training Time for SVM Classifiers



Accuracy vs. Training Time for SVM Classifiers

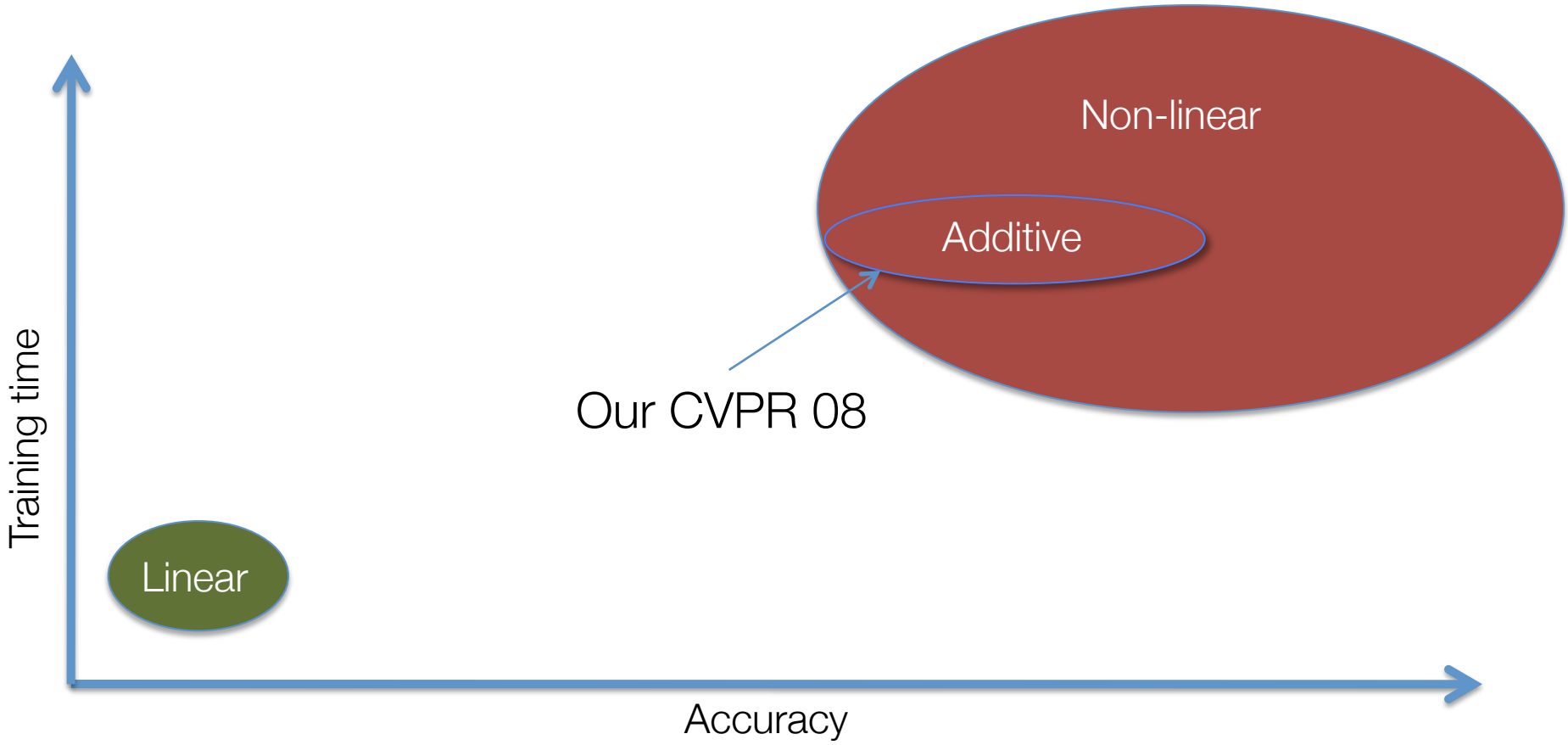


Accuracy vs. Training Time for SVM Classifiers

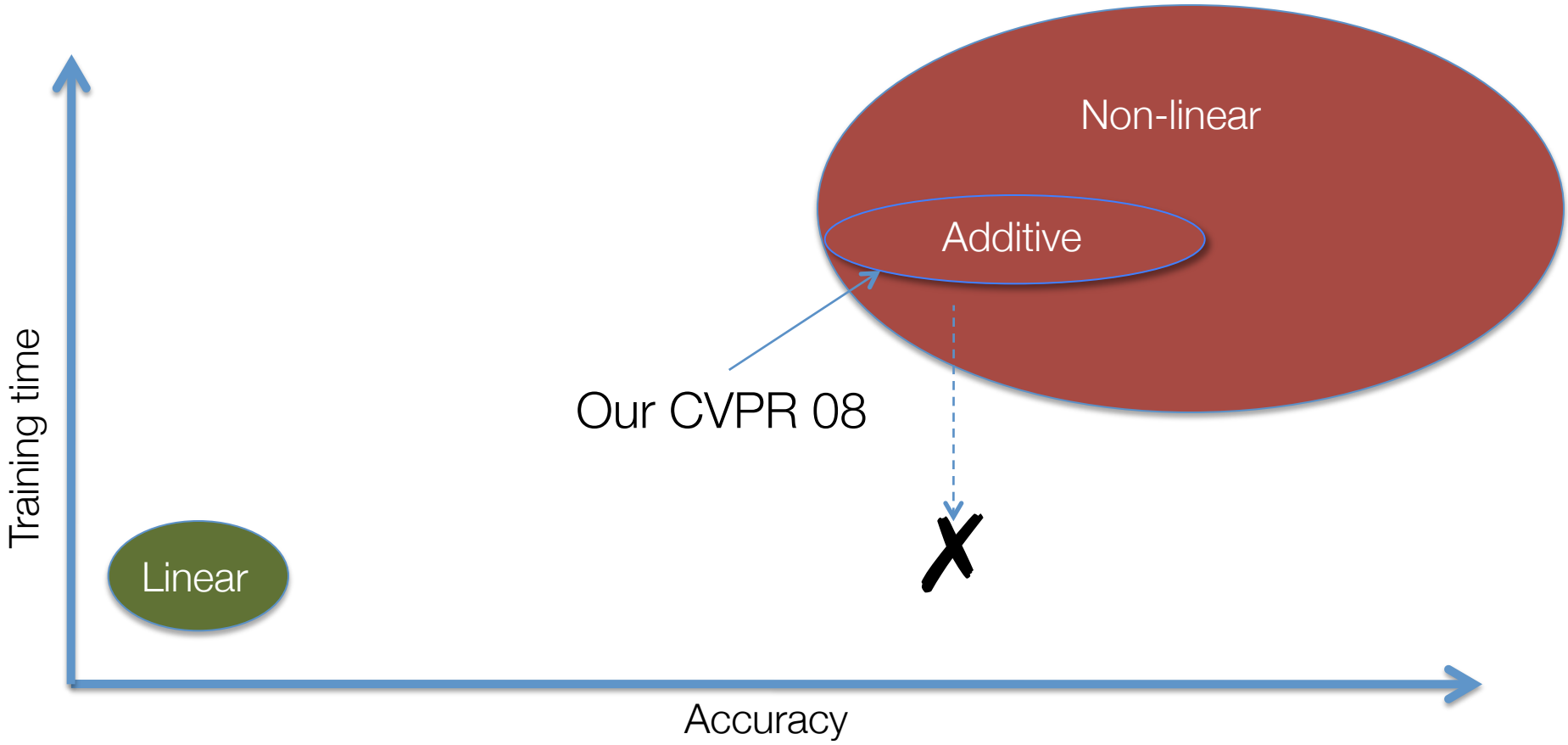


Eg. Cutting Plane, Stoc. Gradient Descent, Dual Coordinate Descent

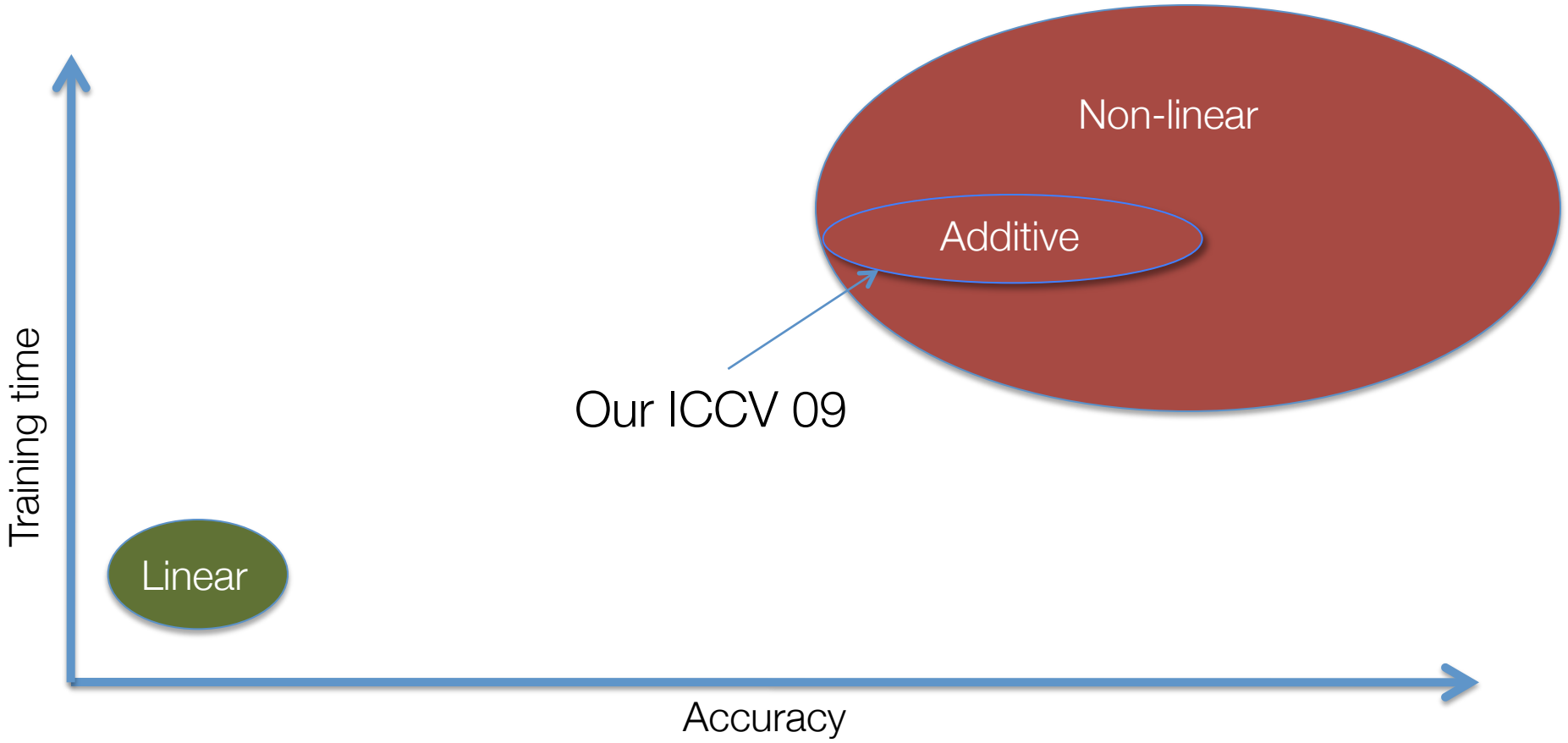
Accuracy vs. Training Time for SVM Classifiers



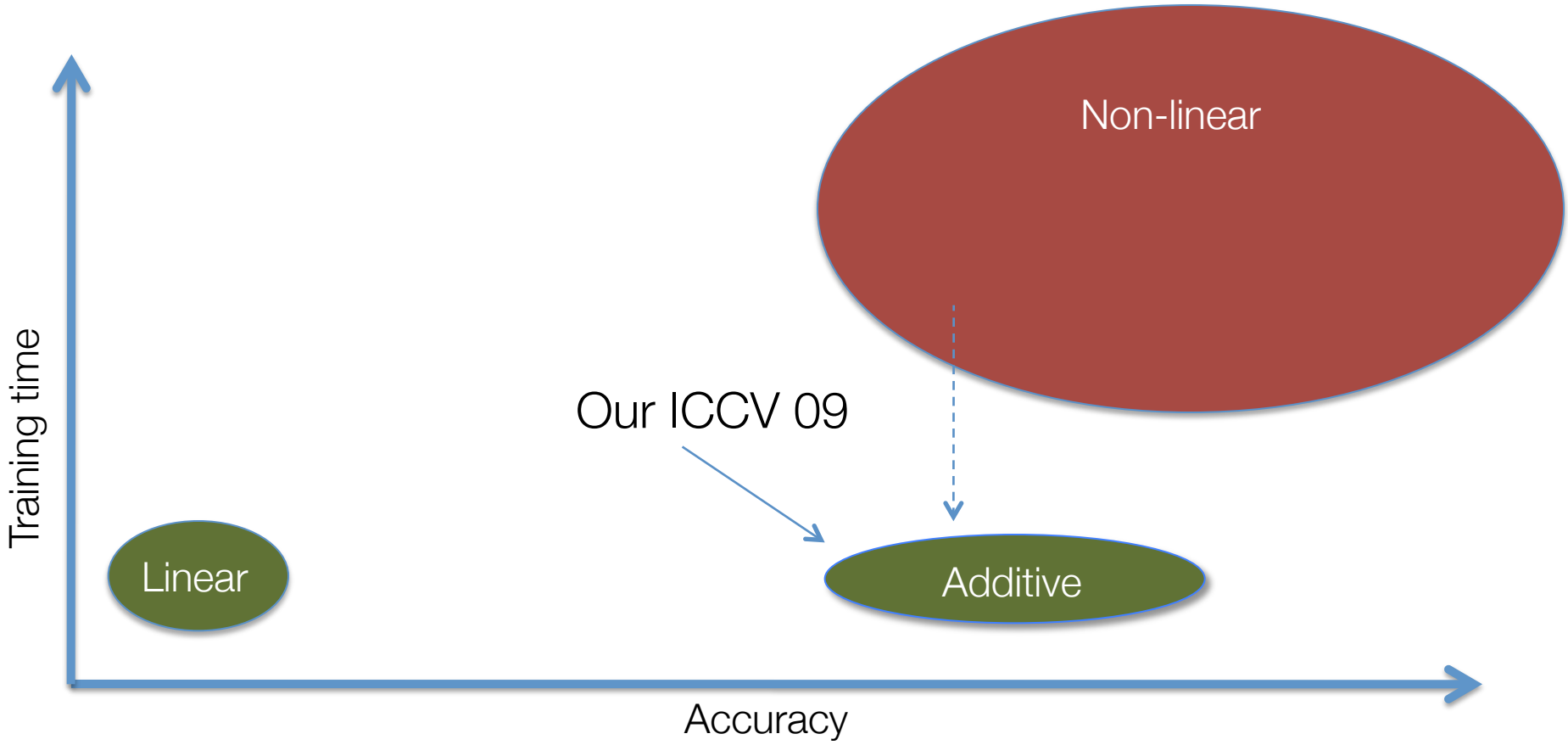
Accuracy vs. Training Time for SVM Classifiers



Accuracy vs. Training Time for SVM Classifiers



Accuracy vs. Training Time for SVM Classifiers



Makes it possible to train additive classifiers very fast.

Summary So Far

- Additive classifiers are widely used and can provide better accuracy than linear
- Our CVPR 08: SVMs with additive kernels are additive classifiers and can be evaluated in O(#dim) -- same as linear.
- Our ICCV 09: additive classifiers can be trained directly as efficiently as linear classifiers using modifications of current state of the art linear training algorithms.

	Additive Kernel SVM	**Our Additive Classifier**	**Linear SVM**
Time	Train 1000 Test 1000	Train 10 Test 1	Train 10 Test 1
Accuracy	95 %	94 %	82 %

Direct Training

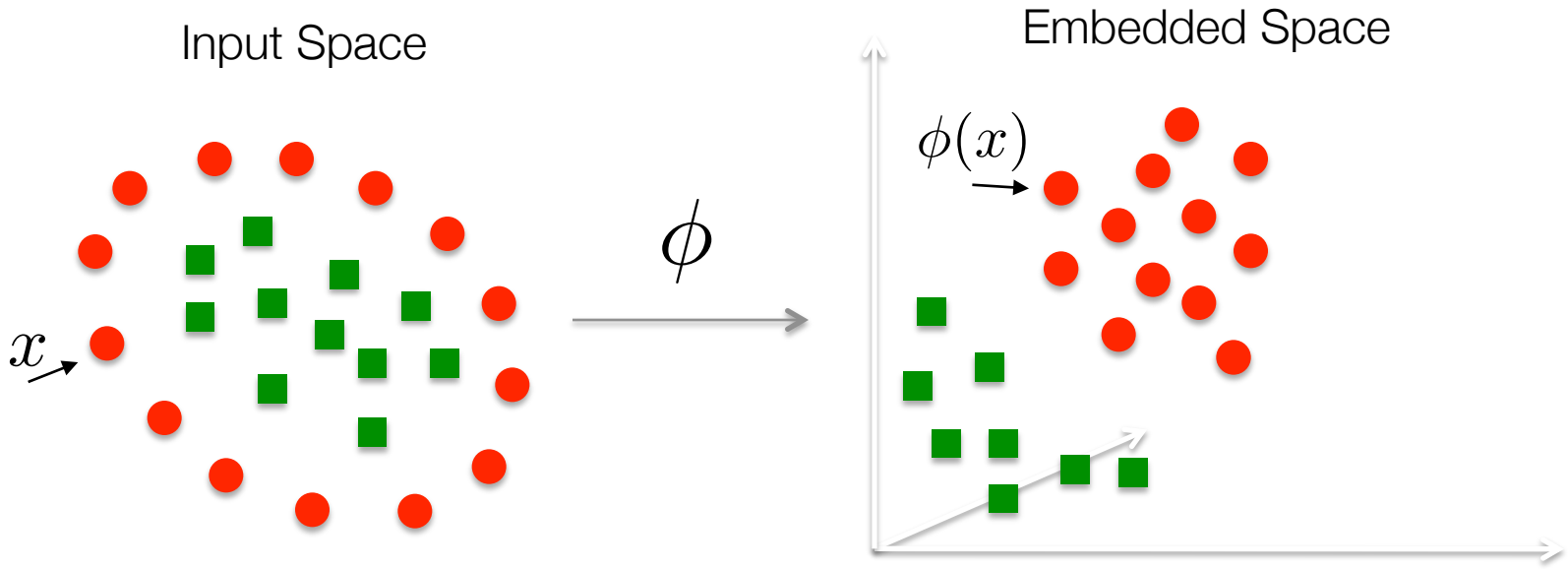
It is possible to directly train additive classifiers without using support vector machines at all.

The formulation is very similar to a linear classifier, with different regularization.

Training uses variants of techniques for fast linear classifier training
(e.g. Pegasos – stochastic subgradient descent with some renormalization)

The key is encoding…

Kernel "Trick" for SVMs



Kernel is inner product in the embedded space $K(x, y) = \phi(x)^T \phi(y)$

Use to represent non-linear boundaries in input space

$$h(x) = \sum_{i=1}^{\#sv} \alpha_i K(x, s_i) + b \qquad h(x) = w^T \phi(x) + b$$

Same Classification Function

Embeddings…

- These embeddings can be high dimensional (even infinite)

- Our approach is based on finite embeddings that **approximate** kernels.

$$\phi(x)^T \phi(y) \sim K(x, y)$$

- We are going to use fast linear classifier training algorithms on the $\phi(x)$ so **sparsity** is important.

Key Idea: Embedding an Additive Kernel

- Additive Kernels are easy to embed –
  just embed each dimension independently

- Linear Embedding for min Kernel for positive integers

$$K_{\min}(x,y) = \sum_{i=1}^{n} \min(x_i, y_i) = U(x)^T U(y)$$

$$\min(3,5) = [1,1,1,0,0]^T [1,1,1,1,1] = 3$$

- For non integers can approximate by quantizing

$$\phi_1(x) = \frac{1}{\sqrt{N}} U(R(Nx)), x \in [0,1]$$

Issues: Embedding Error

- Quantization leads to large errors

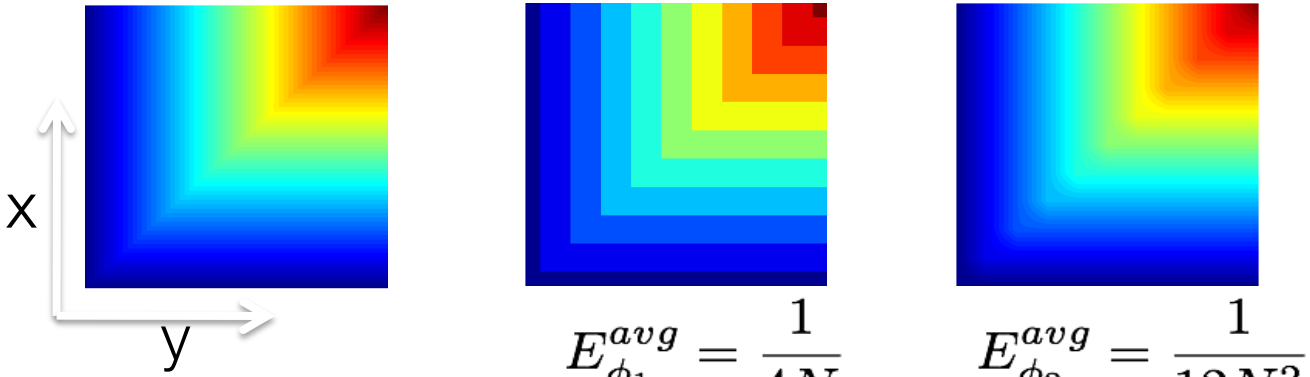
$$x = 3.5; y = 5 \qquad \min(x, x) = 3.5; \min(x, y) = 3.5$$

$$\phi_1(x)^T \phi_1(x) = [1, 1, 1, 0, 0]^T [1, 1, 1, 0, 0] = 3$$
$$\phi_1(x)^T \phi_1(y) = [1, 1, 1, 0, 0]^T [1, 1, 1, 1, 1] = 3$$

- Better encoding

$$\phi_2(x)^T \phi_2(x) = (1, 1, 1, 0.5, 0)^T (1, 1, 1, 0.5, 0) = 3.25$$
$$\phi_2(x)^T \phi_2(y) = (1, 1, 1, 0.5, 0)^T (1, 1, 1, 1, 1) = 3.5$$

$\min(x, y)$ | $\phi_1(x)^T \phi_1(y)$ | $\phi_2(x)^T \phi_2(y)$



$$E^{avg}_{\phi_1} = \frac{1}{4N} \qquad E^{avg}_{\phi_2} = \frac{1}{12N^2}$$

Issues: Sparsity

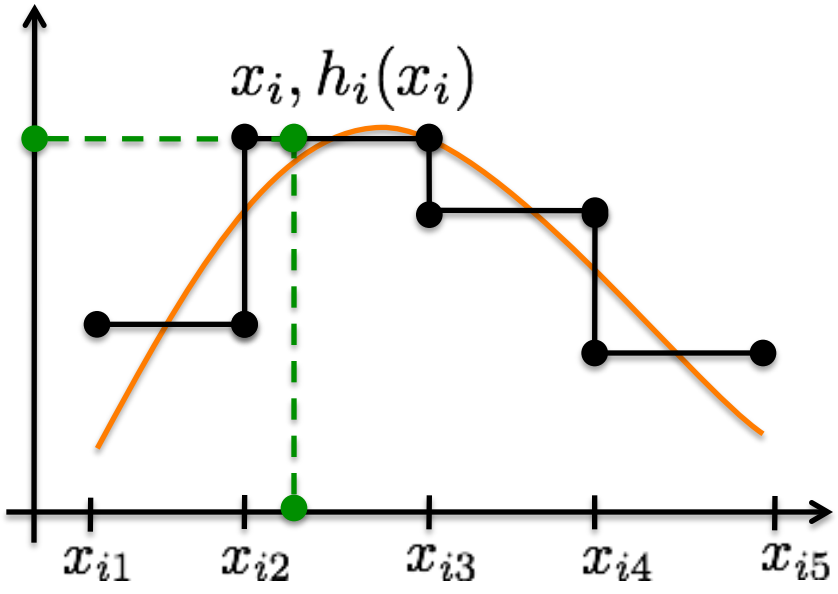
Note:
$\phi_2^s(x) = L\phi_2(x)$

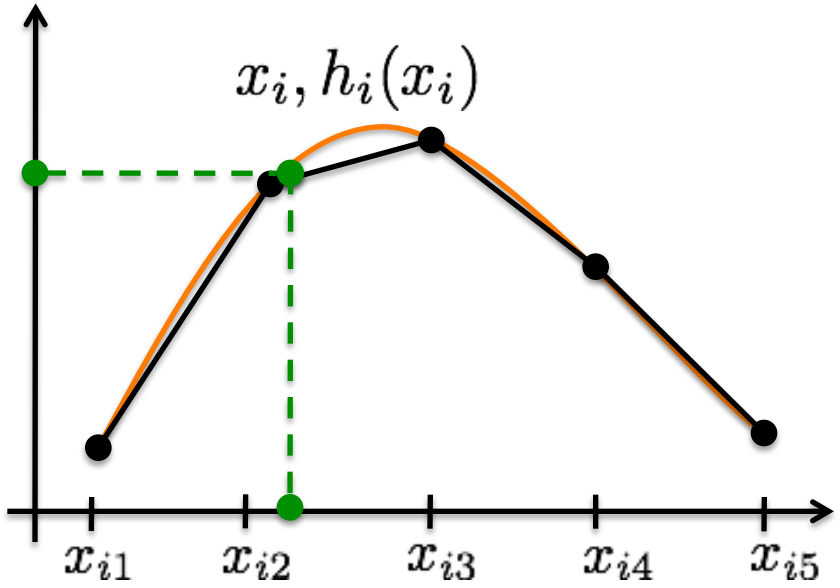
- Represent with sparse values

$$\phi_1(3.5) = (1, 1, 1, 0, 0) \qquad \phi_2(3.5) = (1, 1, 1, 0.5, 0, 0)$$

$$\phi_1^s(3.5) = (0, 0, 1, 0, 0) \qquad \phi_2^s(3.5) = (0, 0, 0.5, 0.5, 0)$$

$$h_i(x_i) = w^T \phi(x) = w^{sT} \phi^s(x)$$





Linear vs. Encoded SVMs

- Linear SVM objective (solve with LIBLINEAR):

$$c(w) = \frac{\lambda}{2} w^T w + \frac{1}{n} \sum_{k=1\ldots n} \max(0, 1 - y^k (w^T x^k)$$

- Encoded SVM:

Standard Solver Impractical

$$c(w) = \frac{\lambda}{2} \hat{w}^T \hat{w} + \frac{1}{n} \sum_{k=1\ldots n} \max(0, 1 - y^k (\hat{w}^T \phi_2(x^k))$$

Custom Solver

$$c(w) = \frac{\lambda}{2} w^{sT} H w^s + \frac{1}{n} \sum_{k=1\ldots n} \max(0, 1 - y^k (w^{sT} \phi_2^s(x^k))$$

Std. Solver Sparse, but "wrong" regularization

$$c(w) = \frac{\lambda}{2} w^{sT} \begin{pmatrix} 1 & -1 & 0 & & \\ -1 & 2 & -1 & & \\ & \cdots & & & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix} w^s + \frac{1}{n} \sum_{k=1\ldots n} \max(0, 1 - y^k (w^{sT} \phi_2^s(x^k))$$

Encourages smooth functions
Closely approximates min-kernel SVM
Custom solver : PWLSGD (see paper)

Linear vs. Encoded SVMs

- Linear SVM objective (solve with LIBLINEAR):

$$c(w) = \frac{\lambda}{2} w^T w + \frac{1}{n} \sum_{k=1 \ldots n} max(0, 1 - y^k (w^T x^k))$$

- Encoded SVM objective (solve with LIBLINEAR) :

$$c(w) = \frac{\lambda}{2} w^T w + \frac{1}{n} \sum_{k=1 \ldots n} max(0, 1 - y^k (w^T \phi_1^s(x)^k))$$

$$c(w) = \frac{\lambda}{2} w^T w + \frac{1}{n} \sum_{k=1 \ldots n} max(0, 1 - y^k (w^T \phi_2^s(x)^k))$$

Very Similar Except for Encoding

Linear

$$\min_w \frac{\lambda}{2} w'w + \frac{1}{m} \sum_i \ell\left(w; (x_i, y_i)\right)$$

Piecewise linear

$$\min_w \frac{\lambda}{2} \hat{w}' H \hat{w} + \frac{1}{m} \sum_i \ell\left(\hat{w}; (\hat{x}_i, y_i)\right)$$

Pegasos: Shalev-Schwartz, Singer, Srebro ICML 2007

$O\left(\frac{d}{\lambda\epsilon}\right)$ for ϵ accuracy

INPUT: S, λ, T, k

INITIALIZE: Choose $\mathbf{w}_1$ s.t. $\|\mathbf{w}_1\| \leq 1/\sqrt{\lambda}$

FOR $t = 1, 2, \ldots, T$

Choose $A_t \subseteq S$, where $|A_t| = k$

Set $A_t^+ = \{(\mathbf{x}, y) \in A_t : y \langle \mathbf{w}_t, \mathbf{x} \rangle < 1\}$

Set $\eta_t = \frac{1}{\lambda t}$

Set $\mathbf{w}_{t+\frac{1}{2}} = (1 - \eta_t \lambda)\mathbf{w}_t + \frac{\eta_t}{k} \sum_{(\mathbf{x},y)\in A_t^+} y\,\mathbf{x}$

Set $\mathbf{w}_{t+1} = \min\left\{1, \frac{1/\sqrt{\lambda}}{\|\mathbf{w}_{t+\frac{1}{2}}\|}\right\} \mathbf{w}_{t+\frac{1}{2}}$

OUTPUT: $\mathbf{w}_{T+1}$

Modified Pegasos for Additive Models

$O\left(\frac{d}{\lambda\epsilon}\right)$ for ϵ accuracy

$\|w\| \rightarrow \sqrt{w'Hw}$

INPUT: S, λ, T, k

INITIALIZE: Choose $\mathbf{w}_1$ s.t. $\sqrt{w_1' H w_1} \leq 1/\sqrt{\lambda}$

FOR $t = 1, 2, \ldots, T$

Choose $A_t \subseteq S$, where $|A_t| = k$

Set $A_t^+ = \{(\mathbf{x}, y) \in A_t : y \langle \mathbf{w}_t, \mathbf{x} \rangle < 1\}$

Set $\eta_t = \frac{1}{\lambda t}$

Set $\mathbf{w}_{t+\frac{1}{2}} = (1 - \eta_t \lambda H)\mathbf{w}_t + \frac{\eta_t}{k} \sum_{(\mathbf{x},y) \in A_t^+} y\,\mathbf{x}$

Set $\mathbf{w}_{t+1} = \min\left\{1, \frac{1/\sqrt{\lambda}}{\sqrt{w'_{t+\frac{1}{2}} H w_{t+\frac{1}{2}}}}\right\} \mathbf{w}_{t+\frac{1}{2}}$

OUTPUT: $\mathbf{w}_{T+1}$

Modified Pegasos

$O\left(\frac{d}{\lambda\epsilon}\right)$ for ϵ accuracy

$\|w\| \rightarrow \sqrt{w'Hw}$

INPUT: S, λ, T, k

INITIALIZE: Choose $\mathbf{w}_1$ s.t. $\sqrt{w_1' H w_1} \leq 1/\sqrt{\lambda}$

FOR $t = 1, 2, \ldots, T$

Choose $A_t \subseteq S$, where $|A_t| = k$

Set $A_t^+ = \{(\mathbf{x}, y) \in A_t : y \langle \mathbf{w}_t, \mathbf{x} \rangle < 1\}$

Set $\eta_t = \frac{1}{\lambda t}$

Set $\mathbf{w}_{t+\frac{1}{2}} = (1 - \eta_t \lambda H)\mathbf{w}_t + \frac{\eta_t}{k} \sum_{(\mathbf{x},y) \in A_t^+} y\,\mathbf{x}$

Set $\mathbf{w}_{t+1} = \min\left\{1, \frac{1/\sqrt{\lambda}}{\sqrt{w'_{t+\frac{1}{2}} H w_{t+\frac{1}{2}}}}\right\} \mathbf{w}_{t+\frac{1}{2}}$

OUTPUT: $\mathbf{w}_{T+1}$

Figure 1. The Pegasos Algorithm.

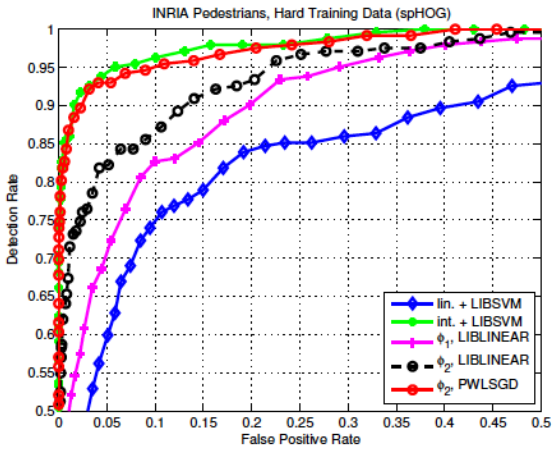
w and x are large but sparse, so we can get computation scingal with # non zeros.

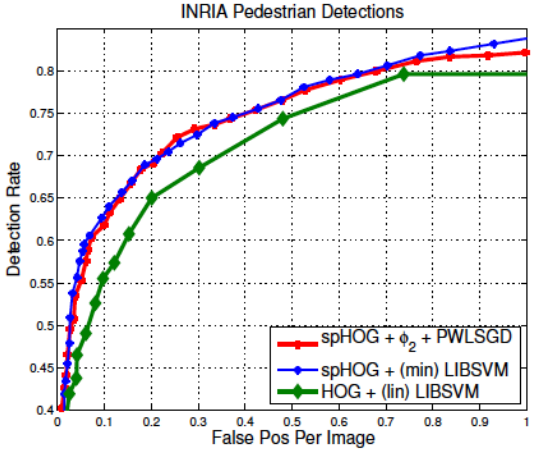
Training Time

Maji, Berg ICCV

Encoding	Training Algorithm	Training Time (HOG)	Training Time (spHOG)
identity	LIBLINEAR	-	20.12s
identity	LIBSVM (lin. lernel)	>180 min	140 min
identity	LIBSVM (int. kernel)	>180 min	148 min
snow=ϕ_1	LIBLINEAR	35.52s	121.81s
ϕ_2	LIBLINEAR	22.45s	26.76s
ϕ_2	PWLSGD	99.85s	76.12s

Table 3. **(HOG)** 47, 327 features of 3780 dimension. Encoding Time 87.22s. Dalal and Triggs use a modified SVMLIGHT which is faster than LIBSVM, but still takes several minutes to train, slower than our PWLSGD on ϕ_2 encoding which produces both better classification using either HOG or spHOG (below) and better detection (Fig. 2 using spHOG). (LIBLINEAR failed to train on the raw HOG data) **(spHOG)** : Training 38, 862 features of 2268 dimension using PWLSGD on the ϕ_2 encoding takes only about 1% of the time taken to train an intersection kernel SVM using LIBSVM, and performs as well for classification (see below).





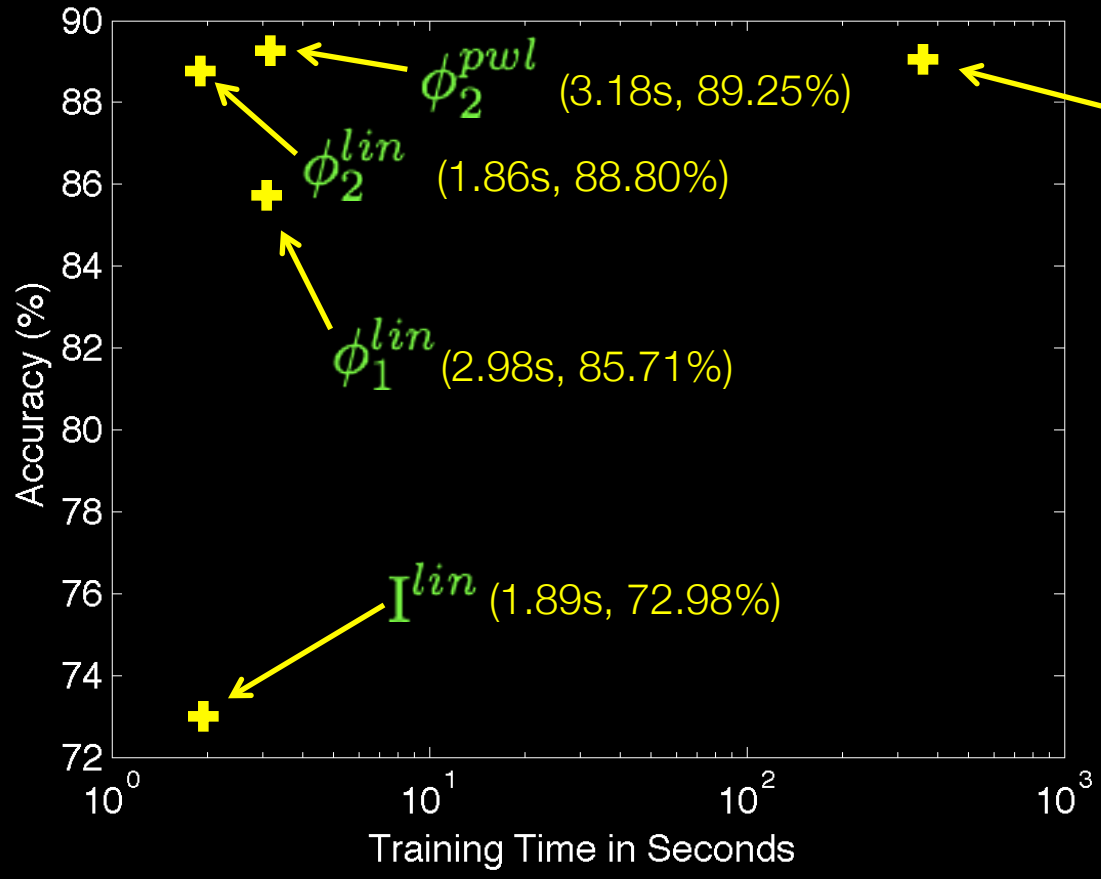
Experiments

- “Small” Scale: Caltech 101 (Fei-Fei, et.al.)

- “Medium” Scale: DC Pedestrians (Munder & Gavrila)

- “Large” Scale : INRIA Pedestrians (Dalal & Triggs)

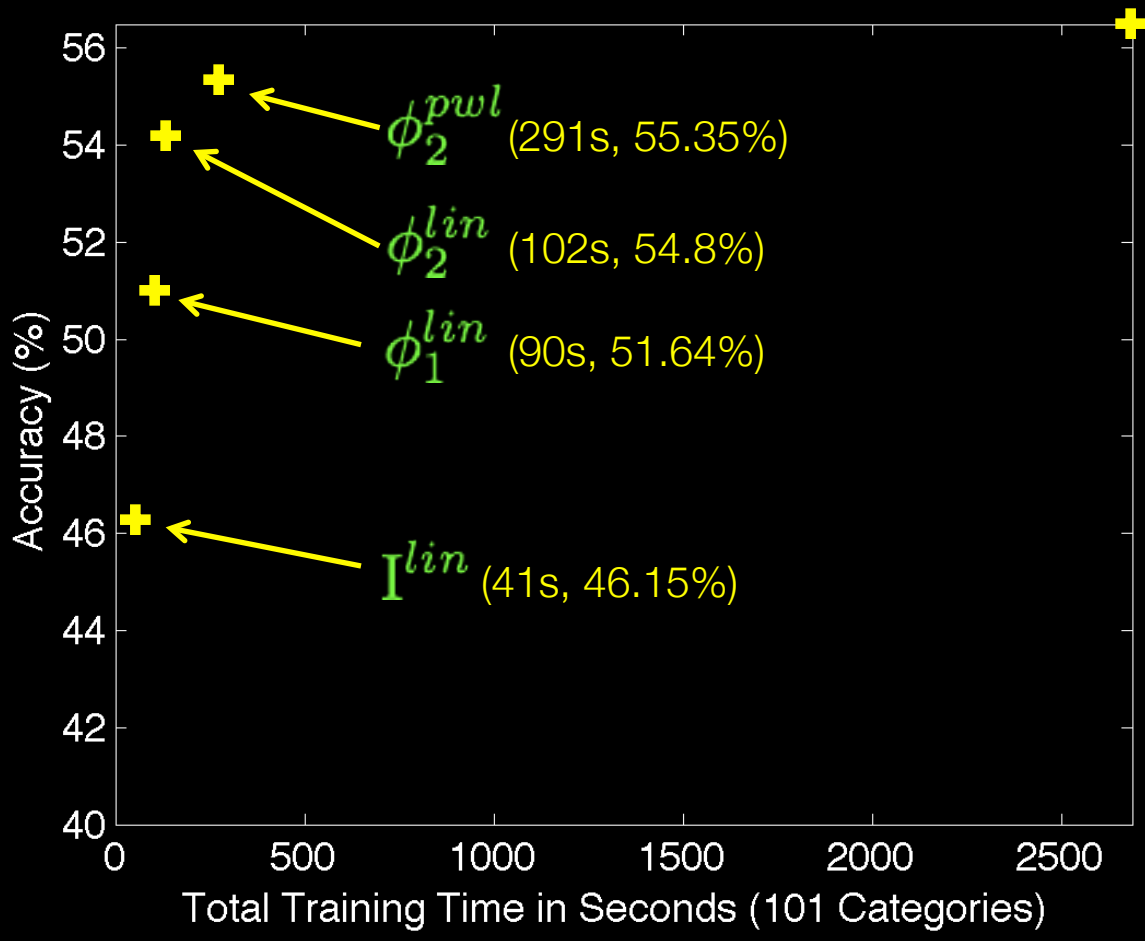
Experiment : DC Pedestrians



100x faster
training time ~ linear SVM
accuracy ~ kernel SVM

20,000 features, 656 dimensional
100 bins for encoding
6-fold cross validation

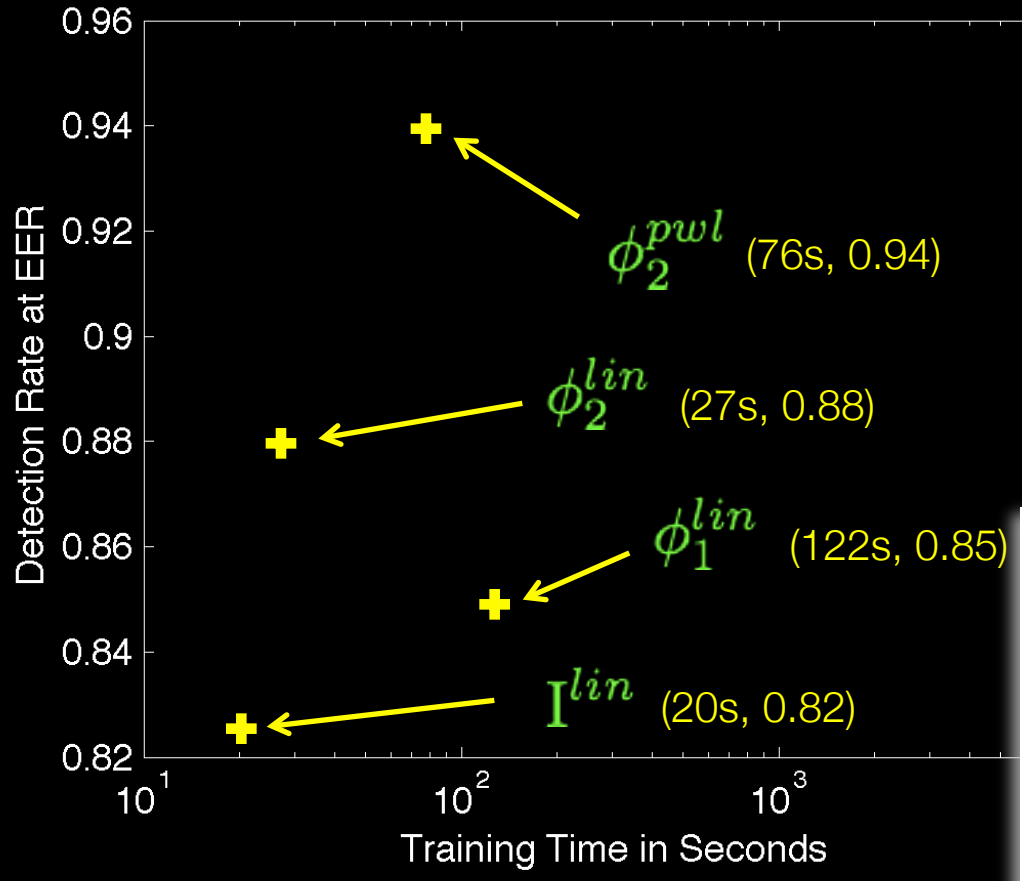
Experiment : Caltech 101



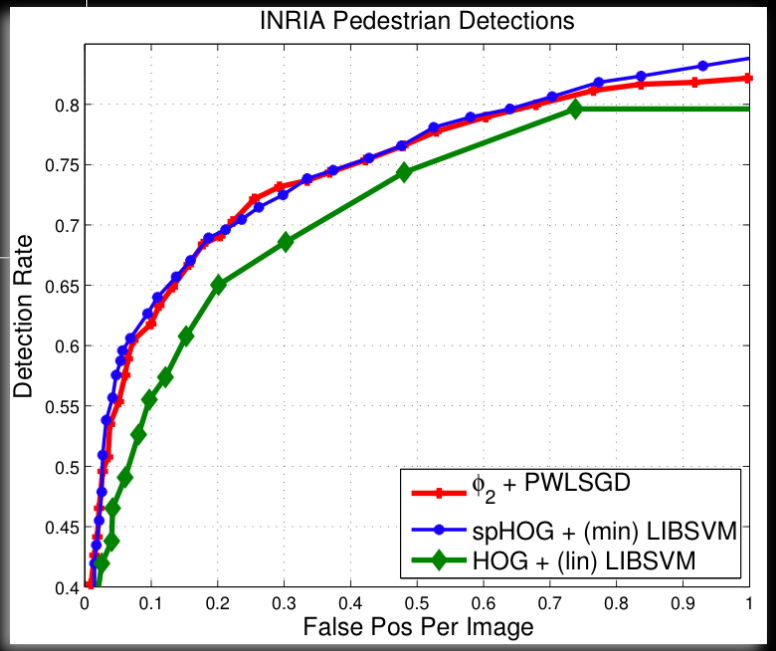
10x faster
Small loss in accuracy

30 training examples per category
100 bins for encoding
Pyramid HOG + Spatial Pyramid Match Kernel

Experiment : INRIA Pedestrians



300x faster
training time ~ linear SVM
accuracy ~ kernel SVM
trains the detector in < 2 mins



SPHOG: 39,000 features, 2268 dimensional
100 bins for encoding
Cross Validation Plots

Take Home Messages

- Additive models are practical for large scale data
- Better accuracy than linear on vision data
- Everyone should try: see code on our websites
  - Fast IKSVM from CVPR'08, Encoded SVMs, etc

Neural Networks

$$S(Lx) = y$$

Fixed non-linearities

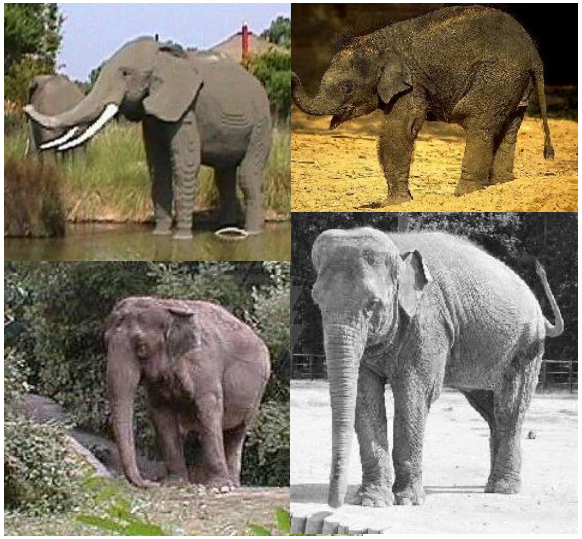
Additive Classifier

$$L(F(x)) = y$$

Learned

Given x Data
y Label

101 Object Categories



How many object categories?

Biederman 1987

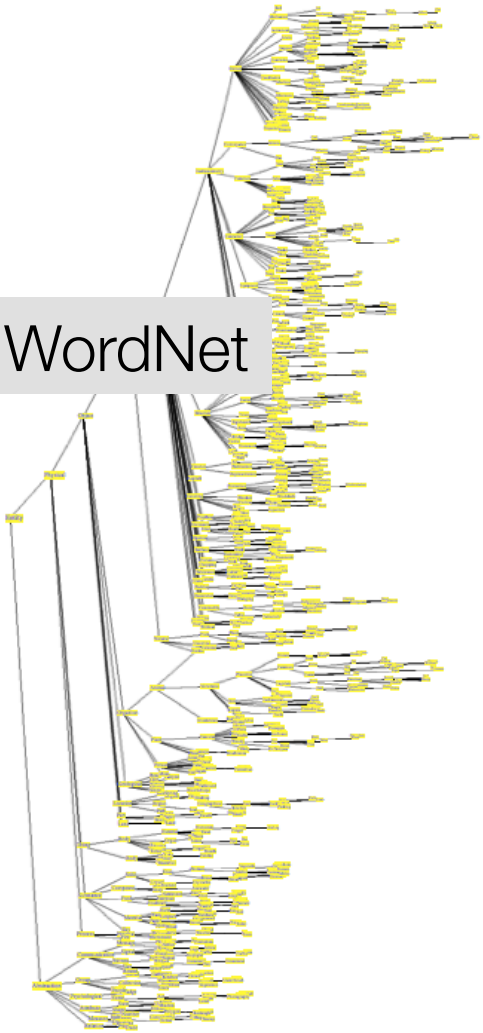
How many object categories?

Biederman 1987

10,000 – 30,000!

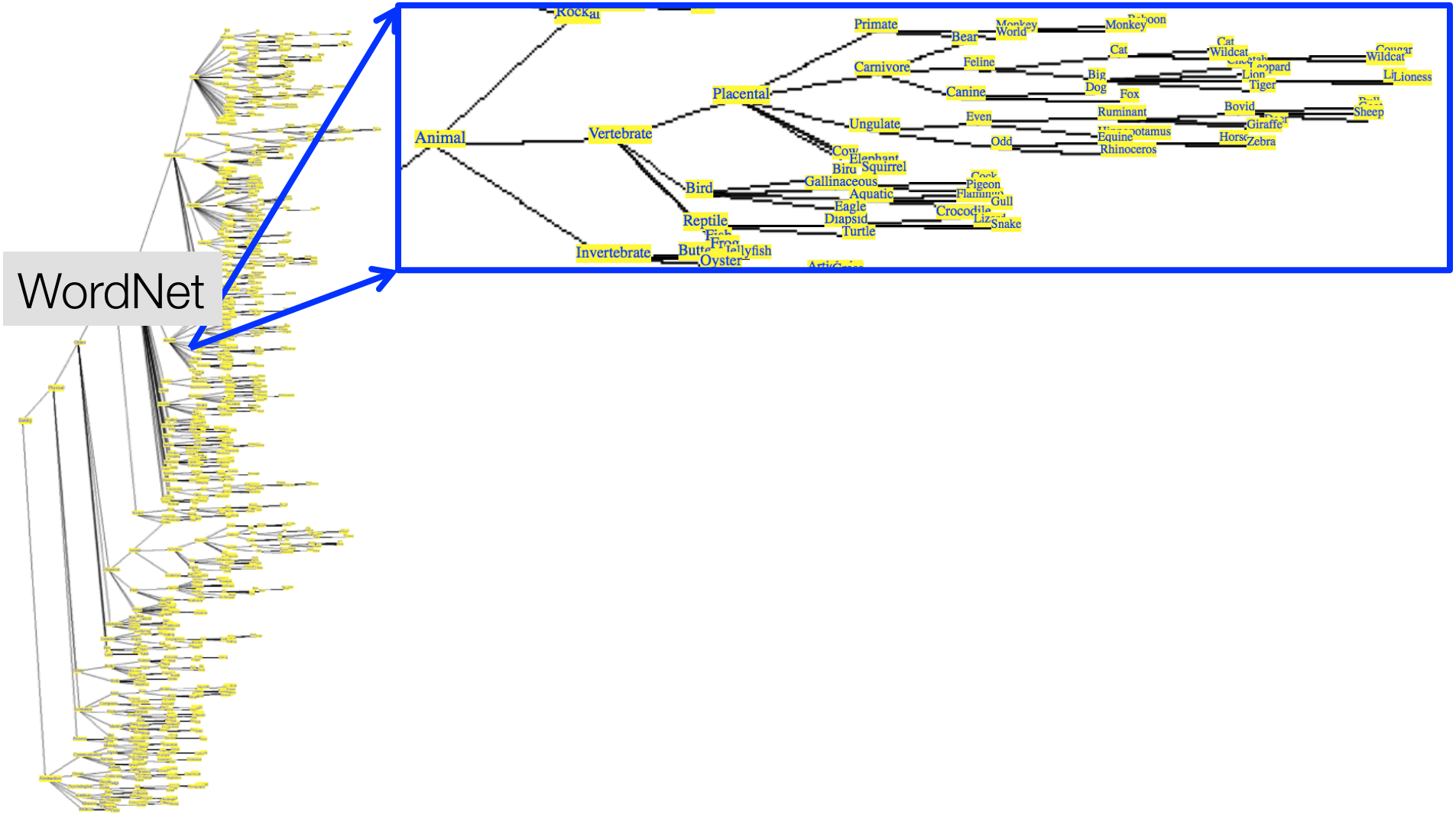
Large Scale Recognition

Large Scale Recognition

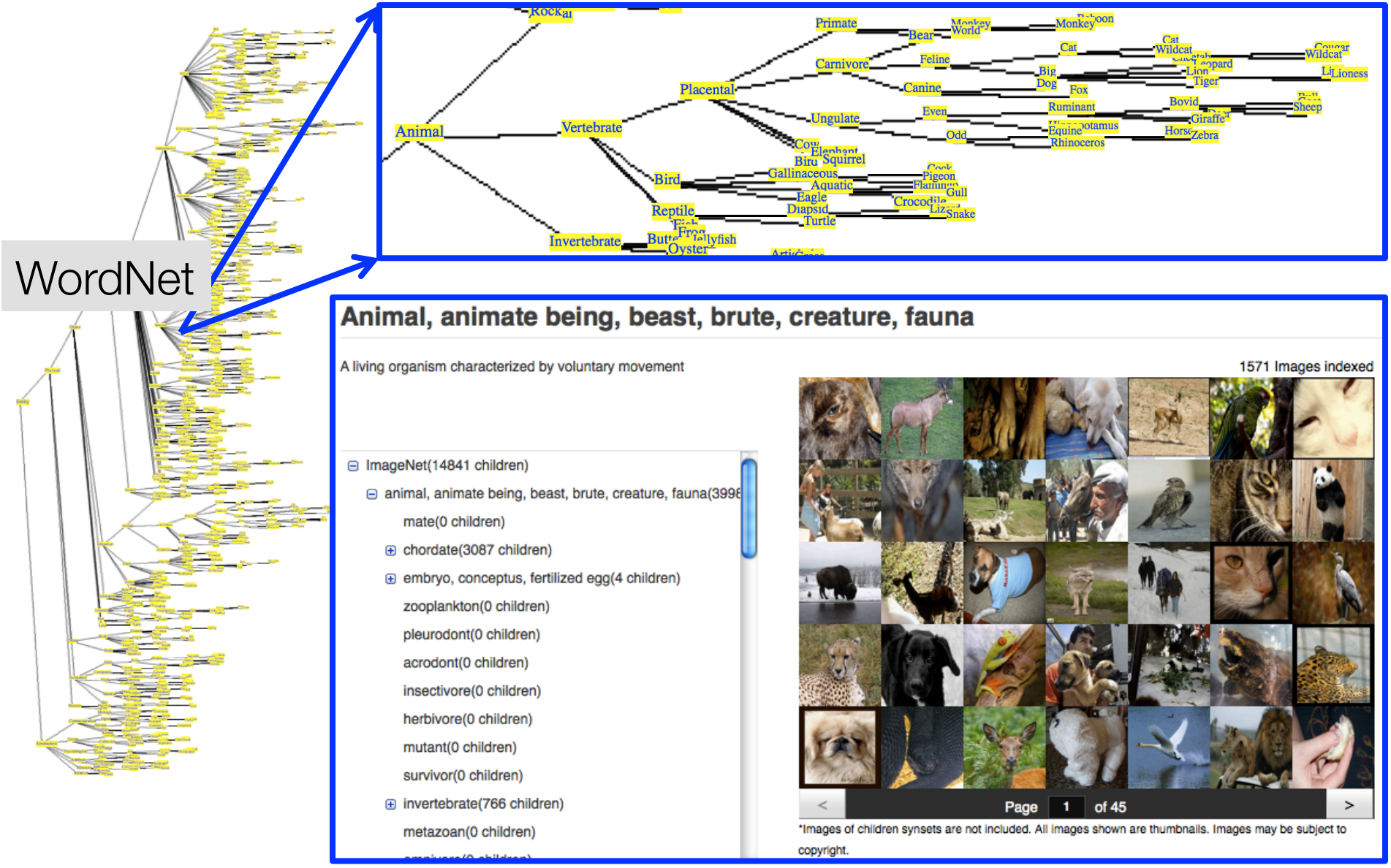


G.A. Miller 1995

Large Scale Recognition



Large Scale Recognition



ImageNet: 10 million images, 10 thousand categories Fei-Fei Li 2009

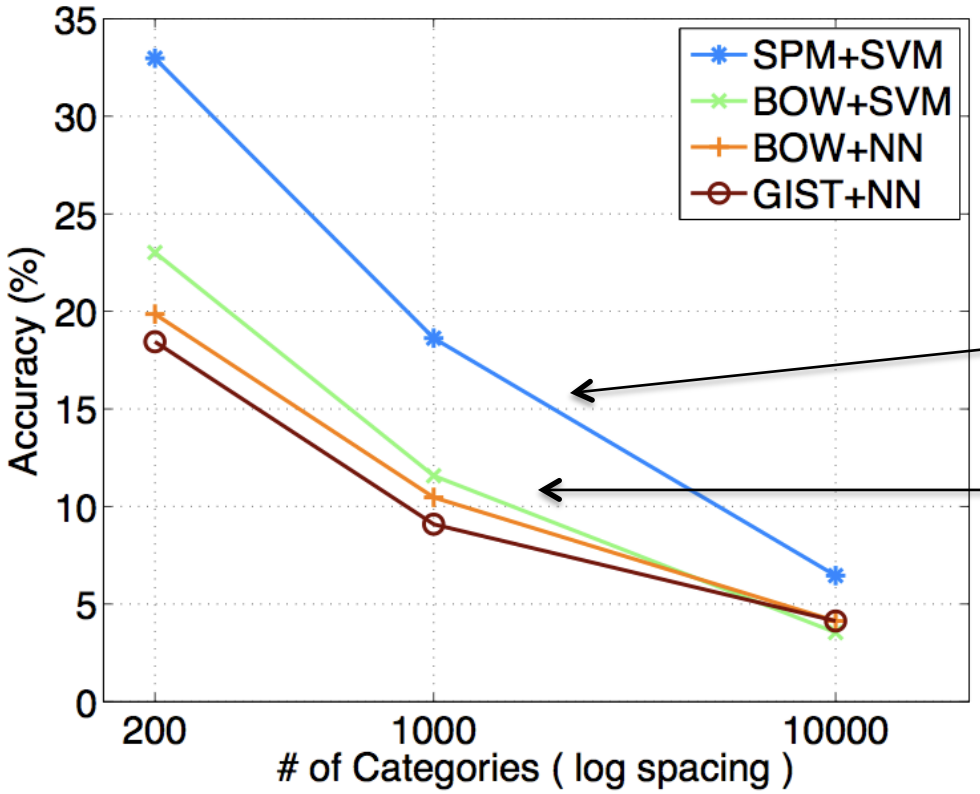
Made Possible by Amazon's Mechanical Turk Service

Costs pennies for a real person to label the content of images!

Radically different cost structure for collecting datasets.

Experiments on ImageNet

Deng, Berg, Li, Fei-Fei in submission



Additive

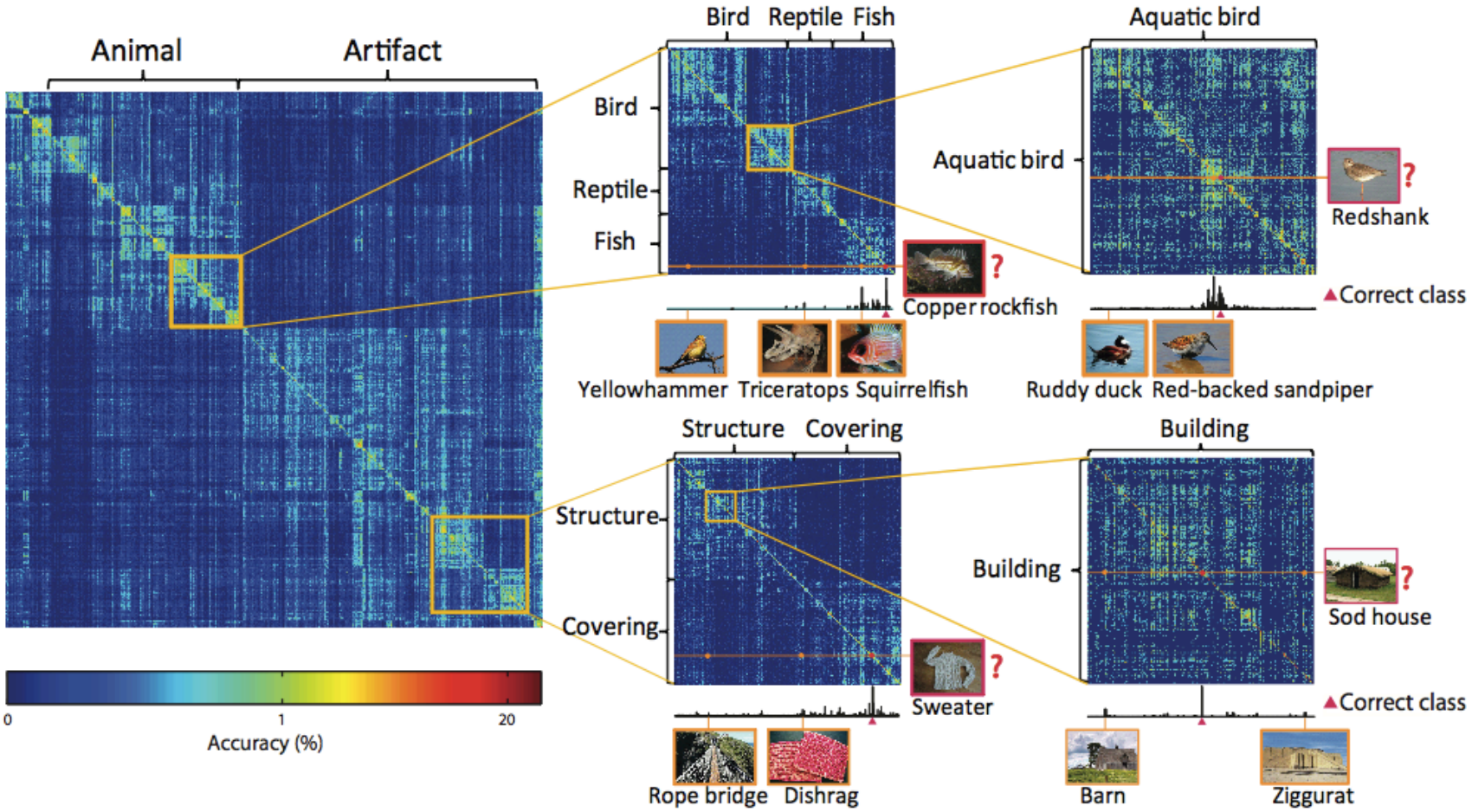
Linear

Scaling not so bad…

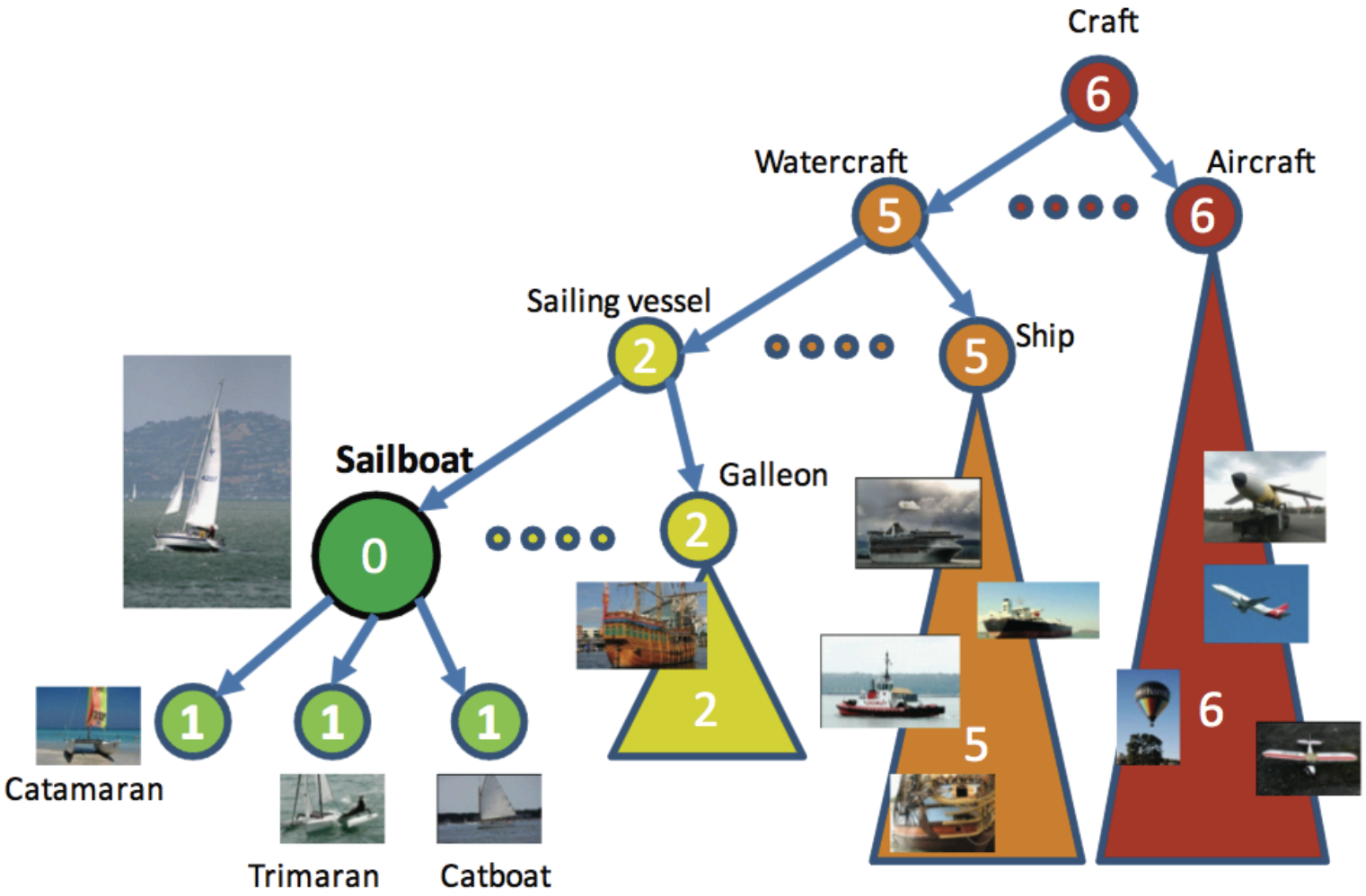
Table 1. Computational requirements of multi-class classification methods for BOW+SVM on ImageNet10K

Method	# of classifiers	T_{train}	T_{test}	Memory	Parallelism
1-vs-all	10,000	10,000 hours	16 hours	7GB	Yes
1-vs-1	50,000,000	10,000 hours	80,000 hours	3MB	Yes
Crammer & Singer	1	unknown/large		7GB	No
Nearest Neighbors (linear scan)	N/A	0	11,000 hours	7GB	Yes

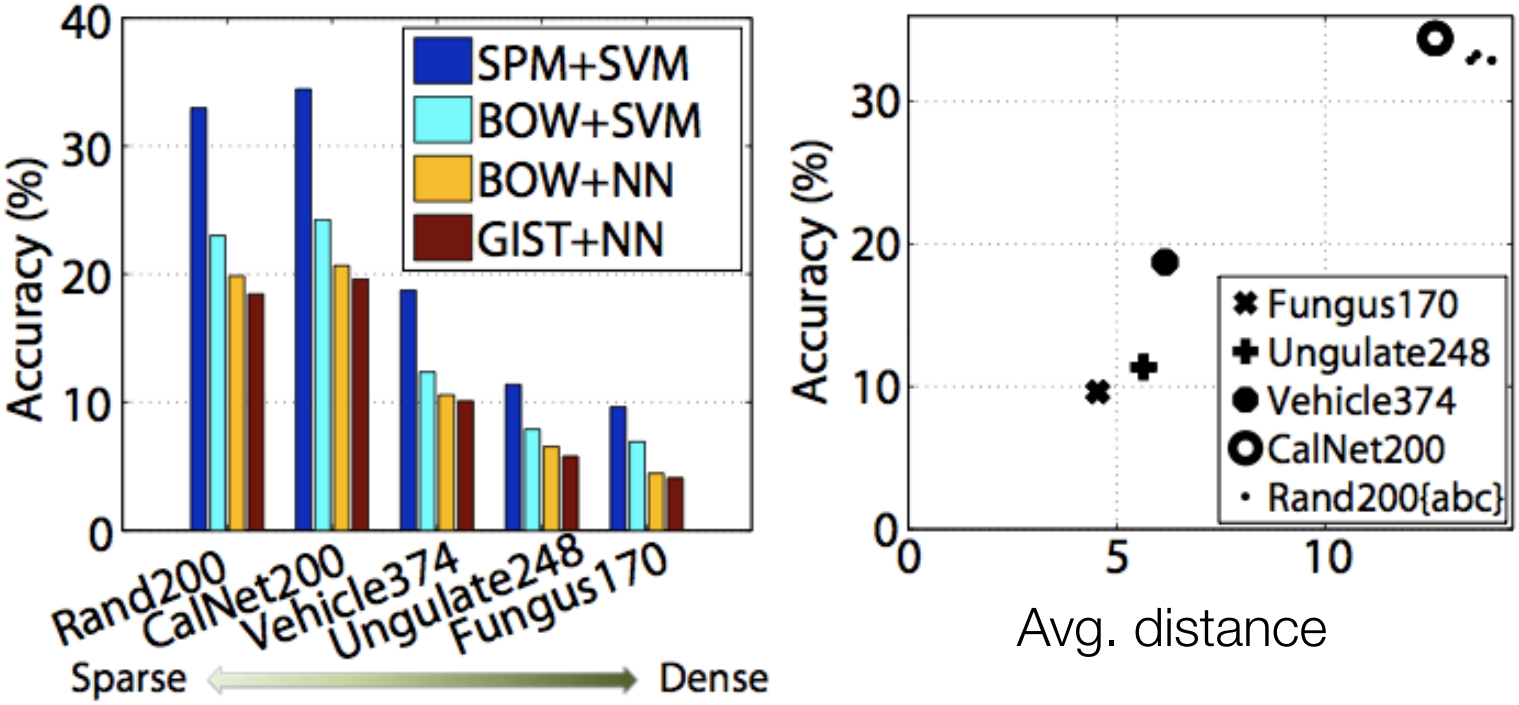
Experiments on ImageNet



ImageNet Hierarchy From WordNet

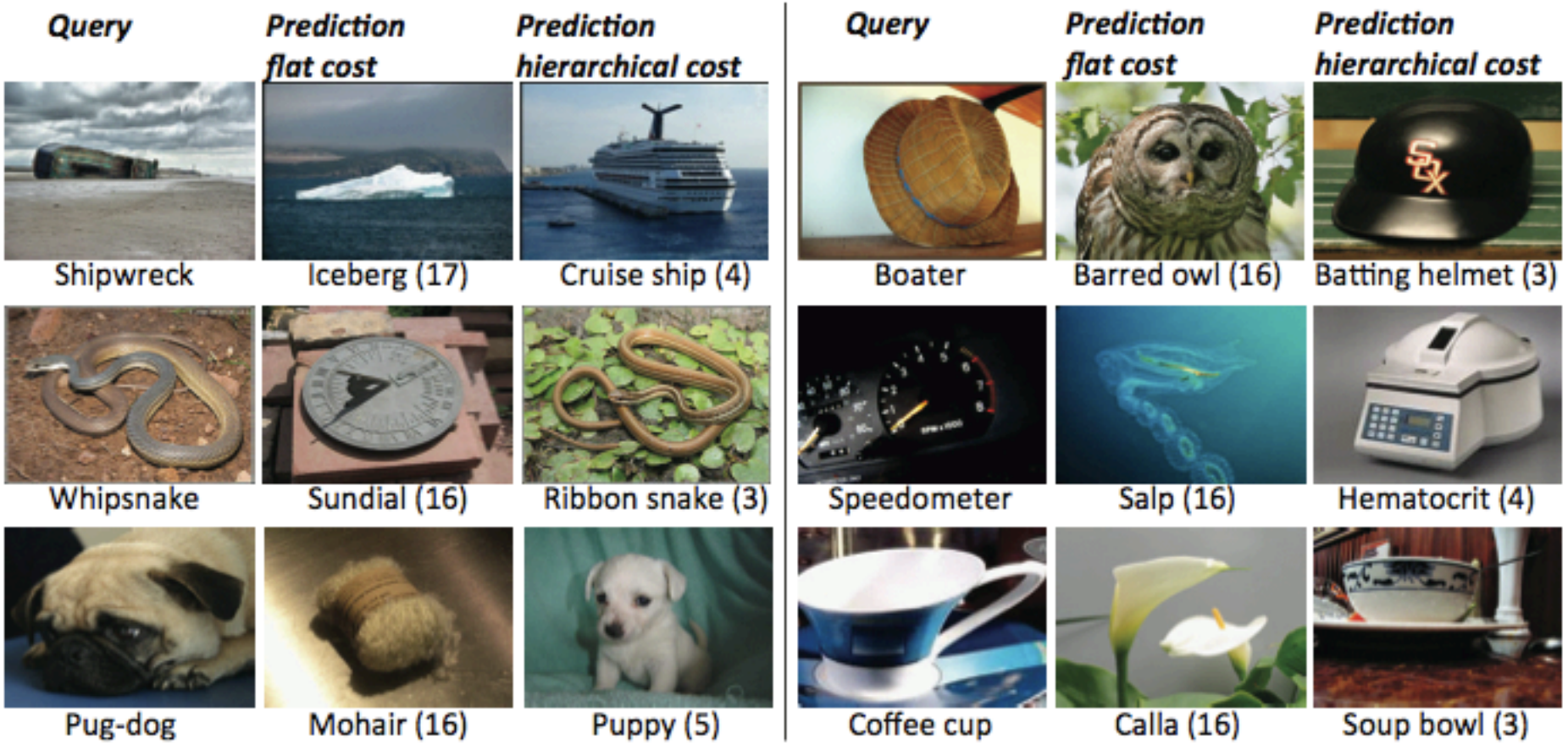


ImageNet Hierarchy From WordNet

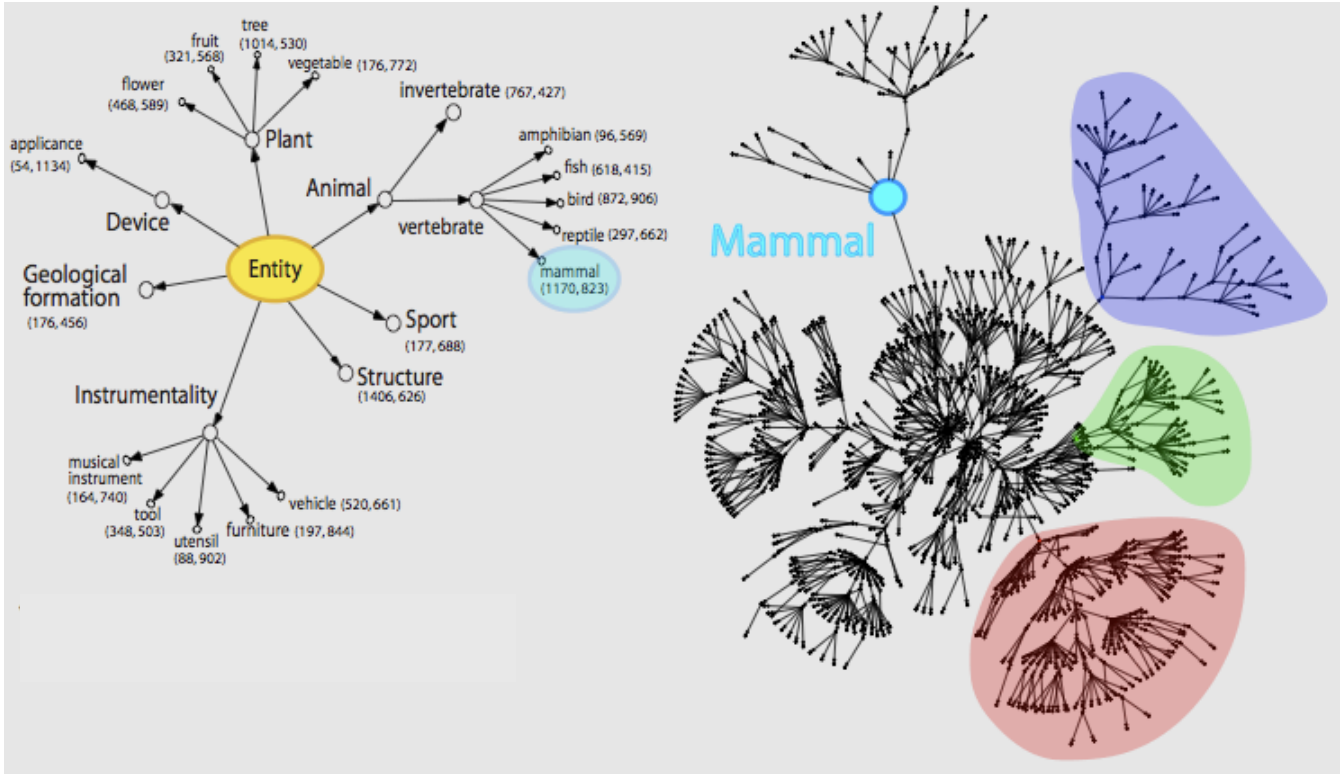


Average distance in WordNet correlates with difficulty of visual recognition!

ImageNet using Hierarchical Cost



Hierarchies vs Attributes



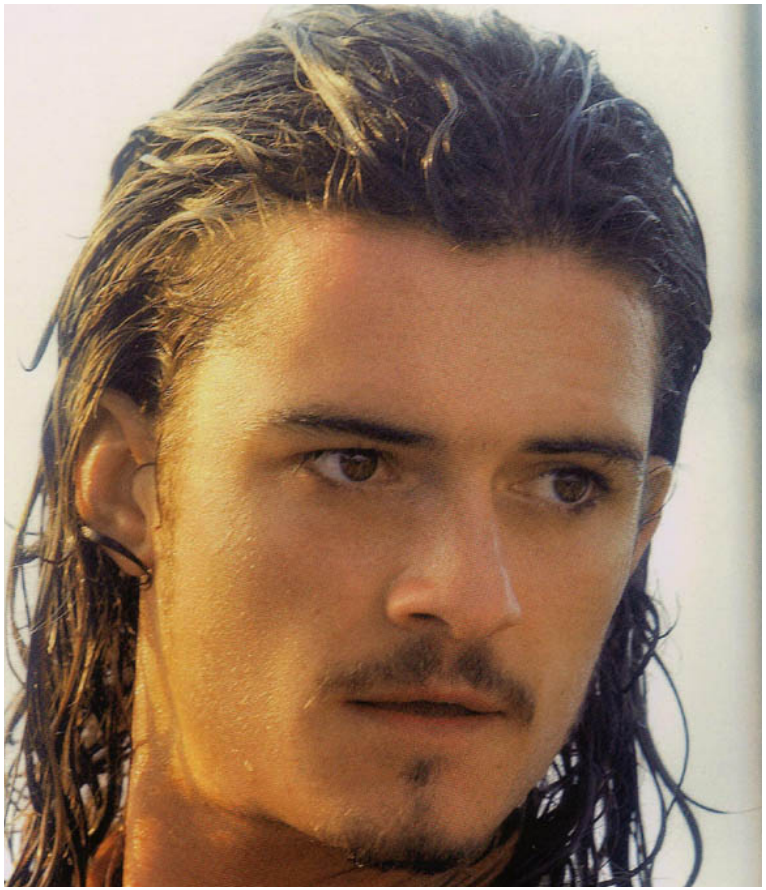
Attributes can be more flexible

Female Eyeglasses Middle-aged Dark hair

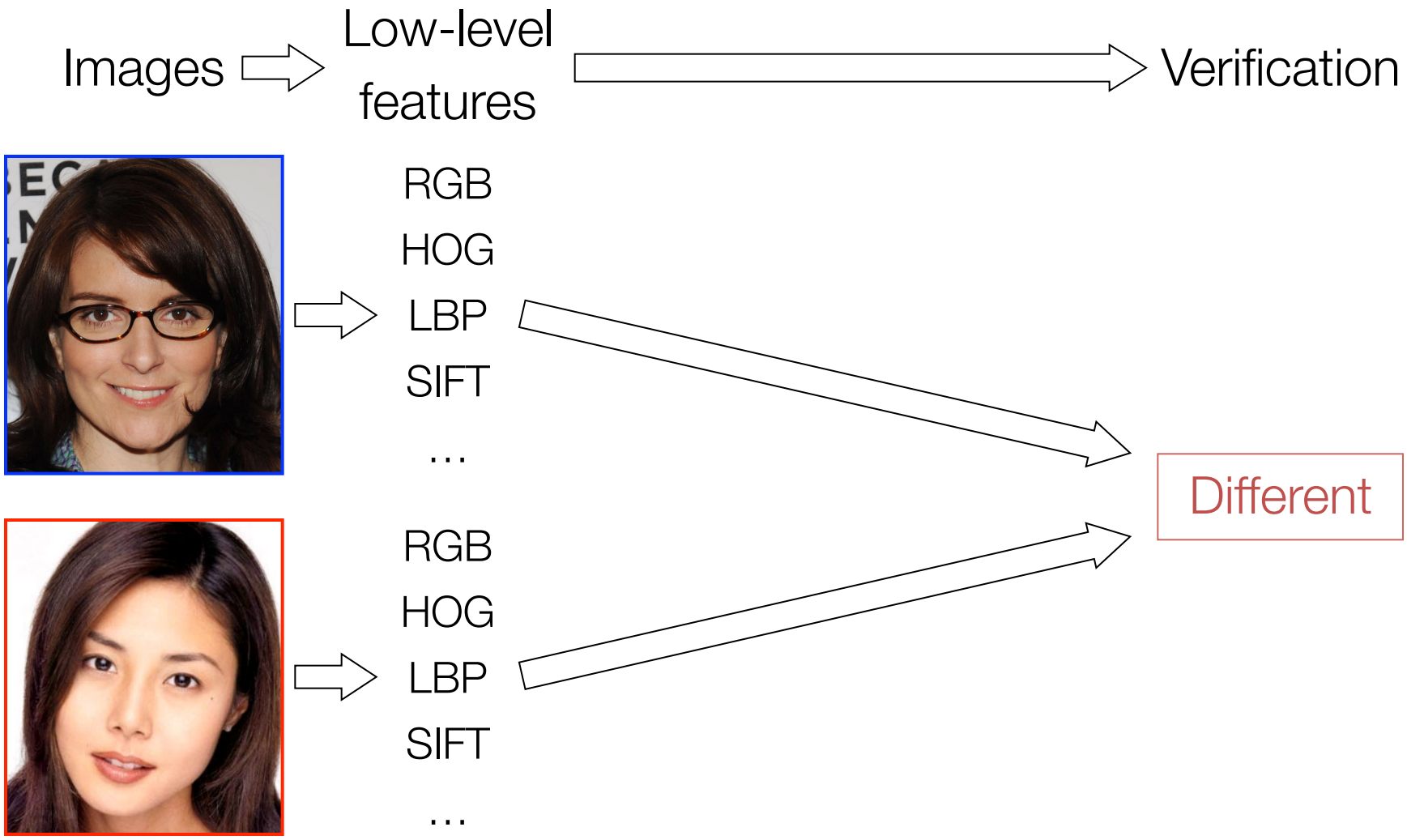
Attributes can be more flexible

Caucasian | Teeth showing | Outside | Tilted head

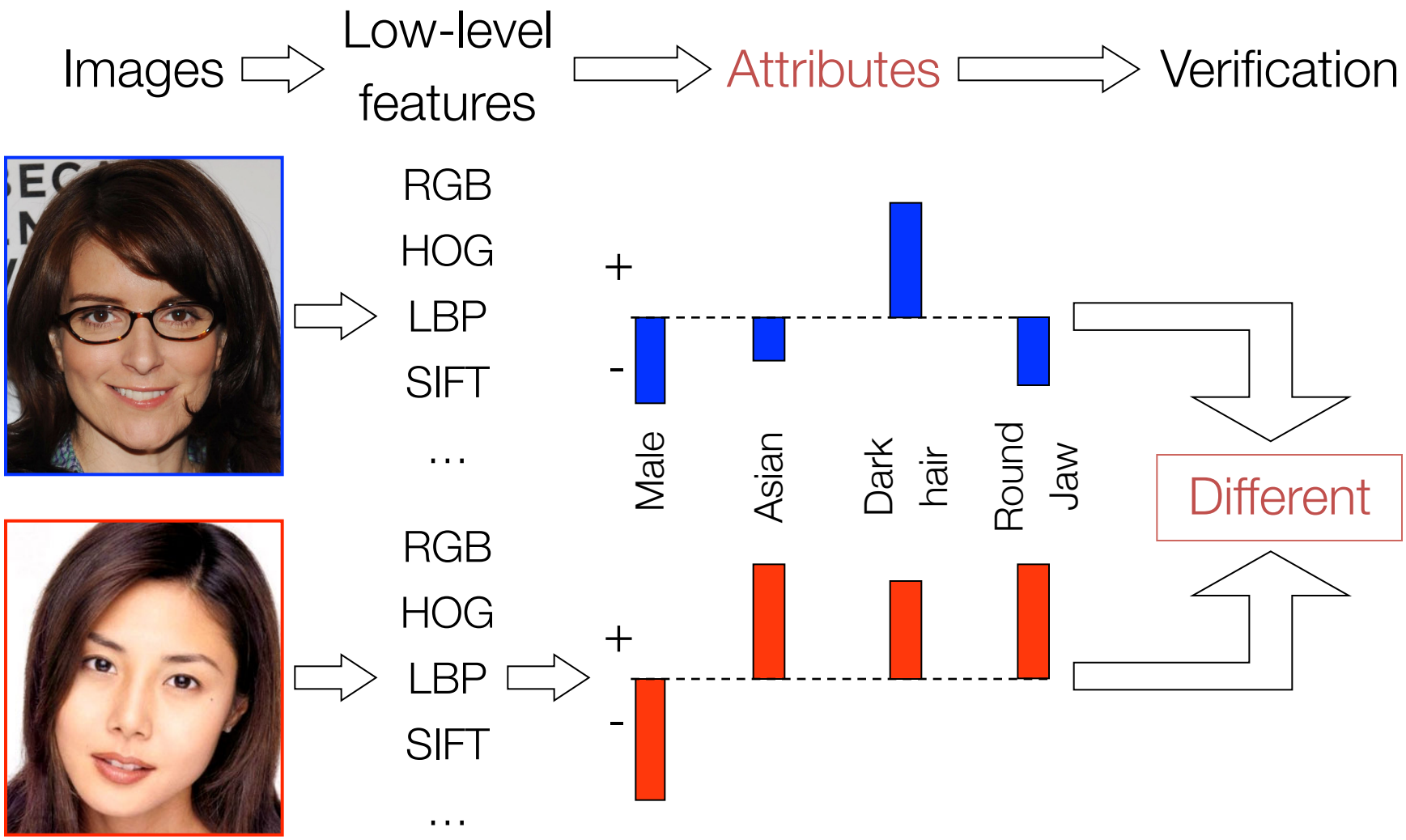
Verification
Are These Images of the Same Person?



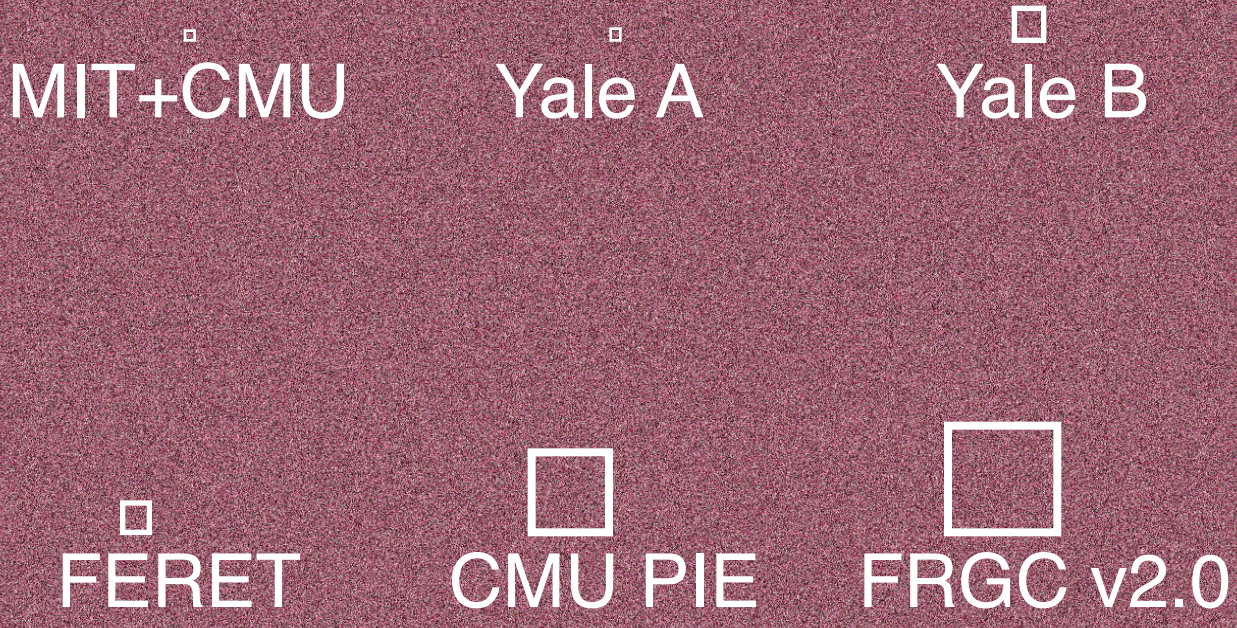
Prior Approaches



Our Approach: Attributes

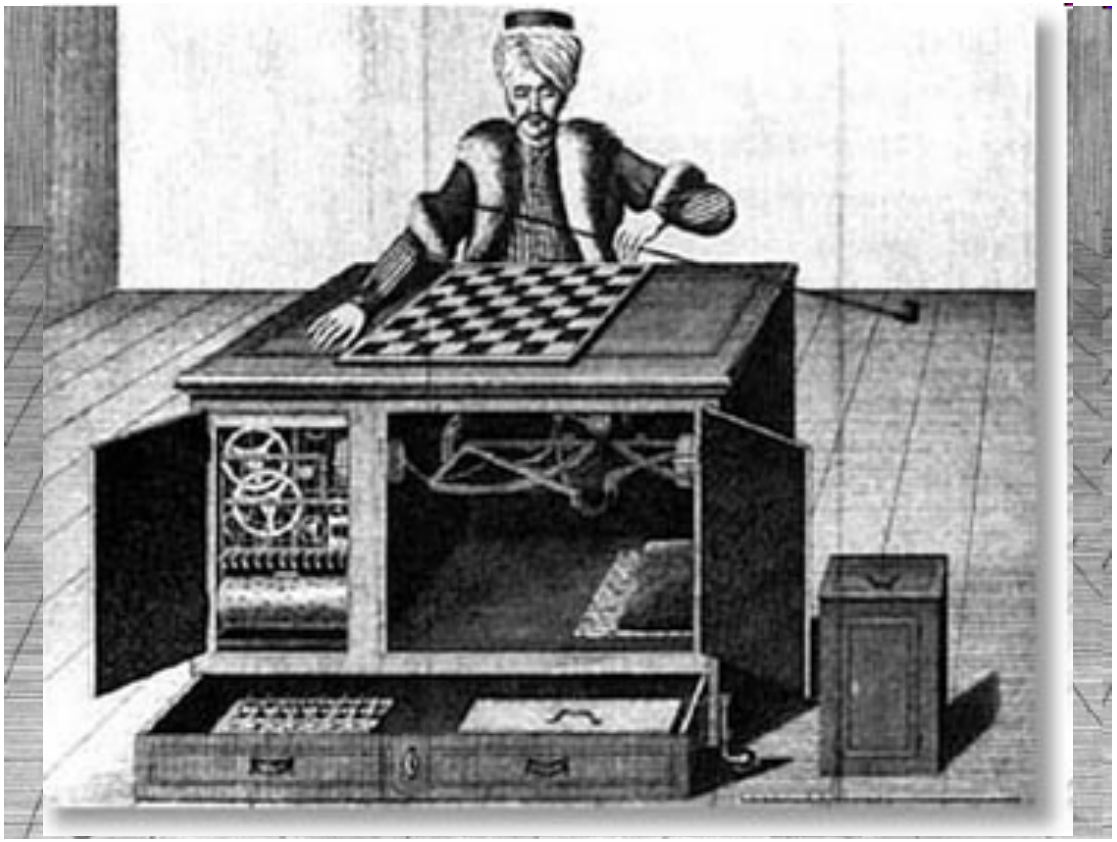


3,000,000 face images



- [ ] Nose Type
- [ ] Race
- [ ] Age
- [ ] Gender
- [ ] Eye Wear
- [ ] Eyebrow Type
- [ ] Hair Color
- [ ] Lip Type
- [ ] Blurry
- [ ] Mustache
- [ ] Eye Type
- [ ] Lighting
- [ ] Smiling
- [ ] Indoor/Outdoor
- [ ] Hair Type

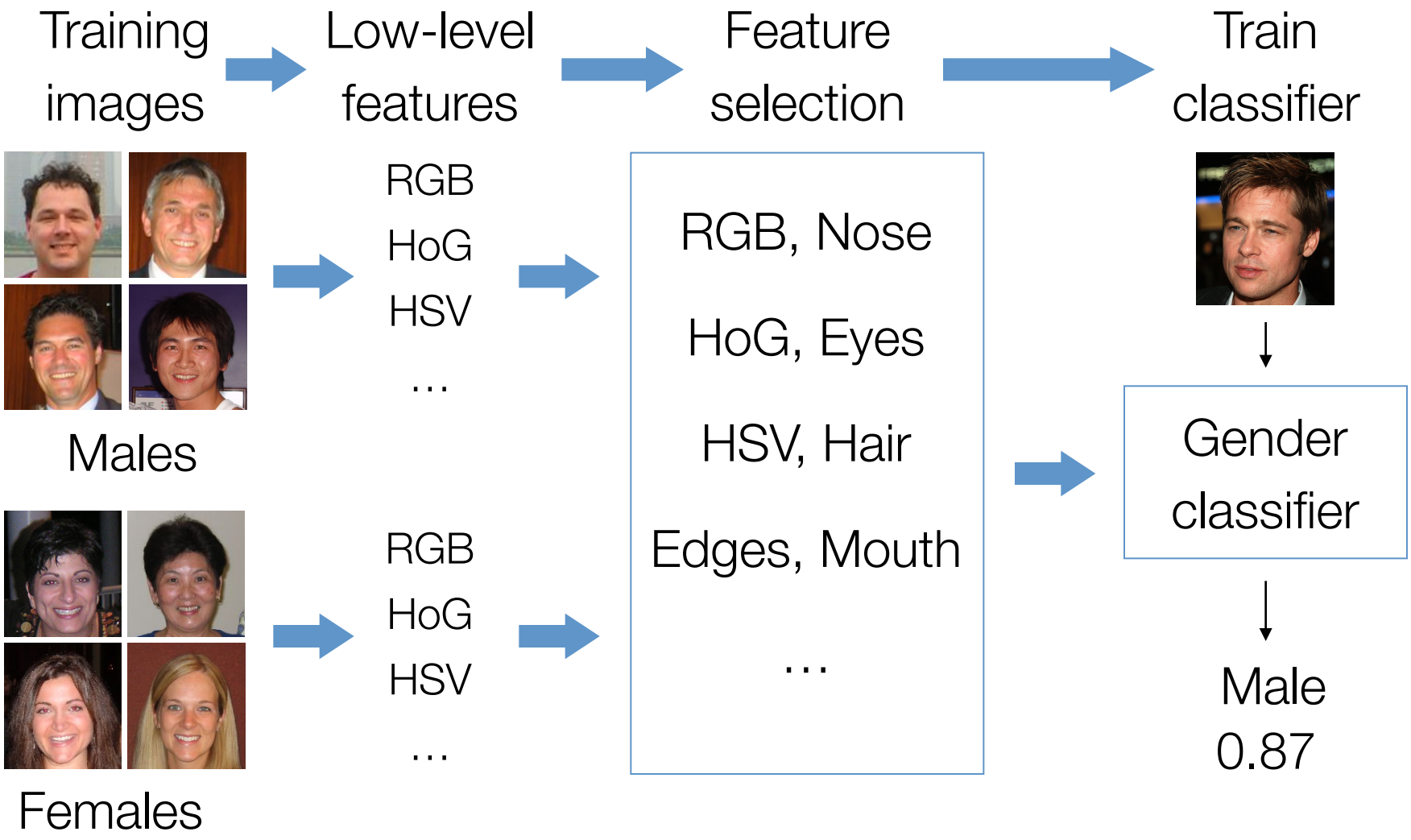
Amazon Mechanical Turk



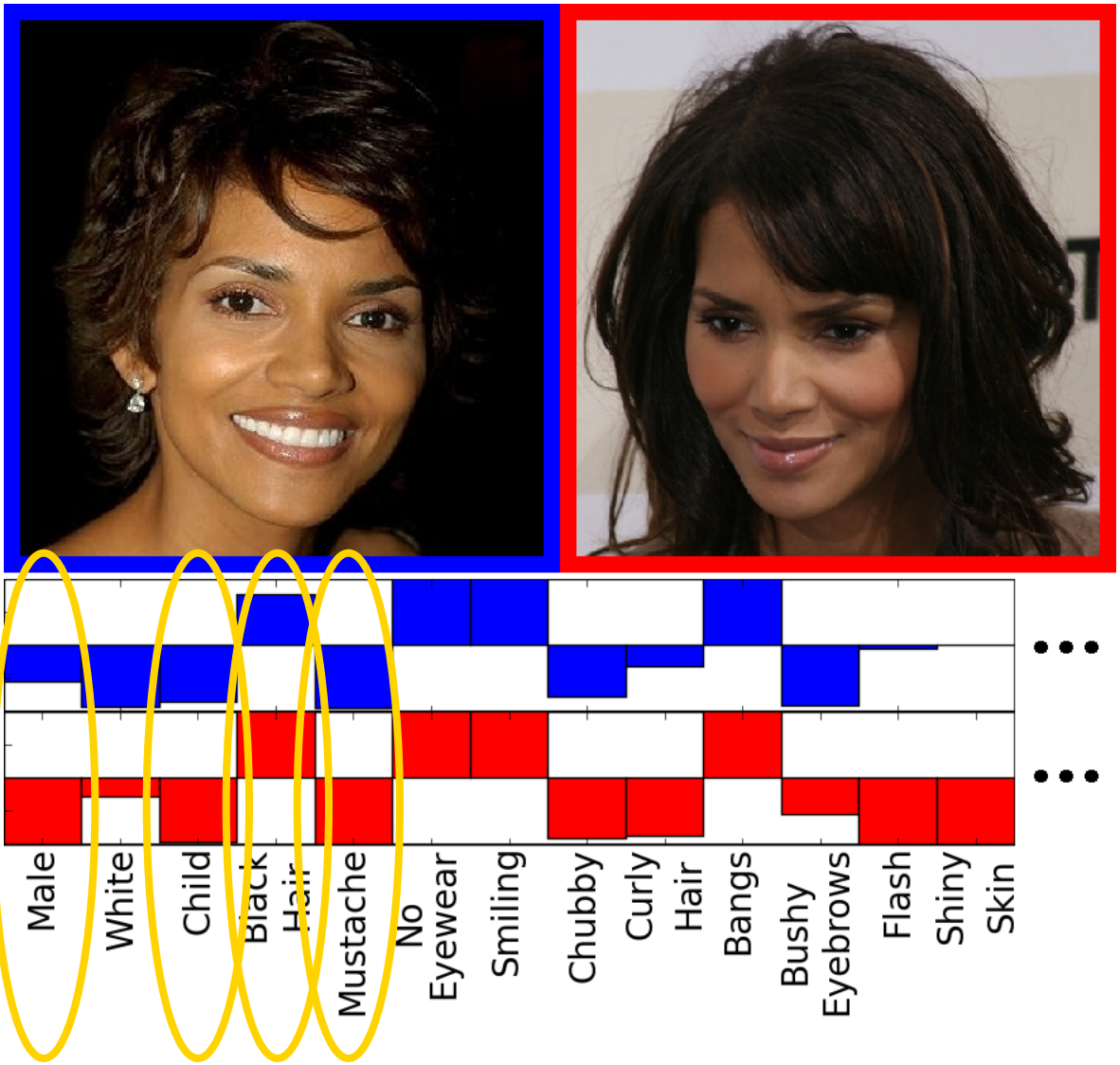
500,000 Attribute Labels = $5,000 + 1 month

See also [Deng, et al., 2009] [Vijayanarasimhan & Grauman, 2009]

Learning an attribute classifier



Using attributes to perform verification



Attributes are intuitive

Female

Young

Attractive

White

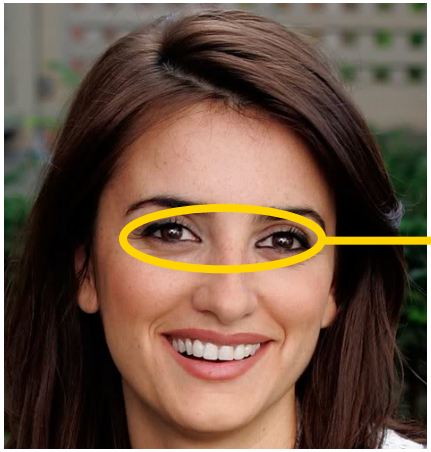
Black hair

Frontal pose

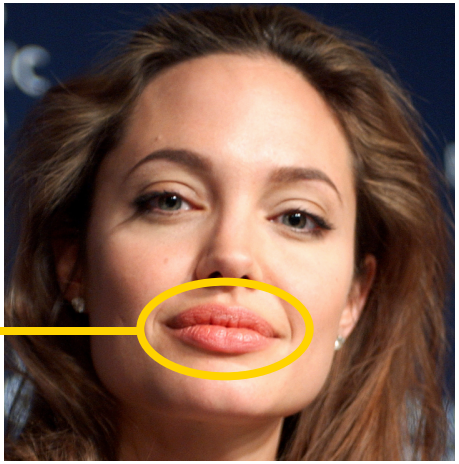
Mouth closed

Eyes open

Describe faces using similes



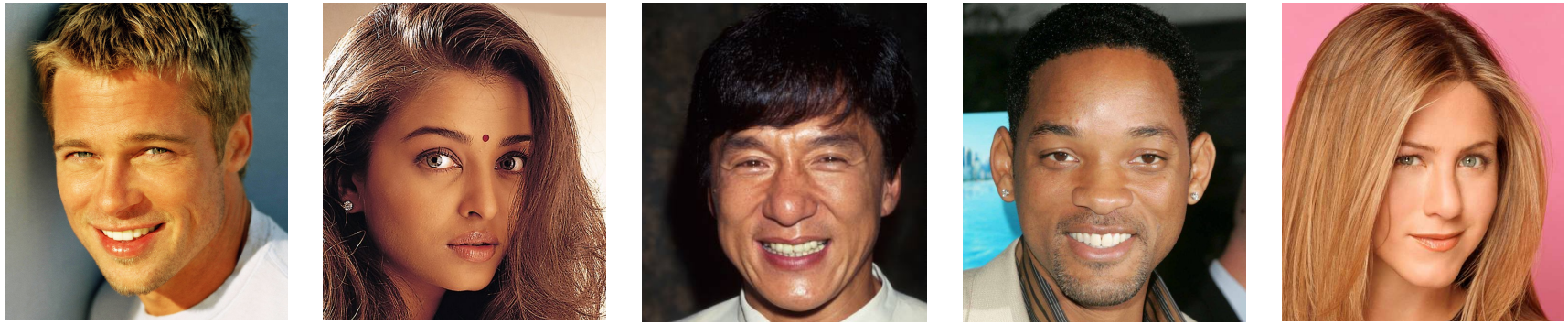
Penelope Cruz



Angelina Jolie

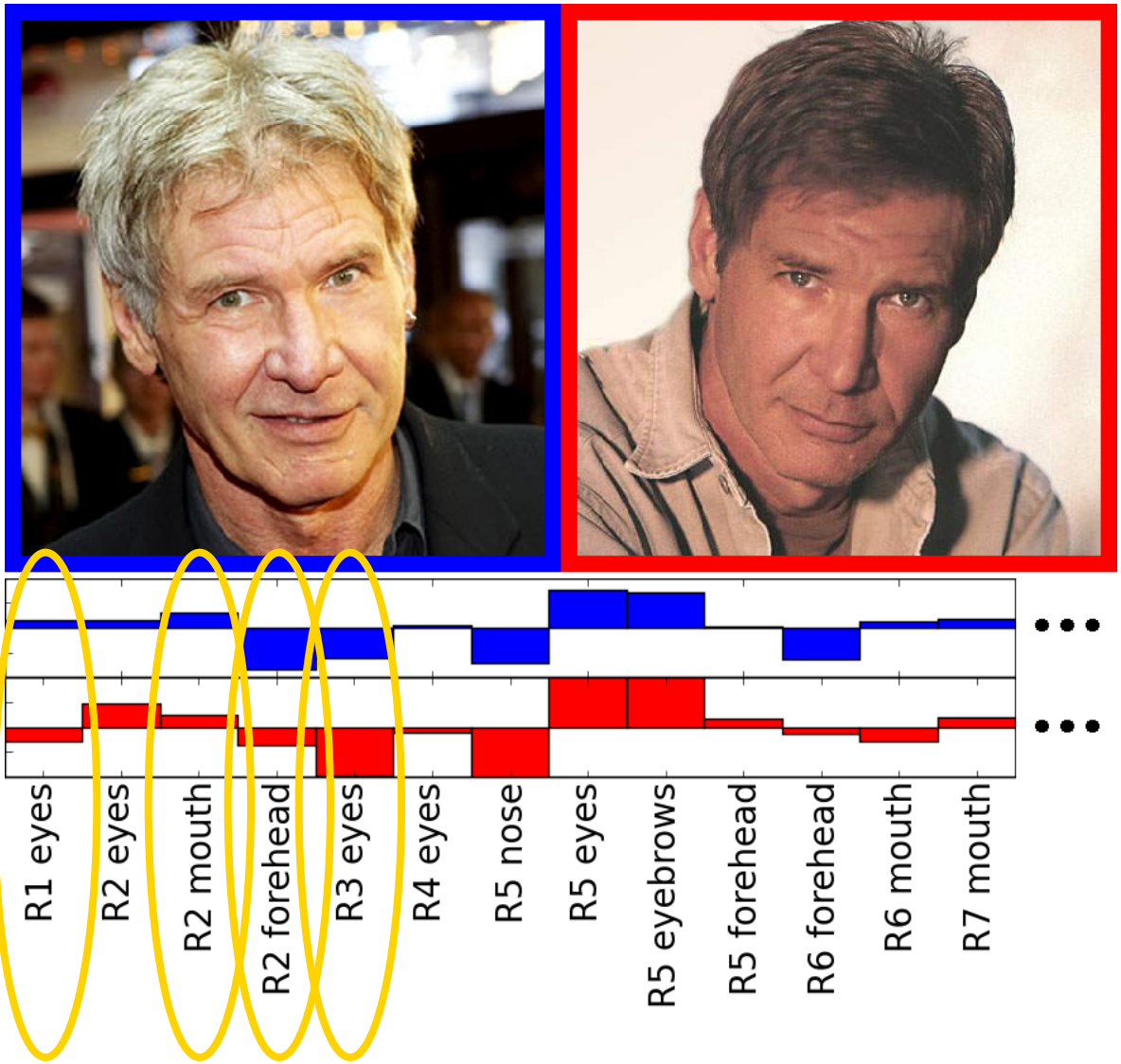
Training simile classifiers

Images of Penelope Cruz 's eyes



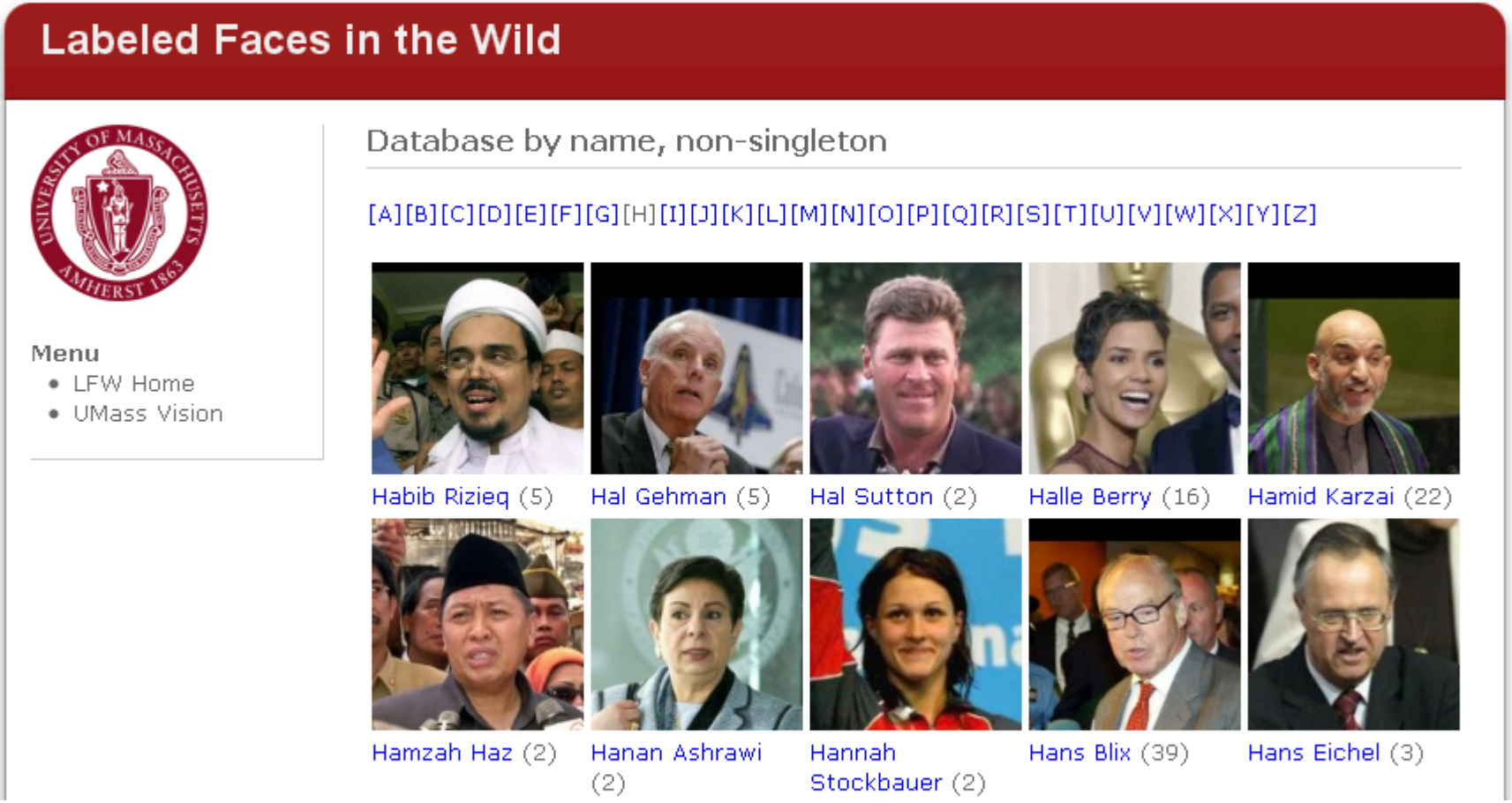
Images of other people 's eyes

Using simile classifiers for verification



Results

Labeled Faces in the Wild (LFW)



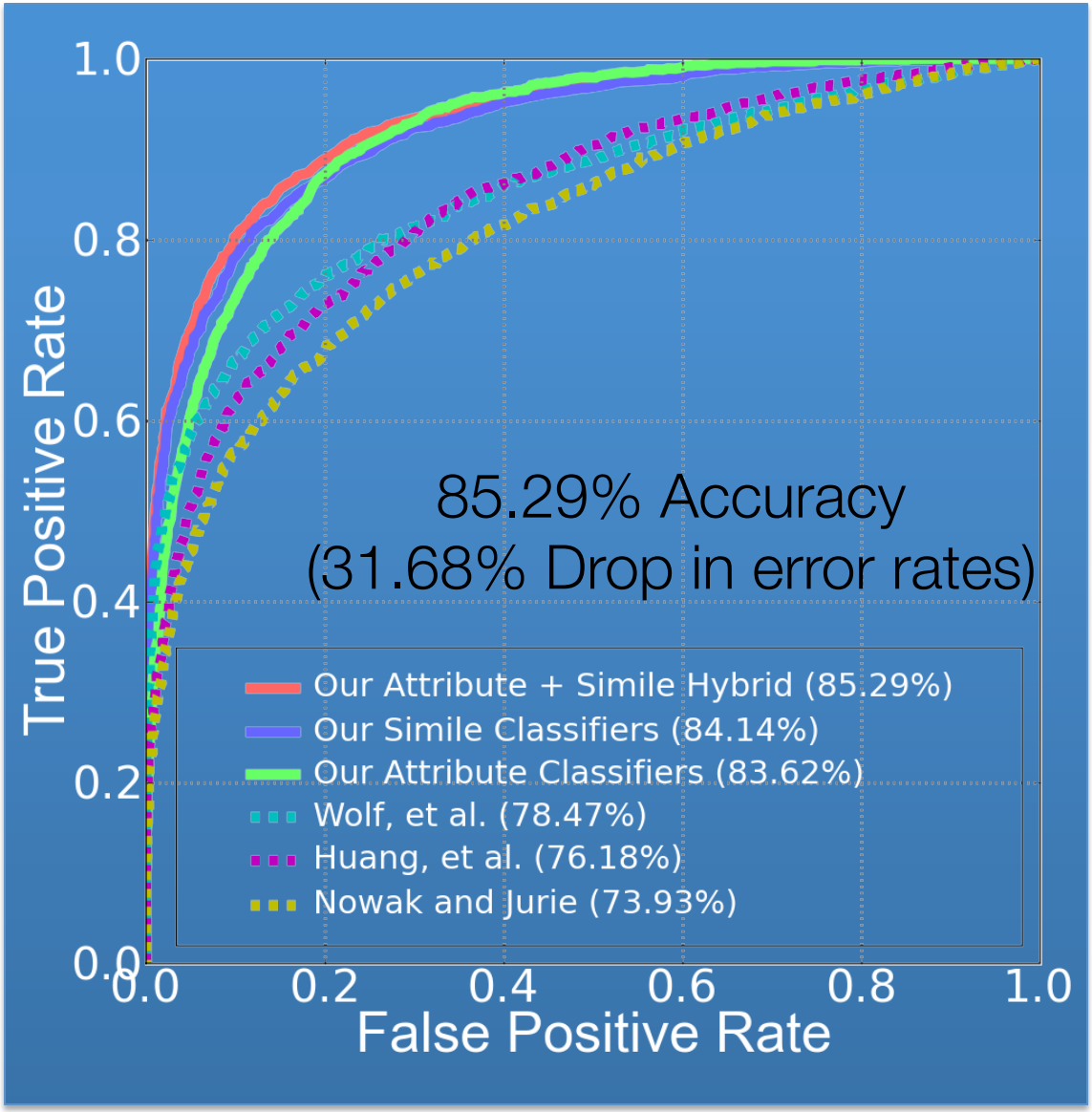
http://vis-www.cs.umass.edu/lfw

Experimental evaluation

LFW Image-Restricted Benchmark:

- 6,000 face pairs (3,000 same, 3,000 different)
- 10-fold cross-validation

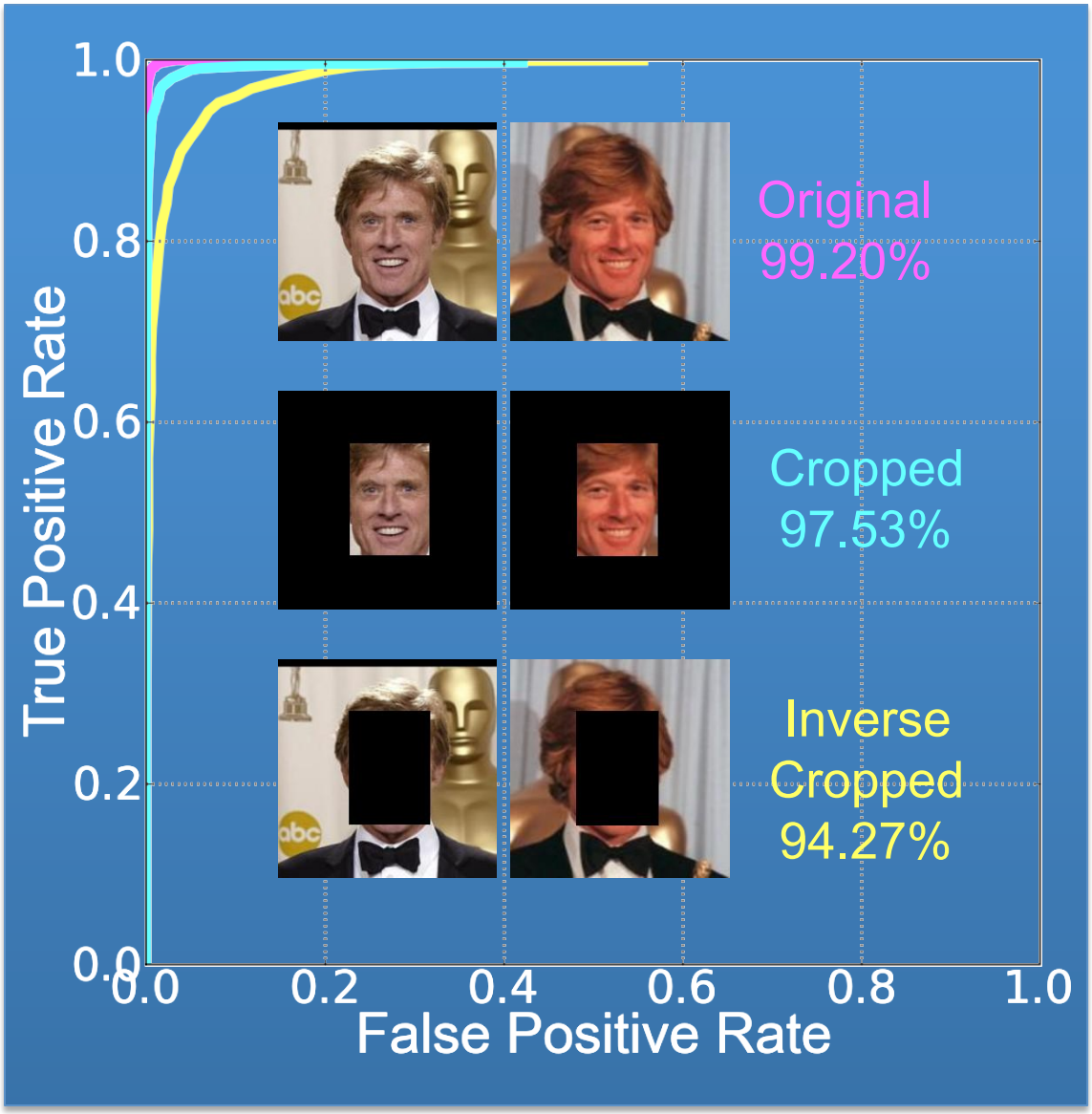
Our performance on LFW



Kumar, Berg, Belhumeur, Nayar ICCV 09

as of May 2009

Human face verification performance

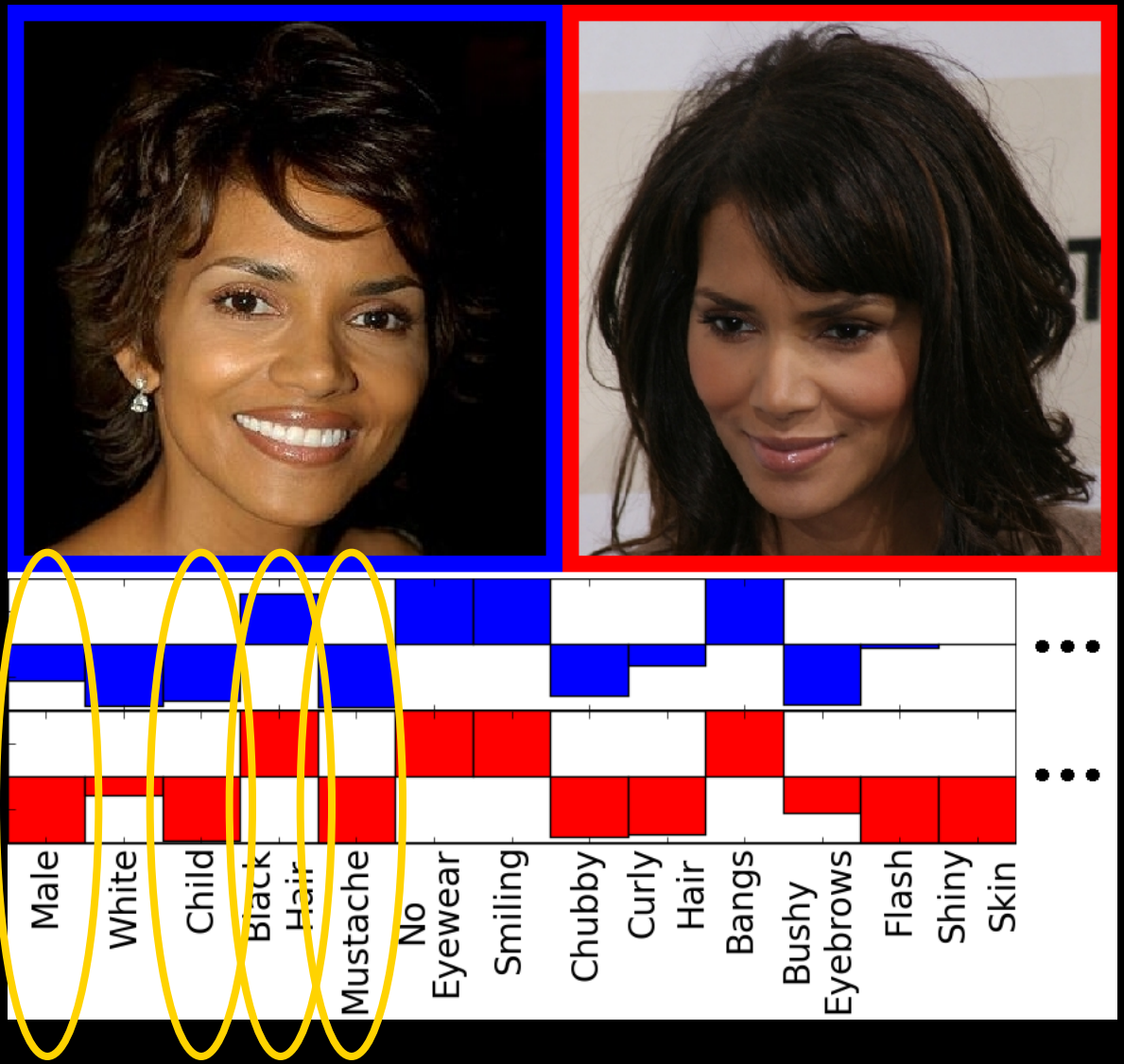


Kumar, Berg, Belhumeur, Nayar ICCV 09

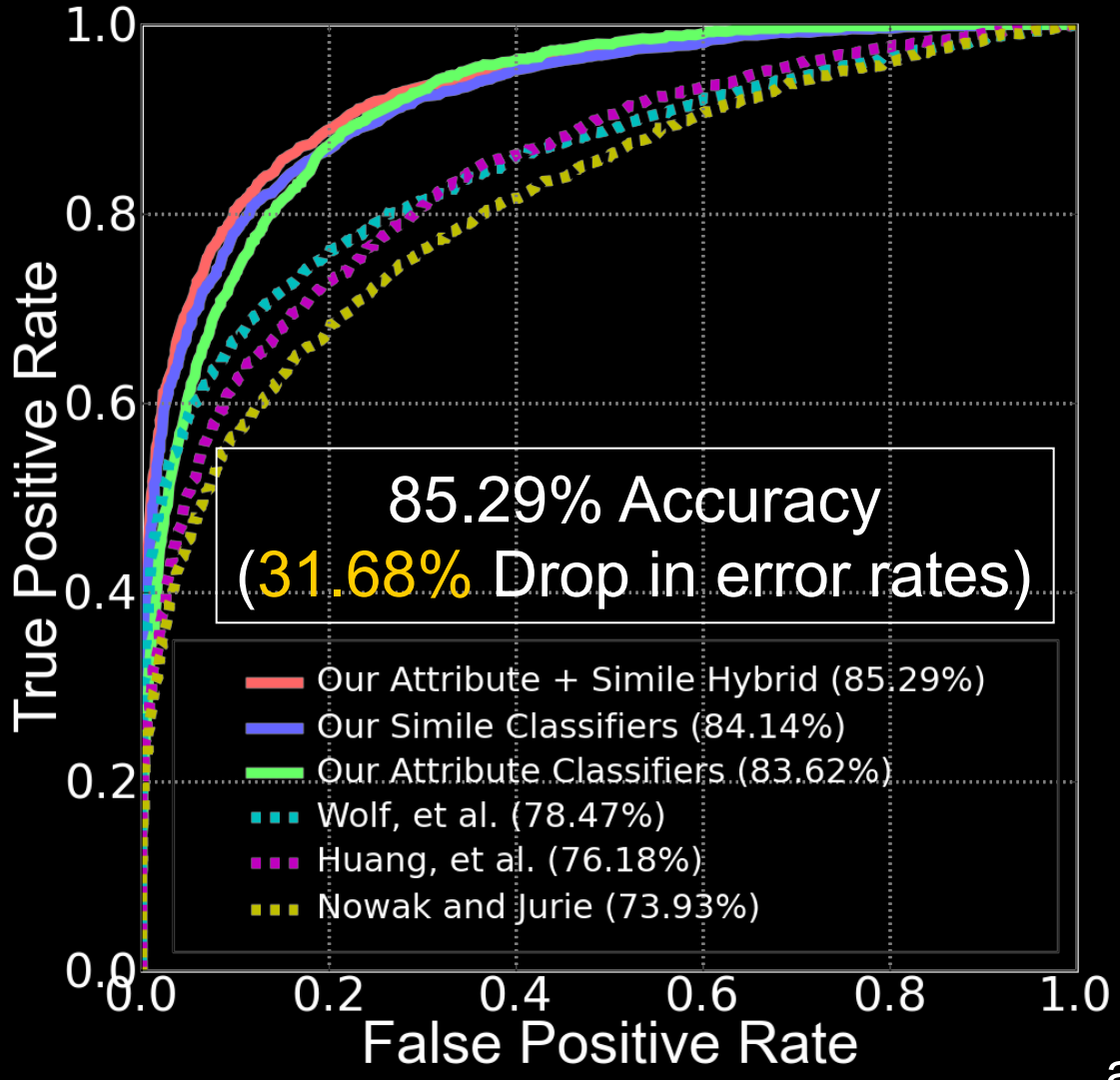
Take home messages

- 1st time using attributes as intermediate features actually improved on the state of the art for a vision problem
- Using describable visual attributes let us leverage humans via Mechanical Turk
- Possible to get closer to human understanding of face similarity
- Open questions on how to find attributes

Using attributes to perform verification

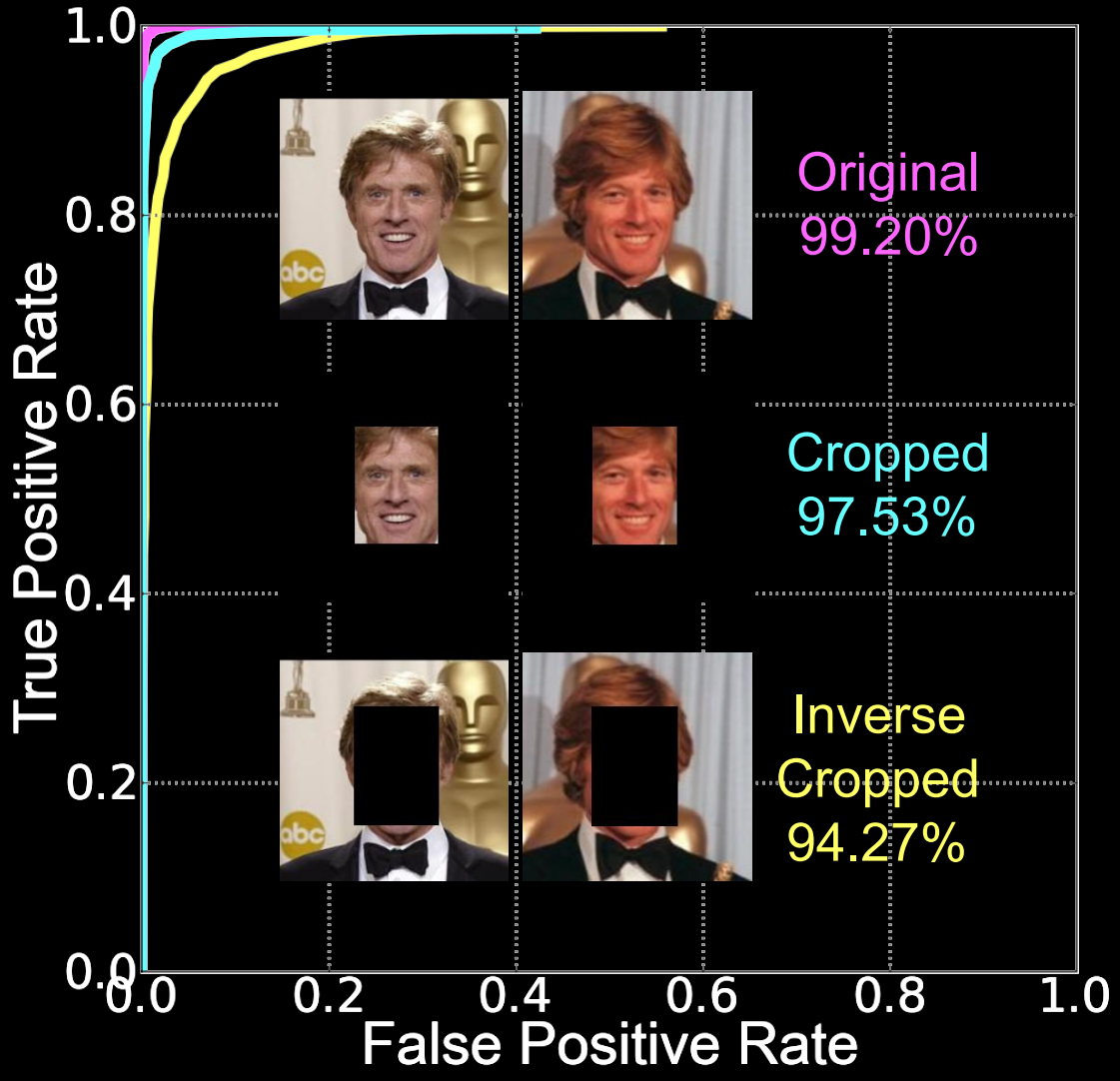


Our performance on LFW



as of May 2009

Human face verification performance



A quick experiment

You will see a mask, then image, then mask.

What do you see?


3ashq Al5ail
Photographer ©


3ashq Alsail


10638

Iconic images

- We want to select “good” images corresponding to textual keywords
- Iconic images as prototypes? (Rosch, 1970s)
  - People can consistently rank typicality w.r.t. a category and the ranks correlate with speed of recognition
  - Note: Here we don’t care about defining categories, only about “mining” prototypes for categories defined by arbitrary keywords

Criteria for canonical view selection

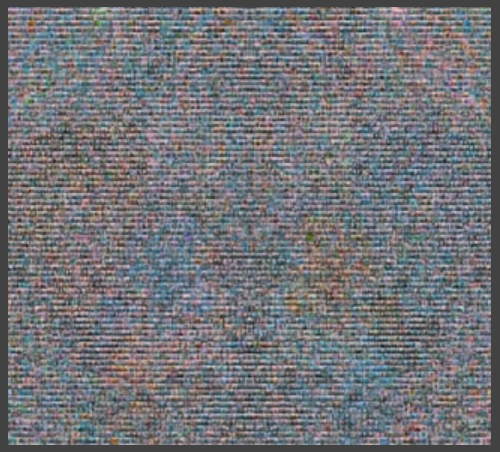
Palmer, Rosch and Chase (1981)

1. Given a set of photos, which view do you like the best?
2. When taking a photo yourself, which viewpoint do you choose?
3. From which viewpoint is the object easiest to recognize?
4. When imagining the object in your head, which view do you see?

Iconic images

- Getting at the “shared mental representations” for general visual categories

Web/
Flickr

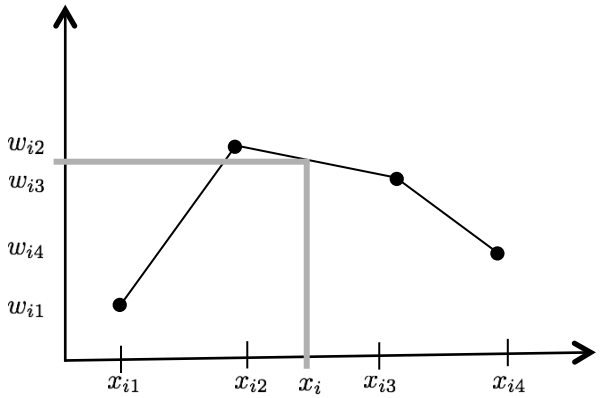


We can do this automatically! Berg & Berg Internet Vision ‘09

Thank you

co-authors and collaborators (so far)

Example Sparse Encoding



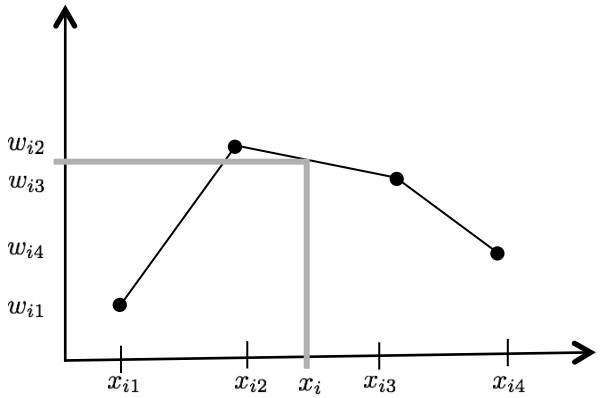
$$x_i = \alpha x_{i2} + (1-\alpha) x_{i3}$$

$$\hat{x}_i = \begin{bmatrix} 0 & \alpha & 1-\alpha & 0 \end{bmatrix}$$

$$\hat{w}_i = \begin{bmatrix} w_{i1} & w_{i2} & w_{i3} & w_{i4} \end{bmatrix}$$

Example Sparse Encoding

$$
\begin{aligned}
h(x) &= \sum_{j=1}^{\#\text{sv}} \alpha^j K(x, x^j) + b \\
&= \sum_{j=1}^{\#\text{sv}} \alpha^j \left(\sum_{i=1}^{\#\text{dimensions}} K_i(x_i, x_i^j) \right) + b \\
&= \sum_{i=1}^{\#\text{dimensions}} \left(\sum_{j=1}^{\#\text{sv}} \alpha^j K_i(x_i, x_i^j) \right) + b \\
&= \sum_{i=1}^{\#\text{dimensions}} h_i(x_i)
\end{aligned}
$$



$$x_i = \alpha x_{i2} + (1 - \alpha) x_{i3}$$

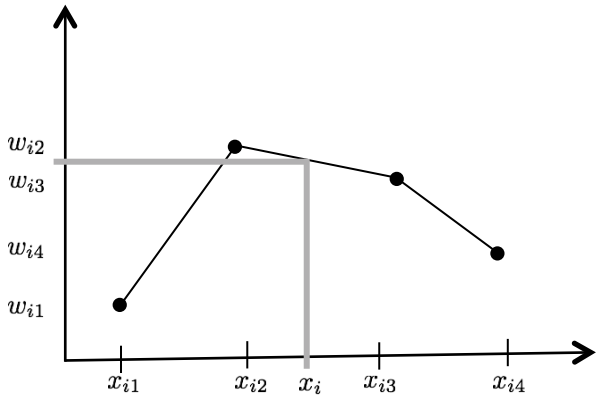
$$\hat{x}_i = \begin{bmatrix} 0 & \alpha & 1-\alpha & 0 \end{bmatrix}$$

$$\hat{w}_i = \begin{bmatrix} w_{i1} & w_{i2} & w_{i3} & w_{i4} \end{bmatrix}$$

Example Sparse Encoding

$$
\begin{aligned}
h(x) &= \sum_{j=1}^{\#\text{sv}} \alpha^j K(x, x^j) + b \\
&= \sum_{j=1}^{\#\text{sv}} \alpha^j \left(\sum_{i=1}^{\#\text{dimensions}} K_i(x_i, x_i^j) \right) + b \\
&= \sum_{i=1}^{\#\text{dimensions}} \left(\sum_{j=1}^{\#\text{sv}} \alpha^j K_i(x_i, x_i^j) \right) + b \\
&= \sum_{i=1}^{\#\text{dimensions}} h_i(x_i)
\end{aligned}
$$

e.g., $h_i(x_i) = \sum_{j=1}^{\#sv} \alpha^j \min(x_i, x_i^j)$



$$x_i = \alpha x_{i2} + (1-\alpha) x_{i3}$$

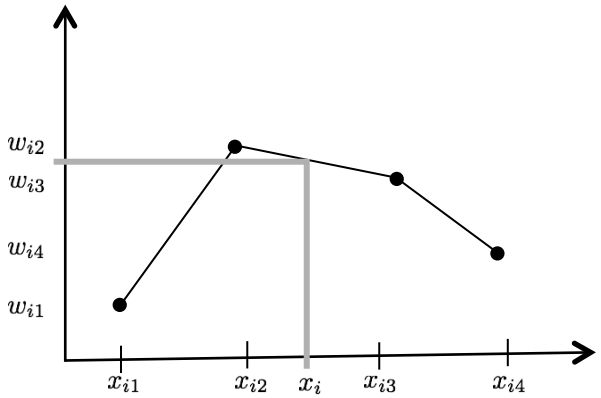
$$\hat{x}_i = \begin{bmatrix} 0 & \alpha & 1-\alpha & 0 \end{bmatrix}$$

$$\hat{w}_i = \begin{bmatrix} w_{i1} & w_{i2} & w_{i3} & w_{i4} \end{bmatrix}$$

Example Sparse Encoding

$$
\begin{aligned}
h(x) &= \sum_{j=1}^{\#\text{sv}} \alpha^j K(x, x^j) + b \\
&= \sum_{j=1}^{\#\text{sv}} \alpha^j \left(\sum_{i=1}^{\#\text{dimensions}} K_i(x_i, x_i^j) \right) + b \\
&= \sum_{i=1}^{\#\text{dimensions}} \left(\sum_{j=1}^{\#\text{sv}} \alpha^j K_i(x_i, x_i^j) \right) + b \\
&= \sum_{i=1}^{\#\text{dimensions}} h_i(x_i)
\end{aligned}
$$

e.g., $h_i(x_i) = \sum_{j=1}^{\#sv} \alpha^j \min(x_i, x_i^j)$



$$x_i = \alpha x_{i2} + (1-\alpha) x_{i3}$$

$$\hat{x}_i = \begin{bmatrix} 0 & \alpha & 1-\alpha & 0 \end{bmatrix}$$

$$\hat{w}_i = \begin{bmatrix} w_{i1} & w_{i2} & w_{i3} & w_{i4} \end{bmatrix}$$

now $h_i(x_i) = \hat{w}' \hat{x}_i$

How?

Encode

$$x \longmapsto \hat{x}$$

So that

$$\hat{w}'\hat{x} \approx \sum_{j=1:\#sv} \alpha_j K(x, x_j)$$

sparse representation is important

Linear

$$\begin{aligned} \text{minimize :} \quad & w'w + c\sum \xi^j & \\ \text{subject to :} \quad & y^i(w'x^j + b) & \geq 1 - \xi^j \\ & \xi^j & \geq 0 \end{aligned}$$

$$\sum h_i(x_i)$$

$$h_i(x_i) = w_i x_i$$

Piecewise Linear

$$\begin{aligned} \text{minimize :} \quad & \hat{w}'H\hat{w} + c\sum \xi^j & \\ \text{subject to :} \quad & \hat{y}^i(\hat{w}'\hat{x}^j + b) & \geq 1 - \xi^j \\ & \xi^j & \geq 0 \end{aligned}$$

$$\sum \hat{h}_i(\hat{x}_i)$$

$$\hat{h}_i(\hat{x}_i) = \hat{w}_i \hat{x}_i$$

How?

Encode

$$x \mapsto \hat{x}$$

So that

$$\hat{w}'\hat{x} \approx \sum_{j=1:\#sv} \alpha_j K(x, x_j)$$

sparse representation is important

Linear

$$\begin{aligned} \text{minimize :} \quad & w'w + c\sum \xi^j & \\ \text{subject to :} \quad & y^i(w'x^j + b) & \geq 1 - \xi^j \\ & \xi^j & \geq 0 \end{aligned}$$

$$\sum h_i(x_i)$$

$$h_i(x_i) = w_i x_i$$

Piecewise Linear

$$\begin{aligned} \text{minimize :} \quad & \hat{w}'H\hat{w} + c\sum \xi^j & \\ \text{subject to :} \quad & \hat{y}^i(\hat{w}'\hat{x}^j + b) & \geq 1 - \xi^j \\ & \xi^j & \geq 0 \end{aligned}$$

$$\sum \hat{h}_i(\hat{x}_i)$$

$$\hat{h}_i(\hat{x}_i) = \hat{w}_i \hat{x}_i$$

$$H = \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & . & & \\ & & & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}$$

An old problem

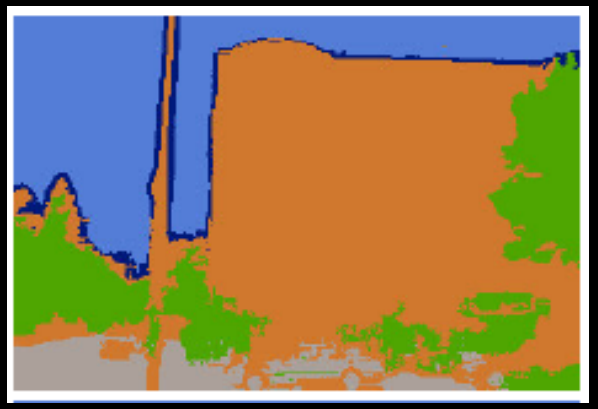
I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have “327”? No. I have sky, house, and trees.

– Max Wertheimer 1923

An old problem

I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have “327”? No. I have sky, house, and trees.

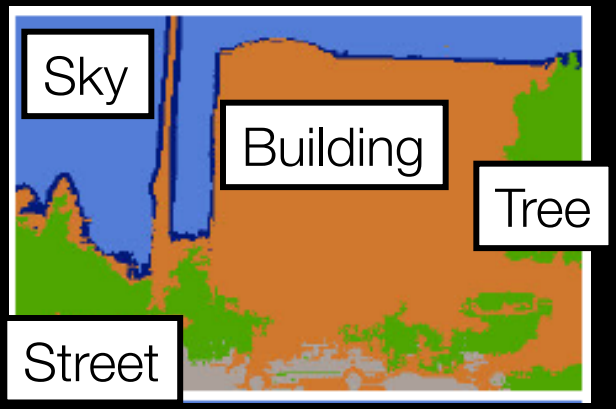
– Max Wertheimer 1923



An old problem

I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have “327”? No. I have sky, house, and trees.

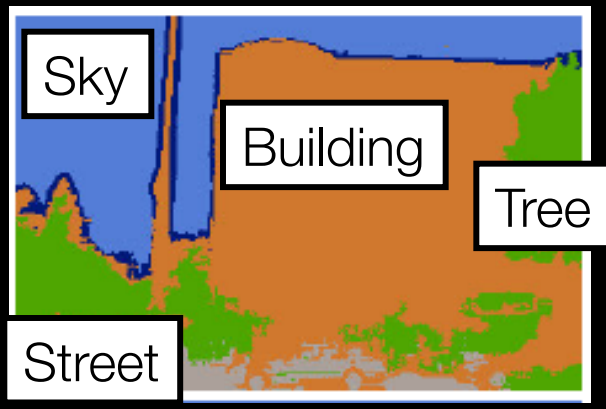
– Max Wertheimer 1923



An old problem

I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have "327"? No. I have sky, house, and trees.

– Max Wertheimer 1923



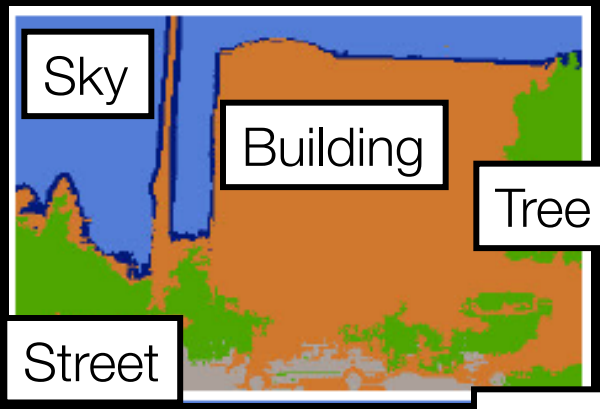
Use this coarse parsing for
more detailed parsing of buildings

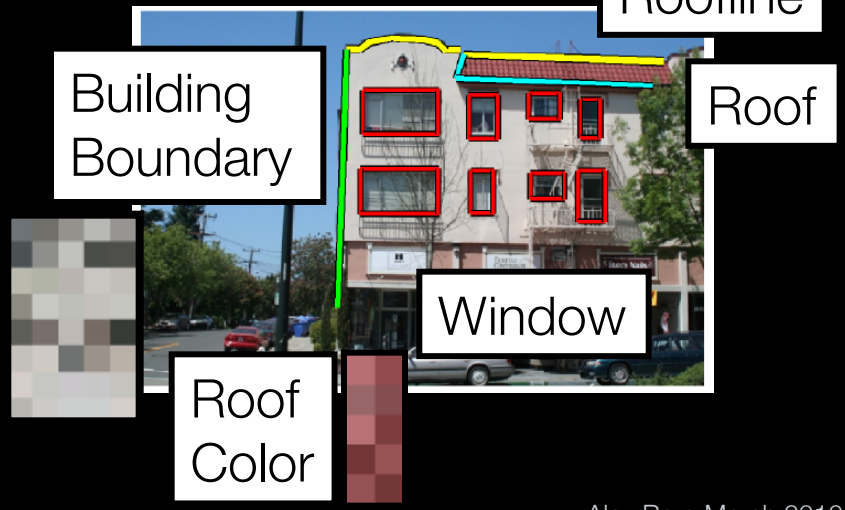
An old problem

I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have "327"? No. I have sky, house, and trees.

– Max Wertheimer 1923

Use this coarse parsing for
more detailed parsing of buildings



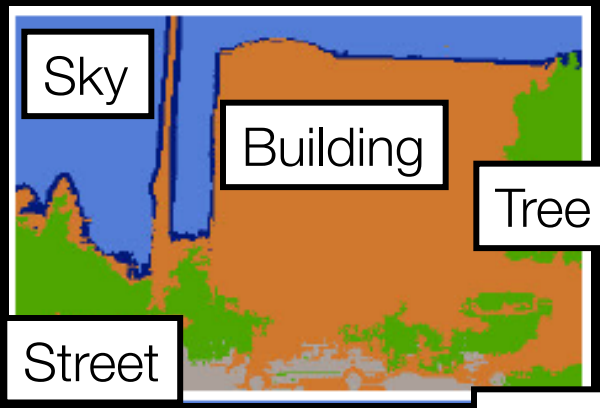


Why Architectural Scenes?

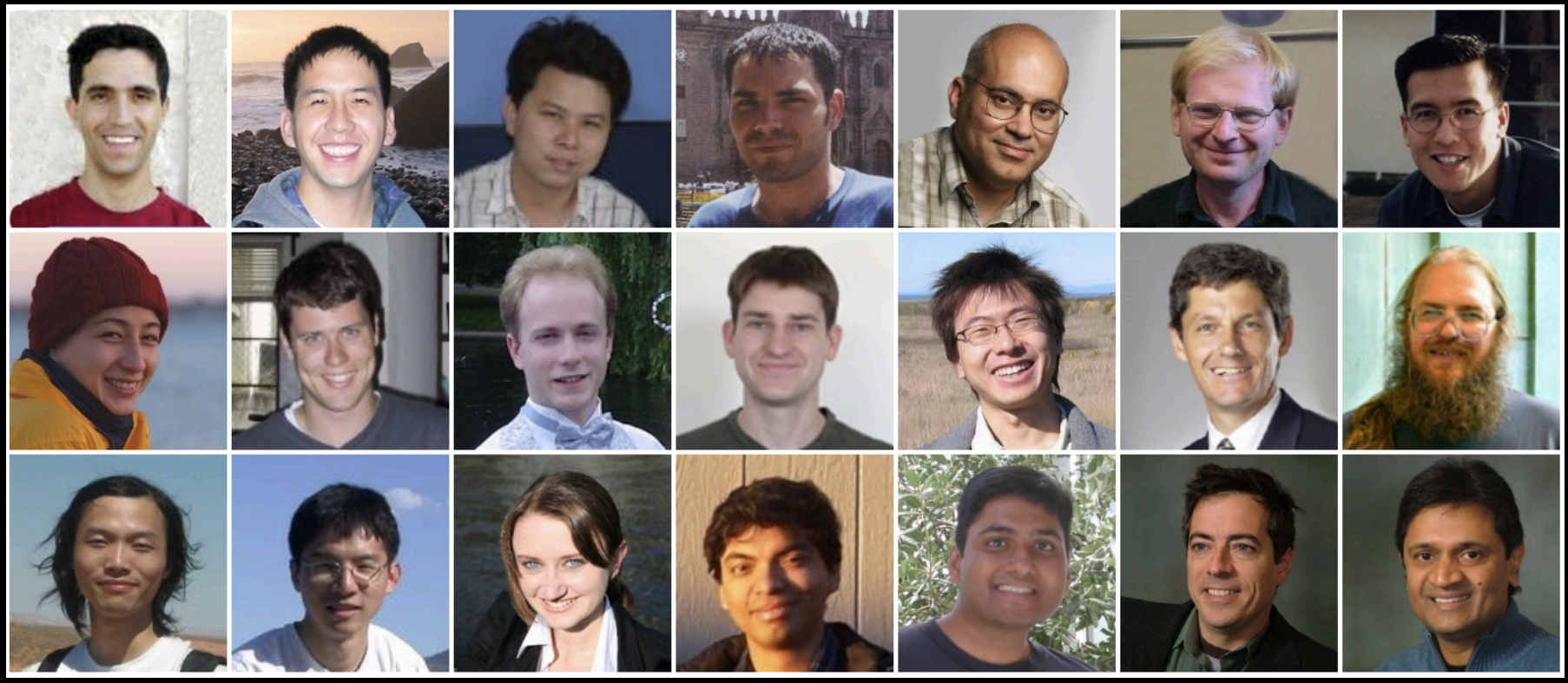
- Make up decent portion of our surroundings

- Microsoft, Amazon, etc. are collecting and trying to use a great deal of this type of data, anything automatic is helpful

- Stress current computational approaches to visual recognition...



Thank you and my coauthors so far…

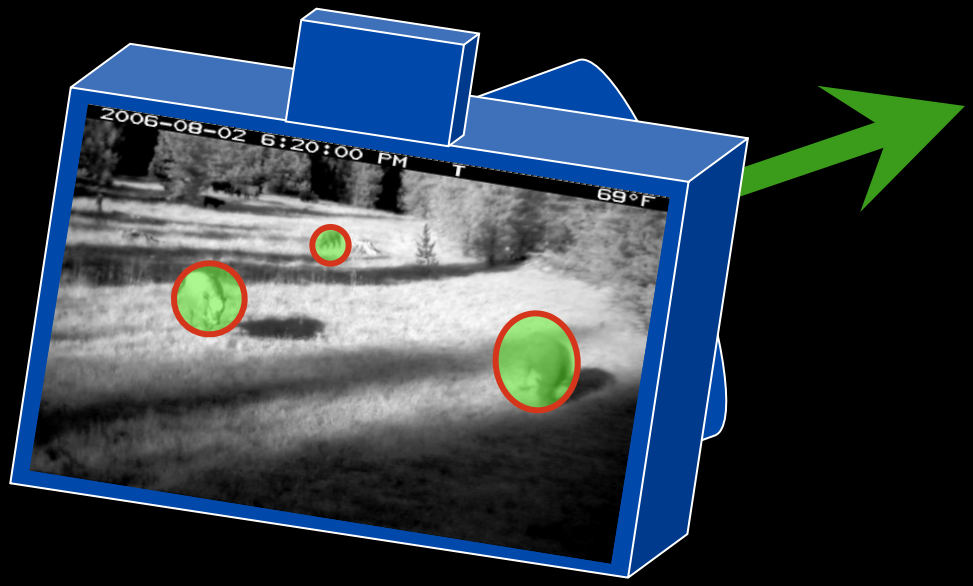


Papers and contact info available on-line. Search for “Alex Berg”

Context can Help Recognition.

Recognizing the road, foliage, and other cars, makes recognizing the red car easier.

Other ways to Help Recognition



3 Cows at 6:20 PM

If only we could convince cows to wear special uniforms designed to aid in detection and tracking.

Alas the proud cow doth protest…

Range of Difficulty in Recognition

Easy

vs

or

Other asopects might be difficult, but the recognition phase is relatively easy.

Range of Difficulty in Recognition

Easy

vs

or

Recognizing the same image

Range of Difficulty in Recognition

Medium

vs

or

Small change

Recognizing the same object

Range of Difficulty in Recognition

Medium

Large change

vs

or

Recognizing the same object

Range of Difficulty in Recognition

Medium to Difficult


vs

or


Recognizing the same object

Range of Difficulty in Recognition

Difficult

Chair vs

Recognizing the same object category

Range of Difficulty in Recognition

Difficult

vs

Recognizing the same object category