Lecture 8:
ML for image classification, then Face and pedestrian detection

COS 429: Computer Vision

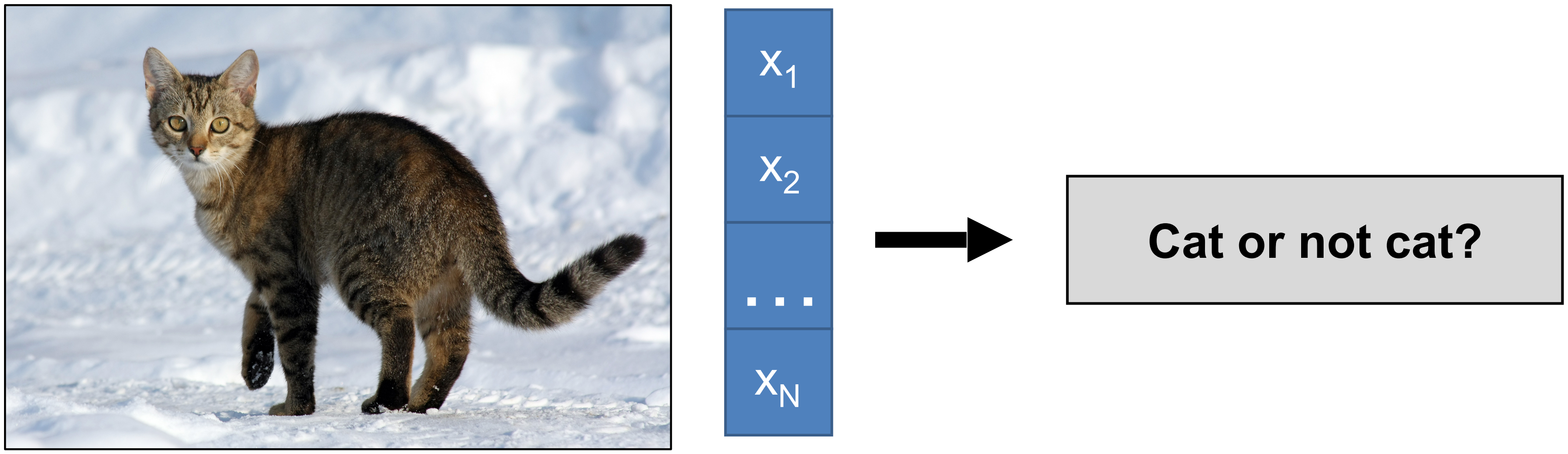
PRINCETON UNIVERSITY

Slides adapted from J. Deng

Building on last lecture: ML for image classification

Classification

Start with simplest example: binary classification



Actually: a feature vector representing the image

Credit: D. Fouhey

Easiest Form of Classification

Just **memorize** (as in a Python dictionary)
Consider cat/dog/hippo classification.

If this:
cat.

If this:
dog.

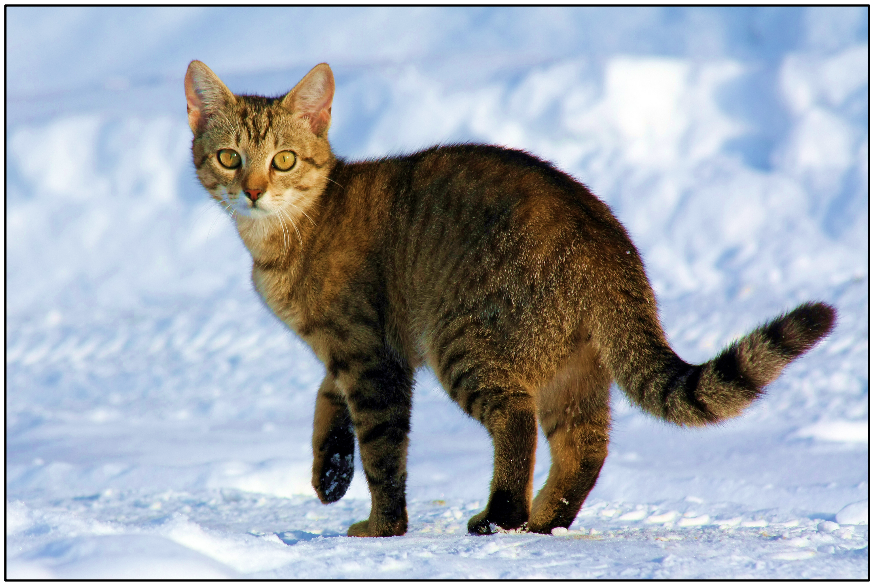
If this:
hippo.

Credit: D. Fouhey

Easiest Form of Classification

Where does this go wrong?

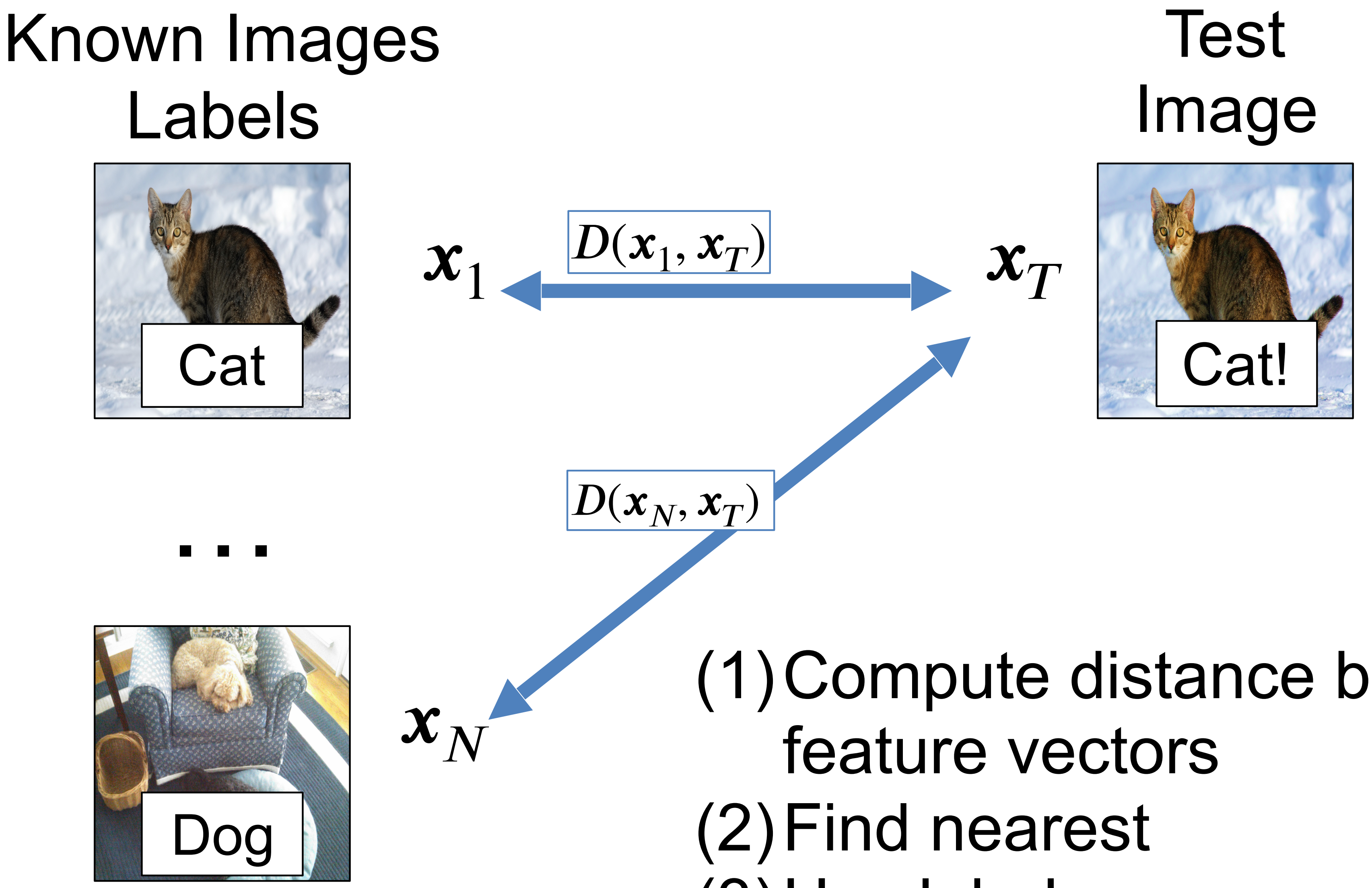
Rule: if this, then cat



Hmmm. Not quite the same.

Credit: D. Fouhey

Easiest Form of Classification



(1) Compute distance between feature vectors

(2) Find nearest

(3) Use label.

Credit: D. Fouhey

Nearest Neighbor

"Algorithm"

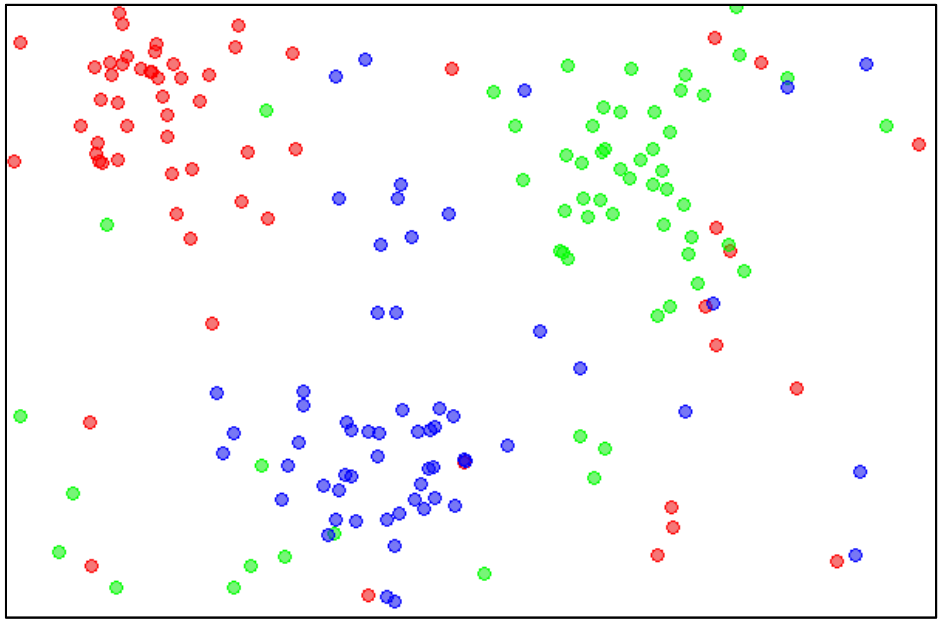
Training ($\mathbf{x}_i$,y_i): Memorize training set

Inference (x):

```
bestDist, prediction = Inf, None
for i in range(N):
    if dist($x_i$,x) < bestDist:
        bestDist = dist($x_i$,x)
        prediction = $y_i$
```

Nearest Neighbor

2D Datapoints
(colors = labels)



2D Predictions
(colors = labels)

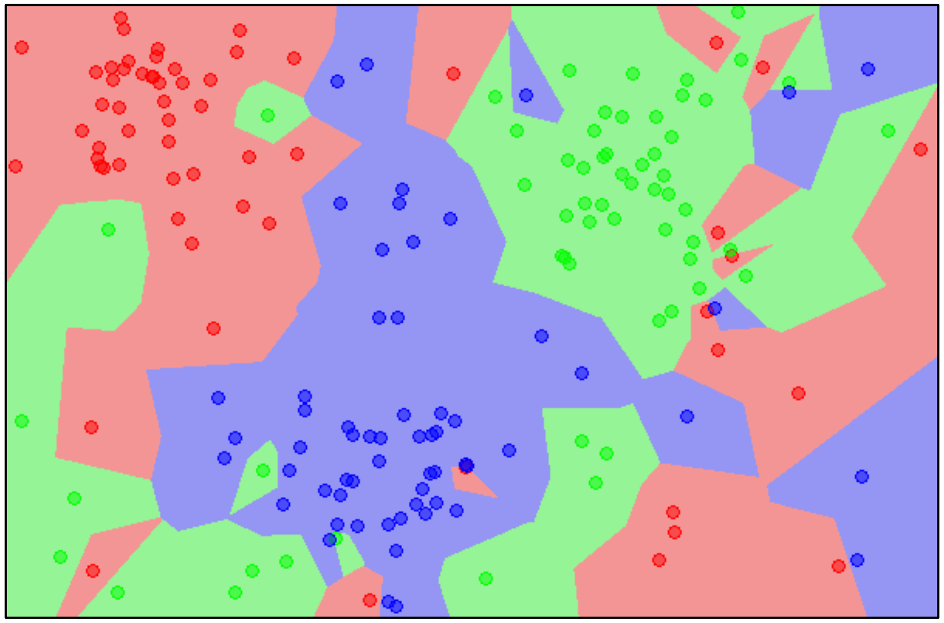


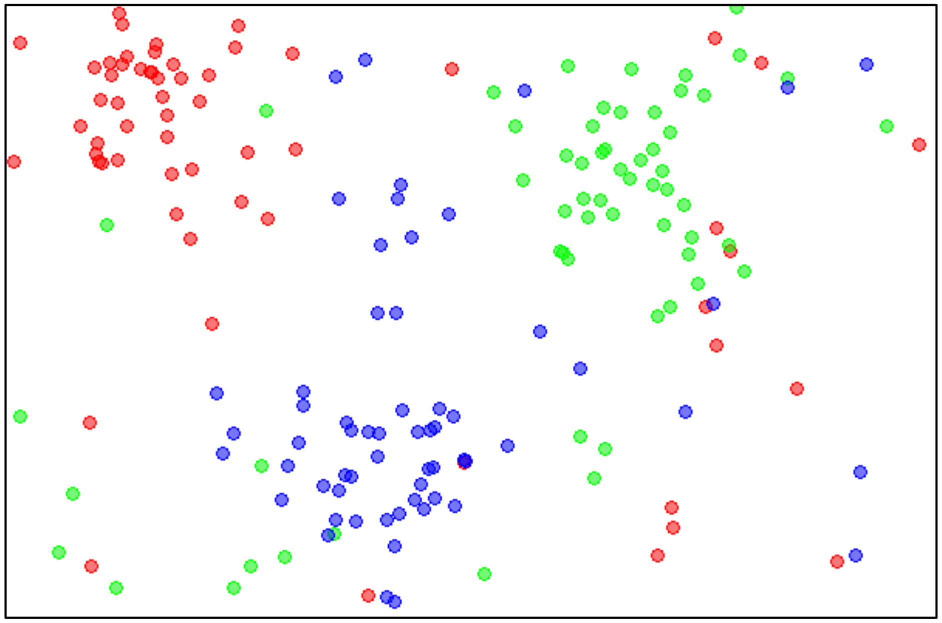
Diagram Credit: Wikipedia

Credit: D. Fouhey

K-Nearest Neighbors

Take top K-closest points, vote

2D Datapoints
(colors = labels)



2D Predictions
(colors = labels)

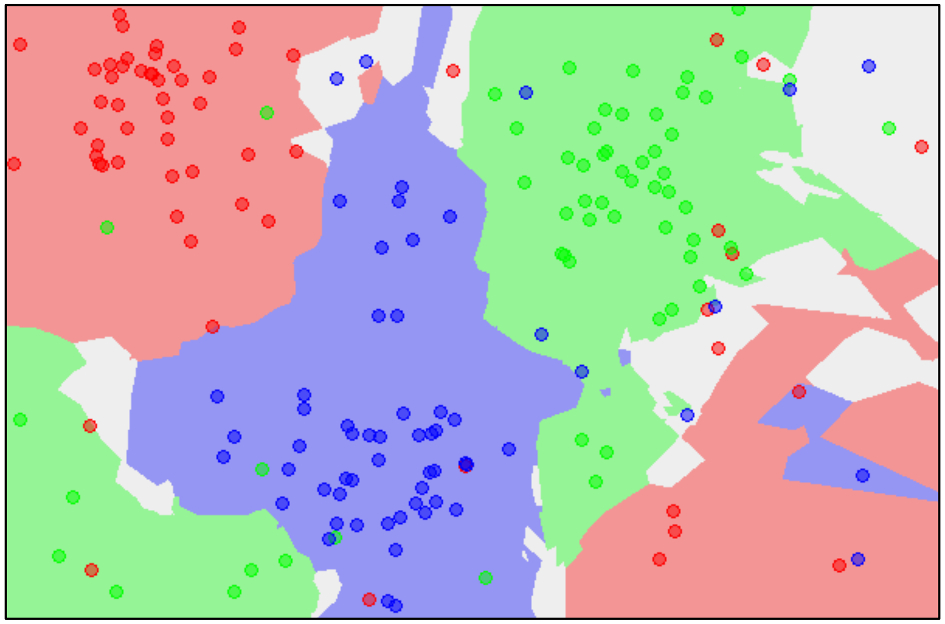


Diagram Credit: Wikipedia

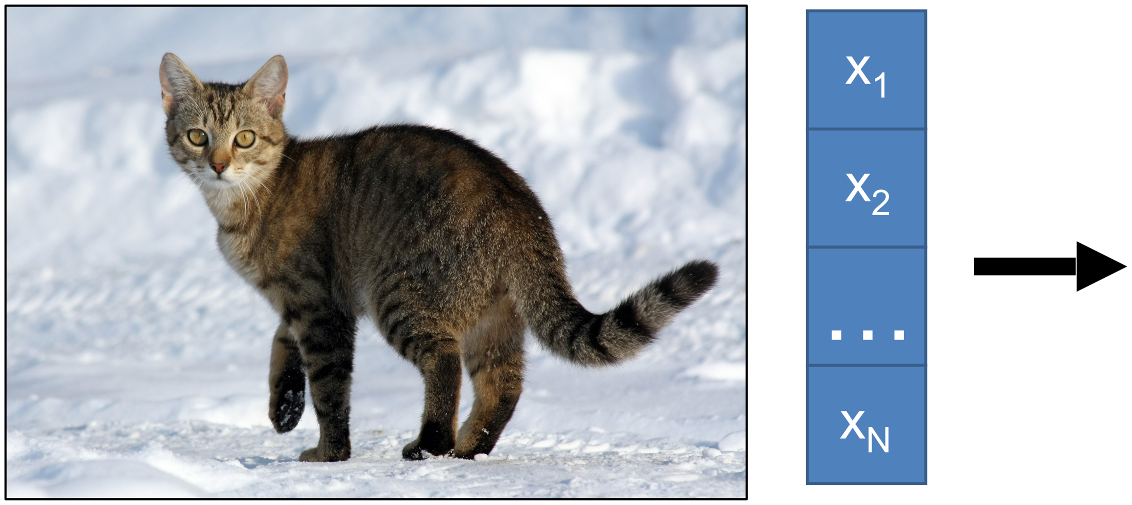
Credit: D. Fouhey

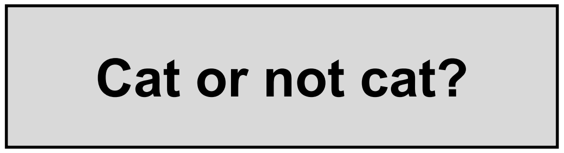
K-Nearest Neighbors

- No learning going on but usually effective
- Same algorithm for every task
- As number of datapoints → ∞, error rate is guaranteed to be at most 2x worse than optimal you could do on data

Credit: D. Fouhey

Alternative: recall last lecture





Classification by Least-Squares

Treat as regression: x_i is image feature; y_i is 1 if it's a cat, 0 if it's not a cat. Minimize least-squares loss.

Training (**X**,y): $$\arg\min_{\boldsymbol{w}} \underbrace{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2}_{\text{Loss}} + \underbrace{\lambda}_{\text{Trade-off}} \underbrace{\|\boldsymbol{w}\|_2^2}_{\text{Regularization}}$$

Inference (x): $$\boldsymbol{w}^T\boldsymbol{x} > t$$

Unprincipled in theory, but often effective in practice
The reverse (regression via discrete bins) is also common

Rifkin, Yeo, Poggio. *Regularized Least Squares Classification* (http://cbcl.mit.edu/publications/ps/rlsc.pdf). 2003
Redmon, Divvala, Girshick, Farhadi. *You Only Look Once: Unified, Real-Time Object Detection.* CVPR 2016.

Credit: D. Fouhey

Solving regularized least squares

Objective: $$\arg\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_2^2$$

$$\boxed{\text{Take } \frac{\partial}{\partial \boldsymbol{w}}, \text{ set to } \mathbf{0}, \text{ solve}}$$

$$\boldsymbol{w}^* = \underbrace{\left(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}\right)}^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

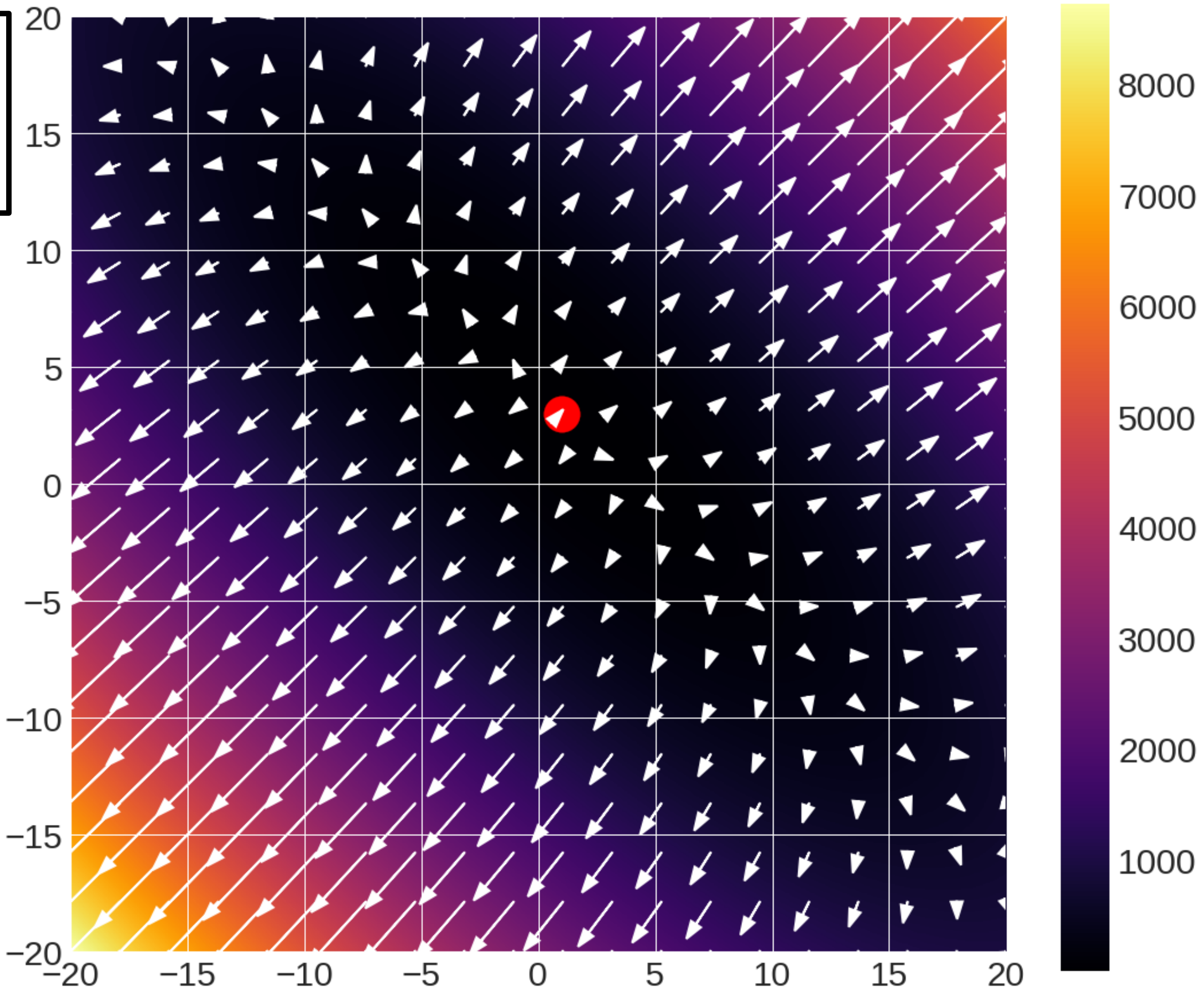
$\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}$ is full-rank (and thus invertible) for $\lambda>0$

Called *lots of things:* regularized least-squares, Tikhonov regularization (after Andrey Tikhonov), ridge regression, Bayesian linear regression with a multivariate normal prior.

Credit: D. Fouhey

But really, use the gradient

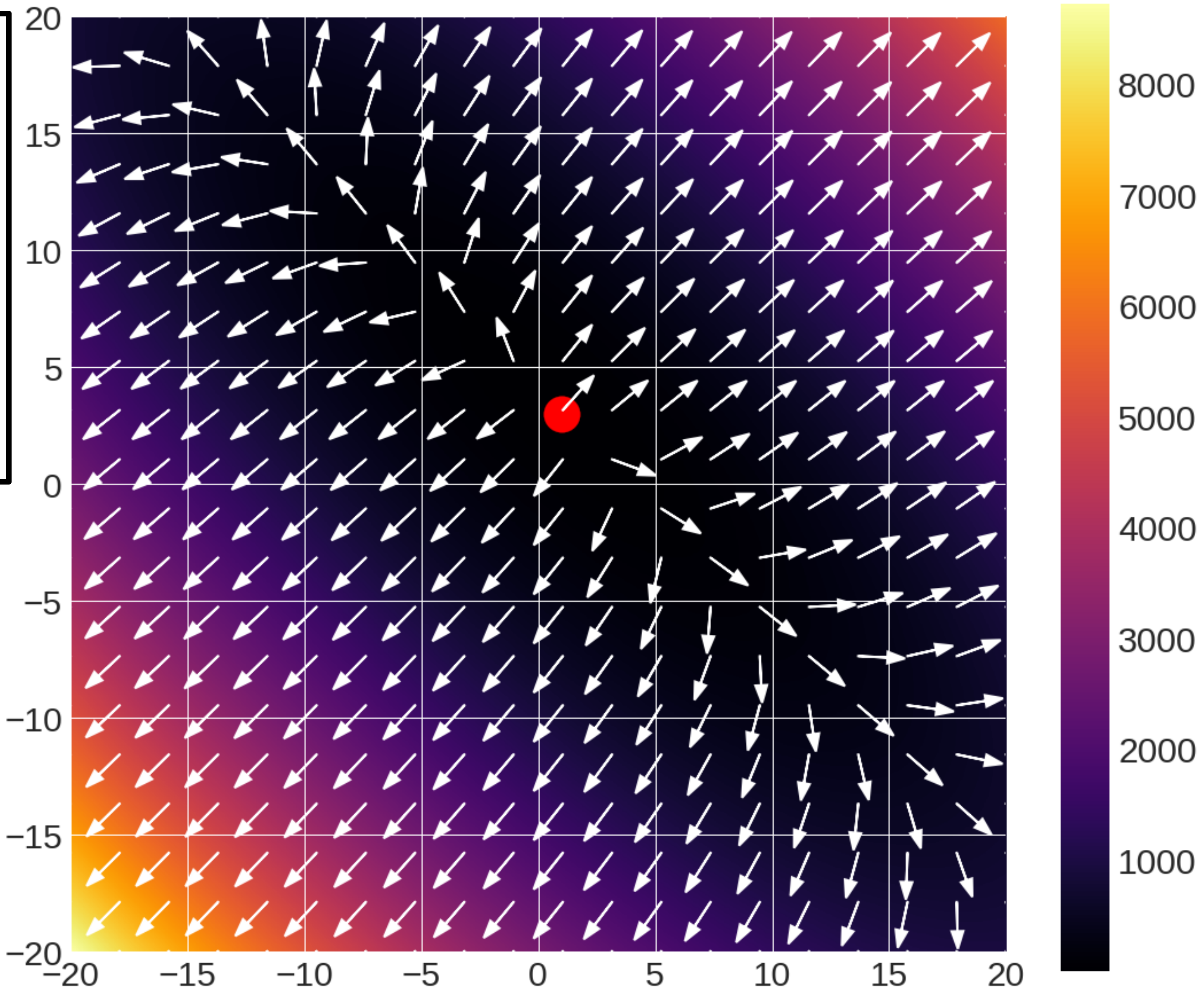
Arrows:
gradient



Credit: D. Fouhey

Use the gradient

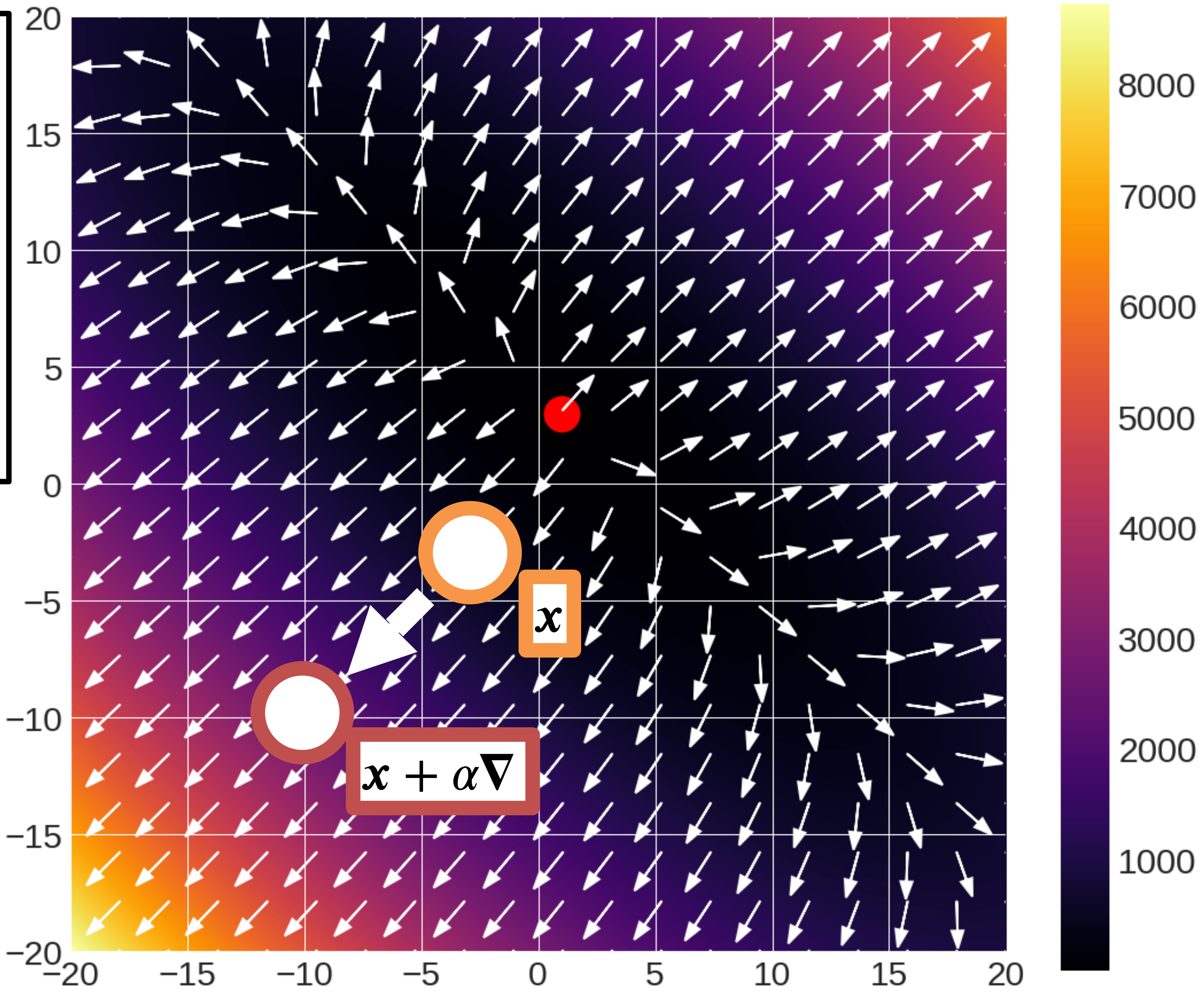
Arrows:
gradient direction
(scaled to unit length)



Credit: D. Fouhey

Use the gradient

Arrows: gradient direction (scaled to unit length)

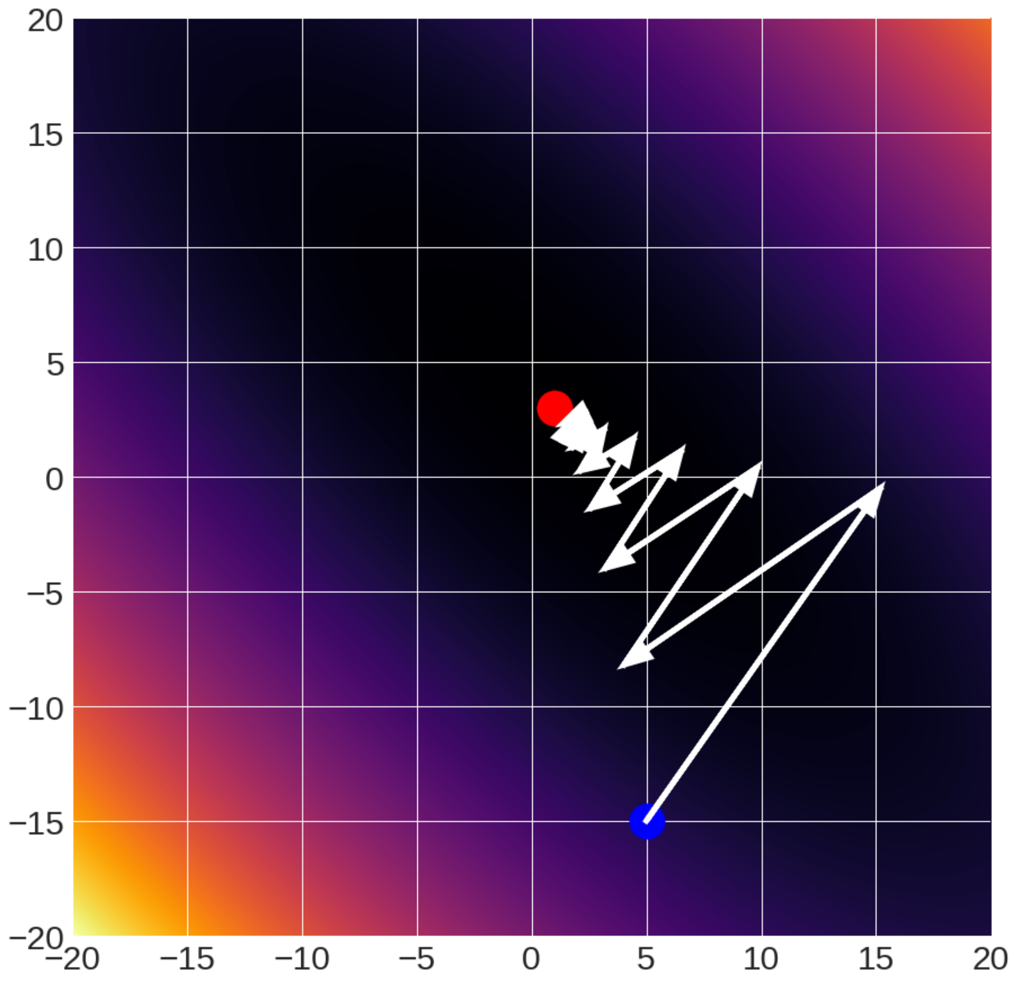


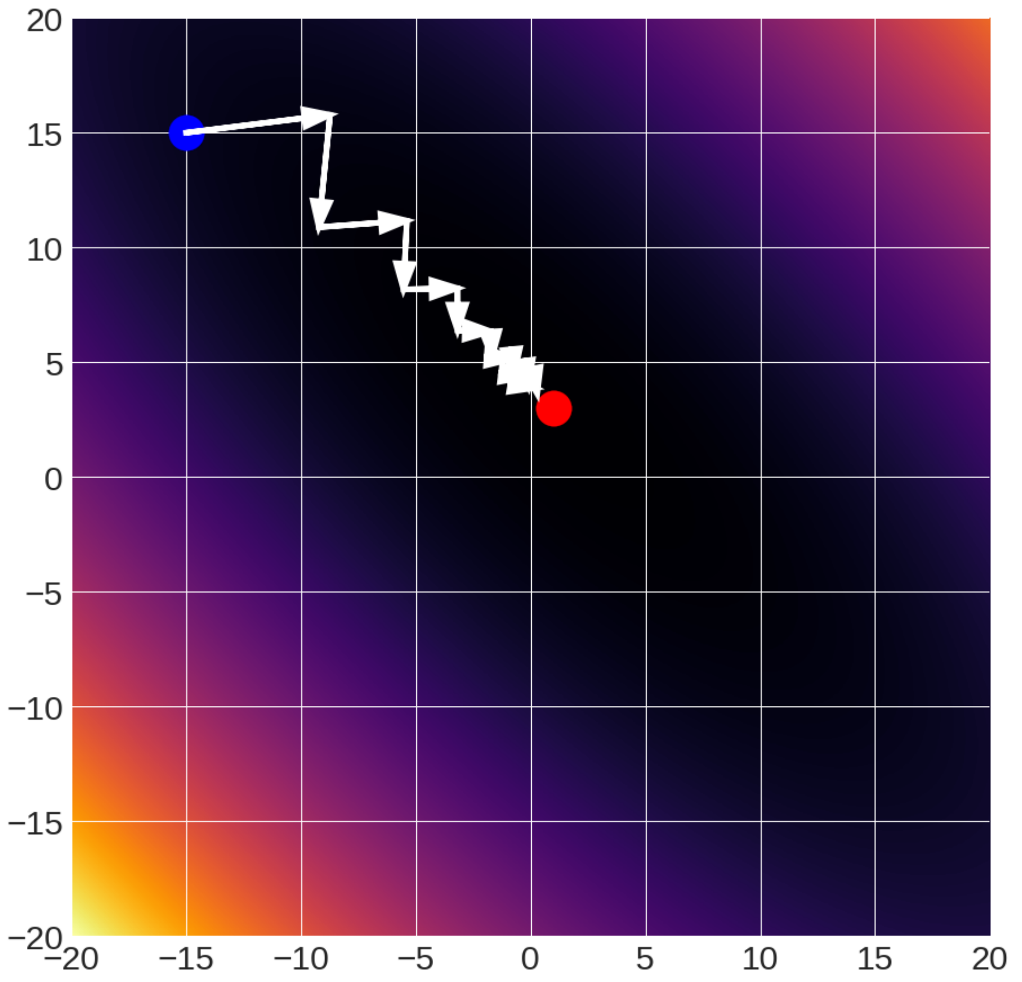
Credit: D. Fouhey

Gradient descent

Given starting point (blue)

$w_{i+1} = w_i + -9.8 \times 10^{-2} \times \text{gradient}$

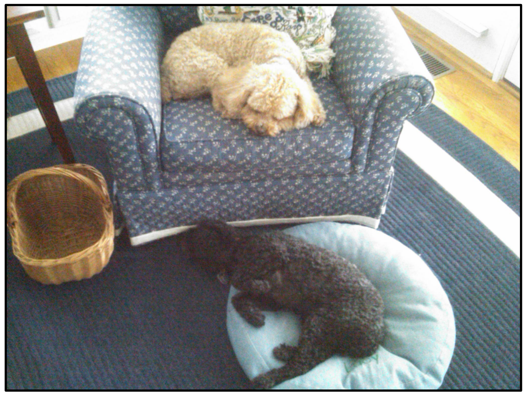




Credit: D. Fouhey

Linear Models

Example Setup: 3 classes



Model – one weight per class: $\boldsymbol{w}_0, \boldsymbol{w}_1, \boldsymbol{w}_2$

$\boldsymbol{w}_0^T \boldsymbol{x}$ big if cat

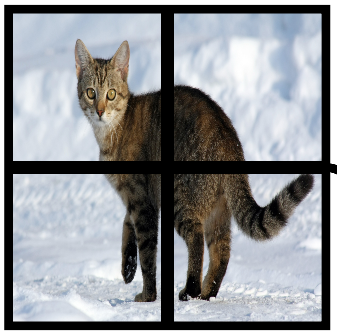
$\boldsymbol{w}_1^T \boldsymbol{x}$ big if dog

$\boldsymbol{w}_2^T \boldsymbol{x}$ big if hippo

Stack together: $\boldsymbol{W}_{3xF}$ where **x** is in R^F

Credit: D. Fouhey

Linear Models



$$
\begin{array}{r|ccccc|}
\text{Cat weight vector} & 0.2 & -0.5 & 0.1 & 2.0 & 1.1 \\
\text{Dog weight vector} & 1.5 & 1.3 & 2.1 & 0.0 & 3.2 \\
\text{Hippo weight vector} & 0.0 & 0.3 & 0.2 & -0.3 & -1.2
\end{array}
\quad
\begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \\ 1 \end{bmatrix}
\rightarrow
\begin{array}{|c|l}
-96.8 & \text{Cat score} \\
437.9 & \text{Dog score} \\
61.95 & \text{Hippo score}
\end{array}
$$

$\boldsymbol{W}$ $\quad$ $\boldsymbol{x_i}$ $\quad$ $\boldsymbol{Wx_i}$

Weight matrix a collection of scoring functions, one per class

Prediction is vector where jth component is “score” for jth class.

Diagram by: Karpathy, Fei-Fei

Credit: D. Fouhey

Geometric Intuition

What does a linear classifier look like in 2D?

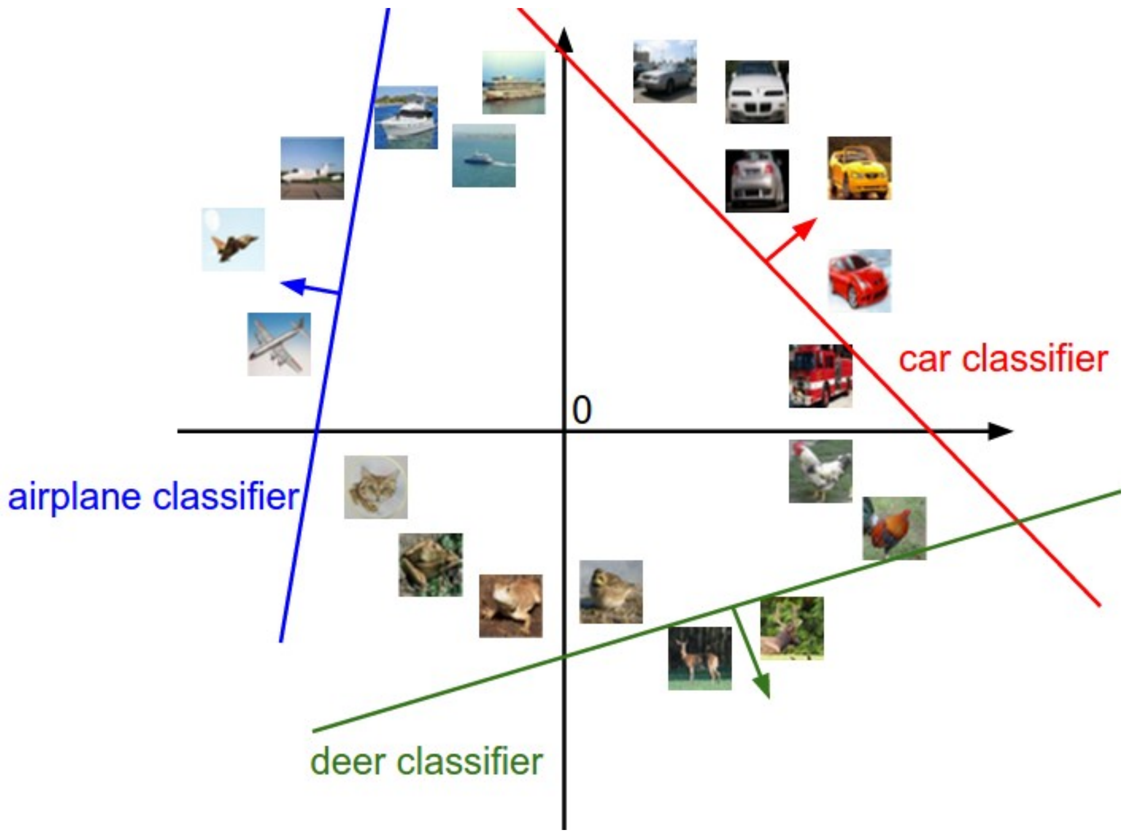
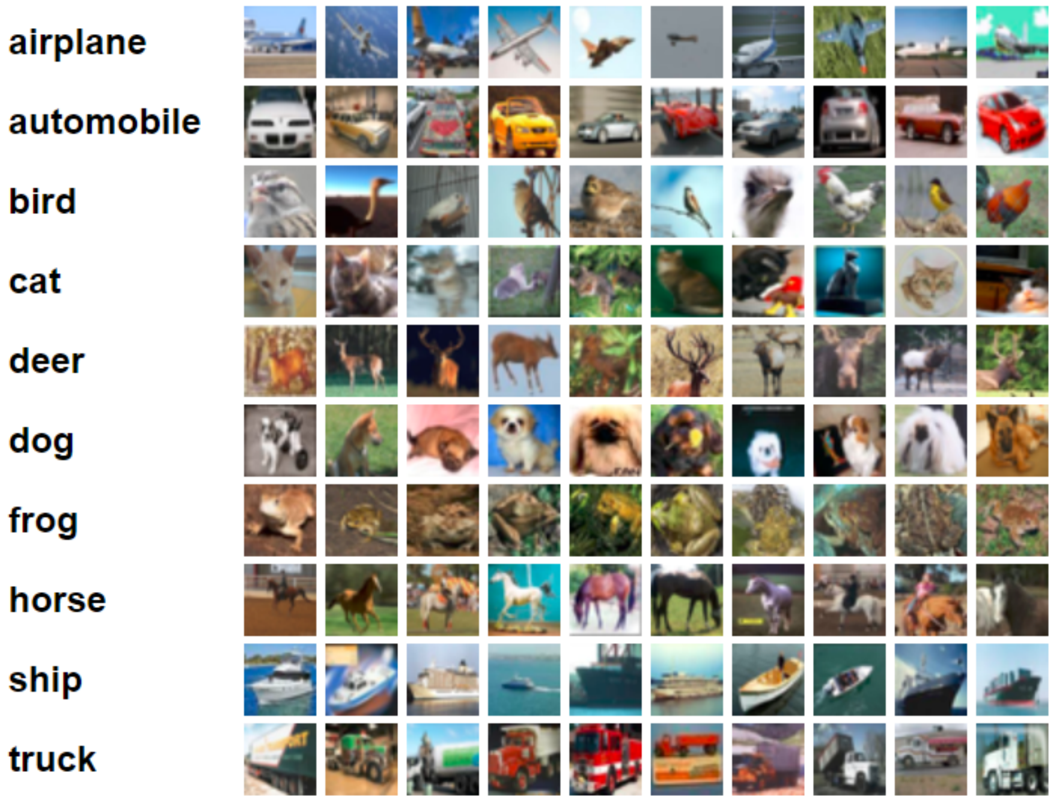


Diagram credit: Karpathy & Fei-Fei

Credit: D. Fouhey

Visual Intuition

CIFAR 10:
32x32x3 Images, 10 Classes



- Turn each image into feature by unrolling all pixels
- Fit 10 linear models

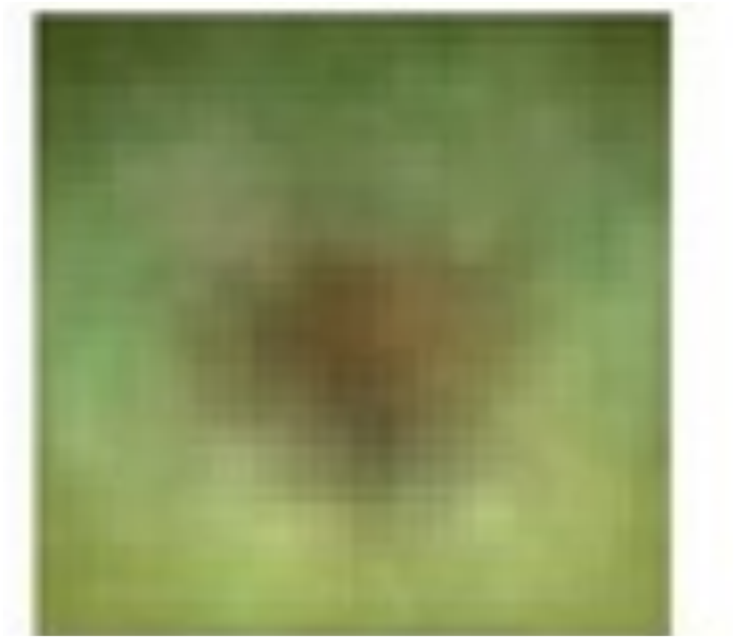
Credit: D. Fouhey

Slide credit: Karpathy & Fei-Fei

Guess The Classifier

Decision rule is $\mathbf{w}^T\mathbf{x}$. If $\mathbf{w}_i$ is big, then big values of x_i are indicative of the class.

Deer or Plane?

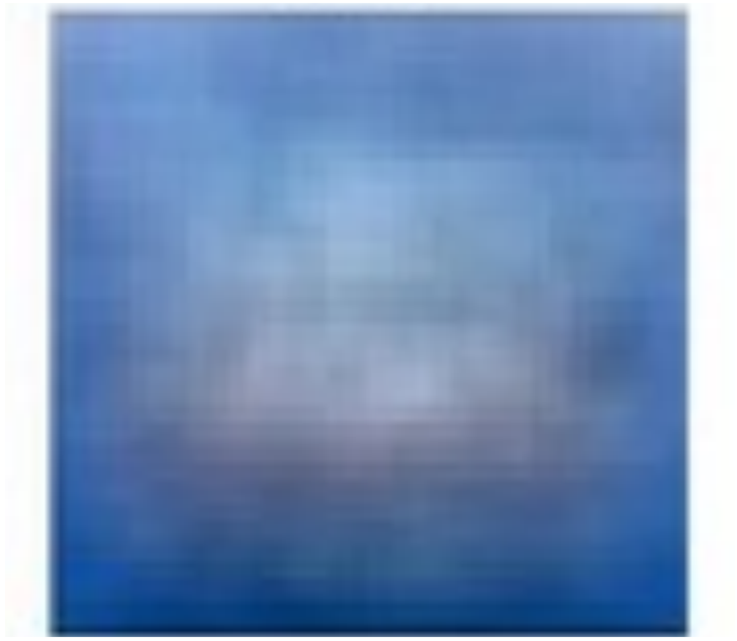


Credit: D. Fouhey Diagram credit: Karpathy & Fei-Fei

Guess The Classifier

Decision rule is $\mathbf{w}^T\mathbf{x}$. If $\mathbf{w}_i$ is big, then big values of x_i are indicative of the class.

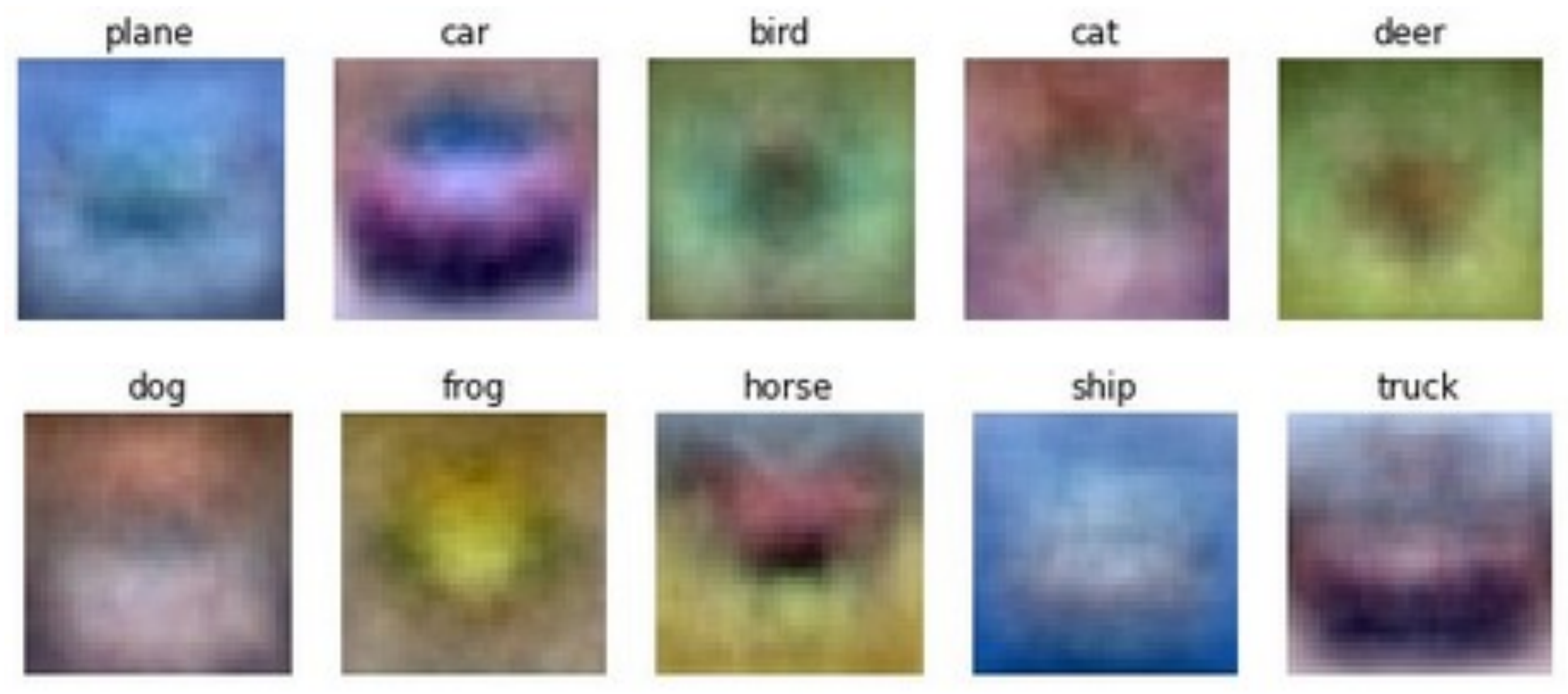
Ship or Dog?



Credit: D. Fouhey   Diagram credit: Karpathy & Fei-Fei

Interpreting a Linear Classifier

Decision rule is $\mathbf{w}^T\mathbf{x}$. If $\mathbf{w}_i$ is big, then big values of x_i are indicative of the class.



Credit: D. Fouhey Diagram credit: Karpathy & Fei-Fei

Going forward

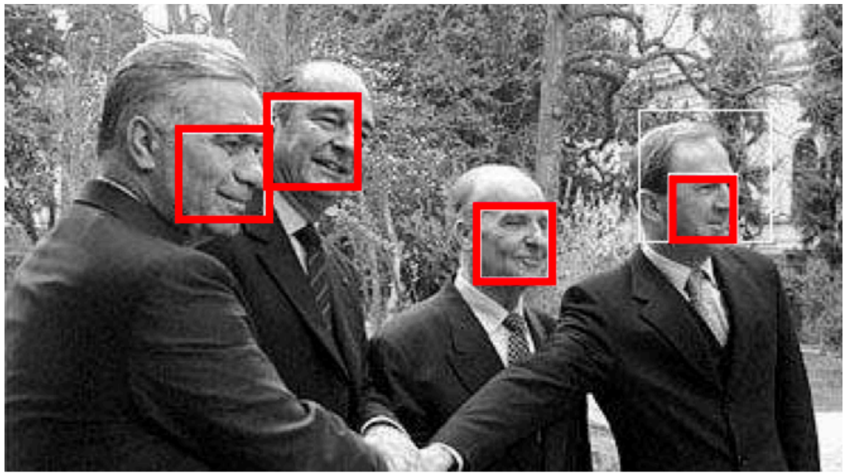
- Consider several computer vision tasks:
  - Today: face detection, pedestrian detection
  - Lectures 9,10: image classification, generic object detection
  - Lectures 11,12: segmentation, texture
- For each one:
  - Discuss **use cases, data, evaluation**
  - Key method: **representation, model, learning**
  - Results: **quantitative**, **qualitative**
- Will revisit many of these tasks in Module 4 on deep learning

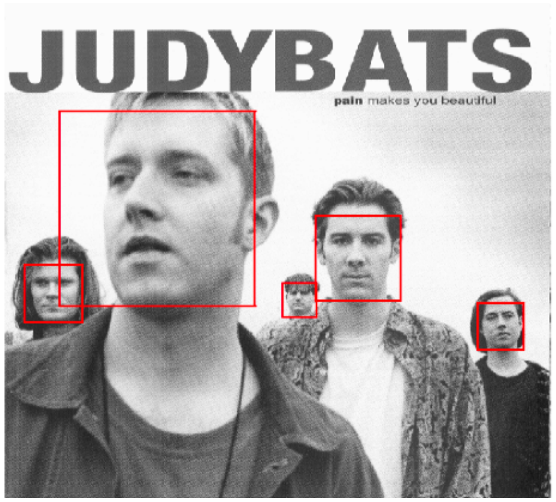
Midterm

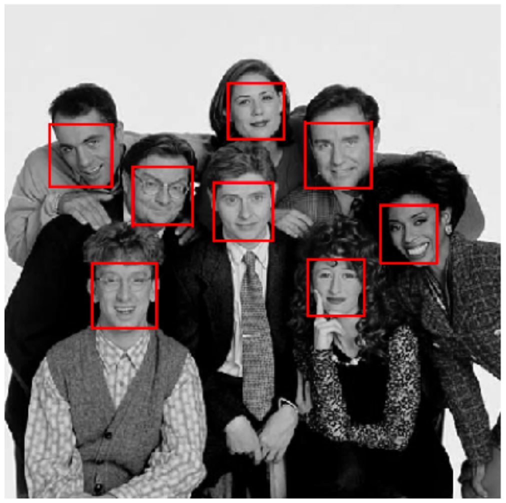
- Everything through lecture 11 on Thursday, Oct 17th
- Will need to be able to articulate the connection between the design decisions (image representation, evaluation criteria, modeling architecture) and the outcome (expected sources of errors, comparative performance)

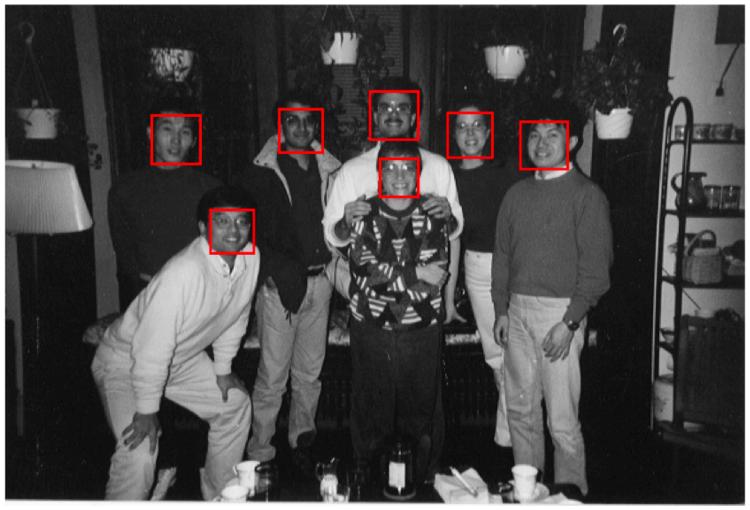
Face detection

Face detection in cameras





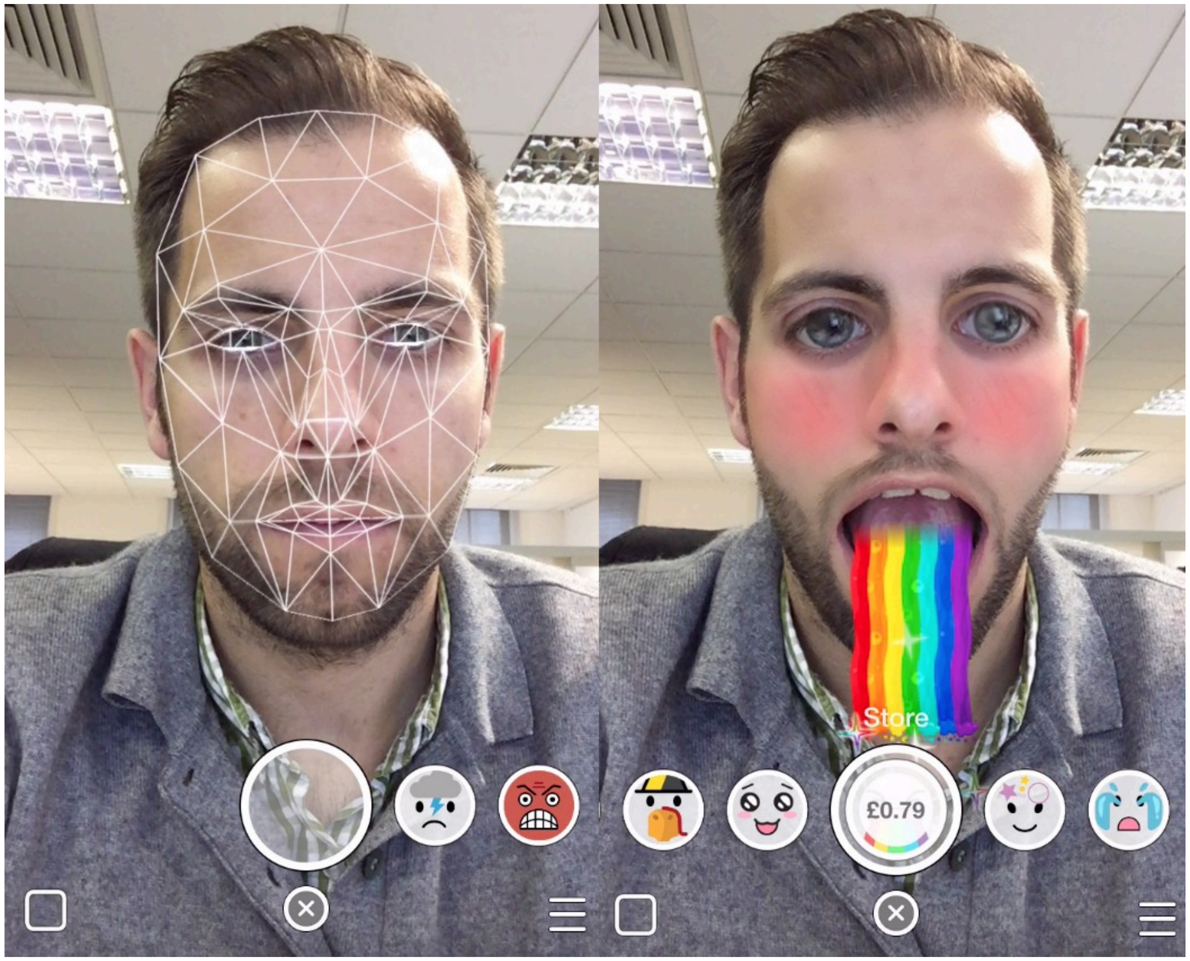




FinePix S6000fd, by Fujifilm, 2006

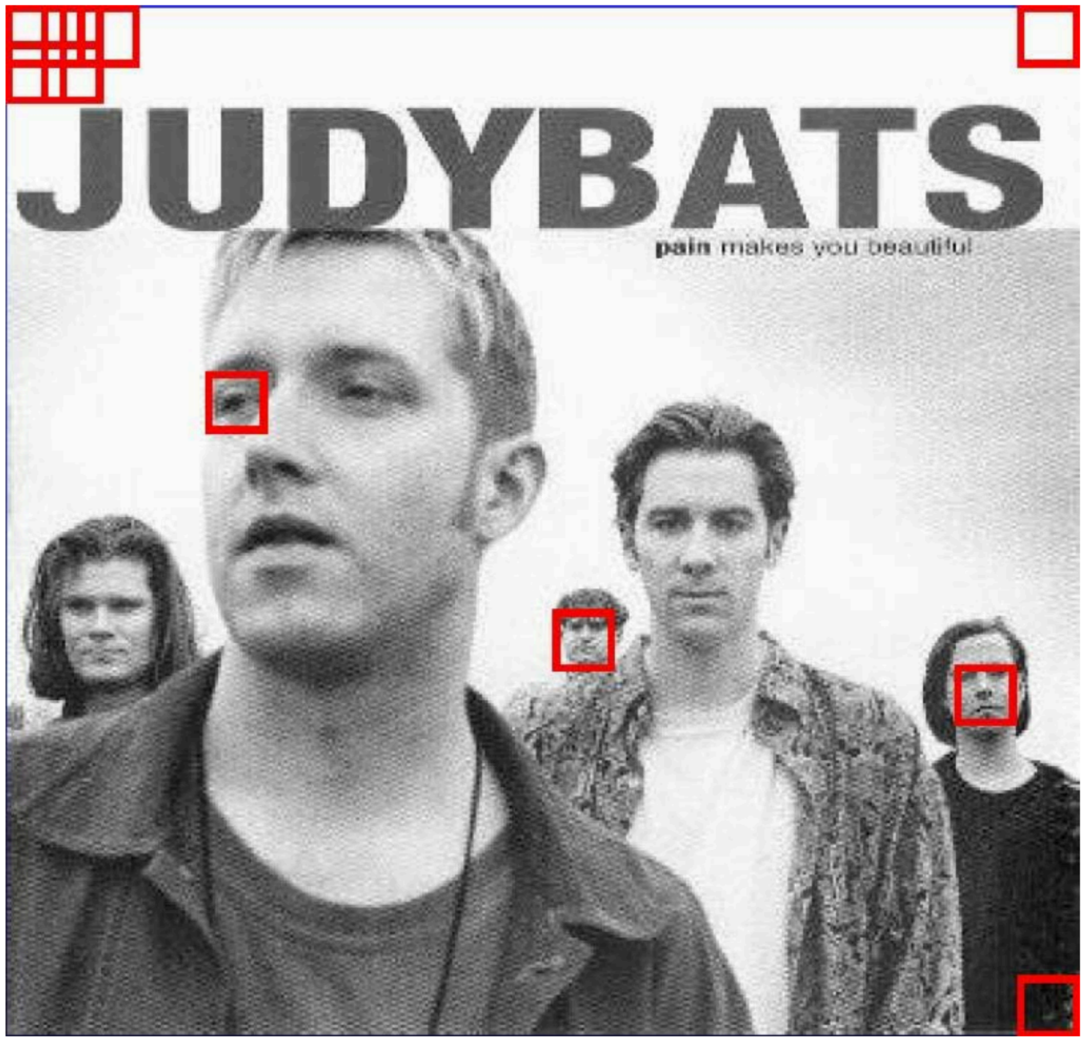
Viola & Jones, 2001

E.g., Snapchat



https://medium.com/@anidaro/how-snapchats-filters-work-86973c3e2e9f

Search over scale and space

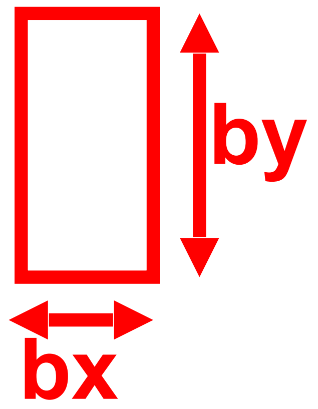


How many sliding window boxes are there?

Given a HxW image and a “template” of size by, bx.

Q. How many sub-boxes are there of size (by,bx)?

A. (H-by)*(W-bx)



This is before considering adding:

- *scales* (by*s,bx*s)
- *aspect ratios* (by*sy,bx*sx)
  - (although not for face detection)

Challenges of face detection

- Sliding window detector must evaluate tens of thousands of location/scale combinations
- Faces are rare: 0–10 per image
  - A megapixel image has ~10^6 pixels and a comparable number of candidate face locations
  - For computational efficiency, we should try to spend as little time as possible on the non-face windows
  - To avoid having a false positive in every image, our false positive rate has to be less than 10^{-6}

Source: S. Lazebnik

The Viola/Jones Face Detector

- A seminal approach to real-time object detection
- Training is slow, but detection is very fast
- Key ideas
  - *Integral images* for fast feature evaluation
  - *Boosting* for feature selection
  - *Attentional cascade* for fast rejection of non-face windows

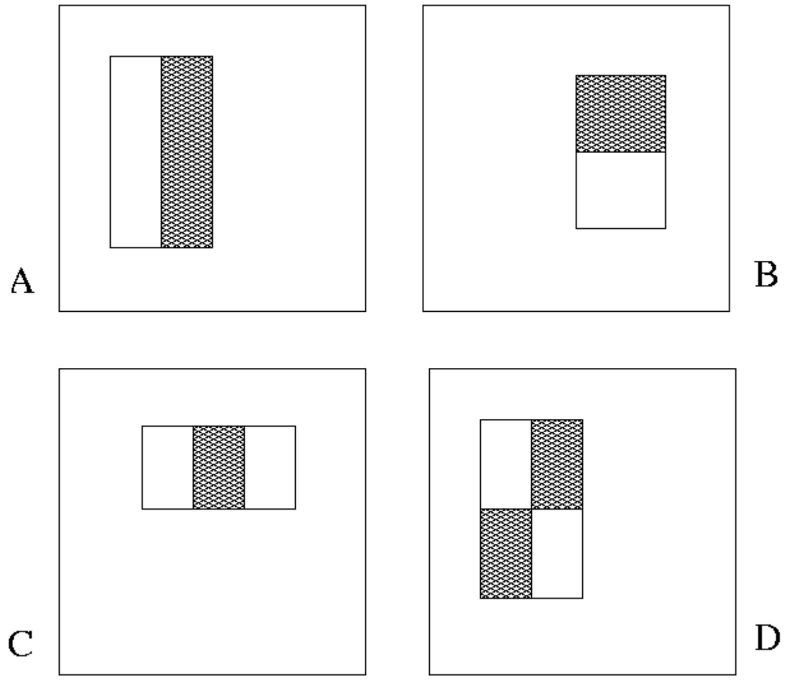
P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features.* CVPR 2001.

P. Viola and M. Jones. *Robust real-time face detection.* IJCV 57(2), 2004.

Source: S. Lazebnik

Image Features

"Rectangle filters"



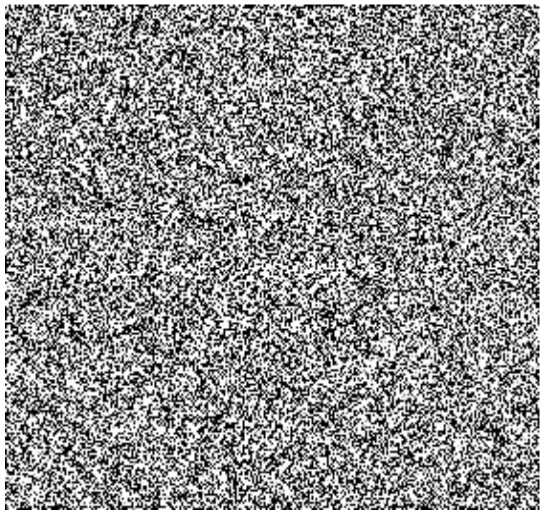
Value =

∑ *(pixels in white area)* –
∑ *(pixels in black area)*

Source: S. Lazebnik

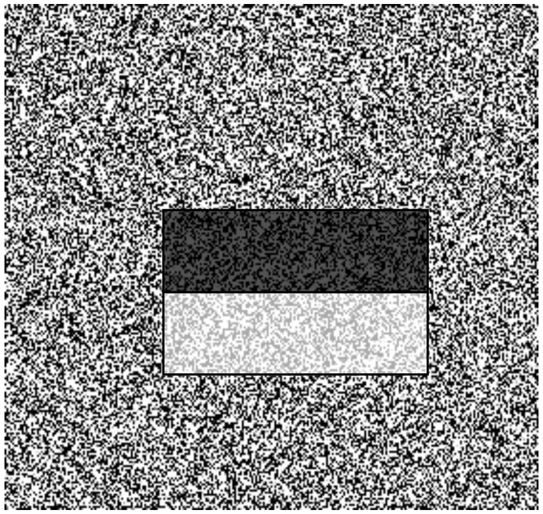
Example

Source

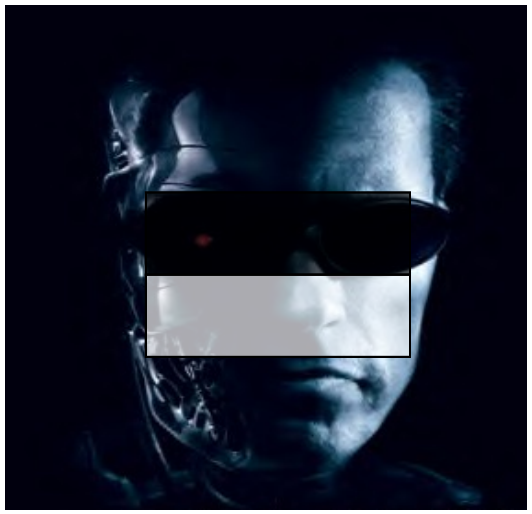


Result

✗



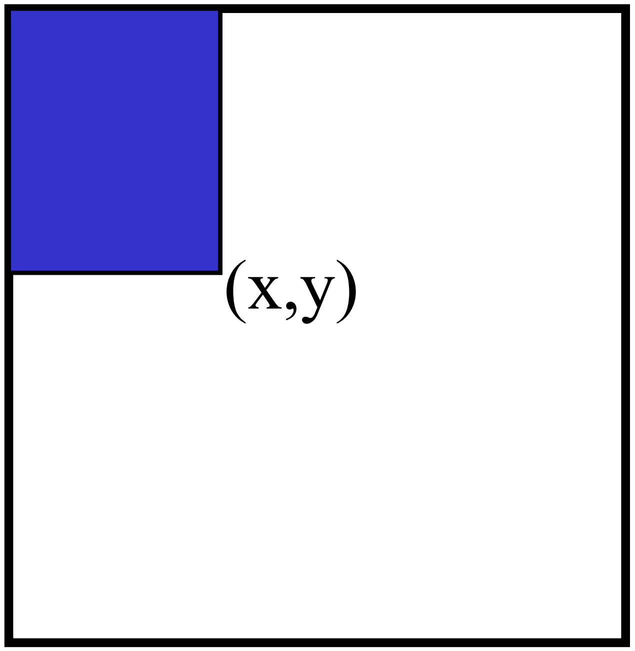
✓



Source: S. Lazebnik

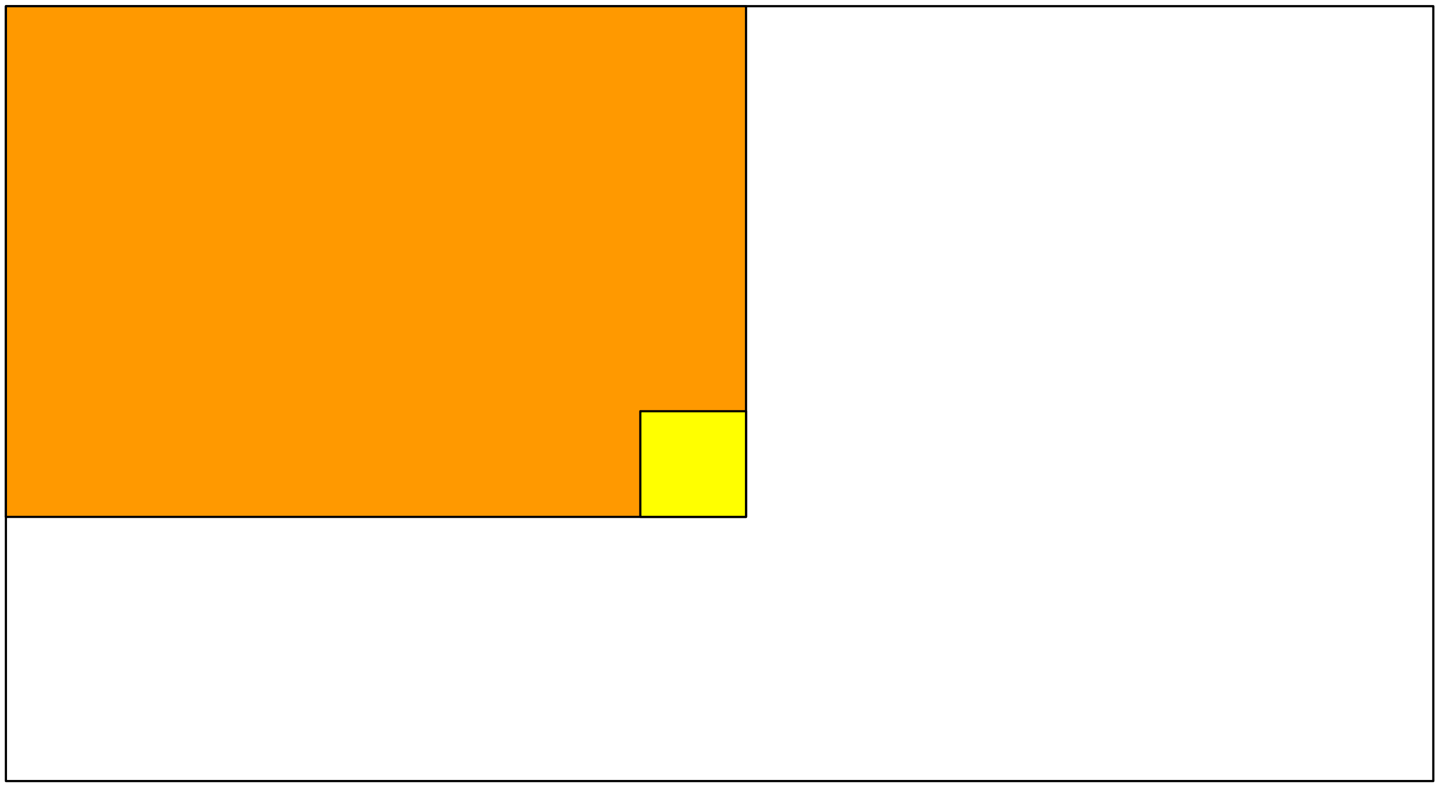
Key idea 1: Fast computation with integral images

- The *integral image* computes a value at each pixel (*x*,*y*) that is the sum of the pixel values above and to the left of (*x*,*y*), inclusive
- This can quickly be computed in one pass through the image



Source: S. Lazebnik

Computing the integral image

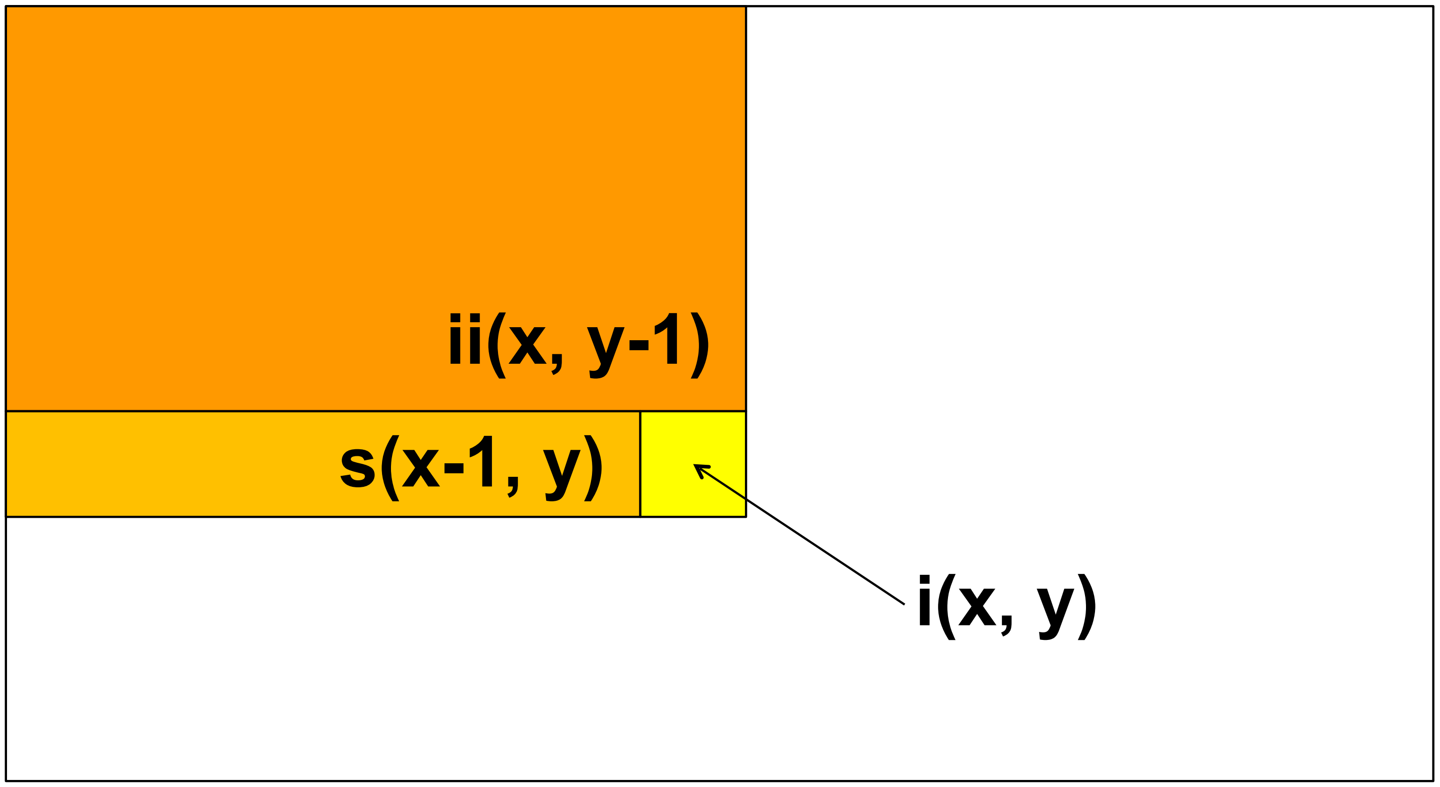


Source: S. Lazebnik

Computing the integral image

Cumulative row sum: $s(x, y) = s(x-1, y) + i(x, y)$

Integral image: $ii(x, y) = ii(x, y-1) + s(x, y)$

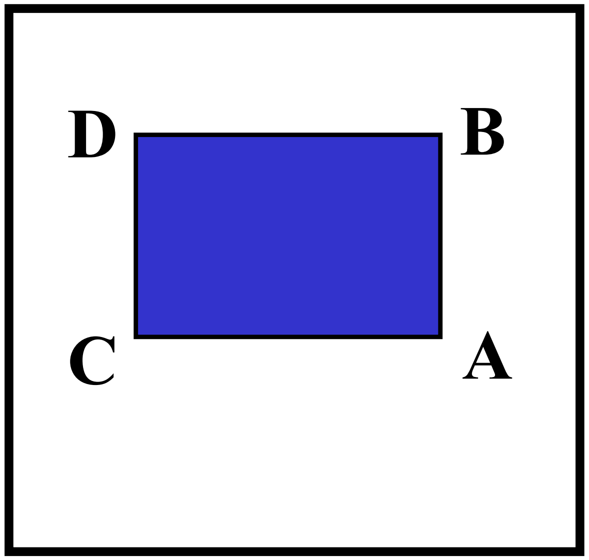


Source: S. Lazebnik

Computing sum within a rectangle

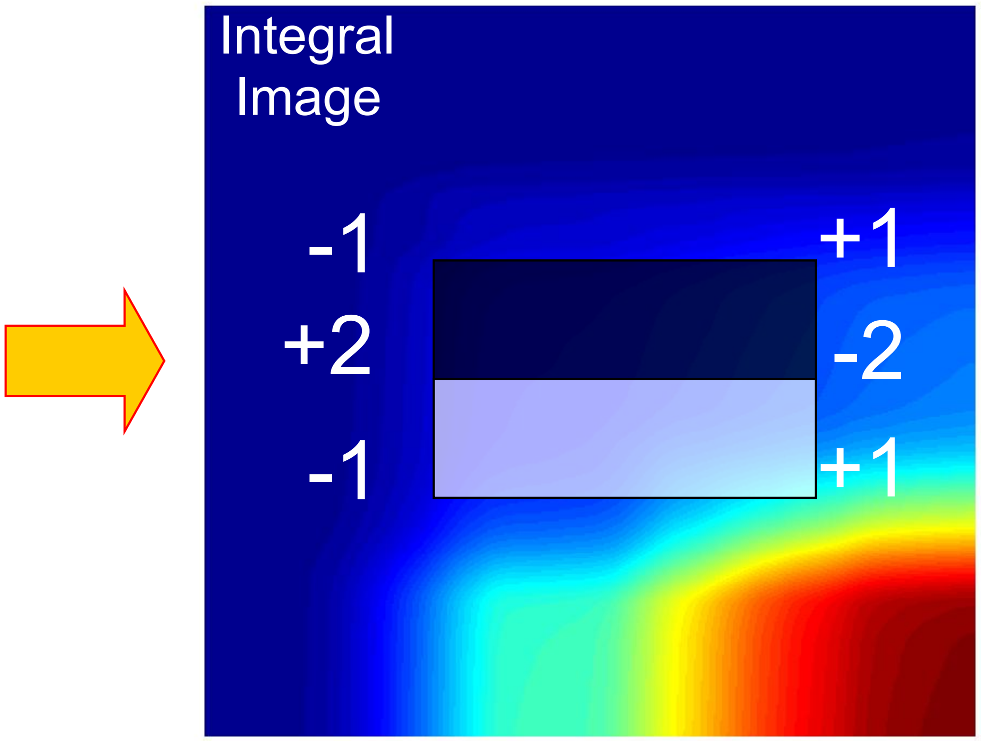
- Let A,B,C,D be the values of the integral image at the corners of a rectangle
- Then the sum of original image values within the rectangle can be computed as:

  sum = A – B – C + D
- Only 3 additions are required for any size of rectangle!



Source: S. Lazebnik

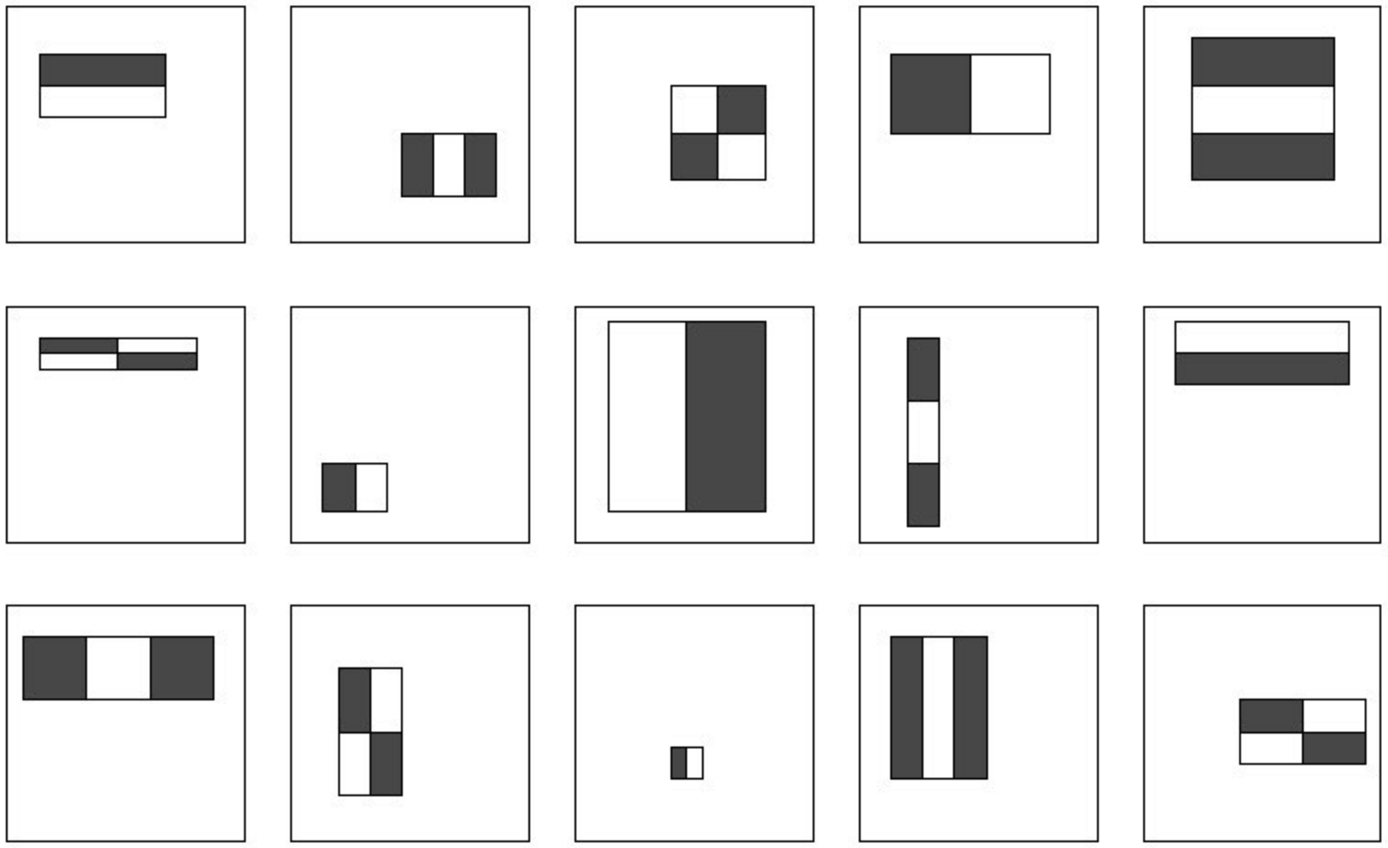
Computing a rectangle feature



Source: S. Lazebnik

Feature selection

- For a 24x24 detection region, the number of possible rectangle features is ~160,000!



Source: S. Lazebnik

Feature selection

- For a 24x24 detection region, the number of possible rectangle features is ~160,000!
- At test time, it is impractical to evaluate the entire feature set
- Can we create a good classifier using just a small subset of all possible features?
- How to select such a subset?

Source: S. Lazebnik

Key idea 2: Boosting

- *Boosting* is a classification scheme that combines *weak learners* into a more accurate *ensemble classifier*
- Weak learners based on rectangle filters:

$$\underbrace{h_t(x)}_{\text{window}} = \begin{cases} 1 & \text{if } p_t \overset{\text{value of rectangle feature}}{f_t(x)} > \underset{\text{parity}}{p_t} \underset{\text{threshold}}{\theta_t} \\ 0 & \text{otherwise} \end{cases}$$

- Ensemble classification function:

$$C(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \alpha_t h_t(x) > \frac{1}{2} \sum_{t=1}^{T} \alpha_t \longleftarrow \text{learned weights} \\ 0 & \text{otherwise} \end{cases}$$

Source: S. Lazebnik

Training procedure

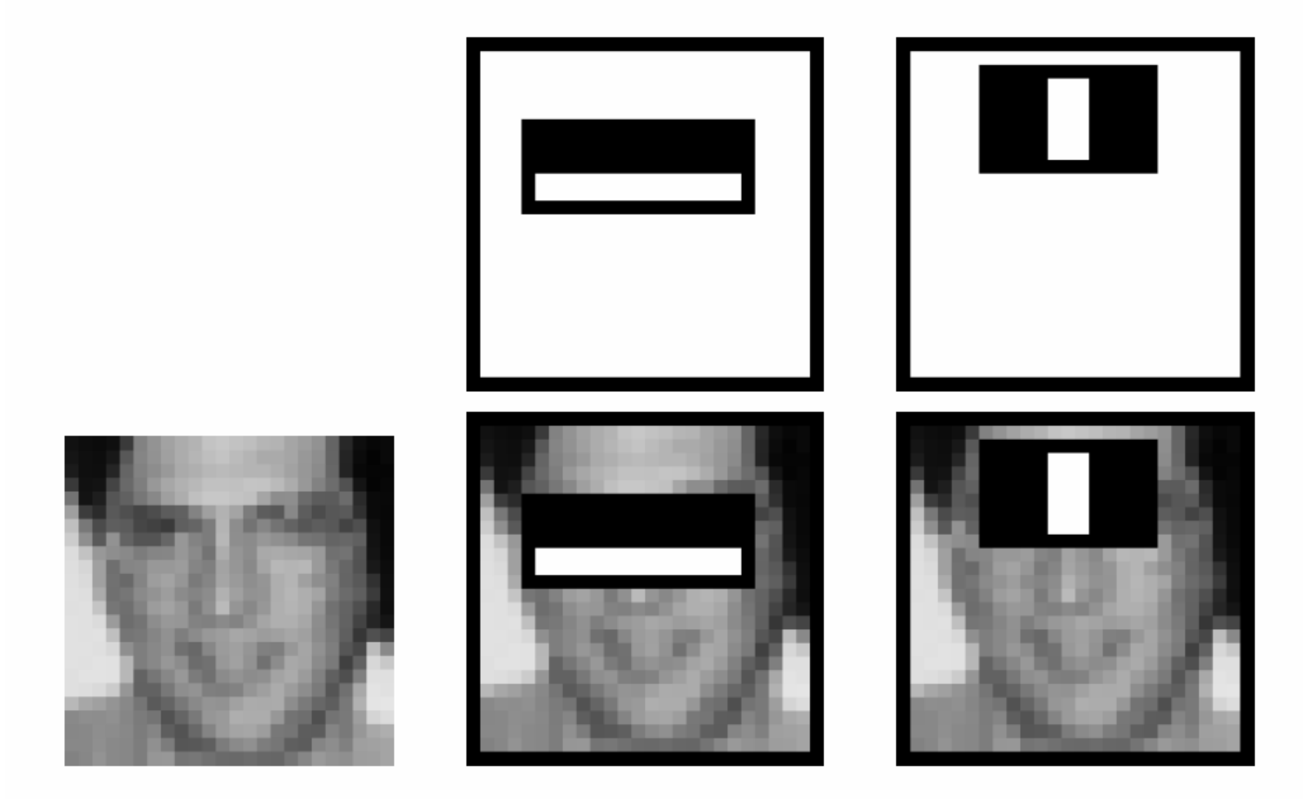
- Initially, weight each training example equally
- In each boosting round:
    - Find the weak learner that achieves the lowest *weighted* training error
    - Raise the weights of training examples misclassified by current weak learner
- Compute final classifier as linear combination of all weak learners (weight of each learner is directly proportional to its accuracy)
    - Exact formulas for re-weighting and combining weak learners depend on the particular boosting scheme (e.g., AdaBoost)

Y. Freund and R. Schapire, A short introduction to boosting, *Journal of Japanese Society for Artificial Intelligence*, 14(5):771-780, September, 1999.

Source: S. Lazebnik

Boosting for face detection

- First two features selected by boosting:



This feature combination can yield 100% detection rate and 50% false positive rate

Source: S. Lazebnik

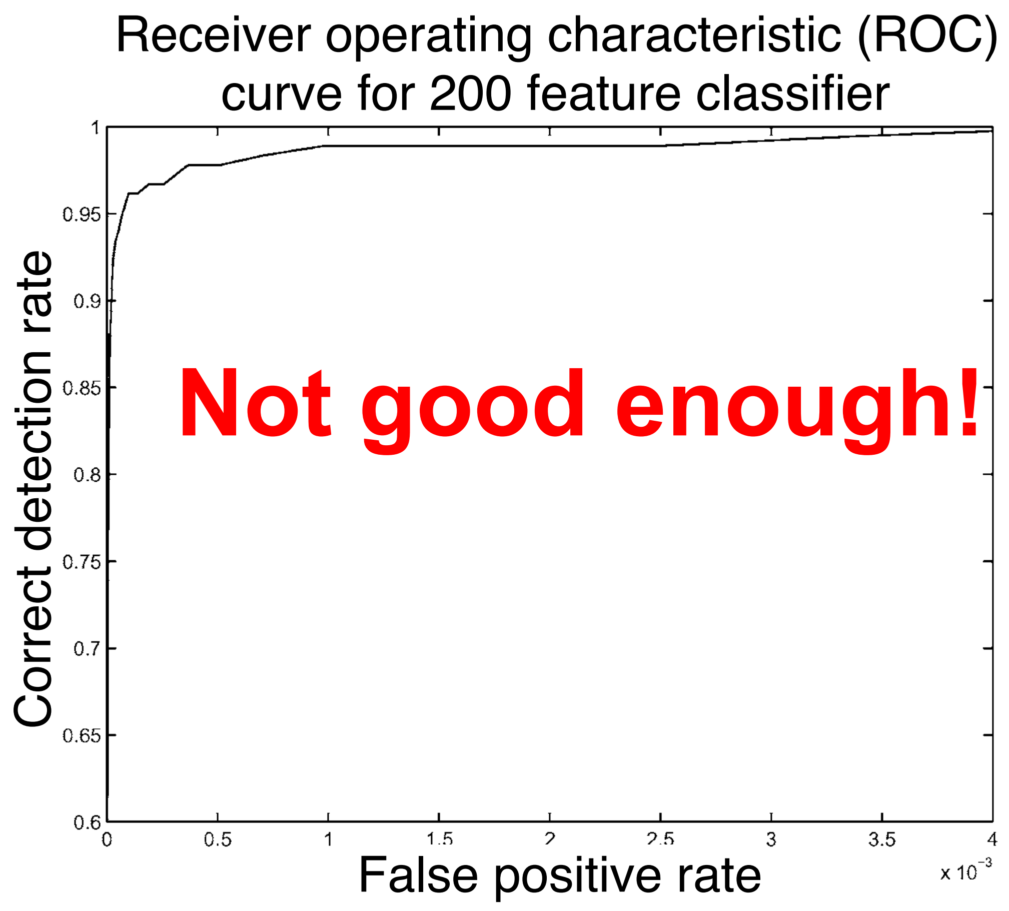
Boosting

- Advantages of boosting
  - Flexibility in the choice of weak learners, boosting scheme
  - Testing is fast
  - Easy to implement
- Disadvantages
  - Needs many training examples
  - Training is slow

Source: S. Lazebnik

Boosting for face detection

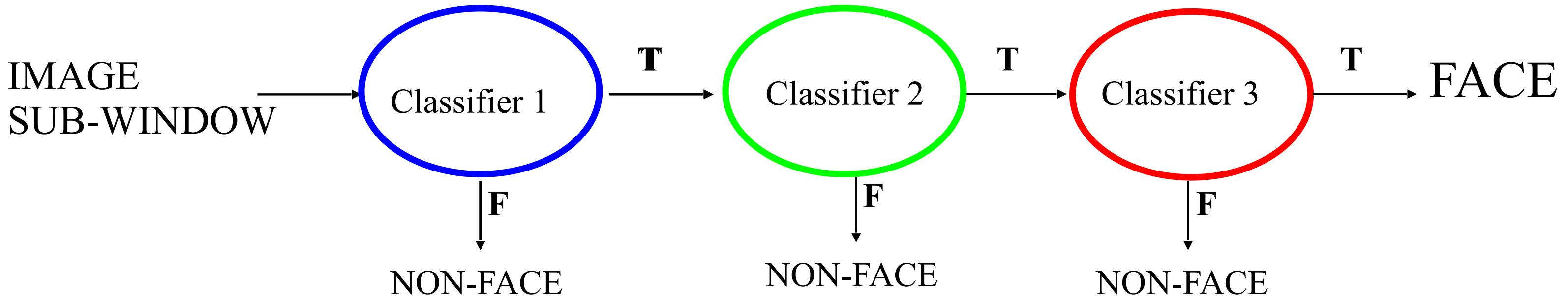
- A 200-feature classifier can yield 95% detection rate and a false positive rate of 1 in 14084



Source: S. Lazebnik

Key idea 3: Attentional cascade

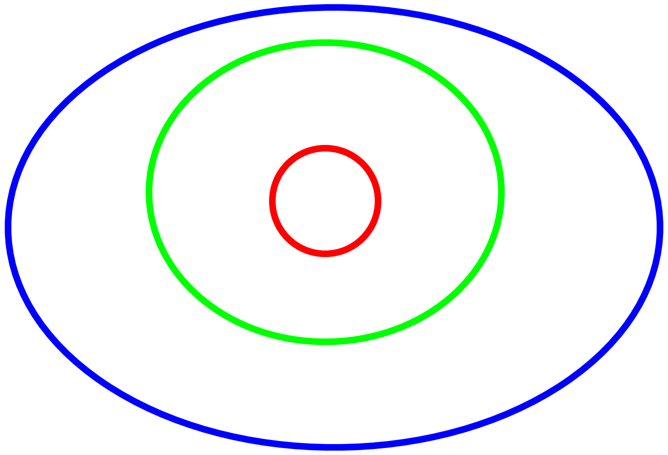
- We start with simple classifiers which reject many of the negative sub-windows while detecting almost all positive sub-windows
- Positive response from the first classifier triggers the evaluation of a second (more complex) classifier, and so on
- A negative outcome at any point leads to the immediate rejection of the sub-window



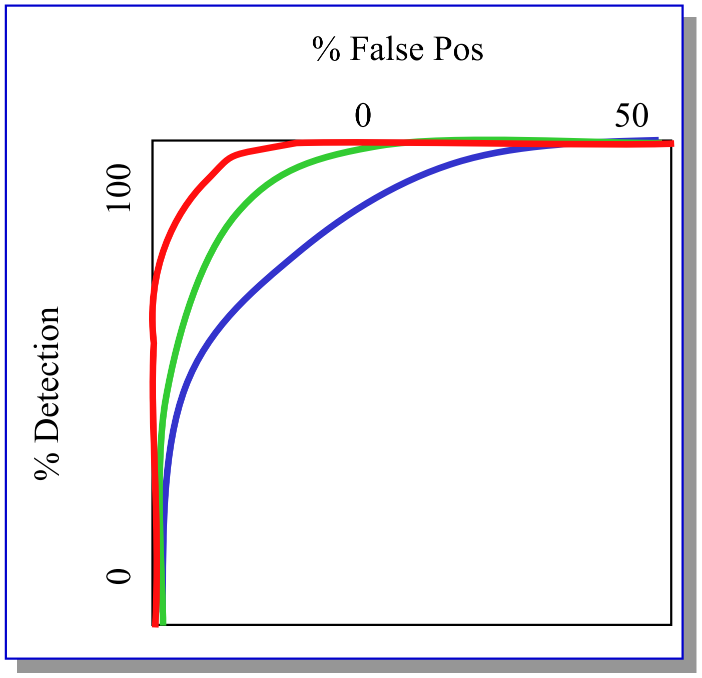
Source: S. Lazebnik

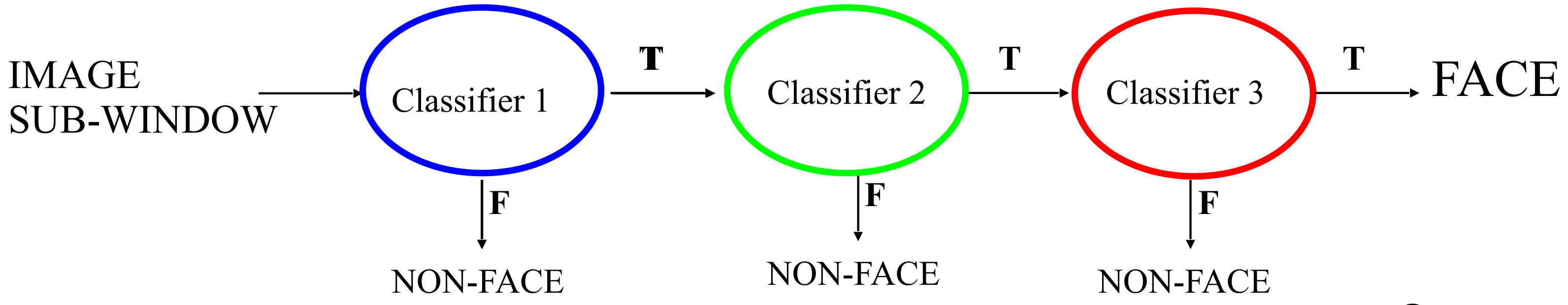
Attentional cascade

- Chain classifiers that are progressively more complex and have lower false positive rates:



Receiver operating characteristic





Source: S. Lazebnik

Training the cascade

- Set target detection and false positive rates for each stage
- Keep adding features to the current stage until its target rates have been met
  - Need to lower AdaBoost threshold to maximize detection (as opposed to minimizing total classification error)
  - Test on a *validation set*
- If the overall false positive rate is not low enough, then add another stage
- Use false positives from current stage as the negative training examples for the next stage

Source: S. Lazebnik

The implemented system

- Training Data
  - 5000 faces
    - All frontal, rescaled to 24x24 pixels
  - 300 million non-faces
    - 9500 non-face images
  - Faces are normalized
    - Scale, translation
- Many variations
  - Across individuals
  - Illumination
  - Pose

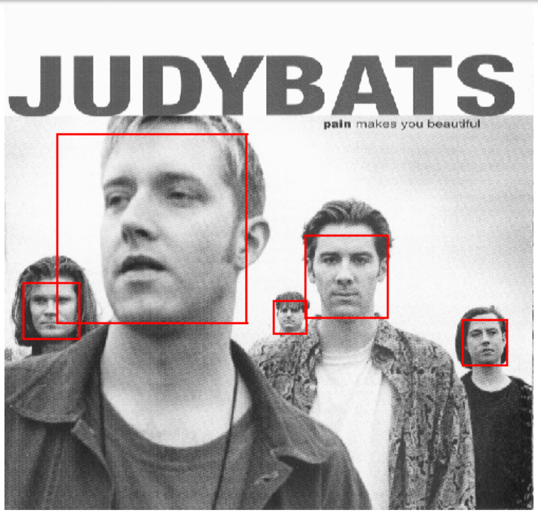
Source: S. Lazebnik

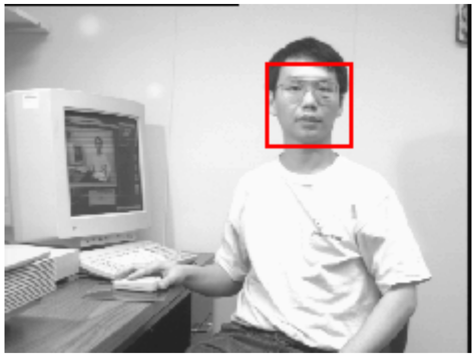
System performance

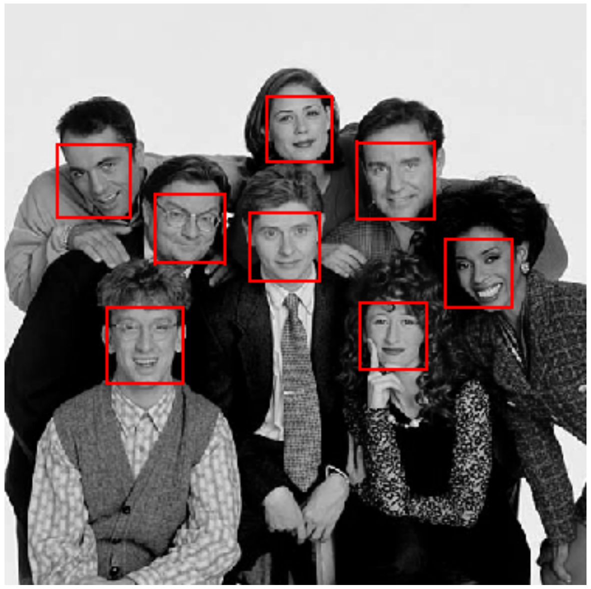
- Training time: “weeks” on 466 MHz Sun workstation
- 38 layers, total of 6061 features
- Average of 10 features evaluated per window on test set
- “On a 700 Mhz Pentium III processor, the face detector can process a 384 by 288 pixel image in about .067 seconds”
    - 15 Hz
    - 15 times faster than previous detector of comparable accuracy (Rowley et al., 1998)

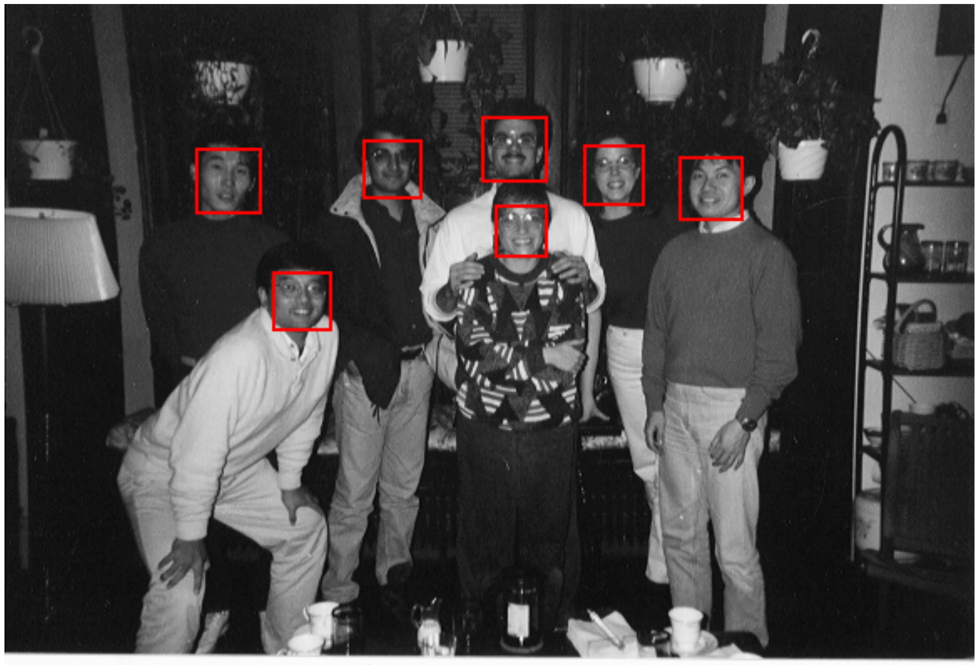
Source: S. Lazebnik

Output of Face Detector on Test Images









Source: S. Lazebnik

Summary: Viola/Jones detector

- Rectangle features
- Integral images for fast computation
- Boosting for feature selection
- Attentional cascade for fast rejection of negative windows

- Caveat: Haar features work very well on structured faces, but much worse on other objects (coming up)

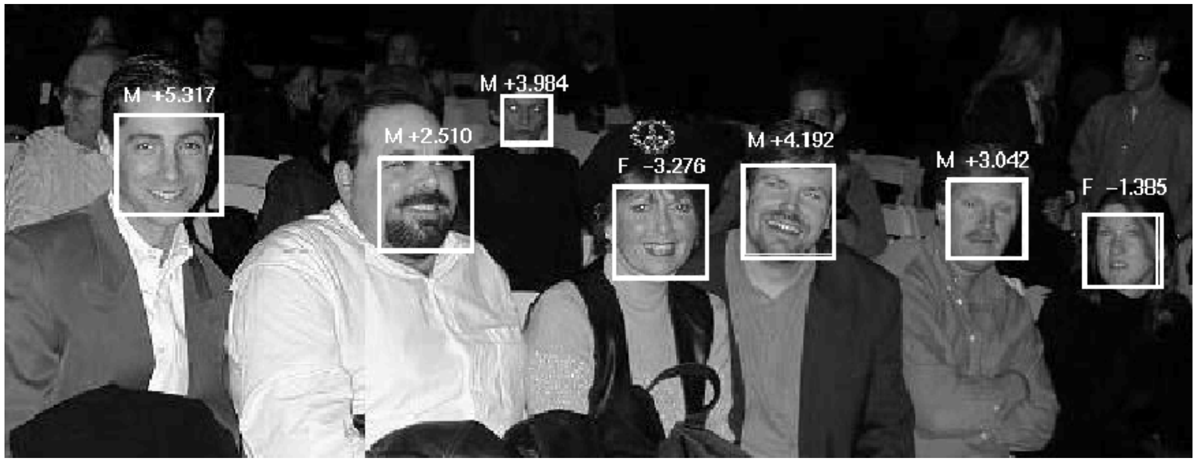
Source: S. Lazebnik

Related detection tasks

Facial Feature Localization

Profile Detection

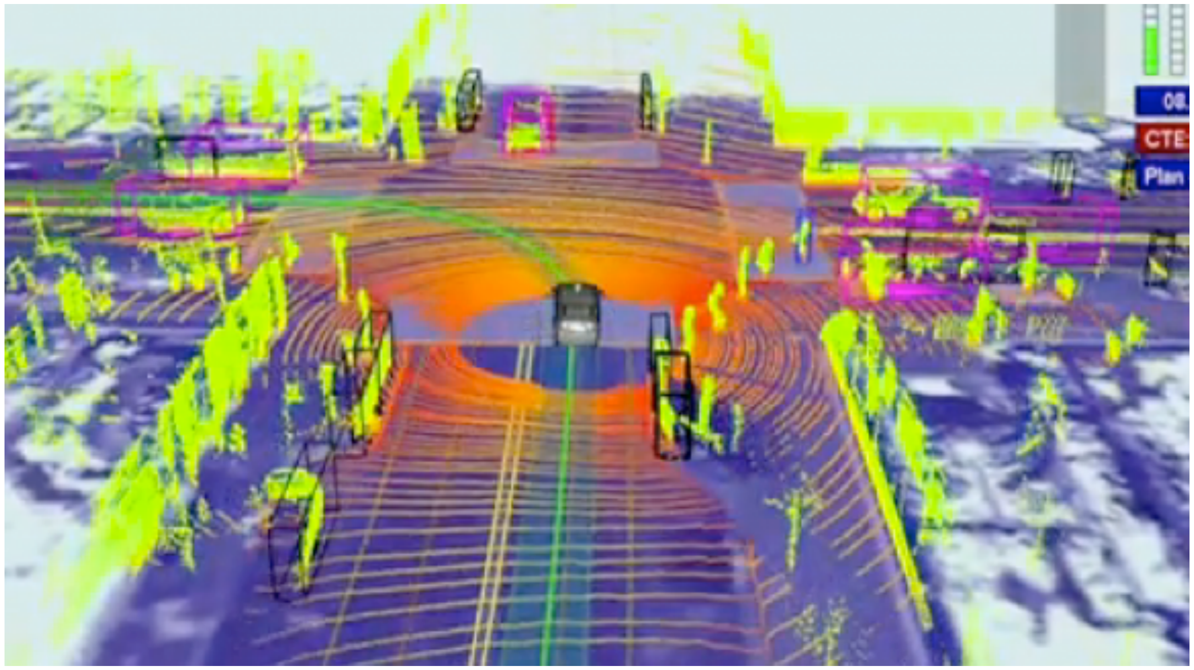
Male vs. female

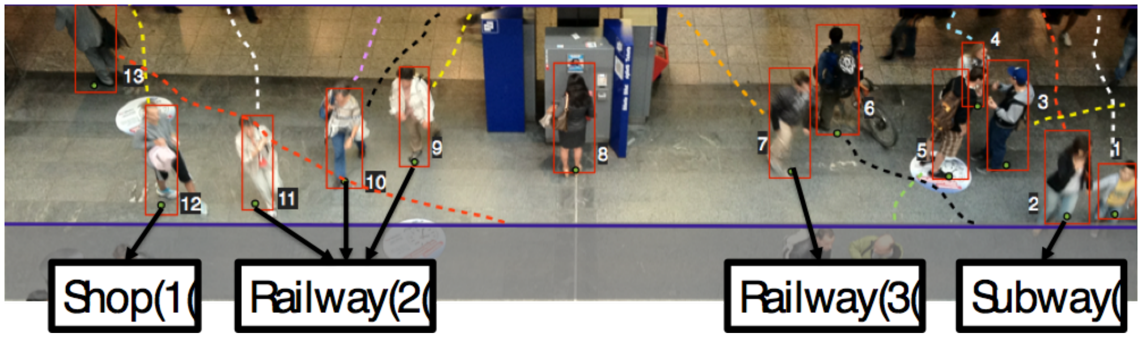


Source: S. Lazebnik

Pedestrian detection

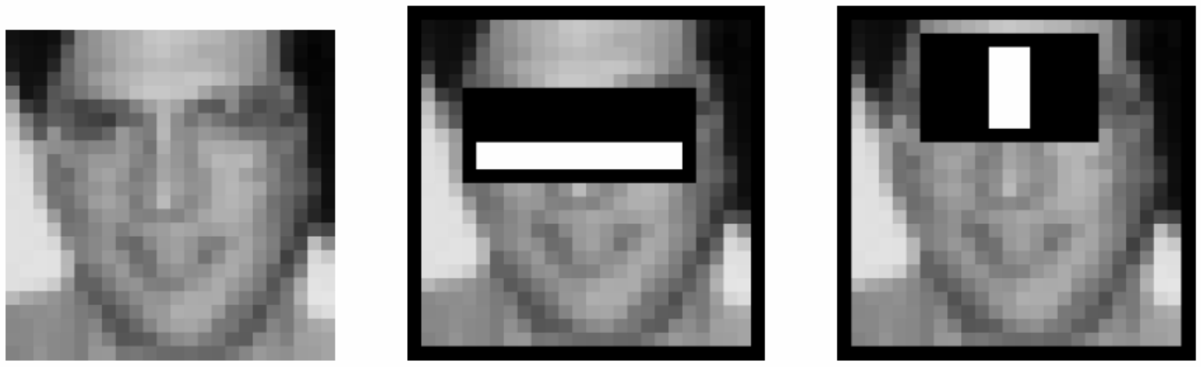
Pedestrian detection use cases





Alahi & Fei-Fei, 2014

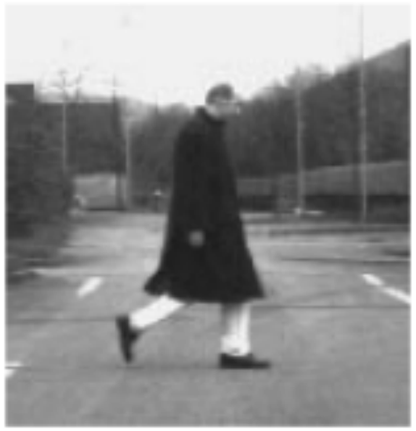
Faces vs pedestrians

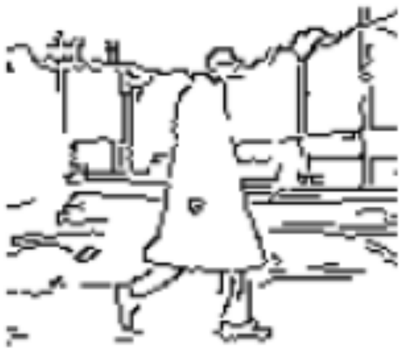


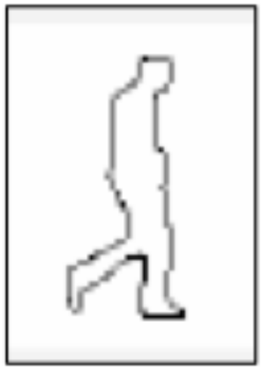
Faces: the identifying features are in the structure

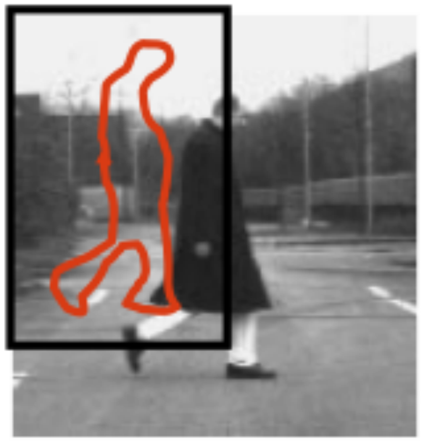
Pedestrian: much more about shape

Chamfer matching

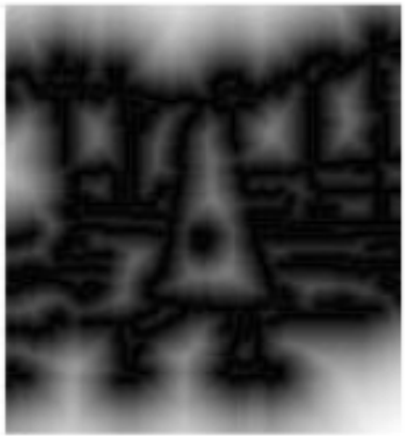

Input Image


Edge Detection


Template


Best Match

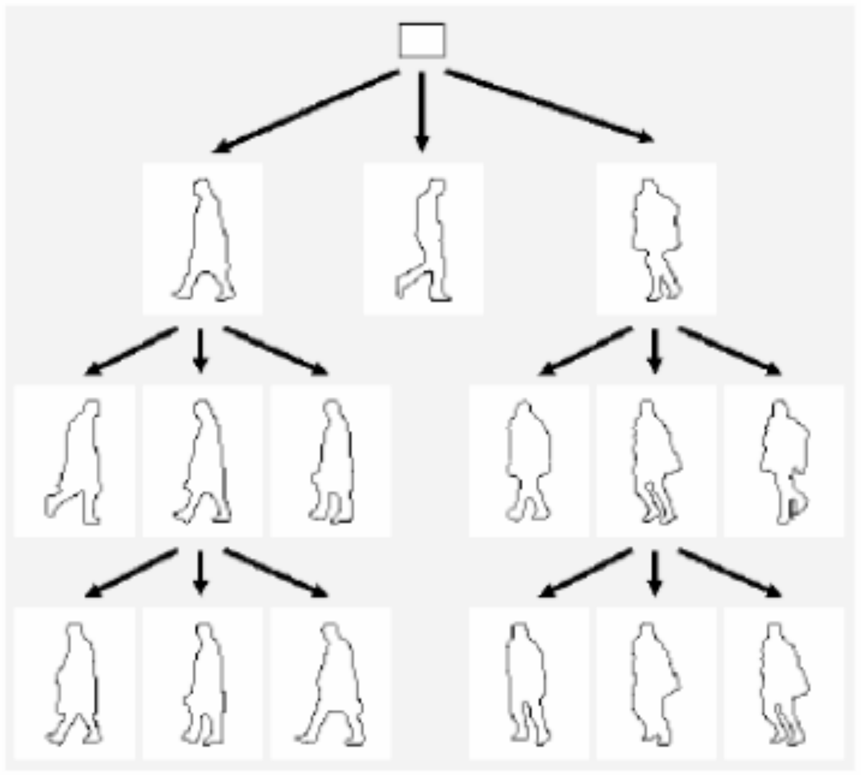
$$D_{chamfer}(T,I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t)$$


Distance Transform

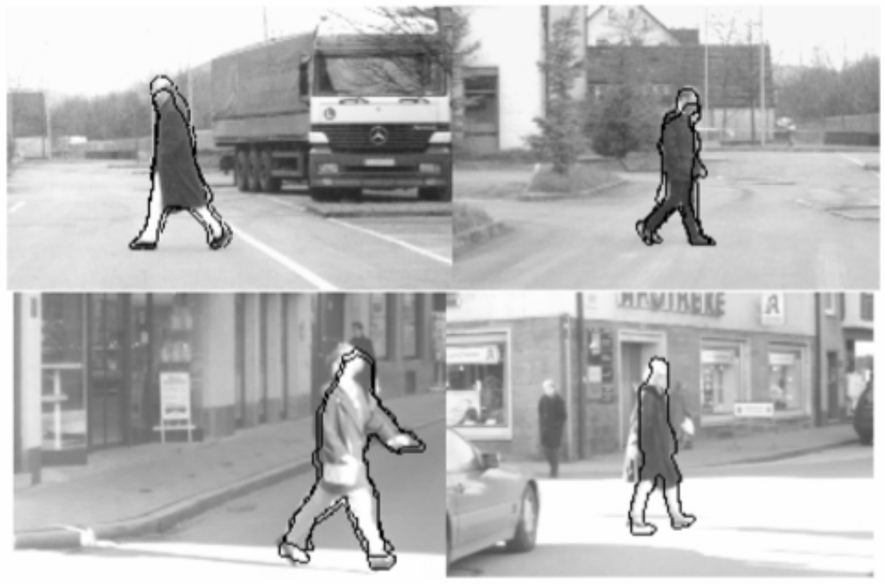
Gavrila & Philomin ICCV 1999

Slides from K. Grauman, B. Leibe, A. Ferencz

Chamfer matching

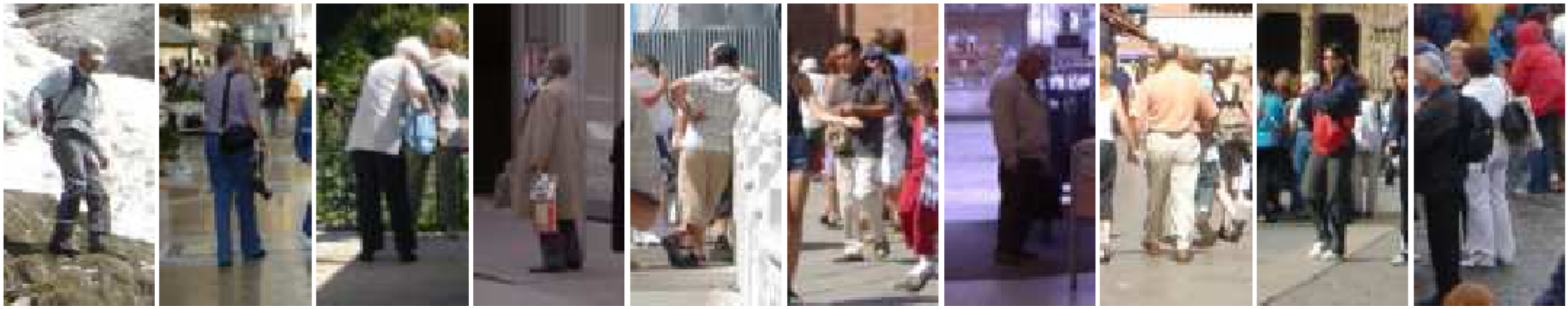


Hierarchy of templates

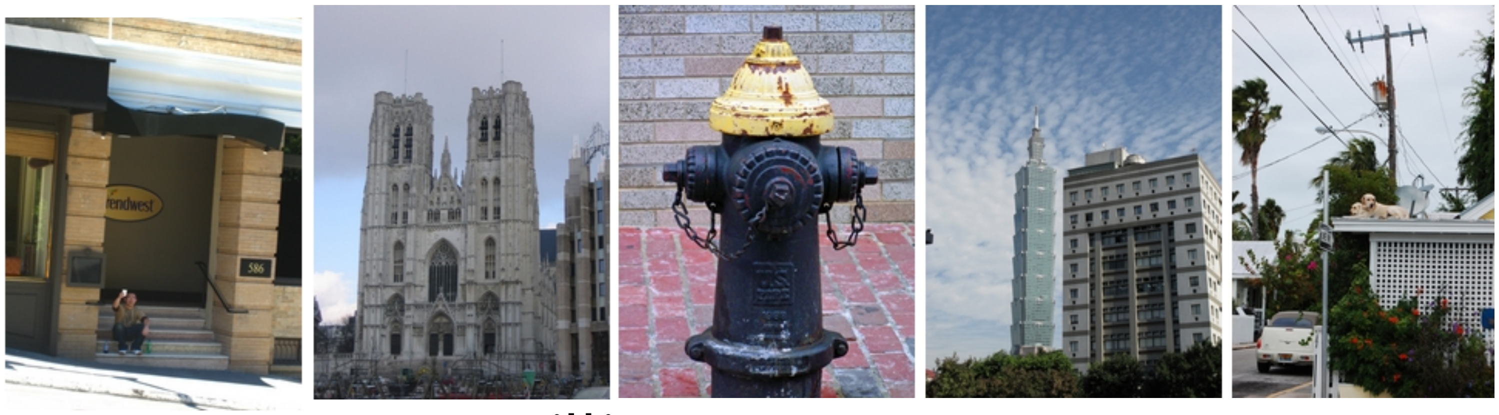


Gavrila & Philomin ICCV 1999

Positive and negative examples



+ thousands more...

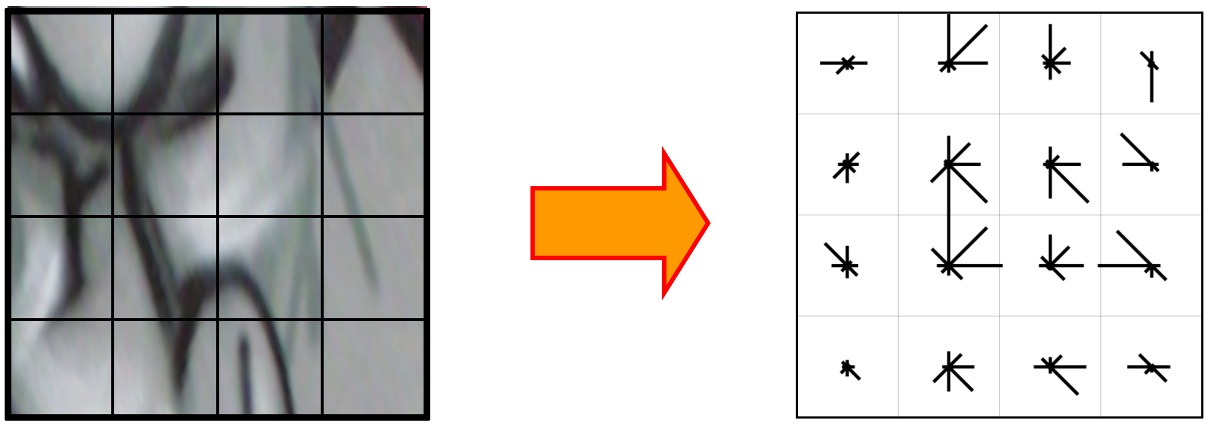


+ millions more...

N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

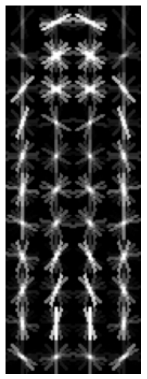
Credit: Noah Snavely

Window representation: edges/gradients



David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

- Used to describe small patches, but now consider describing a whole detection window



Histograms of oriented gradients (HOG)

Partition image into blocks and compute histogram of gradient orientations in each block

HxWx3 Image

H’xW’xC’ Image

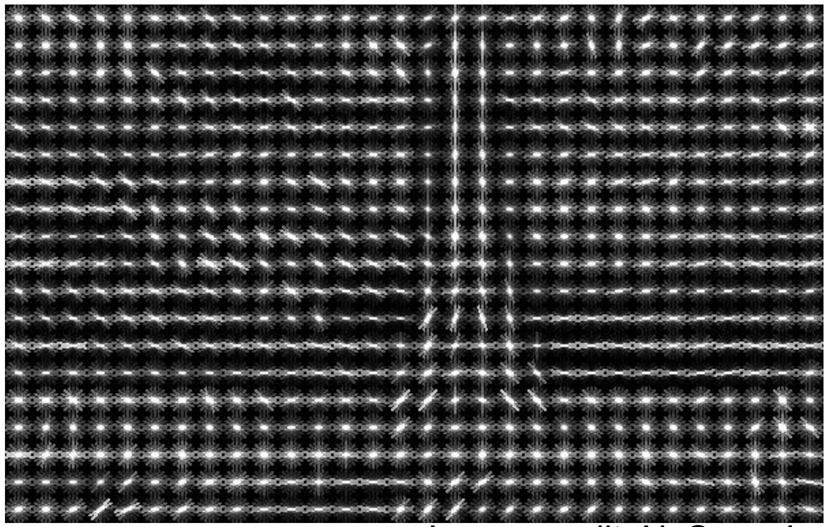


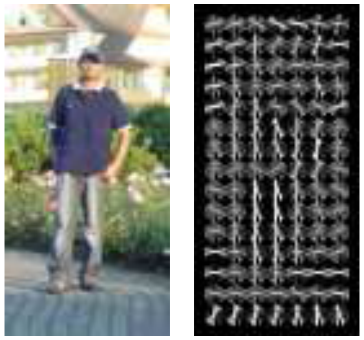
Image credit: N. Snavely

N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

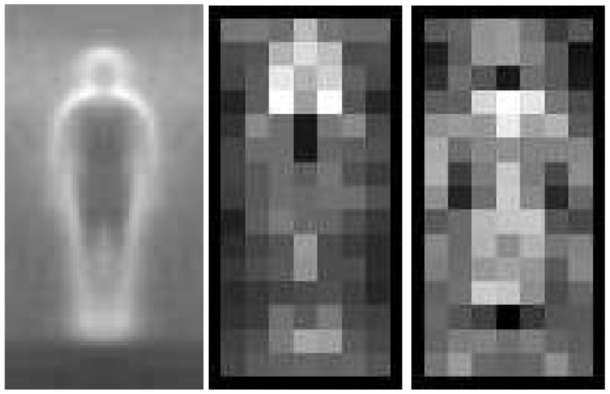
Slide Credit: S. Lazebnik

Person detection, ca. 2005 (Dalal Triggs)

1. Represent each example with a single, fixed HoG template



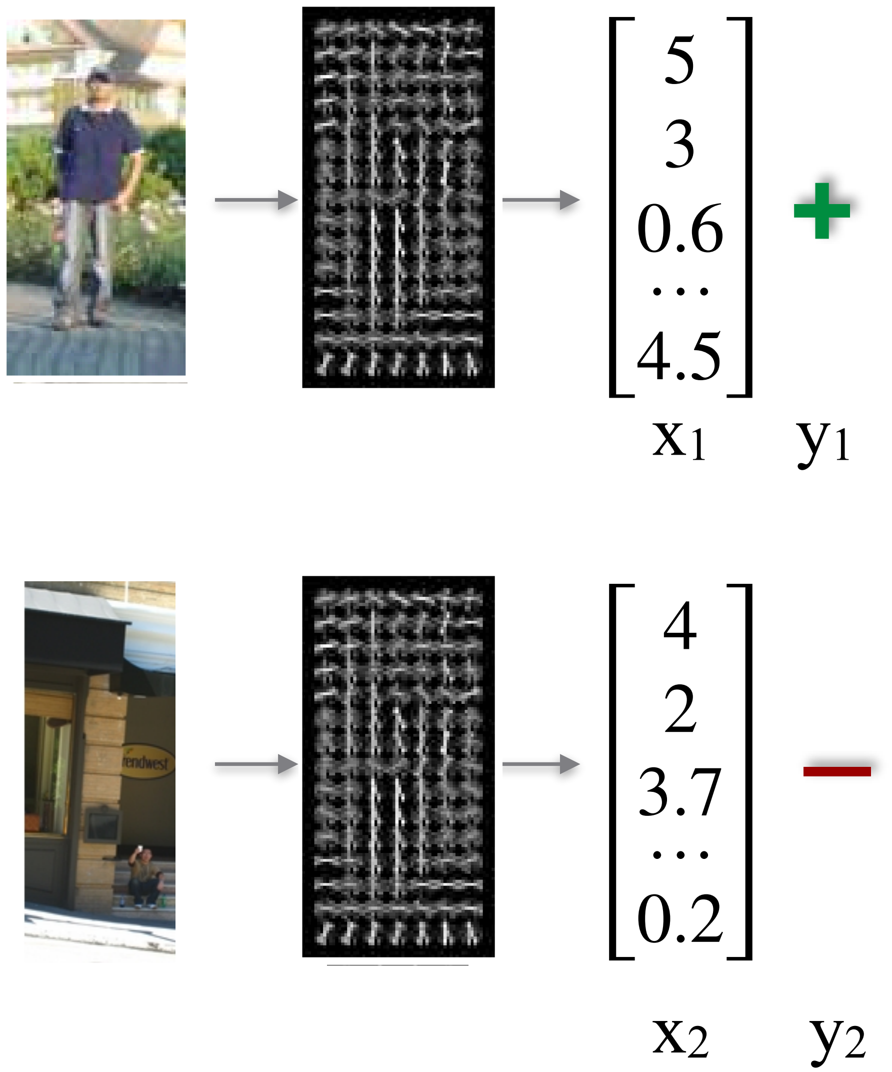
2. Learn a single linear detector



N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

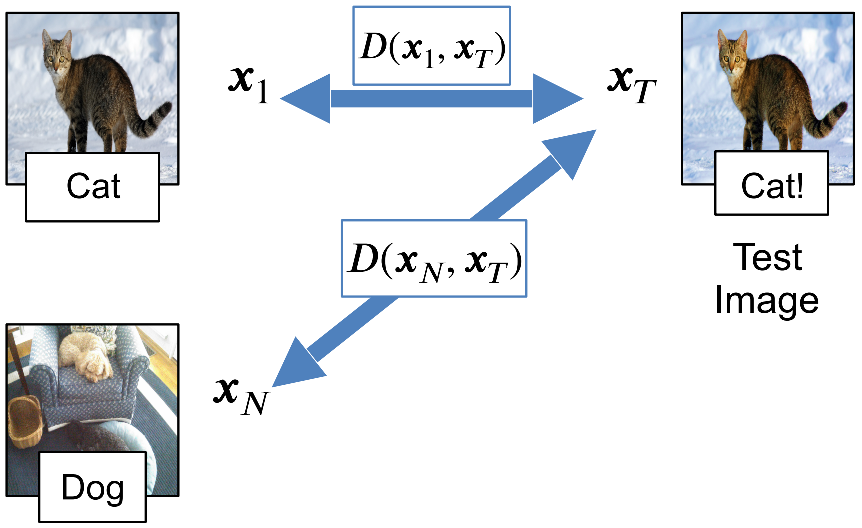
Credit: Noah Snavely

Training



$\begin{bmatrix} 5 \\ 3 \\ 0.6 \\ \cdots \\ 4.5 \end{bmatrix}$
+
x1
y1
$\begin{bmatrix} 4 \\ 2 \\ 3.7 \\ \cdots \\ 0.2 \end{bmatrix}$
−
x2
y2

Nearest neighbor classifier



Least squares classifier

Objective

$$\underset{w}{\operatorname{argmin}} \|y - Xw\|_2^2 + \lambda \|w\|_2^2$$

Inference

$$w^T x > t$$

Alternate objective: Multiclass SVM

Inference (x): $\underset{k}{\text{argmax}}\ (\boldsymbol{W}\boldsymbol{x})_k$

(Take the class whose weight vector gives the highest score)

Credit: D. Fouhey

Alternate objective: Multiclass SVM

Inference (x): $\underset{k}{\operatorname{argmax}}\ (\boldsymbol{Wx})_k$

(Take the class whose weight vector gives the highest score)

Training ($\mathbf{x}_i$,y_i):

$$\underset{W}{\arg\min}\ \lambda \|W\|_2^2 + \sum_{i}^{n} \sum_{j \neq y_i} \underbrace{\max(0, \mathbf{w_j}^T \mathbf{x_i} - \mathbf{w_{y_i}}^T \mathbf{x_i} + m)}$$

Regularization

Over all data points

For every class j that's NOT the correct one (y_i)

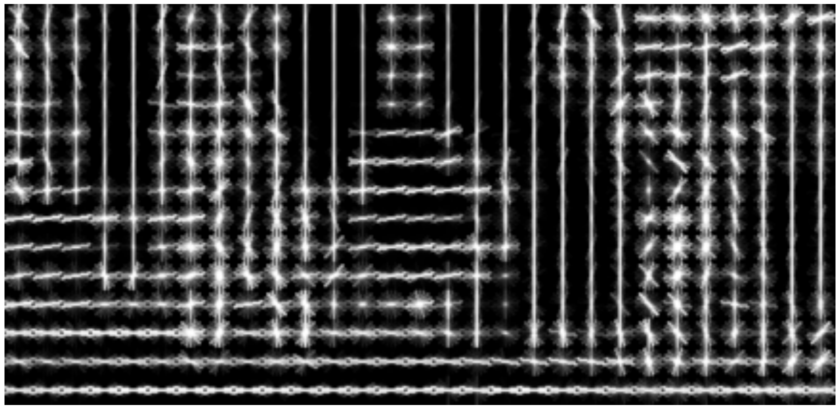
Pay no penalty if prediction for class y_i is bigger than j by m ("margin"). Otherwise, pay proportional to the score of the wrong class.

Credit: D. Fouhey

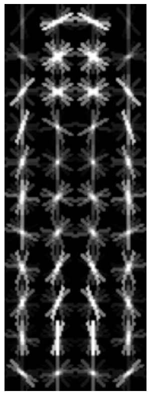
Pedestrian detection with HOG

- Train HOG-based pedestrian "template" with a linear model
- At test time, convolve feature map with template
- Find local maxima of response
- For multi-scale detection, repeat over multiple levels of a HOG *pyramid*

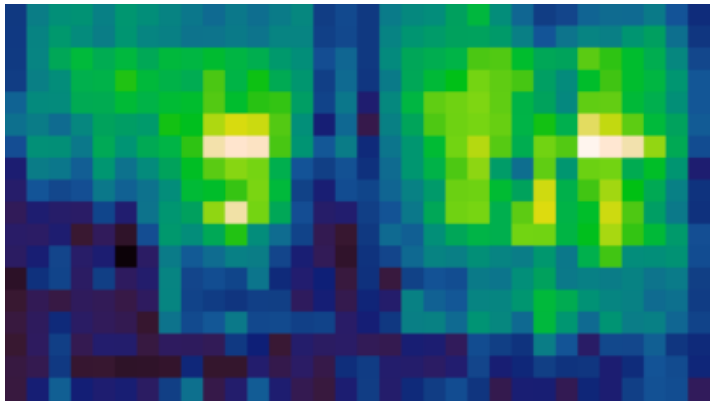
HOG feature map



Template



Detector response map



N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

Slide Credit: S. Lazebnik

What is learned

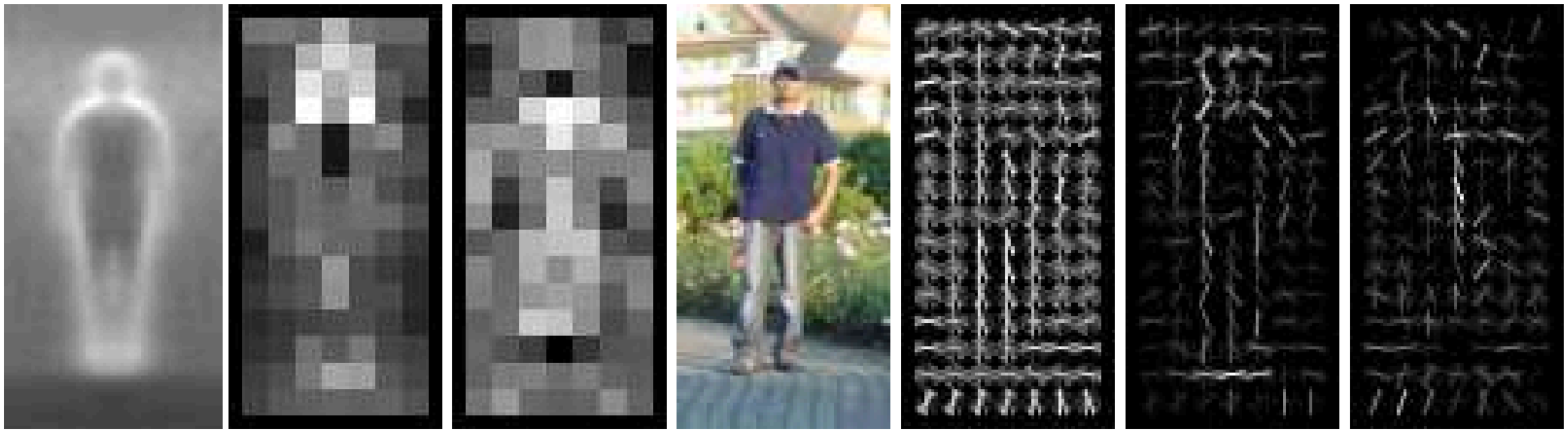
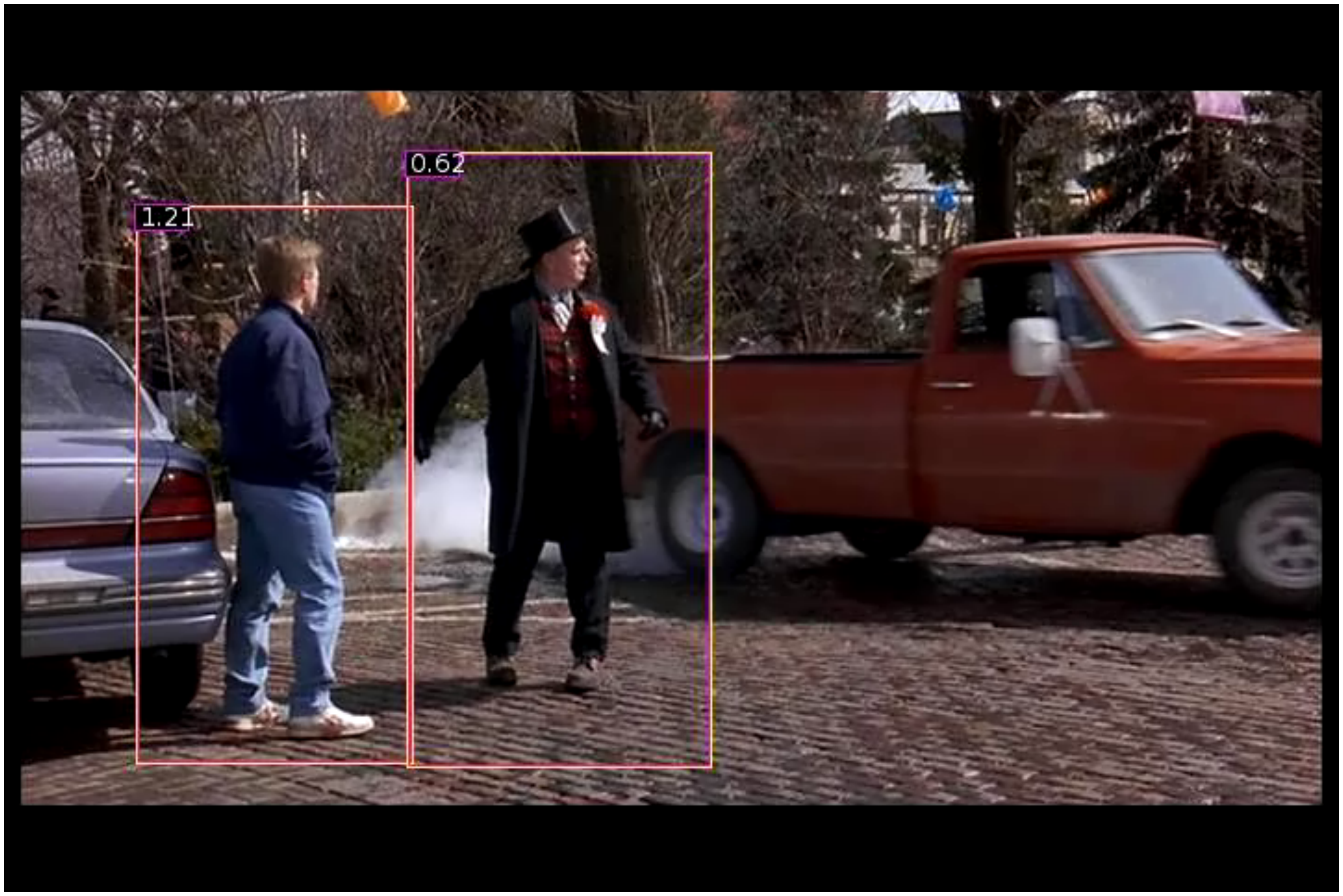


Figure 6. Our HOG detectors cue mainly on silhouette contours (especially the head, shoulders and feet). The most active blocks are centred on the image background just *outside* the contour. (a) The average gradient image over the training examples. (b) Each "pixel" shows the maximum positive SVM weight in the block centred on the pixel. (c) Likewise for the negative SVM weights. (d) A test image. (e) It's computed R-HOG descriptor. (f,g) The R-HOG descriptor weighted by respectively the positive and the negative SVM weights.

N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

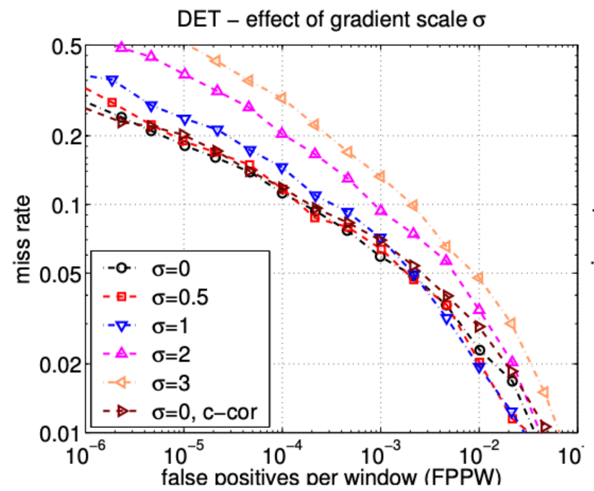
Example detections



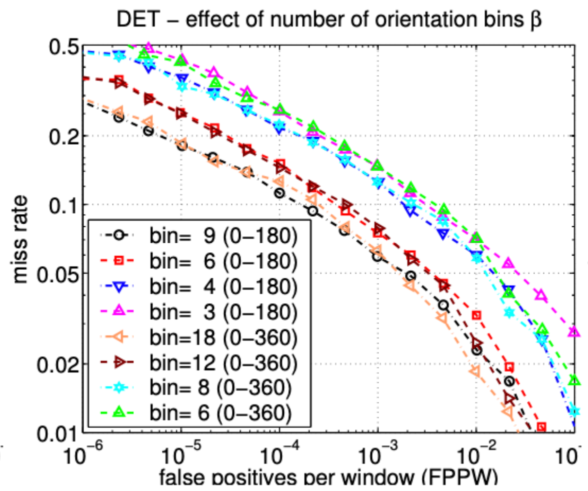
[Dalal and Triggs, CVPR 2005]

Slide Credit: S. Lazebnik

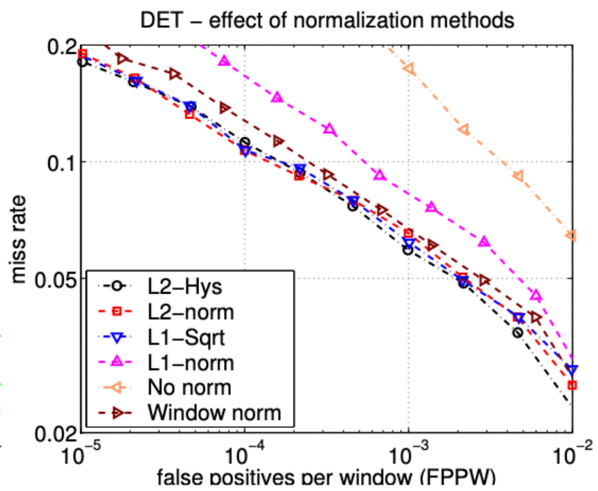
Evaluation: effect of hyperparameters



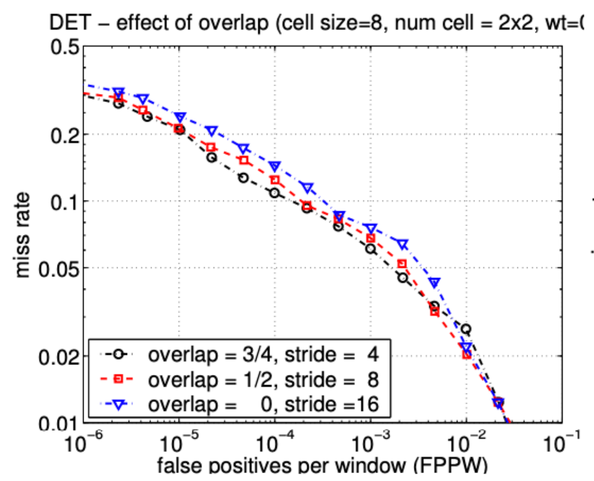
(a)

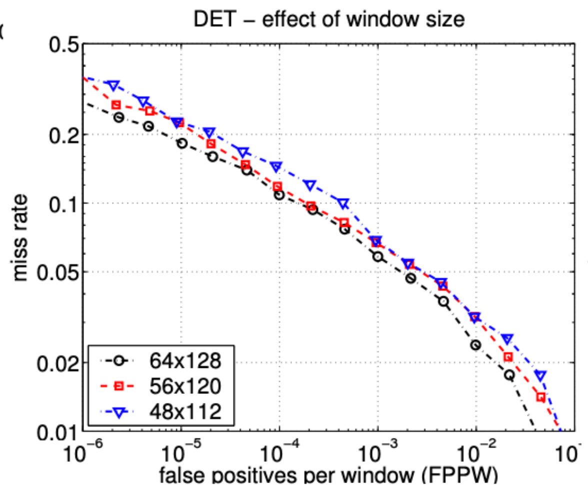


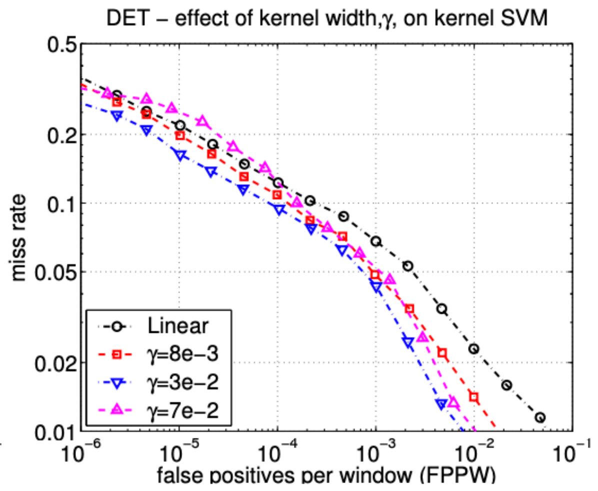
(b)



(c)







[Dalal and Triggs, CVPR 2005]

Slide Credit: S. Lazebnik

Summary: key concepts

1. ML for image classification (NN, kNN, linear models)
2. Viola Jones face detection
3. Dalal Triggs pedestrian detection

Next time: object classification