Visual Recognition

Image Classification & Object Detection

Κώστας Παπουτσάκης, CVRL, Ινστιτούτο Πληροφορικής, ΙΤΕ.

ΔΠΜΣ Προηγμένα Συστήματα Παραγωγής, Αυτοματισμού και Ρομποτικής, ΕΛΜΕΠΑ, 11-12-2020

Visual recognition

- Image classification
- Object detection
- Object recognition

Slides credits:
Juan Carlos Niebles and Ranjay Krishna, Stanford Vision and Learning Lab
Ali Farhadi, CSE 455, University of Washington

Visual Recognition

Visual Recognition

Visual Recognition

Visual Recognition

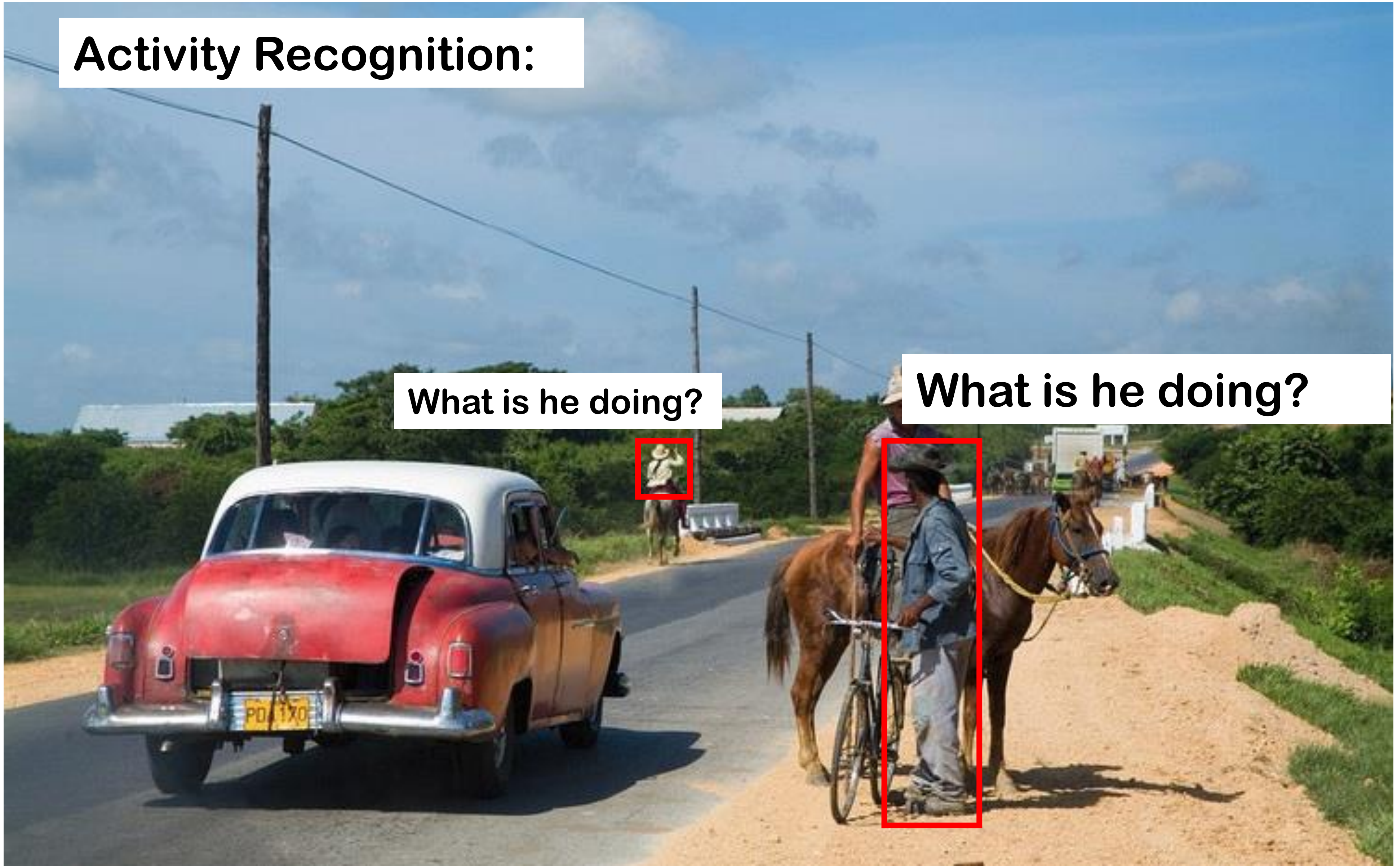
Object Detection + Classification

Is there the car in this picture and where is it located?

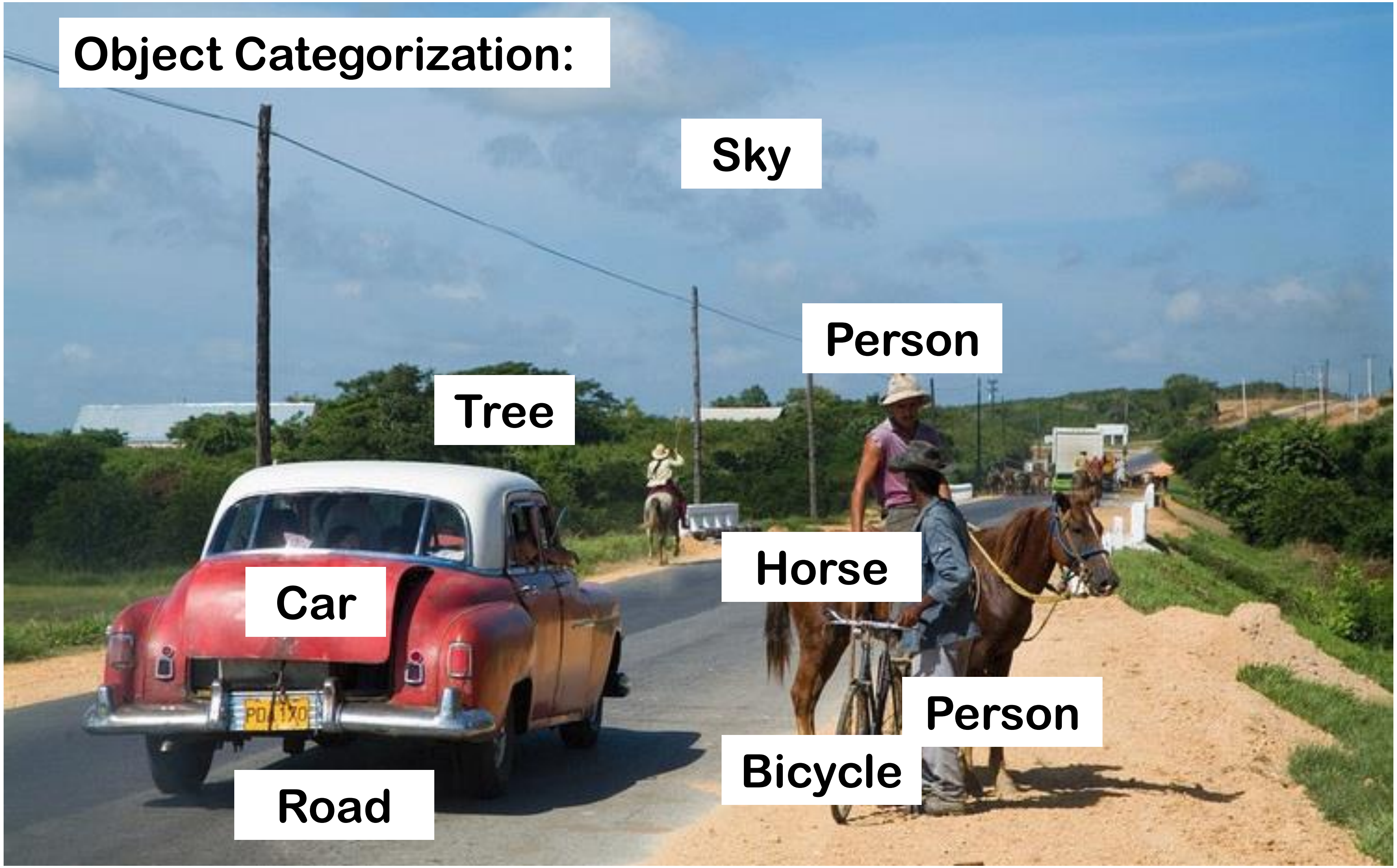
Visual Recognition

Pose Estimation:

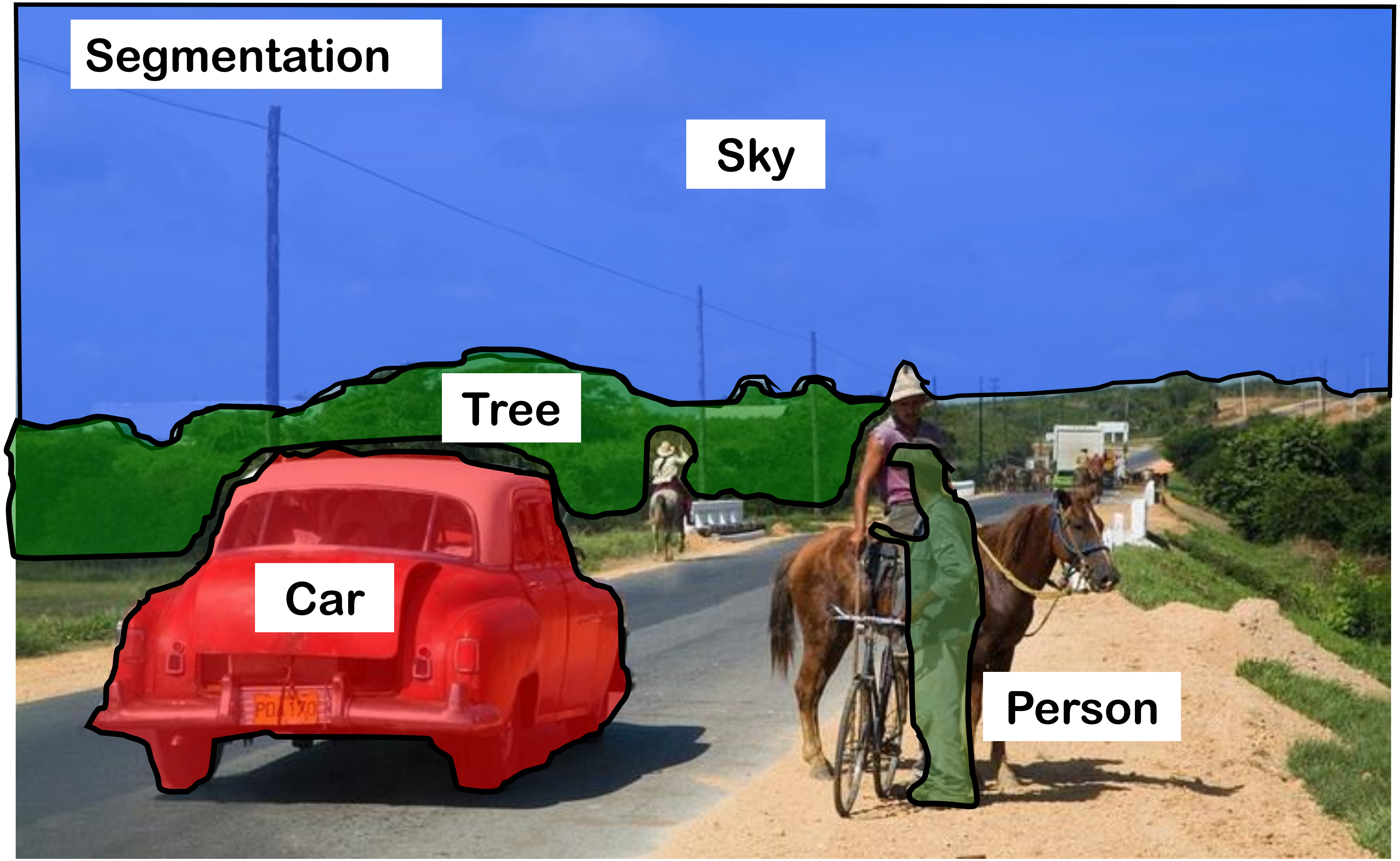
Visual Recognition



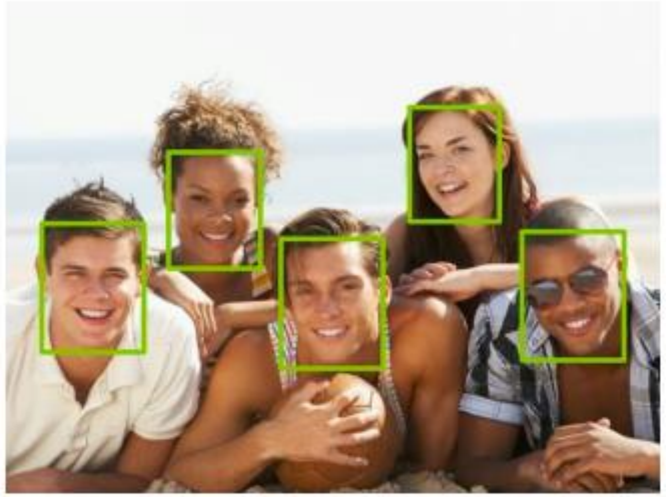
Visual Recognition



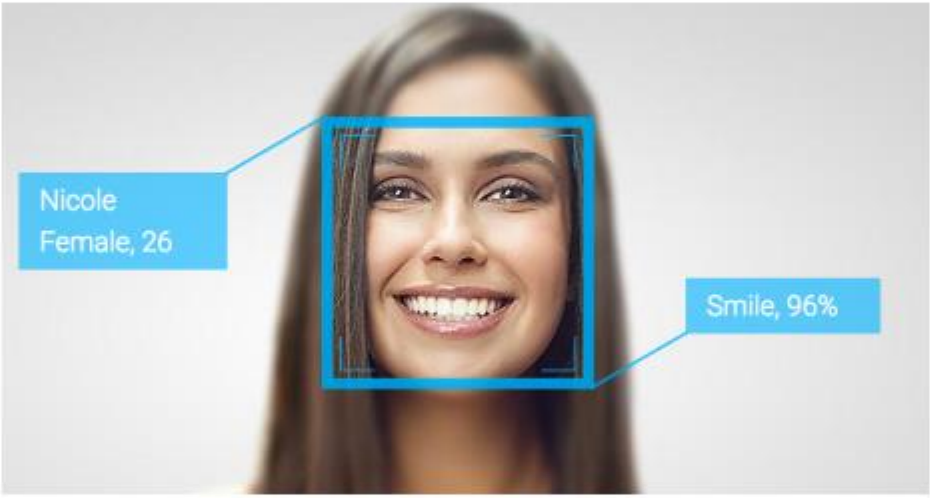
Visual Recognition



Detection versus Recognition



Detection finds the faces in images



Recognition recognizes WHO the person is

Recognition

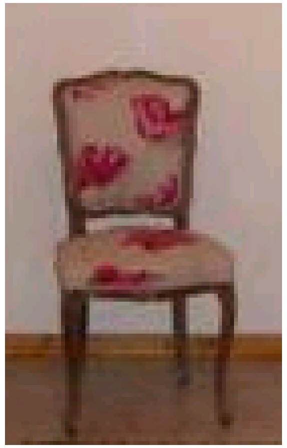
- Design algorithms that have the capability to:
- Object recognition
  - –Classify images or videos
  - –Detect and localize objects
  - –Estimate semantic and geometrical attributes
  - – Classify human activities and events
- Why is this challenging?

Face detection and recognition
Is it really so hard?

- changes in expression, lighting, age, occlusion, **viewpoint**

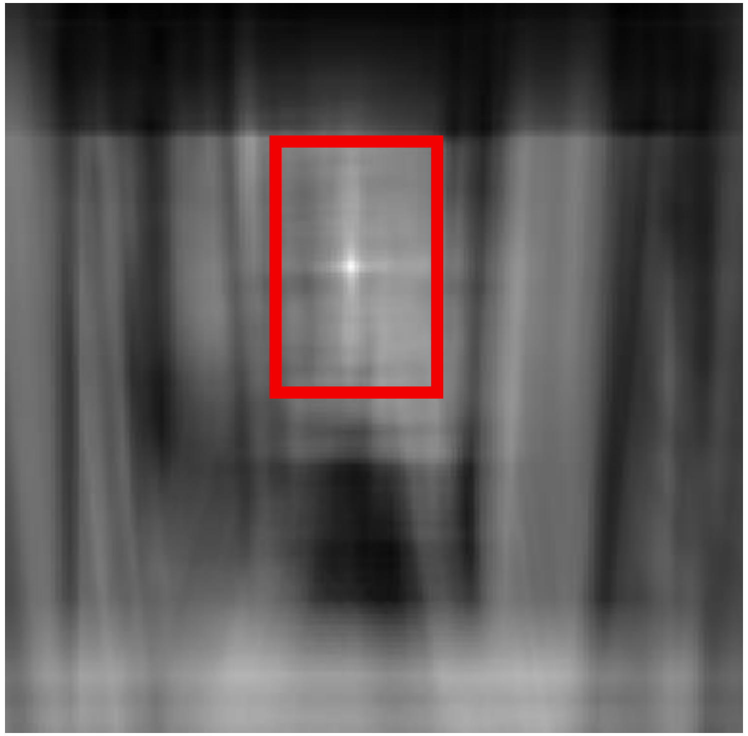
Object recognition
Is it really so hard?

This is a chair



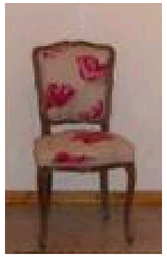
Find the chair in this image

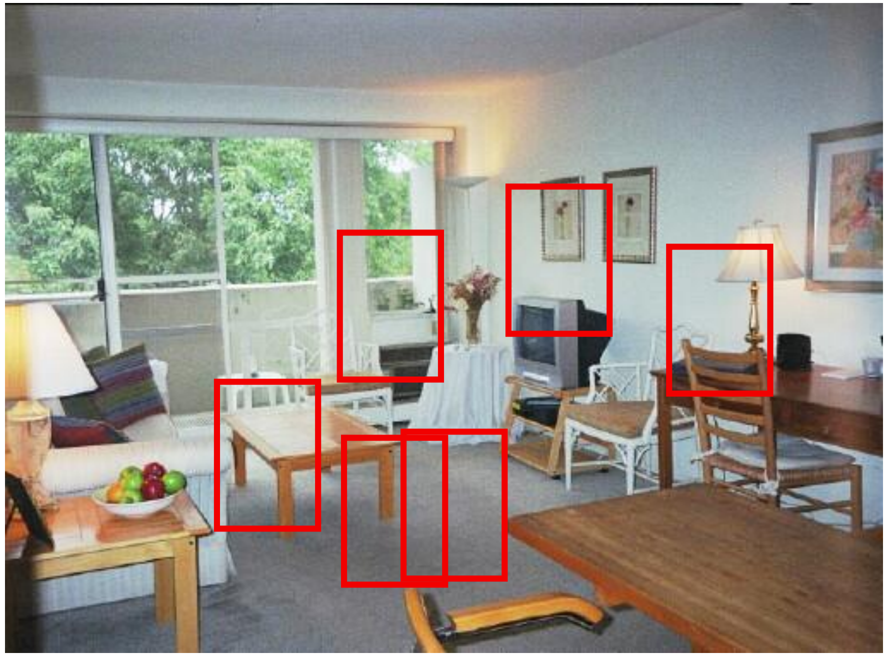
Output of normalized correlation

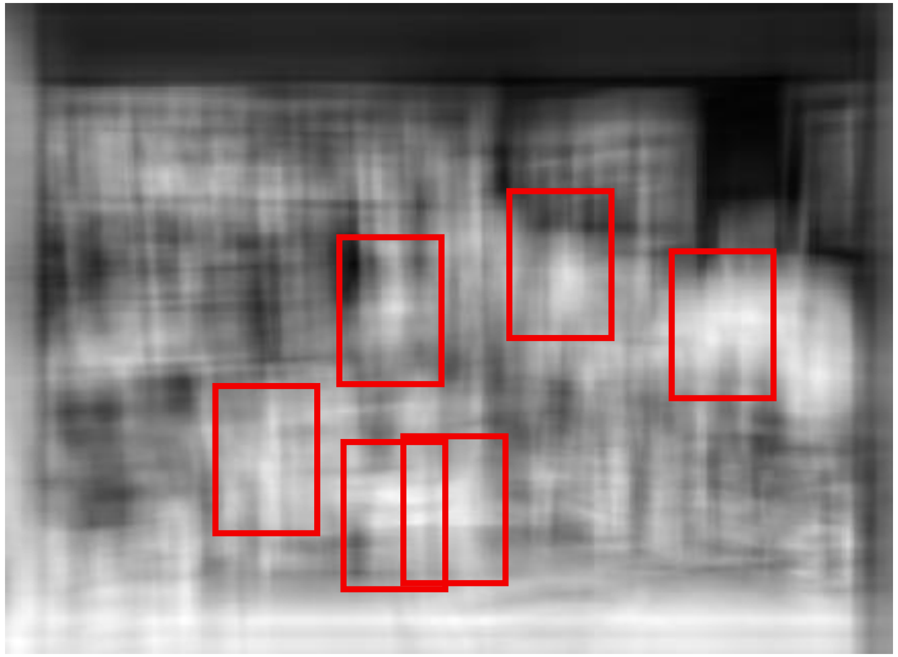


Object recognition
Is it really so hard?

Find the chair in this image







Pretty much garbage

Simple template matching is not going to make it

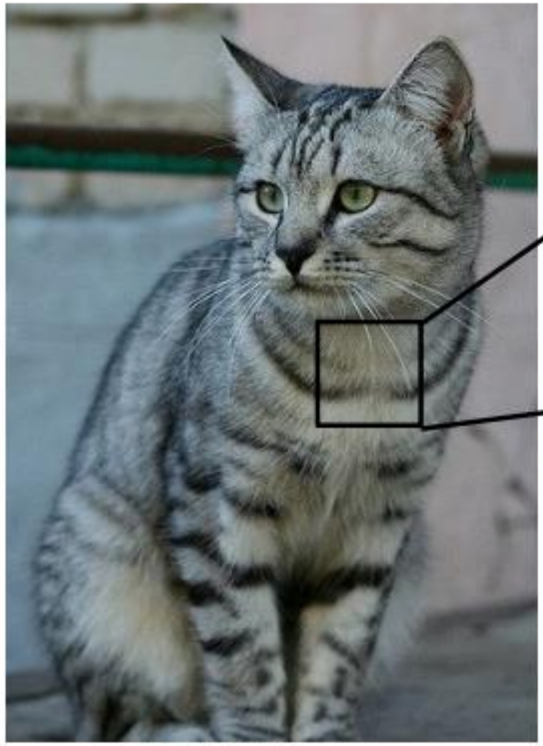
Image Classification: A core task in Computer Vision

(assume given set of discrete labels)
{dog, cat, truck, plane, ...}

$\longrightarrow$ cat

Li, Johnson, Yeung, cs231n, Stanford 2019

The Problem: Semantic Gap



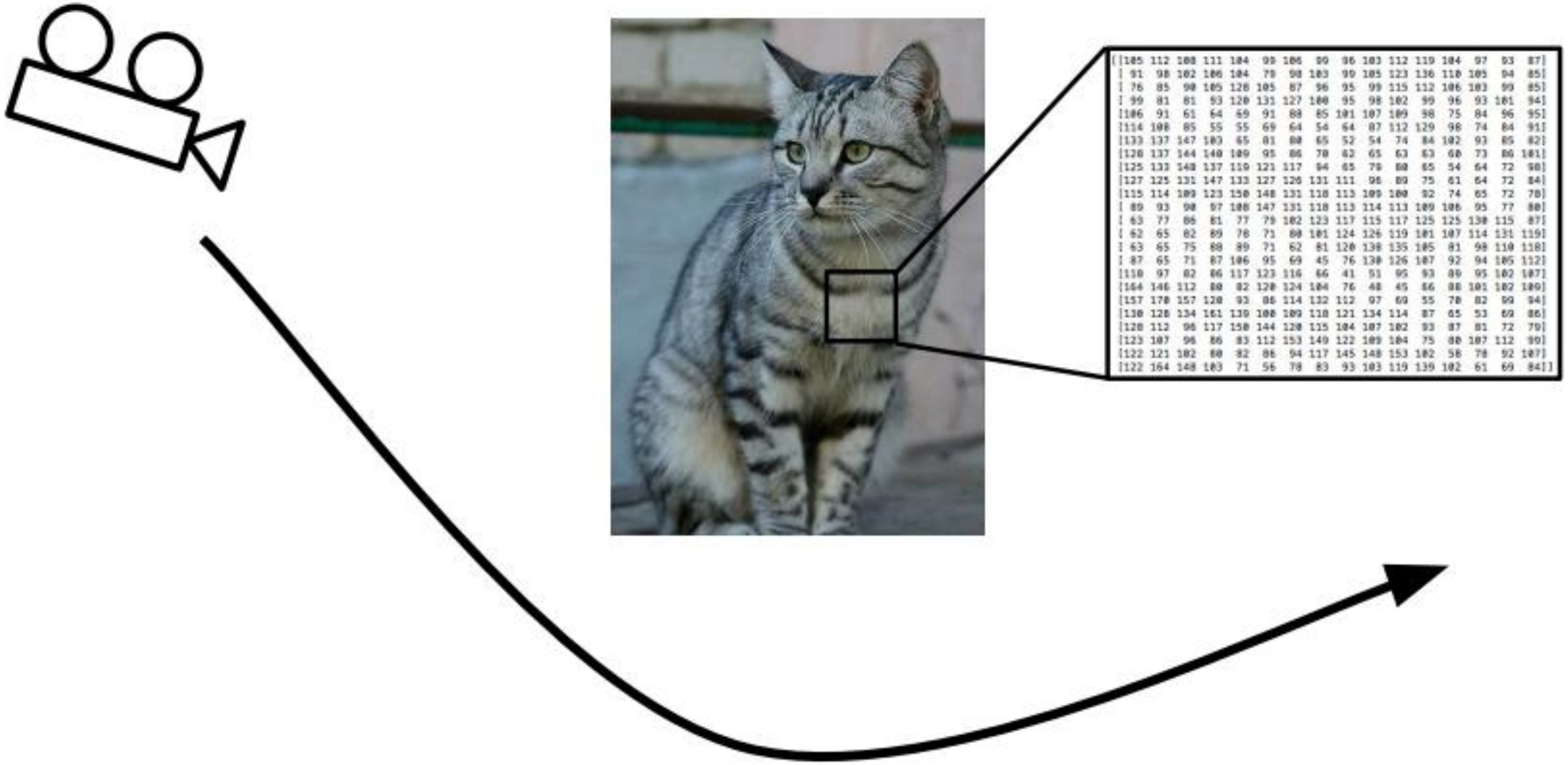

```
[[105 112 108 111 104  99 106  99  96 103 112 119 104  97  93  87]
 [ 91  98 102 106 104  79  98 103  99 105 123 136 110 105  94  85]
 [ 76  85  90 105 128 105  87  96  95  99 115 112 106 103  99  85]
 [ 99  81  81  93 120 131 127 100  95  98 102  99  96  93 101  94]
 [106  91  61  64  69  91  88  85 101 107 109  98  75  84  96  95]
 [114 108  85  55  55  69  64  54  64  87 112 129  98  74  84  91]
 [133 137 147 103  65  81  80  65  52  54  74  84 102  93  85  82]
 [128 137 144 140 109  95  86  70  62  65  63  63  60  73  86 101]
 [125 133 148 137 119 121 117  94  65  79  80  65  54  64  72  98]
 [127 125 131 147 133 127 126 131 111  96  89  75  61  64  72  84]
 [115 114 109 123 150 148 131 118 113 109 100  92  74  65  72  78]
 [ 89  93  90  97 108 147 131 118 113 114 113 109 106  95  77  80]
 [ 63  77  86  81  77  79 102 123 117 115 117 125 125 130 115  87]
 [ 62  65  82  89  78  71  80 101 124 126 119 101 107 114 131 119]
 [ 63  65  75  88  89  71  62  81 120 138 135 105  81  98 110 118]
 [ 87  65  71  87 106  95  69  45  76 130 126 107  92  94 105 112]
 [118  97  82  86 117 123 116  66  41  51  95  93  89  95 102 107]
 [164 146 112  80  82 120 124 104  76  48  45  66  88 101 102 109]
 [157 170 157 120  93  86 114 132 112  97  69  55  70  82  99  94]
 [130 128 134 161 139 100 109 118 121 134 114  87  65  53  69  86]
 [128 112  96 117 150 144 120 115 104 107 102  93  87  81  72  79]
 [123 107  96  86  83 112 153 149 122 109 104  75  80 107 112  99]
 [122 121 102  80  82  86  94 117 145 148 153 102  58  78  92 107]
 [122 164 148 103  71  56  78  83  93 103 119 139 102  61  69  84]]
```

What the computer sees

An image is just a big grid of
numbers between [0, 255]:

e.g. 800 x 600 x 3
(3 channels RGB)

Challenges: Viewpoint variation



All pixels change when the camera moves!

Challenges 0: Too many categories

Challenges 1: view point variation



Michelangelo 1475-1564

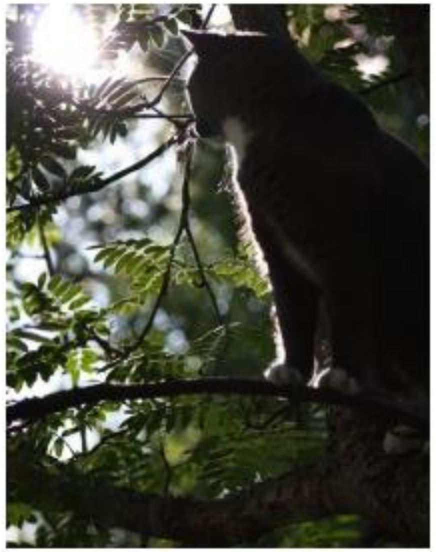
slide by Fei Fei, Fergus & Torralba

Challenges: Background Clutter

This image is CC0 1.0 public domain

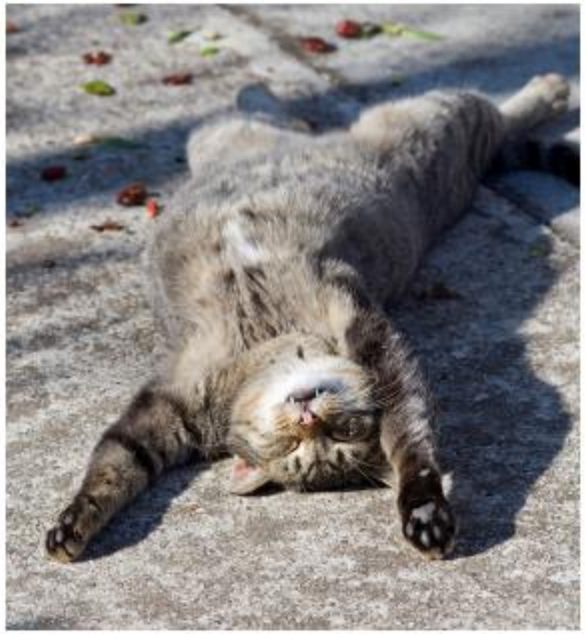
This image is CC0 1.0 public domain

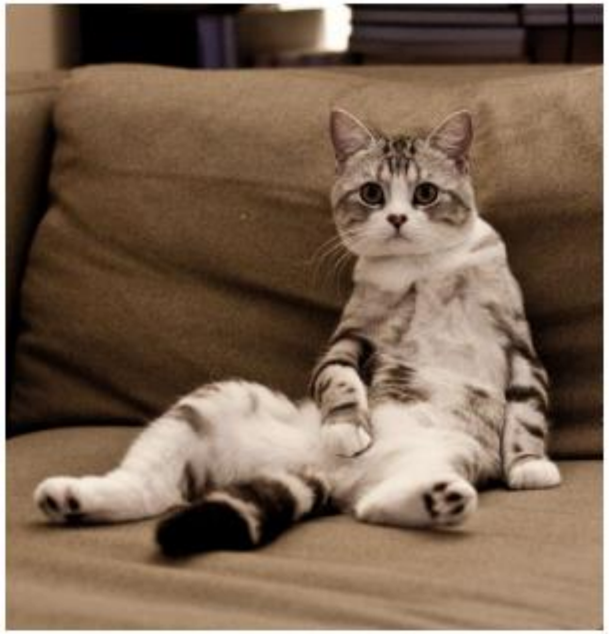
Challenges 2: illumination



Challenges 3: occlusion

Challenges 5: deformation





Challenges 4: scale

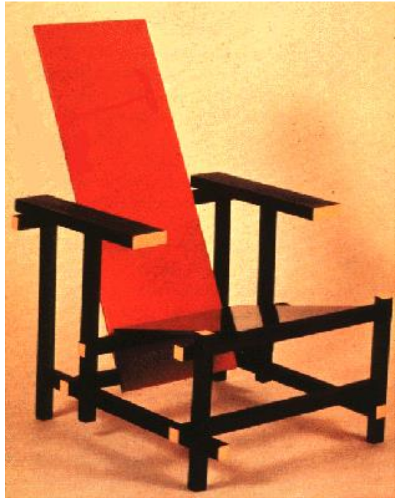
slide by Fei Fei, Fergus & Torralba

Challenges 6: background clutter

Challenges 7: object intra-class variation

Challenges 7: object intra-class variation



slide by Fei-Fei, Fergus & Torralba

Object classification/recognition

The machine learning framework

$$\underbrace{y}_{\text{output}} = \underbrace{f}_{\text{prediction function}}(\underbrace{\mathbf{x}}_{\text{Image feature}})$$

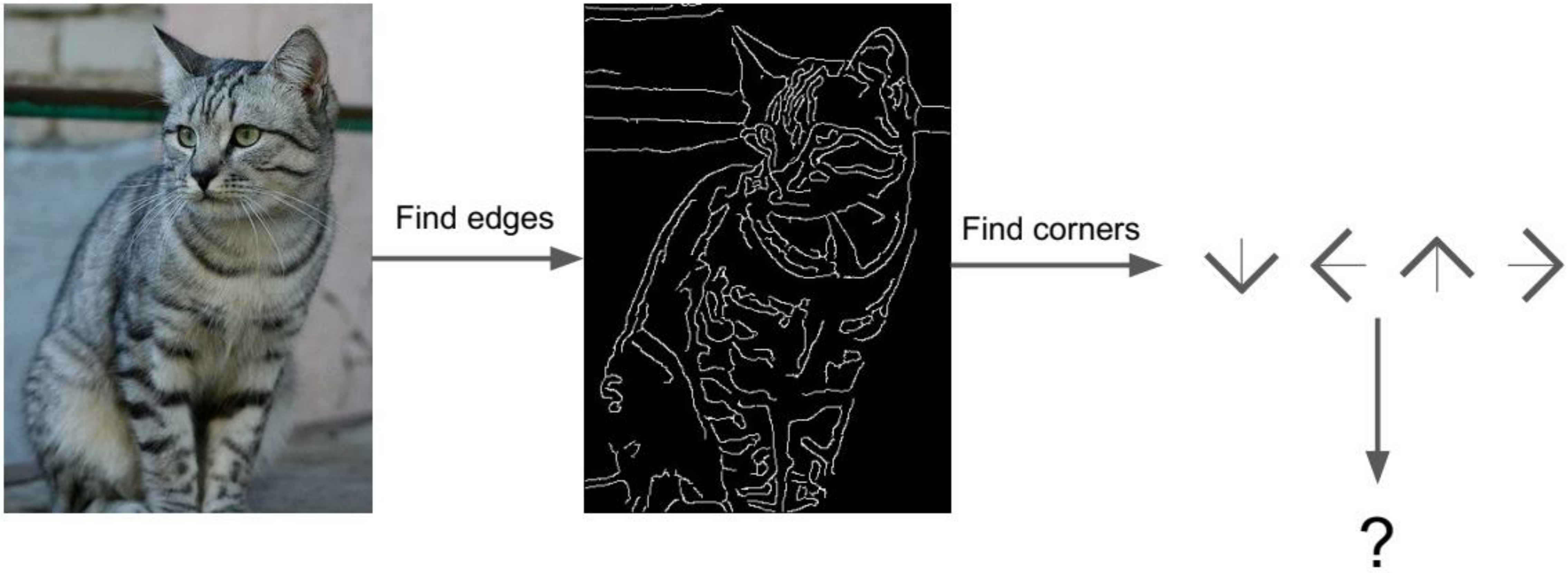
- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \dots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

Slide credit: L. Lazebnik

```
def classify_image(image):
  # Some magic here?
  return class_label
```

Object classification/recognition

Attempts have been made



Object classification/recognition

Machine Learning: Data-Driven Approach

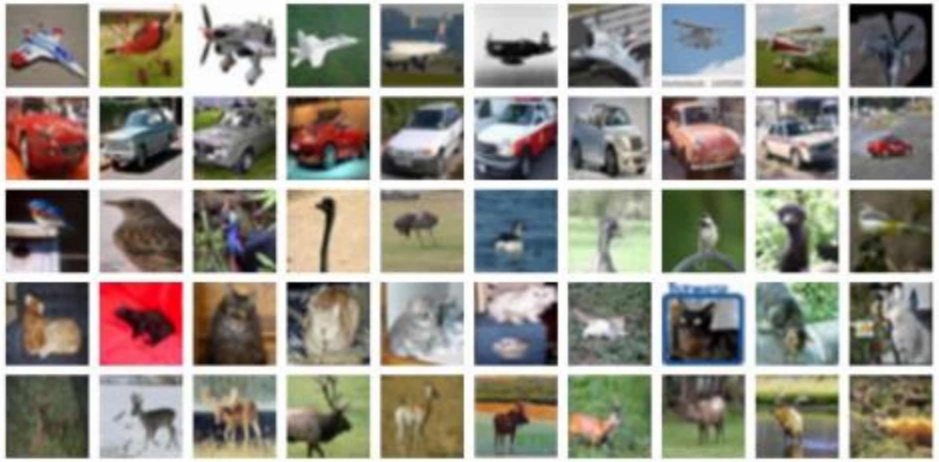
1. Collect a dataset of images and labels
2. Use Machine Learning to train a classifier
3. Evaluate the classifier on new images

```
def train(images, labels):
  # Machine learning!
  return model
```

```
def predict(model, test_images):
  # Use model to predict labels
  return test_labels
```

Example training set

airplane
automobile
bird
cat
deer

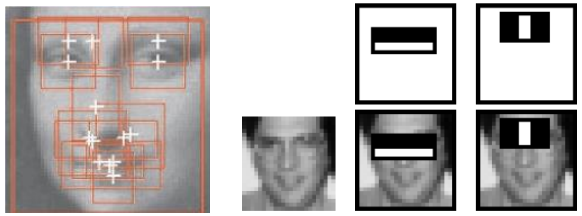


Families of recognition algorithms

Bag of words models

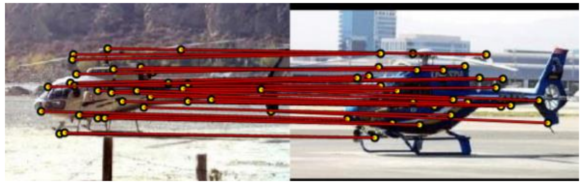
Csurka, Dance, Fan, Willamowski, and Bray 2004
Sivic, Russell, Freeman, Zisserman, ICCV 2005

Voting models



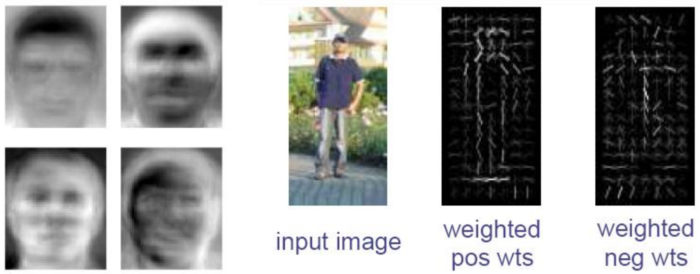
Viola and Jones, ICCV 2001
Heisele, Poggio, et. al., NIPS 01
Schneiderman, Kanade 2004
Vidal-Naquet, Ullman 2003

Shape matching
Deformable models



Berg, Berg, Malik, 2005
Cootes, Edwards, Taylor, 2001

Rigid template models

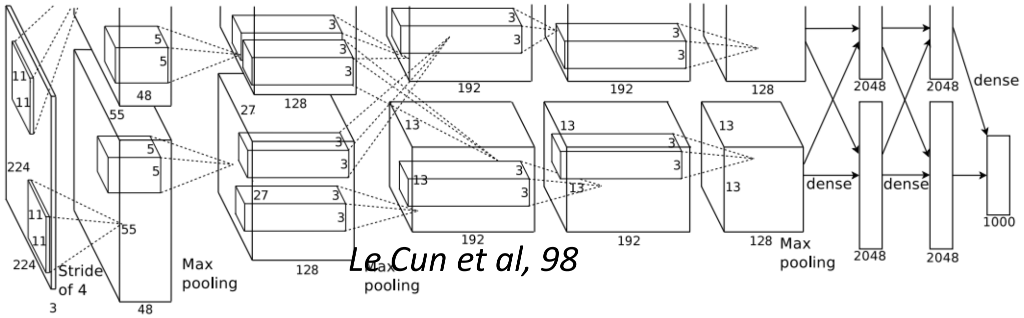


Sirovich and Kirby 1987
Turk, Pentland, 1991
Dalal & Triggs, 2006

Constellation models

Fischler and Elschlager, 1973
Burl, Leung, and Perona, 1995
Weber, Welling, and Perona, 2000
Fergus, Perona, & Zisserman, CVPR 2003

Neural networks



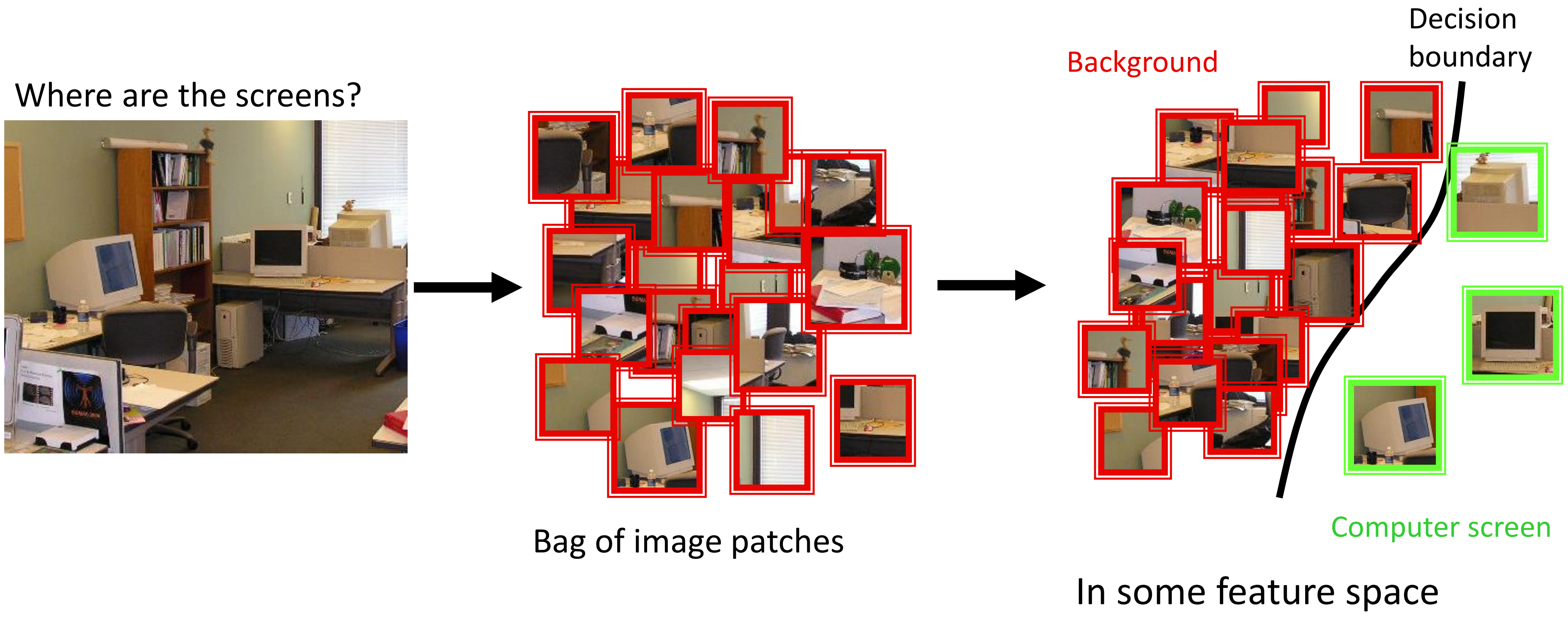
Le Cun et al, 98

Discriminative methods

Object detection and recognition is formulated as a classification problem.

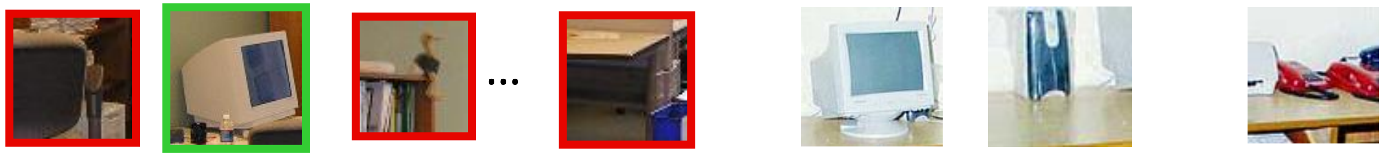
The image is partitioned into a set of overlapping windows

… and a decision is taken at each window about if it contains a target object or not.



Formulation

- Formulation: binary classification



Features $x =$ x_1 x_2 x_3 … x_N x_{N+1} x_{N+2} … x_{N+M}

Labels $y =$ -1 +1 -1 -1 ? ? ?

Training data: each image patch is labeled as containing the object or background

Test data

- Classification function

$$\hat{y} = F(x)$$ Where $F(x)$ belongs to some family of functions

- Minimize misclassification error

(Not that simple: we need some guarantees that there will be generalization)

Object classification/recognition
Parametric Approach: Linear Classifier

Image

Array of **32x32x3** numbers
(3072 numbers total)

$\longrightarrow f(\mathbf{x},\mathbf{W}) \longrightarrow$ **10** numbers giving class scores

$\uparrow$

W

parameters
or weights

Object classification/recognition

Parametric Approach: Linear Classifier

3072x1

$$\boxed{f(x,W)} = \boxed{W}\boxed{x} + \boxed{b}$$

10x1

10x1 10x3072

Image

Array of **32x32x3** numbers
(3072 numbers total)

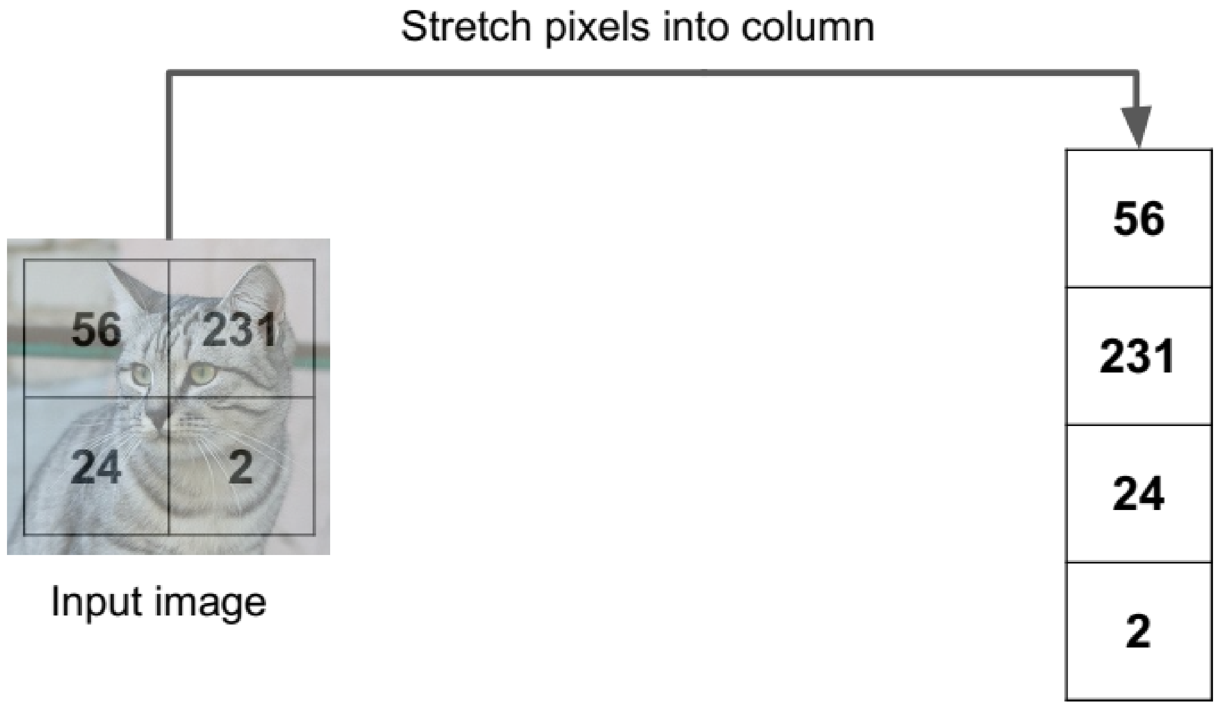
$\rightarrow f(\mathbf{x}, \mathbf{W}) \rightarrow$ **10** numbers giving class scores

W

parameters or weights

Object classification/recognition
Parametric Approach: Linear Classifier

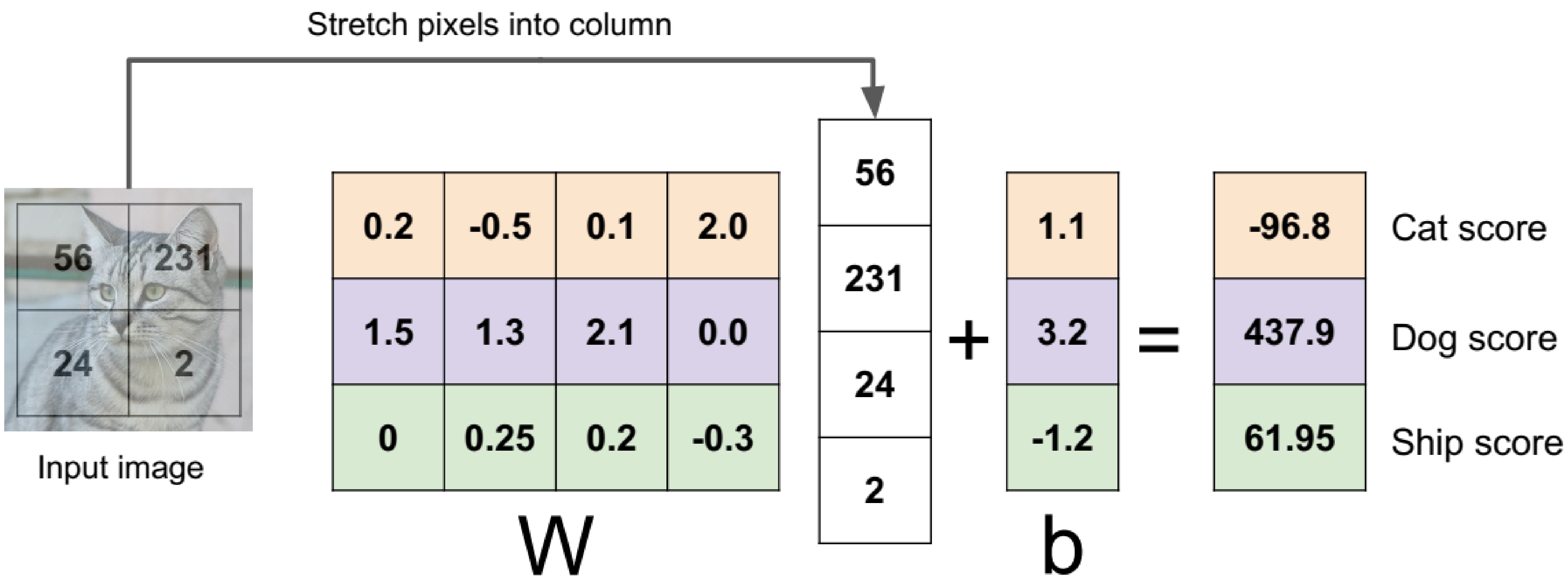
Example with an image with 4 pixels, and 3 classes (cat/dog/ship)



Object classification/recognition

Parametric Approach: Linear Classifier

Example with an image with 4 pixels, and 3 classes (cat/dog/ship)

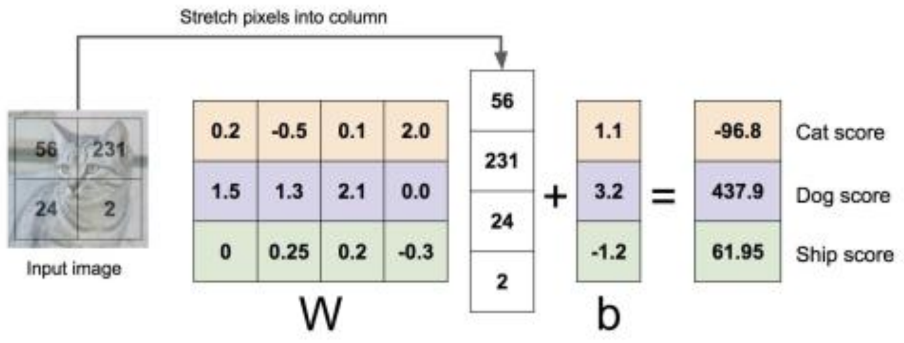


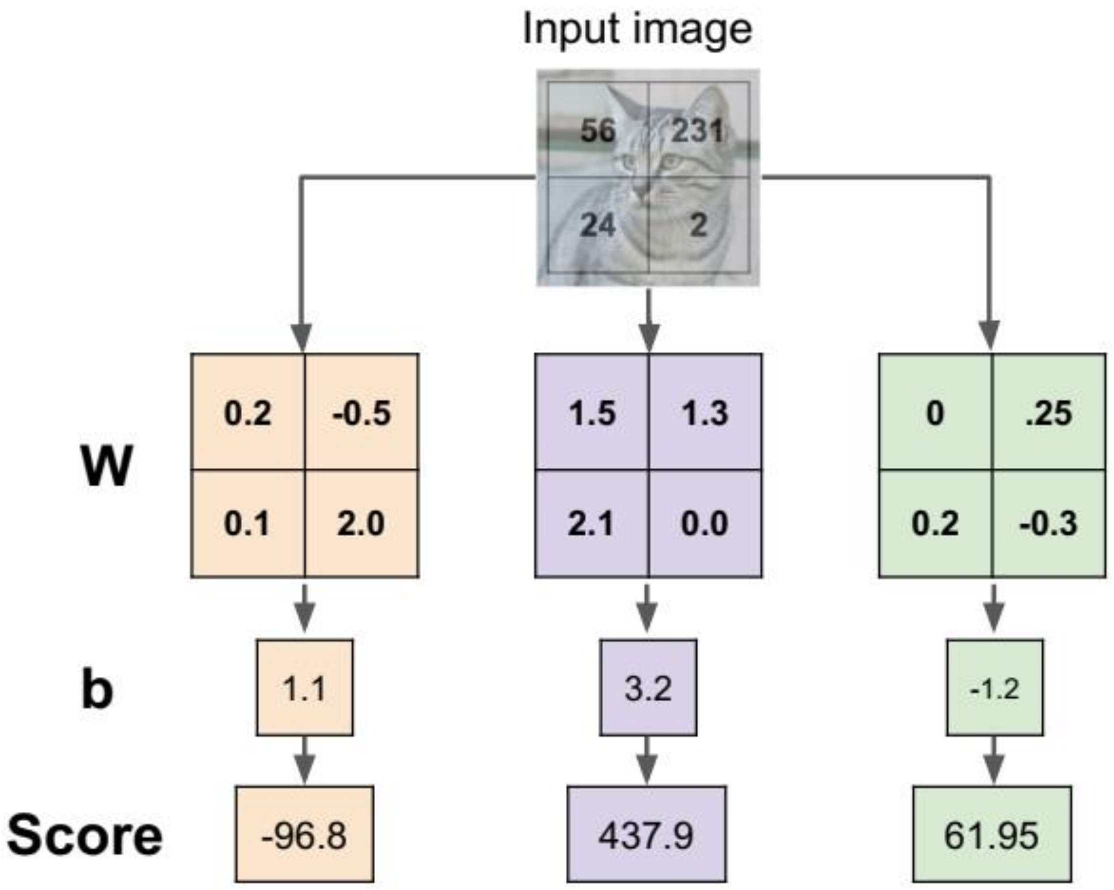
Object classification/recognition
Parametric Approach: Linear Classifier

Example with an image with 4 pixels, and 3 classes (cat/dog/ship)

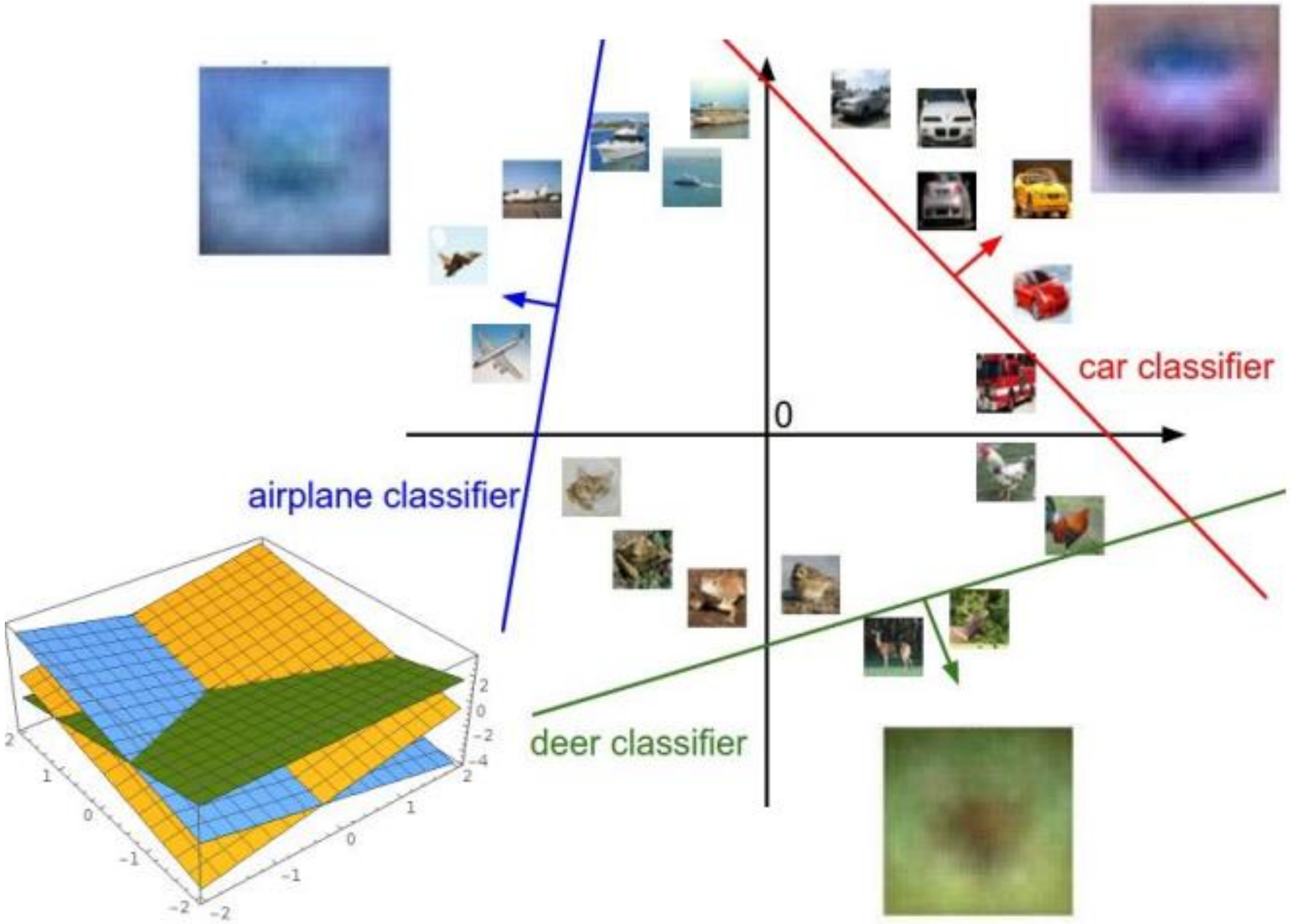
Algebraic Viewpoint

$f(x,W) = Wx$





Interpreting a Linear Classifier: Geometric Viewpoint



$$f(x,W) = Wx + b$$

Array of **32x32x3** numbers
(3072 numbers total)

Plot created using Wolfram Cloud

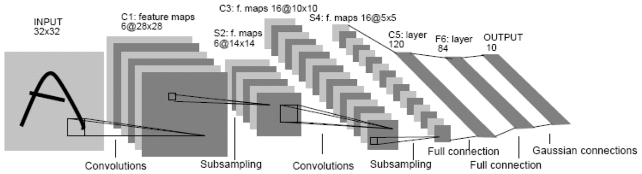
Discriminative methods

Nearest neighbor

10^6 examples

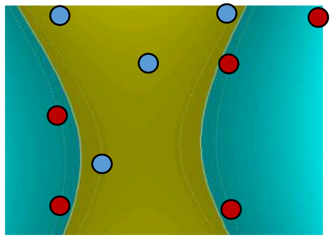
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005
…

Neural networks



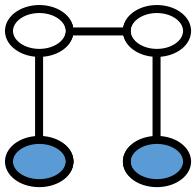
LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
…

Support Vector Machines and Kernels



Guyon, Vapnik
Heisele, Serre, Poggio, 2001
…

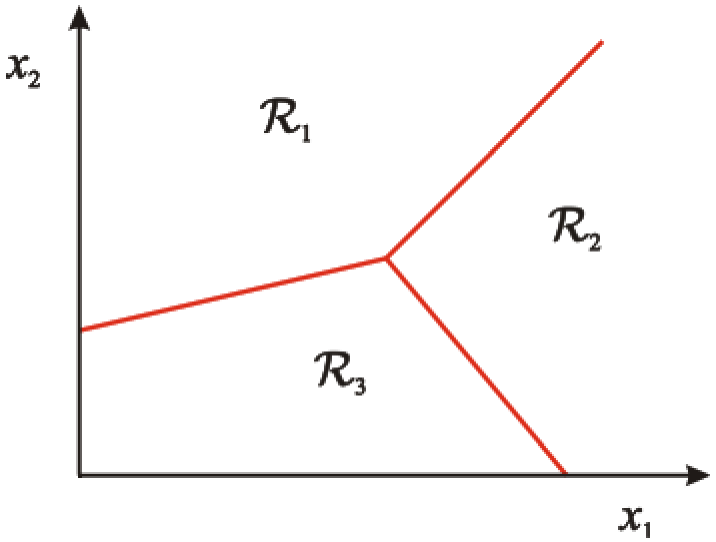
Conditional Random Fields



McCallum, Freitag, Pereira 2000
Kumar, Hebert 2003
…

Object classification/recognition

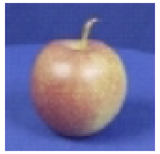
- Assign input vector to one of two or more classes
- Any decision rule divides input space into *decision regions* separated by *decision boundaries*

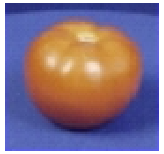


Slide credit: L. Lazebnik

Object classification/recognition

- Apply a prediction function to a feature representation of the image to get the desired output:

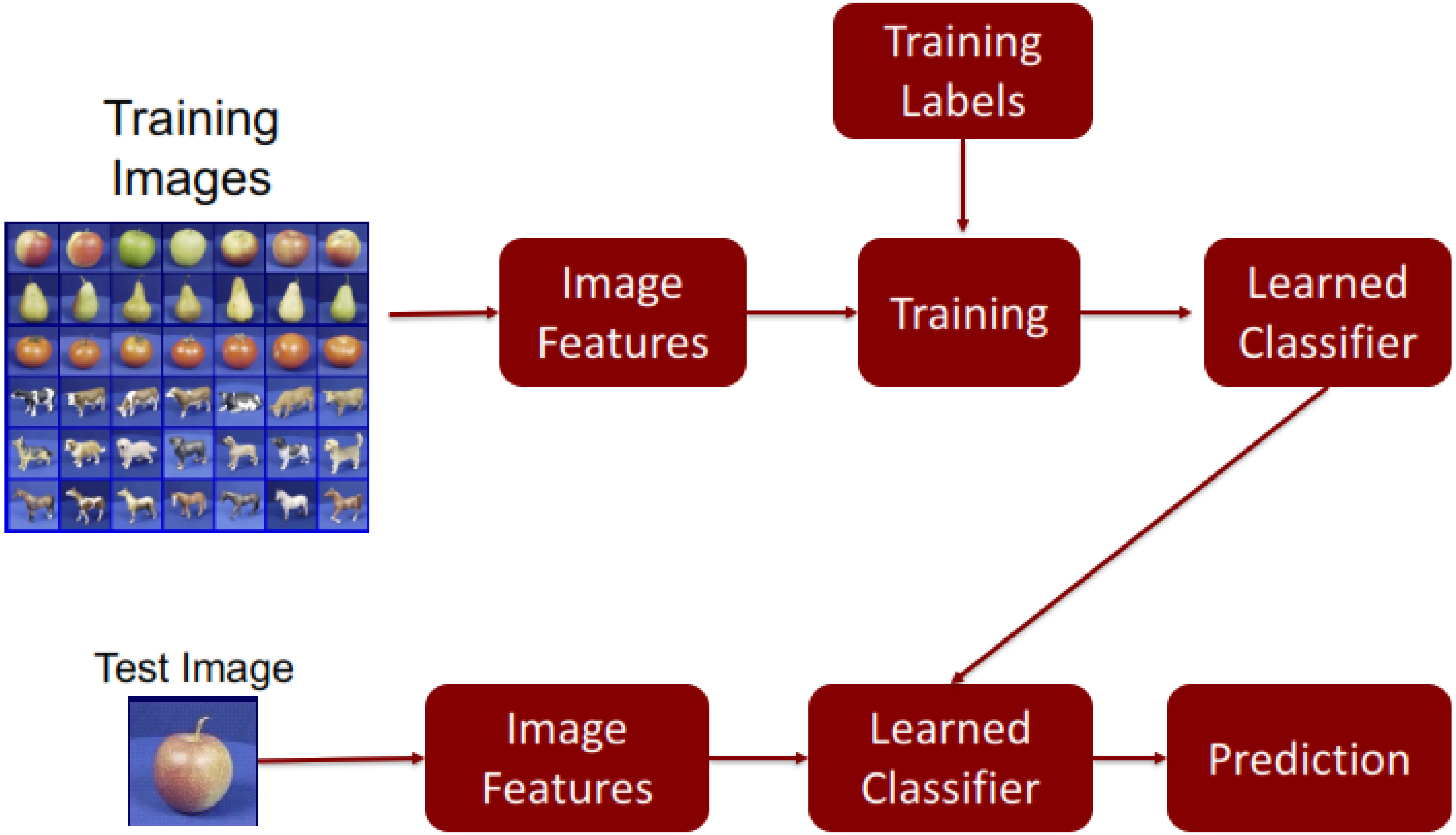
f() = “apple”

f() = “tomato”

f() = “cow”

Dataset: ETH-80, by B. Leibe   Slide credit: L. Lazebnik

A simple pipeline - Training



A simple pipeline - Training

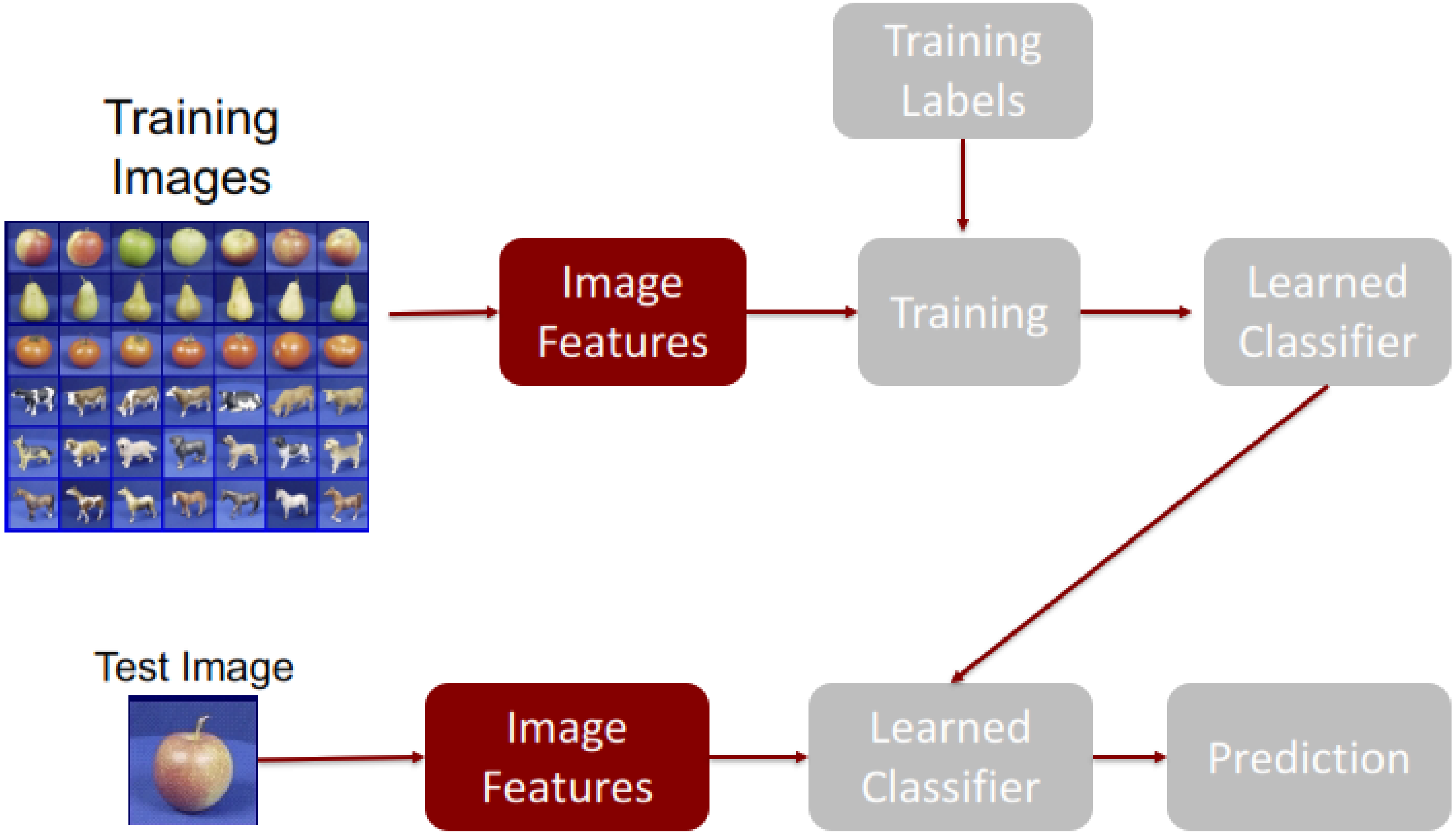
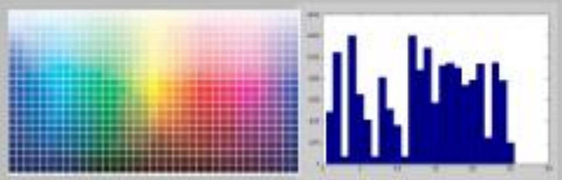


Image features

Input image

Color: Quantize RGB values

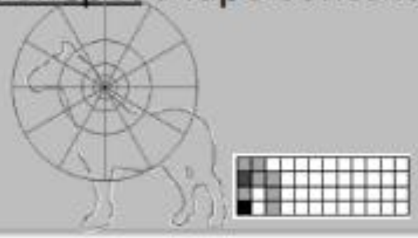


Invariance?
Translation
Scale
Rotation
Occlusion

Global shape: PCA space

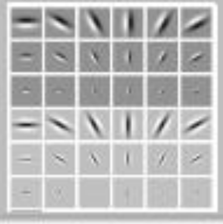
Invariance?
Translation
? Scale
Rotation
Occlusion

Local shape: shape context



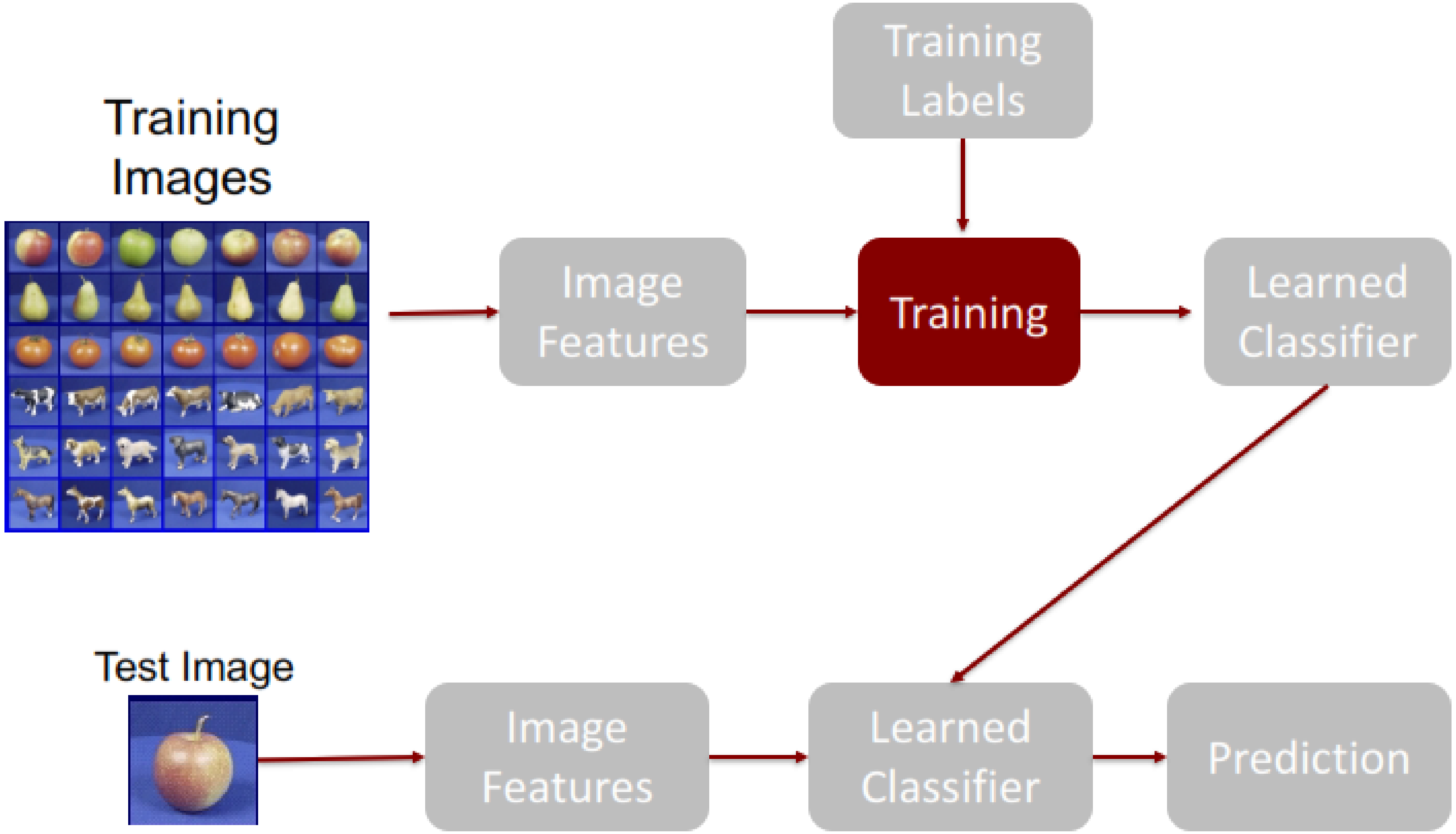
Invariance?
Translation
Scale
? Rotation (in-planar)
Occlusion

Texture: Filter banks

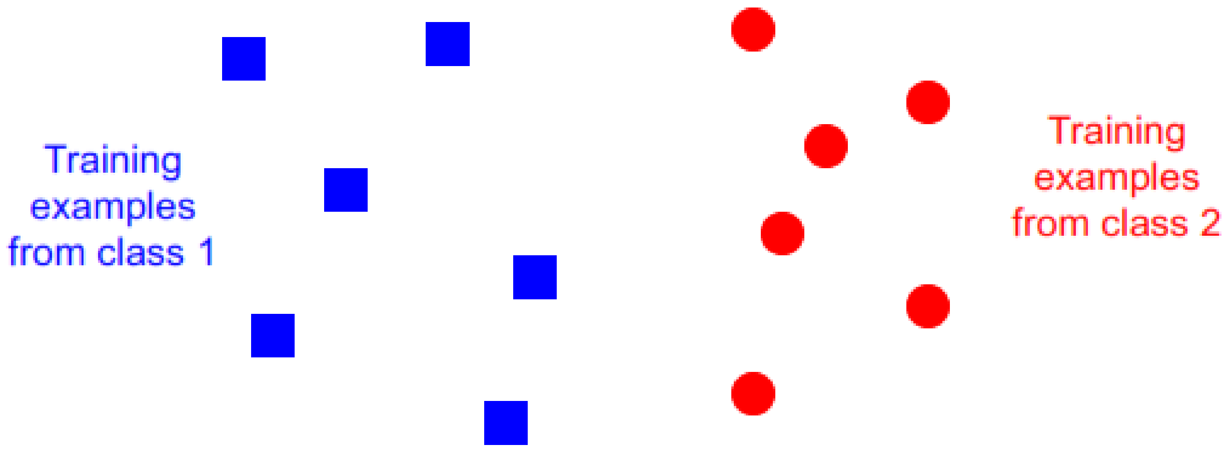


Invariance?
Translation
? Scale
? Rotation (in-planar)
Occlusion

A simple pipeline - Training

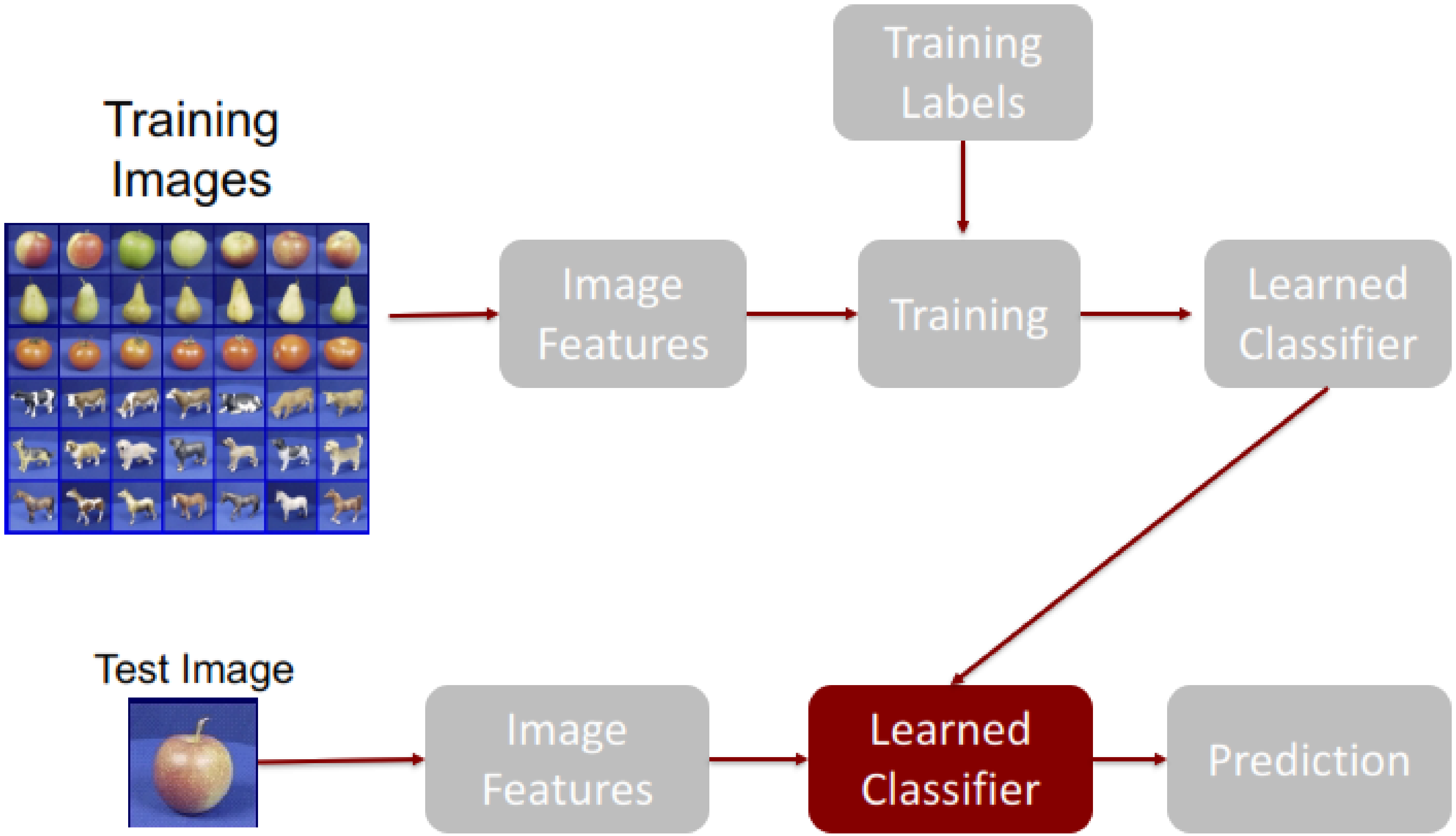


Classifiers: Nearest neighbor

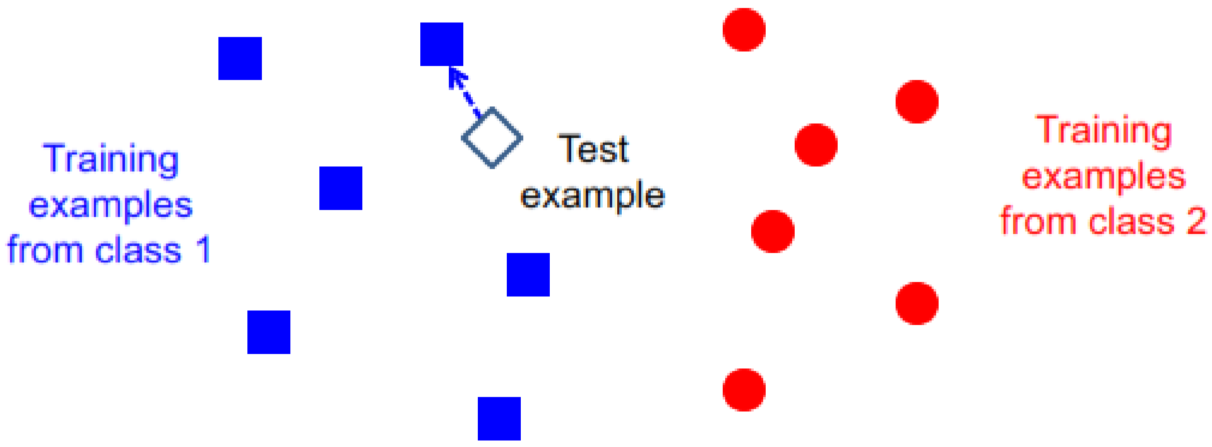


Slide credit: L. Lazebnik

A simple pipeline - Training



Classifiers: Nearest neighbor



Slide credit: L. Lazebnik

Distance Metric to compare images

L1 distance: $$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

test image

56	32	10	18
90	23	128	133
24	26	178	200
2	0	255	220

\-

training image

10	20	24	17
8	10	89	100
12	16	178	170
4	32	233	112

=

pixel-wise absolute value differences

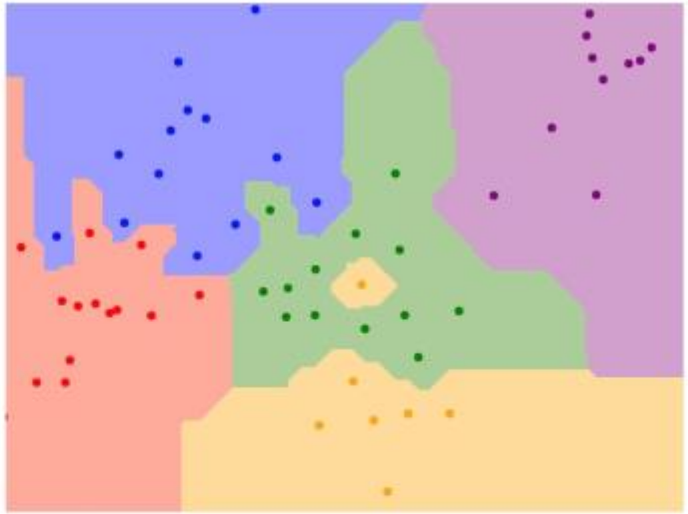
46	12	14	1
82	13	39	33
12	10	0	30
2	32	22	108

$\xrightarrow{\text{add}}$ 456

K-Nearest Neighbors: Distance Metric

L1 (Manhattan) distance

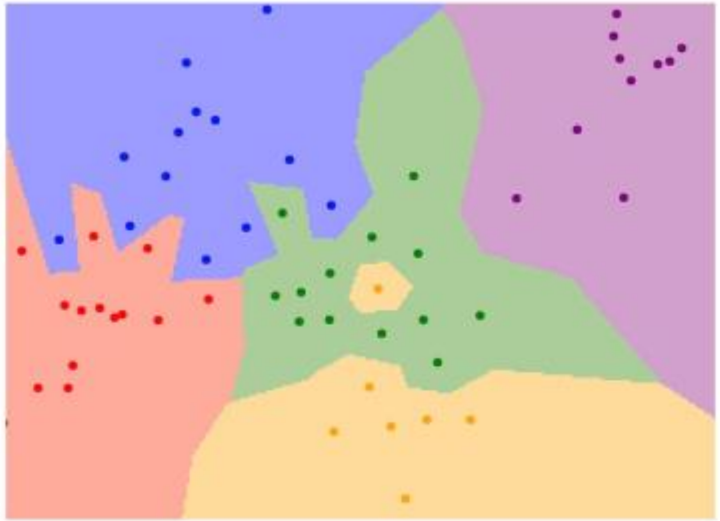
$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$



K = 1

L2 (Euclidean) distance

$$d_2(I_1, I_2) = \sqrt{\sum_p \left(I_1^p - I_2^p\right)^2}$$



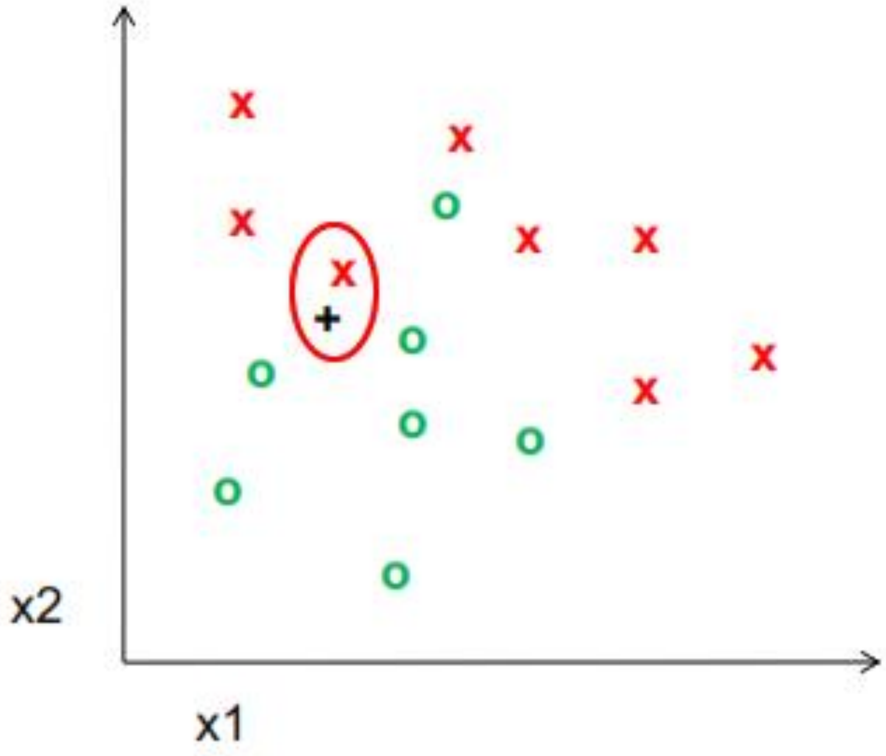
K = 1

1-nearest neighbor

Distance measure - Euclidean

$$Dist(X^n, X^m) = \sqrt{\sum_{i=1}^{D} (X_i^n - X_i^m)^2}$$

Where X^n and X^m are the n-th and m-th data points

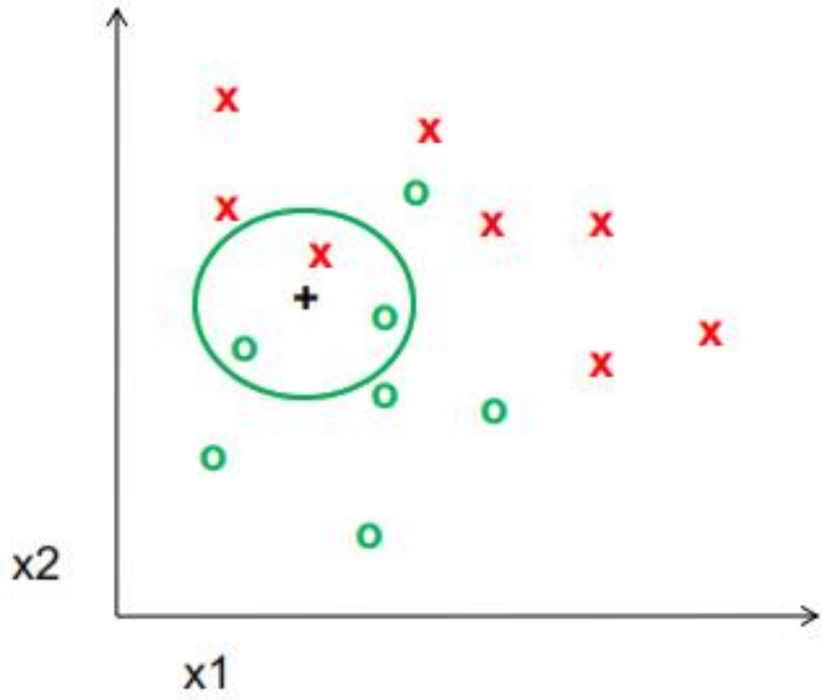


3-nearest neighbor

Distance measure - Euclidean

$$Dist(X^n, X^m) = \sqrt{\sum_{i=1}^{D} (X_i^n - X_i^m)^2}$$

Where X^n and X^m are the n-th and m-th data points

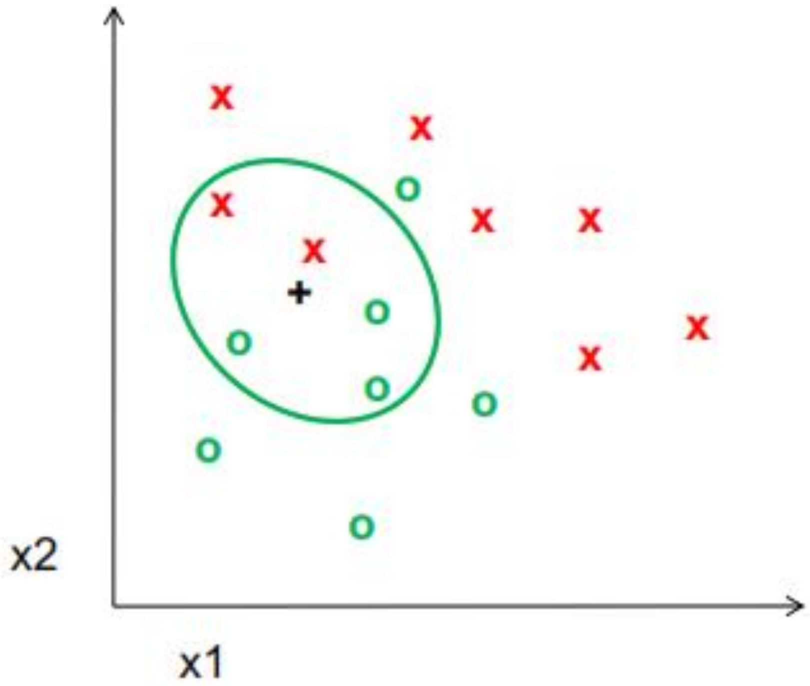


5-nearest neighbor

Distance measure - Euclidean

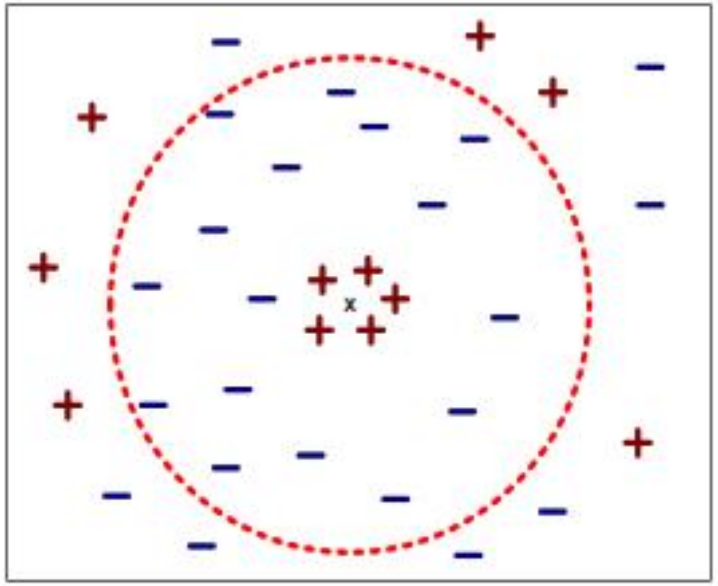
$$Dist(X^n, X^m) = \sqrt{\sum_{i=1}^{D} (X_i^n - X_i^m)^2}$$

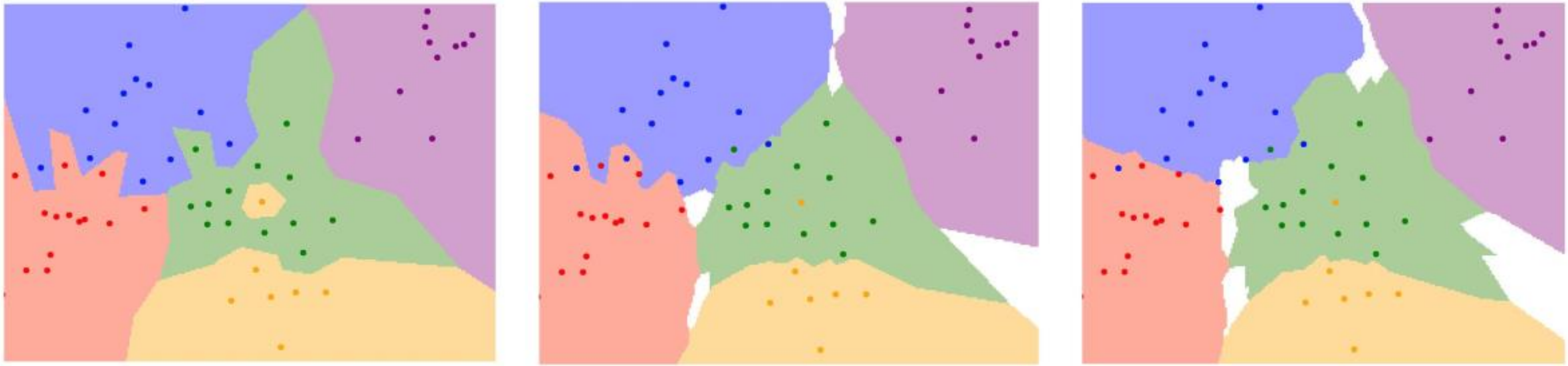
Where X^n and X^m are the n-th and m-th data points



K-NN: issues to keep in mind

- Choosing the value of k:
  - If too small, sensitive to noise points
  - If too large, neighborhood may include points from other classes

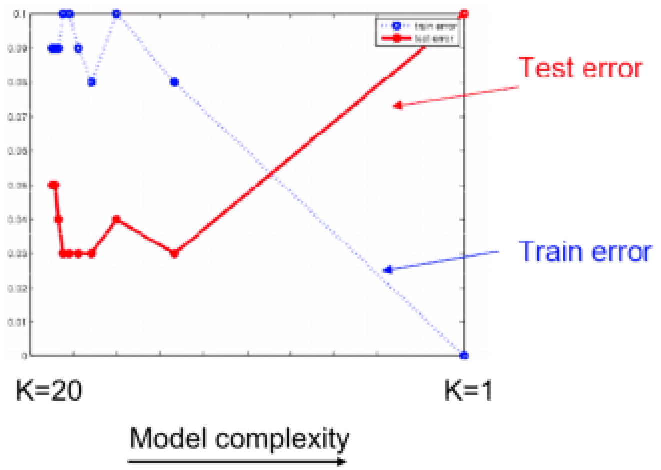




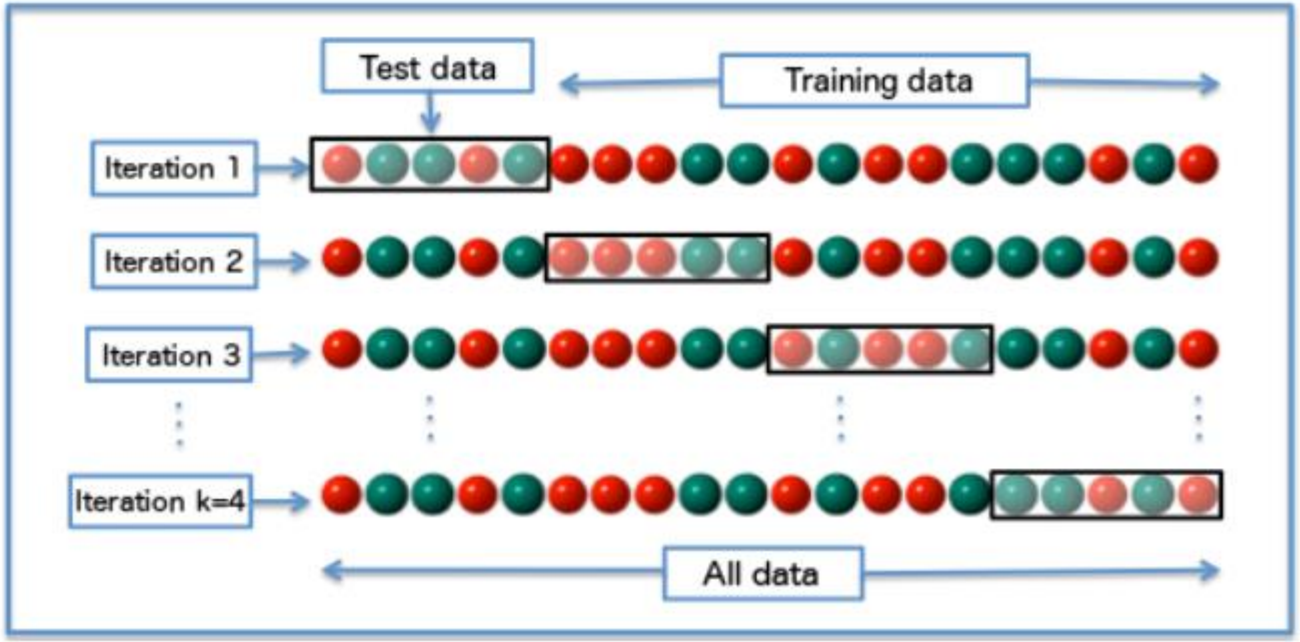
K = 1 K = 3 K = 5

K-NN: issues to keep in mind

- Choosing the value of k:
  - If too small, sensitive to noise points
  - If too large, neighborhood may include points from other classes
  - Solution: cross validate!



Cross validation



Many classifiers to choose from

- **K-nearest neighbor**
- SVM
- Neural networks
- Naïve Bayes
- Bayesian network
- Logistic regression
- Randomized Forests
- Boosted Decision Trees
- RBMs
- Etc.

Which is the best one?

Slide credit: D. Hoiem

Generalization

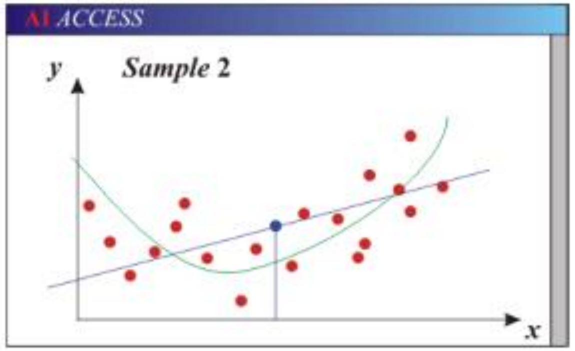
Training set (labels known)

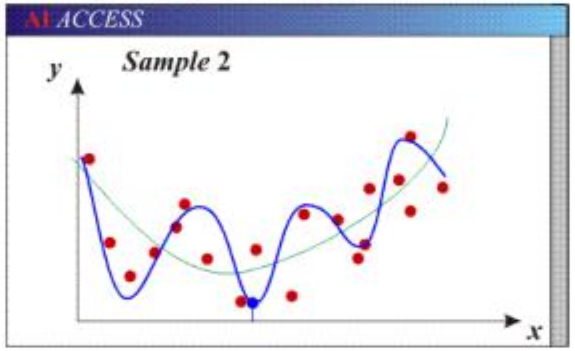
Test set (labels unknown)

- How well does a learned model generalize from the data it was trained on to a new test set?

Slide credit: L. Lazebnik

Bias-Variance Trade-off





- Models with too few parameters are inaccurate because of a large bias (not enough flexibility).
- Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample).

Slide credit: D. Hoiem

Bias versus variance

- Components of generalization error
  - Bias: how much the average model over all training sets differ from the true model?
    - Error due to inaccurate assumptions/simplifications made by the model
  - Variance: how much models estimated from different training sets differ from each other

- Underfitting: model is too “simple ” to represent all the relevant class characteristics

  - High bias and low variance

  – High training error and high test error
- Overfitting: model is too “complex” and fits irrelevant characteristics (noise) in the data

  – Low bias and high variance

  – Low training error and high test error

How do we evaluate object detection?

predictions

ground truth

How do we evaluate object detection?

predictions

ground truth

True positive:

- The overlap of the prediction with the ground truth is MORE than 0.5

How do we evaluate object detection?

predictions

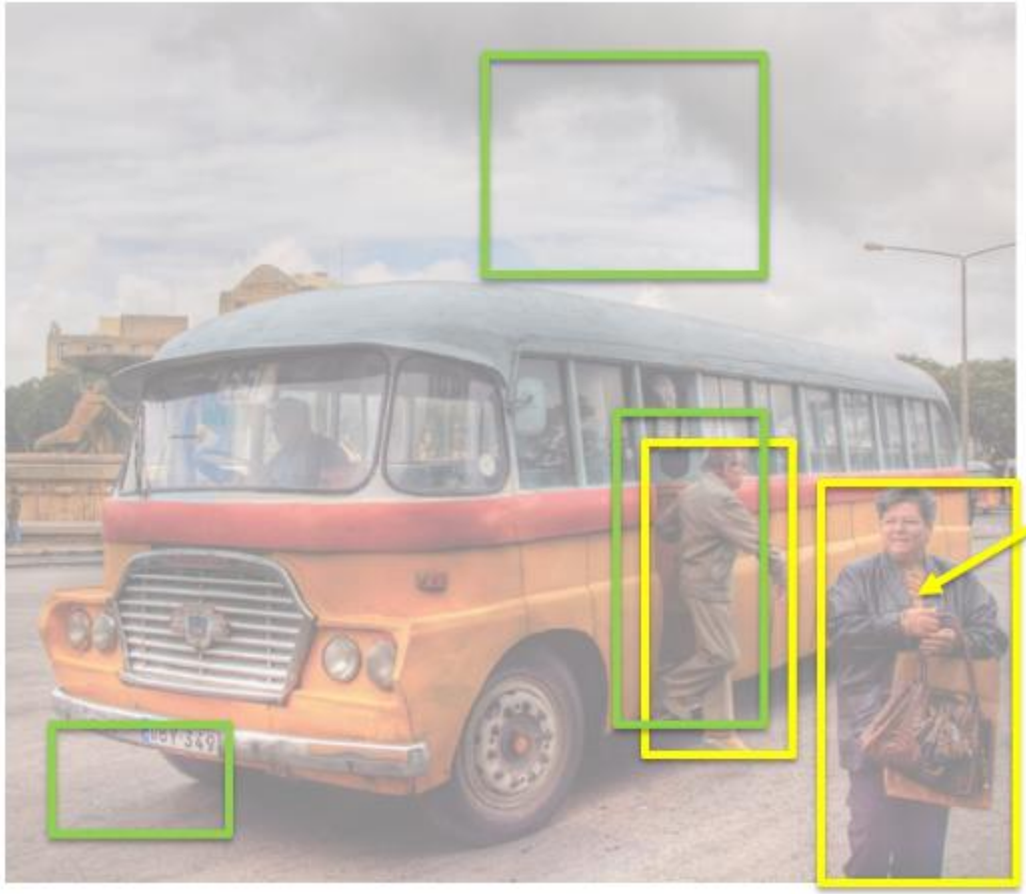
ground truth

True positive:

False positive:

- The overlap of the prediction with the ground truth is LESS than 0.5

How do we evaluate object detection?



predictions

ground truth

True positive:

False positive:

False negative:

- The objects that our model doesn't find

How do we evaluate object detection?

predictions

ground truth

True positive:

False positive:

False negative:

- The objects that our model doesn't find

What is a True Negative?

	Predicted 1	Predicted 0
True 1	true positive	false negative
True 0	false positive	true negative

	Predicted 1	Predicted 0
True 1	TP	FN
True 0	FP	TN

	Predicted 1	Predicted 0
True 1	hits	misses
True 0	false alarms	correct rejections

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

How do we evaluate object detection?

predictions

ground truth

True positive: 1

False positive: 2

False negative: 1

So what is the

- precision?

- recall?

Classification metrics

- Precision versus recall
  - Precision:

    –how many of the object detections are correct?
  - Recall:

    –how many of the ground truth objects can the model detect?

- F1 score: useful when you want to seek a balance between Precision and Recall

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

In reality, our model makes a lot of predictions with varying scores between 0 and 1

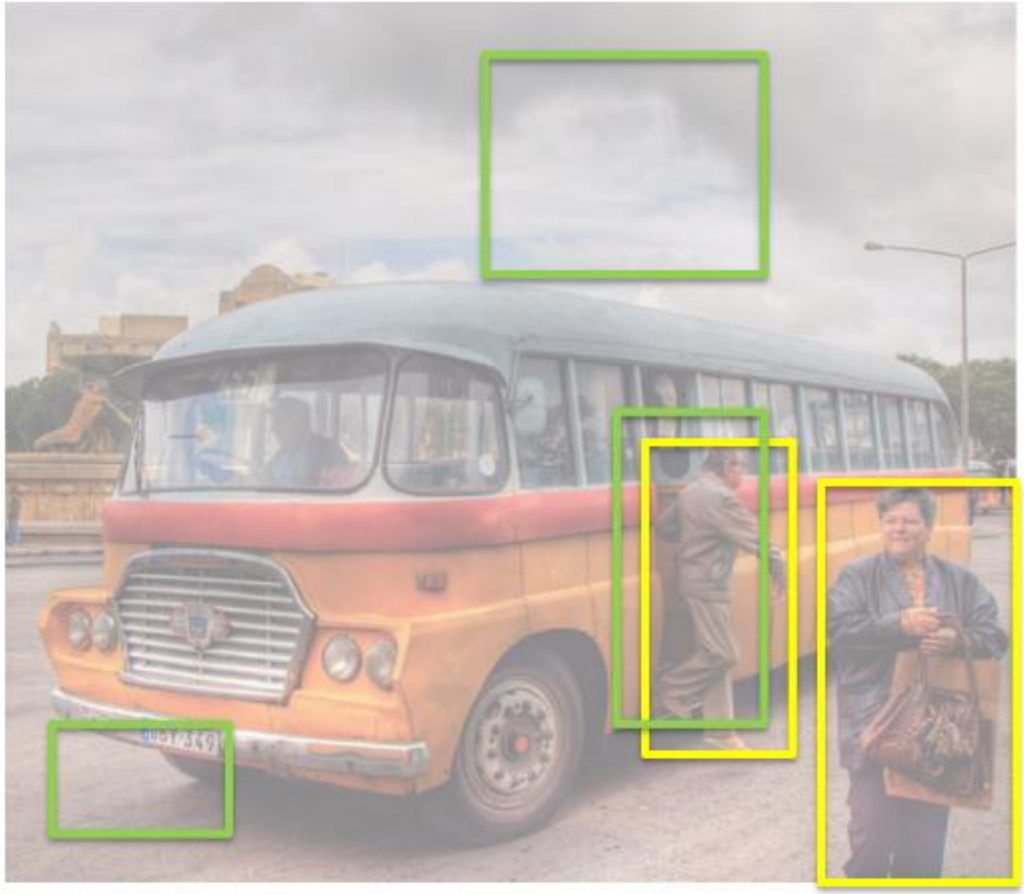
predictions

ground truth

Here are all the boxes that are predicted with score > 0.

This means that our
- Recall is perfect!
- But our precision is BAD!

How do we evaluate object detection?



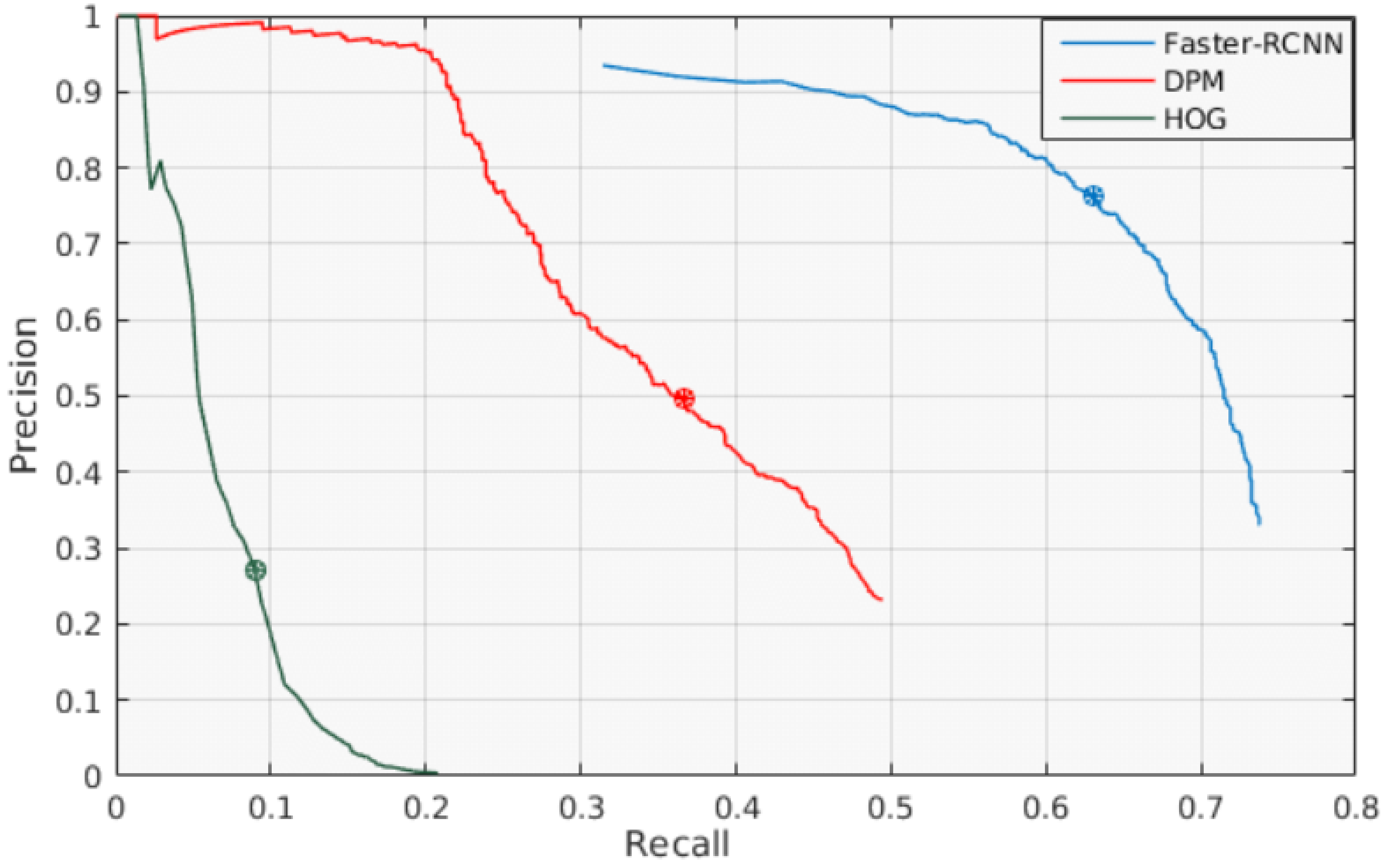
predictions

ground truth

Here are all the boxes that are predicted with score > 0.5

We are setting a threshold of 0.5

Which model is the best?



True Positives - Person

UoCTTI_LSVM-MDPM

MIZZOU_DEF-HOG-LBP

NECUIUC_CLS-DTCT

False Positives - Person

UoCTTI_LSVM-MDPM





MIZZOU_DEF-HOG-LBP





NECUIUC_CLS-DTCT





Non-maximal suppression (NMS)

Many detections above threshold.

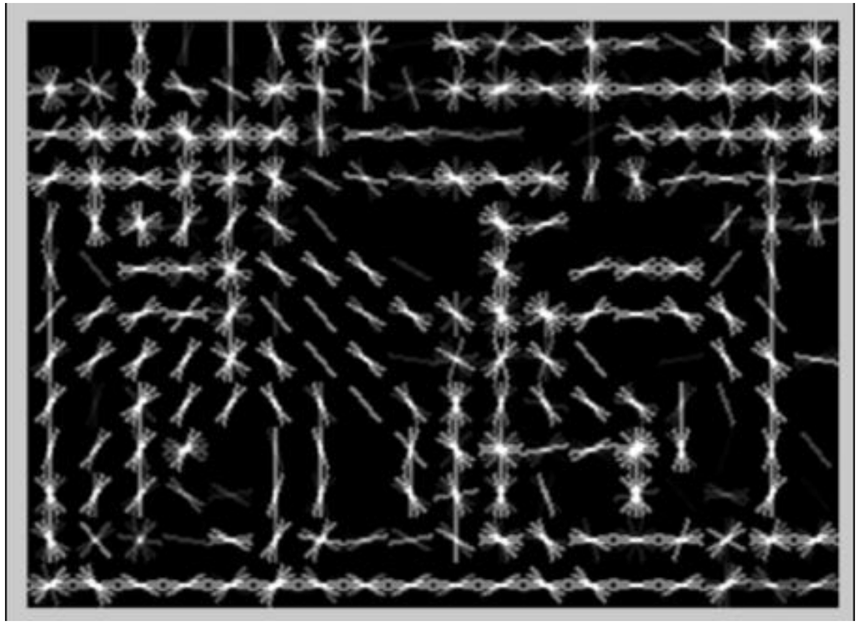
Detections after NMS.

Example: Dalal-Triggs pedestrian detector

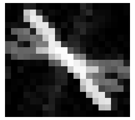
1. Extract fixed-sized (64x128 pixel) window at each position and scale
2. Compute HOG (histogram of gradient) features within each window
3. Score the window with a linear SVM classifier
4. Perform non-maxima suppression to remove overlapping detections with lower scores

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

Histograms of oriented gradients (HOG)

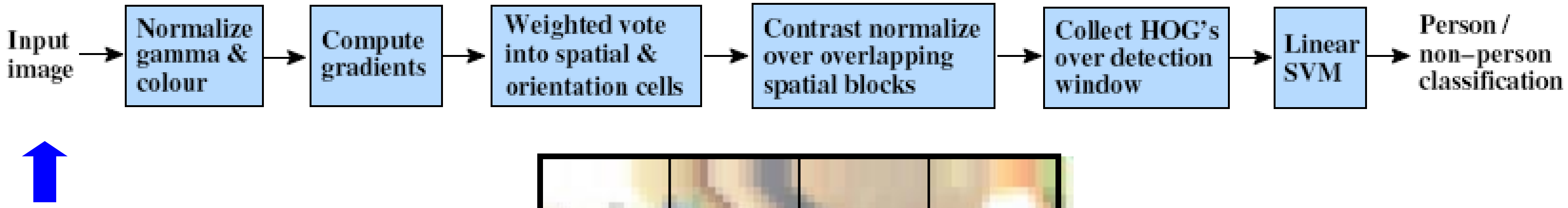


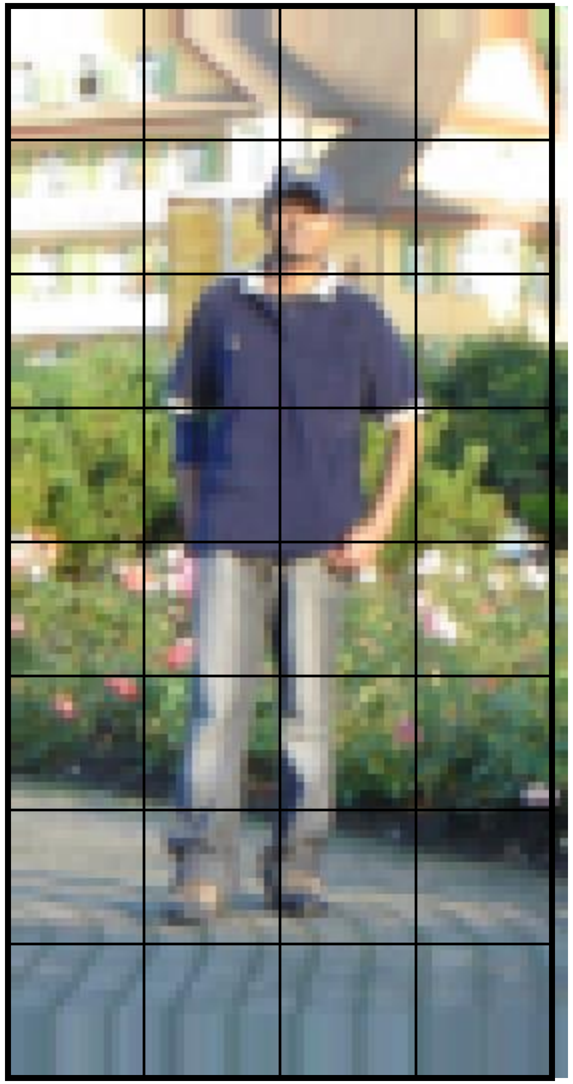
Bin gradients from 8x8 pixel neighborhoods into 9 orientations



(Dalal & Triggs CVPR 05)

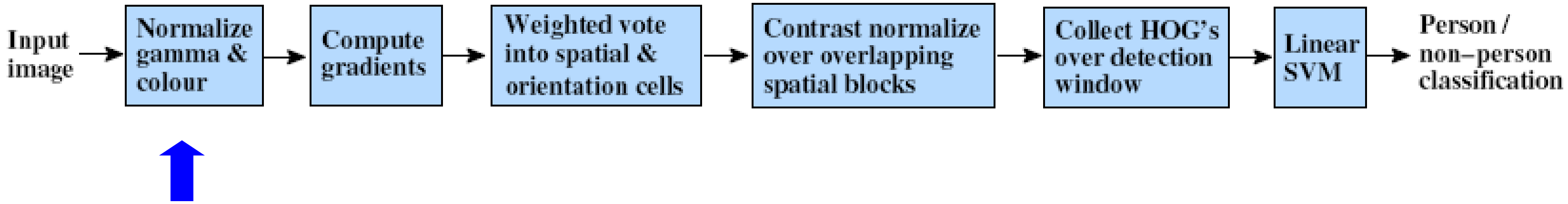
Source: Deva Ramanan





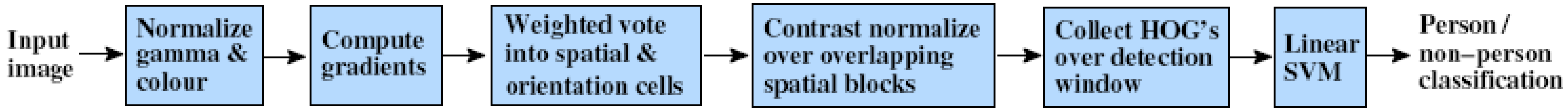
Slides by Pete Barnum

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05



- Tested with
  - RGB
  - LAB
  - Grayscale

Slightly better performance vs. grayscale



Outperforms

-1	0	1

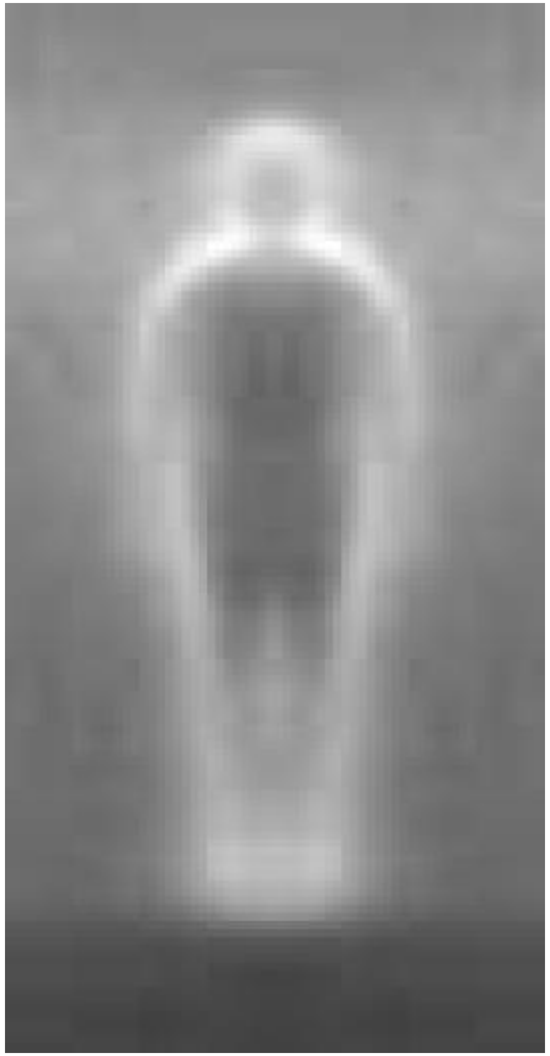
centered

-1	1

uncentered

1	-8	0	8	-1

cubic-corrected



0	1
-1	0

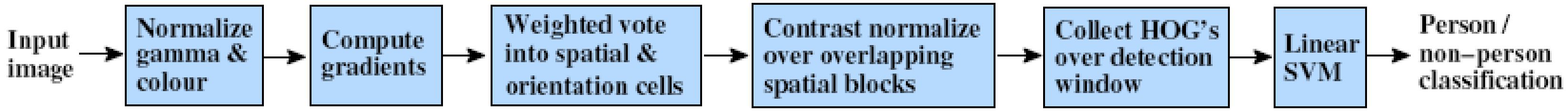
diagonal

-1	0	1
-2	0	2
-1	0	1

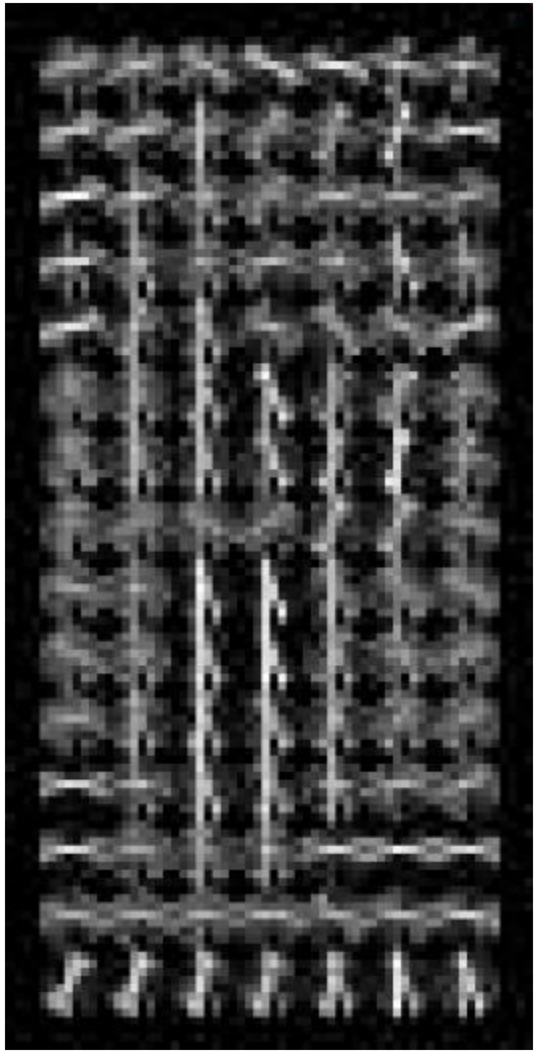
Sobel

Slides by Pete Barnum

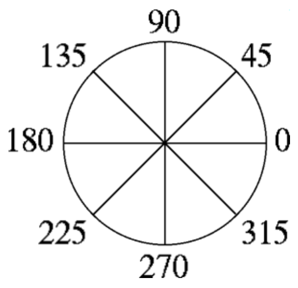
Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05



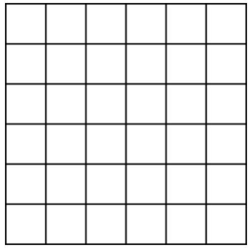
- Histogram of gradient orientations



Orientation: 9 bins (for unsigned angles)

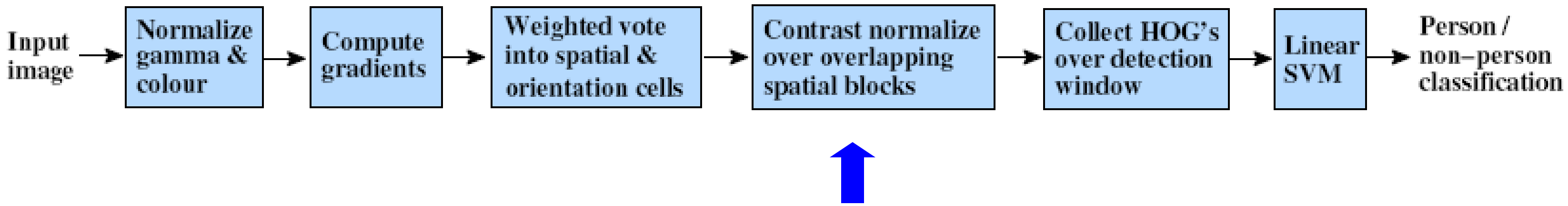


Histograms in 8x8 pixel cells

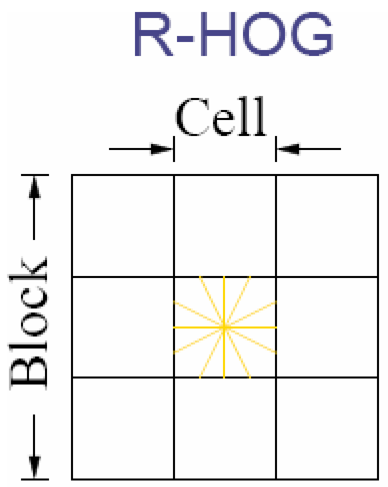


- Votes weighted by magnitude
- Bilinear interpolation between cells

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

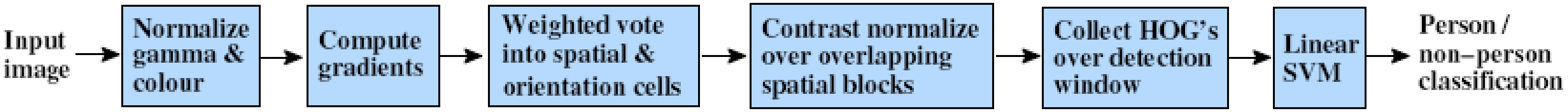


Normalize with respect to surrounding cells

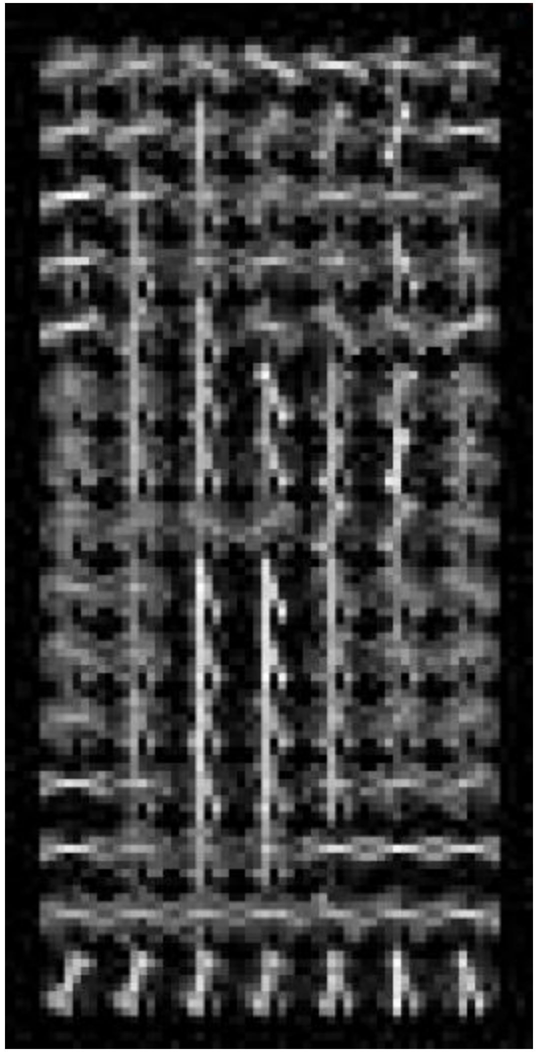


$$L2-norm : v \longrightarrow v/\sqrt{||v||_2^2+\epsilon^2}$$

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05



X=



orientations

features = 15 x 7 x 9 x 4 = 3780

cells

normalizations by neighboring cells

Slides by Pete Barnum

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

Histograms of oriented gradients (HOG)

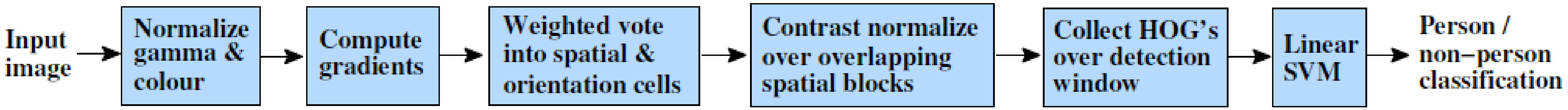
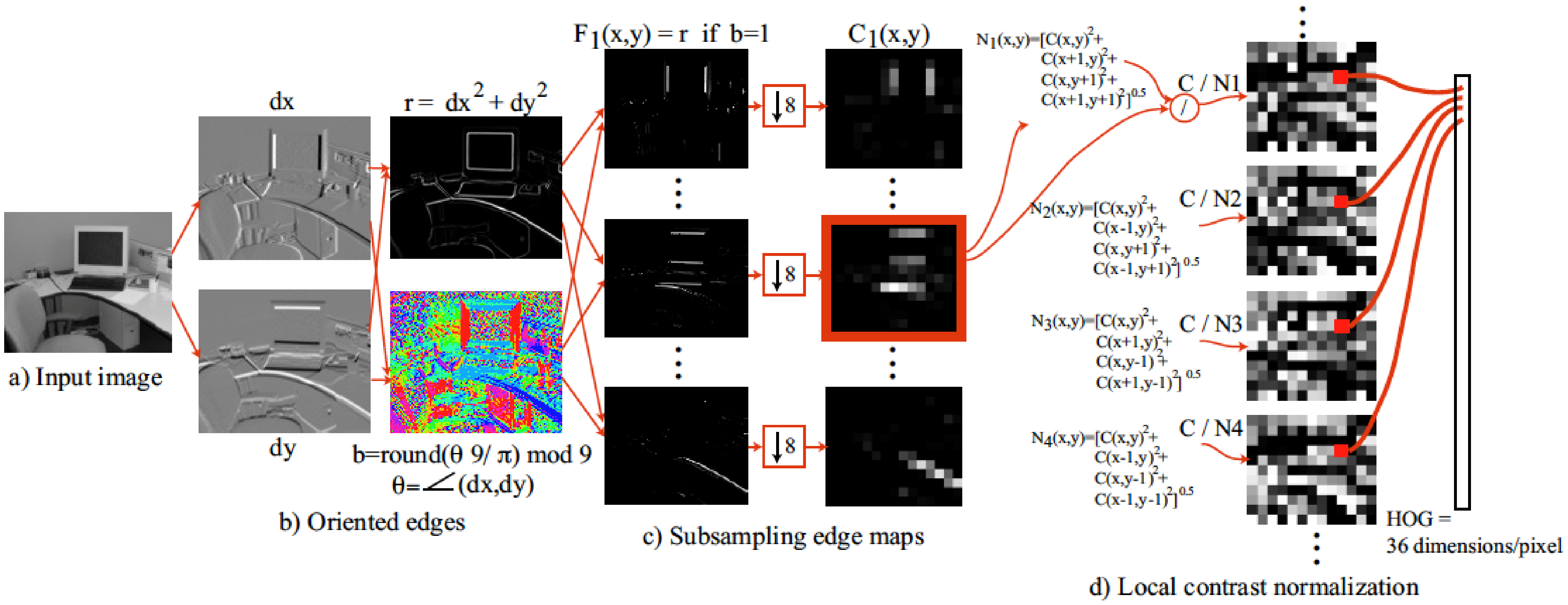


Figure 1. An overview of our feature extraction and object detection chain. The detector window is tiled with a grid of overlapping blocks in which Histogram of Oriented Gradient feature vectors are extracted. The combined vectors are fed to a linear SVM for object/non-object classification. The detection window is scanned across the image at all positions and scales, and conventional non-maximum suppression is run on the output pyramid to detect object instances, but this paper concentrates on the feature extraction process.



Training set

+

−

SVM

A Support Vector Machine (SVM) learns a classifier with the form:

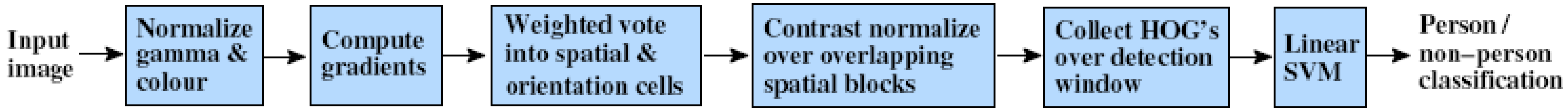
$$H(x) = \sum_{m=1}^{M} a_m y_m k(x, x_m)$$

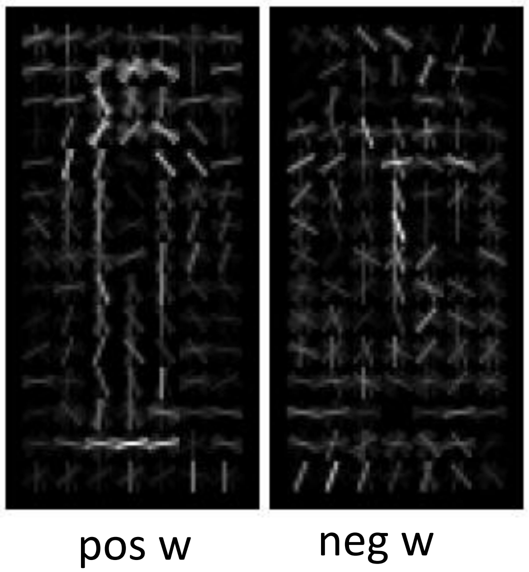
Where $\{x_m, y_m\}$, for m = 1 . . .M, are the training data with x_m being the input feature vector and y_m = +1,-1 the class label. $k(x, x_m)$ is the kernel and it can be any symmetric function satisfying the Mercer Theorem.

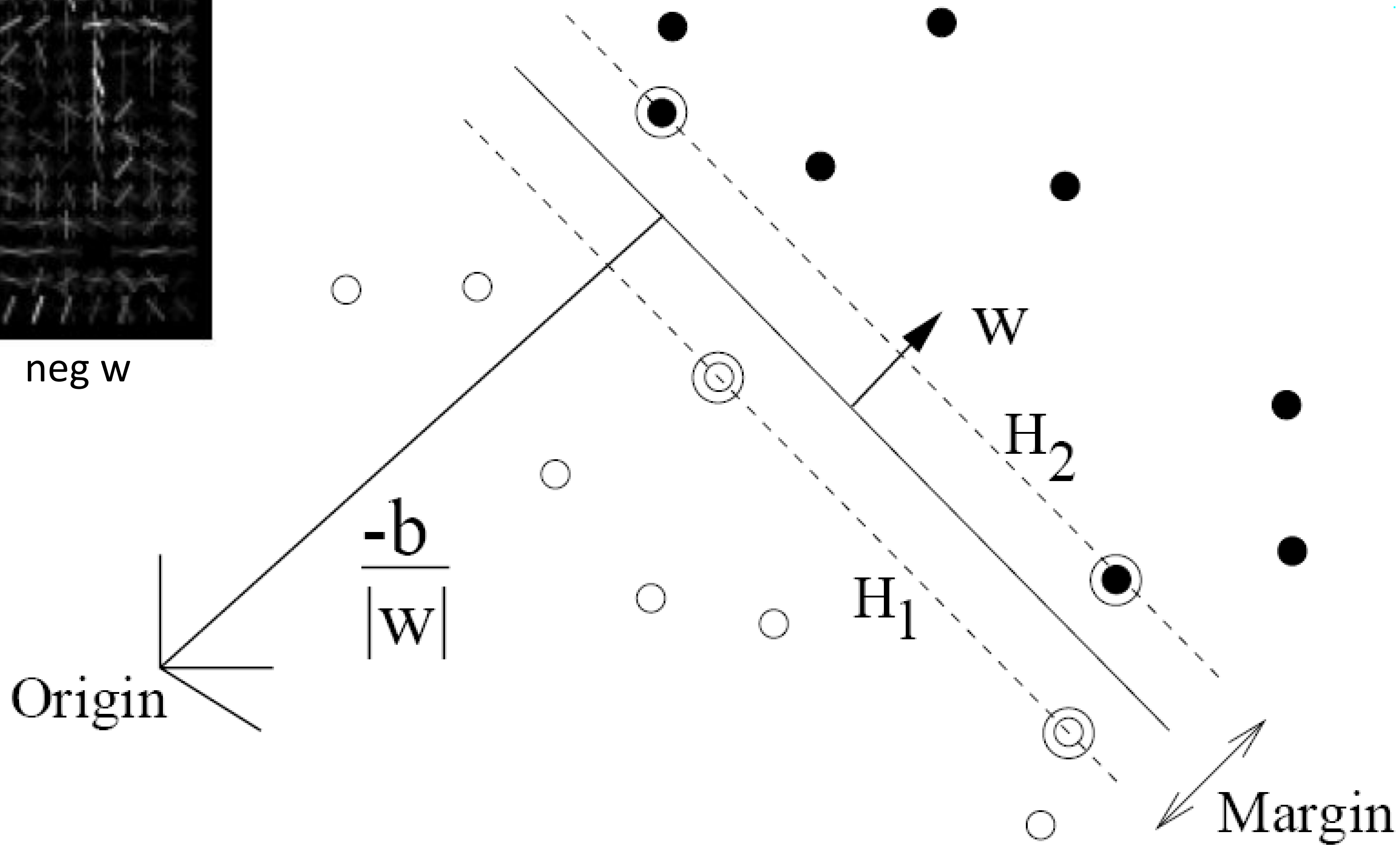
The classification is obtained by thresholding the value of H(x).

There is a large number of possible kernels, each yielding a different family of decision boundaries:

- Linear kernel: $k(x, x_m) = x^T x_m$
- Radial basis function: $k(x, x_m) = \exp(-|x - x_m|^2/\sigma^2)$.
- Histogram intersection: $k(x,x_m) = \mathrm{sum}_i(\min(x(i), x_m(i)))$



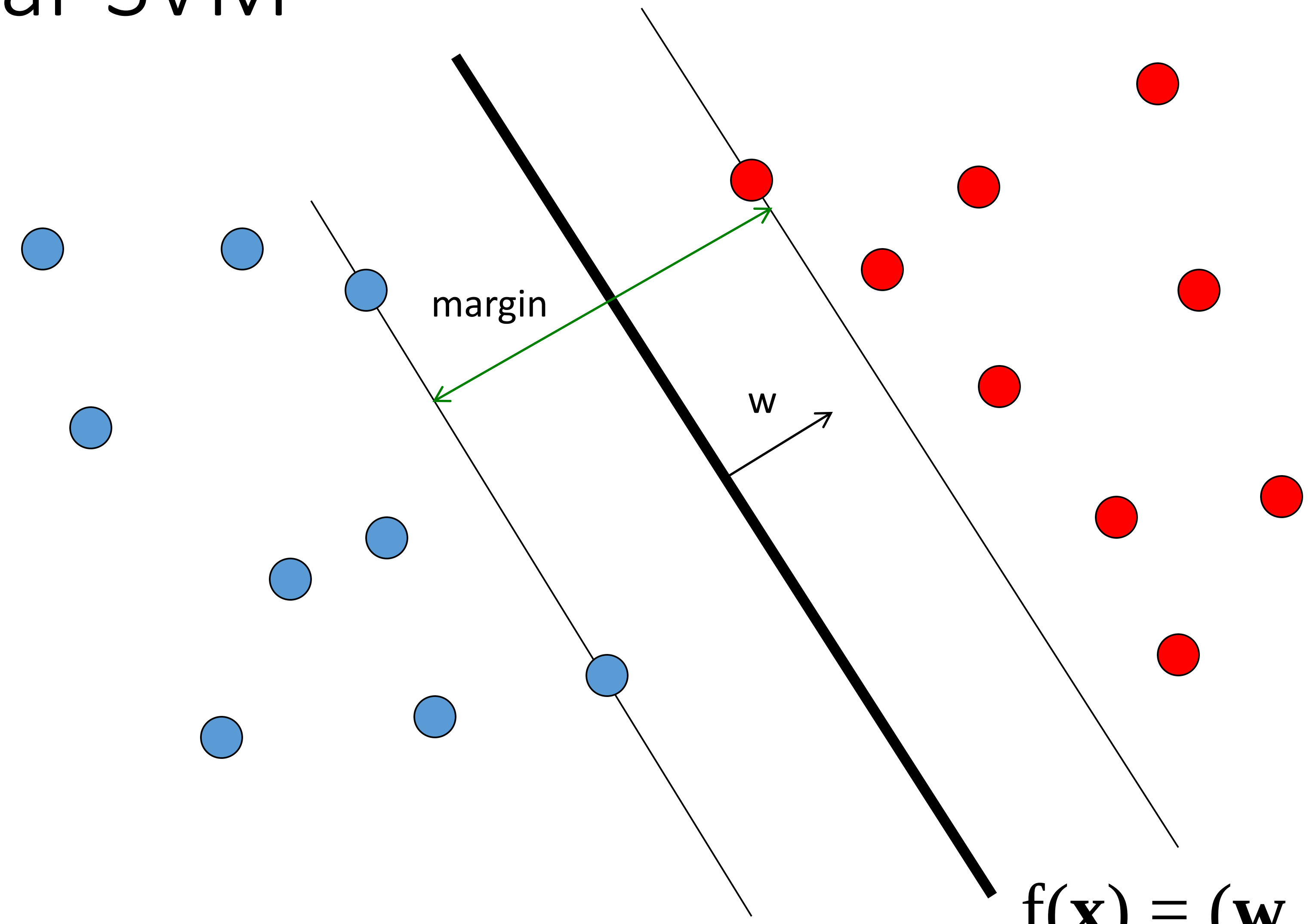




Slides by Pete Barnum

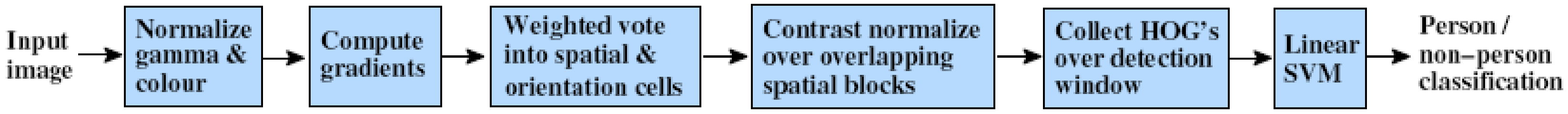
Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

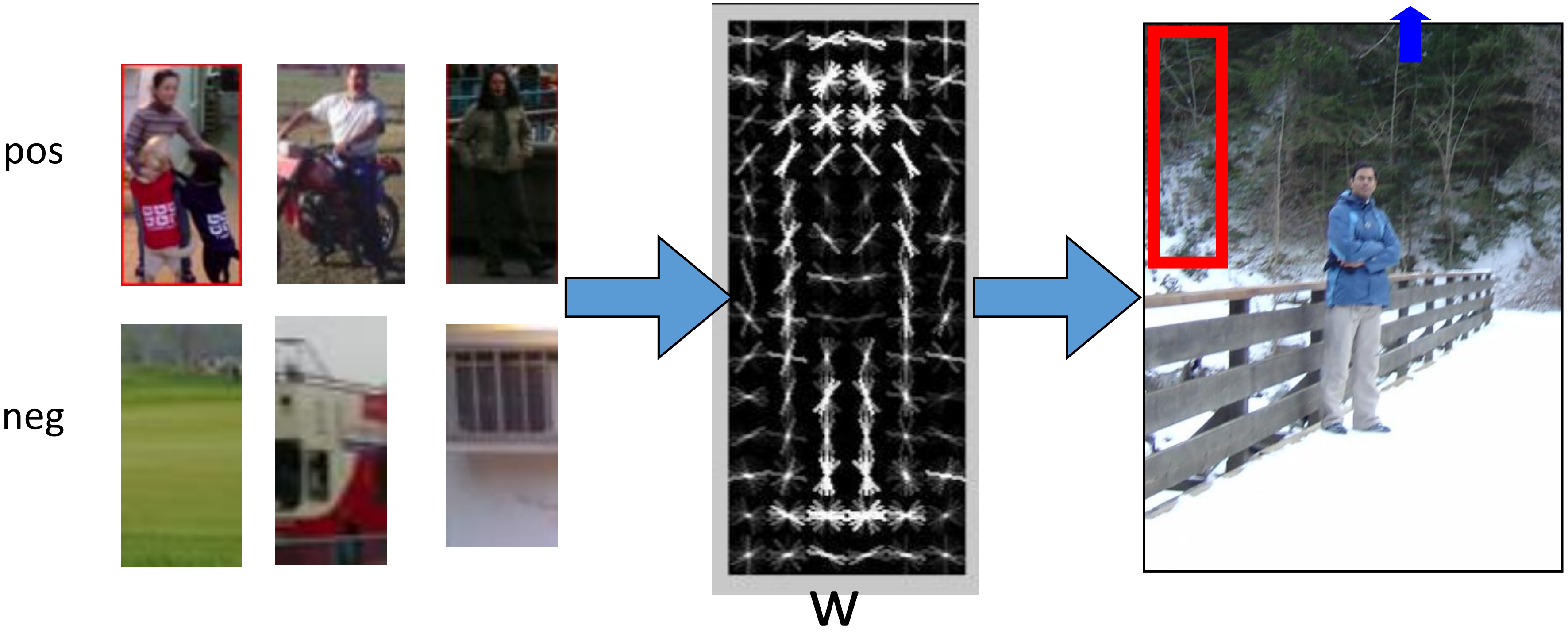
Linear SVM



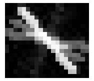
$$f(\mathbf{x}) = (\mathbf{w} \mathbin{.} \mathbf{x} + b)$$

Scanning-window templates





w = weights for orientation and spatial bins



w·x > 0

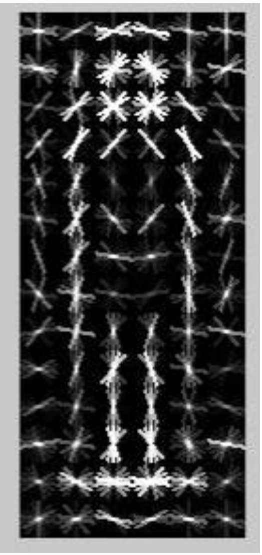
How to interpret positive and negative weights?

$w \cdot x > 0$

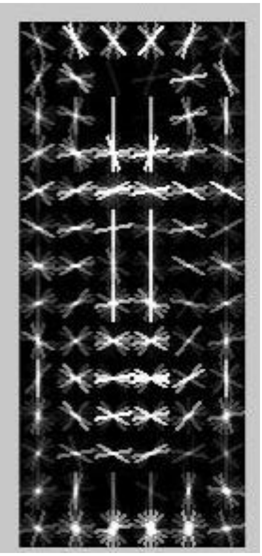
$(w_{pos} - w_{neg}) \cdot x > 0$

$w_{pos} \cdot x > w_{neg} \cdot x$

Pedestrian template



>



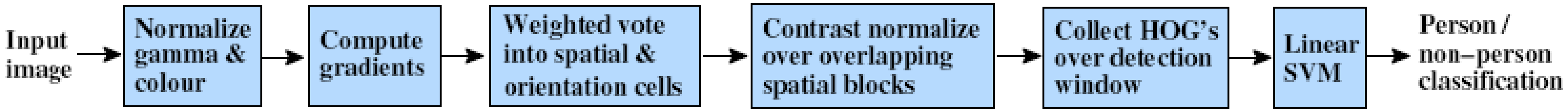
Pedestrian background template

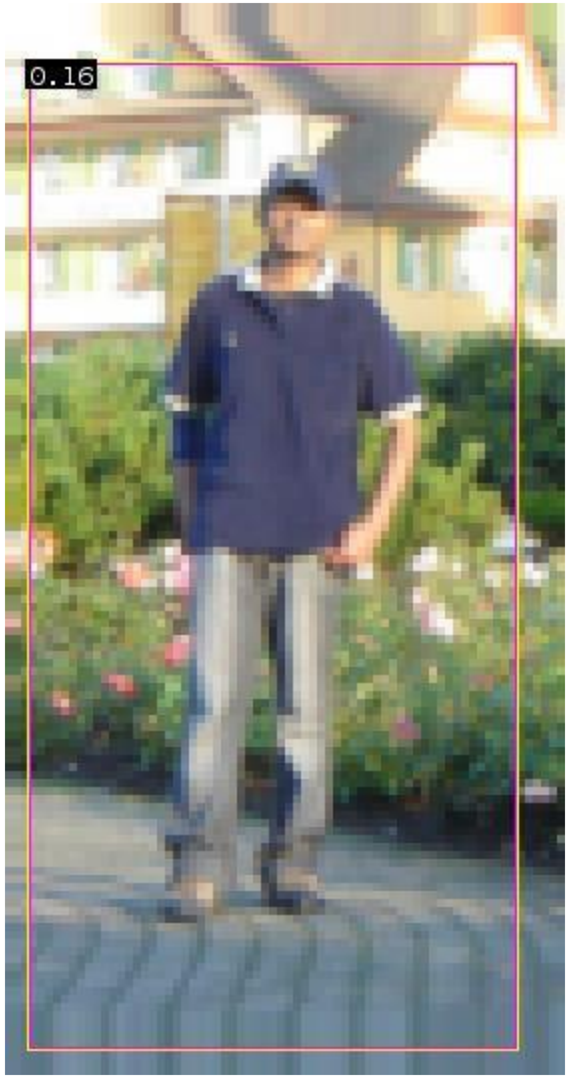
w_{pos}, w_{neg} = weighted average of positive, negative support vectors

Right approach is to compete pedestrian, pillar, doorway... models

Background class is hard to model - easier to penalize particular vertical edges

Source: Deva Ramanan





$$0.16 = w^T x - b$$

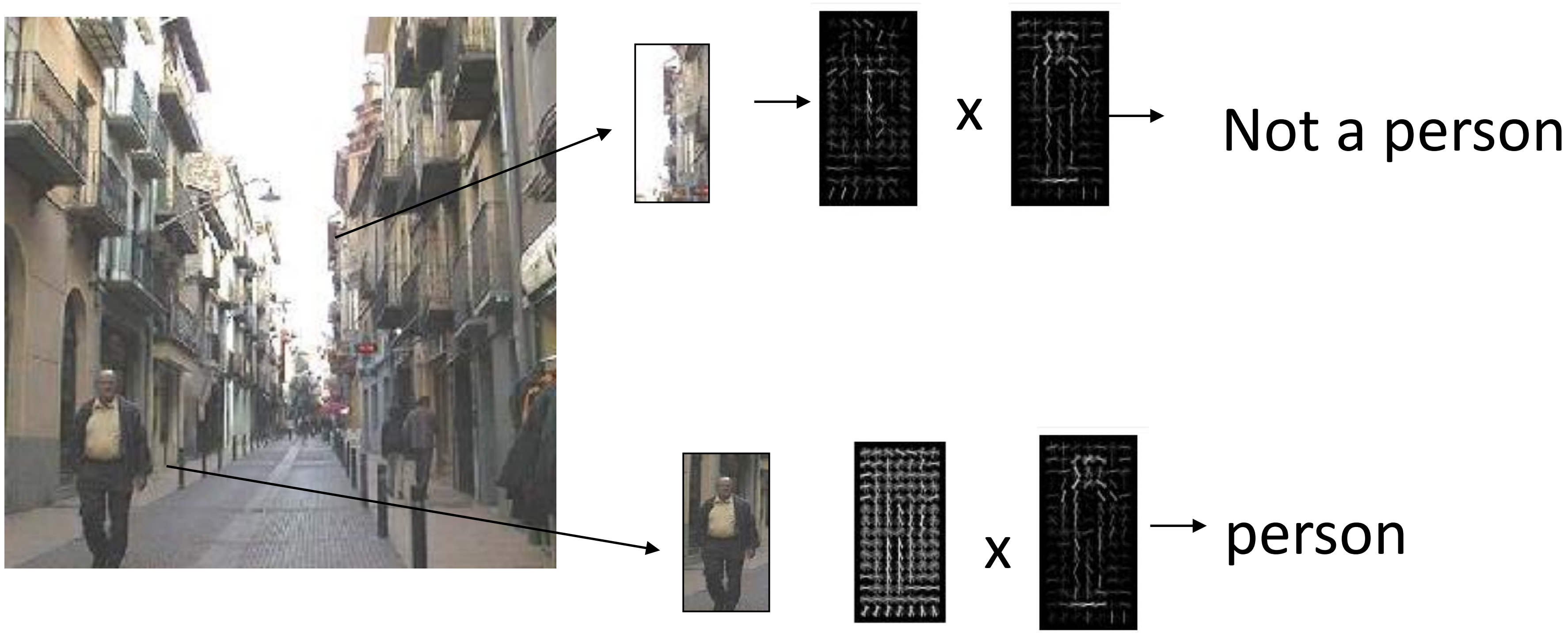
$$sign(0.16) = 1$$

$=>$ pedestrian

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

Histograms of oriented gradients

Dalal & Trigs, 2006



Detection examples

Each window is separately classified