Categorizing objects: global and part-based models of appearance

Kristen Grauman

UT-Austin

Generic categorization problem

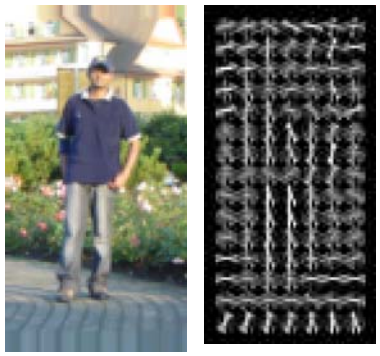
Challenges: robustness

Realistic scenes are crowded, cluttered, have overlapping objects.

Generic category recognition: basic framework

- Build/train object model
  - Choose a representation
  - Learn or fit parameters of model / classifier
- Generate candidates in new image
- Score the candidates

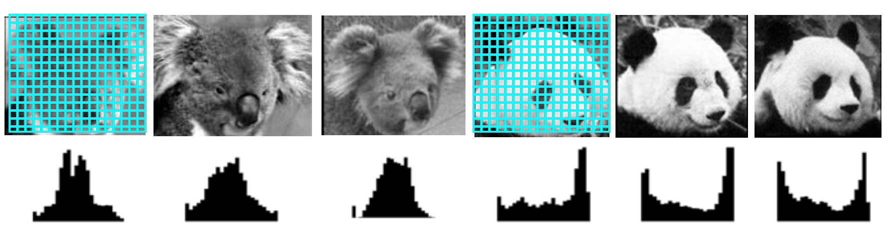
Generic category recognition: representation choice



Window-based

Part-based

Window-based models
Building an object model



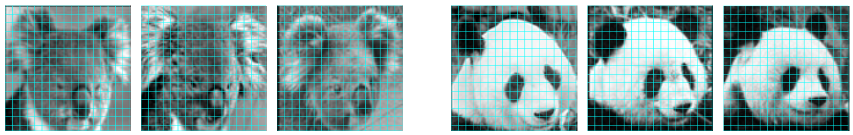
Simple holistic descriptions of image content

- **grayscale / color histogram**
- **vector of pixel intensities**

Kristen Grauman

Window-based models
Building an object model

- Pixel-based representations sensitive to small shifts

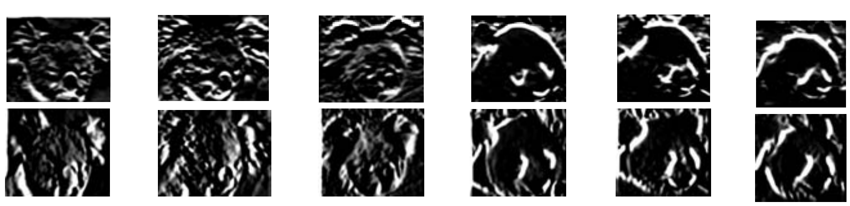


- Color or grayscale-based appearance description can be sensitive to illumination and intra-class appearance variation

Kristen Grauman

Window-based models
Building an object model

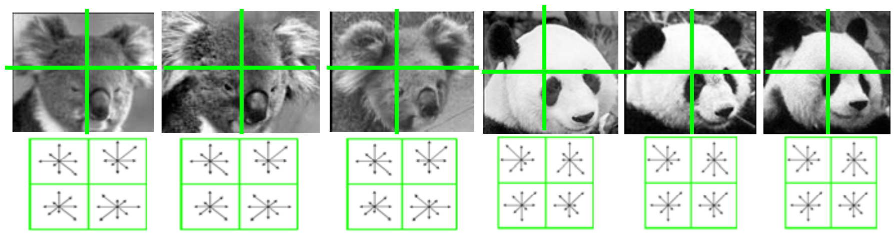
- Consider edges, contours, and (oriented) intensity gradients



Kristen Grauman

Window-based models
Building an object model

- **Consider edges, contours, and (oriented) intensity gradients**

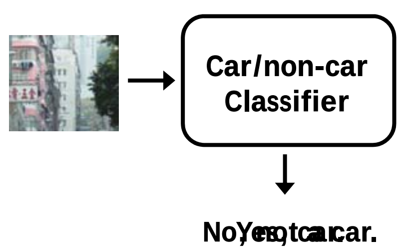


- **Summarize local distribution of gradients with histogram**
  - **Locally orderless: offers invariance to small shifts and rotations**
  - **Contrast-normalization: try to correct for variable illumination**

Kristen Grauman

Window-based models
Building an object model

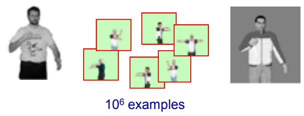
Given the representation, train a binary classifier



Kristen Grauman

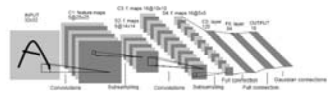
Discriminative classifier construction

Nearest neighbor



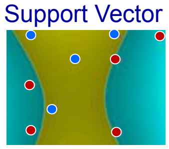
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

Neural networks



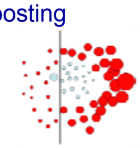
LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
…

Support Vector Machines



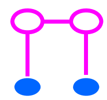
Guyon, Vapnik
Heisele, Serre, Poggio, 2001,…

Boosting



Viola, Jones 2001,
Torralba et al. 2004,
Opelt et al. 2006,…

Conditional Random Fields



McCallum, Freitag, Pereira 2000; Kumar, Hebert 2003
…

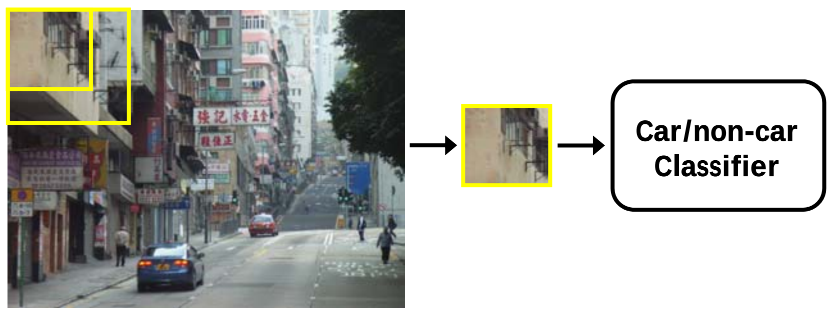
Kristen Grauman

Slide adapted from Antonio Torralba

Generic category recognition: basic framework

- Build/train object model
  - Choose a representation
  - Learn or fit parameters of model / classifier
- **Generate candidates in new image**
- **Score the candidates**

Window-based models
Generating and scoring candidates

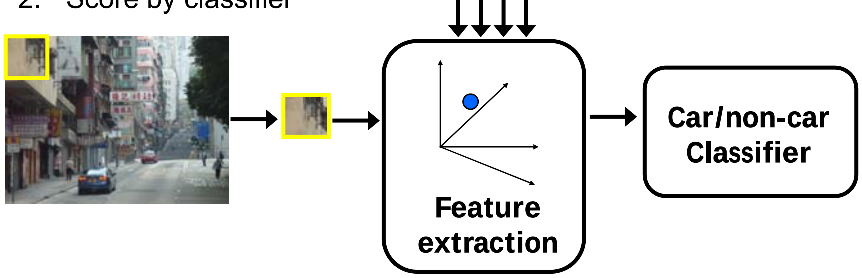


Kristen Grauman

Window-based object detection: recap

Training:
1. Obtain training data
2. Define features
3. Define classifier

Given new image:
1. Slide window
2. Score by classifier



Kristen Grauman

Issues

- What classifier?
  - Factors in choosing:
    - Generative or discriminative model?
    - Data resources – how much training data?
    - How is the labeled data prepared?
    - Training time allowance
    - Test time requirements – real-time?
    - Fit with the representation

Kristen Grauman

Issues

- What classifier?
- What features or representations?
- How to make it affordable?
- What categories are amenable?

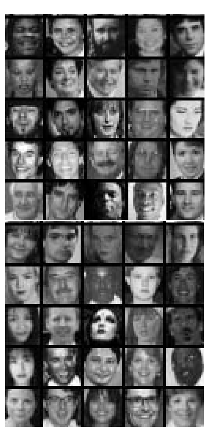
Kristen Grauman

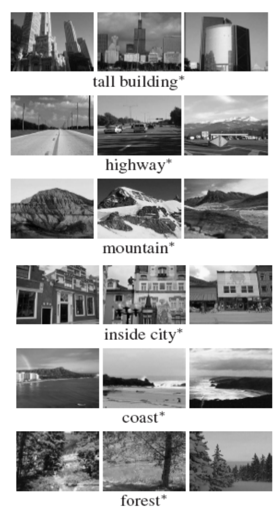
Issues

- What categories are amenable?
  - **Similar to specific object matching,** we expect spatial layout to be fairly rigidly preserved.
  - **Unlike specific object matching**, by training classifiers we attempt to capture intra-class variation or determine required discriminative features.

Kristen Grauman

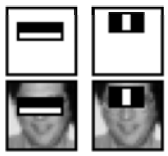
What categories are amenable to window-based reps?





Kristen Grauman

Window-based models: Three case studies

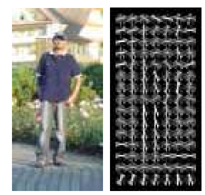


Boosting + face detection

Viola & Jones

NN + scene Gist classification

e.g., Hays & Efros



SVM + person detection

e.g., Dalal & Triggs

Viola-Jones face detector

Main idea:

– Represent local texture with efficiently computable “rectangular” features within window of interest

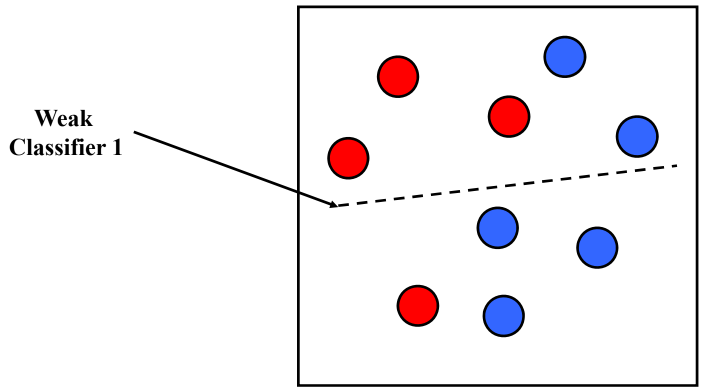
– Select discriminative features to be weak classifiers

– Use boosted combination of them as final classifier

– Form a cascade of such classifiers, rejecting clear negatives quickly

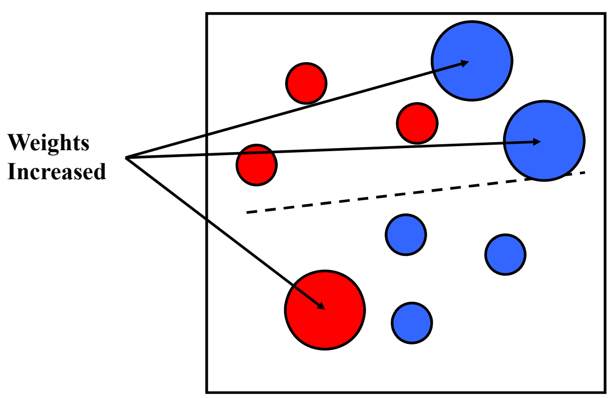
Kristen Grauman

Boosting intuition

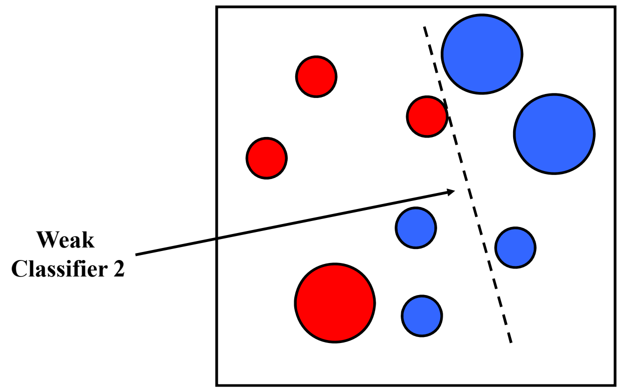


Slide credit: Paul Viola

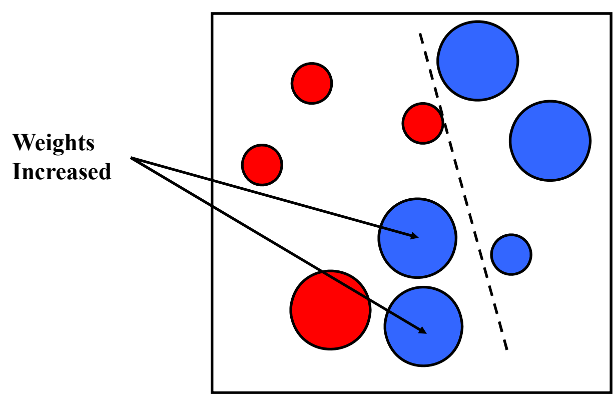
Boosting illustration



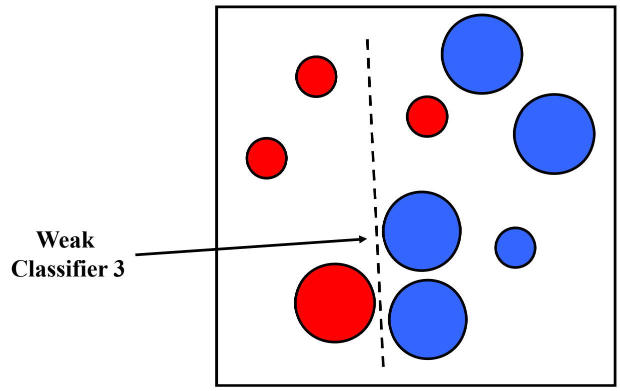
Boosting illustration



Boosting illustration

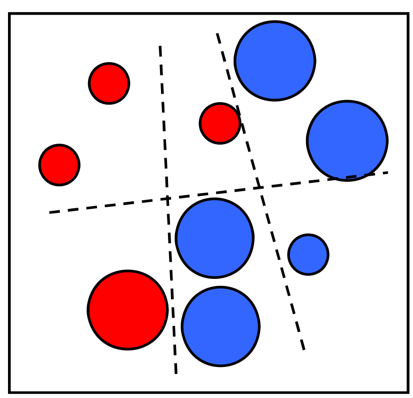


Boosting illustration



Boosting illustration

Final classifier is
a combination of weak
classifiers



Boosting: training

- Initially, weight each training example equally
- In each boosting round:
  - Find the weak learner that achieves the lowest *weighted* training error
  - Raise 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)

Slide credit: Lana Lazebnik

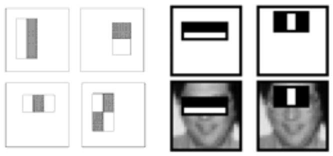
Boosting: pros and cons

- Advantages of boosting
  - Integrates classification with feature selection
  - Complexity of training is linear in the number of training examples
  - Flexibility in the choice of weak learners, boosting scheme
  - Testing is fast
  - Easy to implement

- Disadvantages
  - Needs many training examples
  - Often found not to work as well as an alternative discriminative classifier, support vector machine (SVM)
    - especially for many-class problems

Slide credit: Lana Lazebnik

Viola-Jones detector: features

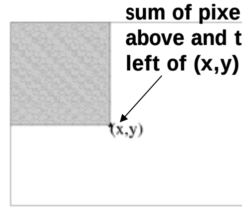


"Rectangular" filters

Feature output is difference between adjacent regions

Efficiently computable with integral image: any sum can be computed in constant time.

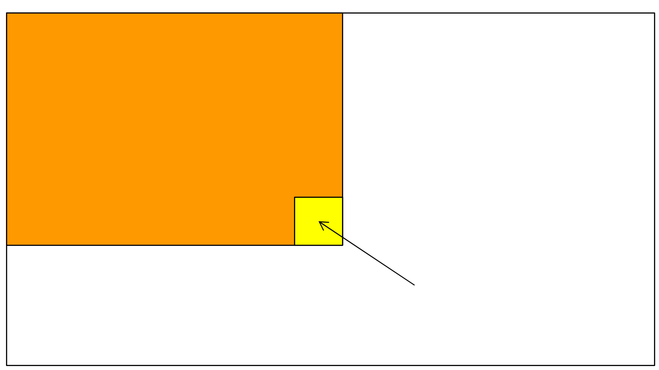
Value at (x,y) is sum of pixels above and to the left of (x,y)



Integral image

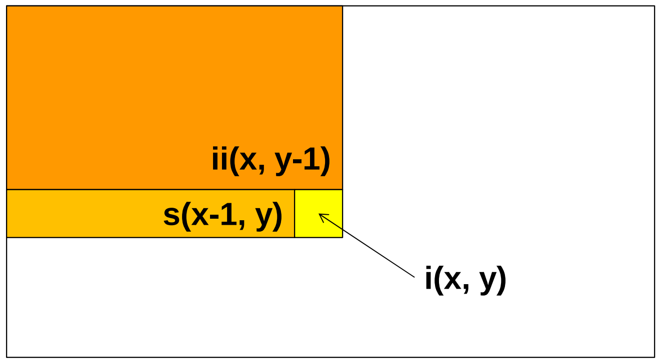
Kristen Grauman

Computing the integral image



Lana Lazebnik

Computing the integral image



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

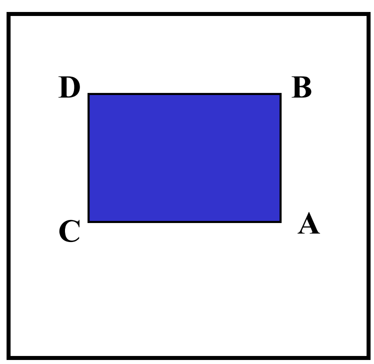
Integral image: $ii(x, y) = ii(x, y-1) + s(x, y)$

Lana Lazebnik

Computing sum within a rectangle

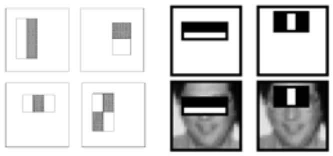
- Let A,B,C,D be the values of the integral image at the corners of a rectangle
- Then the sum of original image values within the rectangle can be computed as:

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



Lana Lazebnik

Viola-Jones detector: features

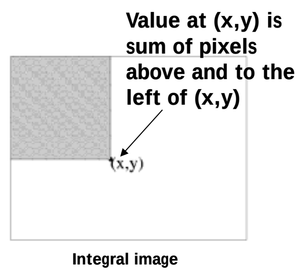


"Rectangular" filters

Feature output is difference between adjacent regions

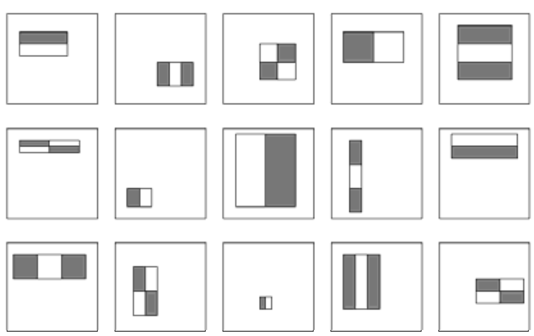
Efficiently computable with integral image: any sum can be computed in constant time

Avoid scaling images → scale features directly for same cost



Kristen Grauman

Viola-Jones detector: features



Considering all possible filter parameters: position, scale, and type:

180,000+ possible features associated with each 24 x 24 window

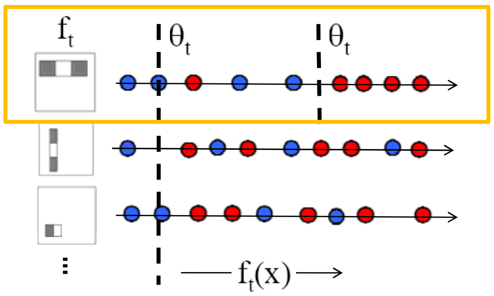
Which subset of these features should we use to determine if a window has a face?

Use AdaBoost both to select the informative features and to form the classifier

Kristen Grauman

Viola-Jones detector: AdaBoost

- **Want to select the single rectangle feature and threshold that best separates positive (faces) and negative (non-faces) training examples, in terms of *weighted* error.**



Outputs of a possible rectangle feature on faces and non-faces.

Resulting weak classifier:

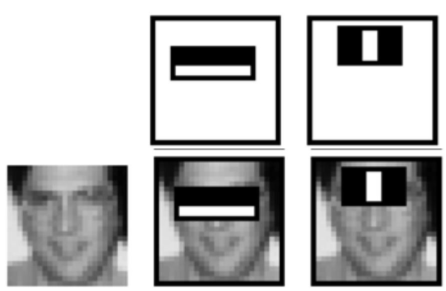
$$h_t(x) = \begin{cases} +1 & \text{if } f_t(x) > \theta_t \\ -1 & \text{otherwise} \end{cases}$$

For next round, reweight the examples according to errors, choose another filter/threshold combo.

Kristen Grauman

Visual Object Recognition Tutorial

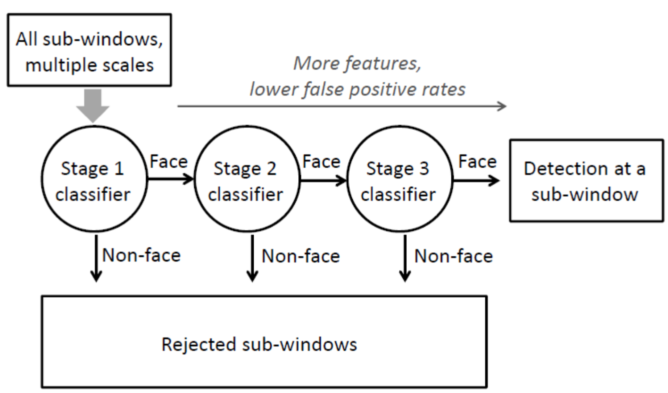
Viola-Jones Face Detector: Results



First two features selected

- Even if the filters are fast to compute, each new image has a lot of possible windows to search.
- How to make the detection more efficient?

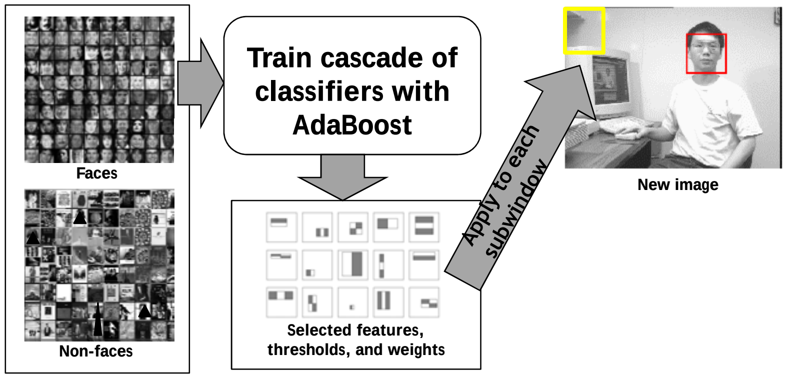
Cascading classifiers for detection



- Form a *cascade* with low false negative rates early on
- Apply less accurate but faster classifiers first to immediately discard windows that clearly appear to be negative

Kristen Grauman

Viola-Jones detector: summary



Train with 5K positives, 350M negatives
Real-time detector using 38 layer cascade
6061 features in all layers

[Implementation available in OpenCV: http://www.intel.com/technology/computing/opencv/]

Kristen Grauman

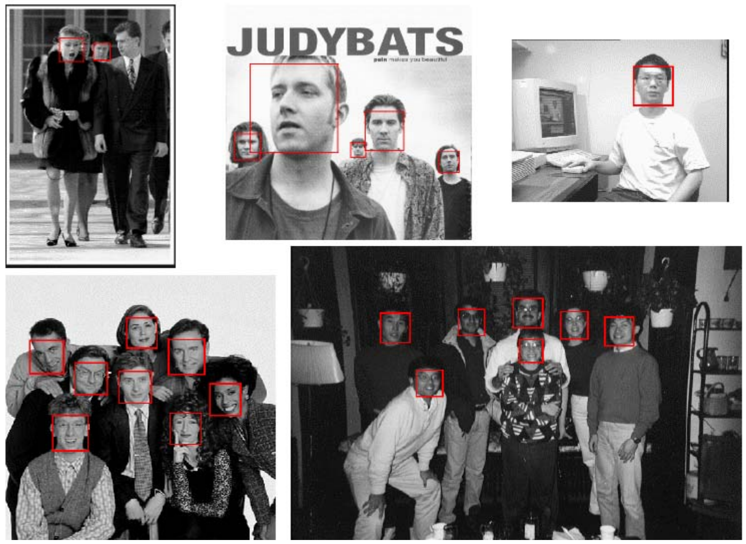
Viola-Jones detector: summary

- 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* of classifiers for fast rejection of non-face windows

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

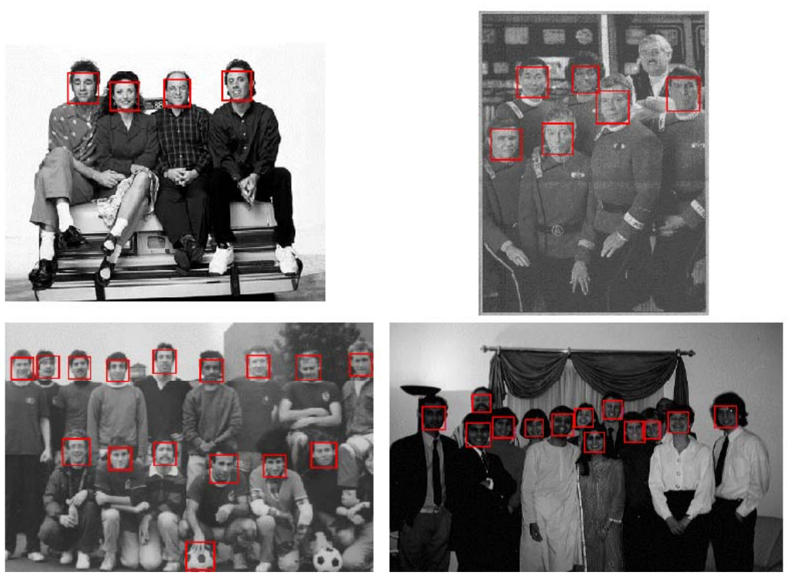
P. Viola and M. Jones. *Robust real-time face detection.* IJCV 57(2), 2004.

Viola-Jones Face Detector: Results



Visual Object Recognition Tutorial

Viola-Jones Face Detector: Results



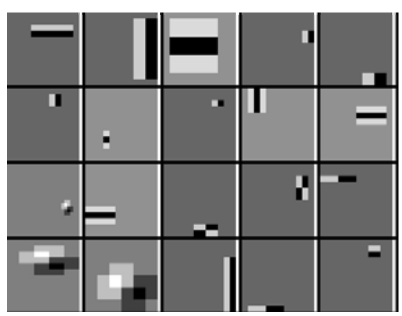
Visual Object Recognition Tutorial

Viola-Jones Face Detector: Results

Visual Object Recognition Tutorial

Detecting profile faces?

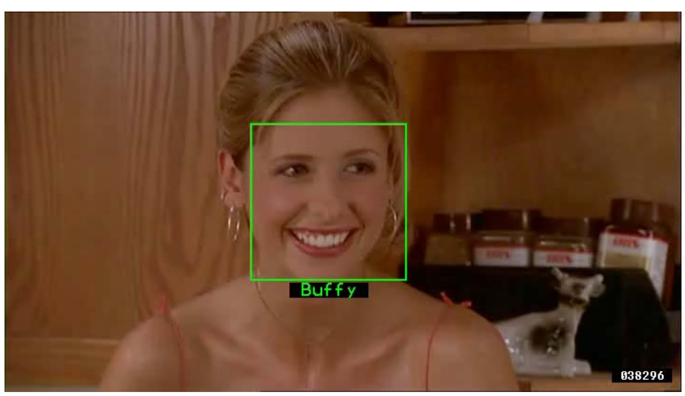
Can we use the same detector?



Visual Object Recognition Tutorial

Viola-Jones Face Detector: Results

Example using Viola-Jones detector



Frontal faces detected and then tracked, character names inferred with alignment of script and subtitles.

Everingham, M., Sivic, J. and Zisserman, A.
"Hello! My name is... Buffy" - Automatic naming of characters in TV video, BMVC 2006. http://www.robots.ox.ac.uk/~vgg/research/nface/index.html

ZDNet Asia
Where Technology Means Business

Home | News | Insight | Reviews | TechGuides | Jobs | Blogs | Videos | Community | Downloads | IT Library

Software | Hardware | Security | Communications | Business | Internet | Photos

News > Internet

Google now erases faces, license plates on Map Street View

By Elinor Mills, CNET News.com
Friday, August 24, 2007 01:37 PM

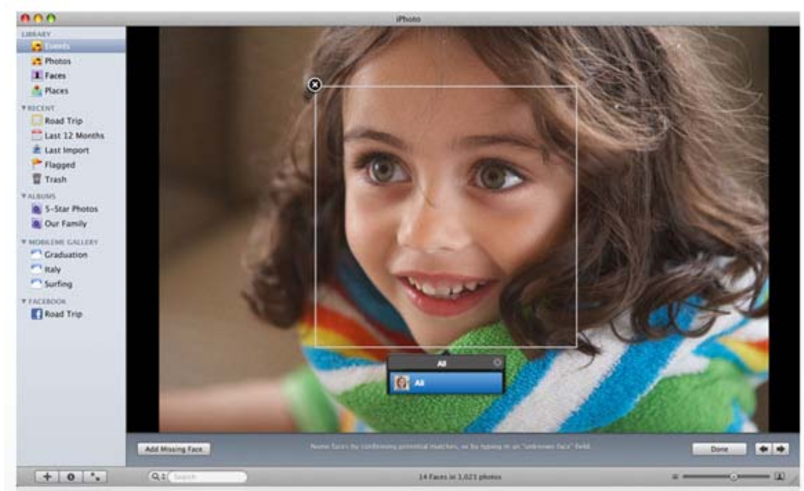
Google has gotten a lot of flack from privacy advocates for photographing faces and license plate numbers and displaying them on the Street View in Google Maps. Originally, the company said only people who identified themselves could ask the company to remove their image.

But Google has quietly changed that policy, partly in response to criticism, and now anyone can alert the company and have an image of a license plate or a recognizable face removed, not just the owner of the face or car, says Marissa Mayer, vice president of search products and user experience at Google.

"It's a good policy for users and also clarifies the intent of the product," she said in an interview following her keynote at the Search Engine Strategies conference in San Jose, Calif., Wednesday.

The policy change was made about 10 days after the launch of the product in late May, but was not publicly announced, according to Mayer. The company is removing images only when someone notifies them and not proactively, she said. "It was definitely a big policy change inside."

Consumer application: iPhoto

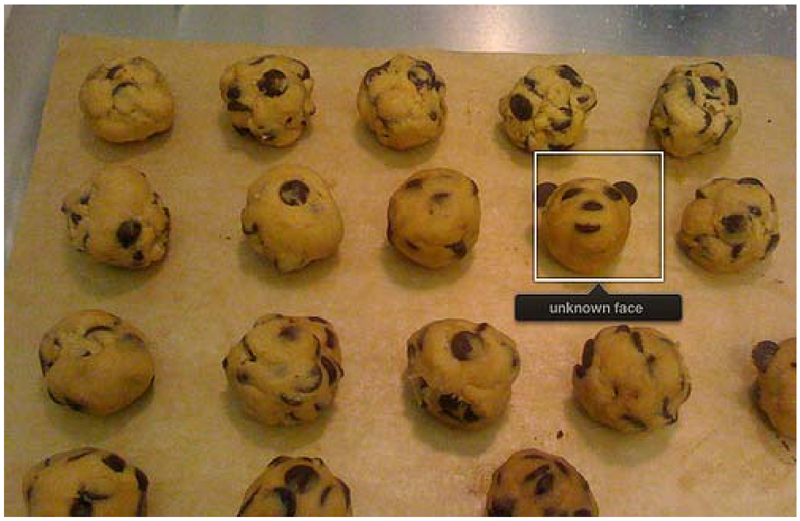


http://www.apple.com/ilife/iphoto/

Slide credit: Lana Lazebnik

Consumer application: iPhoto

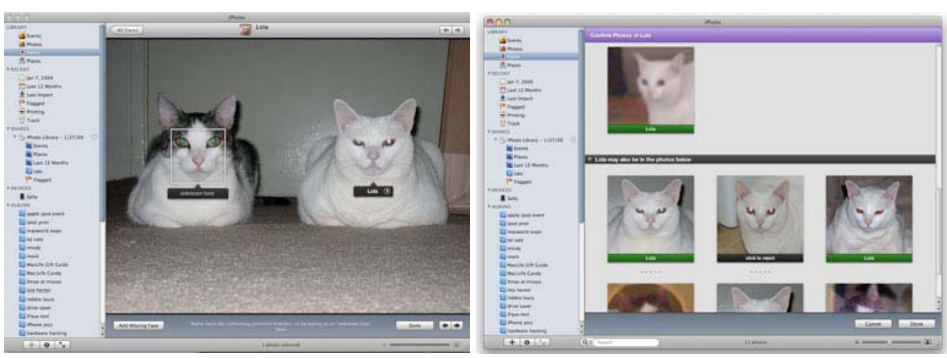
Things iPhoto thinks are faces



Slide credit: Lana Lazebnik

Consumer application: iPhoto

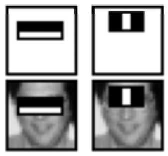
Can be trained to recognize pets!



http://www.maclife.com/article/news/iphotos_faces_recognizes_cats

Slide credit: Lana Lazebnik

Window-based models: Three case studies

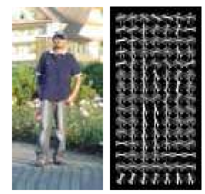


Boosting + face detection

Viola & Jones

NN + scene Gist classification

e.g., Hays & Efros



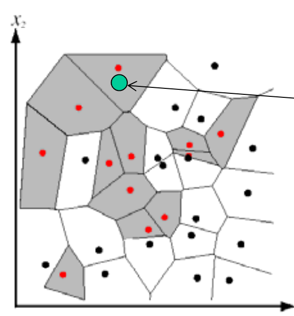
SVM + person detection

e.g., Dalal & Triggs

Nearest Neighbor classification

- Assign label of nearest training data point to each test data point

Black = negative
Red = positive



from Duda *et al.*

Novel test example

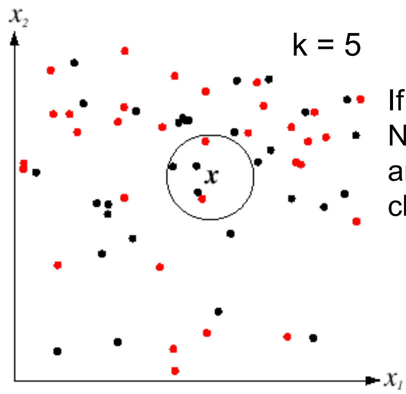
Closest to a positive example from the training set, so classify it as positive.

Voronoi partitioning of feature space for 2-category 2D data

K-Nearest Neighbors classification

- For a new point, find the k closest points from training data
- Labels of the k points “vote” to classify

Black = negative
Red = positive



If query lands here, the 5 NN consist of 3 negatives and 2 positives, so we classify it as negative.

Source: D. Lowe

A nearest neighbor recognition example

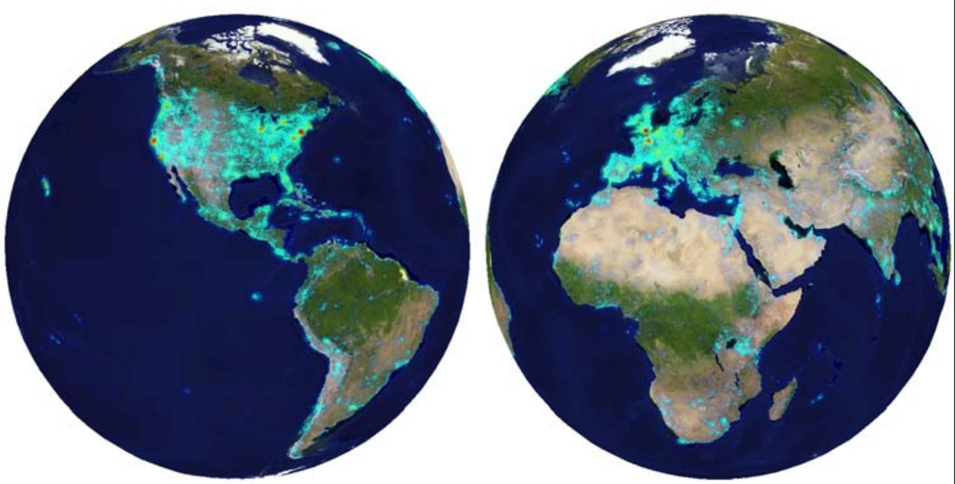
Where in the World?

[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.]

Where in the World?

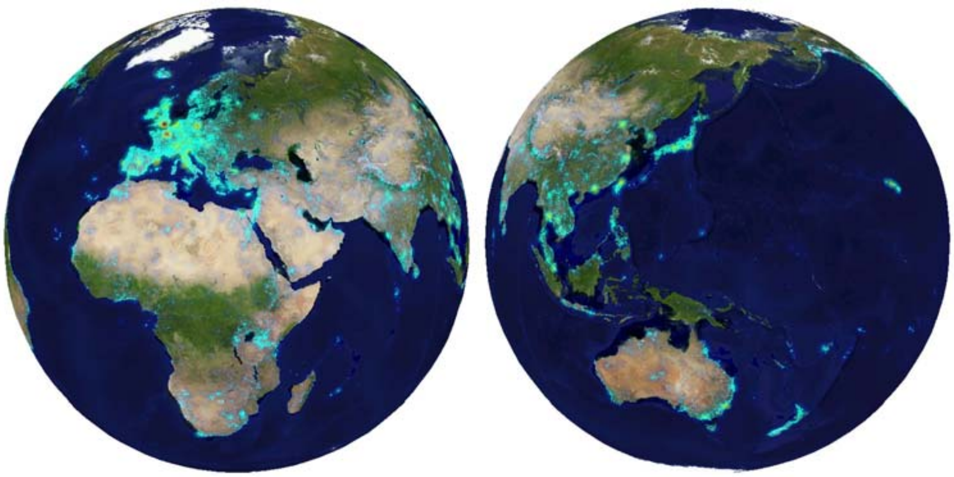
Where in the World?

6+ million geotagged photos by 109,788 photographers



Annotated by Flickr users

6+ million geotagged photos
by 109,788 photographers



Annotated by Flickr users

Which scene properties are relevant?

Spatial Envelope Theory of Scene Representation

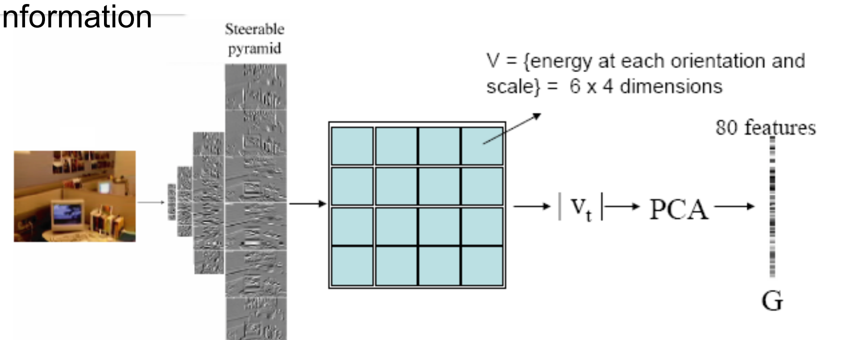
Oliva & Torralba (2001)

A scene is a single surface that can be represented by global (statistical) descriptors

Slide Credit: Aude Olivia

Global texture: capturing the "Gist" of the scene

Capture global image properties while keeping some spatial information



Gist descriptor

Oliva & Torralba IJCV 2001, Torralba et al. CVPR 2003

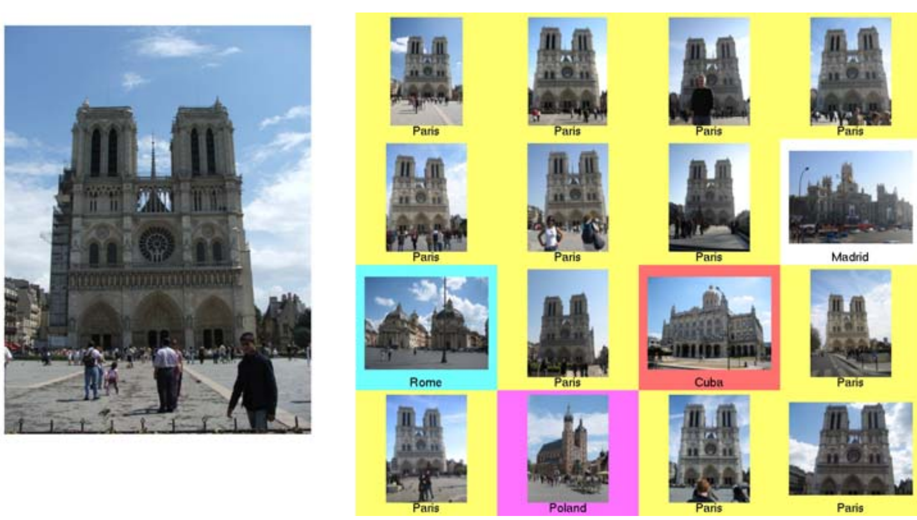
Which scene properties are relevant?

- **Gist scene descriptor**
- **Color Histograms** - L*A*B* 4x14x14 histograms
- **Texton Histograms** – 512 entry, filter bank based
- **Line Features** – Histograms of straight line stats

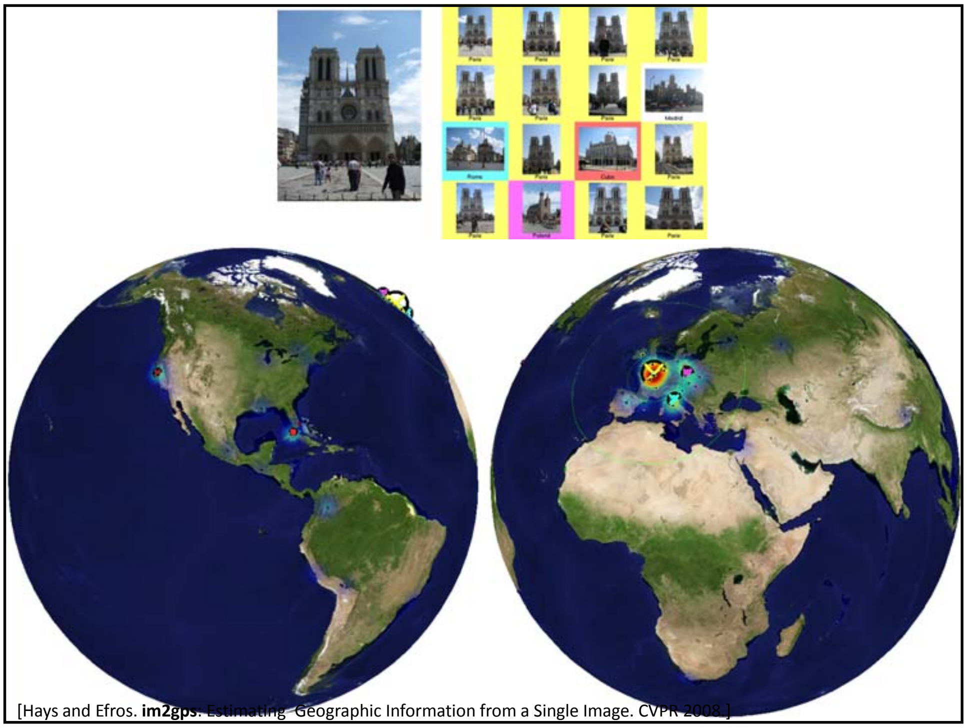
Scene Matches

[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.]

Scene Matches

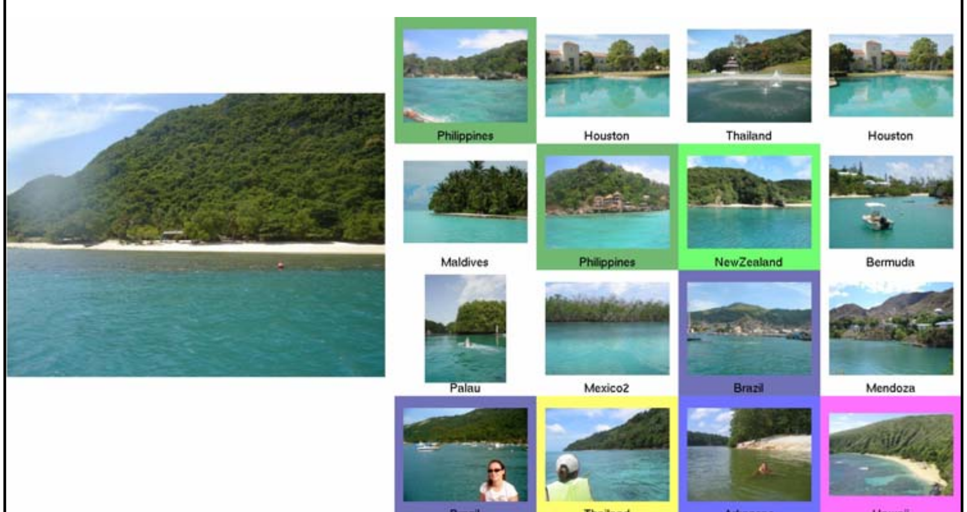


[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.]

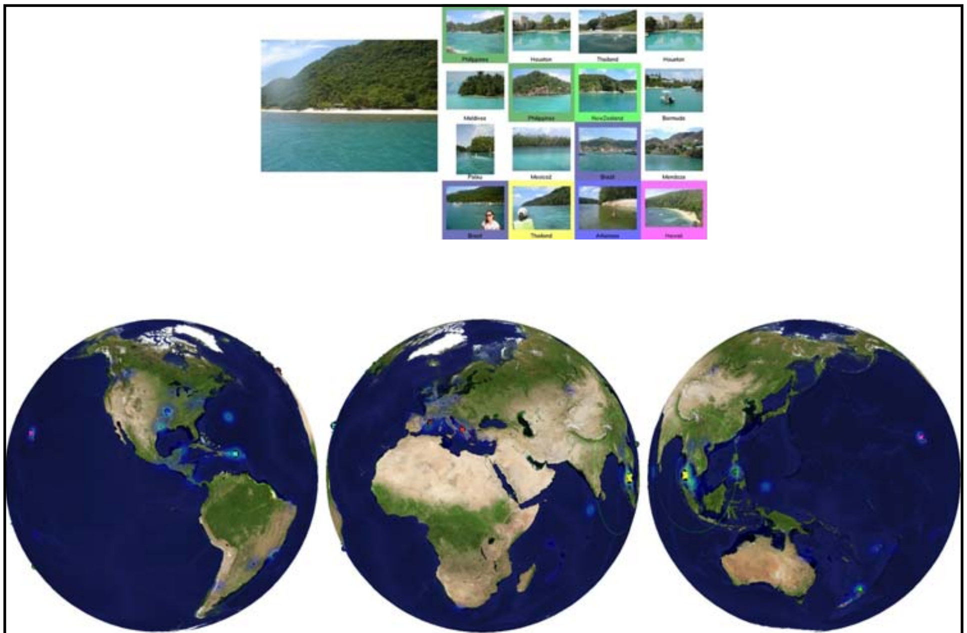


[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.]

Scene Matches

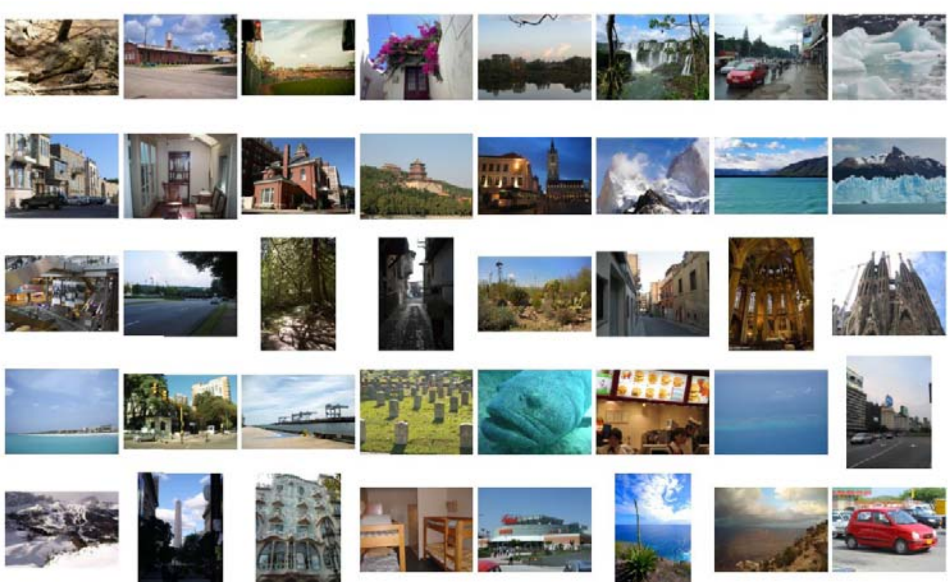


[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.]



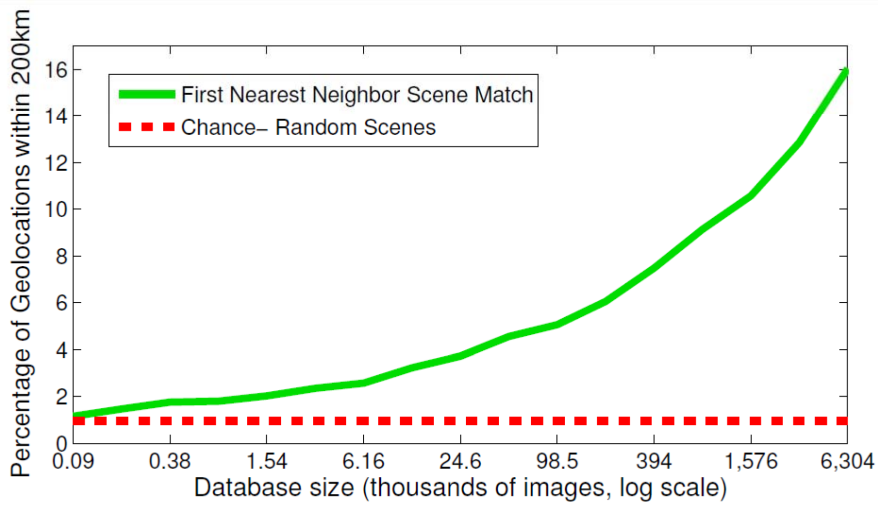
[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.]

Quantitative Evaluation Test Set



• • •

The Importance of Data



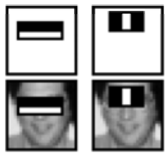
[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.]

Nearest neighbors: pros and cons

- **Pros**:
  - Simple to implement
  - Flexible to feature / distance choices
  - Naturally handles multi-class cases
  - Can do well in practice with enough representative data
- **Cons:**
  - Large search problem to find nearest neighbors
  - Storage of data
  - Must know we have a meaningful distance function

Kristen Grauman

Window-based models: Three case studies

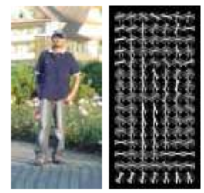


Boosting + face detection

Viola & Jones

NN + scene Gist classification

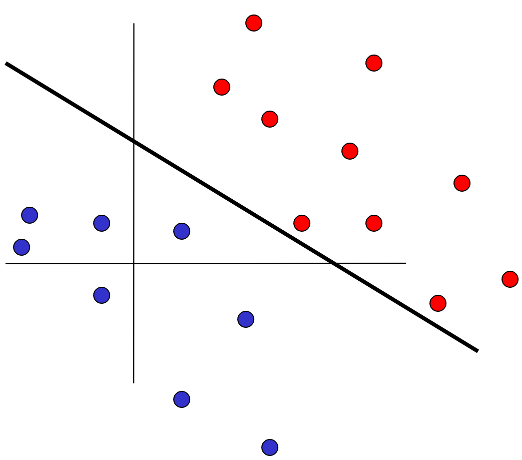
e.g., Hays & Efros



SVM + person detection

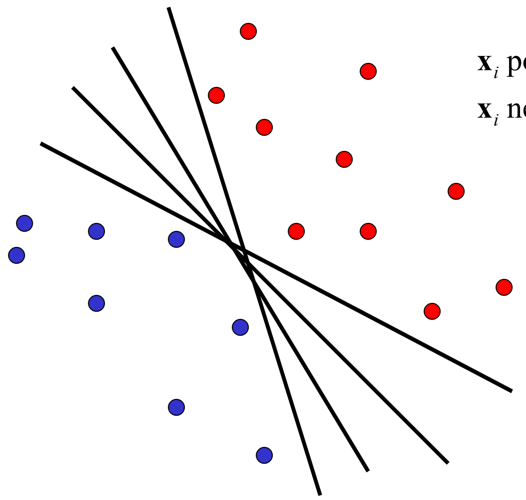
e.g., Dalal & Triggs

Linear classifiers



Linear classifiers

- Find linear function to separate positive and negative examples

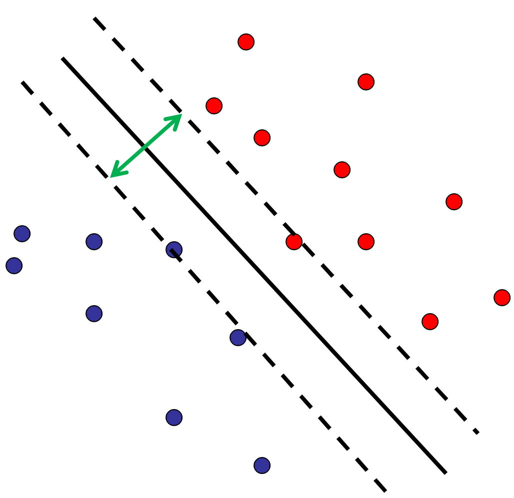


$\mathbf{x}_i$ positive : $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$

$\mathbf{x}_i$ negative : $\quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$

Which line is best?

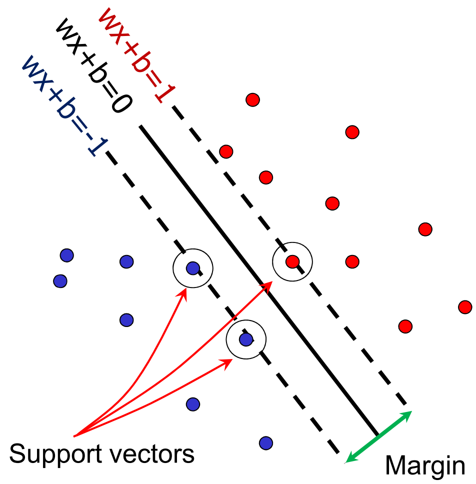
Support Vector Machines (SVMs)



- Discriminative classifier based on *optimal separating line (for 2d case)*
- Maximize the *margin* between the positive and negative training examples

Support vector machines

- Want line that maximizes the margin.



$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

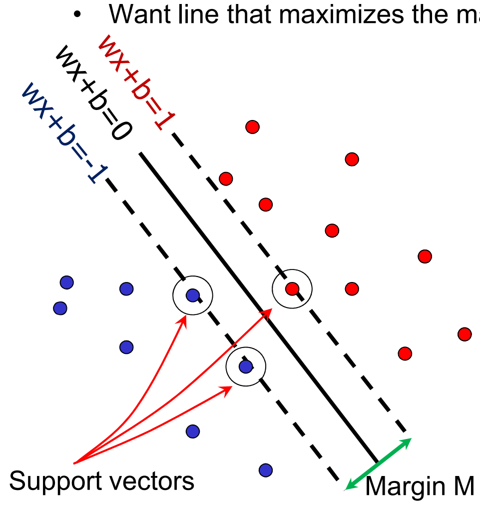
$\mathbf{x}_i$ negative $(y_i = -1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support, vectors, $\quad \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Support vector machines

- Want line that maximizes the margin.



$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ negative $(y_i = -1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support, vectors, $\quad \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

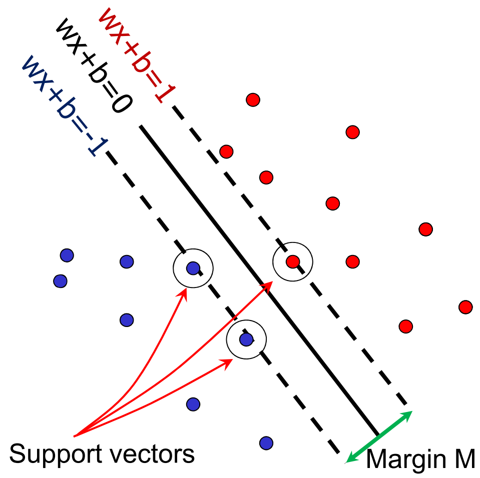
Distance between point and line: $\quad \dfrac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$

For support vectors:

$$\frac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|} = \frac{\pm 1}{\|\mathbf{w}\|} \qquad M = \left| \frac{1}{\|\mathbf{w}\|} - \frac{-1}{\|\mathbf{w}\|} \right| = \frac{2}{\|\mathbf{w}\|}$$

Support vector machines

- Want line that maximizes the margin.



$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ negative $(y_i = -1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support, vectors, $\quad \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

Distance between point and line: $\quad \dfrac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$

Therefore, the margin is $2 / \|\mathbf{w}\|$

Finding the maximum margin line

1. Maximize margin $2/\|\mathbf{w}\|$
2. Correctly classify all training data points:

$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ negative $(y_i = -1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

Quadratic optimization problem:

Minimize $\dfrac{1}{2}\mathbf{w}^T\mathbf{w}$

Subject to $y_i(\mathbf{w} \cdot \boldsymbol{x}_i + b) \geq 1$

Finding the maximum margin line

- Solution: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

learned weight

Support vector

Finding the maximum margin line

- Solution: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

  $b = y_i - \mathbf{w} \cdot \mathbf{x}_i$ (for any support vector)

  $$\mathbf{w} \cdot \mathbf{x} + b = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b$$

- Classification function:

$$f(x) = \text{sign}\,(\mathbf{w} \cdot \mathbf{x} + \text{b})$$

$$= \text{sign}\left(\sum_{\text{i}} \alpha_{\text{i}} \boxed{\mathbf{x}_i \cdot \mathbf{x}} + b\right)$$

If f(x) < 0, classify as negative, if f(x) > 0, classify as positive

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery,

Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alps, 655 avenue de l'Europe, Montbonnot 38334, France
{Navneet.Dalal,Bill.Triggs}@inrialpes.fr, http://lear.inrialpes.fr

Abstract

We study the question of feature sets for robust visual object recognition, adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, we show experimentally that grids of Histograms of Oriented Gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results. The new approach gives near-perfect separation on the original MIT pedestrian database, so we introduce a more challenging dataset containing over 1800 annotated human images with a large range of pose variations and backgrounds.

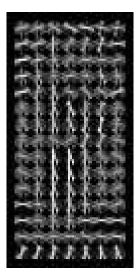
1 Introduction

We briefly discuss previous work on human detection in §2, give an overview of our method §3, describe our data sets in §4 and give a detailed description and experimental evaluation of each stage of the process in §5–6. The main conclusions are summarized in §7.

2 Previous Work

There is an extensive literature on object detection, but here we mention just a few relevant papers on human detection [18,17,22,16,20]. See [6] for a survey. Papageorgiou *et al* [18] describe a pedestrian detector based on a polynomial SVM using rectified Haar wavelets as input descriptors, with a parts (subwindow) based variant in [17]. Depoortere *et al* give an optimized version of this [2]. Gavrila & Philomen [8] take a more direct approach, extracting edge images and matching them to a set of learned exemplars using chamfer distance. This has been used in a practical real-time pedestrian detection system [7]. Viola *et al* [22] build an efficient

Person detection with HoG's & linear SVM's

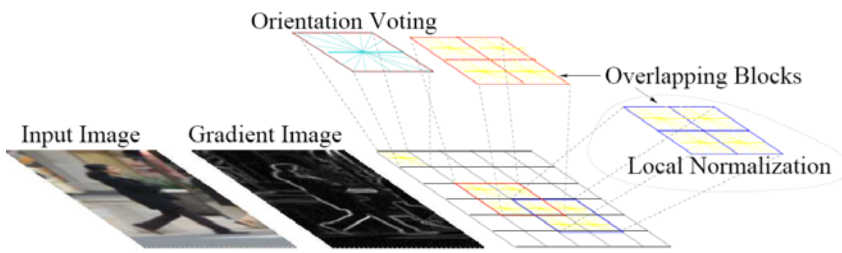


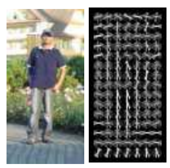
- Map each grid cell in the input window to a histogram counting the gradients per orientation.
- Train a linear SVM using training set of pedestrian vs. non-pedestrian windows.

Dalal & Triggs, CVPR 2005

Code available:
http://pascal.inrialpes.fr/soft/olt/

HoG descriptor





Dalal & Triggs, CVPR 2005

Code available: http://pascal.inrialpes.fr/soft/olt/

Person detection with HoGs & linear SVMs

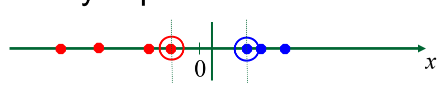
- Histograms of Oriented Gradients for Human Detection, Navneet Dalal, Bill Triggs, International Conference on Computer Vision & Pattern Recognition - June 2005
- http://lear.inrialpes.fr/pubs/2005/DT05/

Questions

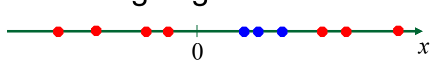
- What if the data is not linearly separable?
- What if we have more than just two categories?

Non-linear SVMs

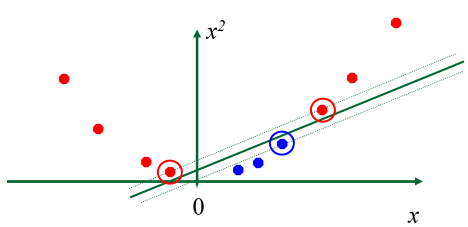
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?

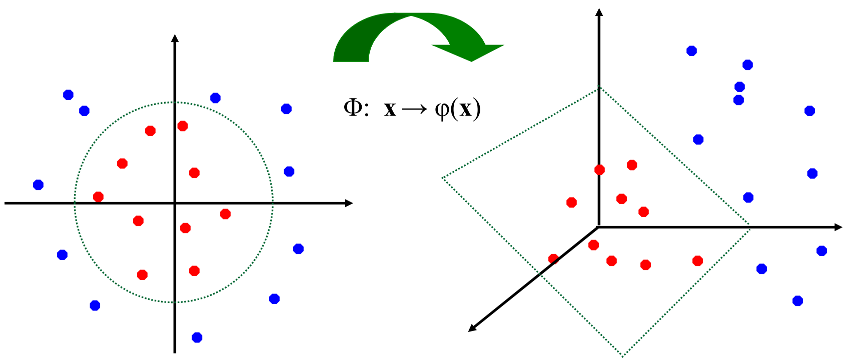


- How about… mapping data to a higher-dimensional space:



Non-linear SVMs: feature spaces

- General idea: the original input space can be mapped to some higher-dimensional feature space where the training set is separable:



Slide from Andrew Moore's tutorial: http://www.autonlab.org/tutorials/svm.html

The "Kernel Trick"

- The linear classifier relies on dot product between vectors $K(x_i,x_j)=x_i^T x_j$
- If every data point is mapped into high-dimensional space via some transformation Φ: $x \rightarrow \varphi(x)$, the dot product becomes:

$$K(x_i,x_j)= \varphi(x_i)^T\varphi(x_j)$$

- A *kernel function* is similarity function that corresponds to an inner product in some expanded feature space.

Slide from Andrew Moore's tutorial: http://www.autonlab.org/tutorials/svm.html

Example

2-dimensional vectors x=$[x_1 \;\; x_2]$;

let $K(\mathrm{x_i},\mathrm{x_j})=(1 + \mathrm{x_i}^{\mathrm{T}}\mathrm{x_j})^2$

Need to show that $K(\mathrm{x_i},\mathrm{x_j}) = \varphi(\mathrm{x_i})^{\mathrm{T}}\varphi(\mathrm{x_j})$:

$$K(\mathrm{x_i},\mathrm{x_j})=(1 + \mathrm{x_i}^{\mathrm{T}}\mathrm{x_j})^2,$$

$$= 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2} + x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

$$= [1 \;\; x_{i1}^2 \;\; \sqrt{2}\, x_{i1}x_{i2} \;\; x_{i2}^2 \;\; \sqrt{2}x_{i1} \;\; \sqrt{2}x_{i2}]^{\mathrm{T}}$$

$$[1 \;\; x_{j1}^2 \;\; \sqrt{2}\, x_{j1}x_{j2} \;\; x_{j2}^2 \;\; \sqrt{2}x_{j1} \;\; \sqrt{2}x_{j2}]$$

$$= \varphi(\mathrm{x_i})^{\mathrm{T}}\varphi(\mathrm{x_j}),$$

where $\varphi(\mathrm{x}) = [1 \;\; x_1^2 \;\; \sqrt{2}\, x_1x_2 \;\; x_2^2 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2]$

Nonlinear SVMs

- *The kernel trick*: instead of explicitly computing the lifting transformation $\varphi(\mathbf{x})$, define a kernel function K such that

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$$

- This gives a nonlinear decision boundary in the original feature space:

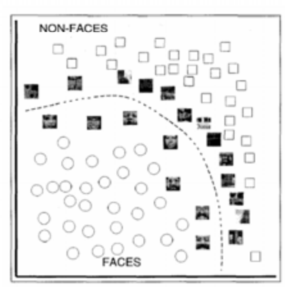
$$\sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

Examples of kernel functions

- Linear: $$K(x_i, x_j) = x_i^T x_j$$
- Gaussian RBF: $$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$
- Histogram intersection: $$K(x_i, x_j) = \sum_k \min(x_i(k), x_j(k))$$

SVMs for recognition

1. Define your representation for each example.
2. Select a kernel function.
3. Compute pairwise kernel values between labeled examples
4. Use this “kernel matrix” to solve for SVM support vectors & weights.
5. To classify a new example: compute kernel values between new input and support vectors, apply weights, check sign of output.



Kristen Grauman

Questions

- What if the data is not linearly separable?
- **What if we have more than just two categories?**

Multi-class SVMs

- Achieve multi-class classifier by combining a number of binary classifiers
- **<u>One vs. all</u>**
  - Training: learn an SVM for each class vs. the rest
  - Testing: apply each SVM to test example and assign to it the class of the SVM that returns the highest decision value
- **<u>One vs. one</u>**
  - Training: learn an SVM for each pair of classes
  - Testing: each learned SVM "votes" for a class to assign to the test example

Kristen Grauman

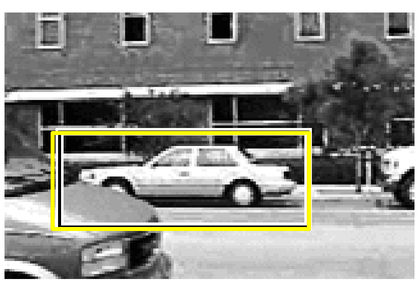
SVMs: Pros and cons

- Pros
  - Kernel-based framework is very powerful, flexible
  - Often a sparse set of support vectors – compact at test time
  - Work very well in practice, even with very small training sample sizes

- Cons
  - No "direct" multi-class SVM, must combine two-class SVMs
  - Can be tricky to select best kernel function for a problem
  - Computation, memory
    - During training time, must compute matrix of kernel values for every pair of examples
    - Learning can take a very long time for large-scale problems

Adapted from Lana Lazebnik

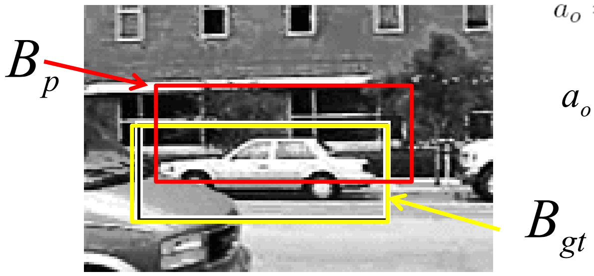
Scoring a sliding window detector



If prediction and ground truth are *bounding boxes*, when do we have a correct detection?

Kristen Grauman

Scoring a sliding window detector



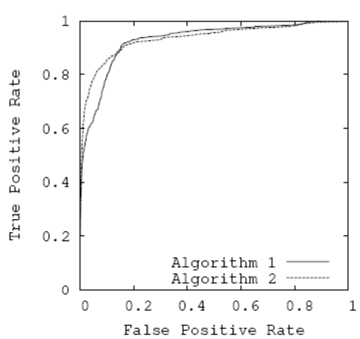
$$a_o = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})}$$

$$a_o > 0.5 \Rightarrow correct$$

We'll say the detection is correct (a "true positive") if the intersection of the bounding boxes, divided by their union, is > 50%.

Kristen Grauman

Scoring an object detector



If the detector can produce a *confidence score* on the detections, then we can plot the rate of true vs. false positives as a threshold on the confidence is varied.

TPR= fraction of positive examples that are correctly labeled.

FPR=fraction of negative examples that are misclassified as positive.

Kristen Grauman

Window-based detection: strengths

- Sliding window detection and global appearance descriptors:
  - Simple detection protocol to implement
  - Good feature choices critical
  - Past successes for certain classes

Visual Object Recognition Tutorial

Kristen Grauman

Window-based detection: Limitations

- High computational complexity
  - For example: 250,000 locations x 30 orientations x 4 scales = 30,000,000 evaluations!
  - If training binary detectors independently, means cost increases linearly with number of classes
- With so many windows, false positive rate better be low

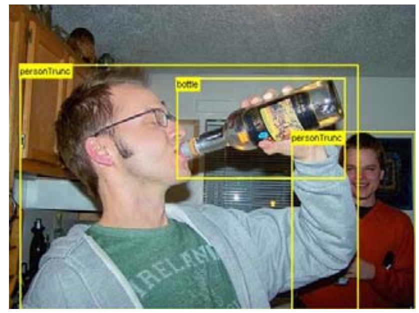
Visual Object Recognition Tutorial

Kristen Grauman

Visual Object Recognition Tutorial

Limitations (continued)

- Not all objects are "box" shaped

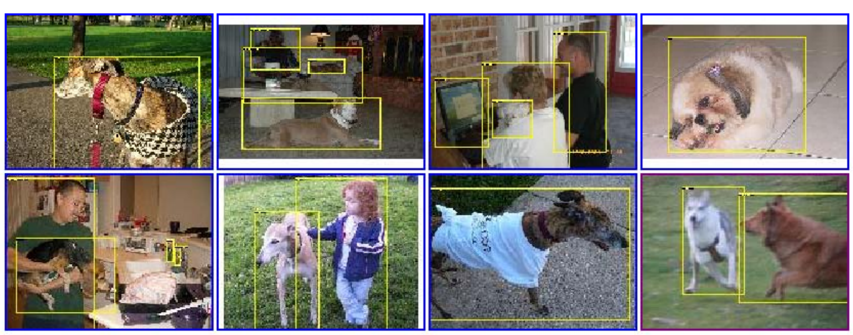


Kristen Grauman

Visual Object Recognition Tutorial

Limitations (continued)

- Non-rigid, deformable objects not captured well with representations assuming a fixed 2d structure; or must assume fixed viewpoint
- Objects with less-regular textures not captured well with holistic appearance-based descriptions



Kristen Grauman

Visual Object Recognition Tutorial

Limitations (continued)

- If considering windows in isolation, context is lost

Sliding window

Detector's view

Figure credit: Derek Hoiem

Kristen Grauman

Visual Object Recognition Tutorial

Limitations (continued)

- In practice, often entails large, cropped training set (expensive)
- Requiring good match to a global appearance description can lead to sensitivity to partial occlusions

Image credit: Adam, Rivlin, & Shimshoni

Kristen Grauman

Summary

- Basic pipeline for window-based detection
  - Model/representation/classifier choice
  - Sliding window and classifier scoring
- Discriminative classifiers for window-based representations
  - Boosting
    - Viola-Jones face detector example
  - Nearest neighbors
    - Scene recognition example
  - Support vector machines
    - HOG person detection example
- Pros and cons of window-based detection