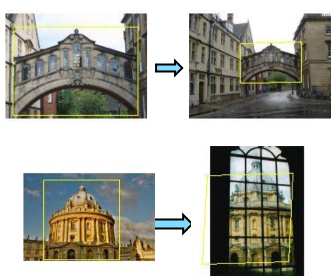
Recognizing object categories

Kristen Grauman
UT-Austin

Announcements

- Reminder: Assignment 1 due Feb 19 on Canvas
- Reminder: Optional CNN/Caffe tutorial on Monday Feb 15, 5-7 pm
- Presentations:
  - Choose paper, coordinate
  - Experiment and paper can overlap
  - Be very mindful of time limit

Last time: Recognizing instances



Last time: Recognizing instances

- 1. Basics in feature extraction: filtering
- 2. Invariant local features
- 3. Recognizing object instances

Recognition via feature matching+spatial verification

Pros:
- Effective when we are able to find reliable features within clutter
- Great results for matching specific instances

Cons:
- Scaling with number of models
- Spatial verification as post-processing – not seamless, expensive for large-scale problems
- Not suited for category recognition.

Kristen Grauman

Today

- Intro to categorization problem
- Object categorization as discriminative classification
  - Boosting + fast face detection example
  - Nearest neighbors + scene recognition example
  - Support vector machines + pedestrian detection example
    - Pyramid match kernels, spatial pyramid match
  - Convolutional neural networks + ImageNet example
- Some new representations along the way
  - Rectangular filters
  - GIST
  - HOG

What does recognition involve?

Fei-Fei Li

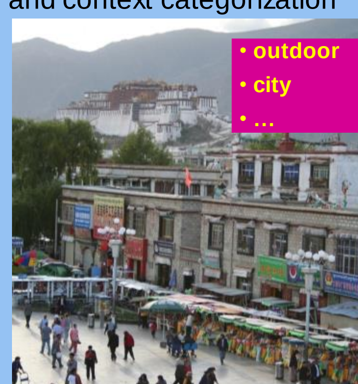
Detection: are there people?

Activity: What are they doing?

Object categorization

Instance recognition

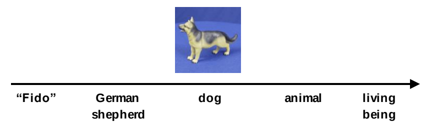
Scene and context categorization



Attribute recognition

Object Categorization

- Task Description
  - "Given a small number of training images of a category, recognize a-priori unknown instances of that category and assign the correct category label."
- Which categories are feasible visually?



Visual Object Recognition Tutorial

K. Grauman, B. Leibe

Visual Object Categories

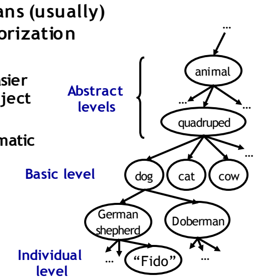
- Basic Level Categories in human categorization [Rosch 76, Lakoff 87]
  - The highest level at which category members have similar perceived shape
  - The highest level at which a single mental image reflects the entire category
  - The level at which human subjects are usually fastest at identifying category members
  - The first level named and understood by children
  - The highest level at which a person uses similar motor actions for interaction with category members

Visual Object Recognition Tutorial

K. Grauman, B. Leibe

Visual Object Categories

- Basic-level categories in humans seem to be defined predominantly visually.
- There is evidence that humans (usually) start with basic-level categorization *before* doing identification.
  - ⇒ Basic-level categorization is easier and faster for humans than object identification!
  - ⇒ How does this transfer to automatic classification algorithms?



Visual Object Recognition Tutorial

K. Grauman, B. Leibe

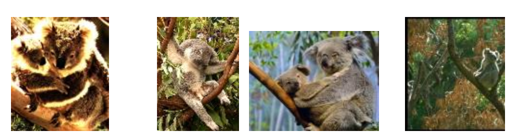
Other Types of Categories

- Functional Categories
  - e.g. chairs = *"something you can sit on"*

Visual Object Recognition Tutorial

K. Grauman, B. Leibe

Challenges: robustness



Illumination | Object pose | Clutter

Occlusions | Intra-class appearance | Viewpoint

Challenges: context and human experience

Context cues

Function

Dynamics

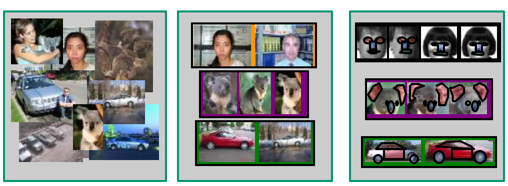
Video credit: J. Davis

Challenges: complexity

- Millions of pixels in an image
- 30,000 human recognizable object categories
- 30+ degrees of freedom in the pose of articulated objects (humans)
- Billions of images online
- 144K hours of new video on YouTube daily
- …
- About half of the cerebral cortex in primates is devoted to processing visual information [Felleman and van Essen 1991]

Challenges: learning with minimal supervision

Less — More



Unlabeled, multiple objects

Classes labeled, some clutter

Cropped to object, parts and classes labeled

Evolution of methods

- Hand-crafted models
- 3D geometry
- Hypothesize and align

- Hand-crafted features
- Learned models
- Data-driven

- "End-to-end" learning of features and models*,**

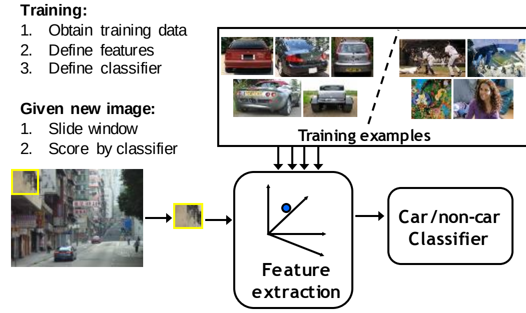
Generic category recognition: basic framework

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

Window-based 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

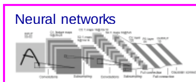
Discriminative classifier construction

Nearest neighbor

10^6 examples

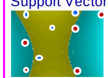
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

Neural networks



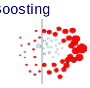
LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
…

Support Vector Machines



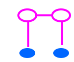
Guyon, Vapnik
Heisele, Serre, Poggio, 2001,…

Boosting



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

Conditional Random Fields



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

Kristen Grauman

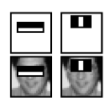
Slide adapted from Antonio Torralba

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.

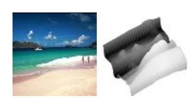
Kristen Grauman

Window-based models: Three landmark case studies



Boosting + face detection

Viola & Jones



NN + scene Gist classification

e.g., Hays & Efros

SVM + person detection

e.g., Dalal & Triggs

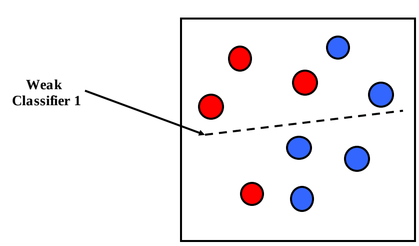
Viola-Jones face detector

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

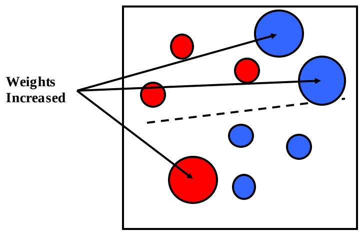
Kristen Grauman

Boosting intuition

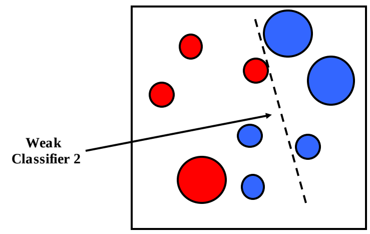


Slide credit: Paul Viola

Boosting illustration



Boosting illustration



Boosting: 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)

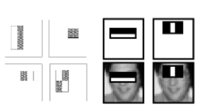
Slide credit: Lana Lazebnik

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

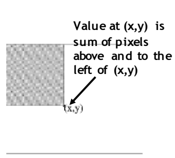
Viola-Jones detector: features



"Rectangular" filters

Feature output is difference between adjacent regions

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

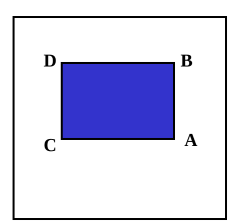


Kristen Grauman

Computing sum within a rectangle

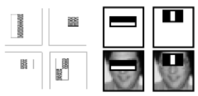
- Let A,B,C,D be the values of the integral image at the corners of a rectangle
- Then the sum of original image values within the rectangle can be computed as:

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



Lana Lazebnik

Viola-Jones detector: features

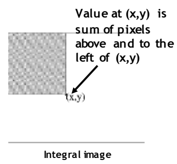


"Rectangular" filters

Feature output is difference between adjacent regions

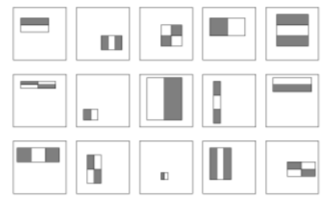
Efficiently computable with integral image: any sum can be computed in constant time

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

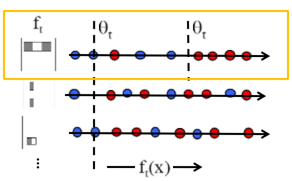
Which subset of these features should we use to determine if a window has a face?

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:

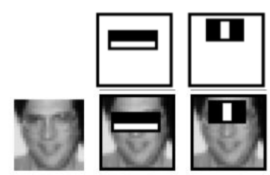
$$h_t(x) = \begin{cases} +1 & \text{if } f_t(x) > \theta_t \\ -1 & \text{otherwise} \end{cases}$$

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

Kristen Grauman

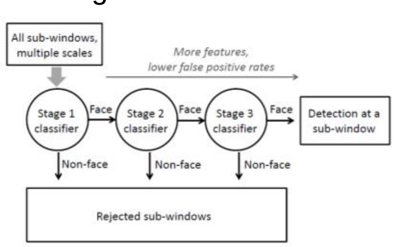
Viola-Jones Face Detector: Results

Visual Object Recognition Tutorial



First two features selected

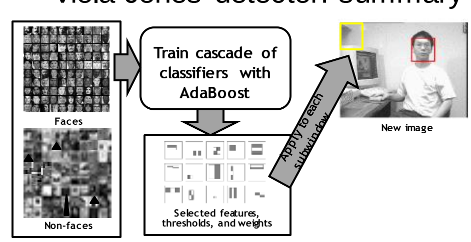
Cascading classifiers for detection



- 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

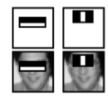
Viola-Jones detector: summary

- 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.

Window-based models: Three landmark case studies

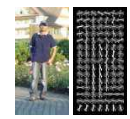


Boosting + face detection

Viola & Jones

NN + scene Gist classification

e.g., Hays & Efros



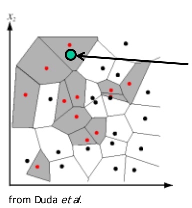
SVM + person detection

e.g., Dalal & Triggs

Nearest Neighbor classification

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

Black = negative
Red = positive



Novel test example

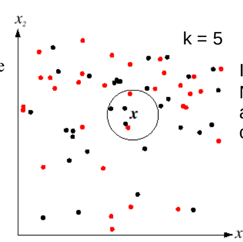
Closest to a positive example from the training set, so classify it as positive.

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

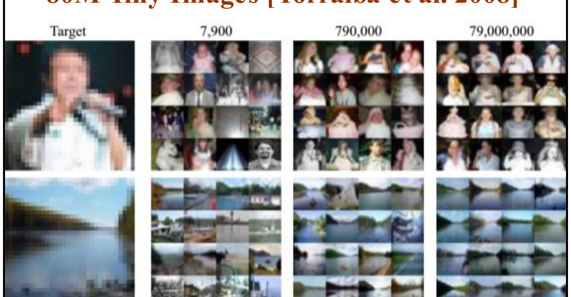
Black = negative
Red = positive



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

Source: D. Lowe

80M Tiny Images [Torralba et al. 2008]

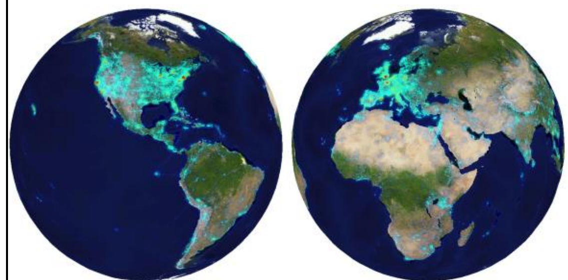


Another nearest neighbor recognition example

Where in the World?

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

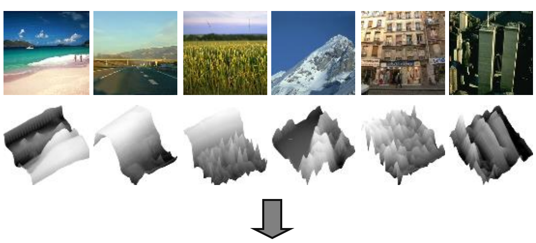
6+ million geotagged photos by 109,788 photographers



Annotated by Flickr users

Spatial Envelope Theory of Scene Representation

Oliva & Torralba (2001)

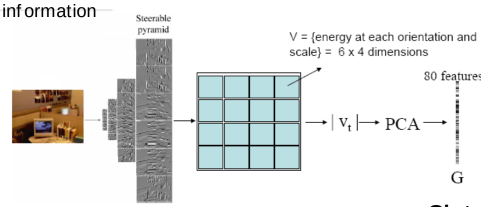


A scene is a single surface that can be represented by global (statistical) descriptors

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

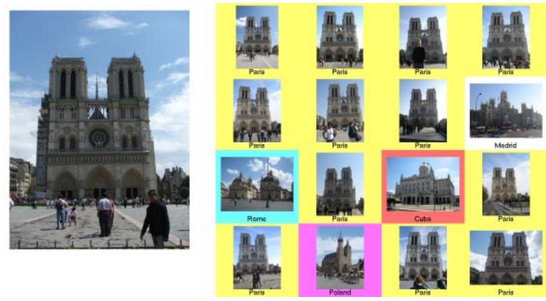
Which scene properties are relevant?

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

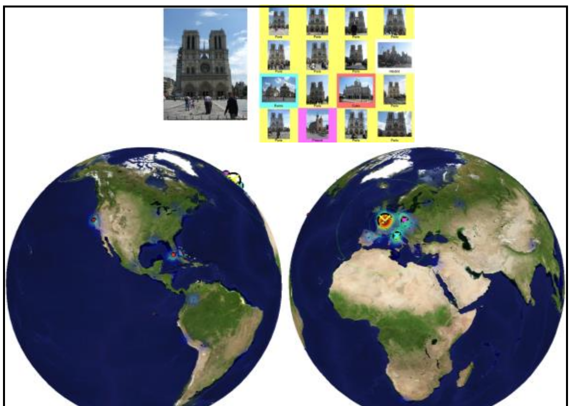
Scene Matches

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

Scene Matches

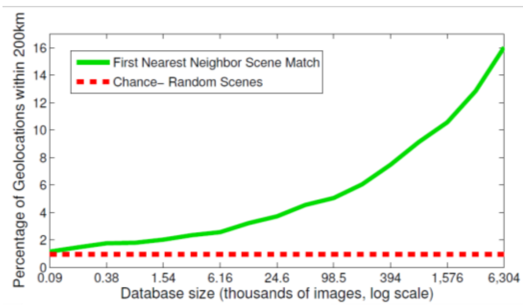


[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.]



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

The Importance of Data



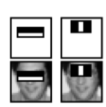
[Hays and Efros. **im2gps**: Estimating Geographic Information from a Single Image. CVPR 2008.]

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

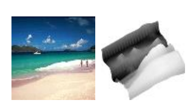
Kristen Grauman

Window-based models: Three landmark case studies



Boosting + face detection

Viola & Jones



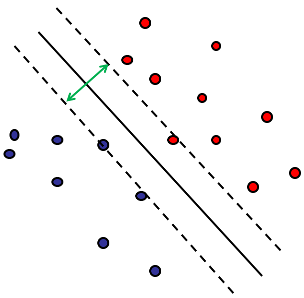
NN + scene Gist classification

e.g., Hays & Efros

SVM + person detection

e.g., Dalal & Triggs

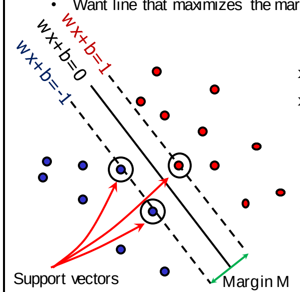
Support Vector Machines (SVMs)



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

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

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

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

For support vectors:

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

Finding the maximum margin line

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

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

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

Quadratic optimization problem:

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

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

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)

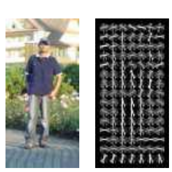
$$\mathbf{w}\cdot\mathbf{x} + b = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b$$

- Classification function:

$$f(x) = \text{sign}\,(\mathbf{w}\cdot\mathbf{x} + \text{b})$$
$$= \text{sign}\left(\sum_i \alpha_i y_i \boxed{\mathbf{x}_i \cdot \mathbf{x}} + b\right)$$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1

Person detection with HoG's & linear SVM's

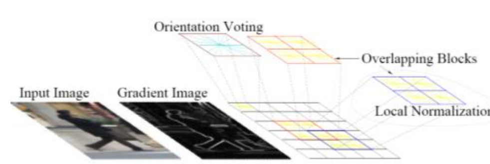


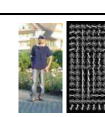
- Map each grid cell in the input window to a histogram counting the gradients per orientation.
- Train a linear SVM using training set of pedestrian vs. non-pedestrian windows.

Dalal & Triggs, CVPR 2005

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

HoG descriptor





Dalal & Triggs, CVPR 2005

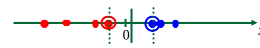
Code available: http://pascal.inrialpes.fr/soft/olt/

Person detection with HoGs & linear SVMs

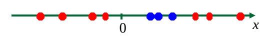
- Histograms of Oriented Gradients for Human Detection, Navneet Dalal, Bill Triggs, International Conference on Computer Vision & Pattern Recognition - June 2005
- http://lear.inrialpes.fr/pubs/2005/DT05/

Non-linear SVMs

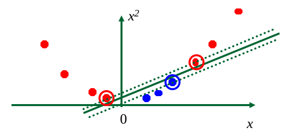
- Datasets that are linearly separable with some noise work out great:



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



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



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:

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

Example

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

let $K(x_i,x_j)=(1 + x_i^T x_j)^2$

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

$K(x_i,x_j)=(1 + x_i^T x_j)^2$,

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

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

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

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

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

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:

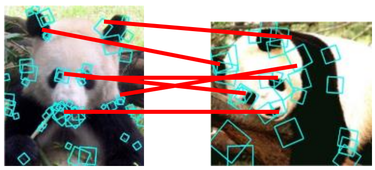
$$K(x_i,x_j) = \sum_k \min(x_i(k), x_j(k))$$

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

What about a *matching* kernel?

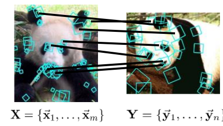


$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

Local feature correspondence useful similarity measure for generic object categories

Kristen Grauman

Partially matching sets of features



Optimal match: O(m^3)
Greedy match: O(m^2 log m)
Pyramid match: O(m)

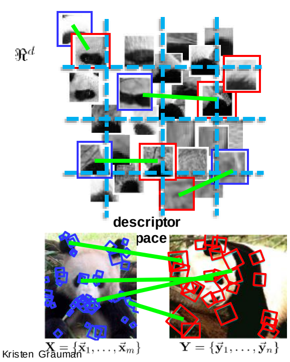
(m=num pts)

$\min_{\pi: \mathbf{X} \rightarrow \mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} ||\mathbf{x}_i - \pi(\mathbf{x}_i)||$ nate matching kernel that makes it practical to compare large sets of features based on their partial correspondences.

[Previous work: Indyk & Thaper, Bartal, Charikar, Agarwal & Varadarajan, …]

Kristen Grauman

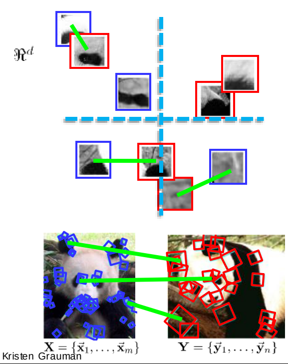
Pyramid match: main idea

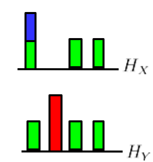


Feature space partitions serve to "match" the local descriptors within successively wider regions.

Kristen Grauman

Pyramid match: main idea





$$\mathcal{I}(H_X, H_Y) = \sum_j \min\left(H_X(j), H_Y(j)\right) = 3$$

Histogram intersection counts number of possible matches at a given partitioning.

Kristen Grauman

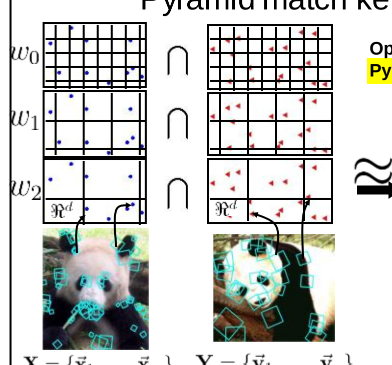
Pyramid match kernel

$$K_\Delta(X,Y) = \sum_{i=0}^{L} \underbrace{2^{-i}}_{\text{measures difficulty of a match at level } i} \underbrace{\mathcal{I}\left(H_X^{(i)}, H_Y^{(i)}\right) - \mathcal{I}\left(H_X^{(i-1)}, H_Y^{(i-1)}\right)}_{\text{number of newly matched pairs at level } i}$$

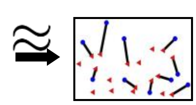
- For similarity, weights inversely proportional to bin size (or may be learned)
- Normalize these kernel values to avoid favoring large sets

[Grauman & Darrell, ICCV 2005]

Pyramid match kernel



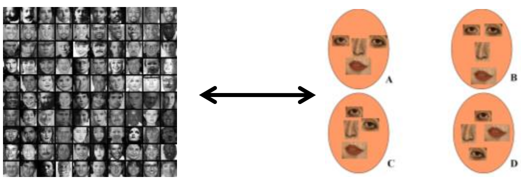
Optimal match: O(m^3)
Pyramid match: O(mL)



optimal partial matching

Kristen Grauman

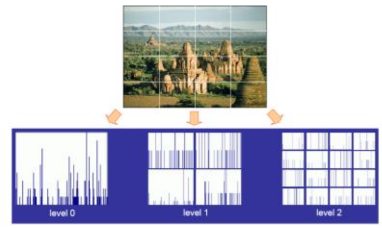
Unordered sets of local features: **No** spatial layout preserved!



Too much? Too little?

Spatial pyramid match

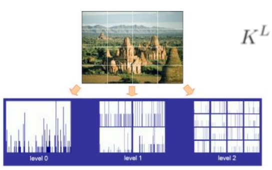
- Make a pyramid of bag-of-words histograms.
- Provides some loose (global) spatial layout information



[Lazebnik, Schmid & Ponce, CVPR 2006]

Spatial pyramid match

- Make a pyramid of bag-of-words histograms.
- Provides some loose (global) spatial layout information



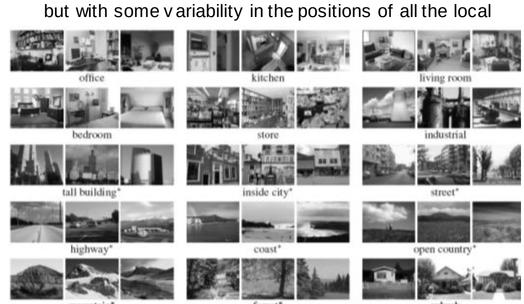
$$K^L(X,Y) = \sum_{m=1}^{M} \kappa^L(X_m, Y_m)$$

Sum over PMKs computed in *image coordinate* space, one per word.

[Lazebnik, Schmid & Ponce, CVPR 2006]

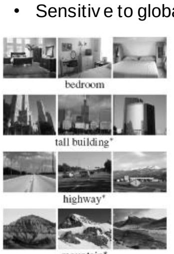
Spatial pyramid match

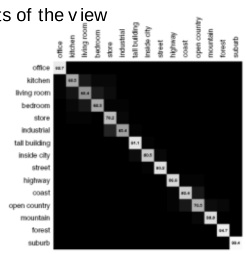
- Can capture **scene** categories well---texture-like patterns but with some variability in the positions of all the local



Spatial pyramid match

- Can capture **scene** categories well---texture-like patterns but with some variability in the positions of all the local pieces.
- Sensitive to global shifts of the view





Confusion table

Multi-class SVMs

- Achieve multi-class classifier by combining a number of binary classifiers
- **One vs. all**
  - 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
- **One vs. one**
  - Training: learn an SVM for each pair of classes
  - Testing: each learned SVM "votes" for a class to assign to the test example

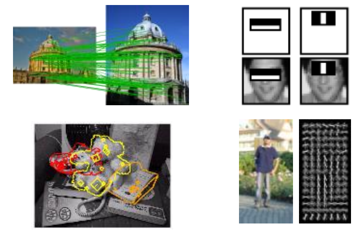
Kristen Grauman

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

Basic recognition models so far



Instances: recognition by alignment

Categories: Holistic appearance models (and sliding window detection)

Kristen Grauman

Summary so far

- 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
    - 80M Tiny Images studies
  - Support vector machines
    - HOG person detection example
    - Pyramid match kernel

Evolution of methods

- Hand-crafted models
- 3D geometry
- Hypothesize and align

- Hand-crafted features
- Learned models
- Data-driven

- "End-to-end" learning of features and models*,**

Next

- Convolutional neural networks
  - Guest lecture by Dinesh Jayaraman