Traditional Object Recognition
Background

• Current state-of-the-art: Neural networks and Deep Learning
Starting next week

• Today: Traditional, pre-DL, approaches
Algorithmic basics
Important to understand the field
Important for various other applications
Outline

• Specific object recognition

Identifying the same object in different images despite variations in pose and illumination

• Object category recognition

Identifying the object category despite variations in pose, illumination and intra-class variation
Outline

• Specific object recognition
  • Problem definition
  • Model-based
  • Image-based
  • Hybrid models
  • Visual words and indexing
  • Geometric constraints with Ransac

• Object category recognition
Specific objects vs. category-level objects

A specific object = an instance of an object class 
e.g. “my car” instead of “a car”
Example app

search photos on the web for particular places

Find these landmarks ...in these images and 1M more

Slide credit: J. Sivic
Application: Large-Scale Retrieval

Query Results from 5k Flickr images (demo available for 100k set)

[Philbin CVPR’07]
Example Applications

Mobile tourist guide
- Self-localization
- Object / landmark recognition
- Augmented reality
- Wine label rec.
  (Vivino, 1st CV app in Samsung SmartWatch powered by kooaba (now Qualcomm))

[Quack, Leibe, Van Gool, CIVR’08]
Main challenges:
• Pose / viewpoint changes
• Illumination variation
• Occlusion
• Clutter
Determine a numerical representation

\[ [2, 3, 19, \ldots, 0.4] \]

\[ [12, -4, 21, \ldots, 7] \]

\[ [0, -21, 5, \ldots, 8] \]

\[ [-4, 9, 11, \ldots, -0.9] \]
Outline

• Specific object recognition
  • Problem definition
  • Model-based
  • Image-based
  • Hybrid models
  • Visual words and indexing
  • Geometric constraints with Ransac

• Object category recognition
Outline

• Specific object recognition
  • Problem definition
  • **Model-based**
  • Image-based
  • Hybrid models
  • Visual words and indexing
  • Geometric constraints with Ransac

• Object category recognition
Once upon a time...

comparing image features with features of objects in a database, trying to figure out type + pose

Representation: Local geometric features

Slow hypothesise-and-verify process
Model-based approaches

Wireframe model for 3D objects
Early attempts…1965

Blocks world model, Roberts et al., 1965

More recent…1985

Dealing with occlusions, Lowe, 1985
More recently: Invariant-based recognition of planar shapes

The crucial advantage of invariants is that they decouple object type and pose issues.

Ex. given here for completely visible planar shapes
- under affine distortions
- using invariant signatures of the outlines

Image on the left is compared against database images of various animals like that of the matching swan on the right.
Example

\[ \int_{\text{start pt}} \left| \bar{x} - \bar{x}_1 \right| \bar{x}^{(1)} \, dt \quad \text{As a function of} \quad \int_{\text{start pt}} \text{abs} \left( \left| \bar{x}^{(1)} \bar{x}^{(2)} \right|^{\frac{1}{3}} \right) \, dt \]
Early attempts…1992

Projective invariance, Rothwell, 1992
Outline

• **Specific object recognition**
  • Problem definition
  • **Model-based** – local geometric features
  • Image-based
  • Hybrid models
  • Visual words and indexing
  • Geometric constraints with Ransac

• **Object category recognition**
Outline

• Specific object recognition
  • Problem definition
  • Model-based – local geometric features
  • Image-based
  • Hybrid models
  • Visual words and indexing
  • Geometric constraints with Ransac

• Object category recognition
Appearance based methods

The model of an object is simply its image(s).

A simple example: Template matching
Shift the template over the image and compare
(e.g. Normalized Cross-Correlation or Sum of Squared Diff.)

The problem is variation in the appearance because of changes in viewpoint / lighting

Zillions of templates!
The power of Principal Component Anal.

You remember PCA?

PCA represents data in a lower-dimensional space keeping most of the variance.

It was seen to be powerful for similar patterns like faces, that exhibit a lot of redundancy.

\[ I = \mu + \sum_{j} \alpha_j U_j \quad I : [\alpha_1, \alpha_2, \ldots, \alpha_J] \]

Representation: PCA weights of the image – global image representations.
Appearance manifold approach

Training
for every object:
- sample the set of viewing conditions (mainly viewpoints in this ex.)
- use these images as feature vectors (after manual bounding-box fitting around the object, rescaling, brightness normalization)
- apply a PCA over all the images (directly on the images)
- keep the dominant PCs (10-20 enough already)
- sequence of views for 1 object represent a manifold in the space of projections (fit data pts with splines, then densely resample if desired)

Recognition
what is the nearest manifold to a test image?

(Nayar et al. ‘96)
The images were put on a turntable, and imaged from a fixed distance and under a fixed elevation angle, hence pose changes were limited to changing azimuth angles.
Object-pose manifold

Appearance changes projected on PCs:
1D pose changes in this case by spinning around vert. axis
and only projection on 3 principal components shown

Sufficient characterization for recognition and pose estimation
Real-time system (Nayar et al. ‘96)
Eigenfaces for compact face representation

\[ S = \bar{S} + \sum_{j=1}^{m} \alpha_j S_j \]

\[ T = \bar{T} + \sum_{j=1}^{m} \beta_j T_j \]
3D reconstruction application

Using the PCA representation identify the 3D face that yields the image when projected

(self?-) portrait of the young

Anthony Van Dijck
3D PCA-based face reconstruction
# Comparison between model-based and appearance-based techniques

<table>
<thead>
<tr>
<th>Pure model-based</th>
<th>Pure appearance-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compact model</td>
<td>Large models</td>
</tr>
<tr>
<td>Can deal with clutter</td>
<td>Cannot deal with clutter</td>
</tr>
<tr>
<td>Slow analysis-by-synthesis</td>
<td>Efficient</td>
</tr>
<tr>
<td>Models difficult to produce</td>
<td>Models easy to produce</td>
</tr>
<tr>
<td>For limited object classes</td>
<td>For wide classes of objects</td>
</tr>
</tbody>
</table>
Outline

• Specific object recognition
  • Problem definition
  • Model-based – local geometric features
  • **Image-based** – global image representation
  • Hybrid models
  • Visual words and indexing
  • Geometric constraints with Ransac

• Object category recognition
Outline

• Specific object recognition
  • Problem definition
  • Model-based – local geometric features
  • Image-based – global image representation
  • Hybrid models
  • Visual words and indexing
  • Geometric constraints with Ransac

• Object category recognition
Outline

• Specific object recognition
  • Problem definition
  • Model-based – local geometric features
  • Image-based – global image representation
  • Hybrid models – local image representations
  • Visual words and indexing
  • Geometric constraints with Ransac

• Object category recognition
Local features: main components

1) Detection: Identify interest points

2) Description: Extract feature vector descriptors around them

3) Matching: Determine correspondence between descriptors in two views

\[
x_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]
\]

\[
x_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]
\]
Euclidean invariant feature

**Training**
1. look for corners (with the Harris detector)
2. take circular regions around these points, of different radii (cope a bit with scale changes)
3. calculate, from the regions, invariants under planar rotation
4. do this from different viewpoints, where the invariance cuts down the number of views needed (here no in-plane rotations necessary)

**Testing**
1. compare these invariants with those found for images in the database -> find matches
2. look for consistent placement of candidate matches / epipolar geometry
3. decide which object based on the number of remaining matches (best matching image: type+appr.pose)

(Schmid and Mohr ‘97)
Euclidean invariant features

Example (rotation) invariant:

\[ G_x G_x + G_y G_y \]

Where \( G_x \) and \( G_y \) represent horizontal and vertical derivatives of intensity weighted by a Gaussian profile ("Gaussian derivatives")

2nd example invariant:

\[ G_{xx} + G_{yy} \]

Where \( G_{xx} \) and \( G_{yy} \) represent 2nd order Gaussian derivatives

(Compute features for circles at different scales, i.e. take scale into account explicitly)
Example

Training examples for one object in the database

Test image

+ deal with cluttered background
+ need less training images
~ problems with uniform objects
Hybrid techniques

+ Rather compact model

+ Can deal with clutter and partial occlusion

+ Efficient

+ Models easy to produce
  (take images, fewer than in pure appearance-based method)

+ For rather wide class of objects
  (almost as wide as in pure appearance based, but there is a problem with untextured objects)
Improving on invariance

• Interest points and local image representation became very popular – state-of-the-art for long time

• The invariance of the descriptors can be improved using features extraction methods you have seen in the previous lectures

• Fewer images for training would be needed
Outline

• Specific object recognition
  • Problem definition
  • Model-based – local geometric features
  • Image-based – global image representation
  • Hybrid models – local image representations
  • Visual words and indexing
  • Geometric constraints with Ransac

• Object category recognition

Robustifying hybrid models
Supporting the matching step

1) Too slow if naively done

2) Will often fail when only based on descriptor matching
Supporting the matching step

1) Too slow if naively done

2) Will often fail when only based on descriptor matching
Supporting the matching step

1) Hierarchical vocabulary tree for speed-up
Indexing local features

With potentially thousands of interest pts + their descriptors per image, and hundreds to millions of images to search, how to efficiently find those relevant to a new test image?

Quantize/cluster the descriptors into `visual words’
And match words hierarchically: vocabulary tree
Use Inverted file indexing schemes
Visual words: main idea

- Extract some local features from several images

  e.g., SIFT descriptor space: each point is 128-dimensional

Slide credit: D. Nister, CVPR 2006
Visual words: main idea
Visual words: main idea
Visual words: main idea
Each point is a local descriptor, e.g. SIFT vector.
Vector quantize feature space
= cluster the features
K-means clustering

1. randomly initialize K cluster centers
2. assign each feature to nearest cluster center
3. recompute cluster center (mean)
4. iterate from 2, until convergence
Hierarchical K-means clustering

Allows to use larger vocabularies and thereby yields better results

In the example $k=3$, but typically it is chosen higher, e.g. $k=10$ and 6 layers could be used for search in about 1M images
Visual words

Ex: each group of patches belongs to same visual word.

Figure from Sivic & Zisserman, ICCV 2003
Indexing local features: inverted file index

For text documents, an efficient way to find all pages on which a word occurs is to use an index...

We want to find all images in which a visual word occurs.
Inverted file index

Database images are loaded into the index, mapping words to image numbers.
New query image is mapped to indices of database images that share a word.
Retrieval with vocabulary tree + inverted file index

[Slide credit: David Nister]

[Nister & Stewenius, CVPR'06]
Performance

Evaluated on large databases
Indexing with up to 1M images
Online recognition for database of 50,000 CD covers
Retrieval in ~1s
Best with very large visual vocabularies

NOTE: object class recognition typically done with smaller vocabularies
Supporting the matching step

1) Too slow if naively done

2) Will often fail when only based on descriptor matching
Supporting the matching step

[Chum, Werner, Matas]
Matching can start from interest points and their descriptors, but such matching is rather fragile. Typically, several `matches’ are wrong, so-called outliers, and one needs to add a test on the configuration of the matches in order to remove the outliers and keep the correct inliers.

Epipolar geometry and projective matching are often used tests, using RANSAC to withstand unavoidable mismatches.  
RANDom SAmple Consensus
The RANSAC test on **epipolar geometry** assumes that there is a fundamental matrix that matches agree with, and

The RANSAC test on **projectivities** assumes that there is a projectivity that maps points in the first image onto the matching points in the second

Such tests allow for the elimination of many outliers
RANSAC

RANSAC assumes the data consists of "inliers", i.e. correct matches, and "outliers", i.e. incorrect matches.

From a set of match candidates,
1. randomly select the minimal nmb of matches to formulate an initial test hypothesis (e.g. 7 for epipolar geometry or 4 for a projectivity; this nmb better be small since the selected tuple must not contain any outlier match for it to work)
2. check how consistent other matches are with this hypothesis, i.e. in how far it is supported
3. use all supporting matches to refine the hypothesis and discard the rest

Finally, RANSAC selects the hypothesis with maximal support after a fixed number of trials or after sufficient support was reached
How often should we draw?.... Suppose

\(n\) - minimum number of data required to fit the model
\(k\) - nmb of iterations / trials performed by the algorithm
\(t\) – threshold to determine when a match fits a model
\(d\) - nmb of `inliers’ needed for a model to be OK

\(t\) and \(d\) are typically chosen beforehand. The nmb of iterations \(k\) can then be calculated. Let \(p\) be the probability that RANSAC only selects inliers for the \(n\) data units generating a valid test at least once, i.e. the probability that the algorithm gets a good output. When \(w\) is the proportion of inliers (estimated),

\[
1 - p = (1 - w^n)^k
\]

is the probability that NO good hypothesis is selected.
Supporting the matching step

Ex. cleaning matches based on RANSAC-Ep.Geom.

[Chum, Werner, Matas]
RANSAC

The RANSAC test on **epipolar geometry** assumes that there is a fundamental matrix that matches agree with, and

The RANSAC test on **projectivities** assumes that there is a projectivity that maps points in the first image onto the matching points in the second

Such tests allow for the elimination of many outliers

but these tests make strong assumptions about the scene:

**Epipolar geometry**: rigidity of the scene (i.e. objects in the scene do not move with respect to each other)

**Projectivity**: the scene is not only rigid, but also (largely) planar

Nonetheless such tests help!
Outline

• Specific object recognition
  • Problem definition
  • Model-based – local geometric features
  • Image-based – global image representation
  • Hybrid models – local image representations
  • Visual words and indexing
  • Geometric constraints with Ransac

• Object category recognition
  • Problem definition
  • Classification: Bag-of-words
  • Detection: Sliding-windows
  • Generalized Hough Transform
Outline

• Specific object recognition
  • Problem definition
  • Model-based – local geometric features
  • Image-based – global image representation
  • Hybrid models – local image representations
  • Visual words and indexing
  • Geometric constraints with Ransac

• Object category recognition
  • Problem definition
  • Classification: Bag-of-words
  • Detection: Sliding-windows
  • Generalized Hough Transform
Specific objects vs. object categories
Object Categorization

How to recognize ANY car

How to recognize ANY cow
On top of factors affecting specific object recognition, the added complexity of intra-class variation

Modified from Kristen Grauman
Intra-class and inter-class variation

The difference between classes can be as small as that between instances of the same class ... yet the distinction needs to be made
Two main tasks: Classification and Detection

**Classification**: *is there* a car in this image? A binary answer is enough

**Detection**: *where* is the car? Need localization
Outline

• Specific object recognition
  • Model-based – local geometric features
  • Image-based – global image representation
  • Hybrid models – local image representations
  • Visual words and indexing
  • Geometric constraints with Ransac

• Object category recognition
  • Problem definition
  • **Classification: Bag-of-words**
  • Detection: Sliding-windows
  • Generalized Hough Transform
Building blocks: Local features and Visual Words

Figure from Sivic & Zisserman, ICCV 2003
Why visual words?

Useful for describing object appearance:

Appearances of objects vary a lot within a category

But appearance of local parts varies less!

Ideally: a part = a visual word, but that is not always so

Sivic & Zisserman 2003;
Csurka, Bray, Dance, & Fan 2004; many others.

Slide adapted from Kristen Grauman
Building Visual Vocabularies

Components:

- Sampling strategy: extract features densely, or at interest points only, randomly, … ?
- Clustering / quantization algorithm

Questions / Issues:

- Unsupervised vs. supervised
- What corpus provides features (universal vocabulary?)
- Vocabulary size, number of words

Slide credit: Kristen Grauman
Sampling Strategies

Sparse, at interest points

Dense, uniformly

Randomly

- To find specific, textured objects, sparse sampling from interest points often more reliable.
- Multiple complementary interest operators offer more image coverage.
- For object categorization, dense sampling offers better coverage.

[See Nowak, Jurie & Triggs, ECCV 2006]
Clustering / Quantization Methods

Typically specific object recognition works better with larger vocabularies, object class recognition with smaller ones (words cover more variability in that case)

k-means (typical choice), agglomerative clustering, mean-shift, …

Hierarchical clustering: allows faster insertion / word assignment while still allowing large vocabularies

Vocabulary tree [Nister & Stewenius, CVPR 2006]
Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that reach the brain from our eyes. For a long time it was thought that the retinal image was transmitted point by point to visual centers in the brain; the cerebral cortex served as a movie screen upon which the image in the eye was projected. Through the discoveries of Hubel and Wiesel we now know that behind the origin of the visual perception there is a considerably more complicated course of events. By following the visual impulses along their path through layers of the optical cortex, Hubel and Wiesel have been able to demonstrate that the message about the image falling on the retina undergoes a step-wise analysis in a system of nerve cells stored in columns. In this system each cell has its specific function and is responsible for a specific detail in the pattern of the retinal image.

China is forecasting a trade surplus of $90bn (£51bn) to $100bn this year, a threefold increase on 2004’s $32bn. The Commerce Ministry said the surplus would be created by a predicted 30% jump in exports this year, against a 18% rise in imports. The figures are likely to further annoy the US, which has long argued that China’s exports are unfairly helped by a deliberately undervalued yuan. Beijing agrees the surplus is too high, but says the yuan is only one factor. Bank of China governor Zhou Xiaochuan said the country also needed to do more to boost domestic demand so more goods stayed within the country. China increased the value of the yuan against the dollar by 2.1% in July and permitted it to trade within a narrow band, but the US wants the yuan to be allowed to trade freely. However, Beijing has made it clear that it will take its time and tread carefully before allowing the yuan to rise further in value.
Object → Bag of ‘words’

Source: ICCV 2005 short course, Li Fei-Fei
Bags of Visual Words

Summarize entire image based on its distribution (histogram) of word occurrences.

Analogous to bag of words representation commonly used for documents.

Advantage of histogram is that irrespective of the nmb of local features, the size is always the same.
Comparing Bags of Words

Any histogram comparison measure can be used here.

E.g. normalized scalar product between histograms
(or with more sophisticated classifiers mentioned shortly)

\[
\begin{align*}
\text{sim}(d_j, q) &= \frac{d_j \cdot q}{|d_j| \times |q|} \\
&= \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{j=1}^{t} w_{i,q}^2}}
\end{align*}
\]

(histograms in database) (histogram query image)
Learning/Recognition with BoW Histograms

Bag of words enable to describe the unordered feature set with a single vector of fixed dimensionality.

Provides easy way to use distribution of feature types with various learning algorithms requiring vector input.

Slide credit: Kristen Grauman
BoW for Object Categorization

Works pretty well for whole-image classification

Csurka et al. (2004), Willamowski et al. (2005), Grauman & Darrell (2005), Sivic et al. (2003, 2005)

{face, flowers, building}
Image Classification (more sophisticated classifiers)

Given bag-of-words of images from different classes, how do we learn a model for distinguishing them?
Discriminative Methods

Learn a decision rule (classifier) assigning bag-of-words representations of images to different classes.

Slide adapted from Svetlana Lazebnik
Classification

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

**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,...
BoW for Image Classification

Caltech6 dataset

<table>
<thead>
<tr>
<th>class</th>
<th>bag of features</th>
<th>bag of features</th>
<th>Parts-and-shape model</th>
</tr>
</thead>
<tbody>
<tr>
<td>airplanes</td>
<td>98.8</td>
<td>97.1</td>
<td>90.2</td>
</tr>
<tr>
<td>cars (rear)</td>
<td>98.3</td>
<td>98.6</td>
<td>90.3</td>
</tr>
<tr>
<td>cars (side)</td>
<td>95.0</td>
<td>87.3</td>
<td>88.5</td>
</tr>
<tr>
<td>faces</td>
<td>100</td>
<td>99.3</td>
<td>96.4</td>
</tr>
<tr>
<td>motorbikes</td>
<td>98.5</td>
<td>98.0</td>
<td>92.5</td>
</tr>
<tr>
<td>spotted cats</td>
<td>97.0</td>
<td>—</td>
<td>90.0</td>
</tr>
</tbody>
</table>

Good performance for classification (object present/absent)

Better than some more elaborate part-based models with spatial constraints…

What could be possible reasons why?
Properties of BoW Representations

The bag of words removes spatial layout.

This is both a strength and a weakness.

Why a strength? $A \Rightarrow B$

Why a weakness? $A \Rightarrow C, D$

Slide adapted from Bill Freeman
BoW Representation: Spatial Information

A bag of words is an *orderless* representation: discarding spatial relationships between features.

Options to bring in some spatial info:
- Visual “phrases” : frequently co-occurring words
- Semi-local features : describe configuration, neighborhood
- Let position be part of each feature descriptor
- Count bags of words only within sub-grids of an image

Slide credit: Kristen Grauman
BoW Representation: Spatial Information

A bag of words is an *orderless* representation: discarding spatial relationships between features

Options to bring in some spatial info:

- Visual “phrases”: frequently co-occurring words
- Semi-local features: describe configuration, neighborhood
- Let position be part of each feature descriptor
- Count bags of words only within sub-grids of an image

Slide credit: Kristen Grauman
Spatial Pyramid Representation

Representation in-between orderless BoW and structural description

Slide credit: Svetlana Lazebnik

[Lazebnik, Schmid & Ponce, CVPR’06]
Spatial Pyramid Representation

Representation in-between orderless BoW and structural description

Slide credit: Svetlana Lazebnik

[Lazebnik, Schmid & Ponce, CVPR’06]
Spatial Pyramid Representation

Representation in-between orderless BoW and structural description
Spatial Pyramid description matching

Can capture scene categories well

office

kitchen

living room

bedroom

store

industrial

tall building*

inside city*

street*

highway*

coast*

open country*

mountain*

forest*

suburb
Spatial Pyramid description matching

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
Example queries and NN with spatial pyramid matching
Discussion: Bag-of-Words

Pros:
- Flexible to geometry / deformations / viewpoint
- Compact summary of image content
- Provides vector representation for sets
- Empirically good recognition results in practice

Cons:
- Basic model ignores geometry – can verify afterwards, or embed within feature descriptors
- Background and foreground mixed when bag covers whole image
- Interest points or sampling: no guarantee to capture object-level parts
- Optimal vocabulary formation remains unclear
Outline

• Specific object recognition
  • Model-based – local geometric features
  • Image-based – global image representation
  • Hybrid models – local image representations
  • Visual words and indexing
  • Geometric constraints with Ransac

• Object category recognition
  • Problem definition
  • Classification: Bag-of-words
  • Detection: Sliding-windows
  • Generalized Hough Transform
Detection:
Sliding-window approaches
Detection via Classification: Main Idea

Basic component: a binary classifier

Car/non-car Classifier

Yes, car.
No, not a car.

Slide credit: Kristen Grauman
Detection via Classification: Main Idea

If object may be in a cluttered scene, slide a window around looking for it.

![Image of a street scene with a highlighted car]

a brute-force approach with many local decisions.

Slide adapted from Kristen Grauman
Detection via Classification: Main Idea

Fleshing out this pipeline a bit more, we need to:
1. Obtain training data
2. Define features
3. Train a classifier

Slide credit: Kristen Grauman
Feature extraction: Global Appearance

Simple holistic descriptions of image content

- Grayscale / color histogram
- Vector of pixel intensities
Feature Extraction: Global Appearance

Pixel-based representations sensitive to small shifts

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

Cartoon example: an albino koala

Slide credit: Kristen Grauman
Gradient-based Representations

Consider edges and (oriented) intensity gradients
Gradient-based Representations

Consider edges and (oriented) intensity gradients

Locally orderless: offers invariance to small shifts and rotations
Contrast-normalization: try to correct for variable illumination

One very popular example: Histogram of Oriented Gradients
[Dalal & Triggs, CVPR 2005]
HoG pedestrian detector

Finally, a decision needs to be made for each window…

example result:
Sliding Windows

Rectangular Integral Image Filters
Feature extraction

“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

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

\[
D = 1 + 4 - (2 + 3) = A + (A + B + C + D) - (A + C + A + B) = D
\]
Example

Integral Image

(x, y)
Large Library of Filters

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

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

Obviously in a classifier you would need to select the informative ones and to form the classifier. Not all of them will be informative!

[Viola & Jones, CVPR 2001]

Slide credit: Kristen Grauman
Classifier Construction

How to compute a decision for each subwindow?
Classifier Choices

**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,…

Slide adapted from Antonio Torralba
Sliding Windows

AdaBoost
Boosting

Build a strong classifier by combining many “weak classifiers”, which need only be better than chance
Sequential learning process: at each iteration, add a weak classifier
Flexible to choice of weak learner
  including fast simple classifiers that alone may be inaccurate

We’ll look at Freund & Schapire’s AdaBoost algorithm
  Easy to implement
  Base learning algorithm for Viola-Jones face detector

Slide adapted from Kristen Grauman
AdaBoost: Intuition

Consider a 2D feature space with **positive** and **negative** examples.

Each weak classifier splits the training examples with at least 50% accuracy.

Examples misclassified by a previous weak learner are given more emphasis at future rounds.

Figure adapted from Freund and Schapire

Slide credit: Kristen Grauman
AdaBoost: Intuition

Weights Increased

Weak Classifier 1

Weak Classifier 2

Slide credit: Kristen Grauman
AdaBoost: Intuition

Final classifier is a combination of the weak classifiers.

Weights Increased

Weak Classifier 1

Weak Classifier 2

Weak classifier 3

Slide credit: Kristen Grauman
AdaBoost Algorithm

Start with uniform weights on training examples

For $T$ rounds

Evaluate weighted error for each feature, pick best.

Re-weight the examples:

- Incorrectly classified $\Rightarrow$ more weight
- Correctly classified $\Rightarrow$ less weight

Final classifier is combination of the weak ones, weighted according to the error they had.

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m} \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where $m$ and $l$ are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$

so that $w_t$ is a probability distribution.
  2. For each feature, $j$, train a classifier $h_j$ which is restricted to using a single feature. The error is evaluated with respect to $w_t$, $e_j = \sum_i w_i |h_j(x_i) - y_i|$.
  3. Choose the classifier, $h_t$, with the lowest error $\epsilon_t$.
  4. Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-\epsilon_t}$$

where $\epsilon_t = 0$ if example $x_i$ is classified correctly; $\epsilon_t = 1$ otherwise, and $\beta_t = \frac{e_t}{\epsilon_t}$.

- The final strong classifier is:

$$h(x) = \begin{cases} 
1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\
0 & \text{otherwise}
\end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$.

[Freund & Schapire 1995]
Example: Face Detection

For frontal faces global appearance models + a sliding window detection approach fit well:

- Regular 2D structure
- Center of face almost shaped like a “patch”/window

Now we’ll take AdaBoost and see how the Viola-Jones face detector works

Slide credit: Kristen Grauman
Large Library of Filters

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

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

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

[Viola & Jones, CVPR 2001]
Want to select the single rectangle feature and threshold that best separates positive (faces) and (non-faces) training examples, in terms of weighted error.

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.

Slide credit: Kristen Grauman

[Viola & Jones, CVPR 2001]
AdaBoost for Efficient Feature Selection

Image features = weak classifiers

For each round of boosting:

- Evaluate each rectangle filter on each example
- Sort examples by filter values
- Select best threshold for each filter (min error)
  
  Sorted list can be quickly scanned for the optimal threshold
- Select best filter/threshold combination
- Weight on this feature is a simple function of error rate

Reweight examples


Slide credit: Kristen Grauman
Efficiency Considerations

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?
→ cascade of classifiers of increasing complexity
Efficiency Considerations

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?

→ cascade of classifiers of increasing complexity
→ conservative classifiers: try to keep in all positives, letting passing some false positives to the next stages as price to pay
→ later classifiers are more expensive (more complex) having to check the fewer surviving patterns
→ this cascaded scheme isn’t discussed in detail

Slide adapted from Kristen Grauman
Viola-Jones Face Detector: Summary

Train cascade of classifiers with AdaBoost

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

- [Implementation available in OpenCV: http://sourceforge.net/projects/opencvlibrary/]

Slide credit: Kristen Grauman
Viola-Jones Face Detector: Results

First two features selected

Speed
384 by 288 pixel images detected at 15 fps on a conventional 700 MHz Intel Pentium III in 2001
Training time was weeks

Slide adapted from Kristen Grauman
Viola-Jones Face Detector: Results

Slide credit: Kristen Grauman
Viola-Jones Face Detector: Results

Slide credit: Kristen Grauman
Viola-Jones Face Detector: Results

Slide credit: Kristen Grauman
You Can Try It At Home…

The Viola & Jones detector was a huge success

First real-time face detector available
But weak on non-frontal faces, strong expressions
Many derivative works and improvements

C++ implementation available in OpenCV [Lienhart, 2002]
http://sourceforge.net/projects/opencvlibrary/

P. Viola, M. Jones, Robust Real-Time Face Detection, IJCV, Vol. 57(2), 2004
Discussion: Sliding-Windows

Pros

- Simple detection protocol to implement
- Good feature choices critical, but part of the process
- Past successes for certain classes
- Good detectors available (Viola&Jones, HOG, etc.)

Cons/Limitations

- High computational complexity
  - More than 100,000 locations in an image, no exception
  - This puts tight constraints on the classifiers we can use.
  - If training binary detectors independently for different classes, this means cost increases linearly with number of classes.
  - One can try to focus on promising locations only, e.g. by first running a generic object detector (‘objectness scores’)
- With so many windows, false positive rate better be low
- Typically need fully supervised training data (= bounding-boxes)

Slide adapted from Kristen Grauman
Limitations (continued)

Not all objects are “box” shaped

Slide credit: Kristen Grauman
Limitations (continued)

Non-rigid, deformable objects not captured well with representations assuming a fixed 2D structure and/or must assume fixed viewpoint

Objects with less-regular textures not captured well with holistic appearance-based descriptions
Limitations (continued)

When considering windows in isolation, context is lost.

Sliding window vs. Detector’s view

Figure credit: Derek Hoiem
Limitations (continued)

In practice, often entails large training set with manually cropped windows (expensive)

Requiring good match to a global appearance description can lead to sensitivity to partial occlusions
Generalized Hough Transform

Implicit Shape Model (ISM)
Implicit Shape Model (ISM)

Basic ideas

- Learn an appearance codebook
- Learn a star-topology structural model
  - Features are considered independent given obj. center

Algorithm: probabilistic Generalized Hough Transform
Implicit Shape Model: Basic Idea

Visual vocabulary is used to index votes for object position [a visual word = a “part”].

Training image

Visual codeword with displacement vectors

Implicit Shape Model: Basic Idea

Objects are detected as consistent configurations of the observed parts (visual words).

Implicit Shape Model - Representation

Learn appearance codebook
- Extract local features at interest points
- Agglomerative clustering => codebook

Learn spatial distributions
- Match codebook to training images
- Record matching positions on object

Training images (+object center and scale)

Appearance codebook

Spatial occurrence distributions
Implicit Shape Model - Recognition

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]
Implicit Shape Model - Recognition

Backprojected Hypotheses

Interest Points

Matched Codebook Entries

Probabilistic Voting

3D Voting Space (continuous)

Backprojection of Maxima

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]
Example Results: Chairs

Office chairs

Dining room chairs
You Can Try It At Home...

Linux binaries available
  Including datasets & several pre-trained detectors
  http://www.vision.ee.ethz.ch/bleibe/code
Discussion: Implicit Shape Model

Pros:
- Works well for many different object categories
  - Both rigid and articulated objects
- Flexible geometric model
  - Can recombine parts seen on different training examples
- Learning from relatively few (50-100) training examples
- Optimized for detection, good localization properties

Cons:
- Needs supervised training data
  - Object bounding boxes for detection
  - Segmentations for top-down segmentation
- Only weak geometric constraints
  - Result segmentations may contain superfluous body parts.
- Purely representative model
  - No discriminative learning