EECS 442 – Computer vision

Object Recognition

• Intro
• Recognition of 3D objects
  • Recognition of object categories:
    • Bag of world models
    • Part based models
    • 3D object categorization

Computer Vision: Algorithms and Applications. R. Szeliski
Pages 696-709
Categorical vs Single Instance
Challenges:

Variability due to:

• View point
• Illumination
• Occlusions
• Etc..
Challenges: intra-class variation
Basic properties

• Representation
  – How to represent an object category; which classification scheme?

• Learning
  – How to learn the classifier, given training data

• Recognition
  – How the classifier is to be used on novel data
Part 1: Bag-of-words models

This segment is based on the tutorial “Recognizing and Learning Object Categories: Year 2007”, by Prof. A. Torralba, R. Fergus and F. Li
Related works

• Early “bag of words” models: mostly texture recognition

• Hierarchical Bayesian models for documents (pLSA, LDA, etc.)
  – Hoffman 1999; Blei, Ng & Jordan, 2004; Teh, Jordan, Beal & Blei, 2004

• Object categorization
  – Csurka, Bray, Dance & Fan, 2004; Sivic, Russell, Efros, Freeman & Zisserman, 2005; Sudderth, Torralba, Freeman & Willsky, 2005;

• Natural scene categorization
  – Vogel & Schiele, 2004; Fei-Fei & Perona, 2005; Bosch, Zisserman & Munoz, 2006
Object \rightarrow \text{Bag of ‘words’}
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 was a movie screen, so to speak, 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 in the brain there is a considerably more complicated course of events. By following the visual impulses along their path to the various cell 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 to $750bn, compared with a 18% rise in imports to $660bn. This is likely to 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.
definition of “BoW”

– Independent features
definition of “BoW”

– Independent features
– histogram representation
Representation

feature detection & representation

image representation

codewords dictionary

category models (and/or) classifiers

recognition

category decision
1. Feature detection and description
1. Feature detection and description

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
1. Feature detection and description

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005

- Interest point detector
  - Csurka, et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic, et al. 2005
1. Feature detection and description

- **Regular grid**
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005

- **Interest point detector**
  - Csurka, Bray, Dance & Fan, 2004
  - Fei-Fei & Perona, 2005
  - Sivic, Russell, Efros, Freeman & Zisserman, 2005

- **Other methods**
  - Random sampling (Vidal-Naquet & Ullman, 2002)
  - Segmentation based patches (Barnard, Duygulu, Forsyth, de Freitas, Blei, Jordan, 2003)
1. Feature detection and description

- Compute SIFT descriptor
  [Lowe'99]

- Normalize patch

- Detect patches
  [Mikojaczyk and Schmid '02]
  [Mata, Chum, Urban & Pajdla, ’02]
  [Sivic & Zisserman, ’03]
2. Codewords dictionary formation
2. Codewords dictionary formation
Example: color feature
Example: color feature

\[
\begin{align*}
\text{b} & \quad \text{r} \\
\text{g} & \quad \text{R}=15 \\
& \quad \text{G}=189 \\
& \quad \text{B}=2
\end{align*}
\]
2. Codewords dictionary formation

Cluster center = code word

Clustering/vector quantization
2. Codewords dictionary formation

- Image patch examples of codewords
2. Codewords dictionary formation

Fei-Fei et al. 2005
2. Codewords dictionary formation

- Typically a codeword dictionary is obtained from a training set comprising all the object classes of interests.
Visual vocabularies: Issues

• How to choose vocabulary size?
  – Too small: visual words not representative of all patches
  – Too large: quantization artifacts, overfitting

• Computational efficiency
  – Vocabulary trees
    (Nister & Stewenius, 2006)
3. Bag of word representation

- Nearest neighbors assignment
- K-D tree search strategy
3. Bag of word representation

Codewords dictionary

frequency

codewords

Codewords dictionary
Representing textures

- Texture is characterized by the repetition of basic elements or *textons*
- For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters


Credit slide: S. Lazebnik
Representing textures


Credit slide: S. Lazebnik
Invariance issues

• Scale? Rotation? View point? Occlusions?
  – Implicit;
  – depends on detectors and descriptors

Kadir and Brady. 2003
1. feature detection & representation

2. codewords dictionary

3. category models
Category models

Class 1

Class N
Recognition

codewords dictionary

category models (and/or) classifiers

category decision
Learning and Recognition

1. Discriminative method:
   - NN
   - SVM

2. Generative method:
   - graphical models
Discriminative classifiers

category models

Model space

Class 1

Class N
Discriminative classifiers

Query image

Winning class: pink

Model space
Nearest Neighbors classifier

Query image

Model space

Winning class: pink

- Assign label of nearest training data point to each test data point
K-Nearest Neighbors classifier

Query image

Model space

Winning class: pink

- For a new point, find the k closest points from training data
- Labels of the k points “vote” to classify
- Works well provided there is lots of data and the distance function is good
K-Nearest Neighbors classifier

- Voronoi partitioning of feature space for 2-category 2-D and 3-D data
- For k dimensions: k-D tree = space-partitioning data structure for organizing points in a k-dimensional space
- Enable efficient search
Functions for comparing histograms

• L1 distance

\[ D(h_1, h_2) = \sum_{i=1}^{N} | h_1(i) - h_2(i) | \]

• \( \chi^2 \) distance

\[ D(h_1, h_2) = \sum_{i=1}^{N} \frac{(h_1(i) - h_2(i))^2}{h_1(i) + h_2(i)} \]

• Quadratic distance (cross-bin)

\[ D(h_1, h_2) = \sum_{i,j} A_{ij} (h_1(i) - h_2(j))^2 \]

Jan Puzicha, Yossi Rubner, Carlo Tomasi, Joachim M. Buhmann: Empirical Evaluation of Dissimilarity Measures for Color and Texture. ICCV 1999
Learning and Recognition

1. Discriminative method:
   - NN
   - SVM

2. Generative method:
   - graphical models
Discriminative classifiers (linear classifier)

category models

Model space

Class 1

Class N
Linear classifiers

• We want to classify two classes of points
• Each point $x_i$ can have two labels \{pos, neg\}
Linear classifiers

• Find linear function (*hyperplane*) to separate positive and negative examples
Linear classifiers

• Find linear function *(hyperplane)* to separate positive and negative examples

\[ x_i \text{ positive: } x_i \cdot w + b \geq 1 \]
\[ x_i \text{ negative: } x_i \cdot w + b < -1 \]
Support vector machines

• Once \( w \) are learnt we can do classification

\[
\mathbf{w} \cdot \mathbf{x} + b = \text{Classification function}
\]

Test point

if \( \mathbf{x} \cdot \mathbf{w} + b \geq 1 \) \( \rightarrow \) class 1

if \( \mathbf{x} \cdot \mathbf{w} + b < -1 \) \( \rightarrow \) class 2
Linear classifiers

- Find linear function (hyperplane) to separate positive and negative examples

\[ x_i \text{ positive: } x_i \cdot w + b \geq 1 \]
\[ x_i \text{ negative: } x_i \cdot w + b < -1 \]

Which hyperplane is best?
Support vector machines

- Find hyperplane that maximizes the margin between the positive and negative examples

Support vectors: \( x_i \cdot w + b = \pm 1 \)

Distance between point and hyperplane:
\[
\frac{|x_i \cdot w + b|}{\|w\|}
\]

Margin = \( \frac{2}{\|w\|} \)

Solution:
\[
w = \sum_i \alpha_i y_i x_i
\]

Credit slide: S. Lazebnik
Support vector machines

- Classification

\[ \mathbf{w} \cdot \mathbf{x} + b = \sum_{i} \alpha_{i} y_{i} \mathbf{x}_{i} \cdot \mathbf{x} + b \]

Test point

if \( \mathbf{x} \cdot \mathbf{w} + b \geq 1 \) → class 1
if \( \mathbf{x} \cdot \mathbf{w} + b < -1 \) → class 2

Nonlinear SVMs

• Datasets that are linearly separable work out great:

• But what if the dataset is just too hard?

• We can map it to a higher-dimensional space:
Nonlinear SVMs

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

\[ \Phi: \mathbf{x} \rightarrow \varphi(\mathbf{x}) \]

lifting transformation

Slide credit: Andrew Moore
Nonlinear SVMs

• Nonlinear decision boundary in the original feature space:

\[ \sum_i \alpha_i y_i x_i \cdot x + b \quad \rightarrow \quad \sum_i \alpha_i y_i K(x_i, x) + b \]

• The kernel \( K \) = product of the lifting transformation \( \varphi(x) \):

\[ K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j) \]

NOTE:
• It is not required to compute \( \varphi(x) \) explicitly:
• The kernel must satisfy the “Mercer inequality”

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

• Histogram intersection kernel:

\[ I(h_1, h_2) = \sum_{i=1}^{N} \min(h_1(i), h_2(i)) \]

• Generalized Gaussian kernel:

\[ K(h_1, h_2) = \exp \left( - \frac{1}{A} D(h_1, h_2)^2 \right) \]

• \( D \) can be Euclidean distance, \( \chi^2 \) distance etc…

Overfitting

- A simple dataset.
- Two models

[Diagram showing linear and non-linear models]
Overfitting

• Let’s get more data.
• Simple model has better generalization.
Overfitting

• As complexity increases, the model overfits the data
• Training loss decreases
• Real loss increases
• We need to penalize model complexity = to regularize
What about multi-class SVMs?

- No “definitive” multi-class SVM formulation
- In practice, we have to obtain a multi-class SVM by combining multiple two-class SVMs
- One vs. others
  - Training: learn an SVM for each class vs. the others
  - 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
SVMs: Pros and cons

- **Pros**
  - Many publicly available SVM packages: [http://www.kernel-machines.org/software](http://www.kernel-machines.org/software)
  - Kernel-based framework is very powerful, flexible
  - SVMs work very well in practice, even with very small training sample sizes

- **Cons**
  - No “direct” multi-class SVM, must combine two-class SVMs
  - 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
Object recognition results

- ETH-80 database
  8 object classes
  (Eichhorn and Chapelle 2004)

- Features:
  - Harris detector
  - PCA-SIFT descriptor, $d=10$

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Complexity</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match [Wallraven et al.]</td>
<td>$O(dm^2)$</td>
<td>84%</td>
</tr>
<tr>
<td>Bhattacharyya affinity [Kondor &amp; Jebara]</td>
<td>$O(dm^3)$</td>
<td>85%</td>
</tr>
<tr>
<td>Pyramid match</td>
<td>$O(dmL)$</td>
<td>84%</td>
</tr>
</tbody>
</table>

Slide credit: Kristen Grauman
Pyramid match kernel

- Fast approximation of Earth Mover’s Distance
- Weighted sum of histogram intersections at multiple resolutions (linear in the number of features instead of cubic)

Spatial Pyramid Matching

\[ I(h_1, h_2) = \sum_{i=1}^{N} \min(h_1(i), h_2(i)) \]

\[ SPM(x_i, x) = \frac{1}{2L}HIK_0(x_i, x) + \ldots + \frac{1}{L-l+1}HIK_l(x_i, x) + \ldots + HIK_L(x_i, x) \]

Scene category dataset
Fei-Fei & Perona (2005), Oliva & Torralba (2001)
http://www-cvr.ai.uiuc.edu/ponce_grp/data

Multi-class classification results (100 training images per class)

<table>
<thead>
<tr>
<th>Level</th>
<th>Weak features (vocabulary size: 16)</th>
<th>Strong features (vocabulary size: 200)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Single-level</td>
<td>Pyramid</td>
</tr>
<tr>
<td>0 ((1 \times 1))</td>
<td>45.3 ±0.5</td>
<td>56.2 ±0.6</td>
</tr>
<tr>
<td>1 ((2 \times 2))</td>
<td>53.6 ±0.3</td>
<td>64.7 ±0.7</td>
</tr>
<tr>
<td>2 ((4 \times 4))</td>
<td>61.7 ±0.6</td>
<td>66.8 ±0.6</td>
</tr>
<tr>
<td>3 ((8 \times 8))</td>
<td>63.3 ±0.8</td>
<td></td>
</tr>
</tbody>
</table>

Fei-Fei & Perona: 65.2%
Scene category confusions

Difficult indoor images

kitchen  living room  bedroom
Caltech101 dataset

Fei-Fei et al. (2004)


Multi-class classification results (30 training images per class)

<table>
<thead>
<tr>
<th>Level</th>
<th>Weak features (16)</th>
<th>Strong features (200)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Single-level</td>
<td>Pyramid</td>
</tr>
<tr>
<td>0</td>
<td>15.5 ±0.9</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>31.4 ±1.2</td>
<td>32.8 ±1.3</td>
</tr>
<tr>
<td>2</td>
<td>47.2 ±1.1</td>
<td>49.3 ±1.4</td>
</tr>
<tr>
<td>3</td>
<td>52.2 ±0.8</td>
<td><strong>54.0 ±1.1</strong></td>
</tr>
</tbody>
</table>
Caltech101 comparison

Zhang, Berg, Maire & Malik, 2006

Pyramid matching
Discriminative models

**Nearest neighbor**

10^6 examples

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

**Support Vector Machines**

Guyon, Vapnik, Heisele, Serre, Poggio...

**Neural networks**

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

**Latent SVM**

Felzenszwalb 00
Ramanan 03...

**Boosting**

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

Courtesy of Vittorio Ferrari
Slide credit: Kristen Grauman

Slide adapted from Antonio Torralba
Learning and Recognition

1. Discriminative method:
   - NN
   - SVM

2. Generative method:
   - graphical models

→ Model the probability distribution that produces a given bag of features
EECS 442 – Computer vision

Object Recognition

- Intro
- Recognition of 3D objects
- Recognition of object categories:
  - Bag of world models
  - Part based models
  - 3D object categorization