Lecture 14

Visual recognition

• Object Classification – BoW models (part 2)
• 2D Object Detection
  • Template based approaches
  • Part-based approaches
Object → Bag of ‘words’
Representation

feature detection & representation

image representation

codewords dictionary

recognition

category models (and/or) classifiers

category decision
Category models

Class 1

Class N
Discriminative classifiers

category models

model space

Class 1

Class N
Discriminative classifiers

Query image

Winning class: pink

- Nearest neighbors
- Linear classifier
- SVM
Nearest Neighbors classifier

Query image

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
category models

Model space

• Instead of modeling decision regions, it estimates a hyperplane $w$ that separates training points that belong to different classes
• The training data is used to learn parameters of $w$ and then discarded
Linear classifiers

Query image

Model space

Winning class: pink

- Only $w$ is needed for classifying new data!
Select two hyperplanes such:
1. They separate the training points
2. There are no points between them
3. Their distance is maximized
### SVM classifiers

#### Query image

![Query image chart](image)

- Winning class: pink

#### Model space

- SVM classifiers

![Model space diagram](image)

- w
SVMs: Pros and cons

• Pros
  – Many publicly available SVM packages:
    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 (unlike neural networks or CNNs)

• Cons
  – Computation, memory
  – Learning can take a very long time for large-scale problems
  – No “direct” multi-class SVM, must combine two-class SVMs
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
Experimental results

- BoW model equipped with SVM and applied on Caltech 101

Fei-Fei et al. (2004)

Caltech 101

Random classification: 0.01%
Major drawback of BOW models

Don’t capture spatial information!
Spatial Pyramid Matching


\[ H = \begin{bmatrix} H_0^1 & H_1^2 & \cdots & H_4^2 & H_1^3 & \cdots & H_{16}^3 \end{bmatrix} \]

or, \( H = \) combination of \( H_i^j \) with appropriate weights
Experimental results

- Pyramid BoW model equipped with SVM and applied on Caltech 101

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

<table>
<thead>
<tr>
<th>Level</th>
<th>Single-level</th>
<th>Pyramid</th>
</tr>
</thead>
<tbody>
<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>54.0 ±1.1</td>
</tr>
</tbody>
</table>
Pyramid matching

Caltech 101

mean recognition rate per class

number of training examples per class

- Zhang, Berg, Maire & Malik (CVPR06)
- Lazebnik, Schmid & Ponce (CVPR06)
- Berg (thesis 06)
- Mutch & Lowe (CVPR06)
- Grauman & Darrell (tech report 06)
- Berg, Berg & Malik (CVPR05)
- Wang, Zhang & Fel-Fel (CVPR06)
- Holub, Welling & Perona (ICCV05)
- Serre, Wolf & Poggio (CVPR05)
- Fei-Fei, Fergus & Perona (GMBV04)
- SSD baseline
Discriminative models

Nearest neighbor
- 10^6 examples
- Shakhnarovich, Viola, Darrell 2003
- Berg, Berg, Malik 2005...

Support Vector Machines
- Guyon, Vapnik, Heisele, Serre, Poggio...

Latent SVM
- Structural SVM
- Felzenszwalb 2000
- Ramanan 2003, 2008...

Neural networks
- LeCun, Bottou, Bengio, Haffner 1998
- Krizhevsky, Sutskever, Hinton, 2012

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
Lecture 14

Visual recognition

• Object Classification – BoW models (part 2)
• 2D Object Detection
  • Template based approaches
  • Part-based approaches
Detection

Which object does this image contain? [where?]
Detection

– Recognition task

– Search strategy: Sliding Windows
  • Simple
  • Computational complexity (x, y, S, θ, N of classes)
    - BSW by Lampert et al 08
    - Also, Alexe, et al 10
Detection

– Recognition task

– Search strategy: Sliding Windows

  • Simple
  • Computational complexity \((x, y, S, \theta, N\) of classes)
    - BSW by Lampert et al 08
    - Also, Alexe, et al 10

• Localization

  • Prone to false positive

Non max suppression:
Canny ’86
....
Desai et al , 2009
Non-max suppression

Score = 0.6
Score = 0.8
Score = 0.8
Score = 0.1
Detection

– Recognition task

– Search strategy: Probabilistic “heat maps”
  
  • Fergus et al 03
  • Leibe et al 04
Lecture 14
Visual recognition

• Object Classification – BoW models (part 2)
• 2D Object Detection
  • Template based approaches
  • Part-based approaches
Template-based detection

1. Slide a window in image
   - E.g., choose position, scale orientation

2. Compare it with a template
   - Compute similarity to an example object or to a summary representation

3. Compute a score for each comparison and compute non-max suppression to remove weak scores

Exemplar
Dalal-Triggs pedestrian detector
Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

Represent an object as a collection of HoG templates
HoG = Histogram of Oriented Gradients

• Like SIFT, but...
  – Sampled on a dense, regular grid around the object
  – Gradients are contrast normalized in overlapping blocks
HOGs can be computed at multiple scales
Dalal-Triggs pedestrian detector

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

Courtesy of J Hayes
Dalal-Triggs pedestrian detector

Results
Tricks of the trade

• Details in feature computation really matter
  – E.g., normalization in Dalal-Triggs significantly improves detection rate at fixed false positive rate

• Template size
  – Typical choice is size of smallest detectable object

• “Jittering” to create synthetic positive examples
  – Create slightly rotated, translated, scaled, mirrored versions as extra positive examples

• Bootstrapping to get hard negative examples
  1. Randomly sample negative examples
  2. Train detector
  3. Keep negative examples that score > T
  4. Repeat until all high-scoring negative examples fit in memory

Courtesy of J Hayes
Limitations of template based approaches

They work

– *very well* for faces
– *fairly well* for cars and pedestrians
– *badly* for cats and dogs

• Why are some classes easier than others?
Limitations of template based approaches

Strengths

• Works very well for non-deformable objects with canonical orientations: faces, cars, pedestrians
• Fast detection

Weaknesses

• Not so well for highly deformable objects or “stuff”
• Not robust to occlusion
• Requires lots of training data if view points need to be encoded

Courtesy of J Hayes
Classic template-based Detectors

  – Basic idea of statistical template detection, bootstrapping to get “face-like”
    negative examples, multiple whole-face prototypes (in 1994)
  – “Parts” at fixed position, non-maxima suppression, simple cascade, rotation,
    pretty good accuracy, fast
  – Careful feature engineering, excellent results, cascade
• Viola-Jones (2001, 2004) : ~11,000
  – Haar-like features, Adaboost as feature selection, hyper-cascade, very fast,
    easy to implement
• Dalal-Triggs (2005) : ~6500
  – Careful feature engineering, excellent results, HOG feature, online code

Courtesy of J Hayes
Lecture 14

Visual recognition

• Object Classification – BoW models (part 2)
• 2D Object Detection
  • Template based approaches
  • Part-based approaches
Part Based Representation

- Object as set of parts
- Model:
  - Relative locations between parts
  - Appearance of each part

Figure from [Fischler & Elschlager 73]
History of Parts and Structure approaches

- Fischler & Elschlager 1973
- Yuille ‘91
- Brunelli & Poggio ‘93
- Lades, v.d. Malsburg et al. ‘93
- Cootes, Lanitis, Taylor et al. ‘95
- Amit & Geman ‘95, ‘99
- Ullman et al. 02
- Felzenszwalb & Huttenlocher ’00, ’04
- Crandall & Huttenlocher ’05, ’06
- Leibe & Schiele ’03, ’04
- Many papers since 2000
Deformations
Presence / Absence of Features

www.corbis.com
Sparse representation

Computationally tractable (\(10^5\) pixels \(\rightarrow\) \(10^1\) -- \(10^2\) parts)
But throw away potentially useful image information
Discriminative

Parts need to be distinctive to separate from other classes
Hierarchical representations

- Pixels → Pixel groupings → Parts → Object

Images from [Amit98, Bouchard05]
Hierarchical representations

- Deep learning architectures and ConvNets

Fukushima, 1980
LeCun, 1987
Hierarchical representations

- Deep learning architectures and ConvNets

(Lee et al., 2009)
Hierarchical representations

S.C. Zhu et al. and D. Mumford
Different connectivity structures

a) Constellation [13]  
Fergus et al. ’03  
Fei-Fei et al. ‘03

b) Star shape [9, 14]  
Crandall et al. ‘05  
Leibe 05; Felzenszwalb 09

c) k-fan (k = 2) [9]  
Crandall et al. ‘05  
Felzenszwalb & Huttenlocher ‘00

d) Tree [12]

e) Bag of features [10, 21]  
Csurka ’04  
Vasconcelos ‘00

f) Hierarchy [4]  
Bouchard & Triggs ‘05

g) Sparse flexible model  
Carneiro & Lowe ‘06

from Sparse Flexible Models of Local Features  
Gustavo Carneiro and David Lowe, ECCV 2006
Different connectivity structures

a) Constellation [13]
Fergus et al. ’03
Fei-Fei et al. ’03

b) Star shape [9, 14]
Crandall et al. ’05
Leibe 05; Felzenszwalb 09

Crandall et al. ’05
Felzenszwalb & Huttenlocher ’00

Fergus et al. ’03
Fei-Fei et al. ’03

Csurka ’04
Vasconcelos ’00

Bouchard & Triggs ’05

f) Hierarchy [4]
Carneiro & Lowe ’06

g) Sparse flexible model

from Sparse Flexible Models of Local Features
Gustavo Carneiro and David Lowe, ECCV 2006
Our first innovation involves enriching the Dalal-Triggs model using a star-structured part-based model defined by a “root” filter (analogous to the Dalal-Triggs filter) plus a set of parts filters and associated deformation models.
Star models by Latent SVM

Felzenszwalb, McAllester, Ramanan, 08
• Source code:
Mixture of components

“side” component

“frontal” component
Results on the PASCAL dataset
Person detection performances from 2005 to 2011

HOG templates

DPM

Courtesy of Girshick
SVM with mixtures of components

• Rather than training a single linear SVM separating positive examples…

• … cluster positive examples into “components” and train a classifier for each (using all negative examples)
Different connectivity structures

- **a)** Constellation [13]
  - \(O(N^6)\)
  - Fergus et al. ‘03
  - Fei-Fei et al. ‘03

- **b)** Star shape [9, 14]
  - \(O(N^2)\)
  - Crandall et al. ‘05
  - Leibe 05; Felzenszwalb 05

- **c)** k-fan (\(k = 2\)) [9]
  - \(O(N^3)\)
  - Crandall et al. ‘05
  - Felzenszwalb & Huttenlocher ‘00

- **d)** Tree [12]

- **e)** Bag of features [10, 21]
  - Csurka ‘04
  - Vasconcelos ‘00

- **f)** Hierarchy [4]
  - Bouchard & Triggs ‘05

- **g)** Sparse flexible model
  - Carneiro & Lowe ‘06

from Sparse Flexible Models of Local Features
Gustavo Carneiro and David Lowe, ECCV 2006
Implicit shape models by generalized Hough voting

B. Leibe, A. Leonardis, and B. Schiele,
Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004
Object representation: Constellation of parts w.r.t object centroid

B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004
Object representation:
How to capture constellation of parts?
Using Hough Voting
Hough transform


Given a set of points, find the curve or line that explains the data points best
Hough transform


Given a set of points, find the curve or line that explains the data points best

\[ y = m x + n \]

Hough space

\[ y_1 = m x_1 + n \]
Hough transform

$\begin{array}{cccccc}
3 & 5 & 3 & 3 & 2 & 2 \\
3 & 7 & 11 & 10 & 4 & 3 \\
2 & 3 & 1 & 4 & 5 & 2 \\
2 & 1 & 0 & 1 & 3 & 3 \\
\end{array}$

$\begin{array}{cccccccc}
\end{array}$
Hough transform


Use a polar representation for the parameter space

\[ x \cos \theta + y \sin \theta = \rho \]  

[Eq. 1]
Hough transform - experiments
Hough transform - experiments

Noisy data

features

votes

IDEA: introduce a grid a count intersection points in each cell
Issue: Grid size needs to be adjusted…
Generalized Hough Transform

- Parts in query image vote for a learnt model
- Significant aggregations of votes correspond to models
- Complexity: \( \# \text{ parts} \times \# \text{ votes} \)
  - Significantly lower than brute force search (e.g., sliding window detectors)
- Popular for detecting parameterized shapes
  - Hough’59, Duda&Hart’72, Ballard’81,...
Generalized Hough Transform

- GOAL: detect arbitrary shapes defined by boundary points and a reference point

Learning a model:

At each boundary point, compute displacement vector: \( \mathbf{r} = \mathbf{a} - \mathbf{p}_i \).  

For a given model shape: store these vectors in a table indexed by gradient orientation \( \theta \).

[Credit slide: C. Grauman]

[Dana H. Ballard, Generalizing the Hough Transform to Detect Arbitrary Shapes, 1980]
Example

Circle model

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>$rx$</th>
<th>$ry$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>90</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>135</td>
<td>-0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>270</td>
<td>0.7</td>
<td>-0.7</td>
</tr>
</tbody>
</table>
Generalized Hough Transform

Detecting the *model shape in a new image:*

- For each edge point
  - Index into table with its gradient orientation $\theta$
  - Use retrieved $r$ vectors to vote for position of reference point
- Peak in this Hough space is reference point with most supporting edges

*Assuming translation is the only transformation here, i.e., orientation and scale are fixed.*
Example

Circle model

Query

\[ \theta = 0 \rightarrow R = [rx, ry] = [1, 0] \rightarrow C_1 = P_1 + R \]
\[ \theta = 45 \rightarrow R = [rx, ry] = [0.7, 0.7] \rightarrow C_2 = P_2 + R \]
\[ \theta = -180 \rightarrow R = [rx, ry] = [-1, 0] \rightarrow C_k = P_k + R \]

<table>
<thead>
<tr>
<th>(\theta)</th>
<th>(rx)</th>
<th>(ry)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>90</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>135</td>
<td>-0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>270</td>
<td>0.7</td>
<td>-0.7</td>
</tr>
</tbody>
</table>
Conceptually similar to
Implicit shape models

• Instead of indexing displacements by gradient orientation, index by “visual codeword”

→ Visual codebook is used to index votes for object position \([\text{center}]\) and scale

B. Leibe, A. Leonardis, and B. Schiele, 
Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision, 2004
Implicit shape models

- Instead of indexing displacements by gradient orientation, index by “visual codeword”

→ Visual codebook is used to index votes for object position \([\text{center}]\) and scale

B. Leibe, A. Leonardis, and B. Schiele,
Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004
Implicit shape models

- Instead of indexing displacements by gradient orientation, index by “visual codeword”

→ Visual codebook is used to index votes for object position \([\text{center}]\) and scale

\[
\begin{array}{c|c|c}
\text{CW} & \text{rx} & \text{ry} \\
1 & 0.9 & 0.1 \\
3 & -1 & 0 \\
\ldots & \ldots & \ldots \\
N & 0.7 & -0.7 \\
\end{array}
\]
Scale Invariant Voting

- Scale-invariant feature selection
  - Scale-invariant interest points
  - Rescale extracted patches
  - Match to constant-size codebook

- Generate scale votes
  - Scale as $3^{rd}$ dimension in voting space
  - Search for maxima in 3D voting space

Source: Bastian Leibe
Implicit shape models: Training

1. Build codebook of patches around extracted interest points using clustering

Credit slide: S. Lazebnik
Implicit shape models: Training

1. Build codebook of patches around extracted interest points using clustering
2. Map the patch around each interest point to closest codebook entry
3. For each codebook entry, store all positions relative to object center [center is given] and scale [bounding box is given]

Credit slide: S. Lazebnik
Implicit Shape Model - Recognition

Interest Points → Matched Codebook Entries → Probabilistic Voting

3D Voting Space (continuous)

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]
Scale Voting: Efficient Computation

- Continuous Generalized Hough Transform
  - Binned accumulator array similar to standard Gen. Hough Transf.
  - Quickly identify candidate maxima locations
  - Refine locations by Mean-Shift search only around those points
  \[ \Rightarrow \text{Avoid quantization effects by keeping exact vote locations.} \]
  \[ \Rightarrow \text{Mean-shift interpretation as kernel prob. density estimation.} \]

Source: Bastian Leibe
Implicit Shape Model - Recognition

Interest Points → Matched Codebook Entries → Probabilistic Voting

Segmentation → Backprojected Hypotheses → Backprojection of Maxima

3D Voting Space (continuous)
Probabilistic Hough Transform

Detection Score

Position Posterior
distribution of the centroid
given the Codeword \( C_i \)
observed at location \( l_j \).

Codeword Match

confidence (or weight) of the
codeword \( C_i \).

\[
S(O, x) \propto \sum_{i,j} p(x', O, C_i, l_j', f_j)
\]

\[
\propto \sum_{i,j} p(x|O, C_i, l_j)p(C_i|f_j)p(O|C_i, l_j)
\]

\( f = \text{features} \)
\( l = \text{location of the features} \)
\( C = \text{codebook entry} \)
\( O = \text{object class} \)
\( x = \text{object center} \)

Learnt using a max margin formulation
Maji et al., CVPR 2009
Example: Results on Cows
Example: Results on Cows

Interest points
Example: Results on Cows

Matched patches
Example: Results on Cows

Prob. Votes
Example: Results on Cows

1st hypothesis
Example: Results on Cows

2^{nd} hypothesis
Example: Results on Cows

3rd hypothesis
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](http://www.vision.ee.ethz.ch/bleibe/code)
Extension: Learning Feature Weights

Subhransu Maji and Jitendra Malik, Object Detection Using a Max-Margin Hough Transform, CVPR 2009

Weights can be learned optimally using a max-margin framework.
Extension: Learning Feature Weights
Subhransu Maji and Jitendra Malik, Object Detection Using a Max-Margin Hough Transform, CVPR 2009

Naïve Bayes

Max-Margin

Important Parts

blue (low), dark red (high)
### Detection Results (ETHZ dataset)

<table>
<thead>
<tr>
<th>Category</th>
<th>Uniform</th>
<th>Naive Bayes</th>
<th>Max-margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applelogos</td>
<td>70.0</td>
<td>70.0</td>
<td>85.0</td>
</tr>
<tr>
<td>Bottles</td>
<td>62.5</td>
<td>71.4</td>
<td>67.0</td>
</tr>
<tr>
<td>Giraffes</td>
<td>47.1</td>
<td>47.1</td>
<td>55.0</td>
</tr>
<tr>
<td>Mugs</td>
<td>35.5</td>
<td>35.5</td>
<td>55.0</td>
</tr>
<tr>
<td>Swans</td>
<td>47.1</td>
<td>47.1</td>
<td>42.5</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>52.4</strong></td>
<td><strong>54.2</strong></td>
<td><strong>60.9</strong></td>
</tr>
</tbody>
</table>
Conclusions

• **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
  – No discriminative learning
Influential Works in Detection

  – Basic idea of statistical template detection, bootstrapping to get “face-like”
    negative examples, multiple whole-face prototypes (in 1994)
  – “Parts” at fixed position, non-maxima suppression, simple cascade, rotation, pretty
    good accuracy, fast
  – Careful feature engineering, excellent results, cascade
• Viola-Jones (2001, 2004) : ~11,000
  – Haar-like features, Adaboost as feature selection, hyper-cascade, very fast, easy to
    implement
• Dalal-Triggs (2005) : ~6500
  – Careful feature engineering, excellent results, HOG feature, online code
• Felzenszwalb-Huttenlocher (2000): ~2100
  – Efficient way to solve part-based detectors
• Weber et al. (2000)
  – Part-based model learnt in a unsupervised fashion; generative
• Felzenszwalb-McAllester-Ramanan (2008): ~1300
  – Excellent template/parts-based blend
• Leibe et al. (2005)
  – Generative approach to detection using hough voting

Courtesy of J Hayes
Next lecture

• 3D Object Detection