Lecture 12

Visual recognition

- Object Classification – BoW models (part 2)
- 2D Object Detection
  - Template based approaches
  - Part-based approaches
Object → Bag of ‘words’
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
2. Codewords dictionary formation

Cluster center = code word

Clustering/vector quantization

E.g., K-Means clustering
2. Codewords dictionary formation

Fei-Fei et al. 2005
3. Bag of word representation

Codewords dictionary

frequency

codewords
Invariance issues

• Scale? Rotation? View point? Occlusions?
  - Implicit
  - Depends on detectors and descriptors

Kadir and Brady. 2003
View invariant descriptors

View 1

View 2

Rectification

SIFT

Rectification

SIFT
Representation

1. feature detection & representation
2. codewords dictionary
3. image representation

recognition

learning

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

model space
Nearest neighbors classifiers

Query image

Winning class: pink

• Assign label of nearest training data point to each test data point
K-nearest neighbors classifiers

For a new query histogram, find the k closest points from training data
• Query histogram is classified by a majority vote of its K nearest neighbors
• Works well provided there is lots of data

Query image
Winning class: pink

model space
• Instead of modeling decision regions, it estimates a hyperplane $w$ that separates training points that belong to different classes.
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.
In training, we only need $w$ for classifying new data!
category models

Model space

Select two hyperplanes such that:
1. They separate the training points
2. There are no points between them
3. Their distance is maximized
Query image

Winning class: pink

SVM classifiers

Model space
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
How to design multi-class SVMs?

• A multi-class SVM can be obtained 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
Random classification: 0.01%
Major drawback of BOW models

Don’t capture spatial information!
Spatial Pyramid Matching


\[ H = [H_0^1, H_1^2, ..., H_4^2, H_1^3, ..., H_{16}^3] \]

or, \( H \) = combination of \( H_i^l \) with appropriate weights
Pyramid matching
Conclusions

• **Pros:**
  – Very simple and effective
  – Still used today for image retrieval problems

• **Cons:**
  – Not ideal for solving detection problems
  – Outperformed by CNNs but require way less training data
Discriminative models

Nearest neighbor

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

Support Vector Machine

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

Latent SVM

Felzenszwalb 2000
Ramanan 2003, 2008...

Structural SVM

Boosting

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

Neural networks

LeCun, Bottou, Bengio, Haffner 1998
Krizhevsky, Sutskever, Hinton, 2012

See lecture 15!

10^6 examples
Lecture 14
Visual recognition

- Object Classification – BoW models (part 2)
- 2D Object Detection
  - Template based approaches
  - Part-based approaches
Which object does this image contain? [where?]
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
Represent an object as a collection of HoG templates

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
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
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 more training data if view points need to be encoded
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
Deformations

A

B

C

D
Intraclass variation
Intraclass variation
Presence / Absence of Features

[Images of cars with occlusion marks]
Sparse representation

Computationally tractable ($10^5$ pixels $\rightarrow 10^1 - 10^2$ parts)

... but throw away potentially useful image information
Larger discrimination power

Parts can help fine-grained discrimination
History of parts-based 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
Different connectivity structures

- **a)** Constellation [13]
- **b)** Star shape [9, 14]
- **c)** $k$-fan ($k = 2$) [9]
- **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

**References**

- Fergus et al. '03
- Fei-Fei et al. '03
- Crandall et al. '05
- Leibe 05; Felzenszwalb 09
- Crandall et al. '05
- Felzenszwalb & Huttenlocher '00
- Bouchard & Triggs '05
- Carneiro & Lowe '06
- Csurka '04
- Vasconcelos '00

From Sparse Flexible Models of Local Features
Gustavo Carneiro and David Lowe, ECCV 2006
Deformable Part Models (DPM)


Score is computed by:
- comparing HOG features for each filter
- computing a “deformation” cost.

root filters (coarse)  
part filters (fine)  
deformation models
Deformable Part Models (DPM)

Mixture of components

SVM with mixtures of components

http://www.cs.berkeley.edu/~rbg/latent/index.html

Source code available!
Pascal Dataset

http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html

• 20 classes: aeroplane, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, train, TV

• Real images downloaded from flickr, not filtered for “quality”

• Complex scenes, scale, pose, lighting, occlusion, ...
Results on the PASCAL dataset

One of the most advanced versions of DPM achieves an average precision of 41%
Results on detecting deformable objects (humans)

HOG templates
Dalal & Triggs 2005

DPM
Felzenszwalb et al 2010

DPM by flexible Mixtures-of-Parts
Yang & Ramanan, 2012
Different connectivity structures

- a) Constellation [13]
- b) Star shape [9, 14]
- c) k-fan (k = 2) [9]
- 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

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
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

\[ y = mx + n \]

\[ y_1 = m x_1 + n \]

Hough space
Hough transform
Hough transform


Use a polar representation for the parameter space

\[ x \cos \theta + y \sin \theta = \rho \quad [\text{Eq. 1}] \]
IDEA: introduce a grid to 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

• Popular for detecting parameterized shapes
  – Hough’59, Duda&Hart’72, Ballard’81,...
For a given shape $S$, at each boundary point $p_i$ of $S$, compute displacement vector $r_i = a - p_i$ [Eq. 2]

Store these vector $r_i$ in a table indexed by the gradient orientation $\theta_i$ computed at $p_i$
Example: a circle

Learning a model:

For a given shape $S$, at each boundary point $p_i$ of $S$, compute displacement vector $r_i = a - p_i$ [Eq. 2]

Store these vector $r_i$ in a table indexed by the gradient orientation $\theta_i$ computed at $p_i$.

<table>
<thead>
<tr>
<th>p</th>
<th>$\theta$</th>
<th>$r_x$</th>
<th>$r_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Example: a circle

Learning a model:

For a given shape S, at each boundary point $p_i$ of S, compute displacement vector $r_i = a - p_i$ [Eq. 2]

Store these vector $r_i$ in a table indexed by the gradient orientation $\theta_i$ computed at $p_i$
Example: a circle

Learning a model:

For a given shape S, at each boundary point $p_i$ of S, compute displacement vector $r_i = a - p_i$ [Eq. 2]

Store these vector $r_i$ in a table indexed by the gradient orientation $\theta_i$ computed at $p_i$

<table>
<thead>
<tr>
<th>p</th>
<th>$\theta$</th>
<th>$r_x$</th>
<th>$r_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>45</td>
<td>0.7</td>
<td>0.7</td>
</tr>
</tbody>
</table>
Example: a circle

Learning a model:

For a given shape S, at each boundary point $p_i$ of S, compute displacement vector $r_i = a - p_i$ [Eq. 2]

Store these vector $r_i$ in a table indexed by the gradient orientation $\theta_i$ computed at $p_i$

<table>
<thead>
<tr>
<th></th>
<th>$\theta$</th>
<th>$r_x$</th>
<th>$r_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>45</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>3</td>
<td>90</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>135</td>
<td>-0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>8</td>
<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: a circle

Query

\[ Q_1 \rightarrow \theta = 0 \rightarrow R = [r_x,r_y] = [1,0] \rightarrow C_1 = Q_1 + R \]
\[ Q_2 \rightarrow \theta = 45 \rightarrow R = [r_x,r_y] = [.7,.7] \rightarrow C_2 = Q_2 + R \]
\[ Q_k \rightarrow \theta = -180 \rightarrow R = [r_x,r_y] = [-1,0] \rightarrow C_k = Q_k + R \]
\[ \vdots \]
Conceptually similar to
Implicit Shape Model
Captures constellation of parts using Hough Voting

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

- Enable detection without sliding windows!

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: Training

1. Build codebook of patches around extracted interest points using clustering (same as for BOW models)

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]

<table>
<thead>
<tr>
<th>CW</th>
<th>rx</th>
<th>ry</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.9</td>
<td>0.1</td>
</tr>
</tbody>
</table>
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]
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]

<table>
<thead>
<tr>
<th>CW</th>
<th>rx</th>
<th>ry</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>15</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>0.7</td>
<td>-0.7</td>
</tr>
</tbody>
</table>
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]
Implicit Shape Model - Recognition

Interest Points → Matched Codebook Entries → Probabilistic Voting

3D Voting Space (continuous)

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

Interest Points → Matched Codebook Entries → Probabilistic Voting

3D Voting Space (continuous)
Implicit Shape Model - Segmentation

Interest Points → Matched Codebook Entries → Probabilistic Voting

Segmentation → Backprojected Hypotheses → Backprojection of Maxima

3D Voting Space (continuous)
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

2nd 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

Many follow up projects:

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

... 

• Hough-based tracking of non-rigid objects, M Godec, PM Roth, H Bischof, 2013
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)

Recall @ 1.0 False Positives Per Window

<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>52.4</td>
<td>54.2</td>
<td><strong>60.9</strong></td>
</tr>
</tbody>
</table>
Conclusions

• Pros:
  – Works well for many different object categories
    • Both rigid and articulated objects
  – Doesn’t need a sliding window strategy
  – Learns from relatively few (50-100) training examples
  – Enables joint detection and segmentation
  – Still used today for 3D object detection problems

• Cons:
  – Requires supervision
  – Outperformed by CNN-based methods, but require way less training data!
Different connectivity structures

a) Constellation [13]

b) Star shape [9, 14]

c) $k$-fan ($k = 2$) [9]

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
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

 prediction of class

high-level parts

mid-level parts

low-level parts

Input image

(Lee et al., 2009)

More on lecture 15!
Next lecture

• 2D scene understanding
Appendix
Hierarchical representations

S.C. Zhu et al. and D. Mumford
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\textsuperscript{rd} dimension in voting space
  - Search for maxima in 3D voting space

Source: Bastian Leibe
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
  => Avoid quantization effects by keeping exact vote locations.
  => Mean-shift interpretation as kernel prob. density estimation.

Source: Bastian Leibe
Probabilistic Hough Transform

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

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

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 \( C_i \).

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

Learnt using a max margin formulation

Maji et al., CVPR 2009
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