Trainable visual models for object class recognition

Andrew Zisserman
University of Oxford

Slides from: Rob Fergus, Dan Huttenlocher, Bastian Leibe, Shimon Ullman
Objectives

• Recognition of visual object classes
• (semi) Unsupervised learning
Recognition

- Identify class (car, face, airplane etc)
- Determine approximate localization
  - multiple instances in a single image
- But not a perfect segmentation
(Semi) Unsupervised learning

- Know if image contains object or not
- But no segmentation of object or manual selection of features
Some object classes

Difficulties:
- Visual aspects
- Size variation
- Background clutter
- Partial occlusion
- Intra-class variation
Class of model: Pictorial Structure

- Fischler & Elschlager 1973

- Model has two components:
  1. parts (2D image fragments)
  2. structure (configuration of parts)

- Why this class of model?
Representation: Parts and Structure
Deformations

A

B

C

D
Presence / Absence of Features
Main issues:

• Parts
  • appearance, shape
• Structure
  • model (e.g. implicit or explicit)
• Model learning
  • from training data
• Model fitting (recognition)
  • complexity
Outline

1. Models that learn parts, then add structure
   - Weber, Welling & Perona, Leibe & Schiele, Agarwal & Roth, Borenstein & Ullman

2. Models for which the structure is primary
   - Felzenszwalb & Huttenlocher, Ramanan & Forsyth

3. Models that learn parts and structure simultaneously
   - Fergus, Perona & Zisserman

4. Summary and open challenges
   - Pascal Challenge: 101 Visual Object Classes
1. Models that learn parts, then add structure
Learning parts by clustering - 1

- Interest point features: textured neighborhoods are selected
- produces 100-1000 regions per image

Weber, Welling & Perona 2000
Learning parts by clustering - 2

“Pattern Space” (100+ dimensions)
Learning parts by clustering - 3

100-1000 images

~100 parts
Detecting part positions

• Detect interest point features
• Correlate parts with regions around detected points
• Candidate parts:
  – Best match at each interest point, or
  – Set of parts above similarity threshold
Leibe & Schiele 2003/2004

- Extraction of local object patches
  - Interest Points (Harris detector)

- Example: training set of 160 car images
  - 16 views of 10 cars
  - results in 8'269 training patches
Visual Vocabulary (Codebook Entries)

- Visual Clustering procedure
  - agglomerative clustering: most similar clusters are merged \((t > 0.7)\)
    \[
    \text{similarity}(C_1, C_2) = \frac{\sum_{p \in C_1, q \in C_2} NGC(p, q)}{|C_1| \times |C_2|} > t
    \]
    \[
    NGC(p, q) = \frac{\sum_i (p_i - \bar{p}_i)(q_i - \bar{q}_i)}{\sqrt{\sum_i (p_i - \bar{p}_i)^2 \sum_i (q_i - \bar{q}_i)^2}}
    \]

- Examples (from 2519 codebook entries)
  - visual similarity preserved
  - wheel parts, window corners, fenders, ...
Structure: Generalized Hough Transform

- **Learning:** For every cluster, store possible “occurrences”
  - Object Identity
  - Pose
  - Relative position

- **Recognition:** For new image, let the matched patches vote for possible object positions
Probabilistic Formulation

- 'Probabilistic Voting'

Learn probabilities from positive and negative examples.
Object Categorization Procedure

Interest Points → Matched Codebook Entries → Probabilistic Voting

Image Patch → Interpretation (Codebook match) → Object & Position

$e$ → $p(I_j|e)$ → $p(o_n, x|I_j)$ → $p(o_n, x|\bar{I_j})p(\bar{I_j}|e)$

Refined Hypothesis (uniform sampling) → Hypothesis → Projection of Maximum

$p(o_n, x|e) = \sum_j p(o_n, x|I_j)p(I_j|e)$
Detection Results

• Qualitative Performance
  - Recognizes different kinds of cars
  - Robust to clutter, occlusion, low contrast, noise
Agarwal & Roth 2002

• Interest points detected

• Extracted fragments from training images

• Clustered Fragments (Dictionary) – 270 parts
Learning: Structure

• Representation: binary feature vector
• Feature vector components
  – Part present/absent (270)
  – Pair wise relation between parts (20 of these for each pair)

Coarse representation of:
  • angles (4 bins)
  • distance (5 bins)

Use sliding window to measure feature vectors from positive and negative examples
Recognition

• Detect parts
• Apply sliding window
• Linear classifier on feature vector for window
• Use SNoW (Sparse network of Winnows)
  • suited to very large, very sparse vectors

Comparison with Leibe & Schiele

Agarawal & Roth:
• looser geometric relations
• more tolerant of structure deformation
Borenstein & Ullman 2002

- Training
- Learn fragments from segmented images
Class-based Recognition/Segmentation
Structure: jigsaw puzzle approach

1. Part matches image
2. Overlap of parts agree on foreground/background
3. Greedy algorithm for fitting

Comparison with Leibe & Schiele, Agarwal & Roth

Borenstein & Ullman:
• geometric constraints too loose
• often gets stuck on background regions
## Summary

<table>
<thead>
<tr>
<th>Parts</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leibe &amp; Schiele</td>
<td>Cluster from positive examples</td>
</tr>
<tr>
<td>Agarwal &amp; Roth</td>
<td>Cluster from positive examples</td>
</tr>
<tr>
<td>Borenstein &amp; Ullman</td>
<td>MI to select fragments from positive &amp; negative examples</td>
</tr>
</tbody>
</table>
So far ..... 

- Recognize class instances under image translation
- Implicit structure model
- No inter-part articulation
- Only single visual aspect

Extend to image scale change and rotation by exhaustive search over scale and orientation
Search over scale
2. Models for which the structure model is primary
New ideas

- Explicit structure model
- Articulated structure
Pictorial Structure Models for Object Recognition

Felzenszwalb & Huttenlocher 2000
Goal

- Detect and localize multi-part objects at arbitrary locations in a scene
  - Generic object models such as person or car
  - Allow for articulated objects
  - Combine 2D geometry and appearance
  - Provide efficient and practical algorithms
Matching Pictorial Structures

- Simultaneous use of appearance and spatial information

- Minimize an energy (or cost) function that reflects both
  - Appearance: how well each part matches at given location
  - Configuration: degree to which model is deformed in placing the parts at chosen locations
Example: Generic Person Model

- Each part represented as rectangle
  - Fixed width, varying length, uniform colour
  - Learn average and variation
    - Connections approximate revolute joints
  - Joint location, relative part position, orientation, foreshortening - Gaussian
  - Estimate average and variation

- Learned 10 part model
  - All parameters learned
    - Including “joint locations”
  - Shown at ideal configuration (mean locations)
Learning

- Manual identification of rectangular parts in a set of training images hypotheses
- Learn relative position \((x & y)\), relative angle, relative foreshortening
Recognition

- Given model $\Theta$ and image $I$, seek “good” configuration(s) $L$
  - Maximum a posteriori (MAP) estimate
    - Highest probability (lowest energy) configuration $L$
    - $L^*=\arg\max_L p(L|I,\Theta)$

- Brute force solutions intractable
  - With $p$ parts and $s$ possible discrete locations per part, $O(s^p)$

- If model is a tree then complexity reduces to $O(ps)$
Example: Recognizing People

NB: requires background subtraction
Variety of Poses
Variety of Poses
Pictorial structures for tracking
Learning articulated pictorial structures using temporal coherence

Ramanan & Forsyth 2003

- Parts detected as parallel lines of contrast
- Parts are clustered together.
- Stationary clusters are rejected.
Results
3. Models that learn parts and structure simultaneously
New ideas

- Explicit structure model – Joint Gaussian over all part positions
  - dates back to Weber, Welling & Perona 2000 and earlier
- Part detector determines position and scale
- Heterogeneous parts
- Simultaneous learning of parts and structure

Constellation model of Fergus, Perona & Zisserman 2003
Detect region for candidate parts

Use salient region operator (Kadir & Brady 01)
Representation of regions

- Find regions within image

**Location**

(x, y) coords. of region centre

**Scale**

Radius of region (pixels)

**Appearance (monochrome)**

- Normalize
- 11x11 patch
- Projection onto PCA basis

Gives representation of appearance in low-dimensional vector space

\[
\begin{pmatrix}
c_1 \\
c_2 \\
\vdots \\
c_{15}
\end{pmatrix}
\]
Generative probabilistic model

**Foreground model**
- Gaussian shape pdf
- Gaussian part appearance pdf
- Gaussian relative scale pdf

**Clutter model**
- Uniform shape pdf
- Gaussian background appearance pdf
- Uniform relative scale pdf
- Poisson pdf on # detections
Example – Learnt Motorbike Model

Samples from appearance model

Shape model
Learning

• Task: Estimation of model parameters

• Chicken and Egg type problem, since we initially know neither:
  - Model parameters
  - Assignment of regions to foreground / background

• Let the assignments be a hidden variable and use EM algorithm to learn them and the model parameters
Learning procedure

• Find regions & their location, scale & appearance over all training, compute PCA

• Initialize model parameters

• Use EM and iterate to convergence:
  
  E-step: Compute assignments for which regions are foreground / background
  M-step: Update model parameters

• Trying to maximize likelihood – consistency in shape & appearance
Recognition

- Detect regions in target image
- Evaluate the likelihood of the model (a search over assignments of parts to features)
- Threshold on the likelihood ratio
Experiments
Experimental procedure

Cal Tech Datasets

Training
- 50% images
- No identification of object within image

Testing
- 50% images
- Simple object present/absent test

Between 200 and 800 images in each dataset
Objects between 100 and 550 pixels in width
Recognized Motorbikes

position of object determined
Background images evaluated with motorbike model
Frontal faces
Airplanes
Spotted cats
Sampling from models

Faces

Motorbikes
## Comparison to other methods

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Ours</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motorbikes</td>
<td>7.5</td>
<td>16.0 Weber et al. [ECCV '00]</td>
</tr>
<tr>
<td>Faces</td>
<td>4.6</td>
<td>6.0 Weber</td>
</tr>
<tr>
<td>Airplanes</td>
<td>9.8</td>
<td>32.0 Weber</td>
</tr>
<tr>
<td>Cars (Side)</td>
<td>11.5</td>
<td>21.0 Agarwal Roth [ECCV '02]</td>
</tr>
</tbody>
</table>

% equal error rate

![Recall-Precision](chart.png)
“Brain damaged” Constellation model

- Learn on full model, but for recognition use only parts or structure probability term.
Constellation Model
Generalization 1:
Conditionally independent model
Shape model

+ Handle more detections per frame (N) - was ~25/image now 100’s/image

+ Handle more parts in model (P) - was 6, now 10-20

- Looser model: lack of inter-part covariance

- Anchor point cannot be occluded
Spotted Cats

• 6 part model
• Using average of 100 detections/frame
Constellation Model
Generalization 2:

Heterogeneous parts
Variety of feature types

- So far patch features using Kadir & Brady regions
- Other region operators (Multiscale Harris, Lowe etc.)
- Curve feature to capture outline of object
- Heterogeneous object models

Multiscale Harris interest point

Canny edge detection
Airplanes – Kadir & Brady operator
Airplanes – Curves
Airplanes – multi-scale Harris operator
Fitting the heterogeneous model

- Learn models with different combinations of Kadir & Brady, Multi-scale Harris, and curve parts

- Choose between models using a validation set

- For the experiments the image datasets are divided into the ratio:
  - 5/12 training
  - 1/6 validation
  - 5/12 testing

- 6 part independent models learnt
Motorbikes

Combination of patches and curves chosen
Motorbike Patch and Curve model
Motorbike results using curve and patch model
Spotted cats

Combination of Kadir & Brady and multi-scale Harris chosen
Spotted cats combination model
Spotted cats results using combination model
4. Summary and open challenges
• 😊 Single visual aspects (e.g. car rear/front)
  • Can learn from unsegmented images
  • Translation and scale invariance
  • Partial occlusion tolerated
  • Background clutter tolerated
  • Futures: greater viewpoint invariance:
  • scale invariant $\rightarrow$ similarity invariant $\rightarrow$ affine invariant

• 😞 Multiple visual aspects (e.g. car from any viewpoint)
  • Multiple 2D models ?
  • 3D models ?
Open Research Areas

- Part representation
  - e.g. Intensity (as here), or
  - orientation (Lowe, Carlsson)
- Structure model
  - tight parametric model (e.g. complete Gaussian)
  - loose model (e.g. pairwise relations)
- Comparison of models/methods on same data sets
Pascal Challenge: 101 Object Classes

- Organized by: Chris Williams, Andrew Zisserman and Luc Van Gool

- Levels of training difficulty:
  - Segmented training images
  - Images known to contain object class
  - Some of the images contain the object class

- Levels of visual difficulty
  - Intra-class variability (e.g. cars rear vs dogs)
  - Varying size and pose
  - Partial occlusion

- Standard test measures
Learning from contaminated data
Learning from contaminated data

- Image search engines give easy access to a vast amount of data.
- Just enter keyword (e.g. Camel)
- Large portion of images are junk (i.e. not instances of the class)
- Use raw output from Google Image search to train model

Fergus, Perona & Zisserman, ECCV 2004
Learning from contaminated data

Benign data sets (e.g. frontal faces):
  • model can use occlusion term to handle a certain level of junk

Google image sets:
  • foreground more varied and weak background model less valid

Approach: frame problem as one of robust estimation

Learning method: Hybrid RANSAC/EM
Robust line estimation - RANSAC

Fit a line to 2D data containing outliers

(RANdom Sample Consensus) [Fishler & Bolles, 1981]

There are two problems

1. a line fit which minimizes perpendicular distance
2. a classification into inliers (valid points) and outliers
RANSAC robust line estimation

- Repeat
  1. Select random sample of 2 points
  2. Compute the line through these points
  3. Measure support (number of points within threshold distance of the line)
- Choose the line with the largest number of inliers
- Compute least squares fit of line to inliers (regression)

Fitting to contaminated data

- Repeat
  1. Select random sample of images (say 10)
  2. Learn a model from these images
  3. Measure support of the model
- Choose the model with the largest number of inliers
RANSAC Scoring Function

Contaminated dataset

Background dataset (from Google ‘things’)
Camel curve model
Raw Camel images & 10 picked
Camel RPC curves

The graph shows the precision-recall curves for different methods:
- Red line: 10 images
- Dotted blue line: Raw Google
- Green dotted line: Unsupervised
Camel filtered results
Tiger filtered results
Bottles filtered results