Object Detection II

Ali Farhadi
CSE 576

Some slides from Derek Hoiem, Larry Zitnick, Ross Girchik
Image Categorization

Training

- Training Images
- Training Labels
- Image Features
- Classifier Training
- Trained Classifier

Testing

- Test Image
- Image Features
- Trained Classifier
- Prediction
  - Outdoor
Training set
Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05
0.16 = w^T x - b

\[ \text{sign}(0.16) = 1 \]

\[ \Rightarrow \text{pedestrian} \]
Detection examples
Each window is separately classified
Problem formulation

\{ airplane, bird, motorbike, person, sofa \}
Evaluating a detector

Test image (previously unseen)
First detection ...

'person' detector predictions
Second detection ...

‘person’ detector predictions
Third detection ...

☐  ‘person’ detector predictions
Compare to ground truth

'person' detector predictions

ground truth 'person' boxes
Sort by confidence

0.9 0.8 0.6 0.5 0.2 0.1
✓ X ✓ ✓ X X
true positive (high overlap)
false positive (no overlap, low overlap, or duplicate)
Evaluation metric

\[ \text{precision}@t = \frac{\# \text{true positives}@t}{\# \text{true positives}@t + \# \text{false positives}@t} \]

\[ \text{recall}@t = \frac{\# \text{true positives}@t}{\# \text{ground truth objects}} \]
Evaluation metric

Average Precision (AP)

mean AP over classes (mAP)
What about this one?

Can the model we trained for pedestrians detect the person in this image?
Specifying an object model

Statistical Template in Bounding Box

- Object is some \((x,y,w,h)\) in image
- Features defined \(\text{wrt bounding box coordinates}\)

Images from Felzenszwalb
When do statistical templates make sense?
Deformable objects

Images from Caltech-256
Deformable objects

Images from D. Ramanan’s dataset

Slide Credit: Duan Tran
Parts-based Models

Define objects by collection of parts modeled by

1. Appearance
2. Spatial configuration
How to model spatial relations?

• One extreme: fixed template
How to model spatial relations?

- Another extreme: bag of words
ISM: Implicit Shape Model

Training overview

• Start with bounding boxes and (ideally) segmentations of objects
• Extract local features (e.g., patches or SIFT) at interest points on objects
• Cluster features to create codebook
• Record relative bounding box and segmentation for each codeword
ISM: Implicit Shape Model

Testing overview

- Extract interest points in test image
- Softly match to codebook entries
- Each matched codeword votes for object bounding box
- Compute modes of votes using mean-shift
- Check which codewords voted for modes
- Refine
Example: Results on Cows
Example: Results on Cows
Example: Results on Cows
Example: Results on Cows
Example: Results on Cows
Example: Results on Cows
Example: Results on Cows
ISM: Detection Results

- Qualitative Performance
  - Robust to clutter, occlusion, noise, low contrast
Explicit Models

Hybrid template/parts model

Detections

Template Visualization

root filters
coarse resolution

part filters
finer resolution

deformation models

Felzenszwalb et al. 2008
How to model spatial relations?

- Explicit Models
- Too expensive
How to model spatial relations?

- Star-shaped model
How to model spatial relations?

• Star-shaped model
How to model spatial relations?

• Tree-shaped model
How to model spatial relations?

- Many others...

<table>
<thead>
<tr>
<th>a) Constellation</th>
<th>b) Star shape</th>
<th>c) $k$-fan ($k = 2$)</th>
<th>d) Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(N^6)$</td>
<td>$O(N^2)$</td>
<td>$O(N^3)$</td>
<td>$O(N^2)$</td>
</tr>
<tr>
<td>Fergus et al. ’03</td>
<td>Leibe et al. ’04, ’08</td>
<td>Crandall et al. ’05</td>
<td>Felzenszwalb &amp; Huttenlocher ‘05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>e) Bag of features</th>
<th>f) Hierarchy</th>
<th>g) Sparse flexible model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(N^6)$</td>
<td>$O(N^2)$</td>
<td>$O(N^2)$</td>
</tr>
<tr>
<td>Csurka ’04</td>
<td>Hierarchy</td>
<td>Carneiro &amp; Lowe ‘06</td>
</tr>
<tr>
<td>Vasconcelos ’00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

from [Carneiro & Lowe, ECCV’06]
Tree-shaped model
Pictorial Structures Model

Part = oriented rectangle  
Spatial model = relative size/orientation

Felzenszwalb and Huttenlocher 2005
Pictorial Structures Model

\[ P(L|I, \theta) \propto \left( \prod_{i=1}^{n} p(I|l_i, u_i) \prod_{(v_i, v_j) \in E} p(l_i, l_j|c_{ij}) \right) \]

Appearance likelihood

Geometry likelihood
Modeling the Appearance

- Any appearance model could be used
  - HOG Templates, etc.
  - Here: rectangles fit to background subtracted binary map

- Can train appearance models independently (easy, not as good) or jointly (more complicated but better)

\[ P(L|I, \theta) \propto \left( \prod_{i=1}^{n} p(I|l_i, u_i) \prod_{(v_i, v_j) \in E} p(l_i, l_j|c_{ij}) \right) \]

Appearance likelihood

Geometry likelihood
Part representation

• Background subtraction
Pictorial structures model

Optimization is tricky but can be efficient

\[ L^* = \arg \min_L \left( \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j) \in E} d_{ij}(l_i, l_j) \right) \]

- For each \( l_1 \), find best \( l_2 \):

\[ \text{Best}_2(l_1) = \min_{l_2} \left[ m_2(l_2) + d_{12}(l_1,l_2) \right] \]

- Remove \( v_2 \), and repeat with smaller tree, until only a single part

- For \( k \) parts, \( n \) locations per part, this has complexity of \( O(kn^2) \), but can be solved in \( \sim O(nk) \) using generalized distance transform
Pictorial Structures

- Model is represented by a graph $G = (V, E)$.
  - $V = \{v_1, \ldots, v_n\}$ are the parts.
  - $(v_i, v_j) \in E$ indicates a connection between parts.
- $m_i(l_i)$ is the cost of placing part $i$ at location $l_i$.
- $d_{ij}(l_i, l_j)$ is a deformation cost.
- Optimal location for object is given by $L^* = (l_1^*, \ldots, l_n^*)$,

\[
L^* = \arg\min_L \left( \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i, v_j) \in E} d_{ij}(l_i, l_j) \right)
\]
\[ L^* = \arg\min_L \left( \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j) \in E} d_{ij}(l_i, l_j) \right) \]

- \( n \) parts and \( h \) locations gives \( h^n \) configurations.
Complexity $O(h^n)$

$h$: number of possible part placements

$n$: number of parts
Efficient minimization

\[ L^* = \arg \min_L \left( \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j) \in E} d_{ij}(l_i,l_j) \right) \]

- \( n \) parts and \( h \) locations gives \( h^n \) configurations.
- If graph is a tree we can use dynamic programming.
  - \( O(nh^2) \), much better but still slow.
Complexity $O(nh^2)$
Efficient minimization

\[ L^* = \arg\min_L \left( \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j) \in E} d_{ij}(l_i,l_j) \right) \]

- \(n\) parts and \(h\) locations gives \(h^n\) configurations.

- If graph is a tree we can use dynamic programming.
  - \(O(nh^2)\), much better but still slow.

- If \(d_{ij}(l_i,l_j) = \|T_{ij}(l_i) - T_{ji}(l_j)\|^2\) can use DT.
  - \(O(nh)\), as good as matching each part separately!!
Distance transform

Given a set of points on a grid \( P \subseteq G \), the quadratic distance transform of \( P \) is,

\[
D_P(q) = \min_{p \in P} \| q - p \|^2
\]
Generalized distance transform

Given a function $f: \mathcal{G} \to \mathbb{R}$,

$$D_f(q) = \min_{p \in \mathcal{G}} \left( \|q - p\|^2 + f(p) \right)$$

— for each location $q$, find nearby location $p$ with $f(p)$ small.
1D case: \[ D_f(q) = \min_{p \in G} ( (q-p)^2 + f(p) ) \]

For each \( p \), \( D_f(q) \) is below the parabola rooted at \( (p, f(p)) \).
There is a simple geometric algorithm that computes $D_f(p)$ in $O(h)$ time for the 1D case.

- similar to Graham’s scan convex hull algorithm.
- about 20 lines of C code.

The 2D case is “separable”, it can be solved by sequential 1D transformations along rows and columns of the grid.

See *Distance Transforms of Sampled Functions*, Felzenszwalb and Huttenlocher.
Pictorial Structures: Summary

\[ L^* = \arg\min_L \left( \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j) \in E} d_{ij}(l_i, l_j) \right) \]

\[ d_{ij}(l_i, l_j) = \| T_{ij}(l_i) - T_{ji}(l_j) \|^2 \]
Results for person matching
Results for person matching
Enhanced pictorial structures
Deformable Latent Parts Model

Useful parts discovered during training

Detections

Template Visualization

root filters
coarse resolution

part filters
finer resolution

deformation models

Felzenszwalb et al. 2008
Deformable Part Models

\[
\text{Score} = F_0 \cdot \Phi(p_0, H) + \sum F_i \cdot \Phi(p_i, H) - \sum d_i \cdot \Phi_d(x, y) \\
\left( \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j) \in E} d_{ij}(l_i, l_j) \right)
\]
HOG Filters

- Array of weights for features in subwindow of HOG pyramid
- Score is dot product of filter and feature vector

Score of $F$ at position $p$ is $F \cdot \phi(p, H)$

$\phi(p, H) =$ concatenation of HOG features from subwindow specified by $p$
Object hypothesis

\[ z = (p_0, \ldots, p_n) \]

- \( p_0 \): location of root
- \( p_1, \ldots, p_n \): location of parts

Multiscale model captures features at two-resolutions
\[
\left( \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i, v_j) \in E} d_{ij}(l_i, l_j) \right)
\]
State-of-the-art Detector: Deformable Parts Model (DPM)

1. Strong low-level features based on HOG
2. Efficient matching algorithms for deformable part-based models (pictorial structures)
3. Discriminative learning with latent variables (latent SVM)

Person model

- Root filters
  - Coarse resolution
- Part filters
  - Finer resolution
- Deformation models
Person detections

high scoring true positives

high scoring false positives (not enough overlap)
Car

root filters
coarse resolution

part filters
finer resolution

deformation models
Car detections

high scoring true positives

high scoring false positives
Cat

root filters
coarse resolution

definitions

part filters
finer resolution

deformation
models
Cat detections

- high scoring true positives
- high scoring false positives (not enough overlap)
Person riding horse
Person riding bicycle
PASCAL VOC detection history

![Graph showing the history of mean average precision (mAP) from 2006 to 2015. The x-axis represents the years, and the y-axis represents the mAP percentage. The graph includes data points for different methods and years, indicating improvements over time.](image)
Part-based models & multiple features (MKL)

- DPM
- DPM, HOG+
- BOW
- DPM, MKL
- DPM++
- Selective Search
- DPM++, MKL

Mean Average Precision (mAP)

- 17%
- 23%
- 28%
- 37%
- 41%
- 41%

Year:
- 2006
- 2007
- 2008
- 2009
- 2010
- 2011
- 2012
- 2013
- 2014
- 2015

Rapid performance improvements
Kitchen-sink approaches

The diagram illustrates the mean Average Precision (mAP) over years from 2006 to 2015. It shows the increasing complexity and plateau in performance for various approaches, including:

- DPM
- DPM, HOG+, BOW
- DPM, MKL
- DPM++, MKL
- Selective Search, DPM++, MKL

The diagram highlights the percentage improvement in performance over the years.
Region-based Convolutional Networks (R-CNNs)

[Region-based Convolutional Networks (R-CNNs).
Girshick et al. CVPR 2014]
Region-based Convolutional Networks (R-CNNs)

~1 year
~5 years

[R-CNN. Girshick et al. CVPR 2014]
Convolutional Neural Networks

• Overview
Standard Neural Networks

\[
\mathbf{x} = (x_{\downarrow 1}, \ldots, x_{\downarrow 784})^T \quad z_{\downarrow j} = g (w_{\downarrow j}^T \mathbf{x}) \quad g(t) = \frac{1}{1 + e^{-t}}
\]
From NNs to Convolutional NNs

- Local connectivity
- Shared ("tied") weights
- Multiple feature maps
- Pooling
Convolutional NNs

• Local connectivity

• Each orange unit is only connected to (3) neighboring blue units
Convolutional NNs

• Shared ("tied") weights

All orange units share the same parameters

Each orange unit computes the same function but with a different input window
Convolutional NNs

• Convolution with 1-D filter: \([w \downarrow 3, w \downarrow 2, w \downarrow 1]\)

All orange units share the same parameters

Each orange unit computes the same function but with a different input window
Convolutional NNs

• Convolution with 1-D filter: $[w↓3, w↓2, w↓1]$
Convolutional NNs

• Convolution with 1-D filter: $[w\downarrow 3, w\downarrow 2, w\downarrow 1]$

All orange units share the same parameters

Each orange unit computes the same function but with a different input window
Convolutional NNs

• Convolution with 1-D filter: \([w \downarrow 3, w \downarrow 2, w \downarrow 1]\)

All orange units share the same parameters

Each orange unit computes the same function but with a different input window
Convolu(tional NNs

• Convolution with 1-D filter: \([w\downarrow 3, w\downarrow 2, w\downarrow 1]\)

All orange units share the same parameters

Each orange unit computes the same function but with a different input window
Convolutional NNs

• Multiple feature maps

Feature map 1 (array of green units)

Feature map 2 (array of orange units)

All orange units share the same parameters
Each orange unit computes the same function but with a different input window
Convolutional NNs

• Pooling \textit{(max, average)}

- Pooling area: 2 units
- Pooling stride: 2 units
- \textit{Subsamples} feature maps
2D input

Pooling

Convolution

Image
Practical ConvNets

Gradient-Based Learning Applied to Document Recognition,
Lecun et al., 1998
Demo

• [http://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html](http://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html)

• ConvNetJS by Andrej Karpathy (Ph.D. student at Stanford)

Software libraries
• Caffe (C++, python, matlab)
• Torch7 (C++, lua)
• Theano (python)
Core idea of “deep learning”

• Input: the “raw” signal (image, waveform, ...)

• Features: hierarchy of features is learned from the raw input
Structure
Structured Prediction

• Prediction of complex outputs
  – Structured outputs: multivariate, correlated, constrained

• Novel, general way to solve many learning problems
Handwriting Recognition

Sequential structure
Object Segmentation

x

Spatial structure

y
Local Prediction

Classify using local information
⇒ Ignores correlations & constraints!
Local Prediction

- building
- tree
- shrub
- ground
Structured Prediction

• Use local information
• Exploit correlations
Structured Prediction

- building
- tree
- shrub
- ground
Structured Models

\[ h(x) = \arg \max_{y \in \mathcal{Y}(x)} \text{score}(x, y) \]

space of feasible outputs

Mild assumptions:

\[ \text{score}(x, y) = w^\top f(x, y) = \sum_p w^\top f(x_p, y_p) \]

linear combination

sum of part scores
Supervised Structured Prediction

Model: \( P_w(y \mid x) \propto \exp\{w^\top f(x, y)\} \)

Data: \((x^1, y^1), \ldots, (x^n, y^n)\)

Learning: Estimate \(w\)

Prediction: \(\arg\max_{y \in \mathcal{Y}(x)} P_w(y \mid x)\)

Example:
- Weighted matching

Generally:
- Combinatorial optimization

Local (ignores structure)

Margin

Likelihood (can be intractable)
Local Estimation

**Model:**

\[ P_w(y \mid x) \propto \prod_{j<k} \exp\{w^\top f(y_{jk}, x)\} \]

- Treat edges as independent decisions
- Estimate \( w \) locally, use globally
  - E.g., naïve Bayes, SVM, logistic regression
  - Cf. [Matusov+al, 03] for matchings
  - Simple and cheap
  - Not well-calibrated for matching model
  - Ignores correlations & constraints
Conditional Likelihood Estimation

Model: \[
P_w(y \mid x) = \frac{\prod_{jk} \exp\{w^\top f(y_{jk}, x)\}}{\sum_{y' \in \mathcal{Y}(x)} \prod_{jk} \exp\{w^\top f(y'_{jk}, x)\}}
\]

- Estimate \( w \) jointly:

\[
\sum_i \log P_w(y^i \mid x^i)
\]

- Denominator is \#P-complete
  
  [Valiant 79, Jerrum & Sinclair 93]

- Tractable model, intractable learning

- Need tractable learning method
  
  \( \Rightarrow \) margin-based estimation
Structured large margin estimation

• We want:

$$\arg \max_y w^T f(\text{brace}, y) = \text{“brace”}$$

• Equivalently:

$$w^T f(\text{brace}, \text{“brace”}) > w^T f(\text{brace}, \text{“aaaaa”})$$
$$w^T f(\text{brace}, \text{“brace”}) > w^T f(\text{brace}, \text{“aaaab”})$$
$$\ldots$$
$$w^T f(\text{brace}, \text{“brace”}) > w^T f(\text{brace}, \text{“zzzzz”})$$

a lot!
Structured Loss

b a x e
b r o x e
b r o c e
b r a c e

2 2 1 0
Large margin estimation

• Given training examples \((x^i, y^i)\), we want:

\[ w^\top f(x^i, y^i) > w^\top f(x^i, y) \quad \forall y \neq y^i \]

- Maximize margin \(\gamma\)

\[ w^\top f(x^i, y^i) \geq w^\top f(x^i, y) + \gamma \ell(y^i, y) \quad \forall y \]

- Mistake weighted margin:

\[ \gamma \ell(y^i, y) \]

\[ \ell(y^i, y) = \sum_p I(y^i_p \neq y_p) \quad \# \text{ of mistakes in } y \]

*Collins 02, Altun et al 03, Taskar 03*
Large margin estimation

\[
\begin{align*}
\max_{\|w\| \leq 1} & \quad \gamma \\
\text{s.t.} & \quad w^\top f(x^i, y^i) \geq w^\top f(x^i, y) + \gamma \ell(y^i, y), \quad \forall i, y
\end{align*}
\]

**Eliminate** \( \gamma \)

\[
\begin{align*}
\min_w & \quad \frac{1}{2} ||w||^2 \\
\text{s.t.} & \quad w^\top f(x^i, y^i) \geq w^\top f(x^i, y) + \ell(y^i, y), \quad \forall i, y
\end{align*}
\]

**Add slacks** \( \xi_i \) for inseparable case (hinge loss)

\[
\begin{align*}
\min_{w, \xi} & \quad \frac{1}{2} ||w||^2 + C \sum_i \xi_i \\
\text{s.t.} & \quad w^\top f(x^i, y^i) + \xi_i \geq w^\top f(x^i, y) + \ell(y^i, y), \quad \forall i, y
\end{align*}
\]
Large margin estimation

• Brute force enumeration

$$\min \quad \frac{1}{2} \|w\|^2 + C \sum_i \xi_i$$

$$w^T f(x^i, y^i) + \xi_i \geq w^T f(x^i, y) + \ell(y^i, y), \quad \forall i, y$$

• Min-max formulation

$$\min \quad \frac{1}{2} \|w\|^2 + C \sum_i \xi_i$$

$$w^T f(x^i, y^i) + \xi_i \geq \max_y [w^T f(x^i, y) + \ell(y^i, y)], \quad \forall i$$

– ‘Plug-in’ linear program for inference

$$\max_y [w^T f(x^i, y) + \ell(y^i, y)]$$
Min-max formulation

\[
\max_y \left[ w^\top f(x^i, y) + \ell(y^i, y) \right]
\]

Structured loss (Hamming):

\[
\ell(y^i, y) = \sum_p \ell_p(y^i_p, y_p)
\]

Inference

\[
\max_y \left[ \sum_p w^\top f(x^i_p, y_p) + \ell_p(y^i_p, y_p) \right]
\]

LP Inference

\[
\max_{q \geq 0; Aq \leq b} \quad q^\top z
\]

Key step: \max_y \quad \text{discrete optim.} \quad \leftrightarrow \quad \max_z \quad \text{continuous optim.}
Matching Inference LP

\[
\max_y \quad w^\top f(x^i, y) + \ell(y^i, y)
\]

\[
\max_z \quad \sum_{jk} z_{jk} \left[ w^\top f(x_{jk}^i) + \ell_{jk}^i \right]
\]

\[
\begin{align*}
q^\top z \quad & q = F^\top w + \ell \\
\text{s.t.} \quad & z_{jk} \geq 0 \\
& \sum_k z_{jk} \leq 1 \\
& \sum_j z_{jk} \leq 1 \\
\end{align*}
\]

\[
Az \leq b
\]
LP Duality

• Linear programming duality
  – Variables ⇒ constraints
  – Constraints ⇒ variables

• Optimal values are the same
  – When both feasible regions are bounded

\[
\begin{align*}
\text{max} \quad & c^T z \\
\text{s.t.} \quad & A z \leq b; \\
& z \geq 0.
\end{align*}
\]

\[
\begin{align*}
\text{min} \quad & b^T \lambda \\
\text{s.t.} \quad & A^T \lambda \geq c; \\
& \lambda \geq 0.
\end{align*}
\]
Min-max Formulation

\[
\begin{align*}
\min_{w, \xi} & \quad \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \\
\text{s.t.} & \quad w^\top f(x^i, y^i) + \xi_i \geq \max_y [w^\top f(x^i, y) + \ell(y^i, y)], \quad \forall i
\end{align*}
\]

\[q_i = F_i^\top w + \ell_i\]

LP duality

\[
\begin{align*}
\max & \quad q_i^\top z_i \\
\text{s.t.} & \quad A_i z_i \leq b_i \quad z_i \geq 0 \\
\min & \quad b_i^\top \lambda_i \\
\text{s.t.} & \quad A_i^\top \lambda_i \geq q_i \quad \lambda_i \geq 0
\end{align*}
\]
Min-max formulation summary

\[
\min_{w, \lambda} \quad \frac{1}{2} \|w\|^2 + C \left( \sum_i b_i^T \lambda_i - w^T f(x^i, y^i) \right)
\]

s.t. \quad A_i^T \lambda_i \geq F_i^T w + \ell_i; \quad \lambda_i \geq 0, \ \forall i.

*Taskar et al 04*
3D Mapping

Data provided by: Michael Montemerlo & Sebastian Thrun

Label: ground, building, tree, shrub
Training: 30 thousand points   Testing: 3 million points
Before and After

Before Decoding

After Decoding

Recognition using Visual Phrases, CVPR 2011
Before and After

Before Decoding

After Decoding