Category-level localization

Josef Sivic
http://www.di.ens.fr/~josef

Slides from Andrew Zisserman
Visual Recognition and Machine Learning Summer School, 2010-2012
http://www.di.ens.fr/willow/events/cvml2012/

Includes slides from: Ondra Chum, Alyosha Efros, Mark Everingham, Pedro Felzenszwalb, Rob Fergus, Kristen Grauman, Bastian Leibe, Ivan Laptev, Fei-Fei Li, Marcin Marszalek, Pietro Perona, Deva Ramanan, Bernt Schiele, Jamie Shotton, Andrea Vedaldi
Announcements

• Assignment 1 was due last week. **Have you sent it?**
  Please check the table with received assignments on the class webpage.

• Assignment 2 was out last week. **Any questions?**

• Topic ideas for the final projects will be out this week:
# Assignment 1 – received reports

<table>
<thead>
<tr>
<th>RecVis12</th>
<th>Received Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key:</strong></td>
<td><strong>R</strong>=received, <strong>L</strong>=late ((\leq3)days), <strong>VL</strong>=very late ((&gt;3)days)</td>
</tr>
<tr>
<td><strong>Student name</strong></td>
<td><strong>Email</strong></td>
</tr>
<tr>
<td>BIENVENU Alexis</td>
<td><a href="mailto:alexis.bienvenu@eleves.enpc.fr">alexis.bienvenu@eleves.enpc.fr</a></td>
</tr>
<tr>
<td>BELGHITI Ismael</td>
<td><a href="mailto:isma.belghiti@gmail.com">isma.belghiti@gmail.com</a></td>
</tr>
<tr>
<td>COLONNA Andréa</td>
<td><a href="mailto:colonnafinance@gmail.com">colonnafinance@gmail.com</a></td>
</tr>
<tr>
<td>CIRSTEA Bogdan</td>
<td><a href="mailto:cirstea.bogdanionut@yahoo.com">cirstea.bogdanionut@yahoo.com</a></td>
</tr>
<tr>
<td>DARDARD Floriane</td>
<td><a href="mailto:floriane.dardard@ens.fr">floriane.dardard@ens.fr</a></td>
</tr>
<tr>
<td>HEDOUIN Renaud</td>
<td><a href="mailto:renaud.hedouin@gmail.com">renaud.hedouin@gmail.com</a></td>
</tr>
<tr>
<td>HUYNH Olivier</td>
<td><a href="mailto:olivier.huynh@mines-paristech.fr">olivier.huynh@mines-paristech.fr</a></td>
</tr>
<tr>
<td>KUMAR KARRI Senanayak Sesh</td>
<td><a href="mailto:seshkumar@gmail.com">seshkumar@gmail.com</a></td>
</tr>
<tr>
<td>LE GUEN Vincent</td>
<td><a href="mailto:vincent.le-guen@telecom-paristech.fr">vincent.le-guen@telecom-paristech.fr</a></td>
</tr>
<tr>
<td>MOISY-MABILLE Kévin</td>
<td><a href="mailto:kevin.moisy-mabille@dptinfo.ens-cachan.fr">kevin.moisy-mabille@dptinfo.ens-cachan.fr</a></td>
</tr>
<tr>
<td>OQUAB Maxime</td>
<td><a href="mailto:oquabm@eleves.enpc.fr">oquabm@eleves.enpc.fr</a></td>
</tr>
<tr>
<td>OYALLON Edouard</td>
<td><a href="mailto:edouard.oyallon@ens-cachan.fr">edouard.oyallon@ens-cachan.fr</a></td>
</tr>
<tr>
<td>PUMIR Thomas</td>
<td><a href="mailto:pumir.thomas@gmail.com">pumir.thomas@gmail.com</a></td>
</tr>
<tr>
<td>RERRONNET Lorraine</td>
<td><a href="mailto:glowable@gmail.com">glowable@gmail.com</a></td>
</tr>
<tr>
<td>REZENDE Rafael</td>
<td><a href="mailto:rafael.sampaio-de-rezende@polytechnique.org">rafael.sampaio-de-rezende@polytechnique.org</a></td>
</tr>
<tr>
<td>RICAUD Bruno</td>
<td><a href="mailto:bruno.ricaud@mines-paristech.fr">bruno.ricaud@mines-paristech.fr</a></td>
</tr>
<tr>
<td>SAHIN Aytunc</td>
<td><a href="mailto:aytunc.sahan@polytechnique.edu">aytunc.sahan@polytechnique.edu</a></td>
</tr>
<tr>
<td>SPISIAK Michal</td>
<td><a href="mailto:michal.spisiak@gmail.com">michal.spisiak@gmail.com</a></td>
</tr>
<tr>
<td>TERRAZZONI Frédéric</td>
<td><a href="mailto:frederic.terrazzoni@gmail.com">frederic.terrazzoni@gmail.com</a></td>
</tr>
<tr>
<td>VU Tuan Hung</td>
<td><a href="mailto:tuan-hung.vu@telecom-paristech.fr">tuan-hung.vu@telecom-paristech.fr</a></td>
</tr>
<tr>
<td>YAO Tao Jin</td>
<td><a href="mailto:tao-jin.yao@mines-paristech.fr">tao-jin.yao@mines-paristech.fr</a></td>
</tr>
</tbody>
</table>
Final project presentations (details later)

• 1\textsuperscript{st} batch during class on Tuesday Dec 11 (16:15-19:15)

• 2\textsuperscript{nd} batch either on:
(a) Wednesday Dec 12 (2pm-6pm)
or(b) Thursday Dec 13 (2pm-6pm)

Which one would you prefer?
What we would like to be able to do…

• Visual scene understanding
• **What** is in the image and **where**

- Dog 1: Terrier
- Dog 2: Sitting on Motorbike
- Person: John Smith, holding Dog 2
- Motorbike: Suzuki GSX 750
- Ground: Gravel
- Gate
- Plant
- Wall

• Object categories, identities, properties, activities, relations, …
Recognition Tasks

• **Image Classification**
  – Does the image contain an aeroplane? (last lecture, assignment 2)

• **Object Class Detection/Localization**
  – Where are the aeroplanes (if any)?

• **Object Class Segmentation**
  – Which pixels are part of an aeroplane (if any)?
**Things vs. Stuff**

**Thing** (n): An object with a specific size and shape.

**Stuff** (n): Material defined by a homogeneous or repetitive pattern of fine-scale properties, but has no specific or distinctive spatial extent or shape.

Ted Adelson, Forsyth et al. 1996.

Slide: Geremy Heitz
Recognition Task

• **Object Class Detection/Localization**
  – Where are the aeroplanes (if any)?

• **Challenges**
  – Imaging factors e.g. lighting, pose, occlusion, clutter
  – Intra-class variation

• **Compared to Classification**
  – Detailed prediction e.g. bounding box
  – Location usually provided for training
Challenges: Scale
Challenges: Background Clutter
Challenges: Occlusion and truncation
Challenges: Intra-class variation
Object Category Recognition by Learning

- Difficult to define model of a category. Instead, **learn** from example images
Level of Supervision for Learning

- Image-level label
- Bounding box
- Pixel-level segmentation
- "Parts"
Preview of typical results

- aeroplane
- bicycle
- car
- cow
- horse
- motorbike
Class of model: Pictorial Structure

- Intuitive model of an object
- Model has two components
  1. parts (2D image fragments)
  2. structure (configuration of parts)
- Dates back to Fischler & Elschlager 1973

Is this complexity of representation necessary?
Which features?
Restrict deformations
Problem of background clutter

- Use a sub-window
  - At correct position, no clutter is present
  - Slide window to detect object
  - Change size of window to search over scale
Outline

1. Sliding window detectors

2. Features and adding spatial information

3. Histogram of Oriented Gradients (HOG)

4. Two state of the art algorithms and PASCAL VOC

5. The future and challenges
Outline

1. Sliding window detectors
   - Start: feature/classifier agnostic
   - Method
   - Problems/limitations

2. Features and adding spatial information

3. Histogram of Oriented Gradients (HOG)

4. Two state of the art algorithms and PASCAL VOC

5. The future and challenges
Detection by Classification

• Basic component: binary classifier
Detection by Classification

• Detect objects in clutter by **search**

• **Sliding window**: exhaustive search over position and scale
Detection by Classification

- Detect objects in clutter by search

- Sliding window: exhaustive search over position and scale
Detection by Classification

- Detect objects in clutter by **search**

- **Sliding window**: exhaustive search over position and scale (can use same size window over a spatial pyramid of images)
Window (Image) Classification

- Features usually engineered
- Classifier learnt from data

Training Data

Feature Extraction $\rightarrow$ Classifier $F(x)$

$P(c|x) \propto F(x)$
Problems with sliding windows …

• aspect ratio
• granularity (finite grid)
• partial occlusion
• multiple responses

See recent work by

• Christoph Lampert et al CVPR 08, ECCV 08
Outline

1. Sliding window detectors
2. Features and adding spatial information
   • Bag of visual word (BoW) models
   • Beyond BoW I: Constellation and ISM models
   • Beyond BoW II: Grids and spatial pyramids
3. Histogram of Oriented Gradients (HOG)
4. Two state of the art algorithms and PASCAL VOC
5. The future and challenges
Recap: Bag of (visual) Words representation

• Detect affine invariant local features (e.g. affine-Harris)

• Represent by high-dimensional descriptors, e.g. 128-D for SIFT

• How to summarize sliding window content in a fixed-length vector for classification?

1. Map descriptors onto a common vocabulary of **visual words**

2. Represent image as a histogram over visual words – a **bag of words**
Local region descriptors and visual words

- Normalize regions to fixed size and shape
- Describe each region by a SIFT descriptor
- Vector quantize into visual words, e.g. using k-means

NB: aff. detectors/SIFT/visual words originally for view point invariant matching
Visual Words

Cluster = Visual Word

Local Descriptors

Vector Quantize (K-means)
Example Visual Words
Intuition

- Visual words represent “iconic” image fragments
- Feature detectors and SIFT give invariance to local rotation and scale
- Discarding spatial information gives configuration invariance
Learning from positive ROI examples

Bag of Words

Feature Vector
Sliding window detector

- Classifier: SVM with linear kernel
- BOW representation for ROI

Example detections for dog

Lampert et al CVPR 08
Discussion: ROI as a Bag of Visual Words

• Advantages
  – No explicit modelling of spatial information -> high level of invariance to position and orientation in image
  – Fixed length vector -> standard machine learning methods applicable

• Disadvantages
  – No explicit modelling of spatial information -> less discriminative power
  – Inferior to state of the art performance
Beyond BOW I: Pictorial Structure

- Intuitive model of an object
- Model has two components
  1. parts (2D image fragments)
  2. structure (configuration of parts)
- Dates back to Fischler & Elschlager 1973

Two approaches that have investigated this spring like model:

- Constellation model
- Implicit shape model
Spatial Models Considered

1. **Fully connected shape model**
   - e.g. Constellation Model
   - Parts fully connected
   - Recognition complexity: $O(N^P)$
   - Method: Exhaustive search

2. **“Star” shape model**
   - e.g. ISM
   - Parts mutually independent
   - Recognition complexity: $O(NP)$
   - Method: Gen. Hough Transform

Slide credit: Rob Fergus
Constellation model

- Explicit structure model – Joint Gaussian over all part positions
- Part detector determines position and scale
- Simultaneous learning of parts and structure
- Learn from images alone using EM algorithm

Given detections: learn a six part model by optimizing part and configuration similarity

Fergus, Perona & Zisserman, CVPR 03
Example – Learnt Motorbike Model

Samples from appearance model

Shape model
Recognized Motorbikes

position of object determined
Airplanes
Discussion: Constellation Model

• Advantages
  – Works well for many different object categories
  – Can adapt well to categories where
    • Shape is more important
    • Appearance is more important
  – Everything is learned from training data
  – Weakly-supervised training possible

• Disadvantages
  – Model contains many parameters that need to be estimated
  – Cost increases exponentially with increasing number of parameters
    ⇒ Fully connected model restricted to small number of parts.
Implicit Shape Model (ISM)

• Basic ideas
  – Learn an appearance codebook
  – Learn a star-topology structural model
    • Features are considered independent given object centre

• Algorithm: probabilistic Generalized Hough Transform
  Good engineering:
  – Soft assignment
  – Probabilistic voting
  – Continuous Hough space
Codebook Representation

• Extraction of local object features
  – Interest Points (e.g. Harris detector)
  – Sparse representation of the object appearance

• Collect features from whole training set

• Example:

Class specific vocabulary
Leibe & Schiele 03/04: Generalized Hough Transform

- **Learning:** for every cluster, store possible “occurrences”

- **Recognition:** for new image, let the matched patches vote for possible object positions
Leibe & Schiele 03/04: Generalized Hough Transform
Scale Voting: Efficient Computation

- Mean-Shift formulation for refinement
  - Scale-adaptive *balloon density estimator*

\[
\hat{p}(o_n, x) = \frac{1}{V_b} \sum_k \sum_j p(o_n, x_j | f_k, \ell_k) K\left(\frac{x - x_j}{b}\right)
\]
Detection Results

- Qualitative Performance
  - Recognizes different kinds of cars
  - Robust to clutter, occlusion, low contrast, noise
Discussion: ISM and related models

Advantages

- Scale and rotation invariance can be built into the representation from the start
- Relatively cheap to learn and test (inference)
- Works well for many different object categories
- Max-margin extensions possible, Maji & Malik, CVPR09

Disadvantages

- Requires searching for modes in the Hough space
- Similar to sliding window in this respect
- Is such a degree of invariance required? (many objects are horizontal)
Beyond BOW II: Grids and spatial pyramids

Start from BoW for ROI

- no spatial information recorded
- sliding window detector
Adding Spatial Information to Bag of Words

Keeps fixed length feature vector for a window

[Concatenate]

[Feature Vector]

[FERGUS ET AL, 2005]
Tiling defines (records) the spatial correspondence of the words

- parameter: number of tiles

If codebook has V visual words, then representation has dimension 4V

Fergus et al ICCV 05
Spatial Pyramid – represent correspondence

- As in scene/image classification can use pyramid kernel

[Grauman & Darrell, 2005]  [Lazebnik et al, 2006]
Dense Visual Words

• Why extract only \textit{sparse} image fragments?

• Good where lots of invariance is needed, but not relevant to sliding window detection?

• Extract \textit{dense} visual words on an overlapping grid

• More “detail” at the expense of invariance
• Pyramid histogram of visual words (PHOW)

[Luong & Malik, 1999]
[Varma & Zisserman, 2003]
[Vogel & Schiele, 2004]
[Jurie & Triggs, 2005]
[Fei-Fei & Perona, 2005]
[Bosch et al, 2006]
Outline

1. Sliding window detectors
2. Features and adding spatial information
3. Histogram of Oriented Gradients + linear SVM classifier
   - Dalal & Triggs pedestrian detector
   - HOG and history
   - Training an object detector
4. Two state of the art algorithms and PASCAL VOC
5. The future and challenges
Dalal & Triggs CVPR 2005 Pedestrian detection

- Objective: detect (localize) standing humans in an image
- Sliding window classifier
- Train a binary classifier on whether a window contains a standing person or not
- Histogram of Oriented Gradients (HOG) feature
- Although HOG + SVM originally introduced for pedestrians has been used very successfully for many object categories
Feature: Histogram of Oriented Gradients (HOG)

- tile 64 x 128 pixel window into 8 x 8 pixel cells
- each cell represented by histogram over 8 orientation bins (i.e. angles in range 0-180 degrees)
Histogram of Oriented Gradients (HOG) continued

• Adds a second level of overlapping spatial bins re-normalizing orientation histograms over a larger spatial area

• Feature vector dimension (approx) = 16 x 8 (for tiling) x 8 (orientations) x 4 (for blocks) = 4096
Window (Image) Classification

- HOG Features
- Linear SVM classifier

Training Data

Feature Extraction

Classifier

\[ F(x) \]

\[ P(c|x) \propto F(x) \]

x

pedestrian/Non-pedestrian
Averaged examples
Advantages of linear SVM:

$\mathbf{f}(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$

• Training (Learning)
  - Very efficient packages for the linear case, e.g. LIBLINEAR for batch training and Pegasos for on-line training.
  - Complexity $O(N)$ for $N$ training points (cf $O(N^3)$ for general SVM)

• Testing (Detection)

  Non-linear
  $$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{S} \alpha_i \mathbf{k}(\mathbf{x}_i, \mathbf{x}) + b$$
  
  ```
  S = # of support vectors
  = (worst case ) N
  size of training data
  ```

  linear
  $$\mathbf{f}(\mathbf{x}) = \sum_{i}^{S} \alpha_i \mathbf{x}_i^T \mathbf{x} + b$$
  
  $$= \mathbf{w}^T \mathbf{x} + b$$
  Independent of size of training data
Review: Binary classification

Given training data \((x_i, y_i)\) for \(i = 1 \ldots N\), with \(x_i \in \mathbb{R}^d\) and \(y_i \in \{-1, 1\}\), learn a classifier \(f(x)\) such that

\[
f(x_i) \begin{cases} 
\geq 0 & y_i = +1 \\
< 0 & y_i = -1
\end{cases}
\]

i.e. \(y_if(x_i) > 0\) for a correct classification.
Review: Linear classifiers

A linear classifier has the form

\[ f(x) = w^\top x + b \]

- in 2D the discriminant is a line
- \( w \) is the normal to the plane, and \( b \) the bias
- \( w \) is known as the weight vector
Review: Linear classifiers

A linear classifier has the form

\[ f(x) = w^\top x + b \]

- in 3D the discriminant is a plane, and in nD it is a hyperplane
Review: Linear classifiers

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

\[ x_i \text{ positive: } x_i \cdot w + b \geq 0 \]
\[ x_i \text{ negative: } x_i \cdot w + b < 0 \]

Which hyperplane is best?
Review: Linear classifiers - margin

- Generalization is not good in this case:
- Better if a margin is introduced:
Support vector machines

• Find a hyperplane that maximizes the _margin_ between positive and negative examples

\[
\begin{align*}
x_i \text{ positive } (y_i = 1) &: \quad x_i \cdot w + b \geq 1 \\
x_i \text{ negative } (y_i = -1) &: \quad x_i \cdot w + b \leq -1
\end{align*}
\]

For support vectors, \( x_i \cdot w + b = \pm 1 \)

The margin is \( 2 / \|w\| \)

• For more details on SVM please see nice slides at http://www.robots.ox.ac.uk/~az/lectures/ml/lect2.pdf
Dalal and Triggs, CVPR 2005
Learned model

\[ f(x) = w^T x + b \]

positive weights

negative weights

average over positive training data
What do negative weights mean?

\[ wx > 0 \]
\[ (w^+ - w^-)x > 0 \]
\[ w^+ > w^-x \]

Complete system should compete pedestrian/pillar/doorway models

Discriminative models come equipped with own bg
(avoid firing on doorways by penalizing vertical edges)
Why does HOG + SVM work so well?

- Similar to SIFT, records spatial arrangement of histogram orientations
- Compare to learning only edges:
  - Complex junctions can be represented
  - Avoids problem of early thresholding
  - Represents also soft internal gradients
- Older methods based on edges have become largely obsolete

• HOG gives fixed length vector for window, suitable for feature vector for SVM
Chamfer Matching

- Match points between template and image
- Measure mean distance
- Template edgel matches nearest image edgel

\[ D(T, I) = \frac{1}{|T|} \sum_{p \in T} \min_{q \in I} d(p, q) \]

- Distance transform reduces min operation to array lookup
- Computable in linear time
- Localize by sliding window search

[Gavrila & Philomin, 1999]
Chamfer Matching

- In practice performs poorly in clutter
- Unoriented edges are not discriminative enough
  (too easy to find…)

[Gavrila & Philomin, 1999]
Contour-fragment models

• Generalized Hough like representation using contour fragments

• Contour fragments learnt from edges of training images

• Hough like voting for detection
Training a sliding window detector

• Object detection is inherently asymmetric: much more “non-object” than “object” data

• Classifier needs to have very low false positive rate
• Non-object category is very complex – need lots of data
Bootstrapping

1. Pick negative training set at random
2. Train classifier
3. Run on training data
4. Add false positives to training set
5. Repeat from 2

- Collect a finite but diverse set of non-object windows
- Force classifier to concentrate on **hard negative** examples
- For some classifiers can ensure equivalence to training on entire data set
Example: train an upper body detector

- Training data – used for training and validation sets
  - 33 Hollywood2 training movies
  - 1122 frames with upper bodies marked

- First stage training (bootstrapping)
  - 1607 upper body annotations jittered to 32k positive samples
  - 55k negatives sampled from the same set of frames

- Second stage training (retraining)
  - 150k hard negatives found in the training data
Training data – positive annotations
Positive windows

Note: common size and alignment
Jittered positives
Jittered positives
Random negatives
Random negatives
Window (Image) first stage classification

- Jittered positives
- random negatives

HOG Feature Extraction

Linear SVM Classifier

\[ f(x) = w^T x + b \]

- find high scoring false positives detections

- these are the hard negatives for the next round of training

- cost = # training images x inference on each image
Hard negatives
Hard negatives
First stage performance on validation set
Precision – Recall curve

- **Precision**: % of returned windows that are correct

- **Recall**: % of correct windows that are returned

Classifier score decreasing
First stage performance on validation set
Effects of retraining

![Graph showing effects of retraining with precision and recall axes. The graph compares retrained (0.44) and initial (0.23) models.]
Side by side

before retraining

after retraining
Side by side

before retraining

after retraining
Side by side

before retraining

after retraining
Tracked upper body detections
Tracked upper body person detections

Combined face, upper body and full body detectors “vote” for upper body bounding boxes.
Detections are tracked and smoothed over video.
[Lezama, MVA thesis 2010]
Accelerating Sliding Window Search

• Sliding window search is slow because so many windows are needed e.g. $x \times y \times \text{scale} \approx 100,000$ for a $320\times240$ image

• Most windows are clearly not the object class of interest

• Can we speed up the search?
Cascaded Classification

• Build a sequence of classifiers with increasing complexity

More complex, slower, lower false positive rate

Window

Classifier 1

Classifier 2

Classifier N

Non-face

Non-face

Non-face

Possibly a face

Possibly a face

Possibly a face

Face

• Reject easy non-objects using simpler and faster classifiers
Cascaded Classification

- Slow expensive classifiers only applied to a few windows ➔ significant speed-up

- Controlling classifier complexity/speed:
  - Number of support vectors [Romdhani et al, 2001]
  - Number of features [Viola & Jones, 2001]
  - Type of SVM kernel [Vedaldi et al, 2009]
Summary: Sliding Window Detection

• Can convert any image classifier into an object detector by sliding window. Efficient search methods available.

• Requirements for invariance are reduced by searching over e.g. translation and scale.

• Spatial correspondence can be “engineered in” by spatial tiling.
Outline

1. Sliding window detectors
2. Features and adding spatial information
3. HOG + linear SVM classifier
4. Two state of the art algorithms and PASCAL VOC
   - VOC challenge
   - Vedaldi et al – multiple kernels and features, cascade
   - Felzenswalb et al – multiple parts, latent SVM
5. The future and challenges
The PASCAL Visual Object Classes (VOC) Dataset and Challenge

Mark Everingham
Luc Van Gool
Chris Williams
John Winn
Andrew Zisserman
The PASCAL VOC Challenge

• Challenge in visual object recognition funded by PASCAL network of excellence

• Publicly available dataset of annotated images

• Main competitions in classification (is there an X in this image), detection (where are the X’s), and segmentation (which pixels belong to X)

• “Taster competitions” in 2-D human “pose estimation” (2007-present) and static action classes

• Standard evaluation protocol (software supplied)
Dataset Content

• 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, ...
Annotation

- Complete annotation of all objects
- Annotated in one session with written guidelines

**Occluded**
Object is significantly occluded within BB

**Truncated**
Object extends beyond BB

**Difficult**
Not scored in evaluation

**Pose**
Facing left
Examples

Aeroplane

Bicycle

Bird

Boat

Bottle

Bus

Car

Cat

Chair

Cow
Examples

Dining Table  Dog  Horse  Motorbike  Person

Potted Plant  Sheep  Sofa  Train  TV/Monitor
Main Challenge Tasks

• Classification
  – Is there a dog in this image?
  – Evaluation by precision/recall

• Detection
  – Localize all the people (if any) in this image
  – Evaluation by precision/recall based on bounding box overlap
Detection: Evaluation of Bounding Boxes

• Area of Overlap (AO) Measure

\[ AO(B_{gt}, B_p) = \frac{|B_{gt} \cap B_p|}{|B_{gt} \cup B_p|} \]

Detection if \( AO > \text{Threshold} \)

Threshold: 50%
## Dataset Statistics

<table>
<thead>
<tr>
<th></th>
<th>train</th>
<th></th>
<th>val</th>
<th></th>
<th>trainval</th>
<th></th>
<th>test</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Images</td>
<td>Objects</td>
<td>Images</td>
<td>Objects</td>
<td>Images</td>
<td>Objects</td>
<td>Images</td>
<td>Objects</td>
</tr>
<tr>
<td>Aeroplane</td>
<td>201</td>
<td>267</td>
<td>206</td>
<td>266</td>
<td>407</td>
<td>533</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bicycle</td>
<td>167</td>
<td>232</td>
<td>181</td>
<td>236</td>
<td>348</td>
<td>468</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bird</td>
<td>262</td>
<td>381</td>
<td>243</td>
<td>379</td>
<td>505</td>
<td>760</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boat</td>
<td>170</td>
<td>270</td>
<td>155</td>
<td>267</td>
<td>325</td>
<td>537</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottle</td>
<td>220</td>
<td>394</td>
<td>200</td>
<td>393</td>
<td>420</td>
<td>787</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bus</td>
<td>132</td>
<td>179</td>
<td>126</td>
<td>186</td>
<td>258</td>
<td>365</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Car</td>
<td>372</td>
<td>664</td>
<td>358</td>
<td>653</td>
<td>730</td>
<td>1,317</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cat</td>
<td>266</td>
<td>308</td>
<td>277</td>
<td>314</td>
<td>543</td>
<td>622</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chair</td>
<td>338</td>
<td>716</td>
<td>330</td>
<td>713</td>
<td>668</td>
<td>1,429</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cow</td>
<td>86</td>
<td>164</td>
<td>86</td>
<td>172</td>
<td>172</td>
<td>336</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diningtable</td>
<td>140</td>
<td>153</td>
<td>131</td>
<td>153</td>
<td>271</td>
<td>306</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dog</td>
<td>316</td>
<td>391</td>
<td>333</td>
<td>392</td>
<td>649</td>
<td>783</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Horse</td>
<td>161</td>
<td>237</td>
<td>167</td>
<td>245</td>
<td>328</td>
<td>482</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Motorbike</td>
<td>171</td>
<td>235</td>
<td>167</td>
<td>234</td>
<td>338</td>
<td>469</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Person</td>
<td>1,333</td>
<td>2,819</td>
<td>1,446</td>
<td>2,996</td>
<td>2,779</td>
<td>5,815</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pottedplant</td>
<td>166</td>
<td>311</td>
<td>166</td>
<td>316</td>
<td>332</td>
<td>627</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sheep</td>
<td>67</td>
<td>163</td>
<td>64</td>
<td>175</td>
<td>131</td>
<td>338</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sofa</td>
<td>155</td>
<td>172</td>
<td>153</td>
<td>175</td>
<td>308</td>
<td>347</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Train</td>
<td>164</td>
<td>190</td>
<td>160</td>
<td>191</td>
<td>324</td>
<td>381</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tvmonitor</td>
<td>180</td>
<td>259</td>
<td>173</td>
<td>257</td>
<td>353</td>
<td>516</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>3,473</td>
<td>8,505</td>
<td>3,581</td>
<td>8,713</td>
<td>7,054</td>
<td>17,218</td>
<td>6,650</td>
<td>16,829</td>
</tr>
</tbody>
</table>
True Positives - Bicycle

UoCTTI_L SVM-MDPM

OXFORD_MKL

NECUIUC_CLS-DTCT
False Positives - Bicycle

UoCTTI_L SVM-MDPM

OXFORD_MKL

NECUIUC_CLS-DTCT
True Positives – TV/monitor

OXFORD_MKL

UoCTTI_LSV-MDPM

LEAR_CHI-SVM-SIFT-HOG-CLS
False Positives – TV/monitor

OXFORD_MKL

UoCTTI_LSVM-MDPM

LEAR_CHI-SVM-SIFT-HOG-CLS
Precision/Recall - Aeroplane
Precision/Recall - Car
Precision/Recall – Potted plant
Wide variety of methods: sliding window, combination with whole image classifiers, segmentation based
Object Detection with Discriminatively Trained Part Based Models

Pedro F. Felzenszwalb, David Mcallester, Deva Ramanan, Ross Girshick

PAMI 2010

Matlab code available online:
http://www.cs.brown.edu/~pff/latent/
Approach

- Mixture of deformable part-based models
  - One component per “aspect” e.g. front/side view
- Each component has global template + deformable parts
- Discriminative training from bounding boxes alone
Example Model

- One component of person model

- Root filters
  - Coarse resolution

- Part filters
  - Finer resolution

- Deformation models
Starting Point: HOG Filter

- Search: sliding window over position and scale
- Feature extraction: HOG Descriptor
- Classifier: Linear SVM

Filter $F$

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

$\phi(p, H) = \text{concatenation of HOG features from subwindow specified by } p$

Dalal & Triggs [2005]
Object Hypothesis

- Position of root + each part
- Each part: HOG filter (at higher resolution)

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

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

Score is sum of filter scores minus deformation costs
Score of a Hypothesis

\[ \text{score}(p_0, \ldots, p_n) = \sum_{i=0}^{n} F_i \cdot \phi(H, p_i) - \sum_{i=1}^{n} d_i \cdot (dx_i^2, dy_i^2) \]

- Linear classifier applied to feature subset defined by hypothesis
Part Detection

head filter

Response of filter in l-th pyramid level

\[ R_l(x, y) = F \cdot \phi(H, (x, y, l)) \]

cross-correlation

Transformed response

\[ D_l(x, y) = \max_{dx, dy} (R_l(x + dx, y + dy) - d_i \cdot (dx^2, dy^2)) \]

max-convolution, computed in linear time (spreading, local max, etc)
System

feature map

feature map at twice the resolution

response of root filter

response of part filters

transformed responses

color encoding of filter response values

combined score of root locations
Training

- Training data = images + bounding boxes
- Need to learn: model structure, filters, deformation costs
Latent SVM (MI-SVM)

Classifiers that score an example $x$ using

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$$

$\beta$ are model parameters
$z$ are latent values

Training data $D = (\langle x_1, y_1 \rangle, \ldots, \langle x_n, y_n \rangle)$ $y_i \in \{ -1, 1 \}$

We would like to find $\beta$ such that: $y_i f_\beta(x_i) > 0$

Minimize

$$L_D(\beta) = \frac{1}{2}||\beta||^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i))$$

SVM objective

"Hinge loss" on one training example

Regularizer

Which component?
Where are the parts?
Latent SVM Training

\[ L_D(\beta) = \frac{1}{2}||\beta||^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i)) \]

• Convex if we fix \( z \) for positive examples

• Optimization:
  – Initialize \( \beta \) and iterate:
    • Pick best \( z \) for each positive example
    • Optimize \( \beta \) with \( z \) fixed

• Local minimum: needs good initialization
  – Parts initialized heuristically from root
Person Model

Handles partial occlusion/truncation
Car Model

root filters
coarse resolution

part filters
finer resolution

deposition
models
Car Detections

high scoring true positives

high scoring false positives
Person Detections

- high scoring true positives
- high scoring false positives (not enough overlap)
Precision/Recall: VOC2008 Person

![Precision/Recall Graph]

- UoCTTIUCI (42.0)
- LEAR_PlusClass (19.7)
- CASIA_Det (11.2)
- XRCE_Det (9.0)
- MPI_struct (2.5)
- Jena (2.0)
Precision/Recall: VOC2008 Bicycle
Comparison of Models
Summary

• **Multiple features** and multiple **kernels** boost performance

• Discriminative learning of model with latent variables for **single feature** (HOG):
  – Latent variables can learn best alignment in the ROI training annotation
  – Parts can be thought of as local SIFT vectors
  – Some similarities to Implicit Shape Model/Constellation models but with discriminative/careful training throughout

NB: Code available for latent model!
Outline

1. Sliding window detectors
2. Features and adding spatial information
3. HOG + linear SVM classifier
4. Two state of the art algorithms and PASCAL VOC
5. The future and challenges
Current Research Challenges

• Context (See class on scenes and objects on Dec 3).
  – from scene properties: GIST, BoW, stuff
  – from other objects
  – from geometry of scene, e.g. Hoiem et al CVPR 06

• Occlusion/truncation
  – Winn & Shotton, Layout Consistent Random Field, CVPR 06
  – Vedaldi & Zisserman, NIPS 09
  – Yang et al, Layered Object Detection, CVPR 10

• 3D
  – Zhu&Ramanan, CVPR’12 (view-based representation of faces)

• Scaling up – thousands of classes
  – Torralba et al, feature sharing
  – ImageNet

• Weak and noisy supervision
Final projects

• The final project amounts to 50% of the final grade.

• You will have the opportunity to choose your own research topic and to work on a method recently published at a top-quality computer vision conference (ECCV, ICCV, CVPR) or journal (IJCV, TPAMI).

• Your task will be to:
  – (i) read and understand the research paper,
  – (ii) implement (a part of ) the paper, and
  – (iii) perform qualitative/quantitative experimental evaluation.
Final projects II.

• We will provide a list of interesting topics.

• If you would like to work on another topic (not from the list below), which you may have seen during the class or elsewhere, please consult the topic with the class instructors (I. Laptev and J. Sivic).

• You may work alone or in a group of 2-3 people. If working in a group, we expect a more substantial project, and an equal contribution from each student in the group.
Final projects III – evaluation and due dates

- **Project proposal** (due on Nov 9th). You will submit a 1-page project proposal indicating (i) your chosen topic, (ii) the plan of work, i.e. what are you going to implement, what data you are going to use, what experiments you are going to do, (iii) if working in a group, who are the members of the group and how you plan to share the work. *The project proposal will represent 10% of the final project grade.*

- **Project report** (due on Dec 23rd). You will write a short report (<3 pages) summarizing your work. *The report will represent 70% of the final project grade.*

- **Project presentation** (on Dec 11 or Dec 12). You will present your work in the class on Dec 11 or Dec 12. *The project presentation will represent 20% of the final project grade.*
Final projects IV.

Re-using other’s people code:
You can re-use other people’s code. However, you should clearly indicate in your report/presentation, what is your own code and what was provided by others (don’t forget to indicate the source).

We expect projects balanced between implementation / experimental evaluation. For example, if you implement a difficult algorithm from scratch, only few qualitative experimental results may suffice. On the other hand, if you completely use someone else’s implementation, we expect a strong quantitative experimental evaluation with analysis of the obtained results and comparison with baseline methods.
Example topics

• Please see
http://www.di.ens.fr/willow/teaching/recvis12/finalproject/

Your own chosen topic:
You can also choose your own topic, e.g. based on a paper, which has been discussed in the class. Please validate the topic with the course instructors (I. Laptev or J. Sivic) first. You can discuss the topic with the course instructors after the class or email to Ivan.Laptev@ens.fr or Josef.Sivic@ens.fr.
Example of a topic defined by students

• Defined their own problem

• Collected data (their own and the Internet)

• Applied visual representations and classification/detection techniques from the class.

Computer Vision
Recognizing playing instrument

Pierre-Adrien Nadal, Axel Barrau

December 24, 2011

Figure 7: Our first results.
Joint projects with other classes

• For example with the “Introduction to graphical models” class (F. Bach and G. Obozinski).

• The joint project between two classes is expected to be more substantial and will have a strong machine learning as well as computer vision component. Please contact the instructors of both courses if you are interested in the joint project. We will discuss and adjust the requirements from each course depending on the size of the group.

• The project should have strong “computer vision” and “graphical models” components.
Example

Activity forecasting

• Page: http://www.cs.cmu.edu/~kkitaniActivityForecasting.html
• This topic is particularly suitable for someone taking also the “Reinforcement learning” class by Remi Munos.

Fig. 1. Given a single pedestrian detection, our proposed approach forecasts plausible paths and destinations from noisy vision-input