Category-level Localization

Andrew Zisserman
Visual Geometry Group
University of Oxford
http://www.robots.ox.ac.uk/~vgg

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, Josef Sivic and Andrea Vedaldi
What we would like to be able to do...

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

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

- **Image Classification**
  - Does the image contain an aeroplane?

- **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.
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: 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**
- **Pixel-level segmentation**
- **Bounding box**
- **“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 spatial 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

No, not a car
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

\[ F(x) \]

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

Car/Non-car
Problems with sliding windows ... 

- aspect ratio
- granuality (finite grid)
- partial occlusion
- multiple responses

See 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
- Map descriptors onto a common vocabulary of visual words

Represent **sliding window** as a histogram over visual words – a **bag of words**

- Summarizes sliding window content in a fixed-length vector suitable for classification
### Examples for visual words

<table>
<thead>
<tr>
<th>Category</th>
<th>Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplanes</td>
<td><img src="image_url" alt="Airplanes Images" /></td>
</tr>
<tr>
<td>Motorbikes</td>
<td><img src="image_url" alt="Motorbikes Images" /></td>
</tr>
<tr>
<td>Faces</td>
<td><img src="image_url" alt="Faces Images" /></td>
</tr>
<tr>
<td>Wild Cats</td>
<td><img src="image_url" alt="Wild Cats Images" /></td>
</tr>
<tr>
<td>Leaves</td>
<td><img src="image_url" alt="Leaves Images" /></td>
</tr>
<tr>
<td>People</td>
<td><img src="image_url" alt="People Images" /></td>
</tr>
<tr>
<td>Bikes</td>
<td><img src="image_url" alt="Bikes Images" /></td>
</tr>
</tbody>
</table>
Intuition

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

Visual Vocabulary
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: Efficient branch and bound search over all windows
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 (ISM)
Spatial Models Considered

Fully connected shape model

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

“Star” shape model

- e.g. ISM
- Parts mutually independent
- Recognition complexity: $O(NP)$
- Method: Gen. Hough Transform

$P$ parts, $N$ positions
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
Example – Learnt Motorbike Model

Samples from appearance model

Shape model
Recognized Motorbikes

Shape model

position of object determined
Airplanes

Airplane shape model

Part 1: Det: 3x10^{-19}
Part 2: Det: 9x10^{-22}
Part 3: Det: 1x10^{-23}
Part 4: Det: 2x10^{-22}
Part 5: Det: 7x10^{-24}
Part 6: Det: 5x10^{-22}
Background: Det: 1x10^{-20}
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.

Recognition complexity: $O(N^P)$

P parts, N positions
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
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

Interest Points → Matched Codebook Entries → Probabilistic Voting → Voting Space (continuous) → Backprojection of Maximum
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

Bag of Words

Concatenate Feature Vector

Keeps fixed length feature vector for a window

[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

[1 BoW]
[4 BoW]
[16 BoW]

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

- Why extract only **sparse** image fragments?

- Good where lots of invariance and matches are needed, but not relevant to sliding window detection?

- Extract **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
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

Feature Extraction → \( \cdots \) → \( F(x) \) → pedestrian/Non-pedestrian

\( P(c|x) \propto F(x) \)
Averaged examples
Advantages of linear SVM:

• 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)

  \[
  f(x) = \sum_{i}^{S} \alpha_i k(x_i, x) + b
  \]

  \[
  \text{Non-linear}
  \]

  \[
  S = \# \text{ of support vectors} = (\text{worst case}) \ N
  \]

  \[
  \text{size of training data}
  \]

  \[
  \text{Linear}
  \]

  \[
  f(x) = \sum_{i}^{S} \alpha_i x_i^\top x + b
  \]

  \[
  = w^\top x + b
  \]

  \[
  \text{Independent of size of training data}
  \]

More on linear/non-linear in the image classification lab
Learned model

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

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
Contour-fragment models

Shotton et al ICCV 05, Opelt et al ECCV 06

- 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 \( \times \) inference on each image
Hard negatives
Hard negatives
First stage performance on validation set

![Graph showing precision vs recall with an initial value of 0.23]
Precision – Recall curve

• **Precision**: % of returned windows that are correct
• **Recall**: % of correct windows that are returned

![Graph](image)

- Classifier score decreasing

- All windows
  - Correct windows
  - Returned windows
First stage performance on validation set
Performance after retraining
Effects of retraining
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
Notes

• Training (bootstrapping, retraining) can be done in a more principled way using Structured Output learning with the cutting plane algorithm
  – See Christoph Lampert’s presentation

• An object category detector can be learnt from a single positive example
  – See Alyosha Efros’ presentation on the Exemplar SVM by Malisiewicz, Gupta, Efros, ICCV 2011
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 $320 \times 240$ 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

- 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]
  – Number of parts [Felzenszwalb et al, 2011]
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 (2010-present)

• 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

- Dining Table
- Dog
- Horse
- Motorbike
- Person
- Potted Plant
- Sheep
- Sofa
- Train
- TV/Monitor
Challenges

20 object classes

1. Classification Challenge: Name Objects
   - Predict whether at least one object of a given class is present in an image

2. Detection Challenge: Localize objects
   - Predict the bounding boxes of all objects of a given class in an image (if any)

3. Segmentation Challenge:
   - For each pixel, predict the class of the object containing that pixel or ‘background’.
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 \( B_{gt} \cap B_{p} > \text{Threshold} \)

> Threshold

50%

• Evaluation: Average precision per class on predictions
Precision/Recall – Potted plant

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

- UVA_SELSEARCH (16.2)
- NLPR_DD_DC (14.7)
- NYUUCLA_HIERARCHY (14.3)
- NUS_CONTEXT_SVM (12.3)
- OXFORD_DPM_MK (12.1)
- MISSOURI_LCC_TREE_CODING (11.6)
- MISSOURI_TREE_MAX_POOLING (11.6)
- UOCTTI_LSVM_MDPM (8.8)
- CMIC_GS_DPM (8.3)
- CMIC_SYNTHDPM (7.5)
- CORNELL_ISVM_VIEWPOINT (1.7)
- BROOKES_STRUCT_DET_CRF (1.5)
“True Positives” - Motorbike

NLPR_DD_DC

NYUUCLA_HIERARCHY

OXFORD_DPM_MK
“False Positives” - Motorbike

NLPR_DD_DC

NYUUCLA_HIERARCHY

OXFORD_DPM_MK
“True Positives” - Cat

NYUUCLA_HIERARCHY

OXFORD_DPM_MK

UVA_SELSEARCH
“False Positives” - Cat

NYUUCLA_HIERARCHY

OXFORD_DPM_MK

UVA_SELSEARCH
Progress 2008-2011

- Results on 2008 data improve for best methods 2009-2011 for almost all categories
  - Caveats: More training data + re-use of test data
Multiple Kernels for Object Detection

Andrea Vedaldi, Varun Gulshan, Manik Varma, Andrew Zisserman
ICCV 2009
Approach

• Three stage cascade
  – Each stage uses a more powerful and more expensive classifier
• Multiple kernel learning for the classifiers over multiple features
• Jumping window first stage
Multiple Kernel Classification

combine one kernel per histogram

\[ K(h, h') = \sum_{i=1}^{l'} d_i K(h_i, h'_i) \]

[Varma & Rai, 2007]
[Gehler & Nowozin, 2009]
Jumping window

Position of visual word with respect to the object

learn the position/scale/aspect ratio of the ROI with respect to the visual word

Handles change of aspect ratio
SVMs overview

- **First stage**
  - linear SVM
  - (or jumping window)
  - time: \#windows

- **Second stage**
  - quasi-linear SVM
  - $\chi^2$ kernel
  - time: \#windows $\times$ \#dimensions

- **Third stage**
  - non-linear SVM
  - $\chi^2$-RBF kernel
  - time:
    \#windows $\times$ \#dimensions $\times$ \#SVs
Results
Results
Results
Single Kernel vs. Multiple Kernels

- **Multiple Kernels** gives substantial boost

- **Multiple Kernel Learning**:
  - small improvement over averaging
  - sparse feature selection
Object Detection with Discriminatively Trained Part Based Models

Pedro F. Felzenszwalb, David Mcallester, Deva Ramanan, Ross Girshick
PAMI 2010
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
**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

- Appearance term
- Spatial prior

- Concatenation of filters and deformation parameters
- Concatenation of HOG features and part displacement features
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 $D = ((x_1, y_1), \ldots, (x_n, y_n))$  $y_i \in \{-1, 1\}$

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

Minimize

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

SVM objective
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

Alternation strategy
Person Model

Handles partial occlusion/truncation

root filters
coarse resolution

part filters
finer resolution

deformation
models
Person model with 3 left-right components

- Mixture model using max over multiple components with left-right pairs
**Car Model**

- **root filters**
  - coarse resolution

- **part filters**
  - finer resolution

- **deformation models**
Car Detections

- high scoring true positives
- high scoring false positives
Person Detections

high scoring true positives

high scoring false positives (not enough overlap)
Comparison of Models

[Graph showing precision and recall for different models labeled as 1 Root, 2 Root, 1 Root+Parts, 2 Root+Parts, and 2 Root+Parts+BB, with different line styles and colors indicating varying performance metrics.]
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**
  – from scene properties: GIST, BoW, stuff
  – from other objects, e.g. Felzenszwalb et al, PAMI 10
  – 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**

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

• **Weak and noisy supervision**