Object detection

Subhransu Maji

CMPSCI 670: Computer Vision

November 29, 2016

Administrivia

- Project presentations
  - December 8 and 13
  - 18 groups will present in a random order
  - 8 mins (6 presentation + 2 mins for questions)
  - Upload your presentation by 10am on December 8 on Moodle. I'll gather all the presentations on a single machine for presentation.
- Writeup
  - December 22 (strictly no extensions)
  - Roughly 6-8 pages
- These details are also on Moodle
  - https://moodle.umass.edu/mod/assign/view.php?id=1148269

Applications of detection

auto-focus based on faces

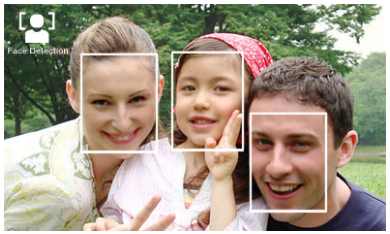
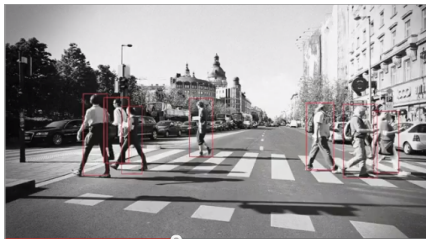
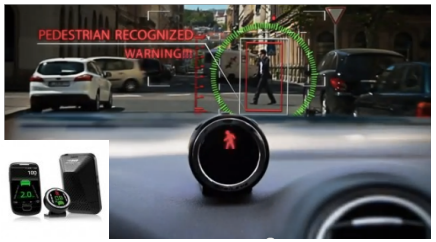


image credit : sony.co.in

pedestrian collision warning





http://www.mobileye.com

Detection = repeated classification

face or not?

Challenges of object detection

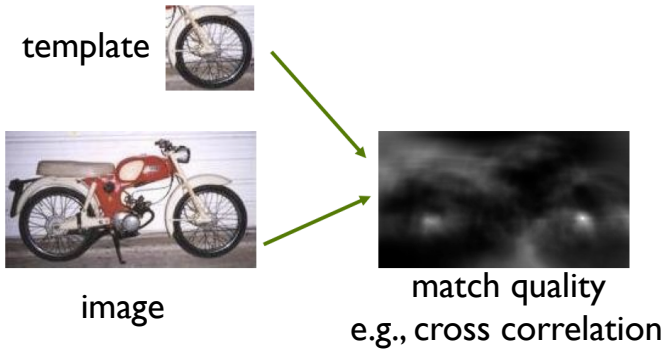
- Must evaluate tens of thousands of location+scale combinations
  - A megapixel image has ~10^6 pixels and a comparable number of candidate face locations. For computational efficiency, we should try to spend as little time as possible on the non-face windows
- Objects are rare
  - To avoid having a false positive in every image, our false positive rate has to be less than 10^{-6}

Lecture outline

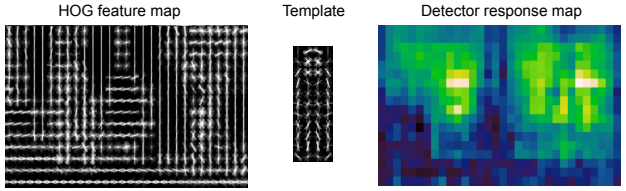
- Sliding-window detection
  - Case study: Dalal & Triggs, CVPR 2005
    - Detection as template matching
      - HOG feature pyramid
      - Non-maximum suppression
    - Learning a template — linear SVMs, hard negative mining
    - Evaluating a detector — some detection benchmarks
- Region-based detectors
  - Case study: Van de Sande et al., ICCV 2013
  - Case study: R-CNN, Girshick et al., CVPR 2014

Detection as template matching

- Consider matching with image patches
  - What could go wrong?

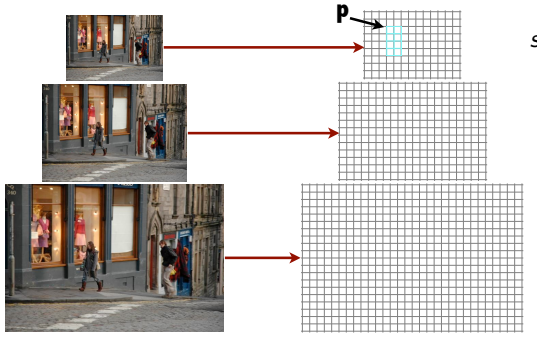


Template matching with HOG



- Compute the HOG feature map for the image
- Convolve the template with the feature map to get score
- Find peaks of the response map (non-max suppression)
- What about multi-scale?

Multi-scale template matching

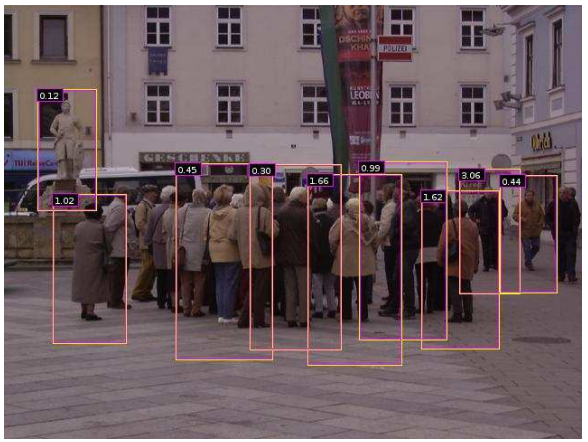


$$score(I, p) = \mathbf{w} \cdot \phi(I, p)$$

Image pyramid HOG feature pyramid

- Compute HOG of the whole image at multiple resolutions
- Score each sub-windows of the feature pyramid
- Threshold the score and perform non-maximum suppression

Example pedestrian detections

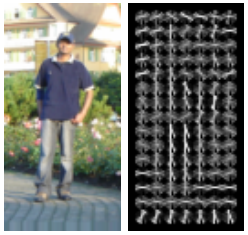


[Dalal05]

Learning a template

Pos = {...  ...}

Annotations



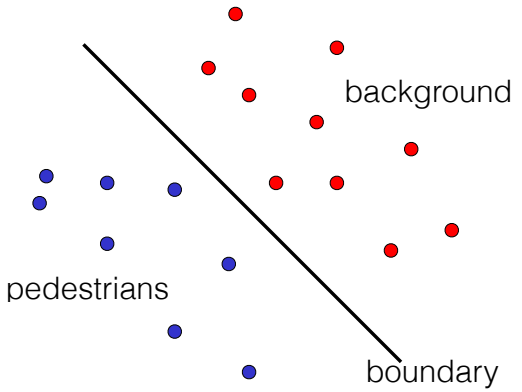
is this template good?

Cropped positive HOG

[Dalal05]

Learning a template

- Score high on pedestrians and low on background patches
- Discriminative learning setting — lets use linear classifiers!



Issue: too many background patches

Initial training

Pos = {...  ...}

Neg = {... random background patches ...}

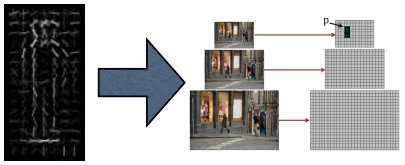
SVM

Test on cropped windows

Mining hard negatives

Pos = {...  ...}

Neg_{rand} = {... random background patches ...}



SVM "Hard" negatives

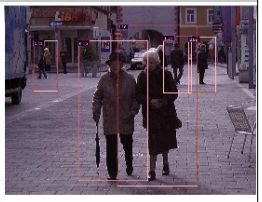
+ Neg_{hard} = {... windows with score >= -1 ...}

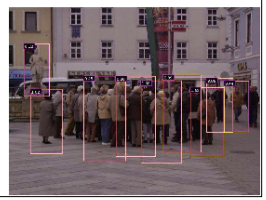
SVM

INRIA person dataset

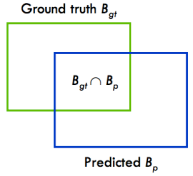
- N. Dalal and B. Triggs, CVPR 2005
- One of the first realistic datasets
  - Wide variety of articulated poses
  - Variable appearance/clothing
  - Complex backgrounds
  - Unconstrained illumination
  - Occlusions, different scales

http://pascal.inrialpes.fr/data/human/

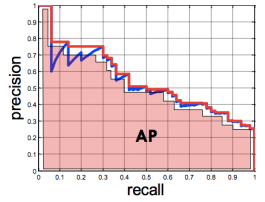




Detection evaluation

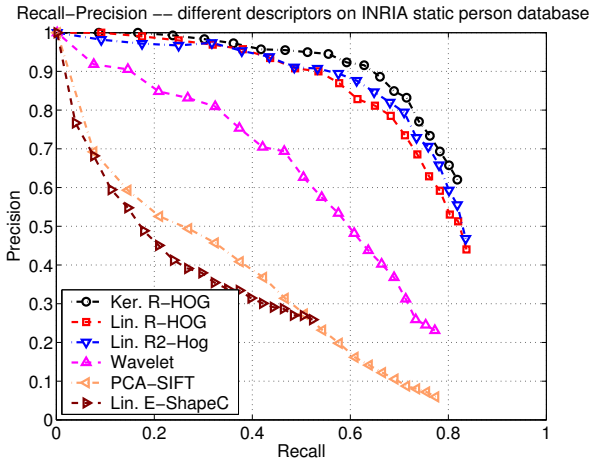


$$\text{overlap}(B_{\text{gt}}, B_{\text{p}}) = \frac{|B_{\text{gt}} \cap B_{\text{p}}|}{|B_{\text{gt}} \cup B_{\text{p}}|}$$



- Assign each prediction to
  - true positive (TP) or false positive (FP)
- $\text{Precision}_{@k}$ = $\#\text{TP}_{@k}$ / ($\#\text{TP}_{@k}$ + $\#\text{FP}_{@k}$)
- $\text{Recall}_{@k}$ = $\#\text{TP}_{@k}$ / #TotalPositives
- Average Precision (AP)

Pedestrian detection on INRIA dataset



- AP = 0.75 with a linear SVM
- Very good, right?

PASCAL VOC Challenge

- Localize & name *(detect)* 20 basic-level object categories
  - Airplane, bicycle, motorbike, bus, boat, train, car, cat, bird, cow, dog, horse, person, sheep, bottle, sofa, monitor, chair, table, plant

- Run from 2005 - 2012
- 11k training images with 500 to 8000 instances / category
- Substantially more challenging images
- Dalal and Triggs detector AP on 'person' category: **12%**

PASCAL examples

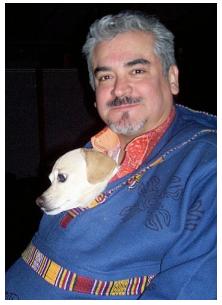


Image credits: PASCAL VOC

PASCAL examples

- Viewpoint

Image credits: PASCAL VOC

PASCAL examples

- Subcategory — “airplane” images

PASCAL examples

- Subcategory — “car” images

Problem with a “sliding window” detector

- Computationally expensive — there are too many windows
  - multiply by scales
  - multiply by aspect ratio (objects are not square)

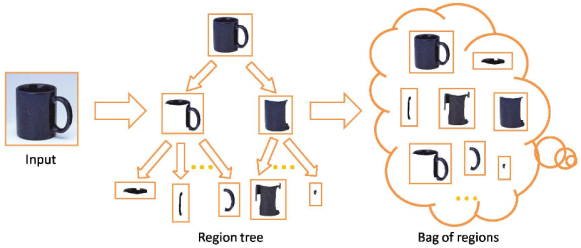
- Need very fast classifiers
  - Typically limited to
    - simple classifiers: linear classifiers and decision trees
    - simple features: gradient features

Intelligent sliding windows

- Instead of exhaustively searching over all possible windows, lets intelligently choose regions where the classifier is evaluated
- Some considerations:
  - We want a small number of such regions (~1000)
  - We want high recall — no objects should be missed
  - Category independent
    - that way we can share the cost of computing features
  - Fast — shouldn’t be slower than running the detector itself

How do we get such regions?

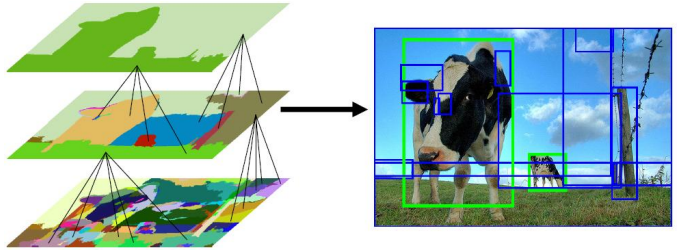
- Use low-level grouping cues to select regions
  - Cues such as color and texture similarity are category independent
  - Often fast to compute
  - Inherently span scale and aspect ratio of objects



Recognition using regions, Gu et al.

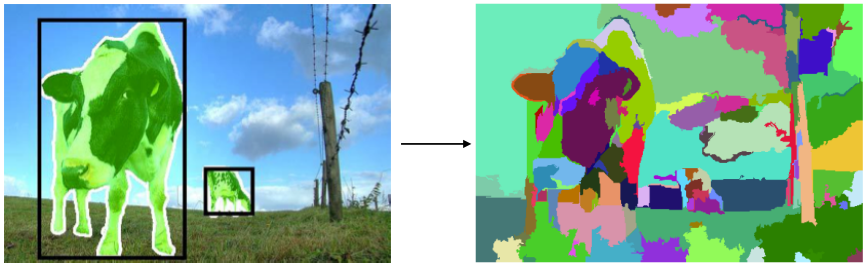
We will look at this approach

Segmentation as Selective Search for Object Recognition, K. Van de Sande, J. Uijlings, T. Gevers, and A. Smeulders, ICCV 2013



Winner of the PASCAL VOC challenge 2010-12

Lets start with segmentations



"Efficient graph-based image segmentation" Felzenszwalb and Huttenlocher, IJCV 2004

- We typically get over-segmentation for big objects, i.e., objects are broken into multiple regions
- How can we fix this?

How to obtain high recall?

- Images are intrinsically hierarchical

Hierarchical clustering

- Compute similarity measure between all adjacent region pairs *a* and *b* as:

$$S(a,b) = S_{size}(a,b) + S_{texture}(a,b)$$

Proportion of the image area that a and b jointly occupy

Histogram intersection of 8-bin gradient direction histogram computed in each color channel

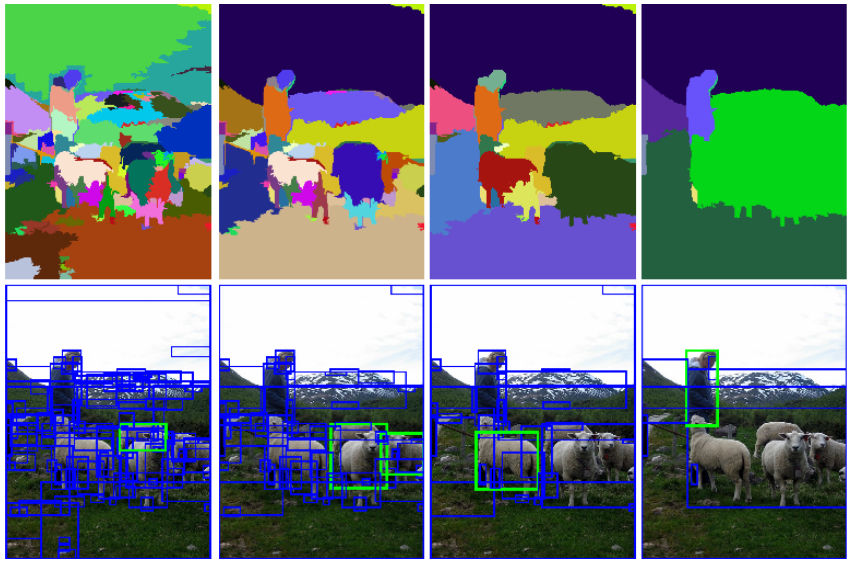
$S_{size}(a,b)$ ⇒ Encourages small regions to merge early and prevents single region from gobbling up all others one by one.

$S_{texture}(a,b)$ ⇒ Encourages regions with similar texture (and color) to be grouped early.

Hierarchical clustering

1. Merge two most similar regions based on S
2. Update similarities between the new region and its neighbors
3. Go back to step 1 until the whole image is a single regions

Example proposals



Example proposals

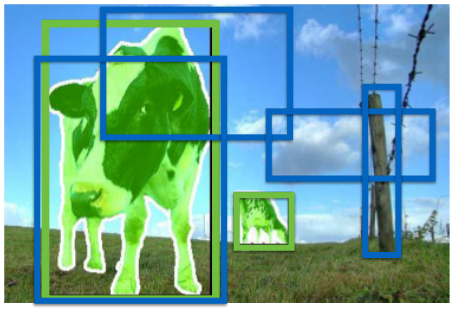
Adding diversity to the proposals

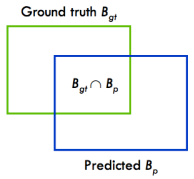
Color cues work best

Texture cues work best

- No single segmentation works for all images
- Use different color spaces
  - RGB, LAB, HSV, etc.
- Vary parameters in the Felzenszwalb segmentation method
  - k = [100, 150, 200, 250] (k= threshold parameter)

Evaluating object proposals





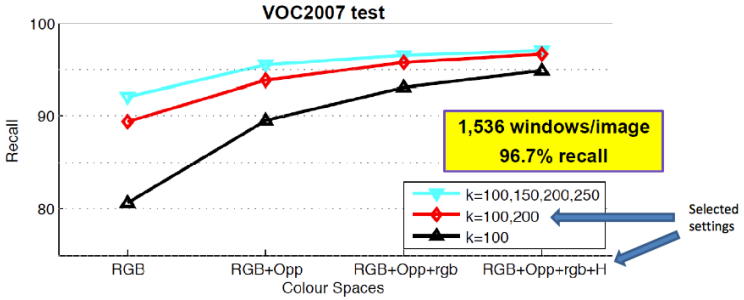
$$\text{overlap}(B_{\text{gt}}, B_{\text{p}}) = \frac{|B_{\text{gt}} \cap B_{\text{p}}|}{|B_{\text{gt}} \cup B_{\text{p}}|}$$

We want:

1. Every ground truth box be covered by at least one proposal
2. We want as few proposals as possible

Evaluating object proposals

- Recall is the proportion of objects that are covered by some box with overlap > 0.5

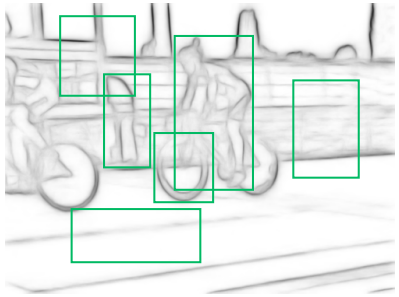


Compare this to ~100,000 regions for sliding windows

Another approach: “Objectness"

- “What is an object?” Alexe et al., CVPR 2010
- Learns to detect objects from background using
  - color, texture, edge cues
  - generic object detector

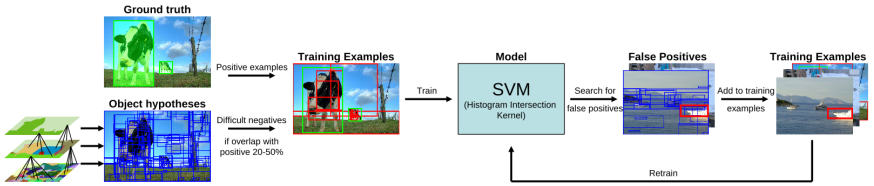
Another approach: “Edge boxes”



- Edge Boxes: Locating Object Proposals from Edges, Zitnick and Dollar, ECCV 2014
- Number of contours that are wholly contained inside the box is an indicative of the likelihood that the box contains an object.
- Very fast (0.25s per image)

Detection using region proposals

- Once again, detection = repeated classification
- But we only classify object proposals
- Training a classifier



Details of the features

- HOG + linear classifiers were used in the DT detector for efficiency
- But we can use complex features and better classifiers
  - In particular SIFT bag of words features + non-linear SVMs

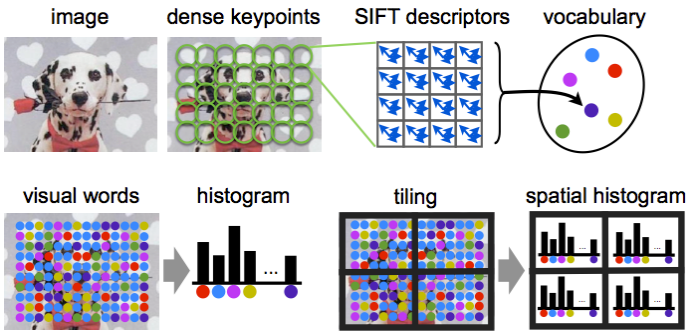
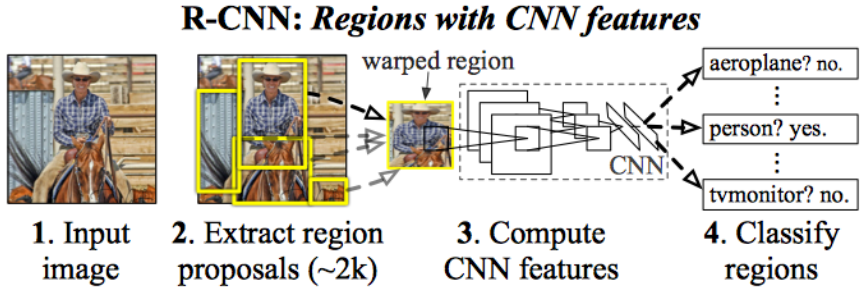


Image credit: Andrea Vedaldi

Current state of the art in detection

- R-CNNs (Girshick et al., CVPR 14)
  - Regions with CNN features



Slides credit

- Some of the slides are based on those by Ross Girshick, Andrea Vedaldi, Van de Sande, and others