CS 395 T: Class Specific Hough Forests for Object Detection

Nona Sirakova

September 2012

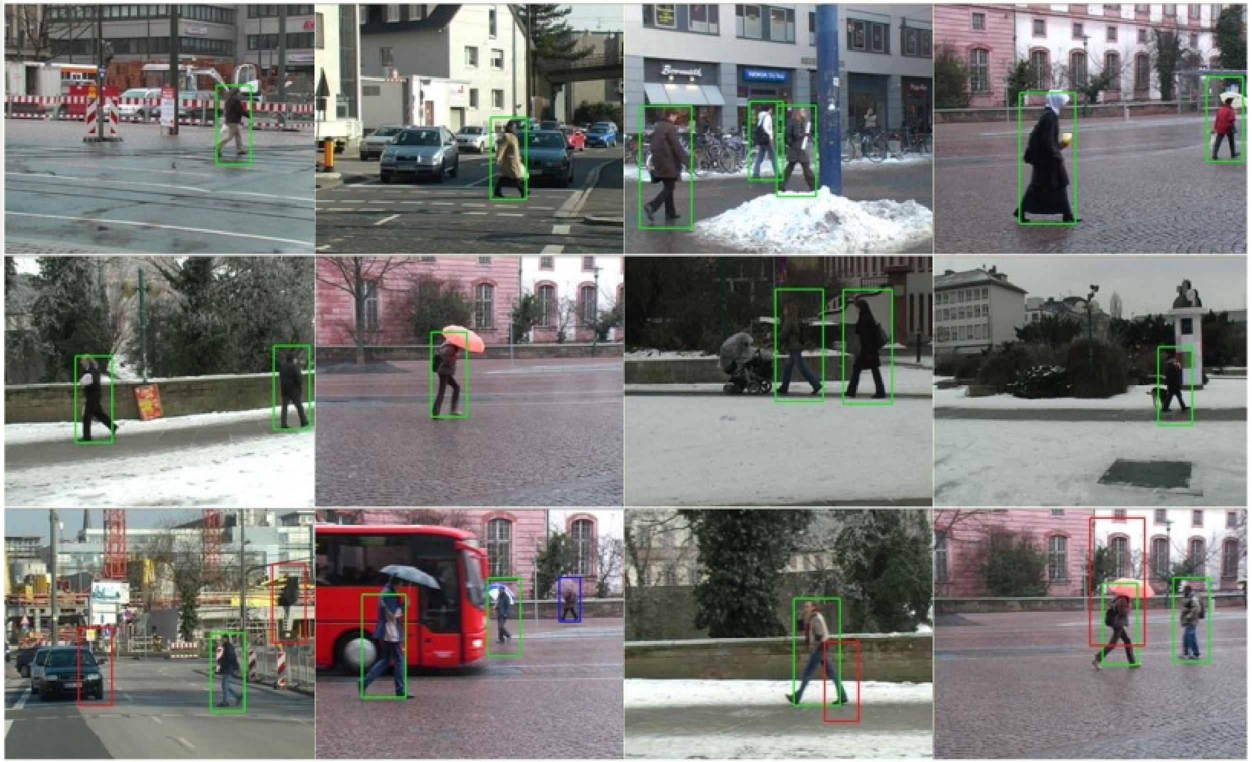
Outline:

1. Goal
2. Theme/Motivation;
3. Importance/Applications;
4. Challenges;
5. Background;
6. Key Ideas;
7. Strengths / Contributions;
8. Weaknesses;
9. Experiments:
   a. Cars
   b. Horses & Pedestrians
10. Open Issues/Extensions;

Goal

Recognize a specific object class in images.

- Denote the object's location with a bounding box.



Theme

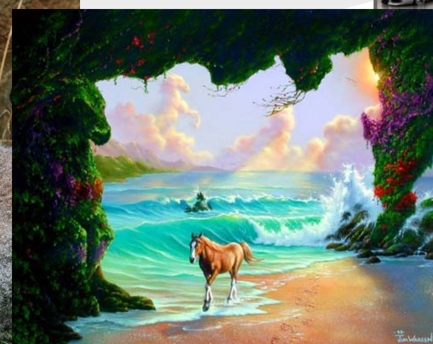
Car or plane?

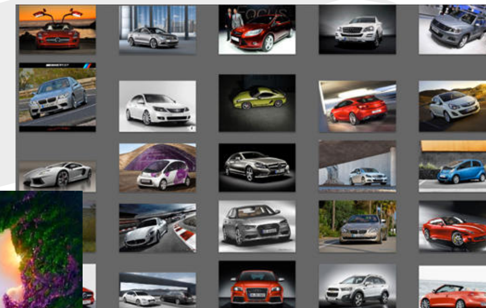
Too Many Pictures!

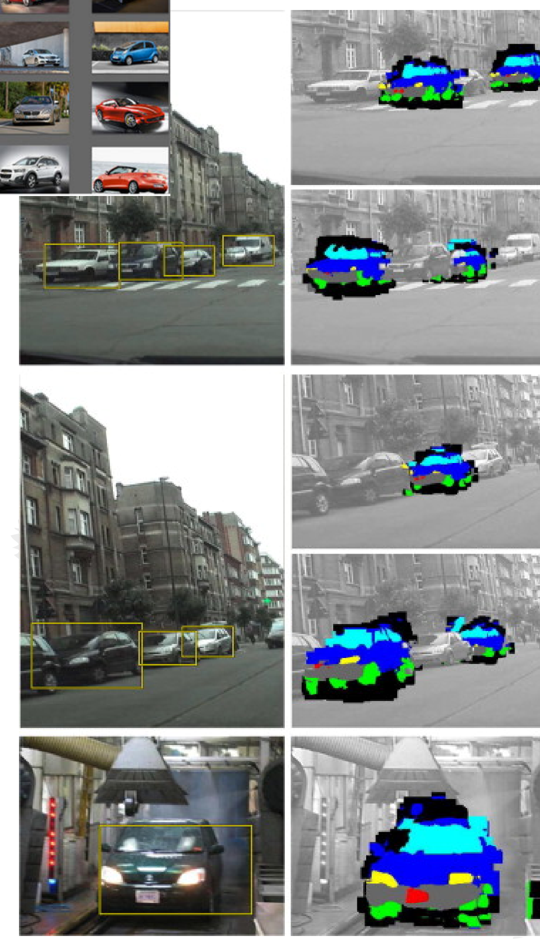
Cat or Lynx?

Importance/ Applications

- Visual search Labeling







- Content-Based Image Indexing
- Object Counting & Monitoring

Challenges

- Objects of same classes vary due to:
  - Illumination
  - Imaging conditions
  - Object articulation
  - Intraclass differences

- Challenges of natural scenes:
  - Clutter
  - Occlusion

Background: (What is done so far)

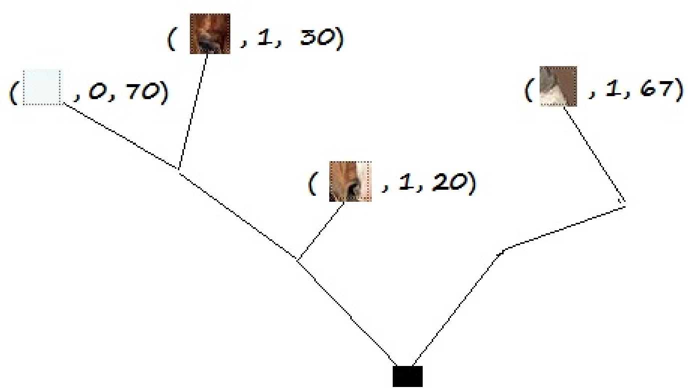
- Generative Codebooks are expensive
  - Opelt et. al
- Bottom-up approach
  - Leive et. al
- Random forests
- Sparse sampling
  - Use interest points which are rather sparse.

Image:

- Image is used to demonstrate the formation of patches, trees and random forests;
- Grid lines show patches;

Key Ideas 1:

- Hough random forests
  - $patch_i$ = (appearance, backgr/foregr, vote);
  - ex: $patch_i$ = ( , 1 , 7.6 in from horse centroid)
  - tree = $patch_i$ + $patch_j$ + ...
  - ex:

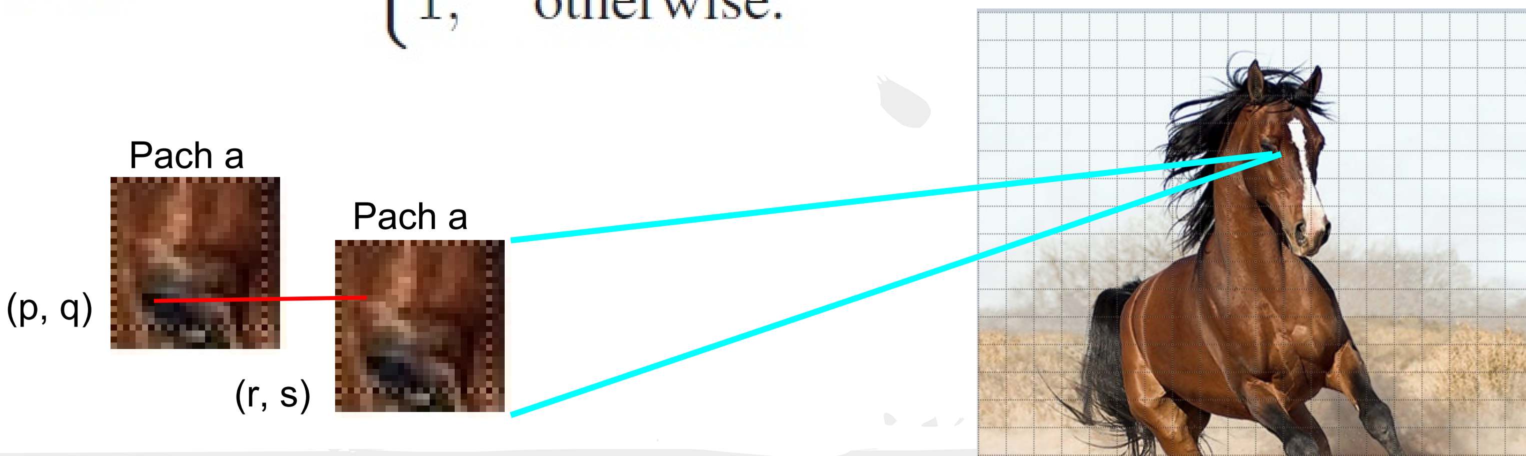


  - forest = $tree_k$ + $tree_l$ + $tree_m$ + ...

Key Ideas 2: Tree training

- How do we assign tests at each node?
  - non-leaf node gets a set of binary tests;
  - Test formation: (p, q) and (r, s) are 2 random pixels of a patch. If they differ by less than threshold t, go down one side of the tree. Else, go down the other side.

$$t_{a,p,q,r,s,\tau}(\mathcal{I}) = \begin{cases} 0, & \text{if } I^a(p,q) < I^a(r,s) + \tau \\ 1, & \text{otherwise.} \end{cases} \qquad (1)$$



Key Ideas 3: Tree training

- How do we pick tests?
  - follow random forest framework;
  - Pick tests that minimize uncertainty in Class Labels and uncertainty in Offset Vectors (votes) as we go down the tree.

Key Ideas 4: Tree training

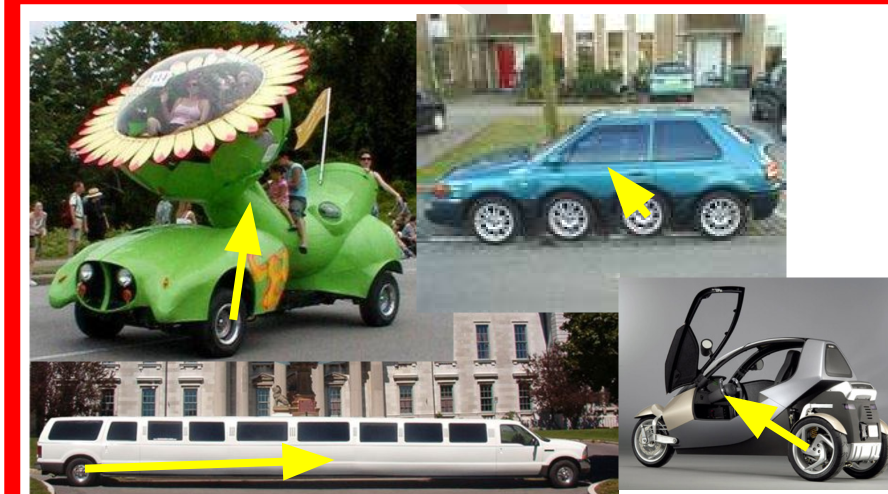
- How do we pick tests?

  2. Measure offset (vote) uncertainty given patch:

Low Uncertainty

Vote vectors point in the similar direction and have similar length

High Uncertainty

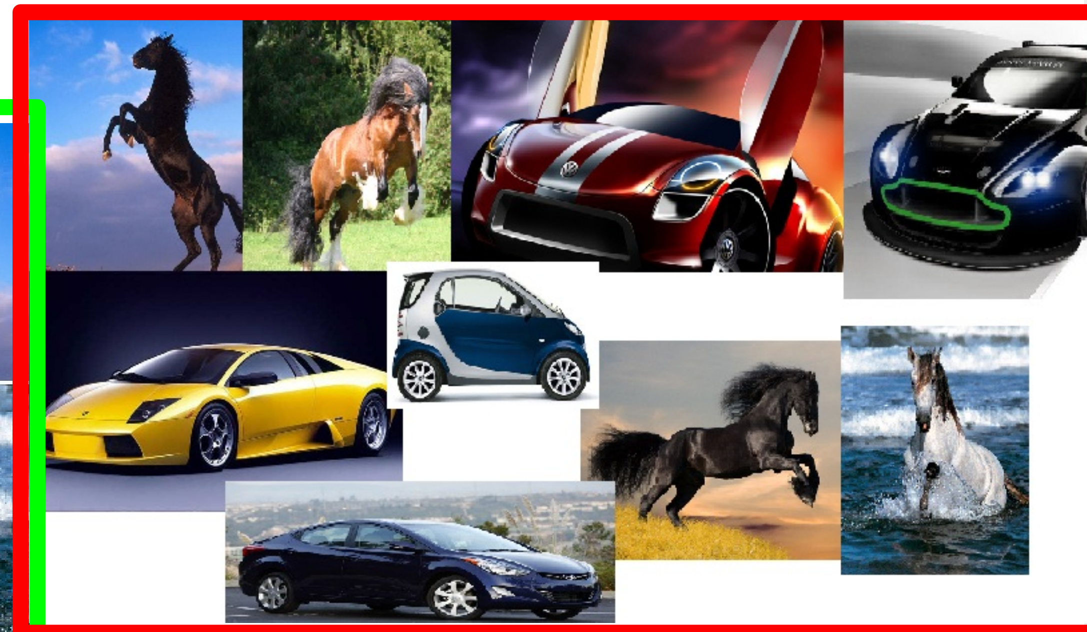


Vote vectors neither point in similar directions no have similar lengths

Key Ideas 5: Tree training

- How do we pick tests?
    1. Class Label Uncertainty.

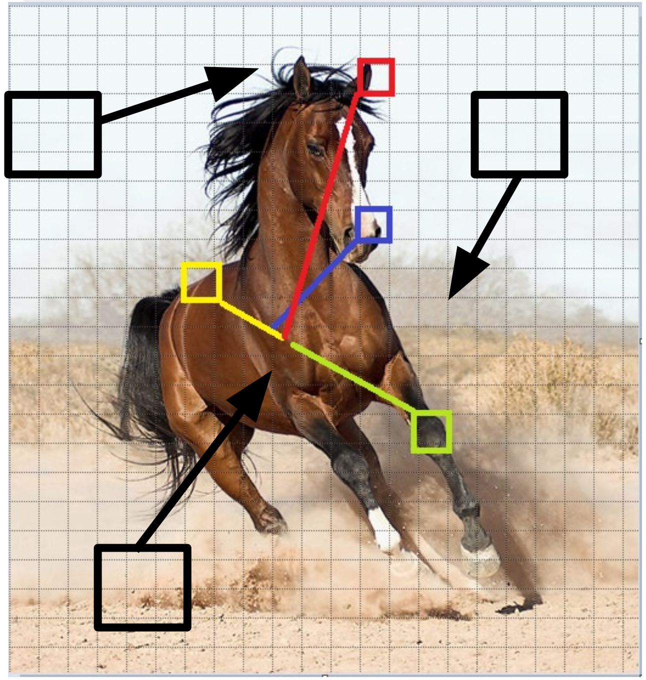
High Uncertainty



Low Uncertainty

Key Ideas 6: Tree training

- How do we pick tests?
  3. Ignore background patches. Because Class Labels of those are 0.



Key Ideas 7: Tree training

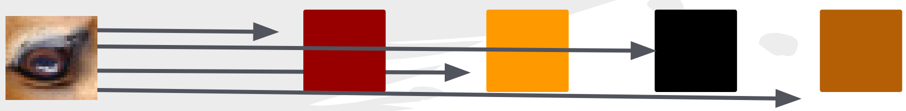
- How do we pick pixels to test?
  a. At each node, randomly choose if you will minimize Label Uncertainty or Offset Uncertainty;

Do I want to be really sure that what I pick is a horse

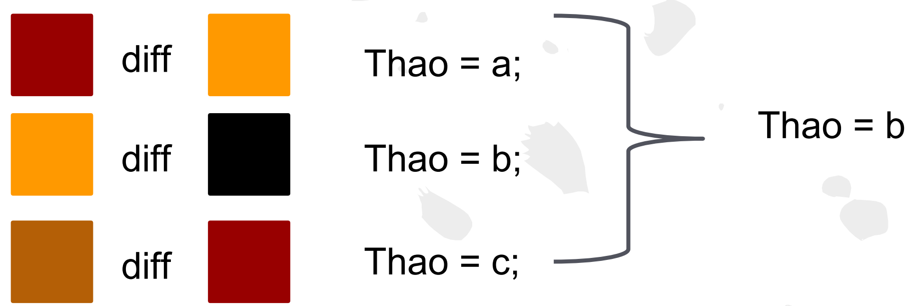
Or do I want to be really sure of that the center of the patch is at location x.

Key Ideas 8: Tree training

- How do we pick pixels to test?
  - Choose a pool of pixels to test from a patch



  - Pick the threshold (thao) randomly from the set of differences between the data;



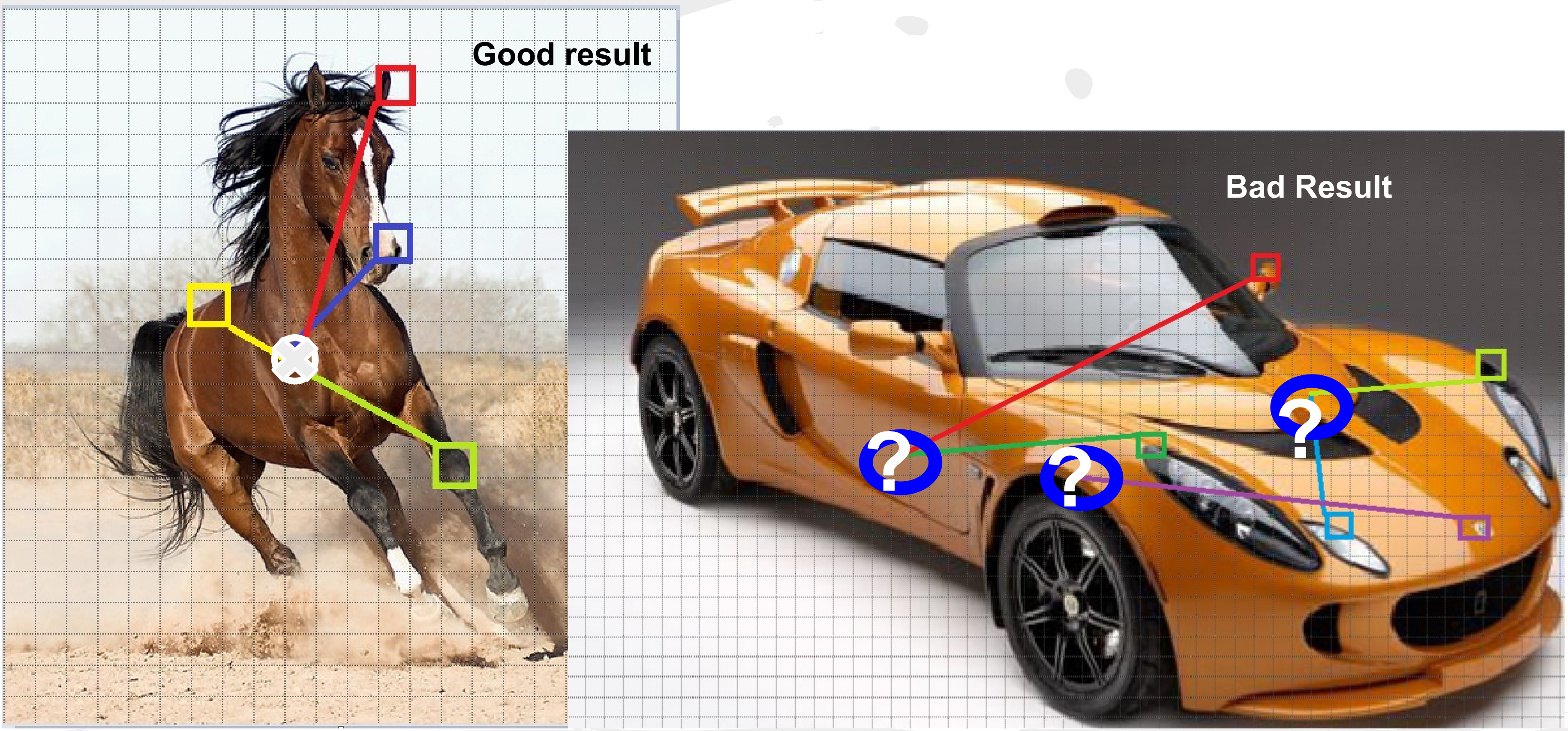
  - Pick the test that gave the min sum of the two types of uncertainties;

Key Ideas 9: Tree training

- What's the result of picking pixels to test in this way?
  - Each node has equal chance to minimize Label Uncertainty or Offset Uncertainty → leaf has low levels of both.

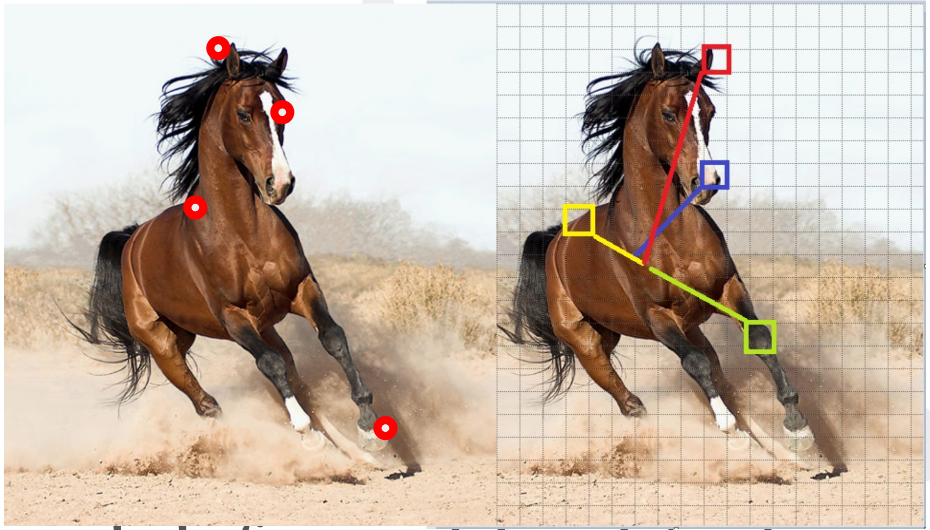
Classification: Find center of object

- Patches vote;
- Center is where we gather the most votes



Strengths / Contributions

- Fast;
- Handles large datasets;
- Matches the performance of state of the art algorithm at the time;



- Dense patch sampling;
- Can work with solid and deformable objects;

Weaknesses

- No option for detecting a variety of objects.
- Must pre-train on the exact object to detect.

- Disregarding background can be a disadvantage.

Weaknesses

- No option for detecting a variety of objects.
- Must pre-train on the exact object to detect.

- Disregarding background can be a disadvantage.

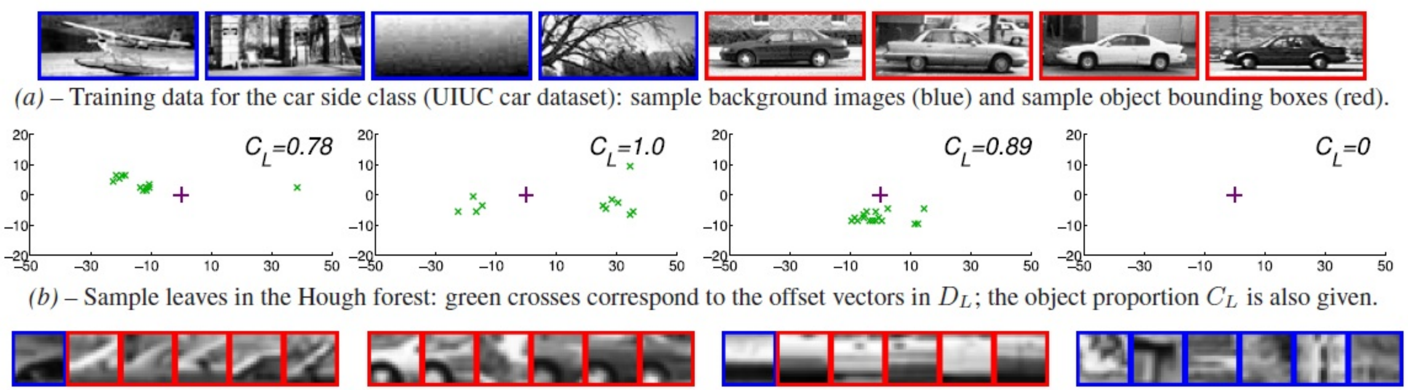
Weaknesses

- No option for detecting a variety of objects.
- Must pre-train on the exact object to detect.

- Disregarding background can be a disadvantage.

Experiments 1: Cars Data

- (UIUC cars)
  - 170 imgs with 210 cars of same scale.
  - 108 imgs with 139 cars of different scale.
  - Variation: occlusion, contrast, background clutter, illumination.
  - Constant in: overall shape of the objects.



(a) – Training data for the car side class (UIUC car dataset): sample background images (blue) and sample object bounding boxes (red).

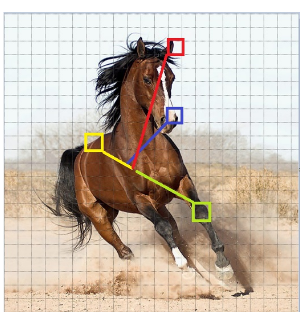
(b) – Sample leaves in the Hough forest: green crosses correspond to the offset vectors in D_L; the object proportion C_L is also given.

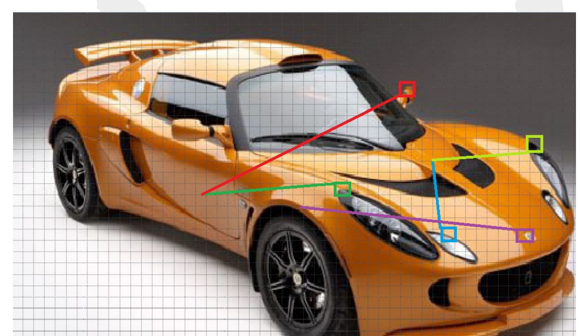
(c) – Sample patches that fall inside each of the leaves above during training (red corresponds to object patches).

Figure 2. For the set of training images shown in *(a)*, we visualize the data recorded in some of the leaves of the constructed class-specific Hough forest in *(b)*. This data consists of the object patch proportion C_L and the list of the offset vectors for object patches D_L. Note that the leaves of the Hough forest form a discriminative class-specific codebook as shown in *(c)*: the training examples falling inside each of the first three leaves can be associated with different parts of a car.

Experiments 2: Cars

- Summary
    - 20 000 binary tests considered for each node;
    - Resized images;
    - Balanced training sets - 25k/ +25k ;
    - 5 scales;
    - Precision Recall curves formed by changing the threshold for acceptance (to be accepted we need: 100 votes, 70 votes, 40 votes...)





Experiments 3: Cars

- Summary of UIUC car implementation:
  - Training
    - 550 positive examples;
    - 450 negative examples;
    - 3 channels:
      1. intensity,
      2. absolute value of x derivative;
      3. absolute value of y derivative;
    - 15 trees;

Experiments 4: Cars

- Results:
  - 98.5% accuracy for UIUC-Single
  - 98.6% accuracy for UIUC-Multi
  - Matches exactly the performance of state of the art algorithm, but is faster.

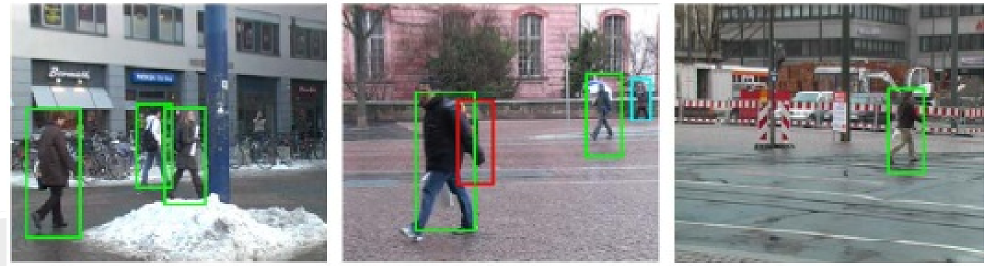
- Explanation:
  - Larger training set
  - Denser patch sample

Experiments 5: Cars

- Significance of results:

  - Outperformed approaches based solely on:
    i. Hough Transform (B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. IJCV, 77(1-3):259– 289, 2008.)
    ii. Boundary Shape (A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. IJCV, 2008.)
    iii. Random Forests (J. M. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. CVPR (1), pp. 37–44, 2006.)

Experiments 1: Horses & Pedestrians



- Data
  - TUD Pedestrians - side views
    - variation in: occlusion, scale, illumination, poses, clothing, weather.
  - INTRA Pedestrians - front & back views
    - variation in: occlusion, scale, illumination, poses, clothing, weather.
  - Weizmann Horses
    - variation in: scale, poses

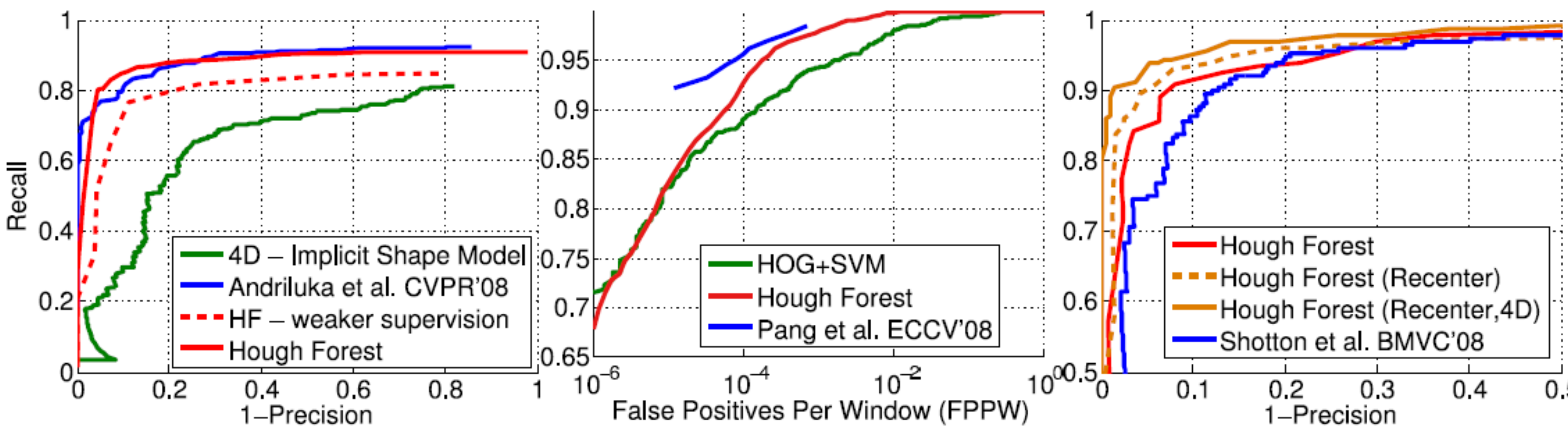
Experiments 2: Horses & Pedestrians

- Summary of data sets:
  - TUD:
    - 400 training images;
    - 250 testing images with 311 pedestrians
  - INTRA
    - 614 training images
    - 288 testing images with pedestrians; 453 imgs with no pedestrians
  - Horses
    - 200 training images, 100 images
    - 228 testing images with horses and 228 without.

Experiments 3: Horses & Pedestrians

- Summary of UIUC car implementation:
  - Training
    - 16 channels:
      1. 3 color channels of LAB color space (insert pic of LAB)
      2. absolute value of x derivative;
      3. absolute value of y derivative;
      4. absolute value of second order x derivative;
      5. absolute value of second order y derivative;
      6. 9 HOG channels
    - 15 trees

Experiments 4: Horses & Pedestrians



(a) TUD Pedestrian dataset *(b)* INRIA Pedestrian dataset *(c)* Weizmann Horse dataset

Figure 5. Hough forests (red and orange curves) demonstrate a competitive performance with respect to the previous state-of-the-art methods (blue curves) on several challenging datasets. See text for a more detailed discussion.

Experiments 5: Horses & Pedestrians

- Significance of results:
    - Outperformed approaches based solely on:
        1. Hough Transform (B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. IJCV, 77(1-3):259– 289, 2008.)
        2. Boundary Shape (A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. IJCV, 2008.)
        3. Random Forests (J. M. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. CVPR (1), pp. 37–44, 2006.)

Open Issues / Extensions

- Multi-class hough forests;
- Testing on more challenging datasets;