Object Detection by Deformable Part Models and Latent support Vector Machines

- Hierarchical Models of Objects.
- Movable Parts.
- Several Hierarchies to take into account different viewpoints.

- Energy– data & prior terms.
- Energy can be computed recursively.
- Data partially supervised – object boxes.
- *Zhu, Chen, Torrabla, Freeman, Yuille (2010)*

Overview

(1). **Hierarchical part-based models** with three layers. 4-6 models for each object to allow for pose.

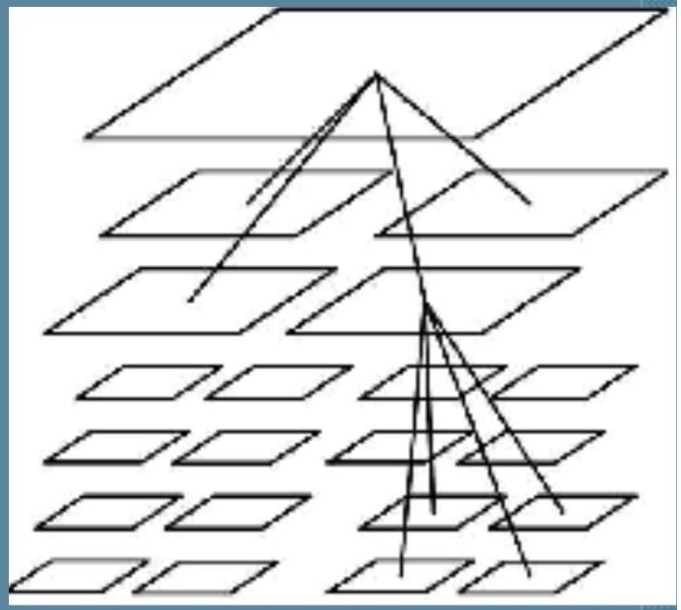
(2). **Energy potential terms:** (a) HOGs for edges, (b) Histogram of Words (HOWs) for regional appearance, (c) shape features.

(3). **Detect objects** by scanning sub-windows using dynamic programming (to detect positions of the parts).

(4). **Learn the parameters** of the models by machine learning: a variant (iCCCP) of Latent SVM.

Graph Structure:

- Each hierarchy is a 3-layer tree.
- Each node represents a part.
- Total of 46 nodes:
- (1+9+ 4 x 9)



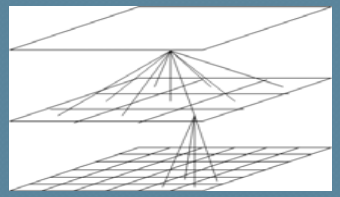
- State variables -- each node has a spatial position.

- Graph edges from parents to child – spatial constraints.

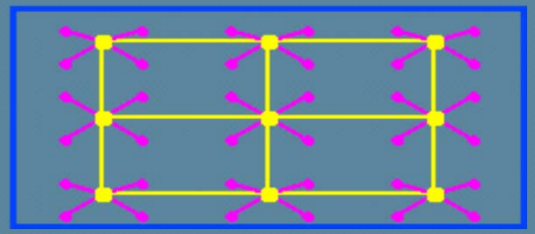
Graph Structure:

- The parts can move relative to each other enabling spatial deformations.
- Constraints on deformations are imposed by edges between parents and child (learnt).

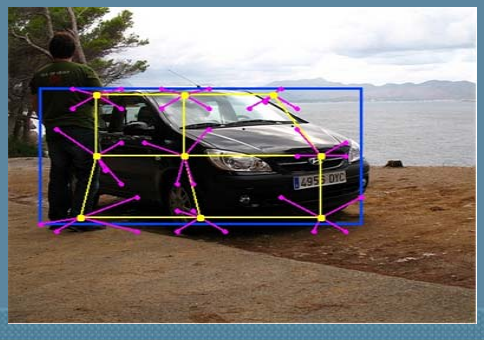
Parent-Child spatial constraints



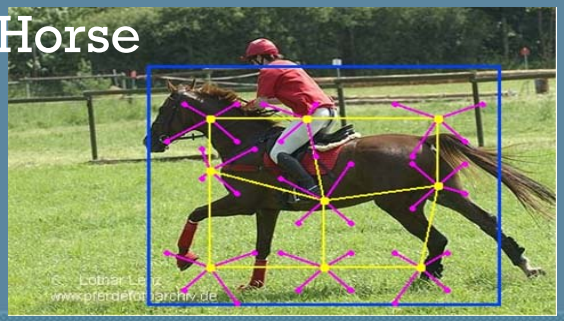
Parts: blue (1), yellow (9), purple (36)



Deformations of the Car

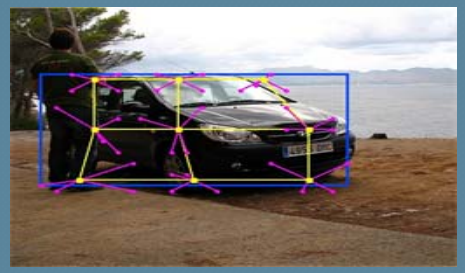


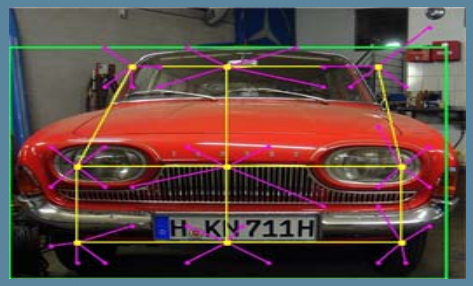
Deformations of the Horse

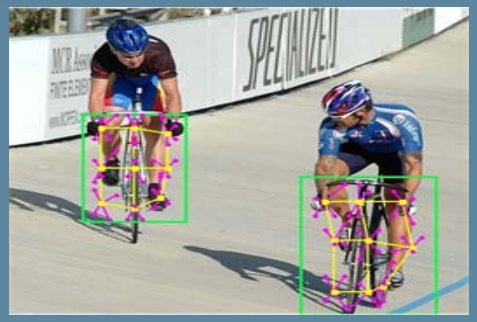


Multiple Models: Pose/Viewpoint:

- Each object is represented by 4 or 6 hierarchical models (mixture of models).
- These mixture components account for pose/viewpoint changes.







Hierarchical Part-Based Models:

The object model has variables:

1. p – represents the position of the parts.
2. V – specifies which mixture component (e.g. pose).
3. y – specifies whether the object is present or not.
4. w – model parameter (to be learnt).

During learning the part positions p and the pose V are unknown – so they are latent variables and will be expressed as V=(h,p)

Energy of the Model:

The "energy" of the model is defined to be: $-\omega \cdot \Phi(x, y, h)$ where x is the image in the region. $y^*, h^* = \arg\max \omega \cdot \Phi(x, y, h)$

- The object is detected by solving:

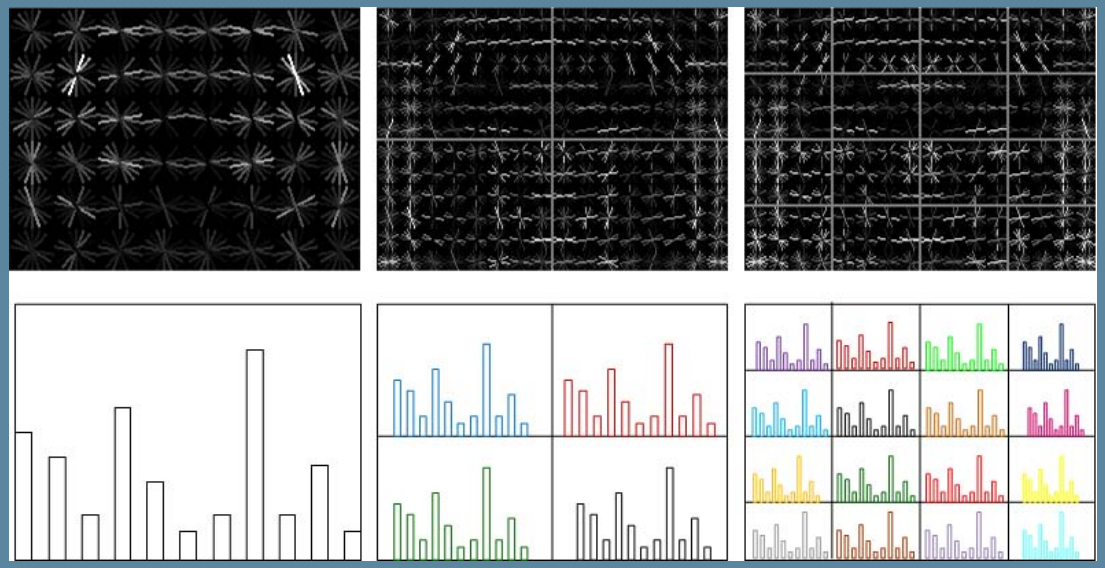
- If $y^* = +1$ then we have detected the object.
- If so, $h^* = (p^*, V^*)$ specifies the mixture component and the positions of the parts.

Energy of the Model:

- Three types of potential terms $\Phi(x, y, h)$
  - (1) Spatial terms $\Phi_{shape}(y, h)$ specify the distribution on the positions of the parts.
  - (2) Data terms for the edges of the object $\Phi_{HOG}(x, y, h)$ defined using HOG features.
  - (3) Regional appearance data terms $\Phi_{HOW}(x, y, h)$ defined by histograms of words (HOWs – grey SIFT features and K-means).

Energy : HOGs and HOWs

- Edge-like: Histogram of Oriented Gradients (Upper row)
- Regional: Histogram Of Words (Bottom row)
- 13950 HOGs + 27600 HOWs



Object Detection

- To detect an object requiring solving:

$$y^*, h^* = \arg\max \omega \cdot \Phi(x, y, h)$$

  for each image region.

- We solve this by scanning over the sub-windows of the image, use dynamic programming to estimate the part positions p and do exhaustive search over the $y \,\&\, V$

Learning by Latent SVM

- The input to learning is a set of labeled image regions. $\{(x\_i, y\_i) : i = 1, \ldots, N\}$
- Learning require us to estimate the parameters ω
- While simultaneously estimating the hidden variables $h = (p, V)$
- Classically EM – approximate by machine learning, latent SVMs.

Latent SVM Learning

- We use Yu and Joachim's (2009) formulation of latent SVM.
- This specifies a non-convex criterion to be minimized. This can be re-expressed in terms of a convex plus a concave part.

$$\min_w \frac{1}{2}\| w\|^2 + C\sum_{i=1}^{N}\left[\max_{y,h}[w\cdot\Phi(x_i,y,h)+L(y_i,y,h)] - \max_h[w\cdot\Phi(x_i,y_i,h)]\right]$$

$$\Rightarrow \quad \min_w\left[\frac{1}{2}\| w\|^2 + C\sum_{i=1}^{N}\max_{y,h}[w\cdot\Phi(x_i,y,h)+L(y_i,y,h)]\right] - \left[C\sum_{i=1}^{N}\max_h[w\cdot\Phi(x_i,y_i,h)]\right]$$

Latent SVM Learning

- Following Yu and Joachims (2009) adapt the CCCP algorithm (Yuille and Rangarajan 2001) to minimize this criterion.
- CCCP iterates between estimating the hidden variables and the parameters (like EM).
- We propose a variant – incremental CCCP – which is faster.
- Result: our method works well for learning the parameters *without* complex initialization.

Learning : Incremental CCCP

- Iterative Algorithm:
  - Step 1: fill in the latent positions with best score(DP)
  - Step 2: solve the structural SVM problem using partial negative training set (incrementally enlarge).
- Initialization:
  - No pretraining (no clustering).
  - No displacement of all nodes (no deformation).
  - Pose assignment: maximum overlapping
- Simultaneous multi-layer learning

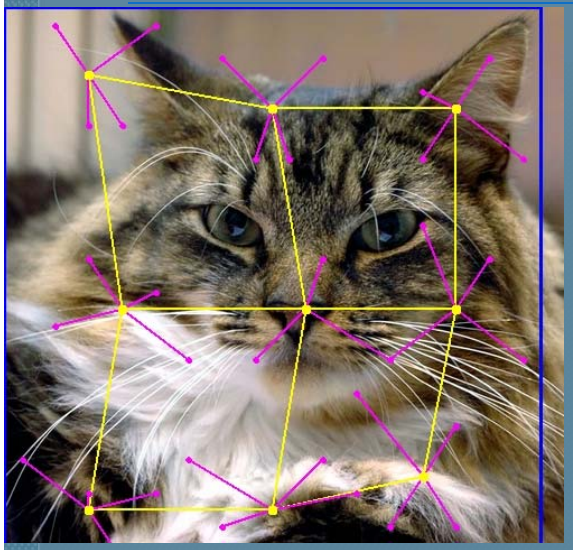
Kernels

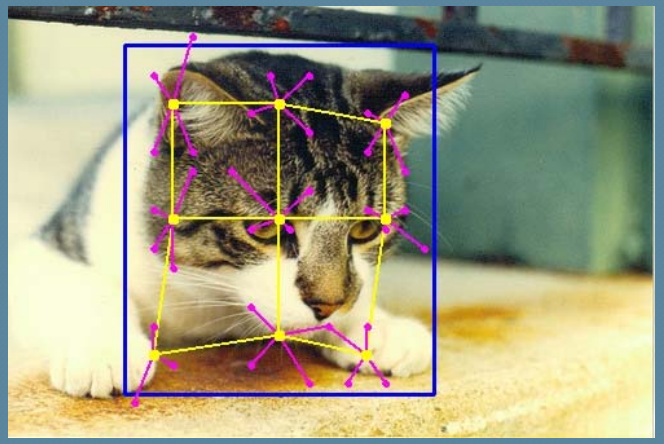
- We use a quasi-linear kernel for the HOW features, linear kernels of the HOGs and for the spatial terms.
- We use:
  - (i) equal weights for HOGs and HOWs.
  - (ii) equal weights for all nodes at all layers.
  - (iii) same weights for all object categories.
- Note: tuning weights for different categories will improve the performance.
- *The devil is in the details.*

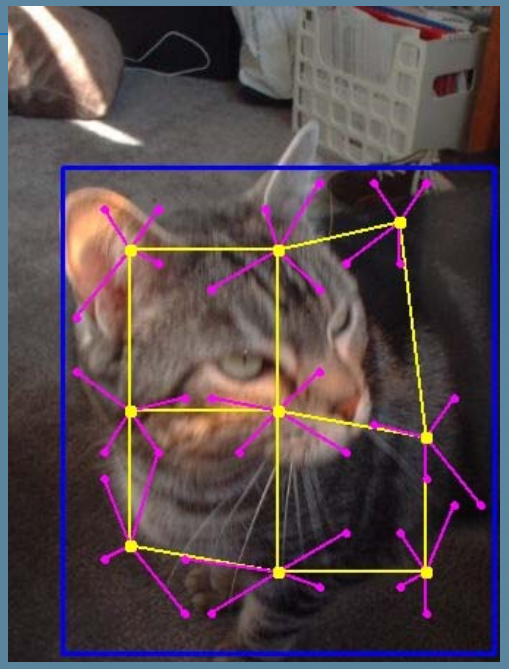
Post-processing: Context Modeling

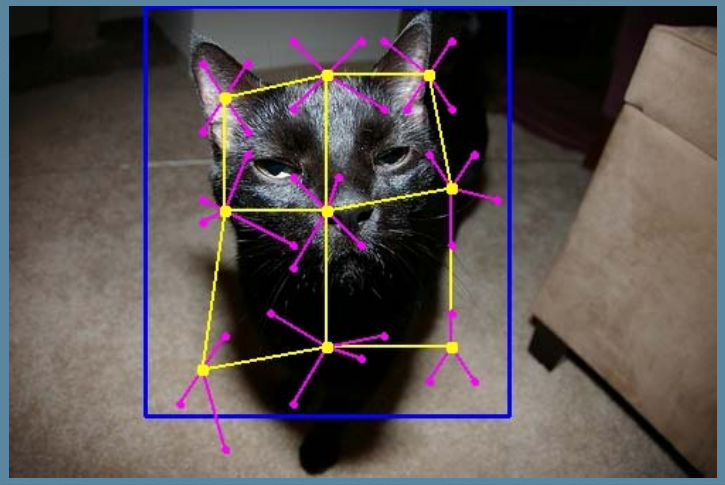
- Post-processing:
  - Rescoring the detection results
- Context modeling: SVM+ contextual features
  - best detection scores of 20 classes, locations, recognition scores of 20 classes
- Recognition scores (Lazebnik CVPR06, Van de Sande PAMI 2010, Bosch CIVR07)
  - SVM + spatial pyramid + HOWs (no latent position variable)

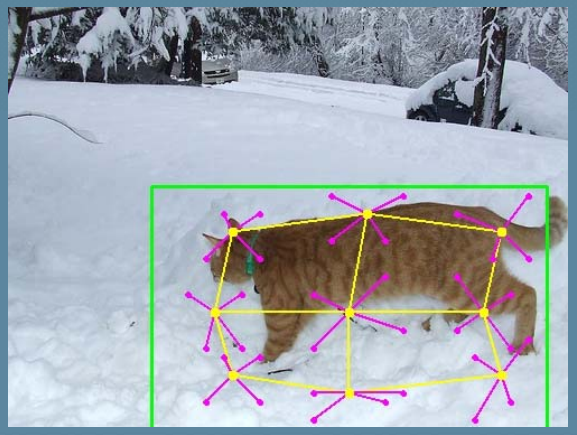
Detection Results on PASCAL 2010: Cats



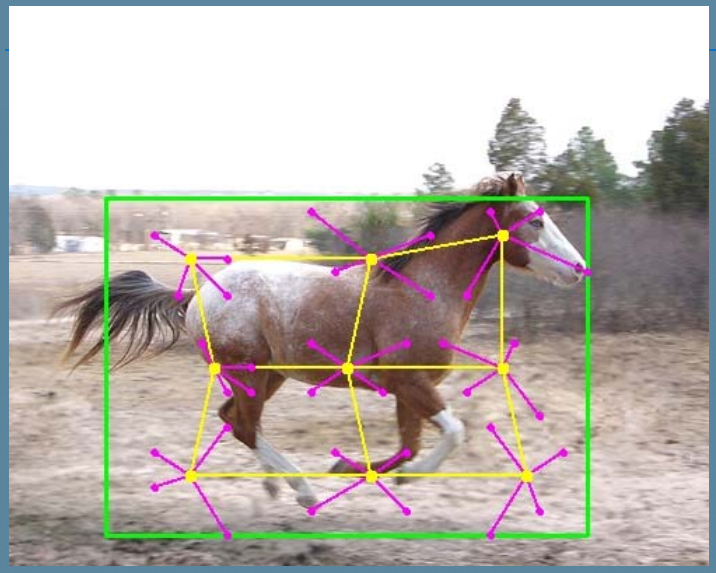


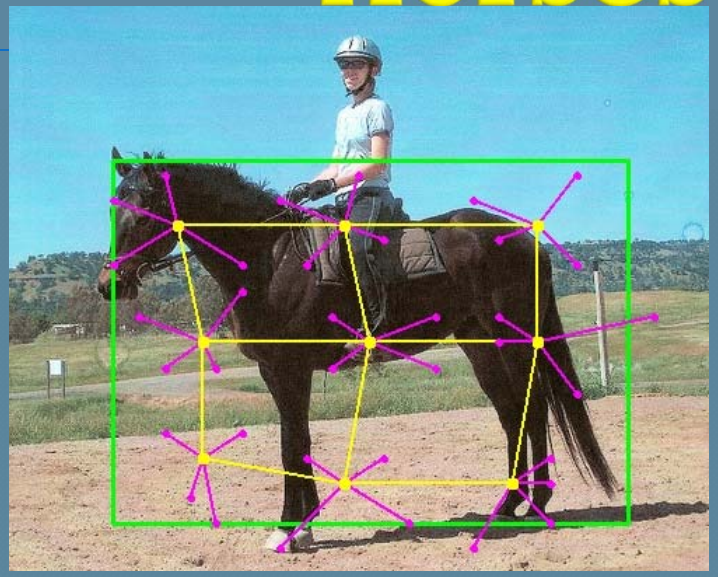


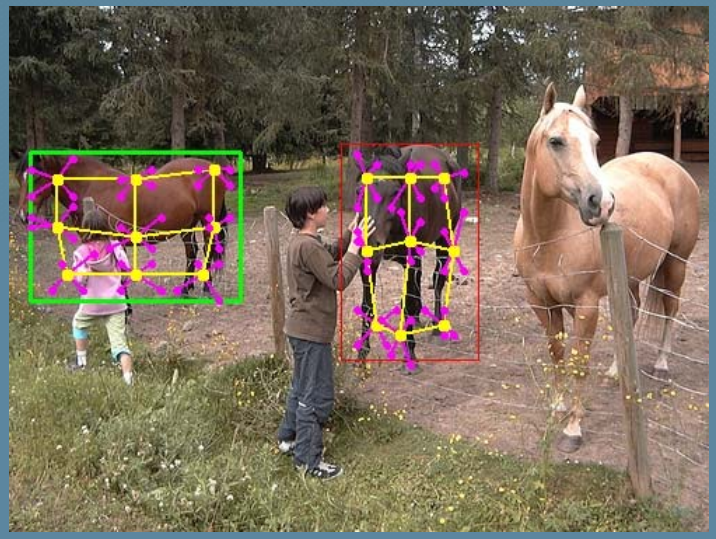


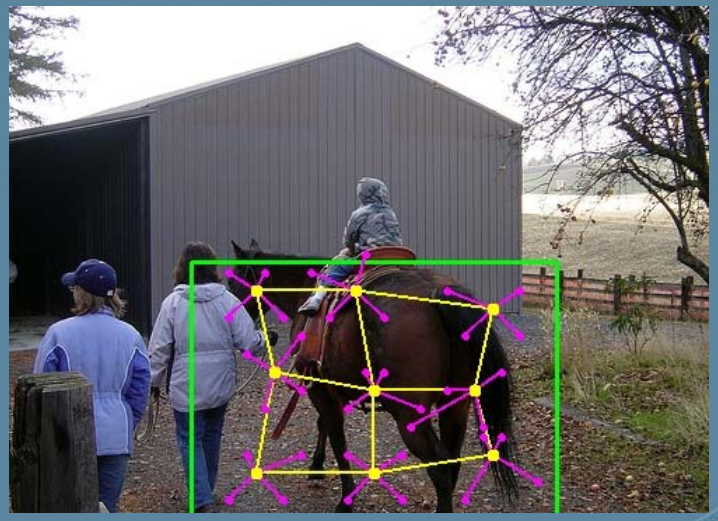


Horses

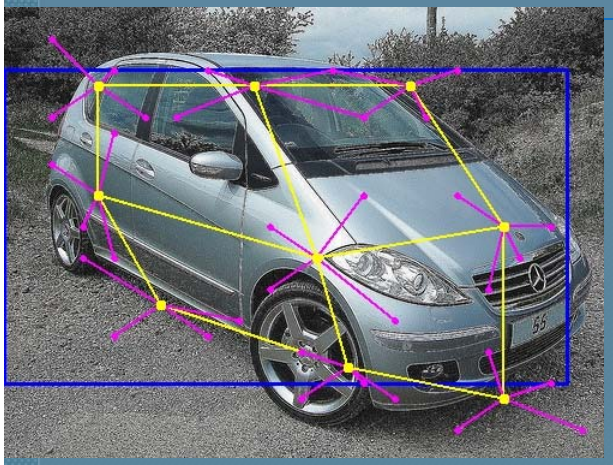


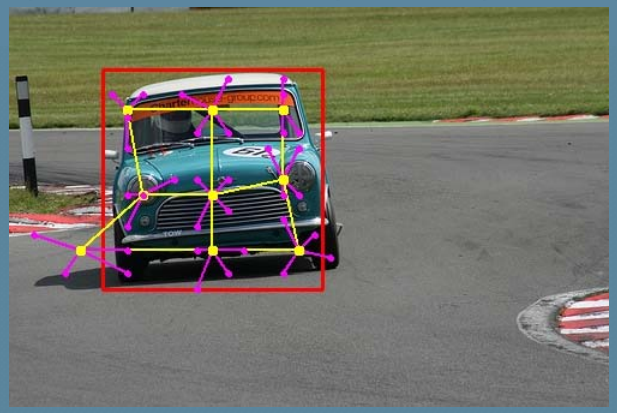


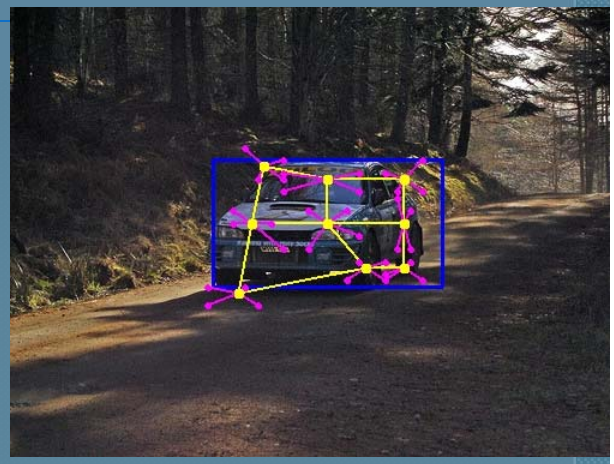


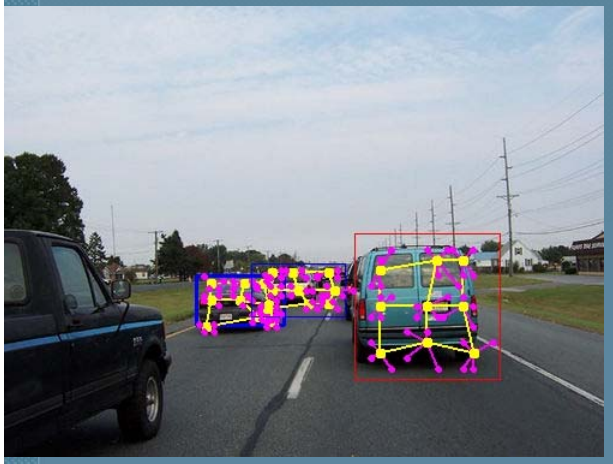


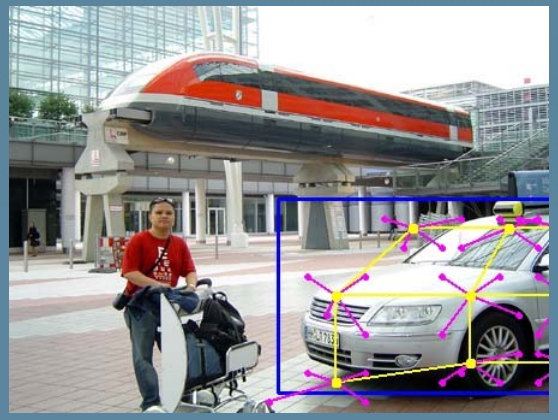
Cars

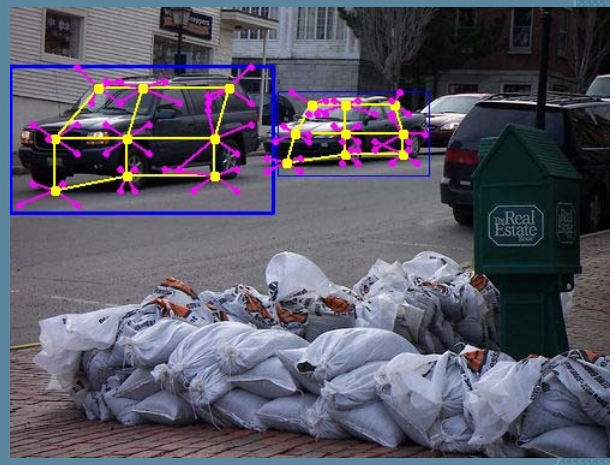




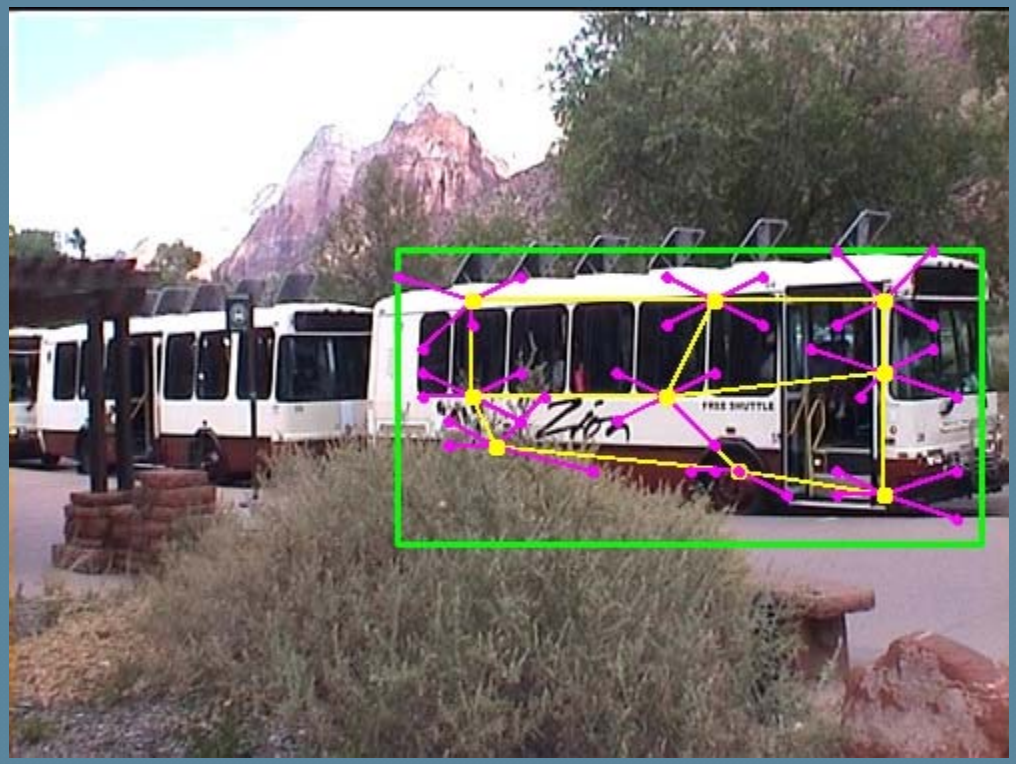


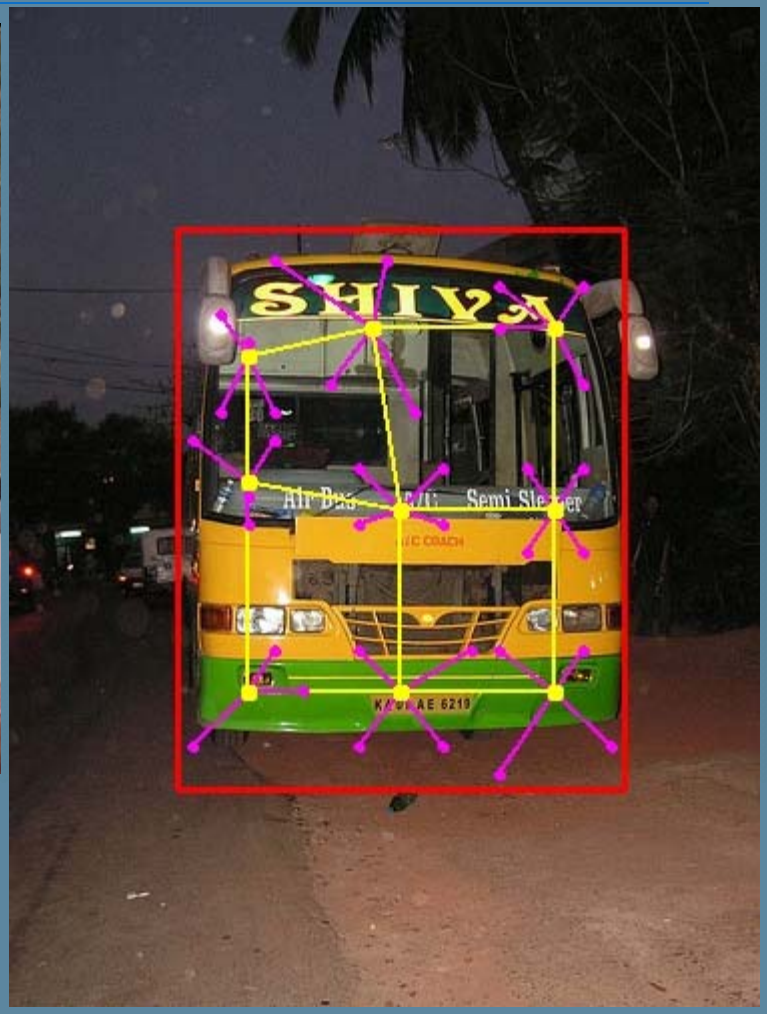






Buses





Comparisons on PASCAL 2010

- Mean Average Precision (mAP).
- Compare AP's for Pascal 2010 and 2009.

Methods (trained on 2010)	MIT-UCLA	NLPR	NUS	UoCTTI	UVA	UCI
Test on 2010	35.99	36.79	34.18	33.75	32.87	32.52
Test on 2009	36.72	37.65	35.53	34.57	34.47	33.63