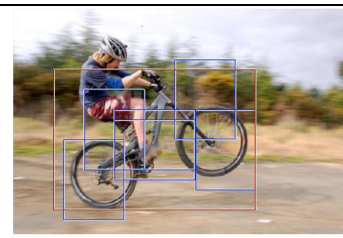
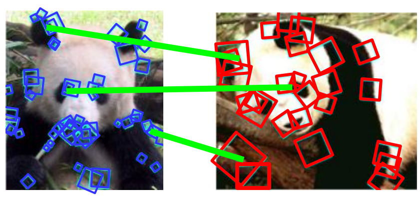


Detecting people & deformable object models

Tues Nov 24
Kristen Grauman
UT Austin



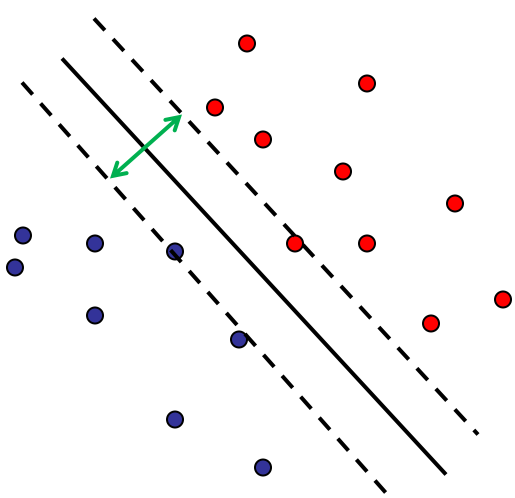
Today

- Support vector machines (SVM)
  - Basic algorithm
  - Kernels
    - Structured input spaces: Pyramid match kernels
  - Multi-class
  - HOG + SVM for person detection
    - Visualizing a feature: Hoggles
- Evaluating an object detector

Review questions

- What are tradeoffs between the one vs. one and one vs. all paradigms for multi-class classification?
- What roles do kernels play within support vector machines?
- What can we expect the training images associated with support vectors to look like?
- What is hard negative mining?

Recall: Support Vector Machines (SVMs)



- Discriminative classifier based on *optimal separating line (for 2d case)*
- Maximize the *margin* between the positive and negative training examples

Finding the maximum margin line

1. Maximize margin $2/\|\mathbf{w}\|$
2. Correctly classify all training data points:

$$\mathbf{x}_i \text{ positive } (y_i = 1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$$
$$\mathbf{x}_i \text{ negative } (y_i = -1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$$

Quadratic optimization problem:

Minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$

Subject to $y_i(\mathbf{w}\cdot\mathbf{x}_i+b) \geq 1$

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery,

Finding the maximum margin line

- Solution: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

  $b = y_i - \mathbf{w}\cdot\mathbf{x}_i$ (for any support vector)

  $$\mathbf{w}\cdot\mathbf{x} + b = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b$$

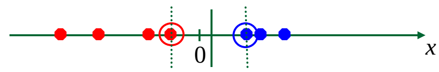
- Classification function:

$$f(x) = \text{sign}\,(\mathbf{w}\cdot\mathbf{x} + \text{b})$$
$$= \text{sign}\left(\sum_{\text{i}} \alpha_{\text{i}} y_i \boxed{\mathbf{x}_i \cdot \mathbf{x}} + b\right)$$

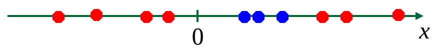
C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery,

Non-linear SVMs

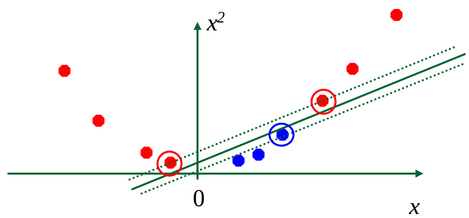
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?



- How about… mapping data to a higher-dimensional space:



Nonlinear SVMs

- *The kernel trick*: instead of explicitly computing the lifting transformation $\varphi(\mathbf{x})$, define a kernel function K such that

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$$

- This gives a nonlinear decision boundary in the original feature space:

$$\sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

Examples of kernel functions

- Linear: $K(x_i, x_j) = x_i^T x_j$

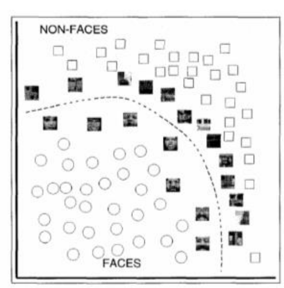
- Gaussian RBF: $K(x_i,x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$

- Histogram intersection:

$$K(x_i, x_j) = \sum_k \min(x_i(k), x_j(k))$$

SVMs for recognition

1. Define your representation for each example.
2. Select a kernel function.
3. Compute pairwise kernel values between labeled examples
4. Use this "kernel matrix" to solve for SVM support vectors & weights.
5. To classify a new example: compute kernel values between new input and support vectors, apply weights, check sign of output.



SVMs: Pros and cons

- Pros
  - Kernel-based framework is very powerful, flexible
  - Often a sparse set of support vectors – compact at test time
  - Work very well in practice, even with small training sample sizes

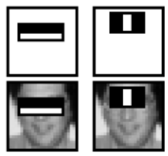
- Cons
  - No “direct” multi-class SVM, must combine two-class SVMs
  - Can be tricky to select best kernel function for a problem
  - Computation, memory
    - During training time, must compute matrix of kernel values for every pair of examples
    - Learning can take a very long time for large-scale problems

Adapted from Lana Lazebnik

Today

- Support vector machines (SVM)
  - Basic algorithm
  - Kernels
    - Structured input spaces: Pyramid match kernels
  - Multi-class
  - HOG + SVM for person detection
    - Visualizing a feature: Hoggles
- Evaluating an object detector

Window-based models: Three case studies

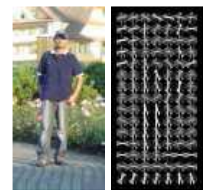


Boosting + face detection

Viola & Jones

NN + scene Gist classification

e.g., Hays & Efros



SVM + person detection

e.g., Dalal & Triggs

Slide credit: Kristen Grauman

Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alps, 655 avenue de l'Europe, Montbonnot 38334, France
{Navneet.Dalal,Bill.Triggs}@inrialpes.fr, http://lear.inrialpes.fr

Abstract

We study the question of feature sets for robust visual object recognition, adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, we show experimentally that grids of Histograms of Oriented Gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results. The new approach gives near-perfect separation on the original MIT pedestrian database, so we introduce a more challenging dataset containing over 1800 annotated human images with a large range of pose variations and backgrounds.

1 Introduction

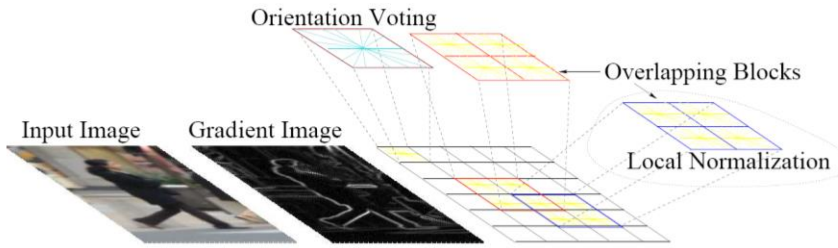
We briefly discuss previous work on human detection in §2, give an overview of our method §3, describe our data sets in §4 and give a detailed description and experimental evaluation of each stage of the process in §5–6. The main conclusions are summarized in §7.

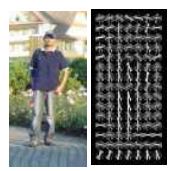
2 Previous Work

There is an extensive literature on object detection, but here we mention just a few relevant papers on human detection [18,17,22,16,20]. See [6] for a survey. Papageorgiou *et al* [18] describe a pedestrian detector based on a polynomial SVM using rectified Haar wavelets as input descriptors, with a parts (subwindow) based variant in [17]. Depoortere *et al* give an optimized version of this [2]. Gavrila & Philomen [8] take a more direct approach, extracting edge images and matching them to a set of learned exemplars using chamfer distance. This has been used in a practical real-time pedestrian detection system [7]. Viola *et al* [22] build an efficient

- CVPR 2005

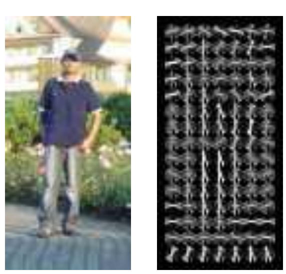
HoG descriptor





Dalal & Triggs, CVPR 2005

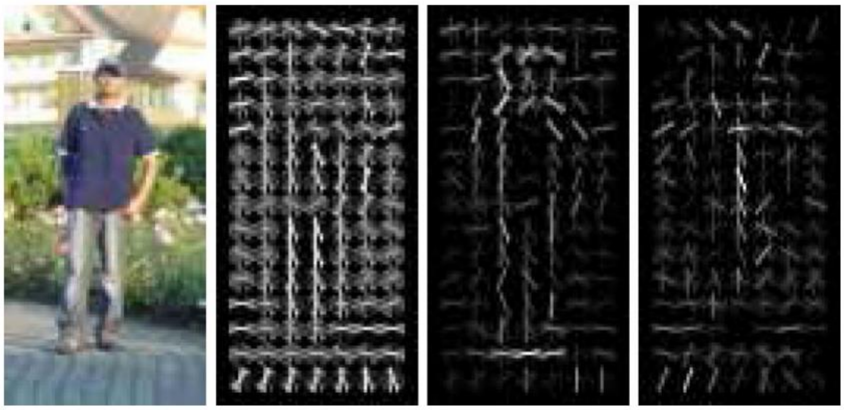
Person detection with HoG's & linear SVM's



- Map each grid cell in the input window to a histogram counting the gradients per orientation.
- Train a linear SVM using training set of pedestrian vs. non-pedestrian windows.

Dalal & Triggs, CVPR 2005

Person detection with HoG's & linear SVM's

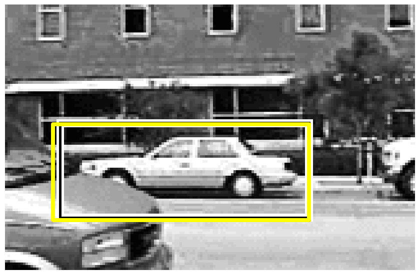


Original test image | HOG descriptor | HOG descriptor weighted by positive SVM weights | HOG descriptor weighted by negative SVM weights

Person detection with HoGs & linear SVMs

- Histograms of Oriented Gradients for Human Detection, Navneet Dalal, Bill Triggs, International Conference on Computer Vision & Pattern Recognition - June 2005
- http://lear.inrialpes.fr/pubs/2005/DT05/

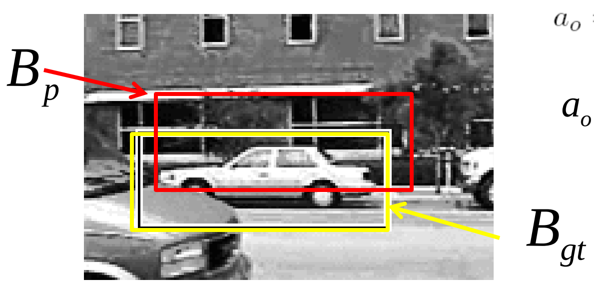
Scoring a sliding window detector



If prediction and ground truth are *bounding boxes*, when do we have a correct detection?

Kristen Grauman

Scoring a sliding window detector



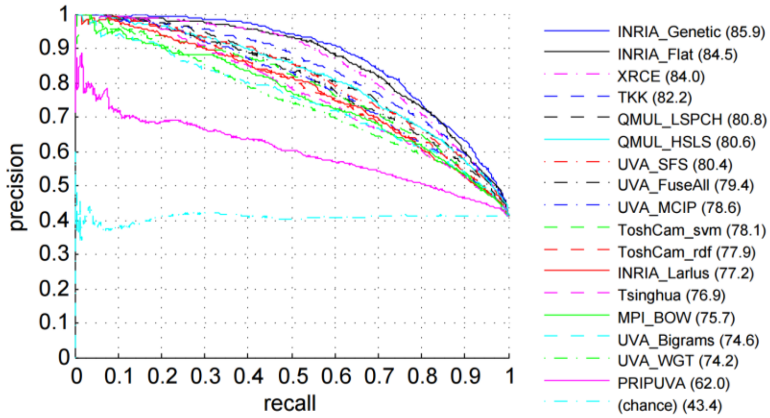
$$a_o = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})}$$

$$a_o > 0.5 \Rightarrow correct$$

We'll say the detection is correct (a "true positive") if the intersection of the bounding boxes, divided by their union, is > 50%.

Kristen Grauman

Scoring an object detector

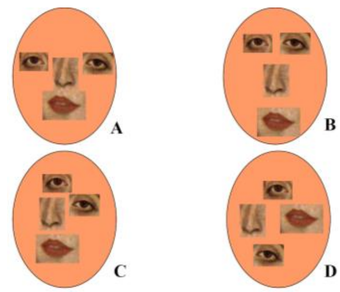


- If the detector can produce a *confidence score* on the detections, then we can plot its precision vs. recall as a threshold on the confidence is varied.
- **Average Precision (AP)**: mean precision across recall levels.

Beyond “window-based” object categories?

Kristen Grauman

Beyond "window-based" object categories?



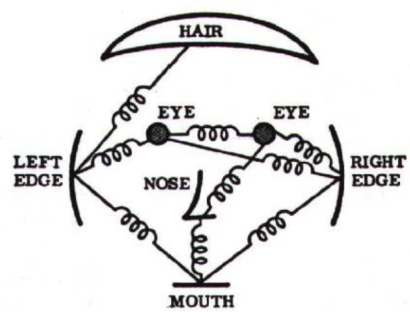
Too much?

Too little?

Slide credit: Kristen Grauman

Part-based models

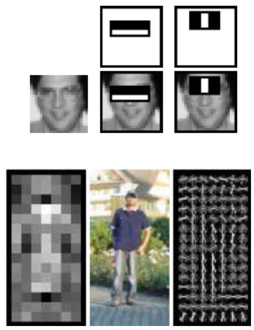
- **Origins in Fischler & Elschlager 1973**
- **Model has two components**
  - **parts (2D image fragments)**
  - **structure (configuration of parts)**



Deformable part model

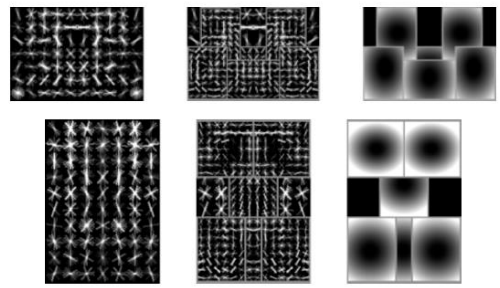
Felzenszwalb et al. 2008

- A hybrid window + part-based model



Viola & Jones
Dalal & Triggs

vs



root filters coarse resolution | part filters finer resolution | deformation models

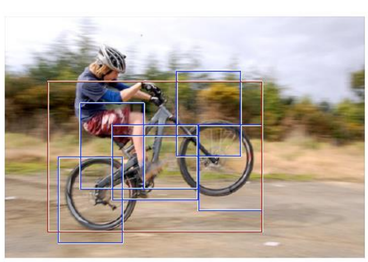
Felzenszwalb et al.

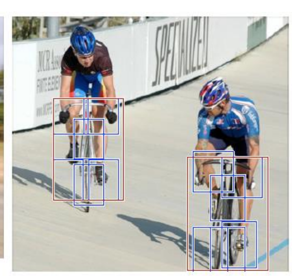
Main idea: Global template ("root filter") plus deformable parts whose placements relative to root are latent variables

Deformable part model

Felzenszwalb et al. 2008

- Mixture of deformable part models
- Each component has global template + deformable parts
- Fully trained from bounding boxes alone

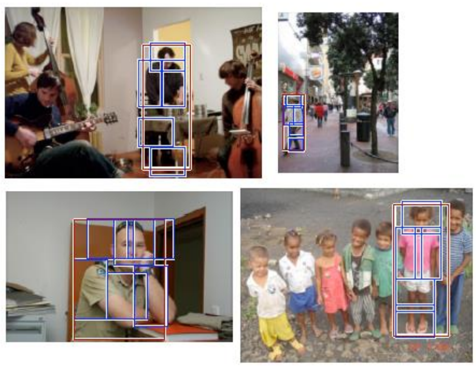




Adapted from Felzenszwalb's slides at http://people.cs.uchicago.edu/~pff/talks/

Results: person detections

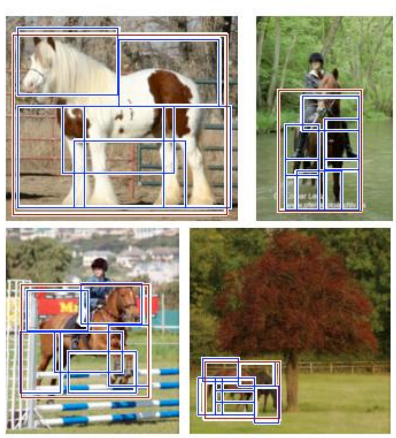
high scoring true positives



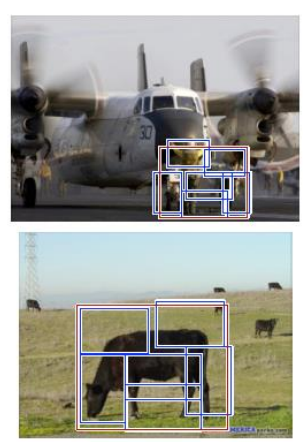
high scoring false positives
(not enough overlap)

Results: horse detections

high scoring true positives



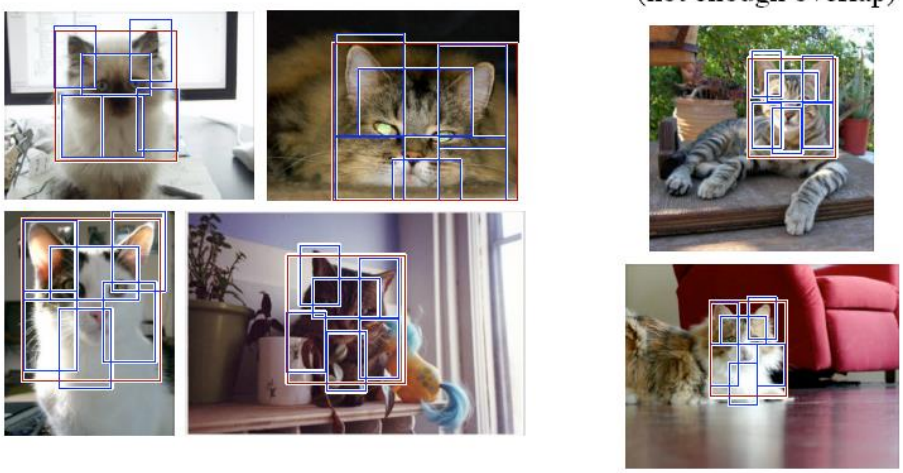
high scoring false positives



Results: cat detections

high scoring true positives

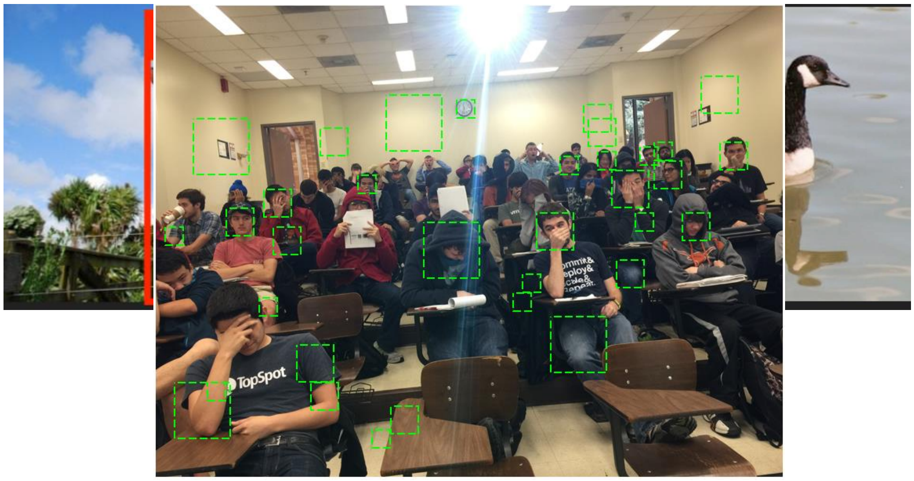
high scoring false positives (not enough overlap)

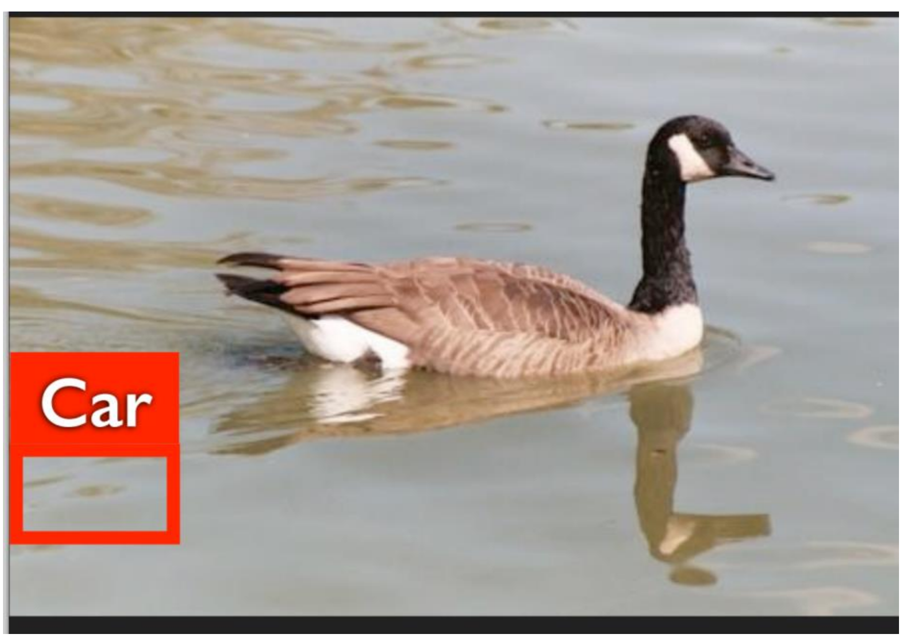


Today

- Support vector machines (SVM)
  - Basic algorithm
  - Kernels
    - Structured input spaces: Pyramid match kernels
  - Multi-class
  - HOG + SVM for person detection
    - Visualizing a feature: Hoggles
- Evaluating an object detector

Understanding classifier mistakes





Carl Vondrick http://web.mit.edu/vondrick/ihog/slides.pdf

What information does HOG have?

Image

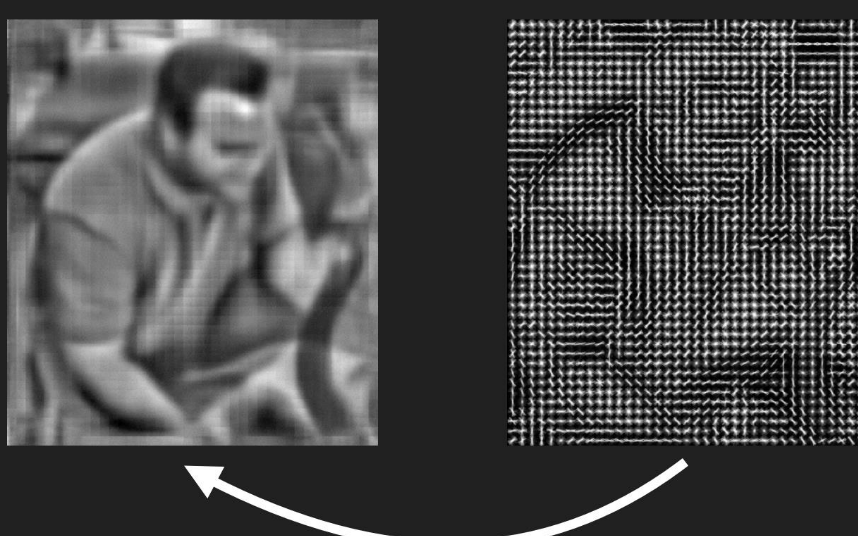
HOG

[redacted]

HOGgles: Visualizing Object Detection Features
Carl Vondrick, MIT; Aditya Khosla; Tomasz Malisiewicz; Antonio Torralba, MIT
http://web.mit.edu/vondrick/ihog/slides.pdf

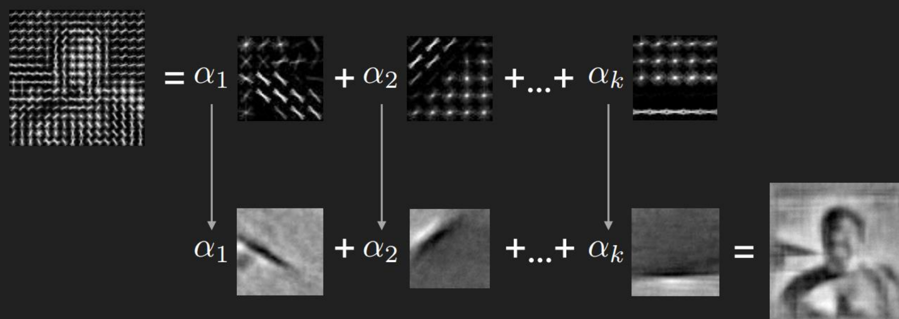
HOGGLES: Visualizing Object Detection Features

What information is lost?



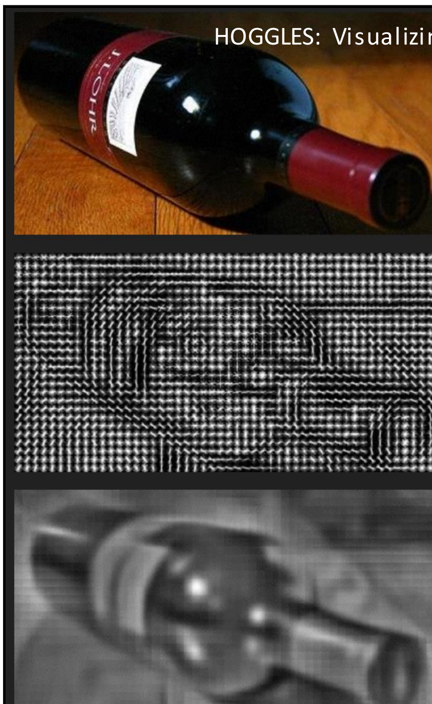
HOGGLES: Visualizing Object Detection Features

Method: Paired Dictionary



HOGgles: Visualizing Object Detection Features
Carl Vondrick, MIT; Aditya Khosla; Tomasz Malisiewicz; Antonio Torralba, MIT
http://web.mit.edu/vondrick/ihog/slides.pdf

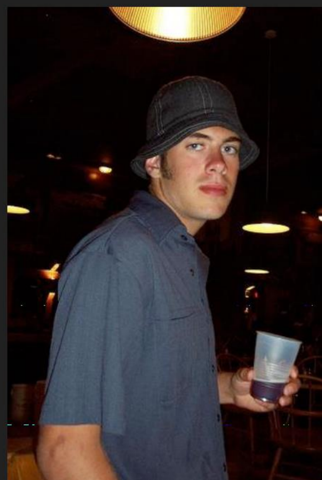
HOGGLES: Visualizing Object Detection Features



A microscope to view HOG

HOGgles: Visualizing Object Detection Features;
Carl Vondrick, MIT; Aditya Khosla; Tomasz Malisiewicz; Antonio Torralba, MIT
http://web.mit.edu/vondrick/ihog/slides.pdf

HOGGLES: Visualizing Object Detection Features



vs

Human Vision

HOG Vision

HOGGLES: Visualizing Object Detection Features

HOGgles: Visualizing Object Detection Features; ICCV 2013
Carl Vondrick, MIT; Aditya Khosla; Tomasz Malisiewicz; Antonio Torralba, MIT
http://web.mit.edu/vondrick/ihog/slides.pdf

Some A4 results

Today

- Support vector machines (SVM)
  - Basic algorithm
  - Kernels
    - Structured input spaces: Pyramid match kernels
  - Multi-class
  - HOG + SVM for person detection
    - Visualizing a feature: Hoggles
- Evaluating an object detector

Recalll: Examples of kernel functions

- Linear: $K(x_i, x_j) = x_i^T x_j$

- Gaussian RBF: $K(x_i,x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$

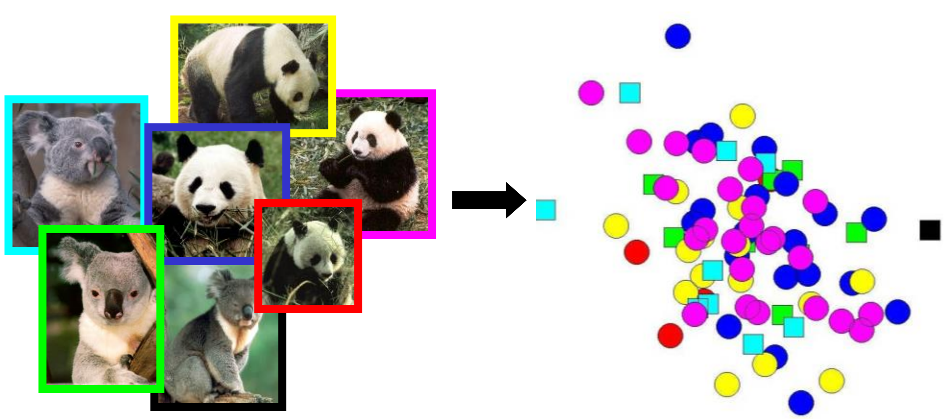
- Histogram intersection:

$$K(x_i, x_j) = \sum_k \min(x_i(k), x_j(k))$$

- Kernels go beyond vector space data
- Kernels also exist for “structured” input spaces like sets, graphs, trees…

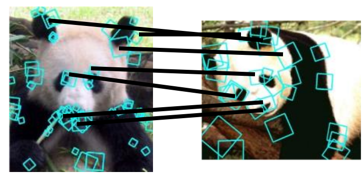
Discriminative classification with sets of features?

- Each instance is unordered set of vectors
- Varying number of vectors per instance



Slide credit: Kristen Grauman

Partially matching sets of features



$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

Optimal match: O(m³)
Greedy match: O(m² log m)
Pyramid match: O(m)

(**m**=num pts)

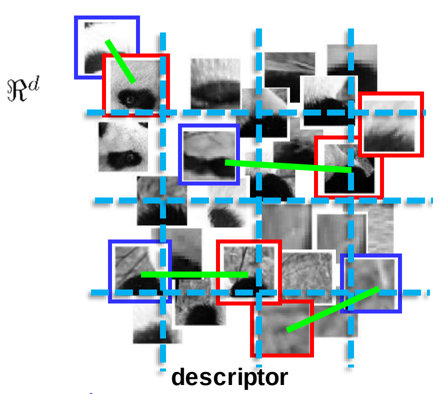
$$\min_{\pi:\mathbf{X}\rightarrow\mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} ||\mathbf{x}_i - \pi(\mathbf{x}_i)||$$

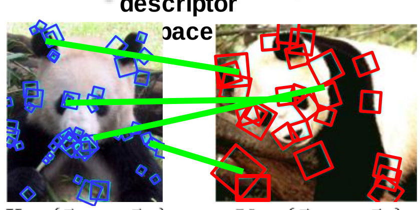
nate matching kernel that makes it practical to compare large sets of features based on their partial correspondences.

[Previous work: Indyk & Thaper, Bartal, Charikar, Agarwal & Varadarajan, …]

Slide credit: Kristen Grauman

Pyramid match: main idea



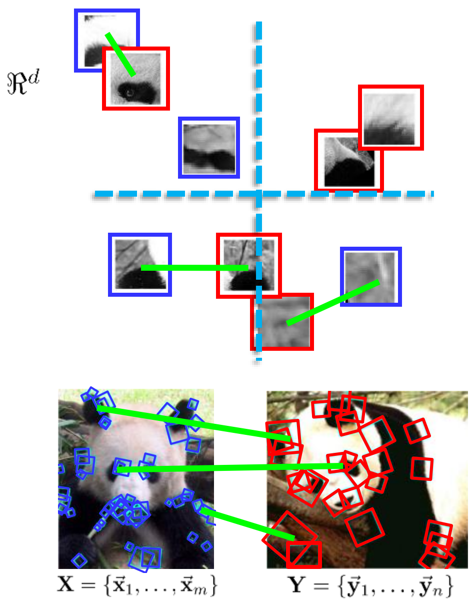


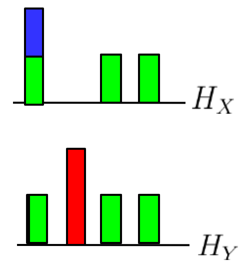
$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

Feature space partitions serve to “match” the local descriptors within successively wider regions.

Slide credit: Kristen Grauman

Pyramid match: main idea





$$\mathcal{I}(H_X, H_Y) = \sum_j \min\left(H_X(j), H_Y(j)\right)$$

$$= 3$$

Histogram intersection counts number of possible matches at a given partitioning.

Slide credit: Kristen Grauman

Pyramid match

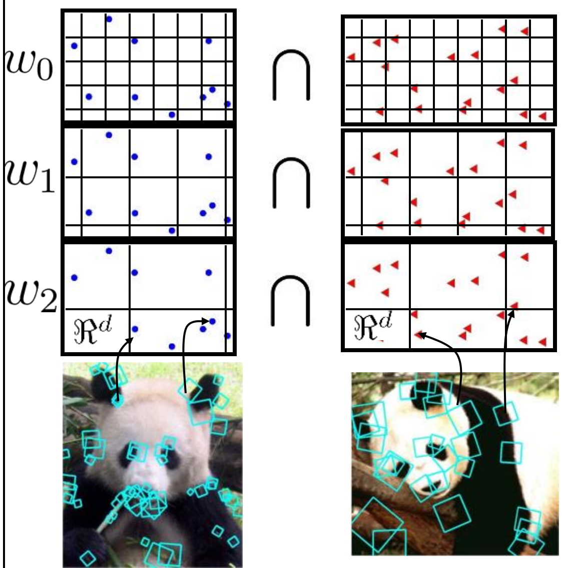
$$K_\Delta(X, Y) = \sum_{i=0}^{L} \underbrace{2^{-i}}_{\text{measures difficulty of a match at level } i} \underbrace{\mathcal{I}\left(H_X^{(i)}, H_Y^{(i)}\right) - \mathcal{I}\left(H_X^{(i-1)}, H_Y^{(i-1)}\right)}_{\text{number of newly matched pairs at level } i}$$

- For similarity, weights inversely proportional to bin size (or may be learned)
- Normalize these kernel values to avoid favoring large sets

[Grauman & Darrell, ICCV 2005]

Slide credit: Kristen Grauman

Pyramid match

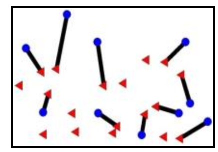


$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\quad$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

Optimal match: $O(m^3)$

Pyramid match: $O(mL)$

$\approx$

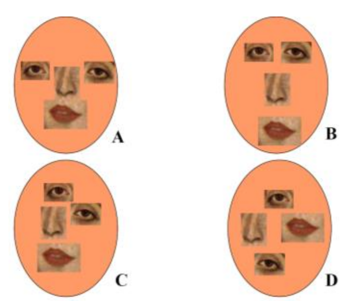


optimal partial matching

The Pyramid Match Kernel: Efficient Learning with Sets of Features. K. Grauman and T. Darrell. Journal of Machine Learning Research (JMLR), 8 (Apr): 725--760, 2007.

BoW Issue: **No** spatial layout preserved!

Too much?

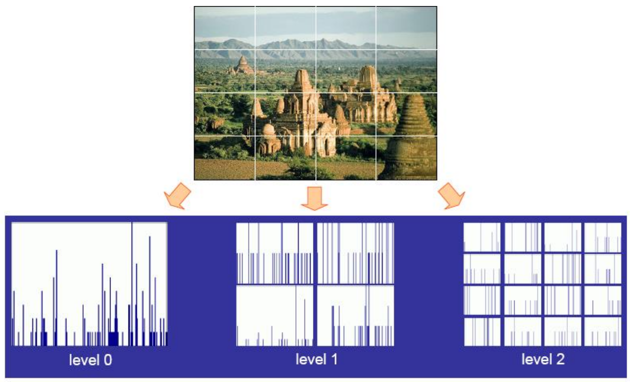


Too little?

Slide credit: Kristen Grauman

Spatial pyramid match

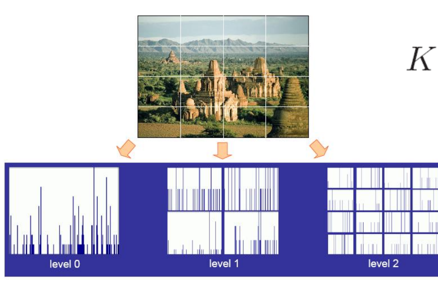
- Make a pyramid of bag-of-words histograms.
- Provides some loose (global) spatial layout information



[Lazebnik, Schmid & Ponce, CVPR 2006]

Spatial pyramid match

- Make a pyramid of bag-of-words histograms.
- Provides some loose (global) spatial layout information



$$K^L(X,Y) = \sum_{m=1}^{M} \kappa^L(X_m, Y_m)$$

Sum over PMKs computed in *image coordinate* space, one per word.

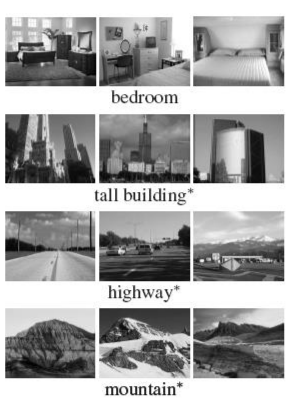
[Lazebnik, Schmid & Ponce, CVPR 2006]

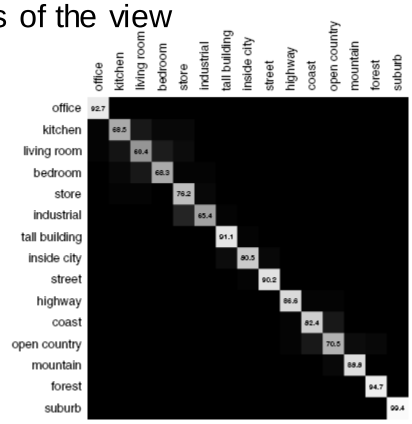
Spatial pyramid match

- Can capture **scene** categories well---texture-like patterns but with some variability in the positions of all the local

Spatial pyramid match

- Can capture **scene** categories well---texture-like patterns but with some variability in the positions of all the local pieces.
- Sensitive to global shifts of the view





Confusion table

Recap: past week

- Object recognition as classification task
  - Boosting (face detection ex)
  - Support vector machines and HOG (person detection ex)
    - Pyramid match kernels
    - Hoggles visualization for understanding classifier mistakes
  - Nearest neighbors and global descriptors (scene rec ex)
- Sliding window search paradigm
  - Pros and cons
  - Speed up with attentional cascade
  - Object proposals as alternative to exhaustive search
- HMM examples
- Evaluation
  - Detectors: Intersection over union, precision recall
  - Classifiers: Confusion matrix

Coming up

- Deep learning and convolutional neural nets
- Attributes and learning to rank