Hyperfeatures –

Multilevel Local Coding for Visual Recognition

Ankur Agarwal & Bill Triggs

GRAVIR-INRIA-CNRS, Grenoble

European Conference on Computer Vision, May 2006

Introduction

- **Target problem**: effectively coding features for recognition in images
- **Approach**: encode feature co-occurence statistics recursively at multiple levels of abstraction
  - use local image patches described by robust descriptors
  - build a hierarchical model starting with texton like representations

Talk outline

- Textons and co-occurence
- Hyperfeatures
- Experimental results on classification

Texton / Bag-of-Feature Approach

- Represent image as a loose collection of individual patches
  - dense, multi-scale, sparse at salient points or edges . . .
- Encode each patch by a vector of local appearance descriptors
  - invariance to lighting, geometric perturbations . . .
- Characterize image / region by the distribution of its patch descriptors

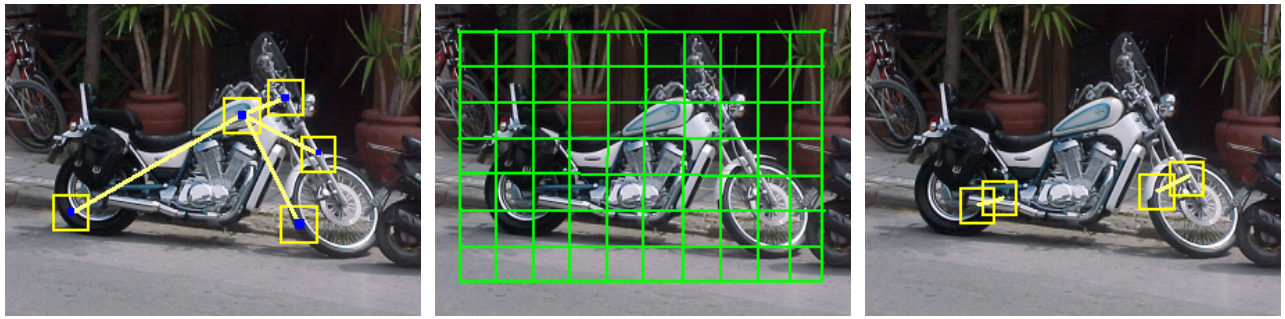
$\{ \times 24, \times 2, \times 6, \times 7, \times 2, \cdots \}$

Capturing Spatial Coherence

- Textons are a very effective model for texture and object images but they encode spatial coherence / object geometry only weakly.

Ways to improve this:

- Explicitly model inter-feature geometry (*e.g.* constellation models)
- Random fields over local labels — MRF, CRF.
- Encoding local (pairwise/neighbourhood) co-occurrence of features.



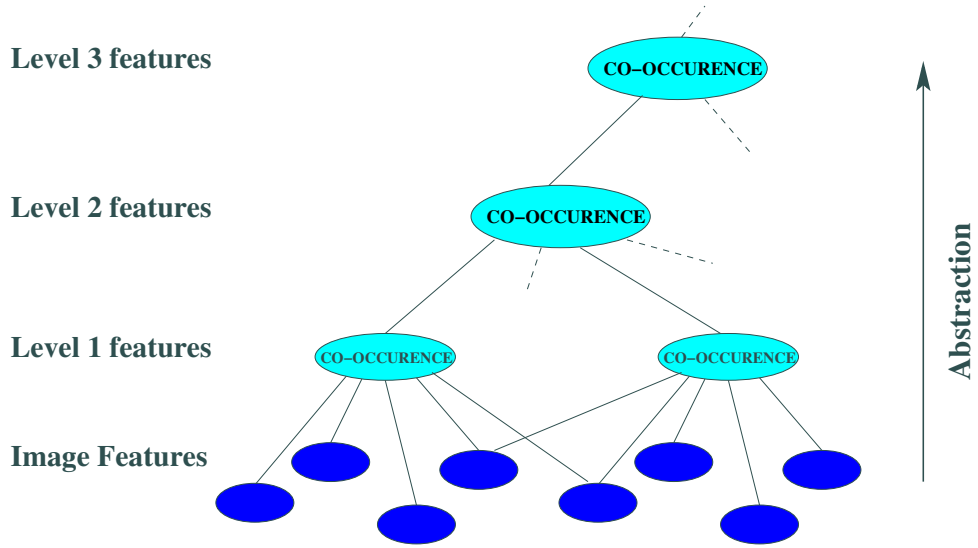
Co-occurence at Multiple Levels

We would like to

- Encode object/scene as a hierarchy of visual parts
- Capture spatial structure *loosely* without precise geometry
- Provide a generic framework to include different feature coding schemes

Cortical suggestiveness

- hierarchical, bottom up, memory based model
- increasing abstraction, less spatial precision in higher levels



Existing Hierarchical Models

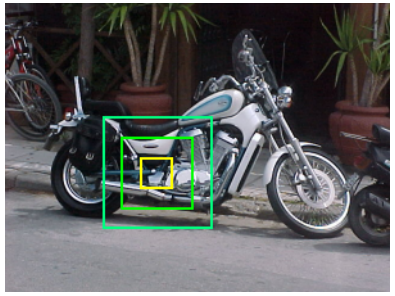
- Take inspiration from biological pattern recognition
- Generally alternating stages of simple and complex cells
- *e.g.* Neocognitron, Convolutional Neural Networks, HMAX ..
  - Neocognitron activates higher level if atleast one cell is active
  - CNNs use a bank of convolution filters against learned templates
  - HMAX performs a *max* operation to retain only dominant signal
- They are all discriminative models

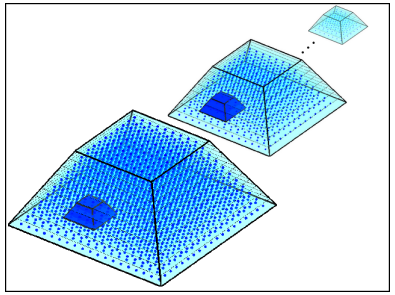
Using co-occurence, we can code *descriptive* statistics and allow for more sophisticated nonlinear codings

Hyperfeatures

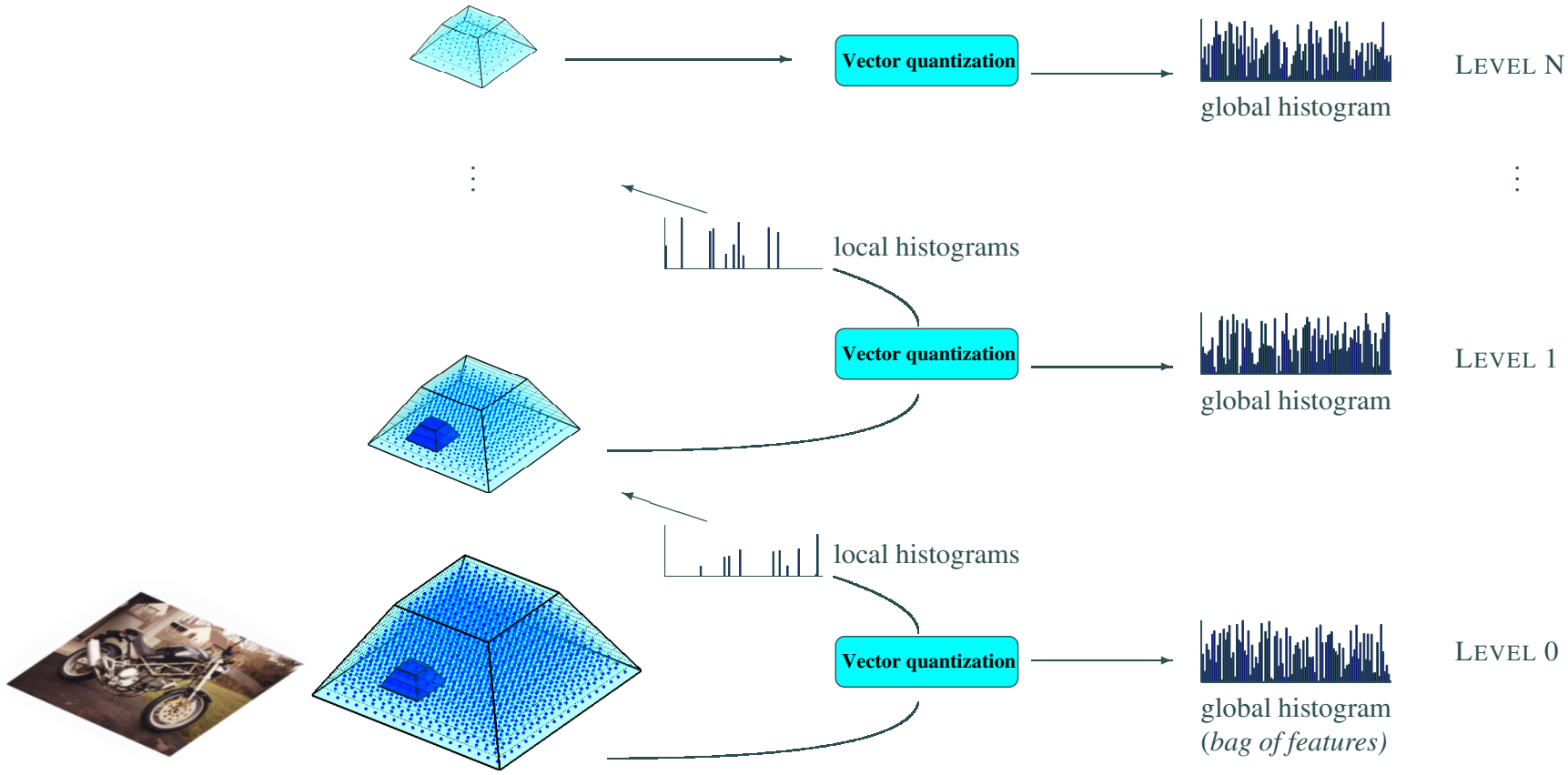
A general principle for multi-level coding: recursively encode neighbourhood co-occurence statistics at multiple levels of abstraction

- **Level 0**: bag of multiscale features
  - Each point is a base feature vector
    e.g. a SIFT descriptor
- **Level 1**: locally collect neighbourhood statistics of these feature vectors
  - Each point is a higher level (feature) vector
- **Repeat recursively** for higher levels to obtain *hyperfeatures*
  - *cf.* the layers of hypercolumns in the cortex





Hyperfeatures for Image Classification



- Global histograms from each level may be fed into a classifer

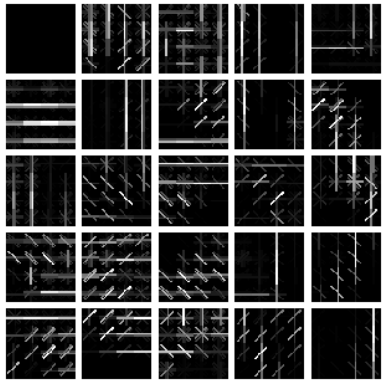
Feature codings at individual levels

Base image patch descriptors

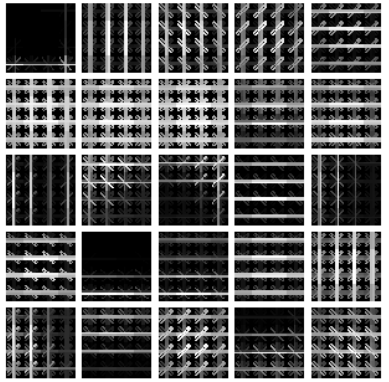
- SIFT-like gradient orientation histograms
  - evaluated on a regular (multiscale) grid without rotation normalization

Distributional coding methods

- Vector Quantization (VQ)
- Gaussian Mixture (GM)
  - training using EM, diagonal covariance model
- Latent Dirichlet Allocation (LDA)
  - document $\rightarrow$ topic $\rightarrow$ word



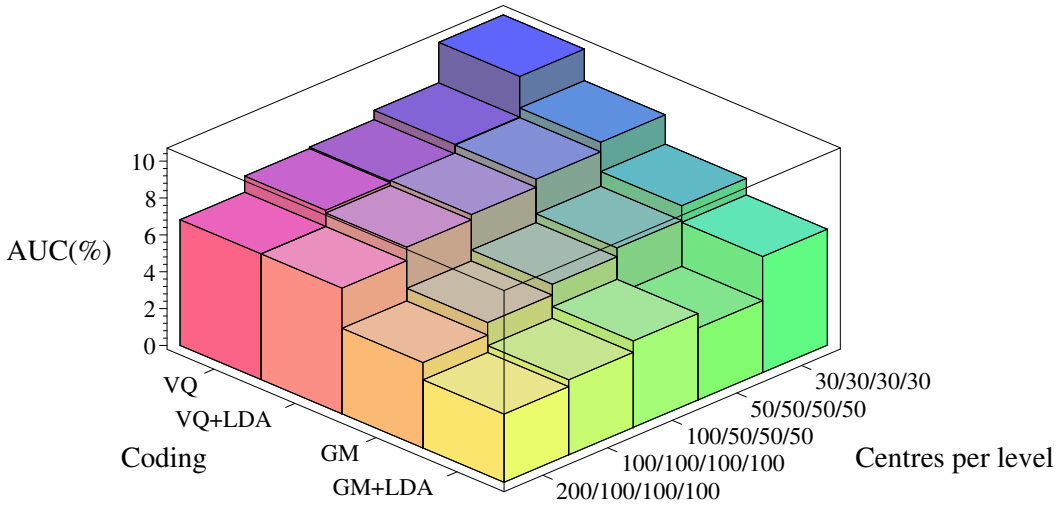
VQ centers



GM centers

Effect of different coding methods

Image Classification using a linear SVM on hyperfeatures

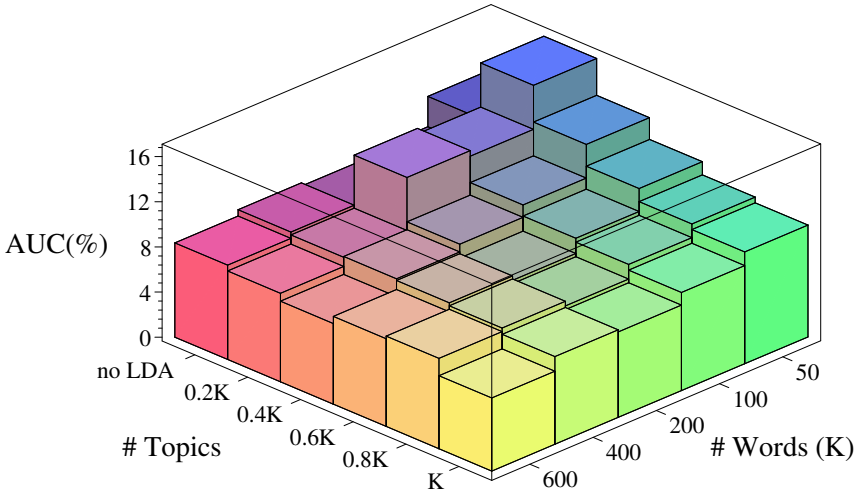


- Performance improves with increasing codebook size
- Gaussian mixtures consistently outperform vector quantization
- Performing LDA futher improves results

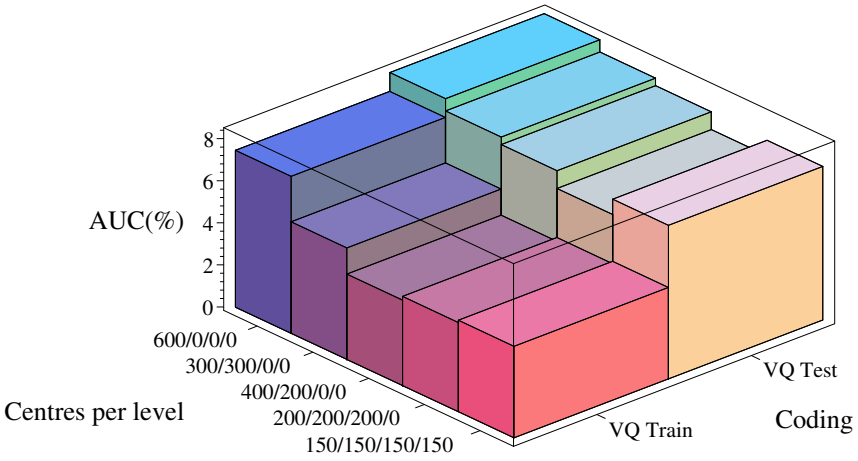
Codebook sizes and hyperfeatures

Number of topics in LDA

- More topics always useful
- Larger vocabularies better



How to distribute centres

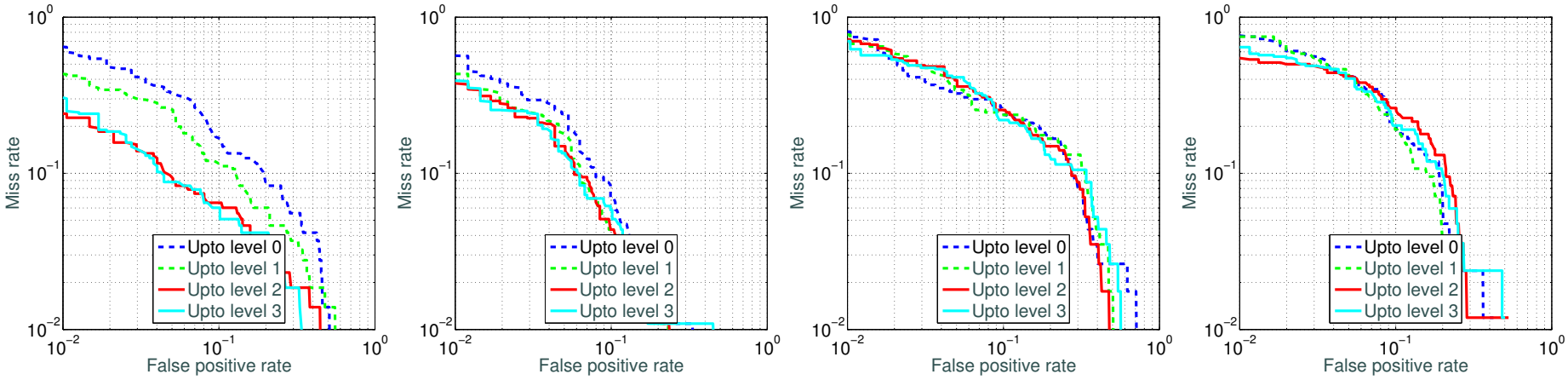


- Distributing centres across levels beneficial
- Too many levels harmful

Classification of object categories

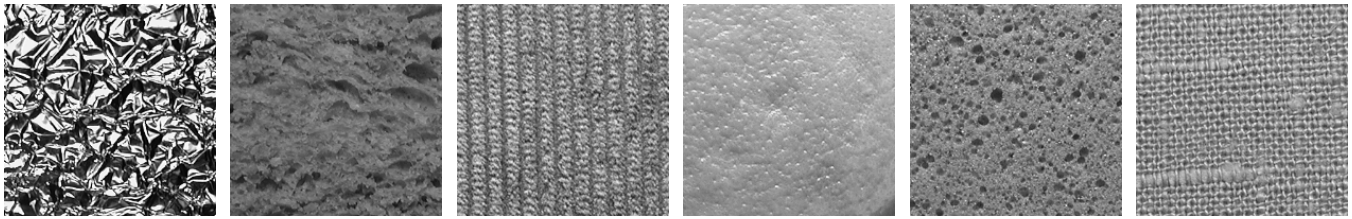
- Linear SVM classifier applied to higher level features, GM coding
- *Structured* objects like motorbikes and cars benefit most
  - saturation after 2-3 extra levels

(a) Motorbikes (b) Cars (c) Bicycles (d) People



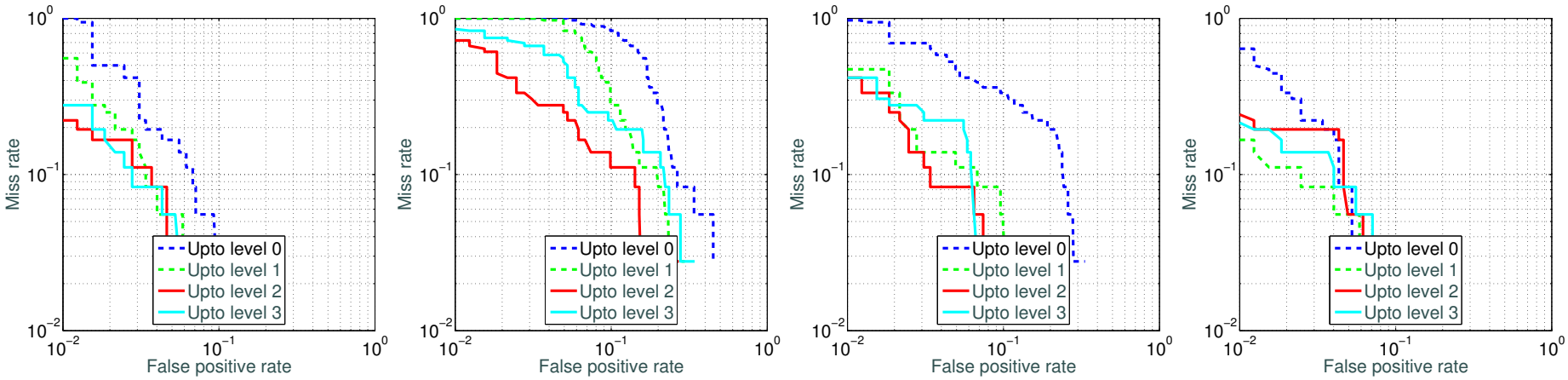
Detection-Error-Tradeoff curves, PASCAL VOC2005 dataset

Classification of textures



- Perfect classification on some textures using simple bag-of-features
- Including 2 or 3 levels of hyperfeatures benefical for rest of classes

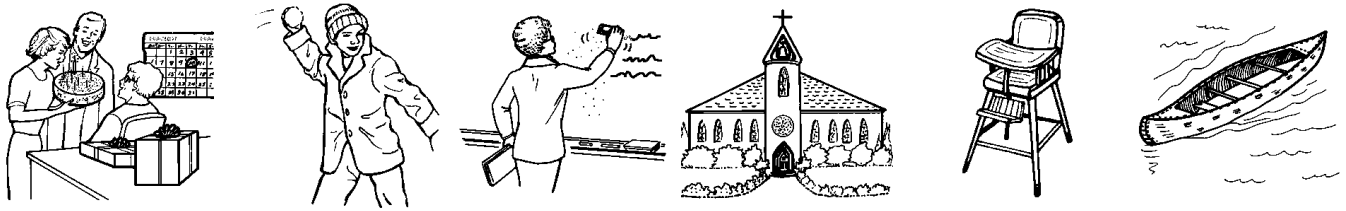
(a) Aluminium foil (b) Cracker (c) Orange peel (d) Sponge



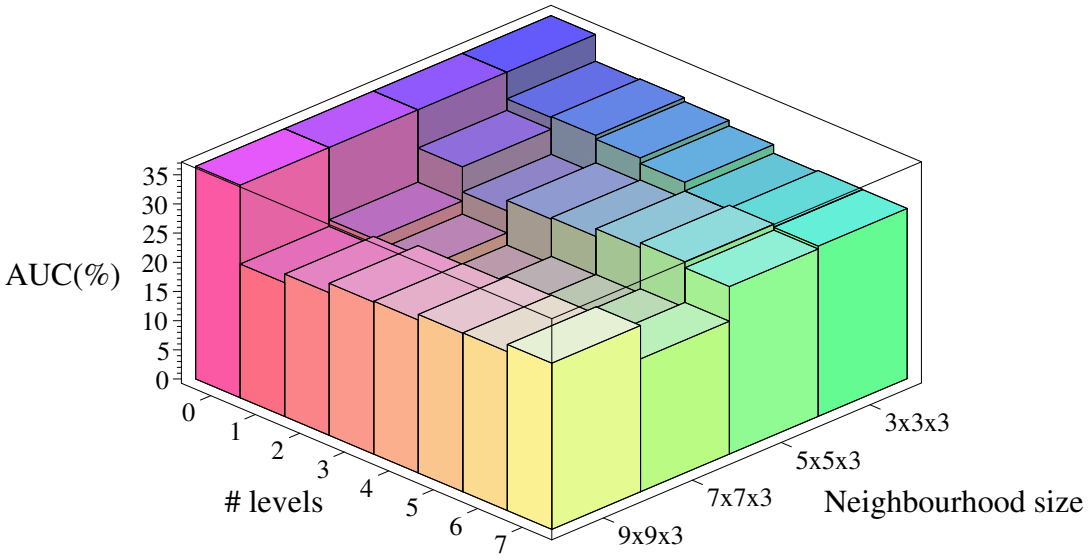
Detection-Error-Tradeoff curves, KTH-TIPS texture dataset

Experiments on Line Drawings

Picture naming project in language research:
Classifying drawings of people from objects/scences



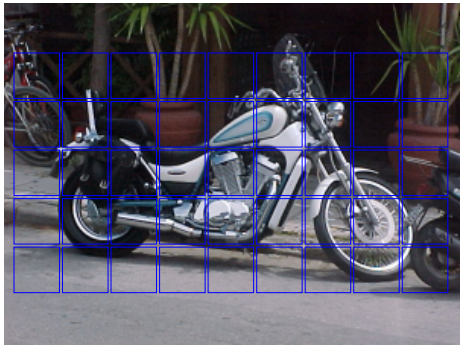
- Extent of local spatial aggregation is important
- More levels needed for smaller neighbourhoods



Classifying Local Image Regions

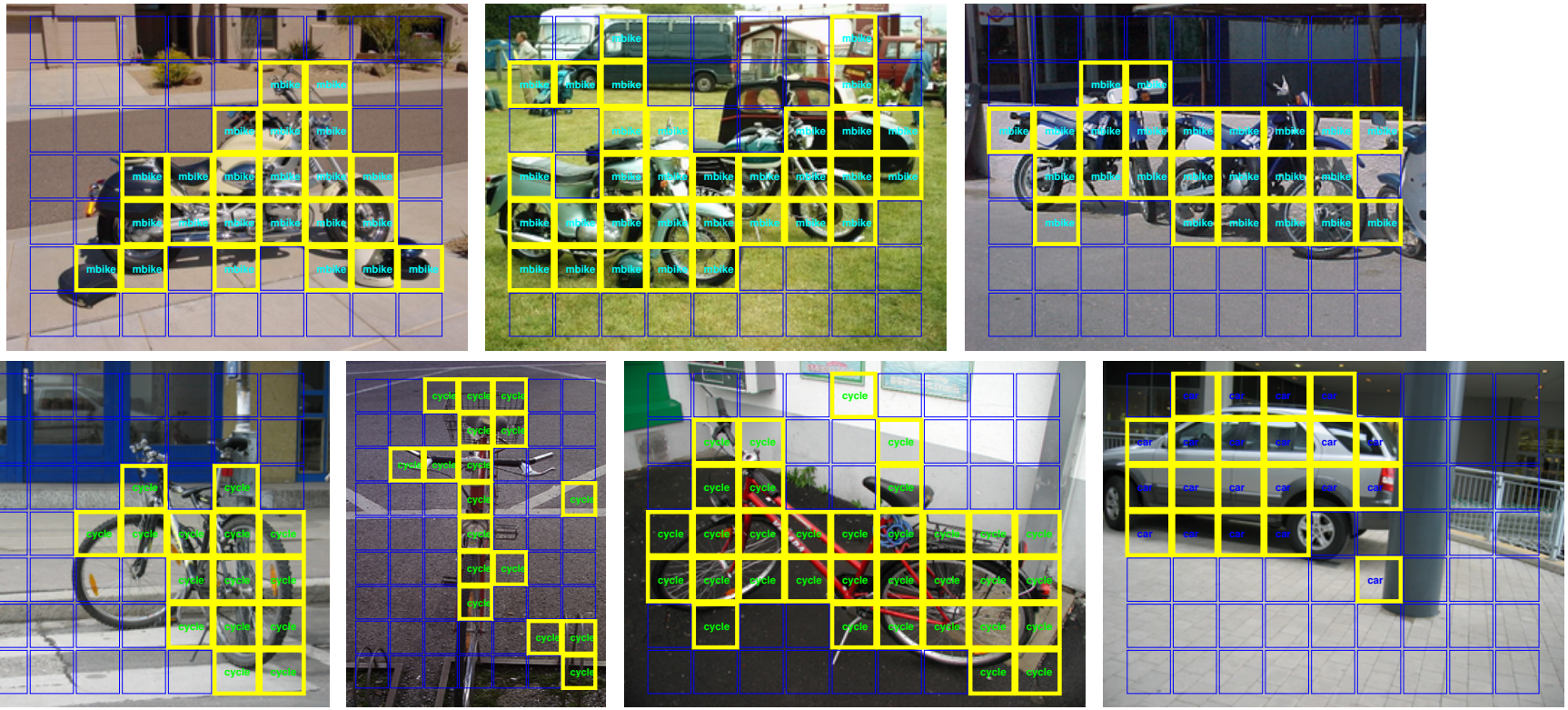
Despite spatial aggregation, high level hyperfeatures remain local

- Construct hyperfeatures individually for each local region of image
  - build a “mini-pyramid” for each region
- Train a *patch classifier* using training bounding boxes
  - patches on object treated as positive
  - patches on background and other classes as negative
- Label each test image region for localizing object parts



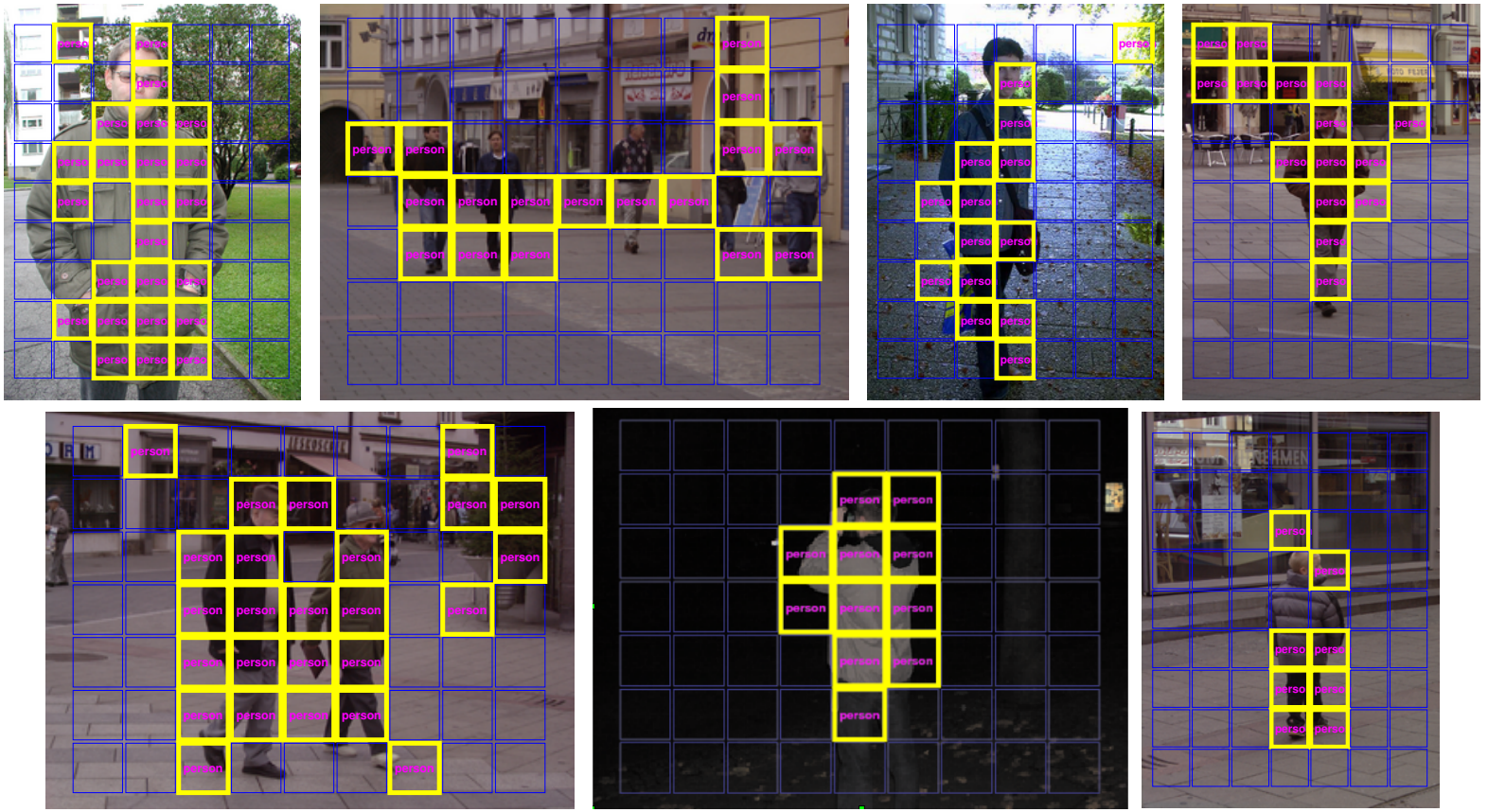
Classifying Local Image Regions – examples

Localizing object parts in images



Classifying Local Image Regions – examples (2)

Localizing human body parts in images



Scalability

Algorithm used in above experiments

- Learns codings one level at a time
- Fixes codebook for lower level before advancing to higher level
- Requires all training features in memory during learning

Alternative algorithm

- Learns all levels in parallel
- Processes training images sequentially to avoid large memory usage
  - uses online EM to learn Gaussian mixture codebooks
- A single pass of training data gives very similar results to first algorithm
- Usable on arbitrarily large datasets

Conclusions

- Hyperfeatures encode spatial information in images, but without a rigid representation
  - –loose structure coded using co-occurence statistics
- Multiple levels of recursive coding improve classification performance in many cases
  - – object classes with distinctive geometric structure benefit most

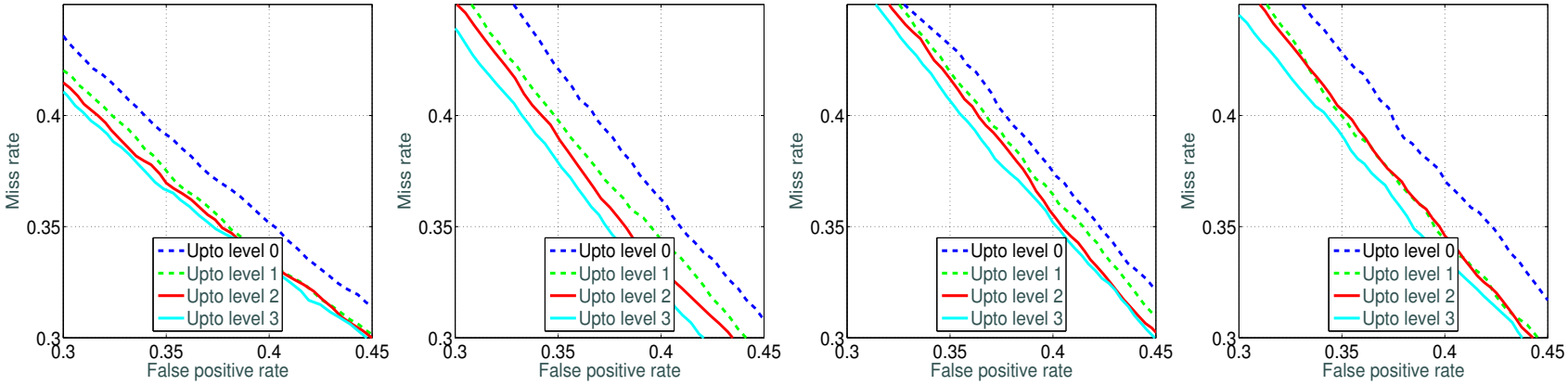
Possible extensions

- Investigation of more discriminative training methods
  - – more general latent aspect models that use local context
- Integration with processes like CRF based segmentation for improved object localization

Thank you

Local region classification performance

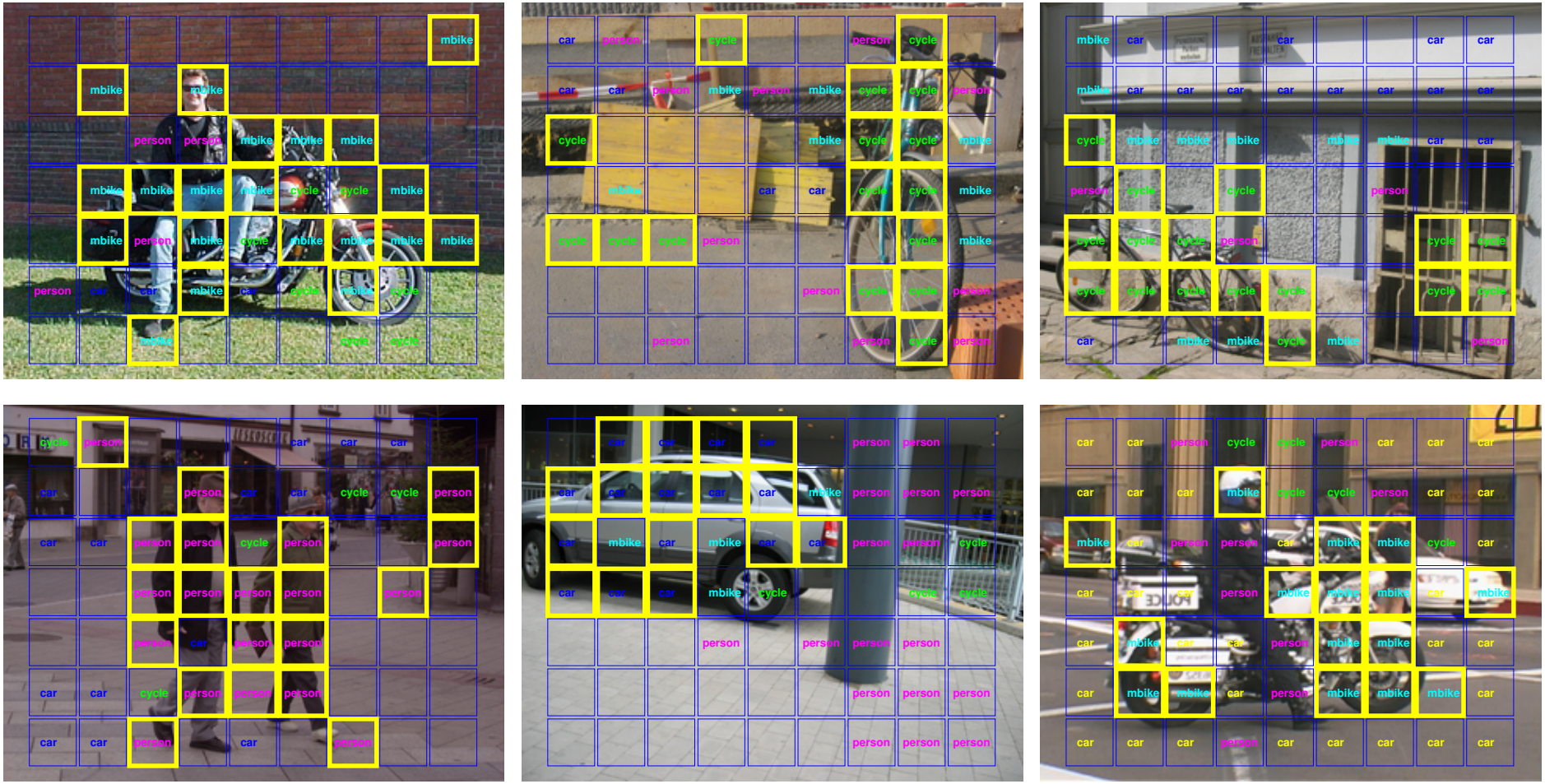
(a) Motorbikes (b) Cars (c) Bicycles (d) People



- adding higher levels improves discrimination of local regions.

Hyperfeatures – Combining Multiple Classifiers

Classifying each region as 1 of 4 classes (no explicit background model)



Hyperfeature coding algorithm – *offline* version

1. $\forall(i,x,y,s)$, $\mathcal{F}^{(0)}_{ixys} \leftarrow$ base feature at point (x,y), scale s in image i.
2. For $l = 0,\ldots,N$:
   - If learning, cluster $\{\mathcal{F}^{(l)}_{ixys} \,|\, \forall(i,x,y,s)\}$ to obtain a codebook of $d^{(l)}$ centres in this feature space.
   - $\forall i$:
     - If global descriptors need to be output, code $\mathcal{F}^{(l)}_{i\ldots}$ as a $d^{(l)}$ dimensional histogram $\mathcal{H}^{(l)}_i$ by globally accumulating votes for the $d^{(l)}$ centers from all (x,y,s).
     - If $l < N$, $\forall(x,y,s)$ calculate $\mathcal{F}^{(l+1)}_{ixys}$ as a $d^{(l)}$ dimensional local histogram by accumulating votes from $\mathcal{F}^{(l)}_{ix'y's'}$ over neighbourhood $\mathcal{N}^{(l+1)}(x,y,s)$.
3. Return $\{\mathcal{H}^{(l)}_i \,|\, \forall i,l\}$.

Hyperfeature coding algorithm – *online* version

1. Initialization: run algorithmn 1 using a very small subset (*e.g.* 10) of the training images
2. Update codebook centres at all levels: $\forall i$:
   - For $l = 0, \ldots, N$:
     - perform one iteration of k-means to update the $d^{(l)}$ centers using $\{\mathcal{F}^{(l)}_{ixys} \,|\, \forall (x, y, s)\}$.
     - If $l < N$, $\forall (x, y, s)$ calculate $\mathcal{F}^{(l+1)}_{ixys}$ as in algorithm 1.
3. If centers have not converged, repeat step 2. Else $\forall i$:
   - For $l = 0, \ldots, N$:
     - If $l < N$, $\forall (x, y, s)$ calculate $\mathcal{F}^{(l+1)}_{ixys}$ as a $d^{(l)}$ dimensional local histogram by accumulating votes from $\mathcal{F}^{(l)}_{ix'y's'}$ over neighbourhood $\mathcal{N}^{(l+1)}(x, y, s)$.
     - If global descriptors need to be output, code $\mathcal{F}^{(l)}_{i\ldots}$ as a $d^{(l)}$ dimensional histogram $\mathcal{H}^{(l)}_i$ by globally accumulating votes for the $d^{(l)}$ centers from all (x, y, s).
4. Return $\{\mathcal{H}^{(l)}_i \,|\, \forall i, l\}$.