Fast Discriminative Visual Codebooks using Randomized Clustering Forests

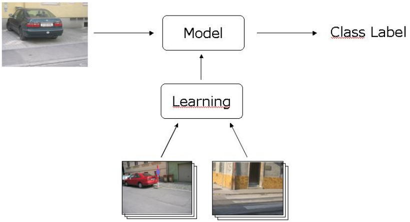
Fredéric Jurie, CNRS - INRIA.
Joint work with F. Moosmann and B. Triggs

To appear in NIPS'06

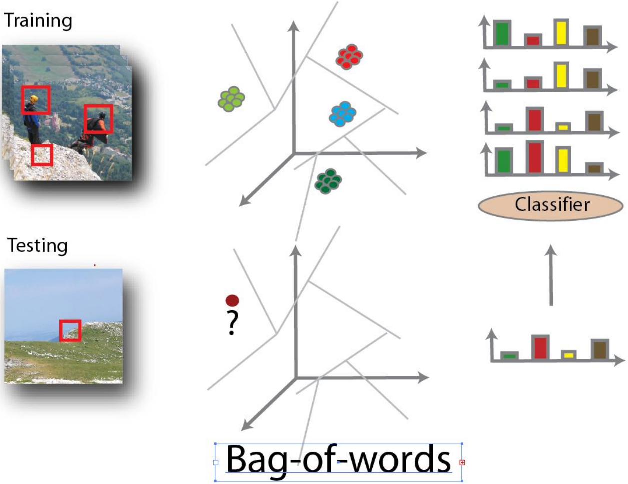
Introduction

Scene categorization

Object Class categorization

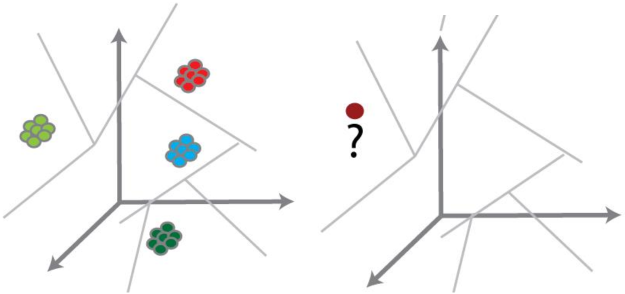


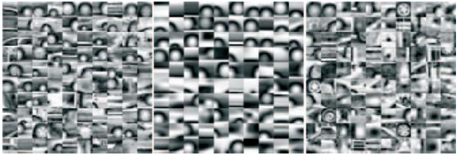
- Challenges:
  - Classes defines by pure 2D-Images
  - High Inner-Class Variance
  - Low Intra-Class Distance
  - Need for robustness towards transformations / illumination changes
  - Large number of images


Training
Testing
?
Classifier
Bag-of-words

Visual Dictionaries

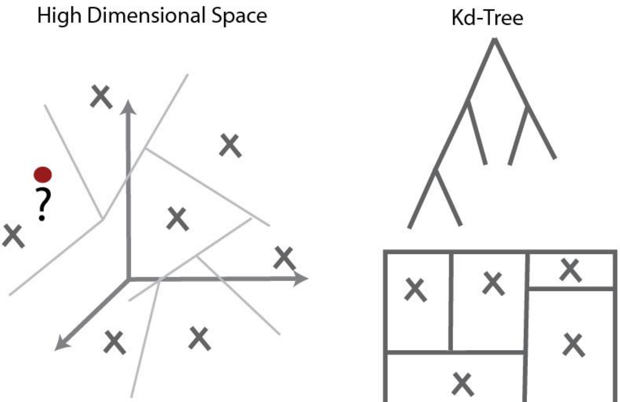
- *Visual Dictionary* = any process (labels $\rightarrow$ descriptors)



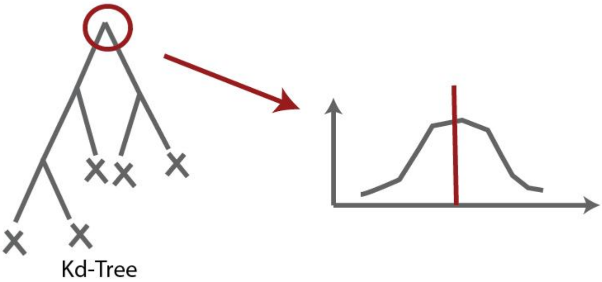


- Creating visual codebooks
  $\longrightarrow$ various methods, $\longrightarrow$ influence on performance.
- K-means clustering: currently the most common [Csurka *et al.*, ECCV-WSLCV, 2004], [Sivic *et al.*, ICCV'03]
- Mean-shift [Jurie *et al.*, ICCV'05] clusterers $\longrightarrow$ advantages.
- Common properties
  - Unsupervised (i.e. class labels are not used in the clustering process)
  - Complexity: at least $\sim$ number of dimensions of the feature space.

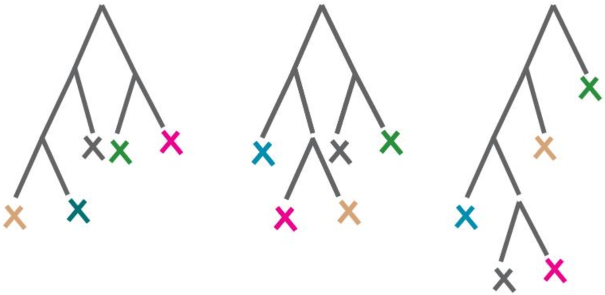
How to be faster: using kd-trees



Kd-trees drawbacks: instability



Adding robustness: ensemble of kd-trees

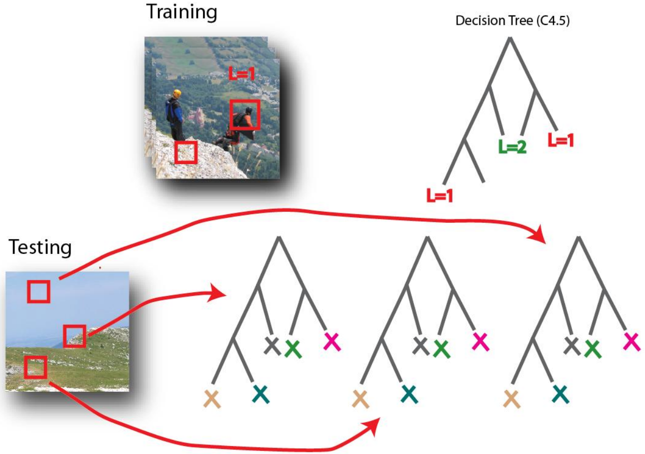


Alternate solutions: [Nister, CVPR'06]: tree coding (hierarchical K-means), uses all components; compromise between speed and loss of accuracy.

- Supervised / Adapted vocabularies
  - [Perronnin *et al.*, ECCV'06]: universal vocabulary adapted for each class.
  - [Winn *et al.*, ICCV'05], large vocabulary → optimal words (mergin GMM).
- Impressive results - computationally expensive (cost of assignation)
- How to be faster?
  - Tree based coding [Nister, CVPR'06] [Lepetit05 *et al.*, CVPR'05] faster but less discriminant.
  - How to achieve both Speed and Good Discrimination?

Decision Trees

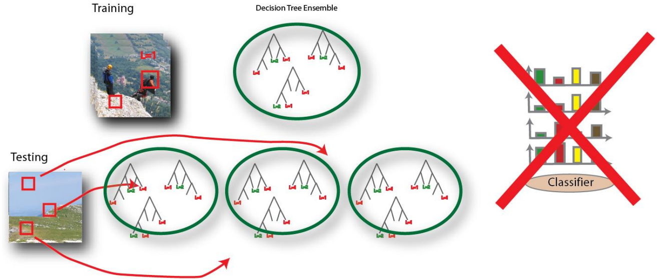
Decision trees for classifing images.



→ instable.

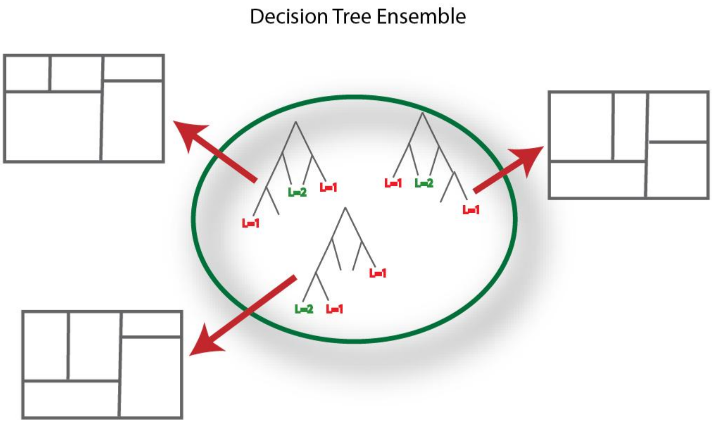
Decision Trees

Ensemble of randomized decision trees.[Geurts et al., ML 2006]

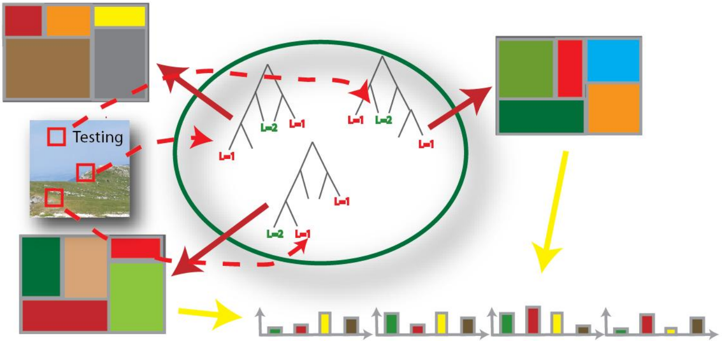


→ Good results, but not as good as 'bag-of-words' approaches.

Decision Trees as clustering trees



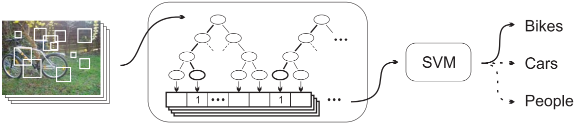
Decision Trees as clustering trees



Main contributions

- Contributions:
  - Small ensembles of decision trees:
    - eliminate many of the disadvantages of single tree based coders
    - without losing the speed advantages of trees.
  - Decision trees: valuable information about locality in descriptor space ($\neq$ class labels).
  - Training tree for classification; ignore class labels;
    - *clustering trees*
    - simple spatial partitioners that assign a distinct label (visual word) to each leaf.
- We show that:
  - Good resistance to background clutter
  - Much faster: for training and testing,
  - More accurate results (than conventional methods like k-means)

Overall framework



Using ERC-Forests as visual codebooks
in bag-of-feature image classification.

Extremely Randomized Clustering Forests (ERC-Forests)

- $\mathbf{d} = (f_1, \ldots, f_D)$, where $f_i, i = 1, \ldots, D$ are elementary scalar features.
- y is the same for all descriptors from a given image.
- training: using a labeled (for now) training set $L = \{(\mathbf{d}_n, y_n), n = 1, \ldots, N\}$.
- Leaves labels: unique leaf indices, not the descriptor labels y associated with the leaves.

Building ERC-Trees.

- Trees construction: recursively top down.
- Each node t
  - descriptor space region $\mathcal{R}_t$,
  - two children l, r = boolean test $\mathcal{T}_t$
  - $\mathcal{R}_t = \mathcal{R}_l \cup \mathcal{R}_r$ with $\mathcal{R}_l \cap \mathcal{R}_r = \phi$.
- Recursion until further subdivision is impossible.
- Thresholds on elementary features $\mathcal{T}_t = \{f_{i(t)} \leq \theta_t\}$
  - Index $i(t)$ is chosen randomly
  - Threshold θ_t is sampled randomly from a uniform distribution
  - resulting node is scored (Shannon entropy)
  - High scores indicate that the split separates the classes well.
- Procedure repeated (threshold $S_{\min}$, maximum number $T_{\max}$ of trials).
- $\mathcal{T}_t$ with highest score adopted
- Pruning

Computational complexity.

- Worst-case complexity for building trees: is $O(T_{\max} Nk)$
- Cannot guarantee balanced trees, but in our experiments on real data we always obtained well balanced trees.
- Practical obseved complexity of around $O(T_{\max} N \log k)$.
- Dependence on data dimensionality:
  - D: hidden in the constant $T_{\max}$ 'filter out irrelevant feature dimensions), better coding, more balanced trees.
  - $T_{\max} \sim O(\sqrt{D})$ has been suggested [Geurts et al., ML 2006], leading to a total complexity of $O(\sqrt{D} N \log k)$.
  - k-means complexity: $O(DNk)$ $-10^4\times$ more for our 768-D wavelet descriptor with $N = 20000$ data points and $k = 5000$ clusters (not counting the number of iterations that k-means has to perform.)
- Faster in use: $O(\log k)$ (k-means costs $O(kD)$)

Experiment Settings

- Visual descriptors.
  - Color descriptor: raw HSL color pixels - 768-D feature vector (16×16 pixels × 3 colors).
  - Color wavelet descriptor: 768-D vector using a 16×16 Haar wavelet transform.
  - Grayscale SIFT descriptor [Lowe, IJCV'04][Marszalek, CVPR'06]: returns 128-D vectors (4×4 histograms of 8 orientations).
- ROC curves classification rates at EER.
- means and variances, 10 learning runs.
- We use $S_{\min} = 0.5$. The exact value is not critical.
- $T_{\max}$: a significant influence, validation set. For the 768-D Color Wavelet: $\rightarrow T_{\max} \approx 50$.

Databases

- 4 different databases. GRAZ-02 test set.
- Bicycles (B), cars (C), persons (P) – and negatives (N).
- Illumination: is highly variable
- Objects: different perspectives and scales, occluded.
- Background: neutral (weak influence of context).

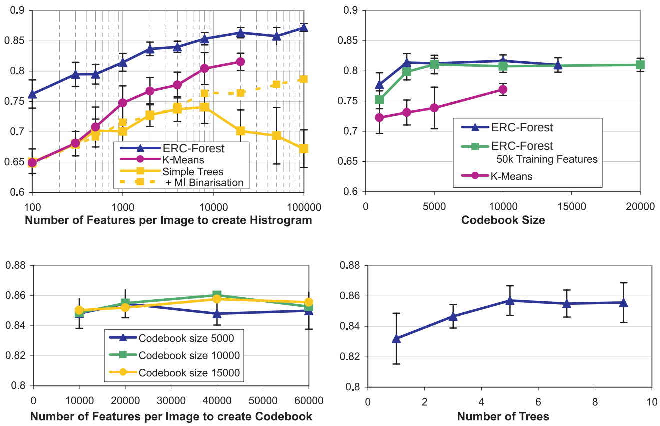
Comparing our random forest with k-mean and kd-clustering trees.

- Individual object categories versus negatives (N).
- 300 images from each category
- Two settings:
  - *Setting 1* did not use the (available) segmentation masks ([Opelt *et al.*, SCIA'05])
  - *Setting 2* uses the provided masks

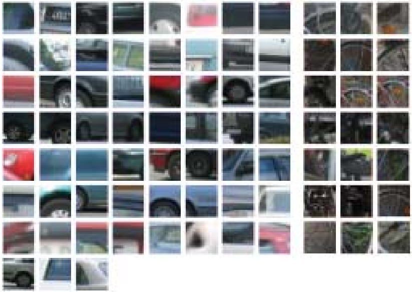
Comparing our random forest with k-mean and kd-clustering trees.

- B vs. N we achieve:
  - 84.4% average EER classification rate for setting 1
  - 84.1% for setting 2,
  - in comparison to 76.5% from [Opelt *et al.*, SCIA'05].
- For C vs. N the respective figures are
  - 79.9% setting 1,
  - 79.8% setting 2
  - in comparison to 70.7% from [Opelt *et al.*, SCIA'05].
- Segmentation masks: not improving results

Comparing our random forest with k-mean and kd-clustering trees.



Visual codebook

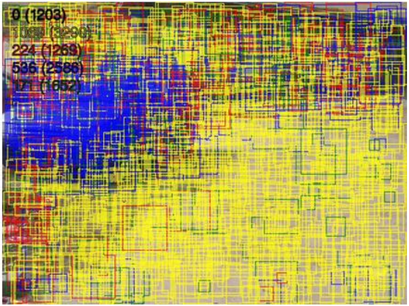


Test patches that were assigned to a particular 'car' leaf (left) and a particular 'bike' one (right).

Conclusions

- Bag-of-words: state-of-the art results, but
  - quantization large numbers of high-dimensional descriptors
  - cluster quality
- Decision trees used as descriptor-space partitions
- *Extremely Randomized Clustering Forests*: rapid, highly discriminative, out-performs k-means based coding
  - training time
  - memory
  - testing time
  - classification accuracy.
- Promising approach for visual recognition, may be beneficial to other areas such as object detection and segmentation.
- Resistant to background clutter: clean segmentation and “pop-out” of foreground classes

Perspective: Biased Sampling



Perspective: Biased Sampling