Content-based Image Retrieval by Indexing Random Subwindows with Randomized Trees

Raphaël Marée, Pierre Geurts, Louis Wehenkel

GIGA Bioinformatics Platform
Dept. EE & CS (Montefiore Institute)
University of Liège
Belgium

ACCV, 22^{th} November 2007, Tokyo

Content-Based Image Retrieval (CBIR)

- Goal
  - Given a reference database of *unlabeled* images, retrieve images similar to a new query image based only on visual content.

- Challenges
  - To be robust to uncontrolled conditions
  - To be fast (efficient indexing structures) and accurate (rich image descriptions)
  - To avoid tedious manual adaptation specific to a task

Content-Based Image Retrieval (CBIR)

- Goal
  - Given a reference database of *unlabeled* images, retrieve images similar to a new query image based only on visual content.

- Challenges
  - To be robust to uncontrolled conditions
  - To be fast (efficient indexing structures) and accurate (rich image descriptions)
  - To avoid tedious manual adaptation specific to a task

Université de Liège

Starting point: our method at CVPR05

- Image classification with *labeled* training images and single class prediction

- Fast method
  - Random subwindow extraction
  - Extremely randomized decision trees [Geurts et al. 2006]
- Good accuracy results on various tasks

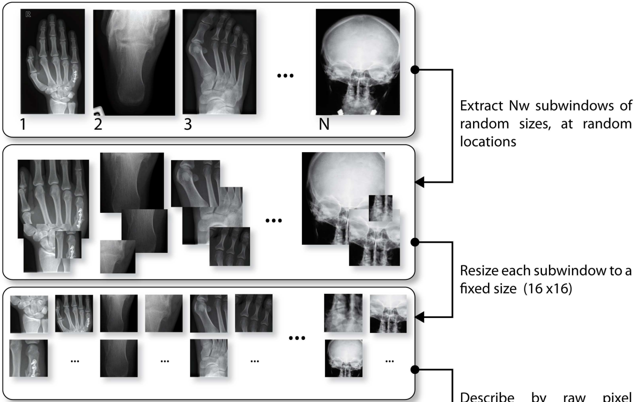
Université de Liège

This work: extension for CBIR

Overview

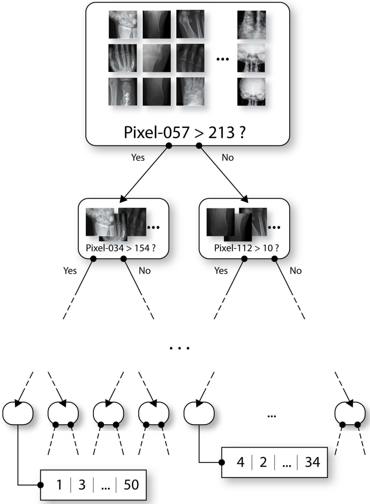
- **Detector**: *random* subwindows
- **Descriptor**: subwindow raw pixel values
- **Indexing subwindows**: *totally randomized* trees
- **Image similarity measure**: derived from similarity measure between subwindows defined by trees

Extraction of Random Subwindows

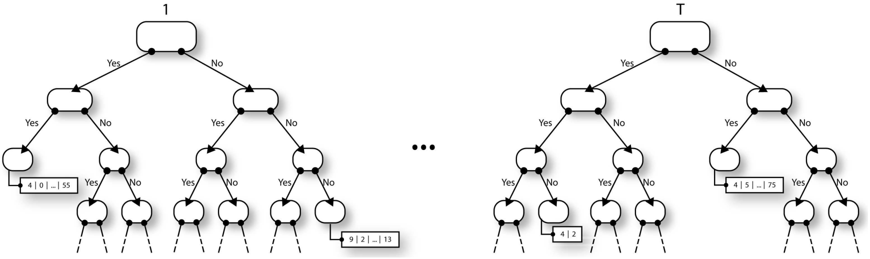


	Pixel-001	Pixel-002		Pixel-256
Subwindow-00	34	201		255
Subwindow-01	78	31		204
Subwindow-02	145	167		32
...				
Subwindow-Nw	14	4		134

Indexing subwindows with one Totally Randomized Tree



Indexing subwindows with an Ensemble of T Trees



- Parameters
  - T: the number of totally randomized trees
  - n_{min}: the minimum node size, stop-spliting of a node if $\#node < n_{min}$

Similarity between two subwindows (one tree)

A tree $\mathcal{T}$ defines a similarity between two subwindows s and s' :

$$k_{\mathcal{T}}(s, s') = \begin{cases} \frac{1}{N_L} & \text{if } s \text{ and } s' \text{ reach the same leaf } L \text{ containing } N_L \text{ subwindows,} \\ 0 & \text{otherwise} \end{cases}$$

Two subwindows are **very similar** *if they fall in a same leaf that has a* **very small** *subset of training subwindows*

Similarity between two subwindows (ensemble of T trees)

The similarity induced by an *ensemble* of T trees is defined by:

$$k_{ens}(s, s') = \frac{1}{T} \sum_{t=1}^{T} k_{\mathcal{T}_t}(s, s') \tag{1}$$

Two subwindows are similar if they are considered similar by a large proportion of the trees

Similarity between two images

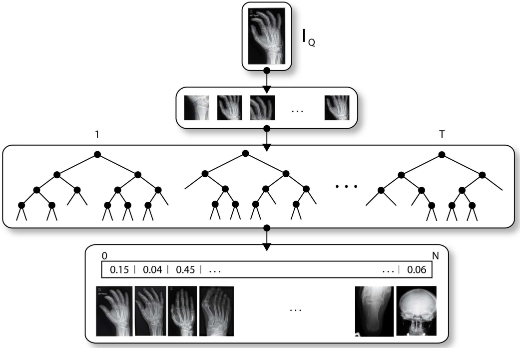
We derive a similarity between two images I and I' by:

$$k(I, I') = \frac{1}{|S(I)||S(I')|} \sum_{s \in S(I), s' \in S(I')} k_{ens}(s, s') \qquad (2)$$

The similarity between two images is thus the average similarity between all pairs of their subwindows

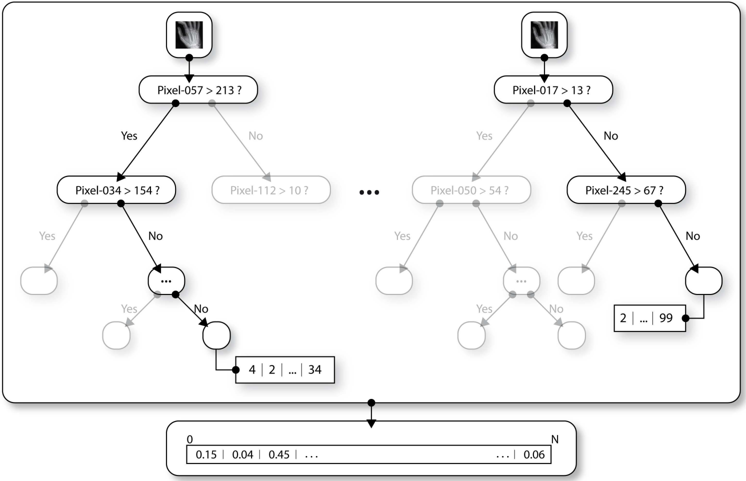
(2) is estimated by extracting at random from each image an a priori fixed number of subwindows

Similarities between I_Q and all reference images...



... are obtained by propagating subwindows from I_Q, and by incrementing, for each subwindow s of I_Q, each tree $\mathcal{T}$, and each reference image (I_R), the similarity $k(I_Q, I_R)$ by the proportion of subwindows of I_R in the leaf reached by s in the tree $\mathcal{T}$, and by normalizing the resulting score.

Propagation of one subwindow into trees



Extensions

- *Model recycling*: Given a large set of unlabeled images we can build an ensemble of trees on these images, and then use this model to compare new images from another set.

- *Incremental mode*: It is possible to incorporate the subwindows of a new image into an existing indexing structure by propagating and recording their leaf counts. If a leaf happens to contain more than $n_{\min}$ subwindows, split the node.

ZuBuD (1/3): images of 201 buildings

ZuBuD (2/3): results

- Protocol
  - 1005 *unlabeled* reference images (640 × 480)
  - 115 *labeled* test images (320 × 240)
  - Recognition rate of the first ranked image
- Results

Dataset	ls/ts	us	OM05	OM02
ZuBuD	1005/115	96.52%	93% to 98.2%	100%

(with 10 trees, 1000 subwindows per image, *nmin* = 2 ie. fully developed trees)

Université de Liège

ZuBuD (3/3): query ⟶ top 10 retrieved images

Université de Liège

IRMA (1/3): X-Ray images (from http://irma-project.org/)

Université de Liège

IRMA (2/3): Results

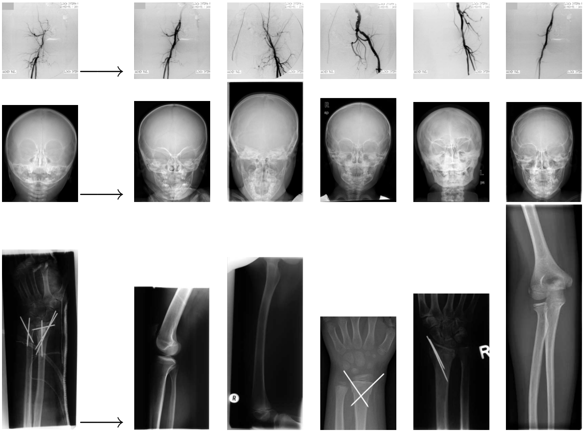
- Protocol
  - 9000 *unlabeled* reference images (approx. 512×512)
  - 1000 *labeled* test images (57 classes)
  - Recognition rate of the first ranked image
- Results

Dataset	ls/ts	us	naïve	NN	KDGN07
IRMA	9000/1000	85.4%	29.7%	63.2%	87.4%

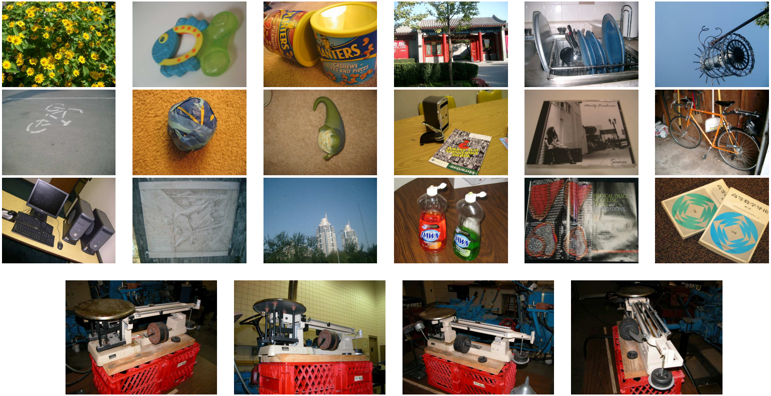
(with 10 trees, 1000 subwindows per image, $nmin = 2$ ie. fully developed trees)

Université de Liège

IRMA (3/3): query $\longrightarrow$ top 5 retrieved images



UkBench (1/2): images of 2550 “objects”



UkBench (2/2): results

- Protocol
  - 10200 *unlabeled* reference images (640 × 480)
  - Same images for test (*labeled*)
  - Recognition rate of the top-4 ranked images

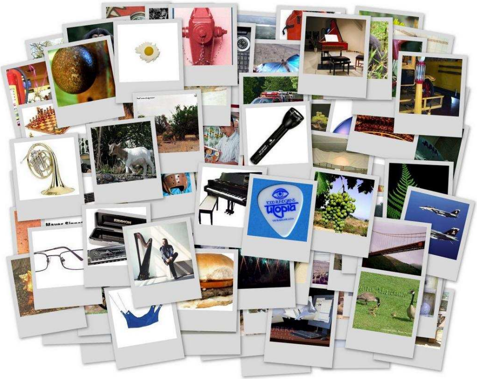
    (Number of correct images in first 4 retrieved images /40800) $*$ 100%

- Results

Dataset	ls=ts	us	NS06	PCISZ07
UkBench	10200	75.25%	76.75% to 82.35%	86.25%

(with 10 trees, 1000 subwindows per image, $nmin = 4$)

META (1/2): images from various sources



Sources: LabelMe Set1-16, Caltech-256, Aardvark to Zorro, CEA CLIC, Pascal Visual Object Challenge 2007, Natural Scenes A. Oliva, Flowers, WANG, Xerox6, Butterflies, Birds.

META (2/2): results

- Protocol
  - 205763 *unlabeled* reference images
  - 10200 UkBench *labeled* test images
  - Recognition rate of the top-4 ranked images

    (Number of correct images in first 4 retrieved images /40800) $*$ 100%

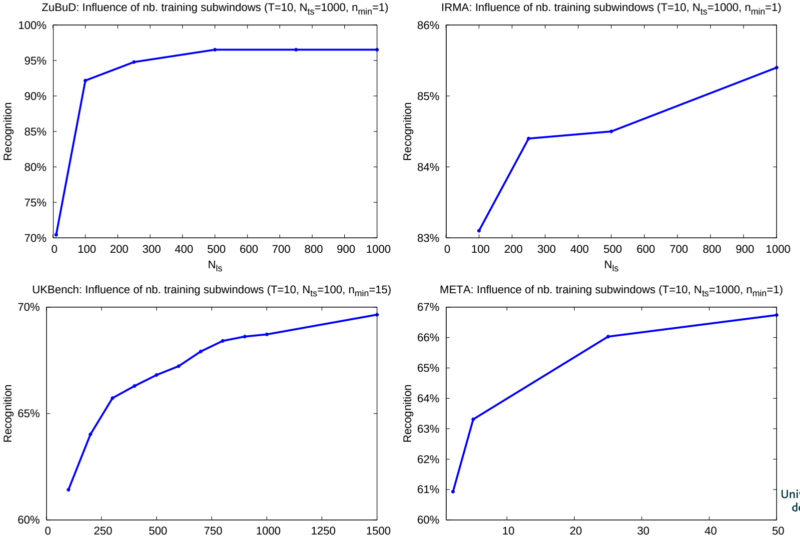
- Results

Dataset	ls/ts	us	NS06
META/UkBench	205763/10200	66.74 %	54% to 79 %

(with 10 trees, 50 subwindows per META image, 1000 subwindows per UkBench image, $nmin = 2$ ie. fully developed trees)

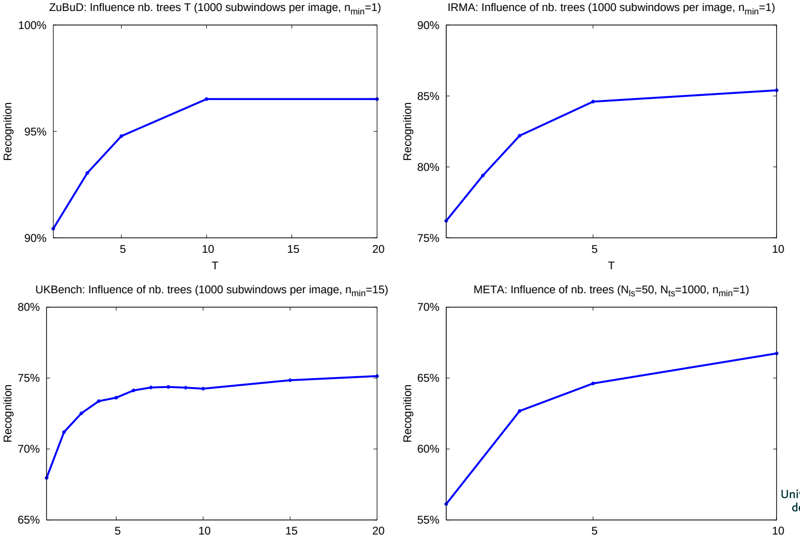
Université de Liège

Number of subwindows per training image: more is better

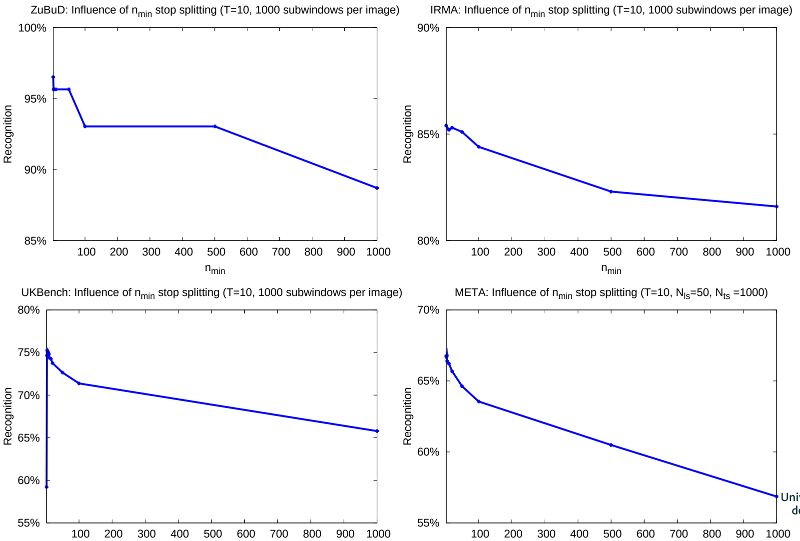


Université de Liège

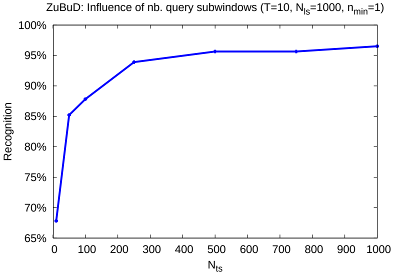
Number of trees T: more is better

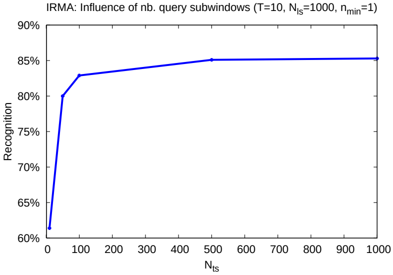


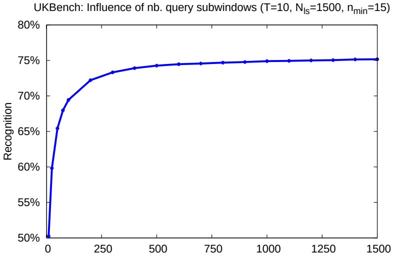
Tree depth (minimum node size n_{min}): deeper is better

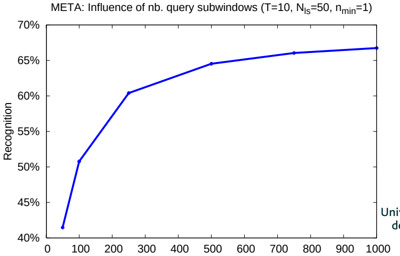


Number of subwindows per query image: more is better









Summary

- A simple method that yields quite good results on various tasks...
    - *Unlabeled* reference images
    - Extraction of *random subwindows*
    - Description by *raw pixel values*
    - Indexing with *totally randomized trees*
    - Image similarity derived from trees
- ... and has some nice practical properties
    - Only a few parameters
    - Fast indexing, fast prediction (parallelization also possible)
    - Model recycling, incremental mode
    - (Implementation in Java, check `http://www.montefiore.ulg.ac.be/~maree/`)

Prospects

- Applications
  - Tackle even more challenging visual tasks
  - Deal with bigger databases (Flickr hits two billion images)
  - Image near-duplicate detection
  - Indexing of other types of data (e.g. audio)
- Method
  - Combination with features/descriptors
  - Mechanisms like relevance feedback, sub-image retrieval, . . .

Acknowledgments

- Vincent Botta for figures
- Walloon Region RÉGION WALLONNE
- European Regional Development Fund
- National Fund for Scientific Research FNRS
- IRMA database courtesy of TM Lehmann and T. Deselaers (RWTH Aachen, Germany)

Université de Liège