Random Subwindows for Robust Image Classification

Raphaël Marée   Pierre Geurts   Justus Piater   Louis Wehenkel
Department of Electrical Engineering and Computer Science
Institut Montefiore, University of Liège
Sart Tilman, B4000, Liège, Belgium
Raphael.Maree@ulg.ac.be

Abstract

We present a novel, generic image classification method based on a recent machine learning algorithm (ensembles of extremely randomized decision trees). Images are classified using randomly extracted subwindows that are suitably normalized to yield robustness to certain image transformations. Our method is evaluated on four very different, publicly available datasets (COIL-100, ZuBuD, ETH-80, WANG). Our results show that our automatic approach is generic and robust to illumination, scale, and viewpoint changes. An extension of the method is proposed to improve its robustness with respect to rotation changes.

1. Introduction

We consider the general problem of image classification as it occurs in a variety of domains such as medecine, biology, geology, astronomy, quality control, office automation, arts, etc. Image classification methods seek to automatically classify previously unseen images of various kinds of "objects" using databases of labeled images provided by human experts. To be useful in practice, image classification methods must be automatic, generic and robust.

In this paper, we propose a novel method that combines a recent machine learning algorithm with a novel technique of extracting subwindows (square patches) from images. The main steps of our approach are described in Section 3. Empirical results are reported in Section 4 on four datasets presented in the literature. In Section 5, we explain the good behavior of our method and a variant is proposed to increase robustness to rotation changes. Some indications about its memory and computing time requirements are also given.

2. Related Work

Many recent image classification methods function according to the following scheme [9]. First, interest points or image regions are detected whose neighbourhood has high informational content and which are thought to be robustly detectable in images under varying conditions [11].

Then, the appearance of the interest points or regions is encoded by a feature vector of numerical values computed in their neighbourhood [10]. Such descriptors are often designed to be discriminative, concise and insensitive to various transformations such as illumination, orientation, scale, and viewpoint changes. They are sometimes compressed by dimensionality reduction (such as PCA or DCT) because local regions contain *too much* data for the traditional learning methods that are not able to deal with very high numbers of variables. These local feature vectors are then stored in a database for use during the recognition step.

To predict the class of a new image, every feature vector computed from a test image is classified using a nearest-neighbor algorithm against the feature vectors in the database. The majority class among the classes assigned to local feature vectors is then assigned to the test image.

3. Method

Our approach largely follows the aforementioned scheme. In earlier work [7], we proposed a method which applies a machine learning technique on fixed-size, square subwindows randomly extracted from images. In the present paper, we greatly improve the autonomy and the robustness of this method to certain image transformations. This improvement is obtained through a novel technique of extracting subwindows that takes into account these transformations in a parameter-free way. During the training phase, subwindows are randomly extracted from training images, and a model is constructed by machine learning based on transformed versions of these (Figure 1). Classification of a new test image similarly entails extraction and description of subwindows, and the application of the learned model to these subwindows. Aggregation of subwindow predictions is then performed to classify the test image, as illustrated in Figure 2.

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 IEEE

3.1. Subwindows

Our method extracts a large number of possibly overlapping, square subwindows of random sizes and at random positions from training images. Each subwindow size is randomly chosen between 1×1 pixels and the minimum horizontal or vertical size of the current training image. The position is then randomly chosen so that each subwindow is fully contained in the image. By randomly selecting a large number (N_w) of subwindows, we are able to cover large parts of images very rapidly. This random process is generic and can be applied to any kind of images. The same random process is applied to test images.

As shown in Section 4, about 100 random subwindows are often sufficient to correctly predict the class of a given test image. Training on about 100000 subwindows yields good overall recognition performance. This number is still substantially smaller than exhaustive sampling of all subwindow sizes and locations.

Subwindows are resized to a fixed scale (16×16 pixels). Experiments have shown that this normalization results in robustness to scale changes (see Section 5.4). It allows us to use generic machine learning methods (see Section 3.2) that work with fixed-size feature vectors. The resized subwindows are transformed to a HSV color space, as this, in comparison to RGB, results in superior robustness to illumination changes. Each subwindow is thus described by a feature vector of 768 ($= 16 \times 16 \times 3$) numerical values. The same descriptors are used for subwindows obtained from training and test images.

3.2. Learning

At the learning phase, a model is automatically built using subwindows extracted from training images. First, each subwindow is labelled with the class of its parent image. Then, any supervised machine learning algorithm can be applied to build a subwindow classification model: neural network, decision tree, SVMs or ensemble of decision trees. Here, the input of a machine learning algorithm is thus a training sample of N_w subwindows, each of which is described by 768 real-value input variables and a discrete output class (Figure 1). The learning algorithm should consequently be able to deal efficiently with a large amount of data, first in terms of the number of subwindows and classes of images in the training set, but more importantly in terms of the number of values describing these subwindows.

In this context, we propose to use a particular ensemble method of decision trees. Ensemble methods improve an existing learning algorithm by combining the predictions of several models. They are very effective when used with decision trees that otherwise are often not competitive in terms of accuracy with other learning algorithms. The method we will use consists in building many extremely randomized trees (extra-trees) [4, 5]. The main difference with respect to other ensemble methods is that the split thresholds at internal nodes of the decision trees are selected fully at random, i.e. not on the basis of any score measure. Each decision tree of the ensemble is then grown until it perfectly classifies the training sample. Because of the extreme randomization, this method is usually much faster than other ensemble methods. It was also shown to perform remarkably well on a variety of tasks in terms of accuracy [5]. This high accuracy can be explained in terms of a bias/variance analysis.

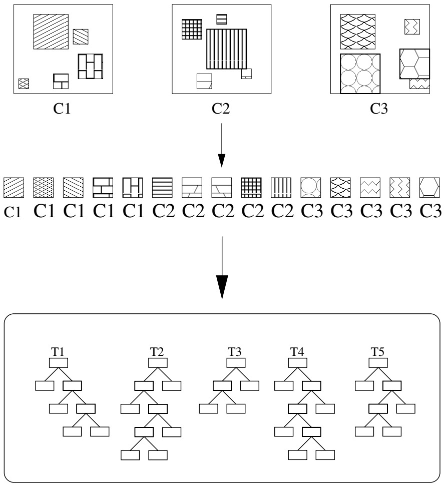

Figure 1. Learning: Our method first randomly extracts multi-scale subwindows in training set images, then resizes them and builds an ensemble of extra-trees.

3.3. Recognition

In our approach, the database of subwindows extracted from the training images are no longer used after training, and can be discarded. We only use the learned model to classify subwindows of a test image. To make a prediction for a test image with an ensemble of extra-trees grown from subwindows, we simply propagate each test subwindow into each extra-tree of the ensemble. Each extra-tree assigns one class to each subwindow. Each subwindow thus receives T votes where T denotes the number of trees in the ensemble. We then aggregate all the predictions by simply adding the votes, as illustrated by Figure 2, and we assign to the image the majority class among the classes assigned to its subwindows.

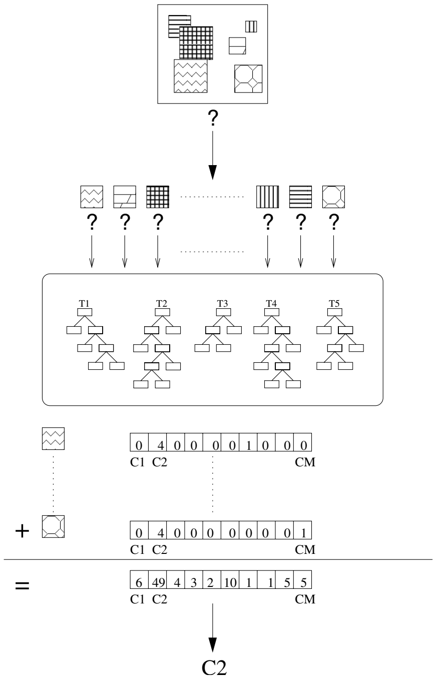

Figure 2. Recognition: Randomly extracted subwindows are propagated through the trees (here $T = 5$). Votes are aggregated and the majority class is assigned to the image.

4. Experiments

Our experiments aim at demonstrating the generality and robustness of our approach based on learning of random subwindows. To this end, we tested it on four well-known and publicly available datasets corresponding to various classification problems: household objects in a controlled environment (COIL-100), buildings in urban scenes (ZuBuD), object categories in a controlled environment (ETH-80), and landscape themes (WANG). The first dataset exhibits important viewpoint changes. ZuBuD and WANG contain images with illumination, viewpoint, scale and orientation changes as well as partial occlusions and cluttered backgrounds. WANG and ETH-80 span large intra-class variabilities. Our results are directly comparable to the state of the art, as we stricly follow published protocols. For each problem and protocol, the parameters of our method are fixed to $N_w = 120000$, $T = 10$, and 100 subwindows are randomly extracted from each test image. These numbers were sufficient to produce good results on every dataset.

4.1. COIL-100

COIL-100[1] [12] is a dataset of 128×128 color images of 100 different 3D objects with 72 images per object at pose intervals of 5 degrees. The goal is to classify images into the correct class among the 100 classes. The best available results are reported by Matas and Obdržálek [9] who evaluated the generalisation ability by considering various training sample sizes. On this dataset, reducing the number of training views increases perspective distortions between learned views and images presented during testing.

In this paper, we evalute the robustness to such viewpoint changes using two experimental protocols. First, we selected for the learning sample 18 views of each of the 100 objects, starting with the pose at 0° and continuing at intervals of 20°. Using this protocol, methods in the literature report error rates from 12.5% to 0.1% [9]. In a second experiment, we selected only one view (the pose at 0°) in the training sample while the remaining 71 views are used for testing. Using this protocol, methods in the literature yield error rates from 50.1% to 24% [9]. As expected, error rates increase when the size of the training set decreases.

Using the first protocol (1800 training images and 5400 test images), we obtain an error rate of 0.5% with our method. Using the second protocol (100 learning images, 7100 test images), we obtain a remarkably low 13.58% error rate which is the best result known so far. Figure 3 illustrates the behavior of the method in presence of viewpoint changes introduced by the second protocol.

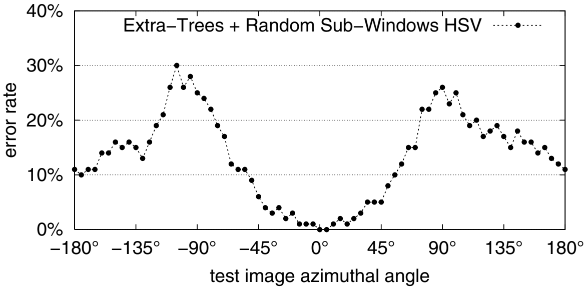

Figure 3. COIL-100: error rates depending on azimuthal test angle, learning from only one view (0°).

4.2. ZuBuD

The ZuBuD dataset[2] created in April 2003 [13] is a database of color images of 201 buildings in Zürich. Each building in the training set is represented by five images acquired at five random arbitrary viewpoints. The training set thus includes 1005 images, while the test set contains 115 images of a subset of the 201 training buildings. Images were taken by two different cameras in different seasons and under different weather conditions, and thus contain a substantial variety of illumination conditions. Partial occlusions and cluttered backgrounds are naturally present (trees,

[1] http://www.cs.columbia.edu/CAVE/

[2] http://www.vision.ee.ethz.ch/showroom/zubud/index.en.html

skies, cars, trams, people, ...) as well as scale and orientation changes due to the position of the photographer. Moreover, training images were captured at 640×480 while testing images are at 320×240 pixels.

About five papers have so far reported results on this dataset that vary from a 59% error rate to 0% [9]. With our method, we obtain a 4.35% error rate (only 5 images misclassified among 115). Figure 4 shows the 5 misclassified images. The first three images correspond to the same building but with substantial differences in illumination, orientation, and viewpoint between training and testing views. The fourth misclassified test image also represent a strong orientation and viewpoint change as well as some occlusion. The last misclassified test image is difficult to distinguish even for a human. For this image, the correct class is ranked second by our model, and obtained only one fewer vote (of one thousand) than the class ranked first.

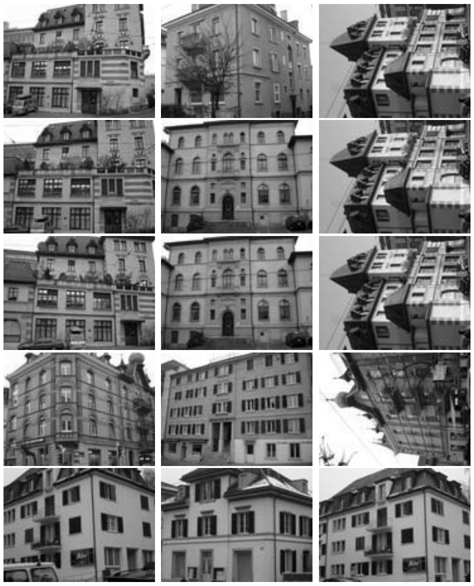

Figure 4. ZuBuD: misclassified test images (left), training images of predicted class buildings (middle), training images of correct buildings (right).

4.3. ETH-80

The Cogvis ETH-80 dataset[3] contains 3280 color images (128×128 pixels) of 8 distinct object categories (apples, pears, tomatoes, cows, dogs, horses, cups, cars). For each category, 10 different objects are provided. Each object is represented by 41 images from viewpoints spaced equally over the upper viewing hemisphere. The protocol used by Leibe and Schiele [6] is leave-one-object-out cross-validation. This means that training is done on all views of 79 objects (a training set of 3239 images), and the test entails the classification of all 41 views of the remaining object. Recognition of one test image is considered successful if the correct category label is assigned. The results are averaged over all 80 possible test objects. Averaged error rates vary from 35.15% to 13.60% [6]. Our method achieves 25.49% error.

4.4. WANG

The WANG dataset[4] is used in the literature [1, 2] to evaluate image retrieval and image categorization methods. It consists of 1000 images subdivided into 10 categories, each represented by 100 images, illustrating the following themes: African people and villages, beach, buildings, buses, dinosaurs, elephants, flowers, horses, mountain and glaciers, and food. Such common categories exhibits high intra-class variability. The images are of size 384×256 (or 256×384). The protocol used to evaluate our method is a leave-one-out cross-validation. That is, for every image, a model is built by using the remaining 999 images for the training, ie. we randomly extract N_w subwindows from the training set of 999 images and build one ensemble of T trees. The test image is classified by that model. The error rate is then averaged over the entire database. Error rates in the literature vary from 62.5% to 15.9% [2]. With our method, we obtain a result equivalent to the best available (15.9% error rate).

5. Discussion

In this section, we explain the very good performance of our method on different tasks by the combination of simple but well-motivated techniques: random subwindow extraction (5.1), HSV pixel representation (5.2), and the recent extra-trees machine learning method (5.3). We propose a variant of this method for better robustness to orientation changes (5.5). We also give some information about memory requirements and computing times (5.6).

5.1 Subwindows

In the object recognition literature, local approaches generally perform better than global approaches. They are more robust to varying conditions because these variations can locally be modelled by simple transformations [9]. These methods are also more robust to partial occlusions and cluttered background. Our approach has the same benefits. Indeed, the correct classification of all subwindows is not required to correctly classify one image. For example, for the ZuBud problem, Figure 5 exhibits one image correctly

[3] http://www.vision.ethz.ch/projects/categorization/eth80-db.html

[4] http://wang.ist.psu.edu/docs/related

classified by our model despite its occluded parts. The figure also shows individual subwindows that were correctly classified (first row) and misclassified (second row).

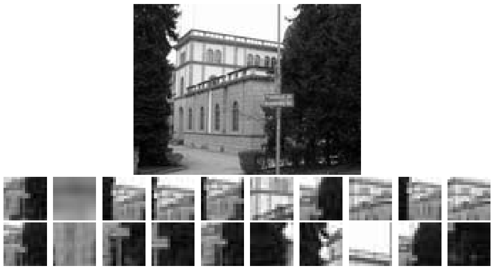

Figure 5. Some subwindows extracted from one test image. Subwindows on the first row are correctly classified while those on the second row are misclassified individually.

Secondly, as an image contains substantial redundancy, spatial image transformations do not alter all elements of the feature vectors to the same extent. Many feature vectors will be left more or less intact by a given image transformation, resulting in remarkably robust performance to viewpoint and orientation changes. Furthermore, robustness to scale changes is improved by normalizing the subwindow to a common size of 16×16 pixels (see Section 5.4).

The labelling and random extraction of a large number of subwindows in the training images provides a large amount of information corresponding to various image regions. Both global and local regions are used. This process also contrasts with some methods in the literature which are not able to extract enough regions or interest points when images are too small or objects are too "simple" (like in the COIL-100 database). In fact, Mikolajczyk et al. [11] recently concluded that current detectors are complementary (some being more adapted to structured scenes while others to textures) and that all of them should ideally be used in parallel. We think that in terms of image coverage, the resulting combination of all these detectors would lead to a representation of overlapping regions quite similar to our randomly extracted subwindows.

Finally, with our method, we have observed that accuracy is a monotonically increasing function of the number N_w of subwindows. A larger value of N_w could further improve the accuracy.

5.2 HSV pixel representation

Since our subwindows are represented in terms of their raw pixel values (in the form of 768-dimensional feature vectors), we do not explicitly discard informational content. To a certain extent, determining which image pixels discriminate over the entire set of subwindows is implicitly done by the machine learning algorithm. High-dimensional feature vectors also allow the algorithm to build a classifier able to distinguish between a large number of classes. Pixel representations are not robust per se, but have to rely on the use of a machine learning algorithm working with a large number of random subwindows extracted from training images that exhibit some varying conditions.

The HSV representation is more robust to illumination changes than the RGB color space because it tends to largely limit the effects of the most important, practically-occurring illumination changes to just one of the three bands. We observed a measurable effect of this phenomenon on classification accuracy. For example, with the second protocol of COIL-100 experiments, we obtain 20.63% error rate using RGB compared to 13.58% using HSV. Similar results were observed on other datasets.[5]

5.3 Extra-trees

Image classification is particularly difficult for traditional machine learning algorithms (e.g., decision tree induction and nearest-neighbor classifiers) mainly because of the high number of input variables that describe images (i.e., pixels). Indeed, with a high number of variables, these methods tend to produce very unstable models with low generalization performance. Moreover, a distance metric in nearest-neighbor classification can be perturbed by irrelevant variables and small pixel-value changes due to misalignment (interest point localization error) or due to other small transformations (translation, rotation, ...).

The success of machine learning techniques in our case is the combination of two factors. First, recent advances in machine learning have produced new methods that are able to handle problems of high dimensionality. Decision tree ensemble methods, including extra-trees, are among these new methods. For comparison, using a single, conventional decision tree instead of an ensemble of trees, we obtain a 19.08% error rate with the second COIL-100 protocol (as opposed to 13.58% with extra-trees), and 13.91% error rate (16 test images misclassified) on ZuBuD dataset (as opposed to 4.35%). Another key to the success of machine learning techniques is the classification of random subwindows instead of full images that at the same time increases the training sample size and decreases the dimensionality. In comparison, the direct application of extra-trees for the classification of full images gives an error rate of 32.61% with the second COIL-100 protocol.

Extra-trees have a high precision (due to their low variance [5]) and an attractive computational efficiency (see Section 5.6). However, any other supervised learning al-

[5] Some experiments were also performed using grey values instead of color information on COIL-100 with different protocols. Most objects are still recognized but the results are not as good because some COIL-100 objects are distinguished only by their color.

gorithm is directly applicable within our random subwindow framework. For example, using Tree Boosting [3], we get even better results with protocol 2 on COIL-100 (11.38% error rate as opposed to 13.58% using extra-trees), and equivalent results (4.35% error rate) on ZuBuD. Note however that Tree Boosting is much slower than extra-trees for learning.

5.4 Robustness to scale changes

Images from COIL-100 and ETH-80 databases occur at a fixed scale, but the ZuBuD and WANG datasets contain images at different sizes or different zoom levels. Further experiments [8] have shown that the size normalization of the subwindows improve robustness to such scale changes. We performed a series of experiments following the COIL-100 protocol 1 where we built a model from 32×32 training images and tested it on scaled versions of the test images ranging from 16×16 to 128×128 pixels. As expected, error rates are always very similar (close to 0.5%) whatever the size of the test image. With our earlier method [7], the error rate with 48×48 test images was close to 7% (in RGB).

5.5 Robustness to orientation changes

Our method yields good results on datasets frequently used in the literature. In the ZuBuD dataset, some natural orientation changes are introduced by the position of the photographer and because pictures were taken at different orientations (landscape or portrait). The same kind of differences exist in the WANG dataset. However, these datasets do not include images with substantial rotational changes between testing and training images so that robustness to rotation could not be evaluated systematically. Some experiments showed that our method is only robust to small rotation changes [8]. We introduced a variant of our method where subwindows (from training and testing images) are randomly rotated before rescaling them to 16×16 pixels, as illustrated in Figure 6. With this variant, we obtain 0.87% errors on the original COIL-100 protocol 1 and 15.58% on the COIL-100 protocol 2 which is only slightly inferior to the variant without rotated subwindows. To evaluate the robustness of this variant, we used COIL-100 protocol 1 where we applied image-plane rotation to test images. The model built from original images was tested on rotated versions of the test images. Results for rotations between 0° and 135° are reported in Table 1. This variant is thus quite robust to rotational changes. With our earlier method [7], the error rate for $ra = 45°$ was more than 30% (in RGB).

Rotation 2D	*subwindows with random rotation*
$ra = 0°$	0.87%
$ra = 45°$	1.56%
$ra = 90°$	1.24%
$ra = 135°$	2.56%

Table 1. Error rates with 2D rotated test images for COIL-100 protocol 1: test images are rotated by ra degrees.

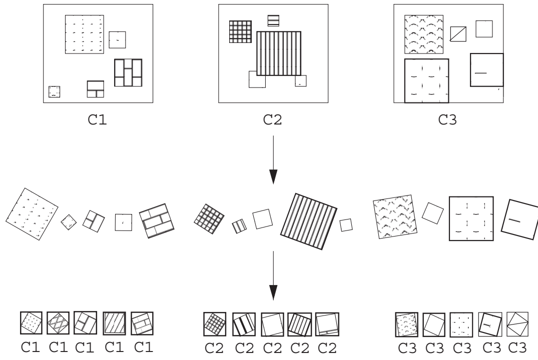

Figure 6. An adaptation of our method for a better robustness to orientation changes. subwindows of random sizes are extracted as stated before, then randomly rotated, then resized in 16×16 pixels.

5.6 Some notes on implementation

Our current implementation cannot be considered optimal but some indications can be given about memory and running-time requirements. With our method, original training images and their subwindows are not necessary to classify new images after the construction of the model. Only ensembles of trees are used for recognition. If the subwindows are in main memory, training $T = 10$ extra-trees takes about 6m30s on a Pentium IV 2.4Ghz processor on the ZuBuD problem. In comparison, building $T = 10$ boosted trees takes about 25h. For this problem, storing the resulting ensemble of $T = 10$ trees on disk require about 12Mb which should be compared to 488Mb for storing the 1005 PNG training images. On average, one extra-tree has about 148025 nodes for this problem. The prediction of one test subwindow with one extra-tree requires on average less than 20 tests (each of which involves comparing the value of a pixel to a threshold) [6].

To classify one unseen image, the number of operations is thus multiplied by T and by the number of subwindows extracted (100 in our experiments). The addition of all votes and maximum search is negligible. Furthermore, extraction of one subwindow is very fast because of its random nature. The variant with rotated subwindows involves more operations to apply random transformations, but the number of

[6] The average is 18.26. It was calculated over the 115000 propagations (100 subwindows for each of the 115 test images, each subwindow propagated to $T = 10$ trees). Depending of the subwindow, the minimum depth was 9, the maximum was 32.

subwindows extracted in a test image is quite low. Note that the method is flexible as one can extract fewer subwindows and use fewer extra-trees if a specific tradeoff between accuracy and computing times is desired.

In terms of scalability, larger data sets would lead to the extraction of more subwindows (larger N_w). The extraction of these subwindows is very fast and extra-trees scale very well with N_w. The complexity of the tree induction algorithm is of order $O(N_w log N_w)$. The size of the extra-trees may grow substantially with larger numbers of images and classes, but the test of one random subwindow remains $O(logNw)$.

6. Summary and conclusions

In this paper, we proposed a novel image classification method. Its main steps are the random extraction of subwindows, their transformation to normalize their representation, and the supervised automatic learning of a classifier based on ensembles of decision trees operating directly on the pixel values. The method has been evaluated on 4 publicly-available datasets corresponding to various image classification tasks. These datasets contain images representing widely varying conditions: occlusions, cluttered background, illumination, viewpoint, orientation and scale changes. These databases contain up to 205 different classes. The last two datasets correspond to image categorization problems with high intra-class variability. Our method yields good results and it is particularly attractive in terms of computational efficiency. Furthermore, it is generic and fully automatic: The same framework (extraction, representation, learning and recognition steps) could be directly applied to any image classification problem without any parameter adaptation.

For future work, it would now be interesting to perform a comparative study with other machine-learning methods (e.g. Tree Boosting or SVMs) and other techniques for extracting image regions. We think that such a comparison is the next important step for practical image classification, following analyses such as those by Mikolajczyk et al. [10, 11] that compare region detectors and local descriptors. In terms of applications, our method will be evaluated on bigger databases in terms of the number of images and/or classes and with images which exhibits higher intra-class variability and heavily cluttered backgrounds (such as Caltech-101[7] or Butterflies[8] datasets).

7. Acknowledgment

Pierre Geurts is a Postdoctoral Researcher at the National Fund for Scientific Research (FNRS, Belgium).

[7] http://www.vision.caltech.edu/feifeili/Datasets.htm

[8] http://www-cvr.ai.uiuc.edu/ponce_grp/data/

References

[1] Y. Chen and J. Z. Wang. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5:913–939, August 2004.

[2] T. Deselaers, D. Keysers, and H. Ney. Features for image retrieval: A quantitative comparison. In *Proc. 26th DAGM Symposium on Pattern Recognition (DAGM 2004)*, volume LNCS 3175, pages 228–236, August/September 2004.

[3] Y. Freund and E. Robert Schapire. Experiments with a new boosting algorithm. In *Proc. Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.

[4] P. Geurts. *Contributions to decision tree induction: bias/variance tradeoff and time series classification*. PhD thesis, Department of Electrical Engineering and Computer Science, University of Liège, May 2002.

[5] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Submitted*, 2004.

[6] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*, Madison, WI, June 2003.

[7] R. Marée, P. Geurts, J. Piater, and L. Wehenkel. A generic approach for image classification based on decision tree ensembles and local sub-windows. In *Proc. 6th Asian Conference on Computer Vision*, January 2004.

[8] R. Marée. *Classification automatique d'images par arbres de décision*. PhD thesis, University of Liège - Electrical Engineering and Computer Science, February 2005.

[9] J. Matas and S. Obdržálek. Object recognition methods based on transformation covariant features. In *Proc. 12th European Signal Processing Conference (EUSIPCO 2004)*, Vienna, Austria, September 2004.

[10] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 257–263, June 2003.

[11] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Submitted to International Journal of Computer Vision*, 2004.

[12] H. Murase and S. K. Nayar. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.

[13] H. Shao, T. Svoboda, and L. Van Gool. Zubud - Zurich building database for image based recognition. Technical Report TR-260, Computer Vision Lab, Swiss Federal Institute of Technology, Switzerland, 2003.