Sampling strategies for bag-of-features image classification

Emmanuel Nowak, Frédéric Jurie, Bill Triggs

Sampling Strategies for Bag-of-Features Image Classification

Eric Nowak[1,2], Frédéric Jurie[1], and Bill Triggs[1]

[1] GRAVIR-CNRS-INRIA,
655 avenue de l'Europe,
Montbonnot 38330, France
{Eric.Nowak, Bill.Triggs, Frederic.Jurie}@inrialpes.fr
http://lear.inrialpes.fr
[2] Bertin Technologie, Aix en Provence, France

Abstract. Bag-of-features representations have recently become popular for content based image classification owing to their simplicity and good performance. They evolved from texton methods in texture analysis. The basic idea is to treat images as loose collections of independent patches, sampling a representative set of patches from the image, evaluating a visual descriptor vector for each patch independently, and using the resulting distribution of samples in descriptor space as a characterization of the image. The four main implementation choices are thus how to sample patches, how to describe them, how to characterize the resulting distributions and how to classify images based on the result. We concentrate on the first issue, showing experimentally that for a representative selection of commonly used test databases and for moderate to large numbers of samples, random sampling gives equal or better classifiers than the sophisticated multiscale interest operators that are in common use. Although interest operators work well for small numbers of samples, the single most important factor governing performance is the number of patches sampled from the test image and ultimately interest operators can not provide enough patches to compete. We also study the influence of other factors including codebook size and creation method, histogram normalization method and minimum scale for feature extraction.

1 Introduction

This paper studies the problem of effective representations for automatic image categorization – classifying unlabeled images based on the presence or absence of instances of particular visual classes such as cars, people, bicycles, etc. The problem is challenging because the appearance of object instances varies substantially owing to changes in pose, imaging and lighting conditions, occlusions and within-class shape variations (see fig. 2). Ideally, the representation should be flexible enough to cover a wide range of visually different classes, each with large within-category variations, while still retaining good discriminative power between the classes. Large shape variations and occlusions are problematic for

A. Leonardis, H. Bischof, and A. Prinz (Eds.): ECCV 2006, Part IV, LNCS 3954, pp. 490–503, 2006.

Fig. 1. Examples of multi-scale sampling methods. (1) Harris-Laplace (HL) with a large detection threshold. (2) HL with threshold zero – note that the sampling is still quite sparse. (3) Laplacian-of-Gaussian. (4) Random sampling.

rigid template based representations and their variants such as monolithic SVM detectors, but more local 'texton' or 'bag-of-features' representations based on coding local image patches independently using statistical appearance models have good resistance to occlusions and within-class shape variations. Despite their simplicity and lack of global geometry, they also turn out to be surprisingly discriminant, so they have proven to be effective tools for classifying many visual classes (e.g. [1, 2, 3], among others).

Our work is based on the bag-of-features approach. The basic idea of this is that a set of local image patches is sampled using some method (e.g. densely, randomly, using a keypoint detector) and a vector of visual descriptors is evaluated on each patch independently (e.g. SIFT descriptor, normalized pixel values). The resulting distribution of descriptors in descriptor space is then quantified in some way (e.g. by using vector quantization against a pre-specified codebook to convert it to a histogram of votes for (i.e. patches assigned to) codebook centres) and the resulting global descriptor vector is used as a characterization of the image (e.g. as feature vector on which to learn an image classification rule based on an SVM classifier). The four main implementation choices are thus how to sample patches, what visual patch descriptor to use, how to quantify the resulting descriptor space distribution, and how to classify images based on the resulting global image descriptor.

One of the main goals of this paper is to study the effects of different patch sampling strategies on image classification performance. The sampler is a critical component of any bag-of-features method. Ideally, it should focus attention on the image regions that are the most informative for classification. Recently, many authors have begun to use multiscale keypoint detectors (Laplacian of Gaussian, Förstner, Harris-affine, etc.) as samplers [4, 1, 2, 5, 6, 7, 8, 9, 10, 11], but although such detectors have proven their value in matching applications, they were not designed to find the most informative patches for image classification and there is some evidence that they do not do so [12, 13]. Perhaps surprisingly, we find that randomly sampled patches are often more discriminant than keypoint based ones, especially when many patches are sampled to get accurate classification results (see figure 1). We also analyze the effects of several other factors including codebook size and the clusterer used to build the codebook. The experiments are performed on a cross-section of commonly-used evaluation datasets to allow us to identify the most important factors for local appearance based statistical image categorization.

2 Related Work

Image classification and object recognition are well studied areas with approaches ranging from simple patch based voting to the alignment of detailed geometric models. Here, in keeping with our approach to recognition, we provide only a representative random sample of recent work on local feature based methods. We classify these into two groups, depending on whether or not they use geometric object models.

The geometric approaches represent objects as sets of parts whose positions are constrained by the model. Inter-part relationships can be modelled pairwise [4], in terms of flexible constellations or hierarchies [2, 14], by co-occurrence [15] or as rigid geometric models [8, 7]. Such global models are potentially very powerful but they tend to be computationally complex and sensitive to missed part detections. Recently, "geometry free" *bag-of-features* models based purely on characterizing the statistics of local patch appearances have received a lot of attention owing to their simplicity, robustness, and good practical performance. They evolved when texton based texture analysis models began to be applied to object recognition. The name is by analogy with the bag-of-words representations used in document analysis (e.g. [16]): image patches are the visual equivalents of individual "words" and the image is treated as an unstructured set ("bag") of these.

Leung *at al.* [3] sample the image densely, on each patch evaluating a bank of Gabor-like filters and coding the output using a vector quantization codebook. Local histograms of such 'texton' codes are used to recognize textures. Textons are also used in content based image retrieval, e.g. [17]. Lazebnik *et al.* [18] take a sparser bag-of-features approach, using SIFT descriptors over Harris-affine keypoints [9] and avoiding global quantization by comparing histograms using Earth Movers Distance [19]. Csurka *et al* [1] approach object classification using k-means-quantized SIFT descriptors over Harris-affine keypoints [9]. Winn *et al.* [13] optimize k-means codebooks by choosing bins that can be merged. Fergus *et al.* [5] show that geometry-free bag-of-features approaches still allow objects to be localized in images.

The above works use various patch selection, patch description, descriptor coding and recognition strategies. Patches are selected using keypoints [4, 1, 2, 5, 6, 7, 8, 9, 10, 11] or densely [3, 13, 15]. SIFT based [1, 6, 8, 10], filter based [3, 13] and raw patch based [4, 2, 5, 7, 11] representations are common. Both k-means [1, 3, 11, 13] and agglomerative [4, 7] clustering are used to produce codebooks, and many different histogram normalization techniques are in use. Our work aims to quantify the influence of some of these different choices on categorization performance.

3 Datasets

We have run experiments on six publicly available and commonly used datasets, three object categorization datasets and three texture datasets.

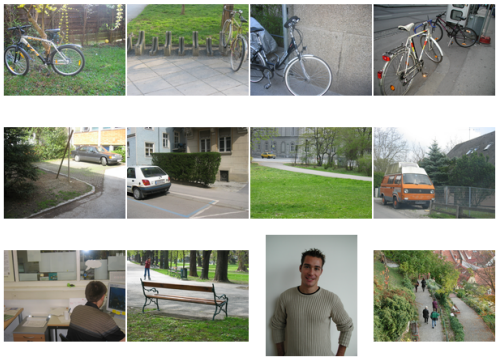

Fig. 2. Example of objects of Graz01 dataset: four images of the categories bike, car, person

Object datasets. *Graz01* contains 667, 640×480 pixel images containing three visual categories (bicycle, car, person) in approximately balanced proportions (see figure 2). *Xerox7*[1] contains 1776 images, each belonging to exactly one of the seven categories: bicycle, book, building, car, face, phone, tree. The set is unbalanced (from 125 to 792 images per class) and the images sizes vary (width from 51 to 2048 pixels). *Pascal-01*[2] includes four categories: cars, bicycles, motorbikes and people. A 684 image training set and a 689 image test set ('test set 1') are defined.

Texture datasets. *KTH-TIPS*[3] contains 810, 200×200 images, 81 from each of the following ten categories: aluminum foil, brown bread, corduroy, cotton, cracker, linen, orange peel, sandpaper, sponge and styrofoam. *UIUCTex*[4] contains 40 images per classes of 25 textures distorted by significant viewpoint changes and some non-rigid deformations. *Brodatz*[5] contains 112 texture images, one per class. There is no viewpoint change or distortion. The images were divided into thirds horizontally and vertically to give 9 images per class.

4 Experimental Settings

This section describes the default settings for our experimental studies. The multiscale Harris and LoG (Laplacian of Gaussian) interest points, and the randomly sampled patches are computed using our team's LAVA library[6]. The default parameter values are used for detection, except that detection threshold for interest points is set to 0 (to get as many points as possible) and – for comparability with other work – the minimum scale is set to 2 to suppress small regions (see §8).

[1] ftp://ftp.xrce.xerox.com/pub/ftp-ipc/
[2] http://www.pascal-network.org/challenges/VOC/
[3] http://www.nada.kth.se/cvap/databases/kth-tips/index.html
[4] http://www-cvr.ai.uiuc.edu/ponce_grp
[5] http://www.cipr.rpi.edu/resource/stills/brodatz.html
[6] http://lear.inrialpes.fr/software

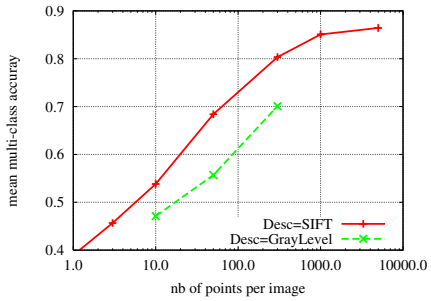

Fig. 3. Classifiers based on SIFT descriptors clearly out-perform ones based on normalized gray level pixel intensities, here for randomly sampled patches on the Graz dataset

We use SIFT [8] descriptors, again computed with the LAVA library with default parameters: 8 orientations and 4×4 blocks of cells (so the descriptor dimension is 128), with the cells being 3×3 pixels at the finest scale (scale 1). Euclidean distance is used to compare and cluster descriptors.

We also tested codings based on normalized raw pixel intensities, but as figure 3 shows, SIFT descriptor based codings clearly out-perform these. Possible reasons include the greater translation invariance of SIFT, and its robust 3-stage normalization process: it uses rectified (oriented) gradients, which are more local and hence more resistant to illumination gradients than complete patches, followed by blockwise normalization, followed by clipping and renormalization.

Codebooks are initialized at randomly chosen input samples and optimized by feeding randomly chosen images into online k-means (the memory required for true k-means would be prohibitive for codebooks and training sets of this size).

Descriptors are coded by hard assignment to the nearest codebook centre, yielding a histogram of codeword counts for each image. Three methods of converting histogram counts to classification features were tested: raw counts; simple binarization (the feature is 1 if the count is non-zero); and adaptive thresholding of the count with a threshold chosen to maximize the Mutual Information between the feature and the class label on the training set. MI based thresholding usually works best and is used as the default. Raw counts are not competitive so results for them are not presented below.

Soft *One-versus-one* SVM's are used for classification. In multi-class cases the class with the most votes wins. The SVM's are linear except in §9 where Gaussian kernels are used to make comparisons with previously published results based on nonlinear classifiers. The main performance metric is the unweighted mean over the classes of the recognition rate for each class. This is better adapted to unbalanced datasets than the classical "overall recognition rate", which is biased towards over-represented classes. By default we report average values over six complete runs, including the codebook creation and the category prediction. For most of the datasets the recognition rates are estimated using two-fold cross validation, but for Pascal-01 dataset we follow the PASCAL protocol and use the specified 'learning set'/'test set 1' split for evaluation.

5 Influence of the Sampling Method

The idea of representing images as collections of independent local patches has proved its worth for object recognition or image classification, but raises the question of which patches to choose. Objects may occur at any position and scale in the image so patches need to be extracted at all scales (e.g. [3, 13]). Dense sampling (processing every pixel at every scale, e.g. [12, 13]) captures the most information, but it is also memory and computation intensive, with much of the computation being spent on processing relatively featureless (and hence possibly uninformative) regions. Several authors argue that computation can be saved and classification performance can perhaps be improved by using some kind of salience metric to sample only the most informative regions. Example-based recognition proceeds essentially by matching new images to examples so it is natural to investigate the local feature methods developed for robust image

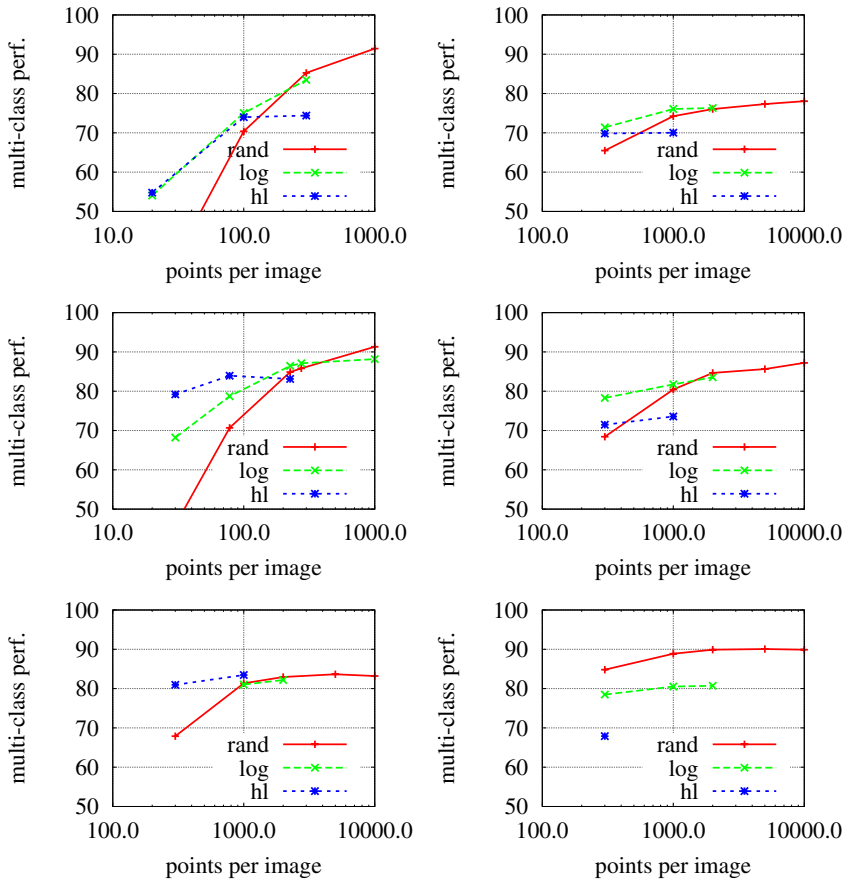

Fig. 4. Mean multi-class classification accuracy as a function of the number of sampled patches used for classification. Reading left to right and top to bottom, the datasets are: Brodatz, Graz01; KTH-TIPS, Pascal-01; UIUCTex and Xerox7.

matching in this context. In particular, many authors have studied recognition methods based on generic interest point detectors [4, 1, 2, 6, 7, 8, 9, 10, 11]. Such methods are attractive because they have good repeatability [8, 9] and translation, scale, 2D rotation and perhaps even affine transformation invariance [20]. However the available interest or salience metrics are based on generic low level image properties bearing little direct relationship to discriminative power for visual recognition, and none of the above authors verify that the patches that they select are significantly more discriminative than random ones. Also, it is clear that one of the main parameters governing classification accuracy is simply the number of patches used, and almost none of the existing studies normalize for this effect.

We investigate these issues by comparing three patch sampling strategies. *Laplacian of Gaussian (LoG):* a multi-scale keypoint detector proposed by [21] and popularized by [8]. *Harris-Laplace (Harris):* the (non-affine) multi-scale keypoint detector used in [18]. *Random (Rand):* patches are selected randomly from a pyramid with regular grids in position and densely sampled scales. All patches have equal probability, so samples at finer scales predominate. For all datasets we build 1000 element codebooks with online k-means and use MI-based histogram encoding (see §7) with a linear SVM classifier.

Figure 4 plots mean multi-class classification rates for the different detectors and datasets. (These represent means over six independent training runs – for typical standard deviations see table 1). Each plot shows the effect of varying the mean number of samples used per image. For the keypoint detectors this is done indirectly by varying their 'cornerness' thresholds, but in practice they usually only return a limited number of points even when their thresholds are set to zero. This is visible in the graphs. It is one of the main factors limiting the performance of the keypoint based methods: they simply can not sample densely enough to produce leading-edge classification results. Performance almost always increases with the number of patches sampled and random sampling ultimately dominates owing to its ability to produce an unlimited number of patches. For the keypoint based approaches it is clear that points with small cornerness are useful for classification (which again encourages us to use random patches), but there is evidence that saturation occurs earlier than for the random approach. For smaller numbers of samples the keypoint based approaches do predominate

Table 1. The influence of codebook optimization. The table gives the means and standard deviations over six runs of the mean classification rates of the different detectors on each dataset, for codebooks refined using online k-means (KM), and for randomly sampled codebooks (no KM).

Dataset	Rand KM	Rand no KM	LoG KM	LoG no KM	H-L KM	H-L no KM
Graz01	74.2 ± 0.9	71.3 ± 0.9	76.1 ± 0.5	72.8 ± 0.9	70.0 ± 1.4	68.8 ± 2.0
KTHTIPS	91.3 ± 1.1	92.1 ± 0.4	88.2 ± 1.0	85.0 ± 1.8	83.1 ± 2.1	81.3 ± 1.1
Pascal-01	80.4 ± 1.4	77.4 ± 0.9	81.7 ± 1.0	78.7 ± 2.3	73.6 ± 2.3	67.8 ± 2.8
UIUCTex	81.3 ± 0.8	75.2 ± 1.4	81.0 ± 1.0	76.0 ± 0.8	83.5 ± 0.8	80.4 ± 0.8
Xerox7	88.9 ± 1.3	87.8 ± 0.5	80.5 ± 0.6	79.9 ± 0.9	66.6 ± 1.8	65.6 ± 1.5

in most cases, but there is no clear winner overall and in Xerox7 the random method is preferred even for small numbers of samples.

6 Influence of the Codebook

This section studies the influence of the vector quantization codebook size and construction method on the classification results.

Codebook size. The number of codebook centres is one of the major parameters of the system, as observed, e.g. by [1], who report that performance improves steadily as the codebook grows. We have run similar experiments, using online (rather than classical) k-means, testing larger codebooks, and studying the relationship with the number of patches sampled in the test image. Figure 5 shows the results. It reports means of multi-class error rates over 6 runs on the Xerox7 dataset for the three detectors. The other settings are as before. For each detector there are initially substantial gains in performance as the codebook size is increased, but overfitting becomes apparent for the large codebooks shown here. For the keypoint based methods there is also evidence of overfitting for

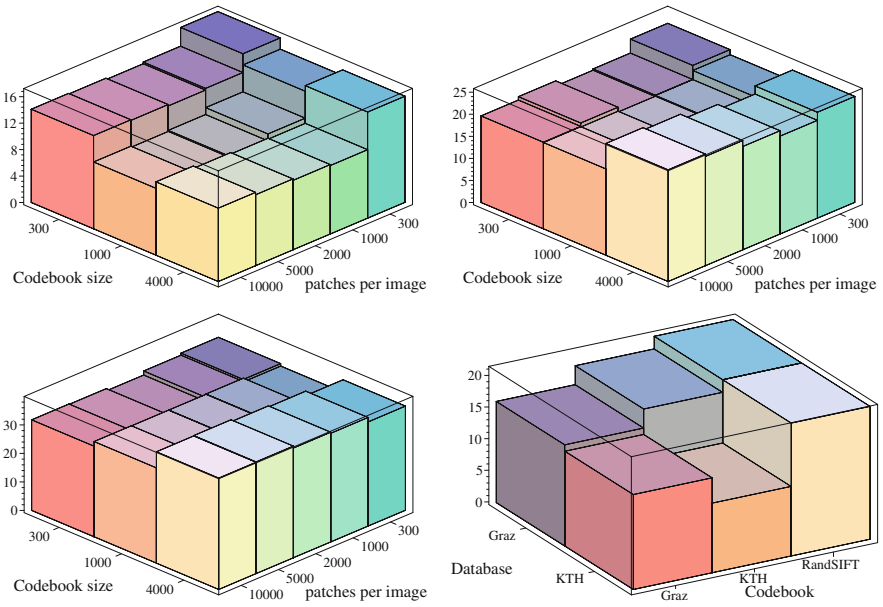

Fig. 5. All but bottom right: The influence of codebook size and number of points sampled per image for: random patches (top left); LoG detector (top right) and Harris detector (bottom left). Bottom right: the influence of the images used to construct the codebook, for KTH, Graz, and random SIFT vector codebooks on the KTH texture dataset and the Graz object dataset. The values are the means of the per-class classification error rates.

large numbers of samples, whereas the random sampler continues to get better as more samples are drawn. There does not appear to be a strong interaction between the influence of the number of test samples and that of the codebook size. The training set size is also likely to have a significant influence on the results but this was not studied.

Codebook construction algorithm. §4 presented two methods for constructing codebooks: randomly selecting centres from among the sampled training patches, and online k-means initialized using this. Table 1 compares these methods, again using 1000 element codebooks, MI-based normalization and a linear SVM classifier. 1000 patches per image are sampled (less if the detector can not return 1000). Except in one case (KTH-TIPS with random patches), the online k-means codebooks are better than the random ones. The average gain (2.7%) is statistically significant, but many of the individual differences are not. So we see that even randomly selected codebooks produce very respectable results. Optimizing the centres using online k-means provides small but worthwhile gains, however the gains are small compared to those available by simply increasing the number of test patches sampled or the size of the codebook.

Images used for codebook construction. One can also ask whether it is necessary to construct a dedicated codebook for a specific task, or whether a codebook constructed from generic images suffices (c.f. [13]). Figure 5(bottom right) shows mean error rates for three codebooks on the KTH-Tips texture dataset and the Graz object dataset. Unsurprisingly, the KTH codebook gives the best results on the KTH images and the Graz codebook on the Graz images. Results are also given for a codebook constructed from random SIFT vectors (random 128-D vectors, not the SIFT vectors of random points). This is clearly not as good as the codebooks constructed on real images (even very different ones), but it is much better than random: even completely random codings have a considerable amount of discriminative power.

7 Influence of Histogram Normalization Method

Coding all of the input images gives a matrix of counts, the analogue of the document-term matrix in text analysis. The columns are labelled by codebook elements, and each row is an unnormalized histogram counting the occurences of the different codebook elements in a given image. As in text analysis, using raw counts directly for classification is not optimal, at least for linear SVM classifiers (e.g. [22]), owing to its sensitivity to image size and underlying word frequencies. A number of different normalization methods have been studied. Here we only compare two, both of which work by rows (images) and binarize the histogram. The first sets an output element to 1 if its centre gets any votes in the image, the second adaptively selects a binarization threshold for each centre by maximizing the mutual information between the resulting binary feature and the class label over the training set [22]. As before we use 1000 element codebooks, online k-means, and a linear SVM. Results for two datasets are shown in figure 6 – other datasets give similar results.

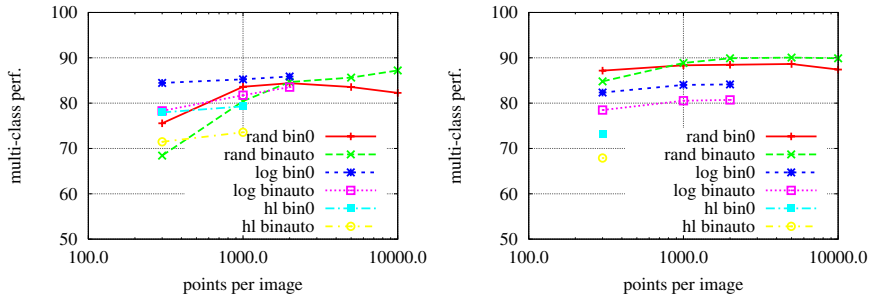

Fig. 6. The influence of histogram normalization on mean classification rate, for the Pascal-01 (left) and Xerox7 (right) datasets. Histogram entries are binarized either with a zero/nonzero rule (bin0) or using thresholds chosen to maximize mutual information with the class labels (binauto). Adaptive thresholding is preferable for dense sampling when there are many votes per bin on average.

Neither method predominates everywhere, but the MI method is clearly preferred when the mean number of samples per bin is large (here 10000 samples/image vs. 1000 centres). For example, on Xerox7, at 1000 samples/image the input histogram density is 27%, rising to 43% at 10000 samples/image. MI-based binarization reduces this to 13% in the later case, allowing the SVM to focus on the most relevant entries.

8 Influence of the Minimum Scale for Patch Sampling

Ideally the classifier should exploit the information available at all scales at which the object or scene is visible. Achieving this requires good scale invariance in the patch selection and descriptor computation stages and a classifier that exploits fine detail when it is available while remaining resistant to its absence when not. The latter is difficult to achieve but the first steps are choosing a codebook that is rich enough to code fine details separately from coarse ones and a binwise normalization method that is not swamped by fine detail in other bins. The performance of descriptor extraction at fine scales is critical for the former, as these contain most of the discriminative detail but also most of the aliasing and 'noise'. In practice, a minimum scale threshold is usually applied. This section evaluates the influence of this threshold on classification performance. As before we use a 1000 element codebook built with online k-means, MI-based normalization, and a linear SVM.

Figure 7 shows the evolution of mean accuracies over six runs on the Brodatz and Xerox7 datasets as the minimum scale varies from 1 to 3 pixels[7]. The performance of the LoG and Harris based methods decreases significantly as the minimum scale increases: the detectors return fewer patches than requested and useful information is lost. For the random sampler the number of patches is

[7] The other experiments in this paper set the minimum scale to 2. SIFT descriptors from the LAVA library use 4×4 blocks of cells with cells being at least 3×3 pixels, so SIFT windows are 12×12 pixels at scale 1.

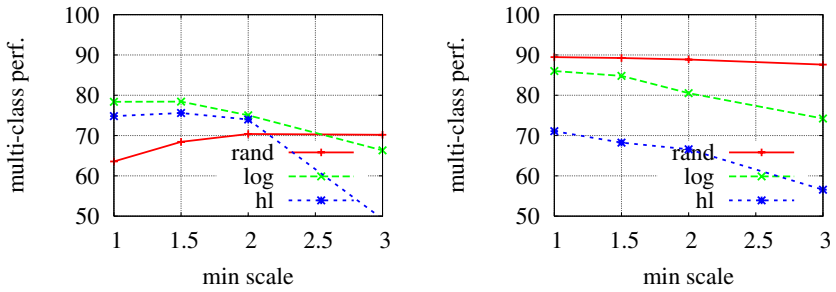

Fig. 7. The influence of the minimum patch selection scale for SIFT descriptors on the Brodatz (left) and Xerox7 (right) datasets

constant and there is no clear trend, but it is somewhat better to discard small scales on the Brodatz dataset, and somewhat worse on the Xerox7 dataset.

9 Results on the Pascal Challenge Dataset

The previous sections showed the usefulness of random sampling and quantified the influence of various parameters. We now show that simply by sampling

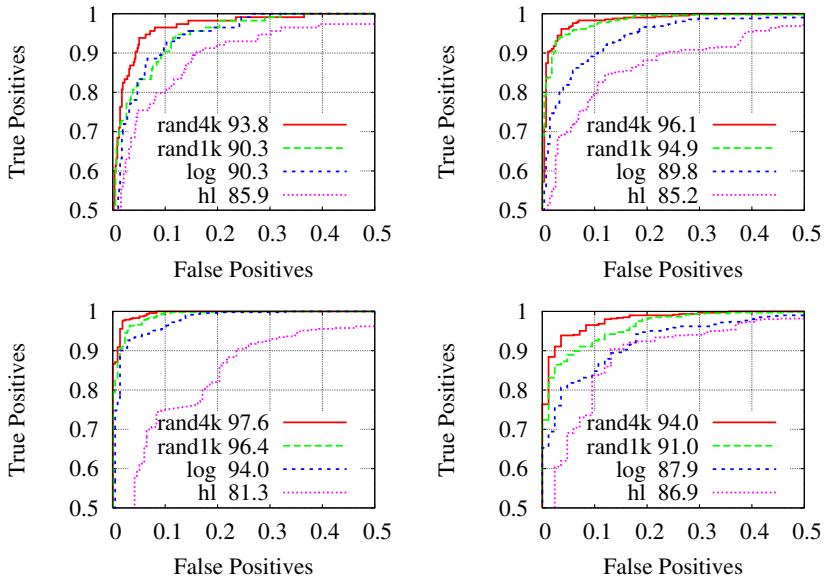

Fig. 8. ROC curves for the 4 categories of the PASCAL 2005 VOC challenge: top-left, bikes; top-right, cars; bottom-left, motorbikes; bottom-right, persons. The codebooks have 1000 elements, except that rand4k has 4000. Equal Error Rates are listed for each method.

Table 2. A comparison of our Rand4k method with the best results obtained (by different methods) during the PASCAL challenge and with the interest point based method of Zhang *et al.*

Method	motorbikes	bikes	persons	cars	average
Ours (rand4k)	97.6	93.8	94.0	96.1	95.4
Best Pascal [23]	97.7	93.0	91.7	96.1	94.6
Zhang *et al* [24]	96.2	90.3	91.6	93.0	92.8

large enough numbers of random patches, one can create a method that outperforms the best current approaches. We illustrate this on the Pascal-01 dataset from the 2005 PASCAL Visual Object Classification challenge because many teams competed on this and a summary of the results is readily available [23]. We use the following settings: 10 000 patches per image, online k-means, MI-based normalization, an RBF SVM with kernel width γ set to the median of the pairwise distances between the training descriptors, and either a 1000 element ('Rand1k') or 4000 element ('Rand4k') codebook. Figure 8 presents ROC curves for the methods tested in this paper on the 4 binary classification problems of the Pascal-01 Test Set 1. As expected the method Rand4k predominates. Table 2 compares Rand4k to the best of the results obtained during the PASCAL challenge [23] and in the study of Zhang *et al* [24]. In the challenge ('Best Pascal' row), a different method won each object category, whereas our results use a single method and fixed parameter values inherited from experiments on other datasets. The method of [24] uses a combination of sophisticated interest point detectors (Harris-Scale plus Laplacian-Scale) and a specially developed Earth Movers Distance kernel for the SVM, whereas our method uses (a lot of) random patches and a standard RBF kernel.

10 Conclusions and Future Work

The main goal of this article was to underline a number of empirical observations regarding the performance of various competing strategies for image representation in bag-of-features approaches to visual categorization, that call into question the comparability of certain results in the literature. To do this we ran head to head comparisons between different image sampling, codebook generation and histogram normalization methods on a cross-section of commonly used test databases for image classification.

Perhaps the most notable conclusion is that although interest point based samplers such as Harris-Laplace and Laplacian of Gaussian each work well in some databases for small numbers of sampled patches, they can not compete with simple-minded uniform random sampling for the larger numbers of patches that are needed to get the best classification results. In all cases, the number of patches sampled from the test image is the single most influential parameter governing performance. For small fixed numbers of samples, none of HL, LOG and random dominate on all databases, while for larger numbers of samples

random sampling dominates because no matter how their thresholds are set, the interest operators saturate and fail to provide enough patches (or a broad enough variety of them) for competitive results. The salience cues that they optimize are useful for sparse feature based matching, but not necessarily optimal for image classification. Many of the conclusions about methods in the literature are questionable because they did not control for the different numbers of samples taken by different methods, and 'simple' dense random sampling provides better results than more sophisticated learning methods (§9).

Similarly, for multi-scale methods, the minimum image scale at which patches can be sampled (e.g. owing to the needs of descriptor calculation, affine normalization, etc.) has a considerable influence on results because the vast majority of patches or interest points typically occur at the finest few scales. Depending on the database, it can be essential to either use or suppress the small-scale patches. So the practical scale-invariance of current bag-of-feature methods is questionable and there is probably a good deal of unintentional scale-tuning in the published literature.

Finally, although codebooks generally need to be large to achieve the best results, we do see some evidence of saturation at attainable sizes. Although the codebook learning method does have an influence, even randomly sampled codebooks give quite respectable results which suggests that there is not much room for improvement here.

Future work. We are currently extending the experiments to characterize the influence of different clustering strategies and the interactions between sampling methods and classification more precisely. We are also working on random samplers that are biased towards finding more discriminant patches.

References

1. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV'04 workshop on Statistical Learning in Computer Vision. (2004) 59–74
2. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR03. (2003) II: 264–271
3. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. IJCV **43** (2001) 29–44
4. Agarwal, S., Awan, A., Roth, D.: Learning to detect objects in images via a sparse, part-based representation. PAMI **26** (2004) 1475–1490
5. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from google's image search. In: ICCV. (2005) II: 1816–1823
6. Grauman, K., Darrell, T.: Efficient image matching with distributions of local invariant features. In: CVPR05. (2005) II: 627–634
7. Leibe, B., Schiele, B.: Interleaved object categorization and segmentation. In: BMVC. (2003)
8. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91–110
9. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: ECCV. (2002) I: 128

10. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV03. (2003) 1470–1477
11. Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for recognition. In: ECCV. (2000) I: 18–32
12. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. In: ICCV. (2005)
13. Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. In: ICCV. (2005)
14. Bouchard, G., Triggs, B.: Hierarchical part-based visual object categorization. In: CVPR. Volume 1. (2005) 710–715
15. Agarwal, A., Triggs, B.: Hyperfeatures – multilevel local coding for visual recognition. In: ECCV. (2006)
16. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: ECML-98, 10th European Conference on Machine Learning, Springer Verlag (1998) 137–142
17. Niblack, W., Barber, R., Equitz, W., Flickner, M., Glasman, D., Petkovic, D., Yanker, P.: The qbic project: Querying image by content using color, texture, and shape. SPIE **1908** (1993) 173–187
18. Lazebnik, S., Schmid, C., Ponce, J.: Affine-invariant local descriptors and neighborhood statistics for texture recognition. In: ICCV. (2003) 649–655
19. Rubner, Y., Tomasi, C., Guibas, L.: The earth mover's distance as a metric for image retrieval. IJCV **40** (2000) 99–121
20. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. Int. J. Computer Vision **65** (2005) 43–72
21. Lindeberg, T.: Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. IJCV **11** (1993) 283–318
22. Nowak, E., Jurie, F.: Vehicle categorization: Parts for speed and accuracy. In: VS-PETS workshop, in conjuction with ICCV 05. (2005)
23. *et al.*, M.E.: The 2005 pascal visual object classes challenge. In Springer-Verlag, ed.: First PASCAL Challenges Workshop. (2006)
24. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classifcation of texture and object categories: An in-depth study. Technical Report RR-5737, INRIA Rhône-Alpes, 665 avenue de l'Europe, 38330 Montbonnot, France (2005)