Creating Efficient Codebooks for Visual Recognition

Frédéric Jurie, Bill Triggs

Creating Efficient Codebooks for Visual Recognition

Frederic Jurie and Bill Triggs
GRAVIR-INRIA-CNRS, 655 Avenue de l'Europe, Montbonnot 38330, France
{*Frederic.Jurie,Bill.Triggs*}*@inrialpes.fr, http://lear.inrialpes.fr*

Abstract

Visual codebook based quantization of robust appearance descriptors extracted from local image patches is an effective means of capturing image statistics for texture analysis and scene classification. Codebooks are usually constructed by using a method such as k-means to cluster the descriptor vectors of patches sampled either densely ('textons') or sparsely ('bags of features' based on keypoints or salience measures) from a set of training images. This works well for texture analysis in homogeneous images, but the images that arise in natural object recognition tasks have far less uniform statistics. We show that for dense sampling, k-means over-adapts to this, clustering centres almost exclusively around the densest few regions in descriptor space and thus failing to code other informative regions. This gives suboptimal codes that are no better than using randomly selected centres. We describe a scalable acceptance-radius based clusterer that generates better codebooks and study its performance on several image classification tasks. We also show that dense representations outperform equivalent keypoint based ones on these tasks and that SVM or Mutual Information based feature selection starting from a dense codebook further improves the performance.

1. Introduction

Representations based on loose collections of invariant appearance descriptors extracted from local image patches have become very popular for texture analysis and visual recognition [1, 5, 7, 9, 12, 17, 18, 19, 25, 24, 26, 27]. They are flexible and relatively easy to construct, they capture a significant proportion of the complex statistics of real images and visual classes in a convenient local form, and they have good resistance to occlusions, geometric deformations and illumination variations. In particular, the statistics of natural image classes can be encoded by vector quantizing the appearance space and accumulating histograms or signatures of patch appearances based on this coding. Traditionally one codes a dense set of patches ('texton' representation) [18], but sparser sets based on keypoints or 'points of interest' detected by invariant local feature detectors have generated a lot of interest recently [1, 5, 7, 17, 26, 27]. Keypoints potentially provide greater invariance and more compact coding, but they were not designed to select the most informative regions for classification and, as we will show below, dense sampling followed by explicit discriminative feature selection gives significantly better results.

Figure 1: Top: sample images from four categories of the 'Xerox 7' dataset. Bottom: the image regions assigned to the 100 codewords with maximal inter-category discrimination in this dataset. The codewords represent meaningful object parts, even though they were learned without using the category labels.

Both sparse and dense codebooks are typically created from a set of training images using a standard clustering algorithm such as k-means. This works well for texture analysis on images containing only a few homogeneous regions, and adequately for keypoint based representations in general, but we will show that it is suboptimal for recognition using dense patches from natural scenes. The wide range of region sizes in such scenes leads to a highly non-uniform, power-law-like sample distribution in descriptor space. This leads k-means to choose suboptimal codebooks in which most of the centers cluster near high density regions [11, 3], thus under-representing equally discriminant low-to-medium density ones. §3 describes our

This work was partially supported by the EU network PASCAL and the French ACI grant MOVISTAR. It benefited from discussions with several members of our team including A. Agarwal, G. Dorkó, D. Larlus, B. Ninassi, E. Nowak and J. Zhang.

method for allocating centers more uniformly, inspired by mean shift [4] and on-line facility location [20].

Finally, most current methods for learning codebooks are *generative* – they capture overall image statistics but not necessarily the fine distinctions needed to discriminate between classes. This can be palliated by generating a large codebook and selecting a discriminant subset of 'parts' (informative classes of patches) from it – see fig. 1 and §4.4. The selected parts tend to have intermediate probabilities, allowing many rare parts to be discarded while still preserving a fairly sparse and hence compact representation.

A codebook algorithm contains three components: the patch selector, the appearance descriptor, and the quantization algorithm. This paper focuses on the first and last. Appearance is represented simply as vectors of raw pixel intensities compared using ZNCC, but other vector valued appearance descriptors can be used and informal experiments with SIFT descriptors [19] gave similar results.

There are indications that despite their good matching performance, existing keypoint detectors (Förstner, Harris-affine, *etc*.) often fail to detect the most informative patches for image classification [14]. Another strategy is to select patches by testing many potential templates and selecting the ones that maximize an informativeness score such as mutual information [25, 24]. We will do this, densely sampling patches at all image scales, quantizing, then selecting the most informative centres using a separate feature selection algorithm.

1.1. Previous Work

By 'textons' (*c.f*. [13]) we mean a set of representative classes of image patches that suffice to characterize an image object or texture. Such textons are typically extracted densely, representing small and relatively generic image micro-structures [10, 18, 28] – blobs, bars, corners, *etc*. When larger, more semantically meaningful structures are wanted, sparse sampling is typically used. For example, Weber *et al*. [26] learn a generative model for face recognition from a set of training images by detecting Förstner keypoints at a single scale and quantizing the selected (small) raw image patches using k-means. Similarly, Leung *et al*. [18] use Gabor texture features for texture classification. Recently, several groups (*e.g*. [5, 7, 15, 19]) have proposed 'bag of features' methods based on normalized patches or SIFT descriptors [19] over Difference of Gaussian, Harris-scale or Harris-affine keypoints [22], vector quantized using k-means. Leibe *et al*. [17] proposed a similar method based on a codebook learned by agglomeratively clustering raw pixel intensity vectors of Harris keypoints. Agarwal *et al*. [1] adopted a similar approach incorporating spatial relations between parts.

2. Local Appearance Statistics

To avoid the suboptimal choices made by keypoint detectors, we base our representation on densely sampled patches. The inter-class discrimination provided by k-means codebooks is highly dependent on the sample distribution in descriptor space, degrading as this becomes more nonuniform. K-means works well for images containing only uniform texture patches, and tolerably well for keypoint based descriptors in more general images, although the distribution is already quite nonuniform in this case. But with densely sampled descriptor sets in general images, the distribution becomes so nonuniform that k-means devotes most of its centres to the immediate neighborhood of the central peak, and the coding suffers. The nonuniformity is essentially due to two effects: (i) certain patches (uniform regions, step edges, *etc*.) occur far more frequently than others (faces, stop signs); and (ii) the amount of any given texture that appears is extremely variable owing to the multiscale region structure of natural scenes (the statistics of this is captured quite well by "dead leaves" [16] – sets of opaque, homogeneous patches occurring randomly at all positions and scales – and similar fractal models). K-means centres drift towards high density regions owing to the 'mean shift like' k-means update rule. Asymptotically, they are distributed according to the underlying density, *i.e*. in the same way as random samples.

This point is critical because the patches that are most informative for classification tend to have intermediate frequencies. Over-frequent patches typically contain generic image structures like edges that occur often in all classes, giving little discriminant information. Conversely, rare patches may be highly discriminant when they occur, but they occur so seldom that they have little influence on overall performance. These effects are probably even greater in man-made environments, where large uniform uninformative regions are common.

To illustrate the non-uniformity, we use our (below) clustering algorithm to analyze the probability density of variance-normalized patches from natural images. These are 11×11 gray level patches, sampled densely from multiscale pyramids with 10 layers spaced by $2^{\frac{1}{4}}$ in scale. Other patch sizes from 6×6 to 20×20 give very similar results. For now, the only important properties of our clusterer are that it chooses the densest regions first and fills them with uniformly-sized balls. Hence, any nonuniformity in the resulting texton counts is direct evidence of nonuniformity at the scale of the balls in the underlying distribution.

We have run several experiments using different image collections: one containing 500 natural images of width 300–800 pixels, and several others that include more specific objects (see §4 for details). Fig. 2 (left) shows the statistics for two of these image sets. The bin densities roughly follow a power law (exponent ≈ -1.6) – *c.f*. Zipf's

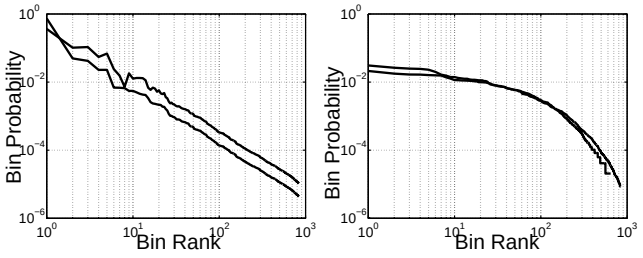

Figure 2: Log-log plots of the local density of image features for the most probable 800 bins, for densely sampled patches (left) and for keypoint based patches (right). The distribution is markedly more peaked for dense patches than for keypoints.

law from textual word frequency analysis [29]. The law persists over a wide range of natural image types and bin probabilities, flattening out to equiprobability only for very rare centers. In contrast, as shown in fig. 2 (right), the corresponding distribution for patches selected using a keypoint detector (here, Lowe's [19]) is much more uniform, with most of the bins having comparatively similar probabilities and only a few being rare.

3. Clustering Algorithm

Existing Methods: Before describing our clustering algorithm, we give a brief review of vector quantization methods for large sets of vectors in high dimensional spaces. In our case the feature dimension is in the hundreds (121 for 11×11 patches) so quantization based on a uniform grid of centers is infeasible and clustering is the preferred method for finding the centers. With densely sampled patches the number of samples to be clustered can be more than 10^8 for medium-sized datasets.

Feature spaces are typically designed to capture perceptually meaningful distinctions while providing good resistance to extraneous details such as changes in illumination. In a well-chosen representation, distances in feature space thus tend to coincide roughly with distinguishability with respect to such perceptual "noise". This implies that a perceptually efficient coding should have roughly uniform cell sizes, or at least enforce a lower bound on the cell size: it is wasteful to code dense regions more finely than the underlying "noise" level.

Owing to its simplicity, k-means is a popular algorithm for perceptual coding. Unfortunately it has the defect that cluster centers are drawn irresistibly towards denser regions of the sample distribution by the 'mean shift' like process that is used to update them, with the result that they tend to be tightly clustered near dense regions and sparsely spread in sparse ones. This effect is especially pronounced in high dimensions where it is infeasible to tile the region of the peak densely with cells. The typical result is a cluster of centers surrounding the peak, with Voronoi cells that extend radially outwards for a significant distance, each including a few more distant points. This gives a perceptually very nonuniform coding. K-means has other drawbacks. It is non-robust – points lying far from any of the centers can significantly distort the position of the centre that they are assigned to – and the number of centers must be known in advance, whereas the overall goal is to code an unknown number of "distinguishable parts".

Several authors have used agglomerative clustering for the centre allocation problem [17, 1]. The proposed methods are rather inefficient but there are interesting alternatives such as the on-line facility location algorithm [20]. These approaches handle the unbalanced density problem well, but they can not be applied to large datasets owing to their high algorithmic complexity. Mean shift has been used in a similar context [11], but it can not be used directly with natural images because informative parts seldom coincide with well-separated density modes.

Several studies of clustering for large data sets have appeared in the data mining literature. Such methods typically use some form of subsampling for speed [6, 21]. In practice, subsampling based methods have trouble with unbalanced densities, especially if uniform subsampling is used [8]. Like k-means, they tend to allocate most of their centres to a few dense regions thus 'starving' lower density ones. Proposed solutions include different forms of resampling such as random oversampling, random undersampling, and importance weighting methods that adjust the cost of the various regions to counter the imbalance [3].

Our algorithm: Our proposed strategy combines the advantages of on-line clustering [20] and mean-shift [4] in an undersampling framework [8]. The algorithm produces an ordered list of centers, with the quantization rule that patches are assigned to the first centre in the list that lies within a fixed radius r of them, or left unlabeled if there is no such centre.

The parameters of the algorithm are the number of subsamples N, the cell radius r and the mean shift radius h (which we always set to r). Centers are created one by one in list order. For each entry, the centre is set when it is added to the list and never changed after that. At each step, we draw N patches uniformly and randomly from the currently unlabeled part of the data set, compute their maximal density region by running a mean-shift estimator [4] with a Gaussian kernel of width h on them, allocate a new centre at the maximal density position, and (notionally – in practice this is done lazily) sweep the dataset, labeling and eliminating all of the patches that are assigned to this centre. Eliminating the labeled patches prevents the algorithm from repeatedly assigning centers to the same high density region. As we are mainly interested in coding regions of intermediate probability, the algorithm can either be stopped

Figure 3: The 100 most frequent codewords produced by quantizing the Agarwal *et al*. [1] dataset. The parameters of the algorithm were N=1000 and h=r=0.8 (which gives clusters with a fairly loose set of appearances).

by monitoring the informativeness of the clusters created (see §4), or simply run until a "sufficiently large" number of clusters has been found.

The computational complexity is governed mainly by the cost of the mean shift iterations, which in practice (given the high dimension) need to be initialized at each of the N sample centers in turn to ensure reliable mode-finding. This quadratic cost is acceptable because it is applied to only to a small (N element) subset of the available patches. Finding and eliminating the labeled patches is at worst linear in the number of training samples, times the average number of centers that need to be tested before elimination. In practice, the centers are found in roughly decreasing order of frequency, so the average number of centres tested is effectively independent of the total number of centres in the list (although it does depend strongly on r). In the experiments below, the average number of centers tested per vector is generally less than 9 for codebooks of several thousand centers. As an illustration, fig. 3 shows the first 100 codewords for the Agarwal-Roth set of side views of cars.

4. Categorization and Object Detection

We demonstrate the performance of our codebooks with experiments on object recognition and image categorization. For simplicity we formulate categorization as supervised learning over a 'bag of features' representation [5, 7, 15]. Despite the fact that they capture only local statistics and ignore geometric relationships, bags of features perform surprisingly well. Here they provide a convenient baseline for comparing the performance of different codebooks, without the complexities inherent in strategies that incorporate more geometry.

We compared two linear classifiers: simple Naive Bayes and linear Support Vector Machines. SVM's are known to produce reliable results in high-dimensional problems, and here they easily outperform Naive Bayes in all of the tests performed – see fig 6.

We tested two ways of producing feature vectors from codebook based labelings: binary indicator vectors and histograms. In the histogram representation, the feature vector is just the normalized (scaled to total sum 1) histogram of occurrence counts for the different codewords. For indicator vectors, each codeword's feature is 1 if a patch with the codeword occurs in the image and 0 otherwise [25]. In fact, rather than using simple presence/absence, we threshold the number of occurrences of the codeword in the image to produce the binary vector. The thresholds are selected automatically to maximize the mutual information (MI) between the thresholded codeword frequency and the class.

Figure 4: Three images from each class of the 'Xerox 7' dataset [27]: faces, buildings, trees, cars, phones, bikes and books. Intra-class variability makes the categorization problem quite challenging.

The size of the codebooks is potentially very large so there may be overfitting (too many "noise" features for good classification). To show how the approach behaves when only a restricted subset of the vocabulary is used, we report on three different feature selection methods: maximization of mutual information (MI) [2, 25], and of odds ratio (OR) [2], and training an initial linear SVM classifier on the full feature set and retaining only the features that have the highest weights (SVM) [2, 23]. Information Gain was also tested [2, 25] but performed less well than MI.

The number of possible combinations of descriptor types, codebooks, feature selectors and classifiers is quite large, so only the most relevant combinations were tested.

4.1. Data sets

We tested on three data sets. The first, side views of cars from Agarwal & Roth [1], was designed to evaluate object detection algorithms. Performance is measured by precision/recall as suggested in [1].

The second, 'Xerox 7' [27], contains 1776 images from seven categories. Fig. 4 shows some examples. The object poses are highly variable and there is a significant amount of background clutter, some of which belongs to the other categories (notably buildings, trees and cars), making the classification task fairly challenging. We report the confusion matrix and the overall error rate under 10-fold cross-validation with a standardized set of folds.

The last dataset contains 4 classes from the ETH80 database [17]: models of cars, horses, dogs and cows. At the individual patch level, the horses, dogs and cows have

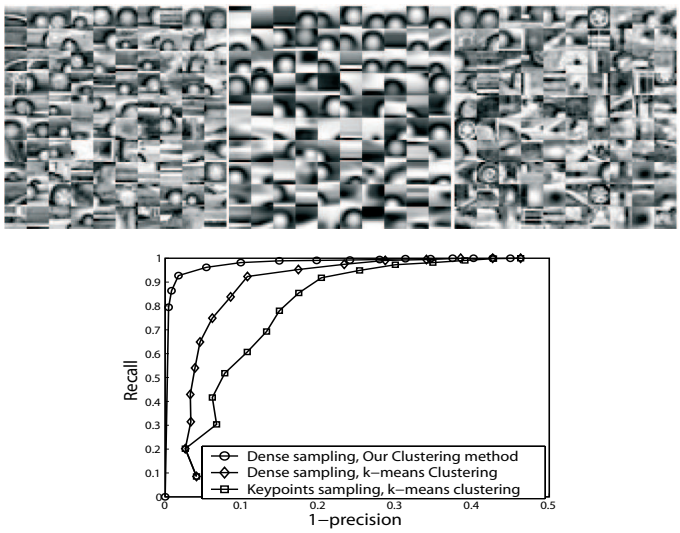

Figure 5: MI based feature selection on the Agarwal-Roth car data set. Top: the 100 best codewords for dense codebooks based on our clusterer (left) and k-means (middle), and a keypoint codebook based on k-means (right). Note the relative lack of diversity in the dense k-means codebook. Bottom: the Precision-Recall curves for these codebooks, using a linear SVM classifier over 600 features selected using MI. Our dense approach performs best.

rather similar appearances. There is no background, and hence neither visual context to help, nor clutter to hinder, the classification. We report overall classification error rates under 10-fold cross-validation.

4.2. Dense versus Sparse Codebooks

The first experiment uses the Agarwal & Roth test set [1]. The problem is to detect the objects (car sides) and return their image locations. There are 500 positive and 500 negative 100×400 pixel training images at 10 scale levels from 100×400 to 17×70. Some positive images are shown in fig. 3 (left). We report results for three 2500-centre codebooks based on both positive and negative images from the training set: one built by densely sampling patches and using our clustering method; one built using dense patches and k-means; and one built with k-means using keypoints detected by Lowe's DoG detector [19]. In each case we select the 600 best codewords by maximizing the mutual information between optimally thresholded codeword occurrence counts and categories, and use a linear SVM classifier with 10-fold cross validation. Fig. 5 shows the Precision-Recall curves. The proposed method clearly outperforms the k-means based dense one, which in turn outperforms the keypoint based one. Further experiments with different classifiers and codebook sizes confirmed these relative performances. Comparisons of our method with agglomerative, k-means and on-line k-means clusterers on several other datasets in the sparse case gave similar results.

We also ran similar experiments on the ETH80 dataset. Fig. 6 (top left) shows the overall error for various numbers of centers, selection methods and classifiers. For every combination of feature selector and classifier tested, our clustering method produced the best codebook.

One major determinant of classifier performance is the number of image patches used to compute the test image histogram. Fig. 6 (top centre) shows the overall misclassification rate for the 'Xerox 7' dataset as a function of this number, for randomly sampled and keypoint based test patches and using a fixed codebook built using our dense approach. Randomly selected patches perform at least as well as keypoint based ones and in both cases performance increases substantially as more patches are sampled, asymptotically reaching that of the dense method (the last 'random' point). Similar results appear to hold for keypoint based codebooks.

There are two main conclusions: (i) adopting a sparse, keypoint based image representation instead of a densely sampled one often leads to a significant loss of discriminative power; (ii) with dense sampling, k-means produces poor codebooks – it is better to use more evenly distributed coding centres, *e.g.* based on an algorithm that enforces fixed-radius clusters.

4.3. Histogram Encoding

We evaluated two methods of converting raw codeword occurrence counts to classification features: normalized histograms (normalizing the sum of the counts to one) and binary feature vectors (thresholding each count at the value that optimizes the MI w.r.t. the class labels on the training set, and coding the result as a binary vector). In each case the dimension of the descriptor vector is the number of codewords after feature selection. Histograms performed better for the Agarwal-Roth and Xerox 7 datasets – see fig. 6 (top right, bottom left & centre) – but the differences are small.

Hoping to reduce the effects of the very non-uniform distribution of occurrence counts, we also tested feature vectors produced by ranking the codeword counts and using $1/(1+\mathrm{rank})$ as a codeword feature, but the results were similar to or worse than those of standard histograms. Histogram features thus seem to be a reasonable choice for codebook based visual detection and categorization.

4.4. Feature Selection and Informativeness

When constructing codebooks for visual learning, it is important to include codewords in regions with intermediate occurrence probabilities. These codewords are not the first ones to be produced by our algorithm, so the initial codebook will generally be larger than the final number of useful features, and subsequent feature selection is advis-

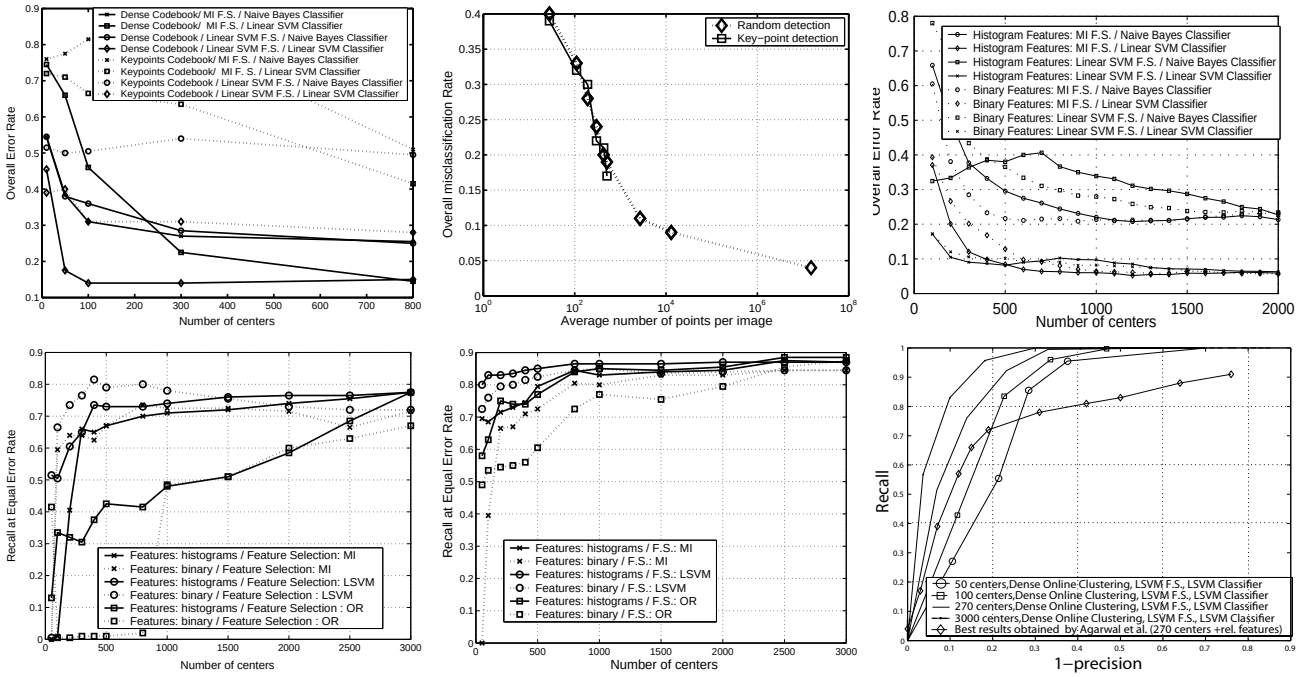

Figure 6: **Top left:** ETH80 dataset. For all feature selectors and classifiers tested, our dense codebook outperforms keypoint-based k-means. **Top centre:** Classification performance depends critically on the number of patches sampled from the test image. Keypoint based patches do no better than random or sparse grid sampling. This is with our dense codebook on the Xerox 7 dataset, but similar results appear to hold for keypoint based codebooks. **Top right:** Our dense method on the Xerox 7 dataset. The linear SVM classifier over histogram features has the lowest error rate, with either MI (for many centres) or LSVM (for few centres) feature selection. **Bottom left & centre:** Car detection on the Agarwal-Roth data set using our dense codebooks. Naive Bayes (left) and linear SVM (centre) classifiers. LSVM feature selection with either histogram or binary based coding performs best. **Bottom right:** Our approach outperforms that of Agarwal *et al.*, even though it does not code spatial relations between features.

able. We studied three feature selection methods: mutual information (MI), odds ratio (OR) and linear SVM weights (LSVM). When there were more than two classes, the OR and LSVM criteria were calculated by averaging the corresponding two-class criteria over the possible one-against-all subproblems. Using the maximum value over the subproblems instead gave similar or slightly worse results.

LSVM selection topped most of our experiments (see fig. 6 top left, bottom left & centre), but on Xerox 7 MI was preferred (fig. 6 top right). However, note that the full codebooks generally outperformed any of the reduced ones, so feature selection should only be used if the computational cost of the algorithm needs to be reduced. Fig. 1 shows the parts corresponding to the top 100 MI codewords for a few images of the Xerox 7 dataset.

4.5. Comparing absolute performances

The high quality of our codebooks allows simple recognition schemes to outperform more complex ones based on other codebooks. Fig. 6 (bottom right) compares our system's precision/recall to that of Agarwal *et al.* [1]. We use (we believe) the same experimental and precision/recall protocol as [1]. Our precision at equal rate error is more than 10% higher than the best results of [1], despite the fact that we rely on a simple linear SVM classifier and incorporate no inter-patch geometry (whereas [1] uses spatial relations between pairs of features).

Label \ True	Faces	Bldgs	Trees	Cars	Phones	Bikes	Books
Faces	98.6	0	0	0.4	1.3	0.8	4.2
Bldgs	0.2	92.0	3.3	1.9	1.8	0	0.7
Trees	0.3	2.6	94.0	0	0	0.8	0
Cars	0.2	1.3	1.3	93.5	0.9	0.8	3.5
Phones	0.1	0	0.67	1.4	91.6	0	3.5
Bikes	0	0	0.6	0	0.4	96.8	1.4
Books	0.3	4.0	0	2.4	3.7	0.8	86.6

Figure 7: The confusion matrix for our method on Xerox7.

On the Xerox 7 dataset, the best results were obtained with our codebooks and a linear SVM. Fig. 7 shows the confusion matrix for this configuration, using 600 codewords. Our overall error rate is 4.8% with standard deviation 1.2%, whereas the (k-means and sparse feature based, but otherwise comparable) method of [27] has an error rate of 15%.

5. Summary and Conclusions

We have presented a simple but effective algorithm for building codebooks for visual recognition. The method is based on two key observations: (i) restricting attention to patches based at sparse keypoints often loses a significant amount of discriminant information, so densely sampled patches should be coded; and (ii) for densely sampled patches, descriptors are distributed very nonuniformly in descriptor space, which leads k-means-like clusterers to concentrate most of their centers in high density regions, thus 'starving' medium density ones and leading to poor codebooks. As an alternative, we described a simple fixed-radius clusterer based on mean shift, that scales well to large datasets and whose codebooks significantly outperform k-means based ones. The approach was validated on several 'bag-of-features' based object detection and image classification tasks, showing results that are clearly superior to other state-of-the-art approaches.

Future work: We are currently consolidating our results on dense versus keypoint methods and evaluating our approach on SIFT descriptors rather than normalized patches. We also hope to integrate discriminative information into the centre selection process so that it directly selects discriminant features.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *PAMI*, 26(11):1475–1490, November 2004.

[2] J. Brank, M. Grobelnik, N. Milic-Frayling, and D. Mladenic. Interaction of feature selection methods and linear classification models. In *ICML'02 Workshop on Text Learning*, 2002.

[3] N. Chawla, N. Japkowicz, and A. Kolcz, editors. *ICML'03 Workshop on Learning from Imbalanced Data Sets*, 2003.

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002.

[5] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV'04 workshop on Statistical Learning in Computer Vision*, pages 59–74, 2004.

[6] D. Cuttin, D. Karger, J. Pedersen, and J. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *SIGR*, pages 318–329, 1992.

[7] G. Dorko and C. Schmid. Selection of scale-invariant parts for object class recognition. In *ICCV*, pages 634–640, 2003.

[8] A. Estabrooks, T. Jo, and N. Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1):18–36, 2004.

[9] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, pages II: 264–271, 2003.

[10] D. Geman and A. Koloydenko. Invariant statistics and coding of natural microimages. In *IEEE Workshop on Statistical and Computational Theories of Vision*, 1999.

[11] B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions: a texture classification example. In *ICCV*, pages 456–463, 2003.

[12] B. Heisele, P. Ho, J. Wu, and T. Poggio. Face recognition: Component-based versus global approaches. *CVIU*, 91(1–2):6–21, July 2003.

[13] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290:91–97, 1981.

[14] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *CVPR*, pages II: 90–96, 2004.

[15] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 27(8):1265–1278, 2005.

[16] A. Lee, D. Mumford, and J. Huang. Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. *IJCV*, 41(1-2):35–59, January 2001.

[17] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *BMVC*, 2003.

[18] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, June 2001.

[19] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, November 2004.

[20] A. Meyerson. Online facility location. In *42nd IEEE Symp. Foundations of Computer Science*, pages 426–433, 2001.

[21] A. Meyerson, L. O'Callaghan, and S. Plotkin. A k-median algorithm with running time independent of data size. *Machine Learning*, 56(1–3):61–87, 2004.

[22] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *ECCV*, page I: 128, 2002.

[23] D. Mladenic, J. Brank, M. Grobelnik, and N. Milic-Frayling. Feature selection using linear classifier weights: Interaction with classification models. In *SIGIR*, pages 234–241, Sheffield, U.K., 2004.

[24] A. Torralba, K. Murphy, and W. Freeman. Sharing features: Efficient boosting procedures for multiclass object detection. In *CVPR*, pages II: 762–769, 2004.

[25] M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *ICCV*, pages 281–288, 2003.

[26] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *ECCV*, pages I: 18–32, 2000.

[27] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. In *Workshop on Learning for Adaptable Visual Systems (LAVS04), Cambridge, U.K.*, 2004.

[28] S. Zhu, C. Guo, Y. Wang, and Z. Xu. What are textons? *IJCV*, 62(1–2):121–143, April 2005.

[29] G. Zipf. *Selected Studies of the Principle of Relative Frequency in Language*. Harvard University Press, 1932.