Heterogeneous bag-of-features for object/scene recognition

Loris Nanni\textsuperscript{a,}\textsuperscript{*}, Alessandra Luminib

\textsuperscript{a} DEI, University of Padua, Viale Gradenigo 6, Padua, Italy
\textsuperscript{b} DEIS, Università di Bologna, Via Venezia 52, 47023 Cesena, Italy

A R T I C L E   I N F O

Article history:
Received 4 April 2012
Received in revised form 14 September 2012
Accepted 5 December 2012
Available online 3 January 2013

Keywords:
Object recognition
Bag-of-features
Interest region description
Texture descriptors
Machine learning
Support vector machine

A B S T R A C T

In this work we propose a method for object recognition based on a random selection of interest regions, heterogeneous set of texture descriptors and a bag-of-features approach based on several k-means clustering runs for obtaining different codebooks. The proposed system is not based on complex region detection as SIFT but on a simple exhaustive extraction of sub-windows of a given image. In the classification step an ensemble of random subspace of support vector machine (SVM) is used. The use of random subspace ensemble coupled to the principal component analysis for reducing the dimensionality of the descriptors permits to reduce the curse of dimensionality problem.

In the experimental section we show that the combination of classifiers trained using different descriptors permits a consistent improvement of the performance of the stand-alone approaches. The proposed system has been tested on four datasets: the VOC2006 dataset, in a wide-used scene recognition dataset, in the well-known Caltech-256 Object Category Dataset and in a landmark dataset, obtaining remarkable results with respect to other state-of-the-art approaches. The MATLAB code of our system is publicly available.

1. Introduction

As an evolution of over 10 years of research in the field of content based image retrieval (CBIR), recent years have witnessed increasing interest in image annotation and object recognition problems. Recently, along with the rapid progress in the application of local descriptors in pattern recognition, computer vision and image retrieval, the use of local features like keypoints or image patches has appeared as promising for several applications of object classification and image retrieval, including wide baseline matching for stereo pairs [15,28], object retrieval in videos [25], object recognition [19], texture recognition [18], robot localization [24], visual data mining [26], and symmetry detection [29].

One of the key issues for existing image annotation methods is to find an effective feature representation for images. Some approaches, e.g. [13,14], proposed constellation models to locate distinctive object parts and determine constraints on the spatial arrangement. The main drawbacks of these approaches are that spatial models typically could not handle significant deformations (i.e. large rotations, occlusions) and they did not consider objects with variable part numbers (i.e. buildings, trees).

Recently, there has been emerging consensus for the bag-of-words (BoW) model for image representation [19]. The BoW approach is based on powerful scale invariant feature descriptors which are used to represent regions in a given image that are covariant to a class of transformations and to match identical regions between images.

The most performing descriptors seem to be those based on histogram distributions [30]: for example the intensity-domain spin image [5], a histogram approach that represents regions using the distance from the center point and intensity values; the Scale-invariant feature transform (SIFT) descriptor [19], a histogram that takes the weighted gradient locations and orientations; the geodesic intensity histogram [31], it provides a deformation invariant local descriptor. Other well-known descriptors include the Principal Components Analysis-based SIFT (PCA-SIFT) [3], moment invariants [16], and complex filters [23].

Moreover, a number of powerful texture operators can be used in region description: for example in [17] a LBP-based texture descriptor (named center-symmetric local binary pattern, CS-LBP) is proposed, which is computationally simpler than SIFT and more robust to illumination problems. Another interesting result is reported in [21], where it is shown that, when the number of regions is quite large, random sampling gives equal or better classification.

\textsuperscript{*} Corresponding author.
E-mail addresses: loris.nanni@unipd.it (L. Nanni), alessandra.lumini@unibo.it (A. Luminii).
rates than the other more complex operators that are in common use.

Starting from these and other results, in this work we improve our previously published system for object recognition [9] considering the ideas reported in PCA-SIFT [3,4] and in [7]:

- We use both local and global descriptors to represent an image; we have tested and fused together several texture descriptors (i.e. local binary/ternary patterns, local phase quantization, histogram of oriented edges, Gabor like features, SIFT).
- We perform a dimensionality reduction of the texture descriptors using principal component analysis (according to the PCA-SIFT approach [3]) to deal with the problems of high correlation among the features and the curse of dimensionality.
- We use a BoW approach computing textons by clustering the descriptors of the regions of each class with k-means; by concatenating the textons over all the classes, we obtain a global texton vocabulary.
- We have built an ensemble not only combining different texture descriptors (local and global) but also considering different codebooks. Since k-means is an unstable clustering approach, it is performed several times to create different codebooks (one for each run), and for each codebook a classifier is trained.
- We represent each image as a histogram of texton labels, and these histograms are used to train a random subspace ensemble (RS) of support vector machines (SVM) classifiers. It has been reported in several papers (e.g. [20]) that RS of SVM performs better than a stand-alone SVM classifier and it is able to successfully manage high dimensional data and to reduce the curse of dimensionality problem.

The novelty of this paper does not consist in presenting a new descriptor, but in having performed extensive tests with many of the state-of-the-art descriptors and showed that they can be combined in the same framework gaining excellent results.

A full-feature MATLAB toolbox containing the source code for the all the steps of the proposed system is available at bias.cs.unibo.it (nanni) objH.rar. Hoping that it will serve as the foundation for further explorations by other researchers in the field.

The rest of the paper is organized as follows: in Section 2 the texture descriptors used in our system are briefly reviewed, in Section 3 the proposed approach for object/scene recognition is explained in details, in Section 4 experimental result are discussed and in Section 5 the conclusions are drawn.

2. Descriptors

In this work we test several state-of-art descriptors that are potentially useful for object/scene classification.

2.1. SIFT descriptor

The SIFT descriptor [19] is a 3D histogram that takes the gradient locations (quantized into a 4 × 4 location grid) and orientations (quantized into eight values) and weights them by the gradient magnitude and a Gaussian window superimposed over the region. The SIFT descriptor, which is obtained by concatenating the orientation histograms over all bins and normalized to unit length, explains how the local gradients around a point are aligned and distributed at different scales.

2.2. Local ternary patterns

Local ternary patterns (LTPs) are a quite recent variant [27] of the widely used local binary patterns (LBPs). The LBP operator is a rotation invariant operator which evaluates the binary difference between the gray value of a pixel x and the gray values of neighboring pixels on a circle of radius R around x. LBP is a powerful texture descriptor based on occurrence histogram of the LBP operator. Conventional LBP is sensitive to noise in the near-uniform image regions: LTP solve this problem by using a three value encoding. Our implementation of LTP is a modification of the original LBP Matlab code which includes a three value encoding and normalized histogram, we have used both rotation invariant and uniform bins (named LTP-R and LTP-U). Each descriptor, obtained concatenating the features extracted with (R = 1, P = 8) and (R = 2, P = 16), is used to train a different classifier.

2.3. Local phase quantization (LPQ)

The local phase quantization operator (LPQ) is a texture descriptor [22] based on the blur invariance property of the Fourier phase spectrum. LPQ uses the local phase information extracted from the 2-D short-term Fourier transform computed over a rectangular neighborhood of radius R at each pixel position of the image. In LPQ only four complex coefficients, corresponding to 2-D frequencies, are considered and quantized using a scalar quantizer between 0 and 255. The final descriptor is the normalized histogram of the LPQ values. In our experiment two different LPQ descriptors are extracted by varying the parameter R (R = 3 and R = 5 in our final system, named LPQ(3) and LPQ(5)), each descriptor is used to train a different classifier.

2.4. GIST

The GIST descriptor [10] computes the energy of a bank of Gabor-like filters evaluated at 8 orientations and 4 different scales. The square output of each filter is then averaged on a 4 × 4 grid.

2.5. The histogram of oriented edges (HOG)

In this work a 2 × 2 version of the Histogram of Oriented Edges (HOG) [4] has been used. The HOG features are extracted on a regular grid at steps of 8 pixels, and stacked together considering sets of 2 × 2 neighboring to form a longer descriptor which provides more descriptive power.

3. Proposed approach

In the following we explain the steps of our proposed approach, see Fig. 1 for a schema of the proposed method:

- Step 1: Pre-processing. The image is normalized by contrast-limited adaptive histogram equalization, and then the image is resized such as lower dimension is at least 50 pixels.
- Step 2: Global descriptors. A global feature extraction is performed, from the image four equal regions without overlap and a central region of size 50% of the original image are extracted. A given descriptor is extracted separately from each of the five regions.

---

1 Textons refer to fundamental micro-structures in generic natural images and the basic elements in early (pre-attentive) visual perception. In practice, the study of textons reduces information redundancy and thus leads to better image representations.


5 Using the function adapthisteq.m in MATLAB.
and used to train five SVMs (one for each region) combined by sum rule.

- **Step 3: Sub-windows.** Each image is divided into overlapping sub-windows whose size is (l/12.5, h/12.5) where l x h is the size of the original image, taken at fixed steps of min(l/25, h/25).
- **Step 4: Local descriptors.** A local feature extraction is performed by evaluating the different texture descriptors from each sub-window.
- **Step 5: Dimensionality reduction by PCA:** Each local descriptor is transformed in a reduced space according to PCA (calculated as in TRAINING2).
- **Step 6: Codebook assignment.** Each descriptor is assigned to one codebook (created as in TRAINING3) according to the minimum distance criterion.
- **Step 7: Classification.** Each global and local descriptor extracted from the image is classified by a RS of SVM (trained as in TRAINING1).
- **Step 8: Fusion.** The classifiers results are then combined using the sum rule, i.e. selecting as final score the sum of the scores of the pool of the classifiers that belong to the ensemble. Notice that before fusion, the scores of each classifier are normalized to a mean of 0 and standard deviation of 1.

**TRAINING1:** RS of SVM. A different RS of SVM is trained for each local or global descriptor. SVM [34] is a widely used general purpose classifier trained to distinguish two classes by finding the equation of a hyperplane that maximally separates all the points between the two classes. SVM can deal with non-linearly separable problems, by using kernel functions to project the data points onto a higher-dimensional feature space, and with multi-class problems, by performing several “one-versus-all” (OVA) classification. OVA is a strategy where a single (usually binary) classifier is trained per class to distinguish that class from all other classes: prediction is then performed by selecting the class with the highest confidence score. According to the result reported in [20] we use RS of SVMs which have been proven to reduce the curse of dimensionality problem. The RS method [2] is a method for creating ensembles that combines classifiers (50 in this work) trained with subsets of the original training set, generated by randomly sampling subsets of features of reduced dimensionality (in this study we half the dimensionality of the dataset). We have used radial basis function kernel for global descriptors and histogram intersection kernel for bag-of-words approach.

- **TRAINING2:** PCA. A set of 250,000 sub-windows is randomly extracted from the training set (considering the different classes) and used to construct the PCA matrix (one projection matrix for each descriptor).
- **TRAINING3:** Codebook creation. A different set of textons is created for each class of the dataset, one for each subgroup of 50 images of the training set. A single texton is created by clustering a local descriptor with k-means (the number of cluster k is randomly selected between 10 and 40). For each descriptor, the final texton vocabulary (codebook) is obtained by concatenating the textons over the all classes. Moreover, since k-means is an unstable clustering approach, we run it several times (20 in this paper, each with a different PCA subspace projection) in order to obtain more codebooks. We have not checked if the different clusterizations obtained by k-means are similar, we combine all the created codebooks. Moreover, in our tests the performance obtained by the fusion of n classifiers based on n different codebooks built by n different k-means outperforms the performance of the best classifier of the ensemble.

### 4. Experimental results

The recognition of object categories and scene is one of the most challenging problems in computer vision. In our experiments

---

6 Images are clustered at groups of 50 due to computational issues.
we use the PASCAL Visual Object Classes Challenge 2006 protocol (VOC2006) [32], a 15 classes scene dataset widely used in the literature [10], the well-known Caltech-256 Object Category Dataset [36] and in a landmark dataset [35].

4.1. Object recognition on VOC2006

The VOC2006 dataset includes ten categories of images: bicyle, bus, car, cat, cow, dog, horse, motorbike, person, and sheep. According to the official protocol [32], we use Trainval images for training and the Test images for testing. Some images of the VOC2006 dataset are illustrated in Fig. 2. Even if VOC2006 dataset is quite old and easier with respect to the most recent VOC competition datasets (VOC2006 contains only 10 classes of object) [33], it is widely used in the literature to assess the performance of object recognition systems.

The classification task in the VOC2006 contest was judged by the Receiver Operating Characteristic (ROC) curve. The official performance indicator [32] for VOC2006 is the area under the Receiver Operating Characteristic curve. The Receiver Operating Characteristic (ROC) curve is a graphical plot of the sensitivity of a binary classifier vs. false positives (1-specificity), as its discrimination threshold is varied; the area under the ROC curve (AUC) [38] is a scalar measure that can be interpreted as the probability that the classifier will assign a lower score to a randomly picked positive samples than to a randomly picked negative samples. Since we deal with a multiclass dataset, a different AUC is calculated for each category (one-versus-all).

The first test is aimed at comparing some variants of the steps of the proposed approach. In particular we evaluate the performance obtained by considering a single texture descriptor: LPQ with \(R = 3\). The performance of the following approaches is reported in Table 1:

- **LocalSa**: this is a version of our system based only on a local descriptor (i.e. LPQ with \(R = 3\)) used to train a stand-alone SVM (only steps 1, 3, 5, 6 of the whole approach are performed, for a single descriptor); the codebook creation is performed just one time as in [9].
- **LocalRs**: the system above (LocalSa) where RS of SVM is employed as classifier. This is a simplified (single descriptor) version of our previous approach proposed in [9];
- **LocalCbSa**: the system LocalSa enhanced by adding a multiple codebook creation (as described in TRAINING3);
- **LocalPatchSa**: the system LocalCbSa enhanced with the step 2 (dimensionality reduction by PCA);
- **Local**: this is a complete version of our system based only on a local descriptor (steps 1, 3, 4, 5, 6 are performed for a single descriptor).

The results reported in Table 1 clearly show that the idea applied in the new approach (Local) drastically improves our previous method (LocalRs) [9].

The second test is aimed at comparing each single descriptor fused in our system and our final system (Final) as described in Section 3 to the state-of-the-art approaches proposed in the literature: a method using SIFT descriptors [17] (SIFT), the best results reported in [17] using the CS-LBP approach (CS-LBP), our previous method based on three descriptors [9] (Old), a method [21] based on random sampling of regions (RanSam), the winner of the VOC2006 competition (WinVOC), the method with highest performance that we have found in the literature [12], which is based on a metric learning (MetLearn). The first lines of Table 2 are referred to the following descriptors (see Section 2) each applied to the whole image (subscription G) or as a Local feature (subscription L): LTP-U, LTP-R, LPQ(3), LPQ(5), GIST, HOG, SIFT.

We want to stress the usefulness to extract the global descriptors from the regions of the image instead from the whole image: if we extract the features from the whole image LPQ(3) obtains an AUC of 91.27 and GIST of 92.99; while using our approach LPQ(3) obtains an AUC of 92.36 and GIST of 94.35.
### Table 1
Performance obtained by some variants of the proposed system.

<table>
<thead>
<tr>
<th>Approach</th>
<th>BUS</th>
<th>Car</th>
<th>Bicycle</th>
<th>Cat</th>
<th>Cow</th>
<th>Dog</th>
<th>Horse</th>
<th>Motorbike</th>
<th>Person</th>
<th>Sheep</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>LocalSa</td>
<td>90.9</td>
<td>95</td>
<td>86.9</td>
<td>83.5</td>
<td>83.7</td>
<td>78.6</td>
<td>81.1</td>
<td>87.6</td>
<td>86.9</td>
<td>90.1</td>
<td>86.43</td>
</tr>
<tr>
<td>LocalRs</td>
<td>91.5</td>
<td>95.7</td>
<td>89.8</td>
<td>86.7</td>
<td>87.4</td>
<td>81.1</td>
<td>84.3</td>
<td>89.6</td>
<td>89.1</td>
<td>91.5</td>
<td>88.67</td>
</tr>
<tr>
<td>LocalProcSa</td>
<td>97.7</td>
<td>96.7</td>
<td>94.5</td>
<td>91.1</td>
<td>89.6</td>
<td>82.7</td>
<td>87</td>
<td>92.8</td>
<td>91.7</td>
<td>93.6</td>
<td>91.7</td>
</tr>
<tr>
<td>LocalProcG5Sa</td>
<td>97.7</td>
<td>96.8</td>
<td>94.4</td>
<td>90.9</td>
<td>89.8</td>
<td>82.9</td>
<td>87.2</td>
<td>93.2</td>
<td>91.7</td>
<td>93.4</td>
<td>91.8</td>
</tr>
<tr>
<td>Local</td>
<td>98.2</td>
<td>96.9</td>
<td>94.6</td>
<td>91.2</td>
<td>89.8</td>
<td>82.9</td>
<td>87.3</td>
<td>93.3</td>
<td>91.9</td>
<td>93.6</td>
<td>92</td>
</tr>
</tbody>
</table>

### Table 2
AUC obtained by the state-of-the-art methods in the VOC2006 dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LocalSa</td>
<td>90.9</td>
</tr>
<tr>
<td>LocalRs</td>
<td>91.5</td>
</tr>
<tr>
<td>LocalProcSa</td>
<td>97.7</td>
</tr>
<tr>
<td>LocalProcG5Sa</td>
<td>97.7</td>
</tr>
<tr>
<td>Local</td>
<td>98.2</td>
</tr>
<tr>
<td>LTP-UL</td>
<td>91.6</td>
</tr>
<tr>
<td>LTP-Rc</td>
<td>90.5</td>
</tr>
<tr>
<td>LTP-Rl</td>
<td>91.5</td>
</tr>
<tr>
<td>LPQ(3)C</td>
<td>92.3</td>
</tr>
<tr>
<td>LPQ(3)L</td>
<td>92.2</td>
</tr>
<tr>
<td>LPQ(5)L</td>
<td>91.5</td>
</tr>
<tr>
<td>LPQ(5)h</td>
<td>90.1</td>
</tr>
<tr>
<td>GISTc</td>
<td>94.3</td>
</tr>
<tr>
<td>GISTl</td>
<td>93.2</td>
</tr>
<tr>
<td>HOGc</td>
<td>91.5</td>
</tr>
<tr>
<td>HOGl</td>
<td>90.7</td>
</tr>
<tr>
<td>Final</td>
<td>96.4</td>
</tr>
<tr>
<td>SIFT [17]</td>
<td>89.4</td>
</tr>
<tr>
<td>CS-LBP [17]</td>
<td>91.4</td>
</tr>
<tr>
<td>RanSum [21]</td>
<td>90.8</td>
</tr>
<tr>
<td>WinVOC [32]</td>
<td>93.6</td>
</tr>
<tr>
<td>Old [9]</td>
<td>94.2</td>
</tr>
<tr>
<td>MetLearn [12]</td>
<td>95.6</td>
</tr>
</tbody>
</table>

### 4.2. Object recognition on Caltech-256

The Caltech-256 dataset includes a challenging set of 256 object categories containing a total of 30,607 images with at least 80 images for each category. Caltech-256 is collected by choosing a set of object categories, downloading examples from Google Images and then manually screening out all images that did not fit the category. According to a suggested protocol [36], we run 5 splits test using $N_{train} = 40$ images per class for training and $N_{test} = 25$ for testing. The performance indicator is the accuracy, which is averaged on the 5 experiments. Some images of the Caltech-256 dataset are illustrated in Fig. 3.

The test reported in Table 3 is aimed at comparing each single descriptor fused in our system and our final system (Final) to the state-of-the-art approaches proposed in the literature: also in this test we show that our fusion permits to obtain performance similar to the state-of-the-art. Moreover, notice that [41–43] use 45 images in the training class for each class, so their training sets are larger than those used in this work. While in [45] several approaches are compared (see Fig. 2 of that paper) and only LPbeta (i.e. [40]) among the approaches based on 40 training images obtains performance higher than 40%.

In this test, since a large training set is available, we slightly modify the algorithm (without change the parameters): we do not consider LTP-Rl and LTP-UL due to their low performance. Moreover we propose another fusion named Final2 where the descriptors (always without using LTP-Rl and LTP-UL) are combined by a weighted sum rule, the global descriptors have a weight equal to one, while local descriptors have a weight of 0.5.

Notice that if in Final2 we use the whole image for the global descriptors and only one codebook for the local descriptors it obtains an accuracy of 33.9%. Moreover if we combine all the local descriptors (considering all the codebooks) the gained accuracy is 27.2%, while if only one codebook is used for each descriptor their fusion obtains an accuracy of 25.0%. These results confirm our idea to combine several simple variants of the same descriptor for improving the performance.

### 4.3. Scene recognition

The scene dataset [10] is composed by the following categories: coast (360 images), forest (328 images), mountain (274 images), open country (410 images), highway (260 images), inside city (308 images), tall building (356 images), street (292 images), bedroom (216 images), kitchen (210 images), living room (289 images), office (215 images), suburb (241 images), industrial (311 images), and store (315 images). Images are about 300 × 250 in resolution. Some sample images are shown in Fig. 4.

The testing protocol for the scene dataset, according to other papers in the literature, requires 5 experiments, each using 100 randomly selected images per category for training and the remaining images for testing. The performance indicator is the accuracy, which is averaged on the 5 experiments.

In Table 4 we report the results obtained by each single descriptor fused in our system, our final system compared to the best method reported in the literature. Even if the proposed approach gains lower performance than the one in [4], the reported results are very valuable, since obtained without any ad hoc optimization. Notice that some state-of-art stand-alone approaches obtain performance similar to our stand-alone best method, e.g. SIFT based method tested in [4] (the best stand-alone method among those that build their ensemble) obtains an accuracy of 81.2%, while the salient coding approach [39] which is a more performing variant of LLC (the winner of VOC2009) obtains 82.6%.

Also in this dataset is very useful to extract the global descriptors from the regions of the image instead from the whole image: if we extract the features from the whole image LPQ(3) obtains an accuracy of 57.35% and GIST of 76.08%; while using our approach LPQ(3) obtains an accuracy of 61.81% and GIST of 75.04%.

---

Fig. 3. Samples from the Caltech-256 dataset.

Fig. 4. Samples from the scene dataset.

Table 4
Accuracy obtained by the state-of-the-art approaches in the scene dataset.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Year</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTP-U&lt;sub&gt;C&lt;/sub&gt;</td>
<td>-</td>
<td>81.2%</td>
</tr>
<tr>
<td>LTP-U&lt;sub&gt;L&lt;/sub&gt;</td>
<td>-</td>
<td>71.2%</td>
</tr>
<tr>
<td>LTP-R&lt;sub&gt;C&lt;/sub&gt;</td>
<td>-</td>
<td>73.5%</td>
</tr>
<tr>
<td>LTP-R&lt;sub&gt;L&lt;/sub&gt;</td>
<td>-</td>
<td>65.6%</td>
</tr>
<tr>
<td>LPQ(3)&lt;sub&gt;C&lt;/sub&gt;</td>
<td>-</td>
<td>76.1%</td>
</tr>
<tr>
<td>LPQ(3)&lt;sub&gt;L&lt;/sub&gt;</td>
<td>-</td>
<td>74.1%</td>
</tr>
<tr>
<td>LPQ(5)&lt;sub&gt;C&lt;/sub&gt;</td>
<td>-</td>
<td>74.7%</td>
</tr>
<tr>
<td>LPQ(5)&lt;sub&gt;L&lt;/sub&gt;</td>
<td>-</td>
<td>74.1%</td>
</tr>
<tr>
<td>GIST&lt;sub&gt;C&lt;/sub&gt;</td>
<td>-</td>
<td>75.0%</td>
</tr>
<tr>
<td>GIST&lt;sub&gt;L&lt;/sub&gt;</td>
<td>-</td>
<td>75.5%</td>
</tr>
<tr>
<td>HO&lt;sub&gt;G&lt;/sub&gt;</td>
<td>-</td>
<td>68.2%</td>
</tr>
<tr>
<td>HO&lt;sub&gt;GL&lt;/sub&gt;</td>
<td>-</td>
<td>74.5%</td>
</tr>
<tr>
<td>Final</td>
<td>2012</td>
<td>87.1%</td>
</tr>
<tr>
<td>Oliva and Torralba [10]</td>
<td>2001</td>
<td>73.3%</td>
</tr>
<tr>
<td>Lazebnik et al. [5]</td>
<td>2006</td>
<td>81.4%</td>
</tr>
<tr>
<td>Liu and Shah [6]</td>
<td>2007</td>
<td>83.3%</td>
</tr>
<tr>
<td>Wu and Rehg [11]</td>
<td>2009</td>
<td>83.1%</td>
</tr>
<tr>
<td>Xiao et al. [4]</td>
<td>2010</td>
<td>88.1%</td>
</tr>
<tr>
<td>Gu et al. [1]</td>
<td>2011</td>
<td>83.7%</td>
</tr>
<tr>
<td>Huang et al. [39]</td>
<td>2011</td>
<td>82.6%</td>
</tr>
<tr>
<td>Meng et al. [8]</td>
<td>2012</td>
<td>84.1%</td>
</tr>
<tr>
<td>Nanni et al. [9]</td>
<td>2012</td>
<td>82.0%</td>
</tr>
<tr>
<td>Elfiky et al. [37]</td>
<td>2012</td>
<td>85.4%</td>
</tr>
</tbody>
</table>

4.4. Landmark recognition

The landmark dataset [35] is composed of 1227 photos of landmarks located in Pisa and crawled from Flickrs. The datasets contains photos from 12 classes having at least 46 images per class. Some sample images are shown in Fig. 5.

The landmark dataset, according to the official testing protocol [35] is divided in a training set consisting of 921 photos (approximately 80% of the dataset) and a test set consisting of 226 (approximately 20% of the dataset). The performance indicator is the accuracy.

Table 5 reports the performance obtained by each single descriptor fused in our system, our final system compared to the best method reported in [35]. In this dataset it is clear that that also global descriptors works very well, notice that several our approaches outperforms SIFT and our proposed fusion drastically outperforms methods reported in [35].

Here LPQ(5)<sub>C</sub> and GIST<sub>C</sub> obtains an accuracy of 94.7% and 88.0%, while if we extract the descriptors from the whole image we obtain an accuracy of 88.5% and 83.6%.

4.5. Comments

The computer vision is one the computer science field with higher performance improvement in the last years, e.g. the winner
of VOC2006 obtained an AUC of 93.6% in 2006, while in 2012 the highest reported performance is 95.6% [12] in this difficult dataset; anyway these results are still far from those obtained by a human being. The same conclusion is obtained in the several datasets:

- the accuracy in the scene dataset used in this paper was 73.3% in 2001, while the highest reported error accuracy in 2012 is 88.1% [4];
- the accuracy in the Caltech–256 dataset (N_train = 40) was ~38% in 2006, while the highest reported error accuracy in 2012 is 48.9% [40].

Our results are very valuable, since they have been obtained without any ad hoc parameter tuning for each dataset, while other approaches at the state-of-the-art which reach comparable performance are designed and tuned specifically for a single problem: object recognition or scene recognition. From the analysis of our experiments it is interesting to note that each descriptor works differently in each dataset, e.g. in the scene/landmark datasets GIST works poorly while in the object classification datasets it works very well. This is a reason why the fusion works well in all the tested datasets.

We investigated, using the average Q-statistic [44], the relationship among the scores of the different classifiers (each for each texton vocabulary) using the same texture descriptor. Yule’s Q-statistic is a measure for evaluating the independence of classifiers: for two classifiers D_i and D_k the Q-statistic is defined as:

\[ Q_{i,k} = \frac{ad - bc}{ad + bc} \]

where a is the probability of both classifiers being correct, d is the probability of both classifiers being incorrect, b is the probability first classifier is correct and second is incorrect, c is the probability second classifier is correct and first is incorrect. Q_{i,k} varies between −1 and 1 and it is 0 for statistically independent classifiers.

Table 5

<table>
<thead>
<tr>
<th>Approach</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT [35]</td>
<td>92%</td>
</tr>
<tr>
<td>Color-SIFT [35]</td>
<td>82%</td>
</tr>
<tr>
<td>SURF [35]</td>
<td>90%</td>
</tr>
</tbody>
</table>

The average Q-statistic (among all the couple of classifiers) in the scene dataset using LPQ(3) is 0.86. This value is enough low, for confirming that the different classifiers bring different information.

5. Conclusions

In this paper we have presented a new method for recognizing object categories and scene. We have used a random sampling instead of a complex region detector, we have used ideas from the PCA-SIFT to deal with the high dimensionality problem and we have combined different codebooks obtained by different clustering runs to enrich the power of codebook representations. Our final approach combines global and local features (obtained from the different codebooks and texture descriptors). Instead of a stand-alone classifier we have used a random subspace ensemble of SVM, which has the ability to reduce the curse of dimensionality problem.

Without an ad-hoc optimization per dataset, our approach obtains a very high performance in four different datasets using always the same parameters.

As a future work we plan to test some non-trained approaches for dimensionality reduction instead of the PCA in our final method, according to some ideas proposed in [37], where a histogram bin selection is performed.

Acknowledgements

The authors would like to thank all the other researchers that have shared their MATLAB code.

References