Open issues on codebook generation in image classification tasks

Luca Piras and Giorgio Giacinto

Department of Electrical and Electronic Engineering University of Cagliari
Piazza D'armi, 09123 Cagliari (Italy)
{luca.piras, giacinto}@diee.unica.it

Abstract. In the last years the use of the so-called bag-of-features approach, often referred to also as the codebook approach, has extensively gained large popularity among researchers in the image classification field, as it exhibited high levels of performance. A large variety of image classification, scene recognition, and more in general computer vision problems have been addressed according to this paradigm in the recent literature. Despite the fact that some papers questioned the real effectiveness of the paradigm, most of the works in the literature follows the same approach for codebook creation, making it a standard *"de facto"*, without any critical investigation on the suitability of the employed procedure to the problem at hand. The most widespread structure for codebook creation is made up of four steps: dense sampling image patch detection; use of SIFT as patch descriptors; use of the k-means algorithms for clustering patch descriptors in order to select a small number of representative descriptors; use of the SVM classifier, where images are described by a codebook whose vocabulary is made up of the selected representative descriptors. In this paper, we will focus on a critical review of the third step of this process, to see if the clustering step is really useful to produce effective codebooks for image classification tasks. Reported results clearly show that a codebook created according to a purely random extraction of the patch descriptors from the set of descriptors extracted from the images in a dataset, is able to improve classification performances with respect to the performances attained with codebooks created by the clustering process.

Keywords: Bag of Word, Visual codebook, Descriptor sampling

1 Introduction

The difficulty in capturing the complex semantics of images prompted the researchers in the fields of computer vision, image retrieval, and image classification to propose new and more effective low-level representation of images (e.g., color features, texture, shapes, local descriptors, etc.) over the years. It is easy to understand that no single representation by itself is capable of capturing all the semantics in the best possible way. In some cases, for example, one can be interested in finding several images of the same category (e.g., car), while, in other cases, one may be interested in searching an image archive to find all images of the same object [35].

In the field of visual concept detection, many global features have been proposed to describe image contents such as color [7, 14], and texture [8] histograms. Indeed, global features are not useful to distinguish different parts of the images, e.g., the foreground from the background, or near-duplicate images. A trade-off between global and local features is the use of the so-called bag-of-features approach, that has extensively exhibited high levels of performance [13, 43].

This approach, originally developed in the field of text classification [17], is based on the idea that images that contain the same or similar objects share specific areas of the image. These areas, identified as patches or interest points, are described through the use of low-level descriptors such as scale invariant features (e.g., SIFT [27], or SURF [2]), or normalized pixel values. These patches form the so-called vocabulary of *"visual words"*, i.e., the basic building blocks that allow describing the images in the dataset. The number of descriptors extracted from each image can vary from a few tens to several hundreds, and this number clearly depends on the image itself, and on the way in which the patches are sampled within the figure, i.e., densely, by using multi-scale resolution, or by focusing on particular points of interest. As the total number of visual words extracted form images of an archive can be quite large, to create a descriptor of uniform length it is necessary to select a subset of the descriptors that will be used to form the 'vocabulary' or 'codebook' [18] used to represent all images.

Codebooks are usually constructed by clustering local descriptors from a set of training images, and selecting the centroids of the clusters as representative of the clusters. An image is thus represented by a vector where the i-th component can represent either the number of local descriptors that falls in the i-th cluster ("hard" assignment) or how close are the local descriptors to the different cluster centroids ("soft" assignment). As the number of points can be quite large, clustering is usually performed by either sampling densely or sparsely the set of local descriptors extracted from the set of training images.

Images are then assigned to one of the data classes by a supervised classification step that is usually called *post-supervised* as the supervision is applied only after the unsupervised codebook creation. It is also possible that the information of the image class is used to construct the codebook, i.e. by selecting different vocabularies for different classes, and then creating the codebook by stacking all the visual vocabolaries. In this case the classification process is called *pre-supervised*. While the clustering step is the most widely used approach to create codebooks, its effectiveness in producing a discriminant representation is not clearly proven [47].

As it can be clearly seen by reading the papers published in recent years in this field, despite the approach for creating codebooks is composed of several parts (i.e., detection of interest points, extraction of effective descriptors, vocabulary coding, classification), and each part can be implemented in several different ways, the most widespread structure is the following [39]:

- dense sampling image patch detection

- use of SIFT as patch descriptors
- use of k-means as clustering technique
- use of SVM as classifier

In this article we will focus primarily on the third step of this process, by analysing the reasons why this approach is so widely used, and whether this choice is supported by either theoretical motivations or by experimental results. In fact, despite the popularity of the k-means algorithm for generating the codebook in the vast majority of research papers published in the last years in the fields of computer vision and image classification [39], few authors have focused on the reasons *why* one approach works better than another, and how the representativeness of a method does not necessarily correspond to an effective ability to discriminate. In this article, we are going to discuss if the use of k-means is able to represent and encode the visual words in an effective way, or if a codebook whose visual words are selected at random is still able to provide good classification performances.

This paper is organized as follows. Section 2 briefly reviews how the codebook creation process has been addressed in the literature. In particular, in Section 2.3 we deeply analyze the third step of this process, i.e., the "clustering phase", while in Section 2.4 we discuss more in details the usefulness of this approach. In Section 3 we argue that random selection of visual words can be as effective as the use of clustering. Experimental results are reported in Section 4, and show that random selection of visual words is actually veru effective, providing better performances than those attained by the usual technique based on clustering. Conclusions are drawn in Section 5.

2 Codebook creation

2.1 Image patch detection

The easiest approach that allows extracting a large number of patches from an image is based on exhaustively sampling different sub-parts of the image at each location and scale [12]. Even if this method has proven to be very effective [18], it suffers from severe drawbacks that lies in its high computational cost. In order to overcome this problem, a fixed grid of image patches has been proposed in [23] in the context of scene classification, where sampling is performed at each n-th pixel, and at fixed multiple scales. Instead of focussing on a more or less dense exhaustive pixels sampling, other approaches focus on the search of a set of a few but informative interest points. In [46] the authors propose a distinction of corner detectors, blob detectors, and region detectors. Among all, the Harris Laplace corner detector [30], and the Hessian-Laplace blob detector [31] have proved successful in many applications. These approaches to detect interesting patches in an image, despite their success, are still under investigation in order to better understand how the classification results are related to the type of application in which they are used [22, 18].

2.2 Patch Descriptors Extraction

Intensity-based descriptors have been widely used so far to extract distinctive invariant features from interest points. The Scale Invariant Feature Transform (SIFT) descriptor proposed in [26] is one of the most used approaches. SIFT describes the patches of an image by edge orientation histograms. In order to try to cope with the main disadvantage of this method, i.e., the 'curse of dimensionality' because each point is described by a 128 dimensional vector, in [19] the dimensionality of the feature space was reduced from 128 to 36 by principal components analysis (PCA), even if in [32] it has been proven that this descriptor is less discriminative than SIFT. These descriptors are not invariant to changes in light color, because the intensity channel is a combination of the R, G and B channels. In order to add color invariance, and increase its discriminative power, several color descriptors have been proposed [41]. More recently, [2] proposed the "Speeded-up Robust Features (SURF)" that are based on Hessian matrix, by using a very basic approximation called 'Fast-Hessian' detector.

2.3 Codebook Generation by Clustering Patch Descriptors

Clustering is the third step in the codebook generation process and is usually performed by taking into account the interest point/patch descriptors of all the images in the training set. The clustering process aims at detecting the so-called *vocabulary* or *codebook*, i.e., the basic low-level building blocks that allows detecting the concepts of interests. The underlying assumption is that each concept can be detected by representing the images in terms of the number of interest points/patches falling in each cluster that represent the *visual words* of the codebook. It is easy to see that the clustering process depends on the way interest points/patches are detected, and on the employed descriptor.

Usually, from a dataset of a few thousands of images, it is possible to extract from a few millions to several dozens of million of interest points depending on the employed detector. To reduce the computational cost of the clustering process, the set of interest points is usually under-sampled up to a few hundreds of thousand of elements. However, this means that, on average, one is actually considering less than ten points for each image in datasets containing some dozens of thousands images.

In [18] the authors argue that the key-points extraction, in spite of a more compact coding and lower computational cost, has not been designed to select the most informative regions for classification, and dense sampling gives significantly better results. In addition, they claim that standard unsupervised clustering algorithm such as k-means works well for texture analysis on images containing only a few homogeneous regions. On the other hand, they are suboptimal for recognition tasks where dense patches from natural scenes are extracted, because they create suboptimal codebooks as most of the cluster centres fall near high density regions, thus under-representing equally discriminant low-to-medium density regions. In that paper the authors proposed a strategy that combines on-line clustering [29] and mean-shift [10]. The algorithm produces an ordered list of centres, and the under-sampled descriptor vectors [15] are assigned to the first centre in the list that lies within a fixed radius r of them, or otherwise it is left unlabelled if there is no such a centre.

In [33] the authors propose to use an ensemble of randomly created clustering trees (Extremely Randomized Clustering Forests) instead of the k-means algorithm to quantize the large numbers of high-dimensional image descriptors. The reported experimental results show that the approach proves to outperform k-means based coding in terms of the training time, the memory occupancy, and the classification accuracy.

In [47], the authors experimentally compare techniques for selecting histogram codebooks for the purpose of image classification. They study some unsupervised clustering algorithms (k-means [4], Linde-Buzo-Gray (LBG) algorithm [24], Self-Organising Map (SOM) [20], and Tree-Structured SOM (TS-SOM) [21]) in a task of histogram codebook generation when post-supervised classification is performed. For comparison, they include in the experimental results those obtained by using a random selection of descriptors as codebook. They also consider several methods for supervised codebook generation that exploit the knowledge of the image classes to be detected. According to the reported experimental results, the authors conclude that at least in the dataset used in the experiments, the k-means and the LBG algorithms produced the best performing codebooks, but they also noticed that, even if the other two algorithms generate good codebooks, their classification performance is worse than that attained by randomly selected codebooks. It is their opinion that this fact demonstrates the non-existence of a direct link between the quality of clustering and the quality of the resulting codebooks.

In the past years, some authors have explored other paradigms than clustering, such as [49] where the author proposed a histogram intersection kernel technique to form the vocabulary. More recently, a dictionary learning method based on manifolds identification has been presented in [25].

In [28] and [3], the authors go as far as to completely eliminate the step of codebook creation using directly the set of descriptors using a random forest of trees and the Naive Bayes Nearest Neighbour (NBNN) [5] approach, respectively. Also in [16], the authors follow the idea that codebooks generated by clustering the descriptors are not sufficiently flexible to model heterogeneous datasets, so they propose an image representation based on a multivariate Gaussian distribution estimated over the extracted local descriptors. Also, in [51], the authors do not code the images using a codebook, but they exploit a sparse coding with locality constraints approach in order to create a raw image representation [48], thus overcoming other similar approaches as NBNN [5].

2.4 "Is clustering useful for codebook definition?"

Some papers such as [34], and, afterwards, [39], and [9], review and compare several techniques to sample image patches from the images, and cluster or reduce them in some way in order to create an effective short codebook. Unfortunately, reported results do not allow drawing sound conclusions on the best technique to be employed to create codebooks, for a number of reasons. First, experimental results heavily depend on the values of the parameters used in the various steps of codebook generation that we mentioned in the Introduction. Secondly, the values of the parameters have to change depending on the dataset, and the application at hand. Indeed, by reviewing all the papers that have been published over the years on this topic, it is possible to see that they share the same basic framework as shown in Section 1 without any critical discussion on its validity. It seems like researchers have "surrendered", more or less consciously, to the idea that there is not much room for improvement in this field, and that the above algorithm should be taken for granted.

From our point of view, a number of questions still remain open: Why, despite of the number of papers providing different solutions, researchers typically rely on the same approach based on k-means clustering?

Is it possible that this behavior is motivated by the ease with which nowadays it is possible to find open-source libraries that implement the basic framework as a black box?

Is it possible that the alternative solutions proposed in the literature, while providing modest improvements in performance, exhibit a very complicated implementation that prevents their wide adoption by the scientific community?

It is also worth noting that there is also one paper that proposed a very much simpler approach whose results can be compared to those attained by much more sophisticated methods [9] but, to the best of our knowledge, this proposal has not gained much attention so far. It is our opinion that the use of a codebook allows attaining an effective representation of images, but the question is: how to make this representation discriminative enough to justify the effort to implement it?

3 Random selection of interest points for codebook creation

The clustering process is the step that transforms the set of interest point descriptors of a set of training images into "bags of visual words" and codebooks. The process of creating a codebook in this way is quite straightforward:

- m interest points are extracted from all the training images (for details see Section 2.1);
- points are clustered by using an unsupervised clustering algorithms (see Section 2.3). Typically, k-means clustering is used, where $k << m$;
- each image is mapped into a codebook of size k by assigning the interest points of the image to the nearest centroid;
- the new feature vector representation is used for image classification.

To say the truth the use of the approach as is, it is not feasible; in fact to ensure that the clustering process is effective some *tricky* settings are needed. First of all it is necessary to estimate the “optimal” number of clusters (i.e., the codebook’s dimension), and then the number of interest points to be clustered.

Let’s examine the issue of the number of interest points to be clustered. From a dataset of a few thousands of images, it is possible to extract up to several dozens of million of interest points depending on the technique that is used. Thus, the process of extracting the subset of points that will form the vocabulary is computationally quite expensive for a clustering task, so the set of interest points is usually under-sampled up to a few hundreds of thousand of elements (Step 2, Figure 1(a)). Clustering is then applied to this reduced set of points. This means that, on average, we are actually considering less than ten points for each image.

A common procedure to set the total number of interest points to be clustered is based on a random extraction of a subset of points of the totality of points belonging to all images in the training set. This number is usually fixed regardless of the number of images in the dataset in order to limit the computational complexity of the clustering algorithm. Consequently, the number of points per image automatically decreases as long as the number of images increases. For example, the *Color Descriptors* toolkit [41] (i.e., one of the most popular toolkits available to generate codebooks) employs the k-means clustering algorithm to produce a “bag of visual words” representation by randomly selecting 250,000 points from the set of points of the images of the input dataset, regardless of the dimension of the dataset. Even if this approach can be considered a reasonable choice with respect to the representativeness of the interest points, it is not possible to say the same with regard to its ability to discriminate.

Thus, it is hard to claim that the output of the clustering process actually summarize the most discriminative point of the image archive. In order to overcome this limitation, in [38] a larger number of interest points are taken into account by resorting to the ensemble paradigm. Different codebooks are created by extracting different subsets by random or pseudo-random under-sampling techniques, then classification results obtained by using each different codebook are combined. In fact, it has been shown that the codebook generated by considering different subsets of interest points provide diverse classification results.

Thus, the first issue we want to point out is that the random extraction of a subset of points for further clustering heavily affect the classification performances. Depending on the number of points extracted, and on the way points are randomly sampled from each image, the resulting codebook may exhibit a diverse degree of discriminative power from a classification point of view.

The choice of the dimension of the codebook is the second parameter that affects the discriminative power in a classification task. Usually, the dimension is chosen so that the computational cost of the clustering phase is constrained within specified limits. We recall that the clustering phase is the most time consuming operation in the pipeline of codebook generation. According to the specifications of the *Color Descriptors* toolkit [41] the *“clustering performed on 250,000 descriptors on 384 (ColorSIFT) dimensions take at least 12 hours per iteration of k-means”*. In addition, the number of clusters strictly depends on the dataset at hand (e.g., number of classes, variability in each class, etc.) so that it is not usually possible to set this parameter in advance. Thus, usually different codebook sizes are tested in order to find the most suitable solution. It is worth recalling that this result is also affected by the number of points we are considering in the clustering process, and thus different codebook sizes my provide similar performances as long as they are computed from a different subset of the set made up of all the interest points extracted from the training images. Finally, let us also recall that the output of the k-means clustering algorithm may depend on the choice of the initial cluster centroids (Steps 3 and 4, Figure 1(a)).

Summing up, the performances of image classification based on the bag-of-visual-words paradigms depends on two random processes

- the initial sampling of interest points that are used in the clustering process
- the initialization of the cluster centroids in the k-means algorithm

Motivated by the above considerations, in this paper we *suggest* to skip the clustering step at all, and generate the codebook by direct random sampling a fixed number of descriptors from the whole set of descriptors, and use them as visual words of the vocabulary to build the codebook (Step 2, Figure 1(b)). Our proposal is based on the idea that a random sub-sampling of the interest points of the images followed by a random initialization of the cluster centroids produces a randomness that can in some way be imitated by an purely random extraction of the codebook's "words" from the dozens of million of interest points of all images in the dataset.

1. m interest points are detected from all the training images;	1. m interest points are detected from all the training images;
2. a subset of n (where $n << m$) interest points is randomly extracted from the previous set;	
3. k points (where $k << n$) are randomly selected as initial cluster centroids;	2. a subset of k (where $k << m$) interest points is randomly extracted from the previous set;
4. k-means algorithms is used to cluster the n points in k clusters;	
5. each image is mapped into a codebook of size k by assigning the interest points of the image to the nearest k^{st} centroid;	3. each image is mapped into a codebook of size k by assigning the interest points of the image to the nearest k^{st} point;
6. the new feature vector representation is used for image classification.	4. the new feature vector representation is used for image classification.
(a) k-means approach	(b) Random approach

Fig. 1. k-means and random approach schemes

Reported results show how a codebook created in this way is able to generate feature histograms able to attain some improvements in a classification task with respect to the performance obtained with a codebook created by the clustering process. In other words, we will show that by completely neglecting the clustering phase we can also attain better performances, and greatly reduce the time required to build the vocabulary.

4 Experimental Results

Experiments have been carried out using two datasets, namely a subset of the MIRFLICKR[1] collection proposed for the ImageCLEF 2012 Photo Flickr Annotation Task [44] and the MICC-Flickr101 dataset [1]. The first dataset comprises 25 thousand multi-labelled images that have been manually annotated using 94 concepts. The MICC-Flickr101 dataset is based on the 101 object categories of the Caltech101 dataset, while the images were obtained by downloading them from Flickr in January 2012. This dataset is made up of 7348 single labelled images with at least about 40 images per class, the median of the number of elements per class being equal to 70. Images are at high resolution, 1024 x 768 pixels on average, and depict objects in daily-life real scenarios. The ImageCLEF 2012 dataset was originally subdivided into a training and a test set, while we randomly subdivided the MICC-Flickr101 dataset.

A dense sampling strategy for interest point detection has been used: features are extracted at 4 scales (0.5, 1, 1.5, and 2) with a regular grid spaced 10 pixels. For extracting the SIFT descriptors, the ISIS Color Descriptors[2] toolkit has been used [41].

In order to run different experiments we performed four random extractions of 512 points to create four different codebooks according to the usual approach based on k-means clustering. The choice of the dimension of the codebook has been aimed to limit the overall computational cost related to all the phases involved in obtaining the "traditional" bag-of-visual-words representation, and the later parameter estimation for each SVM that has to be trained for each representation. It is well known that all these phases are highly time consuming. We compared the proposed approach to the common procedure, i.e. randomly selecting 250,000 points from the set of points extracted from the entire training set. The clustering has been performed on 250,000 128D (SIFT) vectors using two Intel Xeon E5-2630 2.3Ghz with 64 GB of RAM and took on average 8 hours for each of the five iterations of k-means. Also, in this experiments the dimension of the codebooks, i.e. the number of clusters k in the k-means approach, has been fixed to 512. In several preliminary experiments, we also used different values for the k parameter, and the comparison between the proposed mechanism and the usual one exhibits the same behaviour as the one reported in the paper.

The Support Vector Machine with RBF kernel has been used as the base classifier for its good performance on various image classification tasks [11, 6]. In

[1] http://press.liacs.nl/mirflickr/

[2] http://koen.me/research/colordescriptors/

	k-means	Random
F1u	49.51%	50.62%
F1M	23.07%	24.88%

Table 1. Results in terms of F1 micro and macro for the ImageCLEF dataset.

	k-means	Random
F1u	28.12%	30.86%
F1M	24.15%	26.67%

Table 2. Results in terms of F1 micro and macro for the MICC-Flickr101 dataset.

particular, we implemented a multi-label classifier by independently training a SVM classifier for each class [42, 45]. SVM parameters have been set by exploring the set of parameters in order to select the detector with the highest performance in the training set. For each classifier, a different decision threshold has been set according to the threshold optimization approach proposed in [37] aimed at maximizing the overall classification performance when, for each pattern, a score value is available for each class.

4.1 Results

Performance are evaluated in terms of Macro-averaged and micro-averaged F-measures [42, 45]. In multi-label classification tasks, the F-measure [40] over all classes can be defined in terms of empirical averages in two different ways. Macro-averaging (denoted with the capital 'M') consists of averaging over all the classes the corresponding class-related measure. It equally weights each class, and thus tends to be dominated by the performance on rare classes, which is usually lower than that attained for common ones [50]. Micro-averaging (denoted with 'u') consists of computing the measure with respect to the sum of the true positive, false positive and false negative values over all classes.

The results of the experiments on the ImageCLEF and the MICC-Flickr101 datasets are reported in the following Tables. Each value in the table refers to the average performance of four concept detectors, each one trained on different codebooks created either by independently random selected points (Random) or performing *k*-means clustering on 250,000 descriptor randomly extracted from the set of all image descriptors (*k*-means).

Reported results clearly show that the different ways of selecting the visual words for the creation of the codebooks influences the performance, depending on the number of the images in the dataset. Table 1, that is related to the ImageCLEF dataset, shows a limited improvement of the proposed random approach if compared to the improvement that the random approach exhibited over the *k*-means approach on the MICC dataset (see Table 2). In the MICC dataset the average improvement reach 2.5%, while in the ImageCLEF dataset the improvement is around 1.5%. However, it is quite remarkable that by just creating a codebook at random we are able to perform better than with a codebook generated by a structured approach.

The experiments run over the two dataset also show that the same improvement trend is observed by computing the F1M and F1u measures. It is worth to note that the two measures work in a different way. While the F1u measure computes the sum of the true positive values, the false positive values, and the false negative values over all classes, and then evaluates a "global" average, the F1M measure averages the corresponding class-related measure over all the classes, with equal weights for each class. Consequently, the F1M tends to be dominated by the performance of the classes containing a tiny fraction of patterns. The same trend of improvements show how the (Random) approach works well also for the classes with a small number of patterns.

This behavior can be translated into the following observation. When the dataset contains a large number of images, and, consequently, a very large number of interest points are extracted, the (Random) procedure create a codebook that is able to outperfrom the usual approach based on clustering that suffers from the clustering of a so large number of points. On the other hand, when the number of images in the dataset is not so large, the (Random) procedure still allows extracting those points that permit to create codebooks that are able to produce larger increment of the improvement. The reader might think that as the number of images increases, then the variability of image content becomes higher, and it should be better to increase the dimension of the codebook. It is worth to note that such a direct relationship has not been assessed in the literature. However, while some works [33] claim that within certain limits this relation exists, other authors argue that this relation can hold just for some dataset and not for others [34, 9].

In order to better understand the behavior of the random procedure with respect to the usual k-means approach, let us analyze the the correlation (i.e., the diversity) of the codebooks. To this end, Figure 2 shows the degree of correlation among the scores of the 8 concept detectors, four of them trained with the codebooks obtained by randomly extracting the "visual words" of the vocabulary from the set of the descriptors of the images (**R**x), the other four trained with the codebooks obtained by following the common approach (**T**x). It is easy to see that, regardless the technique used to produce different codebooks, the output scores tend to exhibit high correlation values (higher than 0.8 for ImageCLEF and higher than 0.7 for MICC-Flickr), and this support our idea that an offhanded extraction of the "words" of the codebooks exhibits the same randomness of the common procedure where a random sub-sampling of the interest points of the images is followed by a random initialization of the cluster centroids.

5 Conclusion

By extensively reviewing the papers that have been published over the past years in the fields of image classification, scene recognition, and more in general com-

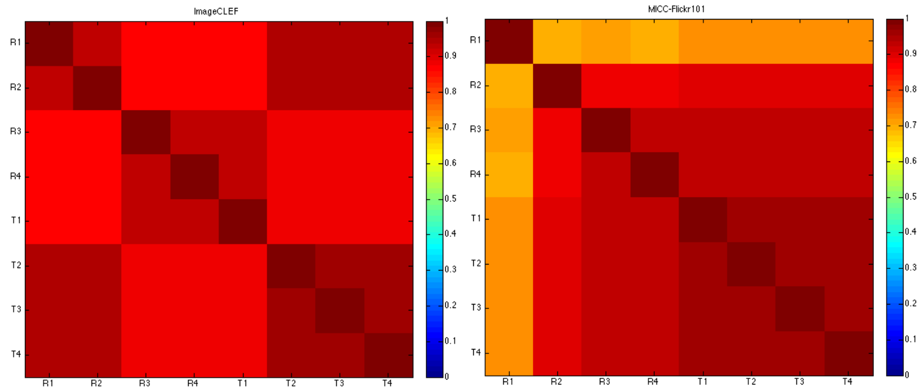

Fig. 2. Correlation matrix between the score obtained using the proposed approach (**R**x) and the score obtained using the common clustering method (**T**x) in the ImageCLEF and MICC-Flickr101 dataset

puter vision, it can be clearly seen that they share the same basic framework, i.e., the use of the bag-of-visual-words paradigm. In this framework, a very important step is the clustering phase that usually is performed using one of the most common unsupervised clustering approach, the k-means algorithm. Despite its wide use to create codebooks, its effectiveness in producing a discriminant representation is not clearly proven, whereas its drawbacks are well known. First of all, the clustering step is the most time consuming operation in the pipeline of codebook generation, and, secondly, it is hardly scalable with the number of points to be clustered. For the latter reason, often this step is not performed over all the points extracted by the training images, but a random sub-sampling step is needed to reduce the total number of interests points, thus reducing the number of points for each image. As a consequence, it is hard to claim that the output of the clustering process actually summarize the most discriminative point of the image archive, as clustering is usually performed over a random selection of a few points per image. In addition, it is also worth noting that the output of the k-means clustering algorithm depends on the choice of the initial cluster centroids. In this light, it is possible to say that the performances of image classification based on the bag-of-visual-words paradigms depends on two random processes.

For these reasons, in this paper we investigated if codebooks generated through a direct random sampling of a fixed number of descriptors from the whole set of descriptors is able to generate feature histograms able to attain similar performances with respect to the usual procedure that involves the use of the k-means clustering algorithm. Surprisingly, we found out that the generation of codebook by randomly selecting the visual words actually produced significant improvements in classification performances. Reported results showed that by completely neglecting the clustering phase it is possible to attain better performances, and greatly reduce the time required to build the vocabulary.

References

1. Ballan, L., Bertini, M., Del Bimbo, A., Serain, A.M., Serra, G., Zaccone, B.F.: Combining generative and discriminative models for classifying social images from 101 object categories. In: Proc. of International Conference on Pattern Recognition (ICPR). Tsukuba, Japan (November 2012), (Poster)
2. Bay, H., Ess, A., Tuytelaars, T., Gool, L.J.V.: Speeded-up robust features (surf). Computer Vision and Image Understanding 110(3), 346–359 (2008)
3. Becker, J.H., Tuytelaars, T., Gool, L.J.V.: Codebook-free exemplar models for object detection. In: WIAMIS. pp. 1–4. IEEE (2012)
4. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer (October 2006), http://www.worldcat.org/isbn/0387310738
5. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR. IEEE Computer Society (2008)
6. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. ACM TIST 2(3), 27 (2011)
7. Chang, S.F., Sikora, T., Puri, A.: Overview of the mpeg-7 standard. IEEE Trans. Circuits Syst. Video Techn. pp. 688–695 (2001)
8. Chatzichristofis, S.A., Boutalis, Y.S.: Fcth: Fuzzy color and texture histogram - a low level feature for accurate image retrieval. In: Proceedings of the 2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services. pp. 191–196. IEEE Computer Society (2008)
9. Chavez, A., Gustafson, D.: Building an effective visual codebook: Is k-means clustering useful? In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Fowlkes, C., Wang, S., Choi, M.H., Mantler, S., Schulze, J.P., Acevedo, D., Mueller, K., Papka, M.E. (eds.) ISVC (2). Lecture Notes in Computer Science, vol. 7432, pp. 517–525. Springer (2012)
10. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. 24(5), 603–619 (2002)
11. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press (2000)
12. Crowley, J.L., Sanderson, A.C.: Multiple resolution representation and probabilistic matching of 2-d gray-scale shape. IEEE Trans. Pattern Anal. Mach. Intell. 9(1), 113–121 (1987)
13. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: In Workshop on Statistical Learning in Computer Vision, ECCV. pp. 1–22 (2004)
14. Deselaers, T., Keysers, D., Ney, H.: Features for image retrieval: an experimental comparison. Inf. Retr. 11(2), 77–107 (2008)
15. Estabrooks, A., Jo, T., Japkowicz, N.: A multiple resampling method for learning from imbalanced data sets. Computational Intelligence 20(1), 18–36 (2004)
16. Grana, C., Serra, G., Manfredi, M., Cucchiara, R.: Image classification with multivariate gaussian descriptors. In: Petrosino [36], pp. 111–120
17. Joachims, T.: Text categorization with suport vector machines: Learning with many relevant features. In: Nedellec, C., Rouveirol, C. (eds.) ECML. Lecture Notes in Computer Science, vol. 1398, pp. 137–142. Springer (1998)
18. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. In: ICCV. pp. 604–610. IEEE Computer Society (2005)

19. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. In: CVPR (2). pp. 506–513 (2004)
20. Kohonen, T.: The self-organizing map. Neurocomputing 21(1-3), 1–6 (1998)
21. Koikkalainen, P., Oja, E.: Self-organizing hierarchical feature maps. In: Neural Networks, 1990., 1990 IJCNN International Joint Conference on. pp. 279–284 vol.2 (1990)
22. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2). pp. 2169–2178. IEEE Computer Society (2006)
23. Li, F.F., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: CVPR (2). pp. 524–531. IEEE Computer Society (2005)
24. Linde, Y., Buzo, A., Gray, R.: An algorithm for vector quantizer design. Communications, IEEE Transactions on 28(1), 84–95 (1980)
25. Liu, B.D., Wang, Y.X., Zhang, Y.J., Shen, B.: Learning dictionary on manifolds for image classification. Pattern Recognition 46(7), 1879–1890 (Jul 2013)
26. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV. pp. 1150–1157 (1999)
27. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
28. Martínez-Muñoz, G., Delgado, N.L., Mortensen, E.N., Zhang, W., Yamamuro, A., Paasch, R., Payet, N., Lytle, D.A., Shapiro, L.G., Todorovic, S., Moldenke, A., Dietterich, T.G.: Dictionary-free categorization of very similar objects via stacked evidence trees. In: CVPR. pp. 549–556. IEEE (2009)
29. Meyerson, A.: Online facility location. In: FOCS. pp. 426–431. IEEE Computer Society (2001)
30. Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. In: ICCV. pp. 525–531 (2001)
31. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. International Journal of Computer Vision 60(1), 63–86 (2004)
32. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. 27(10), 1615–1630 (2005)
33. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. In: Schölkopf, B., Platt, J.C., Hoffman, T. (eds.) NIPS. pp. 985–992. MIT Press (2006), http://eprints.pascal-network.org/archive/00002438/01/nips.pdf
34. Nowak, E., Jurie, F., Triggs, B.: Sampling strategies for bag-of-features image classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV (4). Lecture Notes in Computer Science, vol. 3954, pp. 490–503. Springer (2006)
35. Penatti, O.A.B., Silva, F.B., Valle, E., Gouet-Brunet, V., Torres, R.d.S.: Visual word spatial arrangement for image retrieval and classification. Pattern Recognition 47(2), 705–720 (Feb 2014)
36. Petrosino, A. (ed.): Image Analysis and Processing - ICIAP 2013 - 17th International Conference, Naples, Italy, September 9-13, 2013, Proceedings, Part II, Lecture Notes in Computer Science, vol. 8157. Springer (2013)
37. Pillai, I., Fumera, G., Roli, F.: Threshold optimisation for multi-label classifiers. Pattern Recognition 46(7), 2055 – 2065 (2013), http://www.sciencedirect.com/science/article/pii/S0031320313000320
38. Piras, L., Tronci, R., Giacinto, G.: Diversity in ensembles of codebooks for visual concept detection. In: Petrosino [36], pp. 399–408
39. Ramanan, A., Niranjan, M.: A review of codebook models in patch-based visual object recognition. Journal of Signal Processing Systems 68(3), 333–352 (2012)

40. van Rijsbergen, C.J.: Information Retrieval. Butterworth (1979)
41. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluating color descriptors for object and scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. 32(9), 1582–1596 (2010)
42. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. 34(1), 1–47 (2002)
43. Sivic, J., Zisserman, A.: A text retrieval approach to object matching in videos. In: ICCV. pp. 1470–1477. IEEE Computer Society (2003)
44. Thomee, B., Popescu, A.: Overview of the imageclef 2012 flickr photo annotation and retrieval task. Tech. rep., CLEF 2012 working notes, Rome, Italy (2012)
45. Tsoumakas, G., Katakis, I., Vlahavas, I.P.: Mining multi-label data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 667–685. Springer (2010)
46. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: A survey. Foundations and Trends in Computer Graphics and Vision 3(3), 177–280 (2007)
47. Viitaniemi, V., Laaksonen, J.: Experiments on selection of codebooks for local image feature histograms. In: Sebillo, M., Vitiello, G., Schaefer, G. (eds.) VISUAL. Lecture Notes in Computer Science, vol. 5188, pp. 126–137. Springer (2008)
48. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 3360–3367 (2010)
49. Wu, J., Rehg, J.M.: Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In: ICCV. pp. 630–637. IEEE (2009)
50. Yang, Y.: A study on thresholding strategies for text categorization. In: ACM (ed.) Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval,. p. 137145 (2001)
51. Zhang, C., Wang, S., Liang, C., Liu, J., Huang, Q., Li, H., Tian, Q.: Beyond bag of words: image representation in sub-semantic space. In: Jaimes, A., Sebe, N., Boujemaa, N., Gatica-Perez, D., Shamma, D.A., Worring, M., Zimmermann, R. (eds.) ACM Multimedia. pp. 497–500. ACM (2013)