Class Representative Visual Words for Category-Level Object Recognition

Roberto Javier López Sastre[1,⋆], Tinne Tuytelaars[2], and Saturnino Maldonado Bascón[1]

[1] University of Alcalá, GRAM
{robertoj.lopez,saturnino.maldonado}@uah.es
[2] K.U. Leuven, ESAT-PSI
Tinne.Tuytelaars@esat.kuleuven.be

Abstract. Recent works in object recognition often use visual words, i.e. vector quantized local descriptors extracted from the images. In this paper we present a novel method to build such a codebook with *class representative* vectors. This method, coined *Cluster Precision Maximization* (CPM), is based on a new measure of the cluster precision and on an optimization procedure that leads any clustering algorithm towards class representative visual words. We compare our procedure with other measures of cluster precision and present the integration of a Reciprocal Nearest Neighbor (RNN) clustering algorithm in the CPM method. In the experiments, on a subset of the the Caltech101 database, we analyze several vocabularies obtained with different local descriptors and different clustering algorithms, and we show that the vocabularies obtained with the CPM process perform best in a category-level object recognition system using a Support Vector Machine (SVM).

Keywords: object recognition, clustering, visual words, class representative.

1 Introduction

A popular strategy to represent images in the context of category-level object recognition is the *Bag of Words* (BoW) approach [1]. The key idea is to quantize the continuous high dimensional space of local image descriptors such as SIFT [2], to get a codebook of so-called visual words. In this philosophy, an image is considered like a text document in which it is possible to find visual *words* to describe its content. A Bag of Words is then built as a histogram over visual word occurrences. This image representation has been shown to characterize the images and the objects within in a robust yet descriptive manner, in spite of the fact that it does not capture the spatial configuration between visual words. It is supposed that if a specific set of visual words appears in an image, there will be a high likelihood of finding the object in it. These BoW systems have

[⋆] Work performed during stay at Katholieke Universiteit Leuven.

H. Araujo et al. (Eds.): IbPRIA 2009, LNCS 5524, pp. 184–191, 2009.

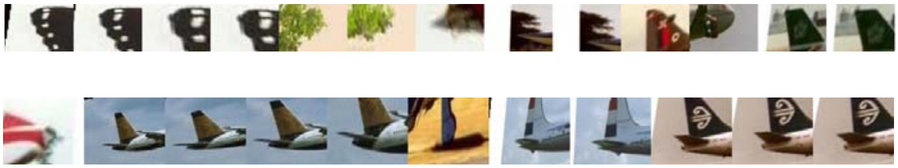


Fig. 1. First row: image patches that have been clustered together. Clearly all come from different object categories and lie on single object instances. Second row: cluster with image patches of the same object category and from different instances of the class, i.e. it contains class representative visual words.

shown impressive results lately, in spite of the simplicity of the scheme [1,3,4]. It is variations on this BoW scheme that have won the recent Pascal Visual Object Classes Challenge on object classification [5].

More precisely, building a BoW representation involves the following steps. First, interest regions are detected, either using a local invariant feature detector [6] or by densely sampling the image. Then, a region descriptor [7] is used to characterize them. After that, a clustering algorithm is run over all these descriptors to perform the vector quantization which finally results in a *visual vocabulary*. The idea is to characterize an image with the number of occurrences of each visual word, so any classifier can then be used for the categorization of this histogram representation.

Different vocabularies can be obtained for the same object class, and their quality depends on several parameters: the number of clusters (frequently fixed empirically), the nature of the descriptor, the chosen distance function to measure the similarity between descriptors, and the clustering algorithm. Our aim in this paper is to describe how to adapt the vector quantization process so as to yield class representative visual words, i.e. how to exploit the information of class labels already during the visual vocabulary construction process.

The main contribution is that we introduce an optimization procedure, the *Cluster Precision Maximization* (CPM), that maximizes the cluster representativeness. High representativeness should be assigned to visual words that generalize well over the within-class variability, yet are discriminative with respect to the object class (between-class variability). The basic idea behind our measure is that a visual word becomes more representative as it is found on a higher number of different instances of the same object class.

Fig.1 shows two examples of clusters of visual words. The first one (upper row) represents a *bad* cluster with visual words that clearly come from different object classes and lie on single object instances, not generalizing well within the class. The second example (lower row), on the other hand, shows a class representative visual word, found almost exclusively on objects of a single class and including different instances of this class. This is the type of cluster we want the CPM to deliver.

Related Work. To date, K-means clustering is still the most widely used vector quantization scheme in this context, in spite of its limitations: it does not take the class-labels into account, its output depends on the initialization, and it is computationally expensive. More efficient and stable alternatives have been proposed. However, here we will focus on work devoted to seeking more class representative visual words. First, there are several works based on frequent itemset mining [8,9,10]. Typically, finding representative visual words then boils down to finding frequent co-occurring groups of descriptors in a transaction database obtained from the training images. Others have tried to add more local geometric information to their codebook generation algorithms. Lazebnik *et al.* [11] constructed a codebook with groups of nearby regions whose appearance and spatial configuration occur repeatably in the training set. In [12] Leibe *et al.* presented how to learn semantic object parts for object categorization. They use what they call co-location and co-activation to learn a visual vocabulary that generalizes beyond the appearance of single objects, and often gets semantic object parts.

Perronnin *et al.* [13] build class representative visual words by enlarging the visual vocabularies, in spite of the increased cost of histograms computations. They propose an approach based on a universal vocabulary plus class specific vocabularies to improve the performance of the recognition system. To get more compact vocabularies Winn *et al.* [14] build an approach based on the bottleneck principle, while Moosmann *et al.* [15] organize the vocabulary using Extremely Randomized Clustering Forests. Finally, Perronnin *et al.* [16] have proposed to use Fisher Kernels, as their gradient representation has much higher dimensionality than a histogram representation, resulting in very compact vocabularies yet highly informative representations.

Closer to our approach are the works of Mikolajczyk *et al.* [17] and Stark *et al.* [18]. In [17] the performance of local detectors and descriptors is compared in the context of the object class recognition problem, and a new evaluation criterion based on the clusters precision is proposed. The problem is that following this approach, many clusters with features from only one object instance get high precision scores. Stark *et al.* [18] decided to give higher scores to feature descriptors that generalize across multiple instances of an object class, and proposed a new cluster precision definition, but their approach gets the best score when each cluster contains only one vector.

Overview. In section 2 we explain the CPM method and the integration of the Reciprocal Nearest Neighbor (RNN) clustering algorithm in it. In section 3 we present results from applying the CPM method to obtain visual vocabularies in a subset of Caltech101 database. Finally, section 4 concludes the paper.

2 Class Representative Visual Words

The main steps of our method can be sketched as follows. First the detection and description of local features for the object classes in the database is done. Then comes the tuning of the clustering algorithm parameters, computing a vocabulary with an efficient vector quantization approach for the obtained local features. The last task is to evaluate the clusters precisions (CP) for every object class, and iterate from the tuning step until the maximum for CP is reached.

There are many different techniques for detecting and describing local image regions [6,7]. Here we briefly describe the detectors and descriptors used in this work[1]. The region detector we use is the Hessian-Laplace [7] that responds to blob-like structures. It searches for local maxima of the Hessian determinant, and selects a characteristic scale via the Laplacian. For the descriptors we experiment with SIFT [2] and Shape Context (SC) [19]. The SIFT descriptor is a 3D histogram over local gradient locations and orientations, weighted by gradients magnitude. SC is based on edge information. For a given interest point, the SC descriptor accumulates the relative locations of nearby edge points in a coarse log-polar histogram.

2.1 Cluster Precision Maximization

Given a set of local features extracted from the images in our database, we could fix the number of clusters we want and apply the clustering algorithm of our choice to obtain a visual vocabulary for the object classes. One strategy to follow is to empirically find the optimal cluster parameters by testing whether this codebook reports an accurate categorization on a validation set. If it does not, we run the algorithm again with different parameters, until we get a codebook casting the lowest empirical risk in categorization. The problem of this kind of approach is that we need to complete the whole pipeline of the system just to validate the clustering.

The Cluster Precision Maximization (CPM) method on the other hand directly searches for class representative visual words, i.e. representative clusters. This new method measures the clusters precisions, so it is not needed to train a classifier and test whether it is accurate enough or not in each iteration. The basic idea behind this approach is that the more images with different object instances contain a particular visual word, the more representative the word is for that object class, and the better the categorization.

Suppose a database $\mathcal{DB}$ which contains N images of M different object classes, $\mathcal{DB} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_N\}$. For each image in the database we first extract local features f, so an image $\mathcal{I}_i$ can be represented with a set of features $\mathcal{I}_i = \{f_{i_1}, f_{i_2}, \ldots\}$. Note that this will not be a BoW representation until the vector quantization is done using the features from all images in the database. After the clustering, a codebook $\mathcal{W} = \{w_1, w_2, \ldots, w_K\}$ with K words is obtained.

Mikolajczyk *et al.* [17] evaluate the codebook W by computing the average cluster precision for all the object classes. Suppose there are K clusters in the vocabulary, but only K_a in which class a dominates. The average precision defined by Mikolajczyk *et al.* is as follows

$$P_a = \frac{1}{K_a} \sum_{j=1}^{K_a} p_{j_a} \ , \qquad (1)$$

where p_{j_a} is the number of features of class a in cluster j, divided by the total number of features in cluster j. In [18] the authors notice the previous definition

[1] The binaries have been taken from http://www.robots.ox.ac.uk/~vgg/research/affine/

of cluster precision gets high scores in those clusters that contain features from only a single object instance. They discount such clusters by summing over the fraction of objects of a class a in cluster j instead of individual features, and weight these fractions by cluster sizes, obtaining,

$$P_a = \left(\sum_{j=1}^{K_a^{'}} s_j \right)^{-1} \sum_{j=1}^{K_a} s_j p_{j_a} \ , \tag{2}$$

where j now ranges over all $K_a^{'}$ clusters in which objects of class a dominate, and s_j is the total number of features in cluster j. This new cluster precision definition gives higher scores to clusters that generalize across multiple instances of an object class, but it casts the maximum score when each cluster contains only one vector. Because neither of the two cluster precision definitions seem to meet our goal of selecting class representative visual words without artefacts, we propose a new cluster precision, this time summing over the number of objects of class a times the number of features of class a in each cluster. We get

$$P_a = K \sum_{j=1}^{K} s_{j_a} n_{j_a} \ , \tag{3}$$

where s_{j_a} is the number of features found in images of object class a in cluster j, n_{j_a} is the number of different objects of class a represented in cluster j, and K is the number of clusters. This cluster precision varies from $S_a \times N_a$ to ${S_a}^2$, where S_a is the total number of features extracted for the object class a, and N_a is the number of different object instances of class a in the database. In each iteration the CPM approach computes the average precision over all object classes, $P = \frac{1}{M} \sum_{m=1}^{M} P_m$, using the definition in equation (3) for P_m, until it gets the maximum value. This maximization leads to the most representative clusters for an object class, and consequently good results in recognition. Any clustering algorithm can be integrated in this methodology. In Algorithm 1 we present in detail how to integrate an efficient average-link agglomerative clustering algorithm based on Reciprocal Nearest Neighbors [20], which has a complexity of $O(N^2d)$ and only linear space requirements.

3 Experimental Results

For all our experiments we use a subset of the Caltech101 database [21] consisting of 20 classes[2]. The total number of images is 3901. For the classification, 50% of the images of each class are used for training and 50% for testing. The number

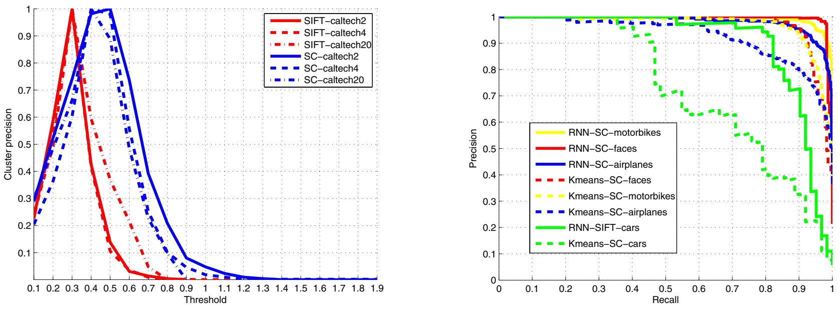
[2] The classes we use in this paper are: airplanes, bonsai, brain, butterfly, car-side, chandelier, faces-easy, grand-piano, hawksbill, ketch, laptop, leopards, menorah, motorbikes, starfish, sunflower, trilobite, umbrella, watch and yin-yang.

Algorithm 1. *CPM* for an Average-Link clustering based on RNN.

```
$CP_{max} = 0$
for $thres = min$ to $max$ do
  $C = \emptyset$; $last \leftarrow 0$; $lastsim[0] \leftarrow 0$; //C will contain a list with the clusters
  $L[last] \leftarrow v \in V$; //Start chain L with a random vector v
  $R \leftarrow V \backslash v$; //All remaining points are kept in R
  while $R \neq \emptyset$ do
    $(s, sim) \leftarrow getNearestNeighbor(L[last], R)$;
    if $sim > lastsim[last]$ then
      $last \leftarrow last + 1$; $L[last] \leftarrow s$; $R \leftarrow R \backslash \{s\}$
    else
      if $lastsim[last] > thres$ then
        $s \leftarrow agglomerate(L[last], L[last-1])$; $R \leftarrow R \cup \{s\}$; $last \leftarrow last - 2$
      else
        $C \leftarrow C \cup L$; $last \leftarrow -1$; $L = \emptyset$;
      end if
    end if
    if $last < 0$ then
      $last \leftarrow last + 1$; $L[last] \leftarrow v \in R$; $R \leftarrow R \backslash s$
    end if
  end while
  $CP \leftarrow getCP(C)$; //Evaluate CP
  if $CP > CP_{max}$ then
    $CP_{max} \leftarrow CP$; $C_{optimum} \leftarrow C$;
  end if
end for
```

of local features used in the experiments varies from 3.000 to 120.000, depending on the number of classes we use.

Fig. 2 shows how the CPM approach is able to find the RNN clustering threshold for which the cluster precision is maximum. We show results computing a vocabulary for 2, 4 and 20 classes in Figure 2(a), and for two descriptors (SIFT and SC). While for SIFT the best threshold for the RNN algorithm does not depend on the number of classes, being always 0.3, for SC descriptor the threshold is between 0.4 and 0.5. Note how for both descriptors the CP quickly drops for suboptimal threshold values. Results from applying an SVM for the category-level object recognition of 4 classes of objects (airplanes, faces, motorbikes and cars) are given in Fig.2(b). For the experiments we have used SVMs with radial basis kernels. The input vectors for the SVMs are the normalized histograms of visual words for each labeled image. We compare the performance of the codebook obtained with a classical BoW approach with Kmeans (K was fixed to 2000 in the experiments), with the vocabularies obtained with the CPM optimization process using RNN. As expected, the SVM trained with the CPM vocabularies outperforms the Bow+Kmeans for all classes. From Table 1 it is clear that good cluster precision is indeed a good prediction for a good classification accuracy: CPM applied to the SC descriptor gets the maximum both with respect to cluster precision and average precision per class.



(a) RNN for 2, 4 and 20 classes

(b) Precision-recall curves

Fig. 2. (a) shows the normalized CP running the CPM with different number of object classes (2, 4 and 20). (d) Precision-Recall curves obtained with the classification results.

Table 1. Cluster Precision vs. Average Precision

Clustering	K	CP	**airplane**	**cars**	**faces**	**motorbikes**
Kmeans-SIFT	2000	4.64e+9	0.88	0.65	0.79	0.92
Kmeans-SC	2000	4.72e+9	0.90	0.70	0.94	0.93
CPM-RNN-SIFT	1527	2.60e+11	0.95	0.88	0.96	0.97
CPM-RNN-SC	3080	**3.46e+11**	0.96	0.87	0.97	0.98

4 Conclusion

In summary, we have introduced an optimization procedure, coined Cluster Precision Maximization, that maximizes the clusters representativeness. The CPM method measures the cluster precision for each class and finds the clustering parameters that cast class representative visual words. A complete description of the method has been given, showing how to integrate a RNN agglomerative clustering algorithm in it. CPM evaluates the intrinsic quality of the clusters for a classification task and as a result, allows to compare the quality of clusters computed with different descriptors, as well as the quality of clusterings with different number of clusters. Results confirm that the vocabularies obtained with CPM get better results.

As future work, we plan to experiment with other descriptors and detectors, as well as with more challenging image databases like the PASCAL VOC challenge data [5]. Also, instead of using a single, global threshold, the method can be extended towards optimizing the threshold for each cluster separately. Another line of research involves bringing local information in the clustering process to discover semantic visual words.

Acknowledgments. We acknowledge support from the Flemish Fund for Scientific Research (FWO) and the Spanish projects MEC TEC2008-02077 and CAM CCG07-UAH/TIC-1740.

References

1. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Proceedings of the ECCV (2004)
2. Lowe, D.: Object recognition from local scale-invariant features. In: ICCV (1999)
3. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: Proceedings of the CVPR (2008)
4. van de Sande, K., Gevers, T., Snoek, C.: Evaluation of color descriptors for object and scene recognition. In: Proceedings of the CVPR (2008)
5. Everingham, M., et al.: The PASCAL voc 2008 Results (2008), `http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html`
6. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: A survey. Foundations and Trends in Computer Graphics and Vision 3(3), 177–280 (2008)
7. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on PAMI 27(10), 1615–1630 (2005)
8. Sivic, J., Zisserman, A.: Video data mining using configurations of viewpoint invariant regions. In: Proceedings of the CVPR, pp. 488–495 (2004)
9. Quack, T., Ferrari, V., Leibe, B., Van Gool, L.: Efficient mining of frequent and distinctive feature configurations. In: Proceedings of the ICCV (2007)
10. Yuan, J., Wu, Y.: Context-aware clustering. In: Proceedings of the CVPR (2008)
11. Lazebnik, S., Schmid, C., Ponce, J.: Semi-local affine parts for object recognition. In: Proceedings of the BMVC (2004)
12. Leibe, B., Ettlin, A., Schiele, B.: Learning semantic object parts for object categorization. Image and Vision Computing 26(1), 15–26 (2008)
13. Perronnin, P., Dance, C., Csurka, G., Bressan, M.: Adapted vocabularies for generic visual categorization. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 464–475. Springer, Heidelberg (2006)
14. Winn, J., Criminisi, A., Minka, A.: Object categorization by learned universal visual dictionary. In: Proceedings of the ICCV (2005)
15. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. In: Advances in NIPS (2006)
16. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: Proceedings of the CVPR (2007)
17. Mikolajczyk, K., Leibe, B., Schiele, B.: Local features for object class recognition. In: Proceedings of the ICCV (2005)
18. Stark, M., Schiele, B.: How good are local features for classes of geometric objects. In: Proceedings of the ICCV, pp. 1–8 (2007)
19. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Transactions on PAMI 24(24), 509–522 (2002)
20. Leibe, B., Mikolajczyk, K., Schiele, B.: Efficient clustering and matching for object class recognition. In: Proceedings of the BMVC (2006)
21. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: Proceedings of the CVPR (2004)