Experimental Evaluation of the Bag-of-Features Model for Unsupervised Learning of Images

Conference Paper · January 2015

DOI: 10.5244/C.29.146

Mariana Afonso
University of Bristol
4 PUBLICATIONS **5** CITATIONS

SEE PROFILE

Experimental Evaluation of the Bag-of-Features Model for Unsupervised Learning of Images

Mariana Afonso
marianafza@fe.up.pt

Department of Electrical and Computer Engineering, Faculty of Engineering, University of Porto

Luis F. Teixeira
luisft@fe.up.pt

INESC TEC and Department of Informatics Engineering, Faculty of Engineering, University of Porto

Abstract

This paper presents the results of an experimental study of the popular Bag-of-Features (BoF) model for the application of unsupervised learning of images, or image clustering. Although this method has been extensively applied for image classification and scene recognition, there has been few works which employ it in an unsupervised way. Also, due to the fact that the BoF model requires a great amount of steps, algorithms and parameter settings, we felt like there was a lack of detailed studies about the subject. We implemented testing routines in Python which we made publicly available in GitHub. In order to assess the performance of the model, three image datasets were used, namely, Coil-20 dataset, Natural and Urban dataset and Event dataset. The results obtained indicate that the BoF method provides a good representation of simple image collections for the purpose of clustering. However, it requires fine tunning of the parameters and algorithms for each dataset and obtains poor results for more complex scene datasets. We can therefore conclude that more advanced techniques are required in order to be able to effectively extract information from large image collections.

1 Introduction

The Bag-of-Features (BoF) is a model that aims to represent images as an orderless collection of features without the use of any spatial information. Each image is represented by a frequency histogram of visual words from a codebook. A visual word is a local segment in an image, defined either by a region (image patch or blob) or by a reference point with its neighborhood. The name comes from an analogy with the Bag-of-Words representation used in textual information retrieval (text mining). Although the model is quite simple with regards to the implementation, there are several steps in which parameters and algorithms need to be chosen.

This work aimed to assess the performance of this model for the application of unsupervised learning for a set of images, also called image clustering. Additionally, it aims to provide valuable insight on the different steps of the model and to compare different algorithms in order to achieve the best performance for a given dataset.

Pages 146.1-146.12
DOI: https://dx.doi.org/10.5244/C.29.146

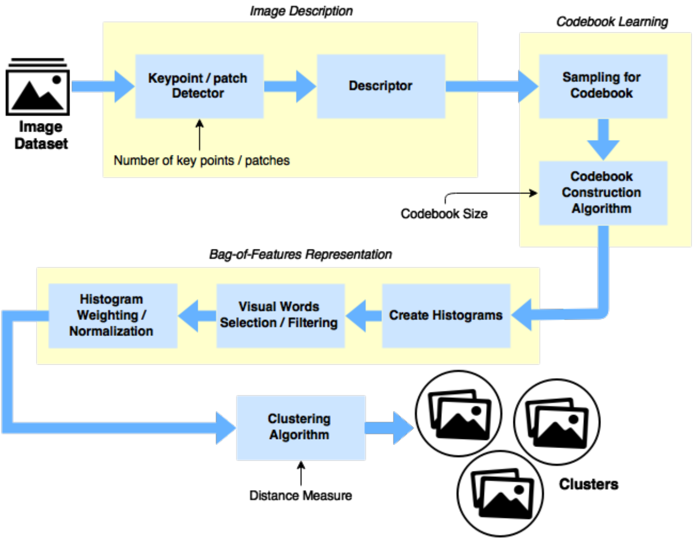
Image Description
Codebook Learning
Image Dataset
Keypoint / patch Detector
Descriptor
Sampling for Codebook
Number of key points / patches
Codebook Construction Algorithm
Codebook Size
Bag-of-Features Representation
Histogram Weighting / Normalization
Visual Words Selection / Filtering
Create Histograms
Clustering Algorithm
Distance Measure
Clusters

Figure 1: Main steps of the BoF model for the application of image clustering.

The fundamental difference between supervised learning (e.g. classification) and unsupervised learning (e.g. clustering) is that the data is not annotated and thus there is no previous information about the categories. For this reason, the methods used aim to find an underlying structure of the data and obtain relevant partitions.

The applications of image clustering are endless and could include social network mining, more specifically for summarization of the huge amount of content shared everyday by millions of users. This could also provide a new visualization of what is happening at a given time, based on the pictures being shared online.

The process of the BoF model and the main steps are summarized in Figure 1. As shown in this Figure, there are three main parts in the BoF model designed for image clustering. The first one is the image description step, in which the input images from the dataset are processed by first detecting keypoints or patches and then describing them using a certain strategy. The number of keypoints per image is a parameter that can be varied in the implementation for almost all the detection algorithms. The next step is codebook learning, where a portion of the feature vectors from the images are used in order to obtain a codebook of visual words. Here, the codebook size is a very important parameter that can be specified to obtain different codebooks. The following step is the BoF representation of the images where each image is represented by a histogram of frequencies based on the codebook obtained. The words are then filtered and the histograms are normalized following a chosen methodology. Finally, the images are clustered using a clustering algorithm of choice.

2 Related Work

Clustering is an important tool that has been applied extensively for many types of data. In computer vision, this type of unsupervised learning has been used for several applications such as for image annotation [7], image summarization [23] and the performance improvement of Content-based Image Retrieval (CBIR) systems [8]. Different techniques have been applied but recently the most popular approach for image clustering has been to use the Bag-of-Features, also called Bag-of-Visual-Words model [24].

Due to the popularity of the BoF model, a number of works have been focused on evaluating its performance. Moreover, due to the great number steps needed to apply the model to a given problem, these studies also compare different strategies for each of the steps. For instance, in [16] the authors presented the results of an experimental study concerning the BoF model applied to the problem of image classification. Several key steps of the model were tested using different algorithms and parameters, including the detection of the interest points, the size of the codebook and the histogram normalization procedure. Their results show the most influential parameter is the number of patches extracted from the images. Additionally, they have also determined that the codebook learning method does not have a significant impact on the performance provided that even randomly sampled codebooks also performed fairly well.

Another empirical study presented in [27] evaluated the impact of applying techniques used in text categorization to the BoF model for the application of scene classification. More specifically, they tested, among others, the use of term weighting, stop word removal and feature selection. The results indicate that these techniques successfully improve the classification results. Other example of a similar work that proposes and evaluates the use of text classification techniques for the BoF model is [14].

The main contributions of this study are: (1) the experimental analysis of the BoF model for image clustering, (2) the addition of a number of steps and algorithms (e.g. sampling the features for codebook learning and visual words selection) and (3) the proposal of a sampling technique for the features obtained from the images for the codebook learning procedure (called SAMPLEI in our routines).

3 Experimental Design

As mentioned before, the first step of the BoF method consists of obtaining image descriptors for each image in the dataset. Among the different methods, the detectors used in this study were: SIFT (Scale-Invariant Feature Transform) [12], SURF (Speeded Up Robust Features) [4], FAST (Features from Accelerated Segment Test) [20], STAR - derived from CenSurE (Center Surrounded Extrema) [1] and ORB (Oriented FAST and Rotated BRIEF) [21] and the descriptors were: SIFT, SURF, BRIEF (Binary Robust Independent Elementary Features) [6], ORB, and FREAK (Fast Retina Keypoint) [2]. More details about the parameters used for these algorithms can be found in Section 2.1 of the Supplementary Material.

Similarly to [16], in order to test whether the clustering performance is influenced by the keypoint detection algorithm, a random generator of patches (RANDOM) was also used. It works by randomly sampling the output of the a DENSE detector [5], which produces a regular grid of interest patches.

After the descriptors for each image are obtained, the codebook learning method is performed. For this purpose, two clustering algorithms were selected: K-Means [13] and Mini Batch K-Means [22]. These algorithms were chosen due to their high scalability property. K-Means is by far the most popular algorithm for this application and has been used in almost all the works that applied the BoF for either image classification or image clustering [7, 9, 18]. In order to attempt to reduce the computational complexity of the codebook learning algorithm, the Mini Batch K-Means algorithm was tested. Additionally, with the aim of testing whether the codebook learning algorithm is significantly relevant to the performance of BoF model, the last methods adopted for constructing the codebook was using randomly selected feature vectors from the images (RANDOMV) and also entirely random vectors (RANDOM).

Next, instead of using all the features obtained from the images to produce the codebook, two types of sampling strategies were adopted. The first one is simply selecting random keypoints from the entire dataset. However, given that some images generate more interest points than others, we believe that this could potentially have a negative impact on the codebook and consequently on the performance of the model. For this reason, we tested a simple algorithm for adaptative sampling of the images in order to reduce the standard deviation of the keypoints detected per image. The algorithm first selects a random sample of images. Then, for each image that was chosen, it randomly selects a proportion of the keypoints in order to construct the codebook. This proportion of keypoints sampled per image depends on the relation between the number of keypoints of that image and the average number of keypoints per image of the entire dataset. Therefore, this algorithm attemps to reduce the variability in the number of keypoints per image when selecting the visual words.

After obtaining the codebook, each image is represented by a histogram of frequency of visual words from the codebook. However, before the features are ready for clustering, they are filtered and the histograms are be normalized. In terms of feature selection, three alternative were tested: removing frequent visual words, removing rare visual words or both. In relation to histogram normalization, the methods tested were simple binarization and different forms of the term frequency-inverse document frequency technique (tf-idf) [24].

In the last step of the process, the clusters are obtained using a given clustering algorithm. A number of different approaches were tested including K-Means, DBSCAN, BIRCH and two Hierarchical Clustering implementations, HIERAR1 and HIERAR2. Additionally, some of these methods allow the choice of the dissimilarity metric used. That parameter was also varied in order to obtain different clustering results.

Three datasets were used in this empirical study in order to obtain different levels of difficulty and complexity for the purpose of image clustering. Examples of images from each dataset can be found in Figure 2. The first one is the popular Coil-20 dataset [15], which is an object-based dataset. It is composed by 1440 small pictures of size 26×26 pixels divided into 20 classes of objects taken under different perspectives (lightning, rotation, etc). This was considered the simplest dataset. The second dataset used was the Natural and Urban Scenes dataset [17], which is made from 8 nature and human-made scenes such as coastlines and buildings. It has 2688 images which have a dimension of 264×264 pixels. it was considered medium difficulty. Finally, the last and more complex image dataset used was the Event Dataset [11] which is composed by 1580 images of 8 sports event categories. The images from this datasetwere reduced to 500 pixels in the largest side for simplicity and to reduce the execution time.

In relation to the performance measures, external clustering indexes were used. An information-based method, Normalized Mutual Information (NMI) [26] and a decision-based method, Adjusted Rand index (ARI) [28], were selected. Both these indexes are popular choices for clustering validity and since they have different natures, a more complete evaluation was possible.

The software for testing the BoF model was developed in Python and it is openly available on GitHub [1]. The implementation requires three Python libraries: OpenCV [5] for the functions related to image description, Scikit-Learn [19] and Scipy [10] for the implementa-

[1] The source code of this project can be found in the link: https://github.com/marianaAfonso/BOFClustering

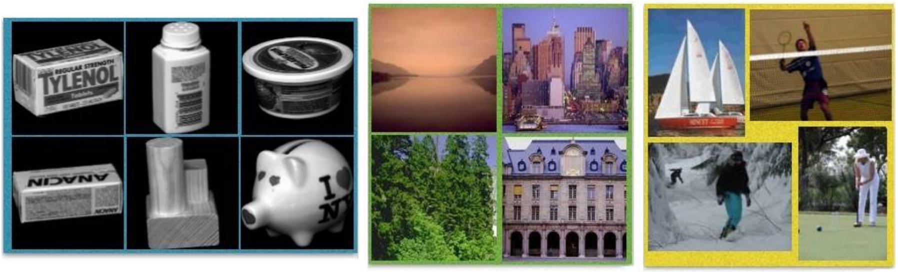

Figure 2: Examples of images from the public image datasets. From Left to right: Coil-20 dataset, Natural and Urban dataset and Event dataset.

tion of the machine learning algorithms tested.

4 Results

4.1 Image Description

4.1.1 Detectors and Descriptors

First, the different detectors and descriptors for the stage of image description were tested. For these tests, all the other settings of the BoF model for image clustering were fixed. The K-Means algorithm was selected as the codebook learning algorithm and the final clustering algorithm. Also, the size of the codebook and the proportion of keypoints to be used for the process of codebook learning were fixed to a certain value for each dataset. These values can be found in Section 2.1 of the Supplementary Material. Additionally, as the K-Means clustering algorithm does not take into account if the features have different scales, a whitening transformation of the features from the histograms was applied prior to the application of this clustering algorithm.

The results of this analysis for all three datasets can be found in Table 1, where the performance of the best and the worst combination of detectors and descriptors are presented. The table contains the following information: average ARI, standard deviation of the ARI, average NMI score and standard deviation of the NMI score, average number of keypoints per image and finally a relative qualitative value for the computational time required. In order to obtain the average and the standard deviation of the indexes, every test was repeated 10 times.

By analyzing the results for the Coil-20 dataset, it can be seen that the best performing combination was the FAST detector with the FREAK descriptor and the SURF descriptor. In contrast the worst combinations of detectors and descriptors for this dataset was found to be the RANDOM detector with the SURF descriptor. In general, the RANDOM detector performed poorly for this dataset. These results are not surprising since the images represent objects with a black background which will most likely generate a great number of keypoints and will be seen as noise for the BoF model.

In relation to the Natural and Urban dataset, the best performing descriptor is definitely the SIFT descriptor. An interesting result is that the RANDOM detector achieved very good results, by which can be concluded that using specific interest point detectors can yield worse

Table 1: Performance and additional information concerning the application of the BoF model for image clustering using different detectors and descriptors for the three datasets.

Dataset	Detector	Descriptor	Avg ARI	Std ARI	Avg NMI	Std NMI	Avg. # of keypoints / img.	Computational time
Coil-20	**FAST**	**FREAK**	**52,2%**	**3,9%**	**78,7%**	**1,5%**	**88**	**High**
	FAST	**SURF**	**48,8%**	**4,3%**	**76,2%**	**1,5%**	**88**	**Medium**
	SIFT	**SIFT**	**46,4%**	**4,9%**	**75,3%**	**2,1%**	**51**	**High**
	RANDOM	FREAK	32,8%	2,8%	53,7%	2,0%	50	High
	ORB	ORB	19,3%	2,1%	40,6%	1,8%	11	Low
	RANDOM	SURF	12,4%	0,9%	28,4%	1,4%	50	Medium
Natural and Urban	**STAR**	**SIFT**	**34,2%**	**2,2%**	**46,0%**	**1,6%**	**130**	**Low**
	RANDOM	**SIFT**	**31,2%**	**0,8%**	**41,8%**	**1,2%**	**500**	**Medium**
	SURF	**SIFT**	**27,1%**	**1,6%**	**38,7%**	**1,0%**	**332**	**High**
	SIFT	SURF	14,0%	1,1%	25,2%	1,4%	393	Medium
	STAR	BRIEF	13,8%	1,3%	23,4%	1,4%	130	Very Low
	FAST	FREAK	11,8%	0,5%	21,1%	0,4%	851	Low
Events	**RANDOM**	**SURF**	**18,7%**	**0,8%**	**27,1%**	**0,6%**	**1000**	**High**
	STAR	**SIFT**	**16,5%**	**1,0%**	**26,5%**	**0,9%**	**554**	**High**
	SURF	**SIFT**	**15,9%**	**0,8%**	**25,9%**	**0,3%**	**972**	**Very High**
	FAST	BRIEF	5,4%	0,3%	13,0%	0,2%	1038	Low
	FAST	FREAK	5,2%	0,2%	11,1%	0,4%	972	Medium
	ORB	ORB	4,1%	0,3%	8,1%	0,5%	957	Low

results for scene datasets than randomly selecting patches from the whole image.

Finally, concerning the Event dataset, the descriptors SIFT and SURF achieved the best results in contrast to the binary descriptors BRIEF, FREAK and ORB. It is clear from the results that this is a very challenging dataset with regards to unsupervised learning.

In relation to the computational time, the SURF and SIFT detectors and descriptors are among the fastest algorithms. Therefore, for larger datasets and/or real-time applications a more efficient combination of detectors and descriptors should be selected, for instance, using the FAST, STAR or RANDOM detectors and the BRIEF or FREAK descriptors.

In conclusion, after analyzing these results, it can be observed that the performance of the BoF model applied to unsupervised learning of image data highly depends on the algorithms for the description of the images and not so much on the detector (except for object datasets). Also, the choice of the algorithms is dependent of the dataset.

4.1.2 Number of Keypoints and Codebook Size

In this step, several combinations of the average number of keypoints per image and the size of the codebook were tested. These parameters are correlated since the more features extracted from the images, the more diversity of visual words will exist and, therefore, the size of the codebook can increase. Figure 3 presents the results for all the datasets. Here, the performance index was chosen as the NMI score since both the NMI score and the ARI followed the same trends.

As shown in the charts of Figure 3, regardless of the codebook size used, the performance almost always increased with the average number of keypoints per image. As referred in Section 2, this result was also obtained in [16] for the problem of image classification.

Another interesting conclusion, also observed in [16], is that the performance increases with the codebook size until a certain point in which the performance starts to go down. This behavior can probably be attributed to the curse of dimentionality [25], in which the sparsity of high dimensional features results in unpredictable effects that the clustering algorithm cannot handle. Additionally, it could be observed that the ideal size of the codebook increases with the complexity of the dataset.

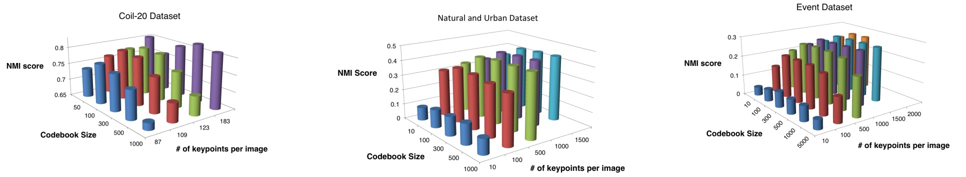

Figure 3: Performance of the BoF model for the three datasets using different values for the average number of keypoints per image and the codebook size.

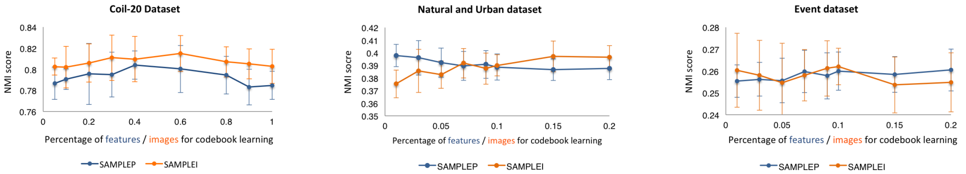

Figure 4: Results for the three datasets using different values for the proportion of images and features used for the codebook learning step and using two different methods for selecting or sampling these features SAMPLEP and SAMPLEI.

4.2 Sampling for Codebook Learning

Next, the influences of the sampling technique and the proportion of keypoints used for the codebook learning algorithm were tested. As mentioned before, two methods for sampling were evaluated, named in our testing framework SAMPLEP and SAMPLEI. For both techniques, the number of keypoints used was varied considering the time that would be required for each dataset: for the Coil-20 dataset, up to 100% of the keypoints were tested whereas for the two most complex datasets, the maximum portion tested was 20%.

The NMI scores for the three datasets are presented as charts in Figure 4. The error bars represents the standard deviation of the NMI score. It is important to note that for the SAMPLEI method, the x-axis refers to the percentage of images used for codebook learning whereas for the SAMPLEP method it is the percentage of keypoints. This is because SAMPLEI method downsamples the images with above average number of keypoints per image and therefore less features will be sampled.

In terms of the sampling algorithm, it can be seen that the SAMPLEI method performs, in average, slightly better than the SAMPLEP only for the Coil-20 dataset. This could be caused by the level of complexity of the images, since for the object dataset, the ratio between the category with the most keypoints and the category with the least keypoints was approximately 11 whereas for the other two datasets it was around 2. Additionally, it was observed that varying the percentage of keypoints or images used for the codebook learning procedure only improved the results up to 4%, which is not very significant. For this reason, even if no sampling was used when selecting the subset of keypoints used for codebook learning, the performance would probably remain under the same values.

4.3 Codebook Learning Method

Regarding the codebook learning method, the results are presented in Table 2. For each algorithm, the average NMI score is presented.

For all the datasets, the K-Means was, on average, the best performing algorithm. Nonetheless, for the Natural and Urban dataset, the Mini-Batch K-Means had almost the same score, and therefore, would be more desirable considering the less time computational required. As for the Event dataset, K-Means, Mini-Batch K-Means and Random Vectors, RANDOMV, got the same result, and therefore, it would be more efficient to use RANDOMV. In contrast, for the Coil-20 dataset, there is a significant difference between K-Means and Mini-Batch. Finally, the completely random codebook, RANDOM, got significantly poorer results in terms of the clustering validity indexes. This result might come from the fact that the visual words could be placed very far from the feature vectors, in the feature space, and thus creating a poor representation of the images.

Given that codebook obtained by randomly selecting feature vectors achieved good results, it can be concluded that the choice of the codebook learning method does not significantly influence the performance for image clustering and this influence reduces with the complexity increase of the images in the dataset.

Table 2: Performance of the BoF model for image clustering evaluated by the NMI score for different algorithm for codebook learning.

Algorithm	Coil-20 Dataset	Natural and Urban Dataset	Event Dataset
K-Means	81.0%	42.8%	27.2%
Mini-Batch	77.9%	42.4%	27.4%
Random Vectors	75.9%	40.8%	27.7%
Random	53.9%	16.3%	21.7%

4.4 Feature Selection

Next, an attempt was made in order to apply simple feature selection methods to the visual words of the obtained codebook by filtering the most and least frequent features. The clustering results obtained have shown that this step did not improve the results and even made them worse. For this reason, there is evidence to suggest that applying these types of methods, which are usually used in text mining, does not improve the results for the application of image clustering. More details can be found in Section 2.3 of the Supplementary Material.

4.5 Histogram Weighting and Normalization

After obtaining the histograms of frequency of visual words for each image in the dataset, the normalization and weighting of the histograms can be performed. For this purpose, five types of normalization and weighting procedures were tested. Details concerning the techniques used are presented in Table 3. The variable $f_((t,d)$ refers to the frequency of the word t in the document (or in this case, image) d. The number of images is given by N and the number of images that have the visual word t is given by n_t. Furthermore, the results of the application of these normalization procedures for the datasets tested can be found in Figure 5. In the tf-idf variant, $N*$ is the total number of features (sum of all visual word frequencies) and n_t* is the total number of incidences of that visual word of all images.

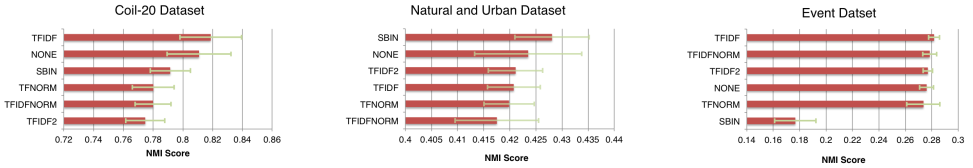

Figure 5: Performance of the BoF model for image clustering evaluated by the NMI score for different techniques for weighting and normalization of the histograms.

By analyzing the charts in Figure 5, it is clear that no technique outperforms the others in all three datasets used. More specifically, for the Coil-20 dataset, the method that achieved the best performance was the tf-idf and the one that got the worst was the tf-idf variation. In relation to the Natural and Urban dataset, the best was simple binary normalization while the worst was the tf-idf normalized. Finally, concerning the Event dataset, almost all methods got similar result apart from the simple binarization, which obtained significantly worse results.

In summary, although the use of normalization and weighting can help to improve the results of the BoF model for image clustering, it does not significantly influence it, and therefore is not a core step.

Table 3: Methods tested for histogram normalization and weighting.

Method	Mathematical Expression
Simple Binarization (SBIN)	1 if ti is present, 0 if not
tf-idf (TFIDF)	$f_{(t,d)} \cdot \log(1+\frac{N}{n_t})$
tf-idf variation (TFIDF2)	$f_{(t,d)} \cdot \log(1+\frac{N*}{n_t*})$
tf Normalized (TFNORM)	$\frac{f_{(t,d)}}{\sum_d f_{(t,d)}}$
tf-idf Normalized (TFIDFNORM)	$\frac{f_{(t,d)} \cdot \log(1+\frac{N}{n_t})}{\sum_d f_{(t,d)} \cdot \log(1+\frac{N}{n_t})}$

4.6 Clustering Algorithm

The last step of the testing procedure is the clustering algorithm. Both the clustering algorithm and the distance measure for computing the dissimilarity between images were varied. The results are presented in Table 4.

Among the different algorithms tested only DBSCAN and HIERAR2 do not require the number of clusters as a parameter, which is desirable for a total unsupervised fashion. However, by analyzing the results, only in the Coil-20 dataset, the HIERAR2 achieved comparable results with the other methods. It was verified that regardless of the attempts in changing the parameters of the DBSCAN algorithm, it either found too many data points as noise, or considered a great number of images to be part of the same cluster. In contrast, both BIRCH and K-Means performed very well.

Regarding the distance measures, for the Hierarchical Clustering approaches, the best performance was achieved using the cosine and correlation distance measures. This shows the applicability of those metrics for this application.

The conclusion of this last step of the BoF method applied for unsupervised learning of images is that, although an algorithm which does not require the number of clusters is desirable, it is not an easy task, since usually those algorithms require other parameters that need to be adjusted and can be very specific to a given set of images. For this reason, a better alternative might be to compute the clustering algorithm for different number of clusters and then select the one that maximizes a given internal index, such as the silhouette index [3].

Table 4: Results for the three datasets using different algorithms for the final clustering step.

Coil-20 dataset

Normalization	Clusting Algorithm	Distance measure	Avg. ARI	Avg. NMI score	Number of clusters
SBIN	BIRCH	hamming	**67,4%**	**84,8%**	**20**
TFIDF	KMEANS	euclidean	59,6%	81,9%	20
NONE	HIERAR1	correlation	56,9%	82,7%	20
NONE	HIERAR2	cosine	54,5%	81,4%	>150
SBIN	DBSCAN	correlation	18,0%	64,2%	15 avg.
Natural and Urban dataset					
NONE	KMEANS	euclidean	**30,2%**	**40,6%**	**8**
NONE	BIRCH	euclidean	27,4%	37,9%	8
SBIN	HIERAR1	cosine	25,7%	37,6%	8
NONE	HIERAR2	cosine	8,1%	44,6%	>700
NONE	DBSCAN	correlation	5,8%	36,6%	>1200
Event dataset					
NONE	KMEANS	euclidean	**19,4%**	**27,4%**	**8**
TFIDF	BIRCH	euclidean	17,1%	25,6%	8
TFIDF	HIERAR1	correlation	15,8%	23,6%	8
NONE	HIERAR2	correlation	9,5%	48,0%	>860
TFIDF	DBSCAN	cosine	2,3%	35,0%	>700

5 Conclusion

This work aimed to evaluate the performance of a very popular model for image representation, called Bag-of-Features and to test different algorithms and parameters for the various steps of the model.

As a result of these experiments, the steps or parameters that most influenced the performance of the model for image clustering were the algorithm for image description, the average number of keypoints per image, the size of the codebook and the final clustering algorithm.

Another interesting observation was that, although having been proposed several decades ago, the K-Means algorithm continues to be a very fast and robust alternative for the codebook learning algorithm and for the clustering algorithm compared to other recent approaches.

Additionally, from all the different experiences developed and presented in this work, it can be concluded that although the Bag-of-Features model can be successfully applied to the problem of unsupervised learning for visual data, it provides a poor representation of the images when the datasets represent complex scenes. This was clearly illustrated by the results for the Event dataset.

For this reason, more advanced techniques are required in order to be able to effectively extract information from large image collections in an unsupervised way. Also, there needs to be further research towards a better understanding of visual data and the way humans evaluate the similarity between images.

References

[1] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. Censure: Center surround extremas for realtime feature detection and matching. In *Computer Vision–ECCV 2008*, pages 102–115. Springer, 2008.

[2] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517. IEE, 2012.

[3] Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, Jesús M Pérez, and Iñigo Perona. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1): 243–256, 2013.

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010*, pages 778–792. Springer, 2010.

[7] Jie Cao, Zhiang Wu, Junjie Wu, and Wenjie Liu. Towards information-theoretic k-means clustering for image indexing. *Signal Processing*, 93(7):2026–2037, 2013.

[8] Yixin Chen, James Ze Wang, and Robert Krovetz. Clue: cluster-based retrieval of images by unsupervised learning. *Image Processing, IEEE Transactions on*, 14(8): 1187–1201, 2005.

[9] Vincent Delaitre, Ivan Laptev, and Josef Sivic. Recognizing human actions in still images: a study of bag-of-features and part-based representations. In *BMVC 2010-21st British Machine Vision Conference*, 2010.

[10] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL `http://www.scipy.org/`. [Online; accessed 2015-02-26].

[11] Li-Jia Li and Li Fei-Fei. What, where and who? classifying events by scene and object recognition. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[12] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[13] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. California, USA, 1967.

[14] Christophe Moulin, Cécile Barat, and Christophe Ducottet. Fusion of tf. idf weighted bag of visual features for image classification. In *Content-Based Multimedia Indexing (CBMI), 2010 International Workshop on*, pages 1–6. IEEE, 2010.

[15] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-20). Technical report, Technical Report CUCS-005-96, 1996.

[16] Eric Nowak, Frédéric Jurie, and Bill Triggs. Sampling strategies for bag-of-features image classification. In *Computer Vision–ECCV 2006*, pages 490–503. Springer, 2006.

[17] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3): 145–175, 2001.

[18] Symeon Papadopoulos, Christos Zigkolis, Yiannis Kompatsiaris, and Athena Vakali. Cluster-based landmark and event detection for tagged photo collections. *IEEE MultiMedia*, 18(1):52–63, 2011.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.

[21] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[22] D Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178. ACM, 2010.

[23] Ian Simon, Noah Snavely, and Steven M Seitz. Scene summarization for online image collections. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[24] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003.

[25] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition.* Academic Press, 2008.

[26] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:2837–2854, 2010.

[27] Jun Yang, Yu-Gang Jiang, Alexander G Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 197–206. ACM, 2007.

[28] Ka Yee Yeung, David R. Haynor, and Walter L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17(4):309–318, 2001.