BMAX: a bag of features based method for image classification

Pedro Senna†, Isabela Neves Drummond *, Guilherme Sousa Bastos †
† Instituto de Engenharia de Sistemas e Tecnologias da informação
* Instituto de Matemática e Computação
Universidade Federal de Itajubá - UNIFEI
{pedrosennapsc, isadrummond, sousa} @unifei.edu.br

***Abstract*—This work presents an image classification method based on bag of features, that needs less local features extracted for create a representative description of the image. The feature vector creation process of our approach is inspired in the cortex-like mechanisms used in "Hierarchical Model and X" proposed by Riesenhuber & Poggio. Bag of Max Features - BMAX works with the distance from each visual word to its nearest feature found in the image, instead of occurrence frequency of each word. The motivation to reduce the amount of features used is to obtain a better relation between recognition rate and computational cost. We perform tests in three public images databases generally used as benchmark, and varying the quantity of features extracted. The proposed method can spend up to 60 times less local features than the standard bag of features, with estimate loss around 5% considering recognition rate, that represents up to 17 times reduction in the running time.**

***Keywords*-Image classification; bag-of-features; HMAX; low feature usage;**

I. Introduction

The problem of classifying an image content is a difficult challenge for Computer Vision. The main reference in this problem is the human natural ability in classifying images semantically, that is impressive, both for its performance and by its response time.

Starting with response time, the simple Haar cascade [1] is the first method to achieve face recognition in real-time. It needs a huge amount of positive and negative samples from the target object for training, and do not deal well with intra-class variance. For example, if a Haar cascade model is trained to detect a red traffic cone with two white strips, it will not detect one with three white strips.

In other way, some methods emphasize performance working just because of the evolution in computational power and use of graphics processing units (GPUs). One example is the deep convolutional neural networks [2]–[4] that can achieve human-like performance, but is non-viable to real time applications. However, we are still far from achieve human-like performance and response time.

In this work, we proceed to investigate a technique that can achieve good relation between recognition rate and computational cost. Our proposal is based in two methods: *bag of features* (BOF) [5]–[8], also known as *bag of visual words*, and *Hierarchical Model and X* (HMAX). The BOF extracts local features from the image, and match them with the vocabulary of visual words. Each feature is attributed to a visual word, and the occurrence of the visual words are counted. Then, we create a histogram of visual words for the image, also known as image bag of features. The bag of features is the image representation, and is used as the image feature vector. The second method, HMAX, was proposed by [9] and extended by [10]. It is a model that simulates the primate prefrontal cortex functions for classifying image objects, and is divided into four layers, where simple layers employ local filter convolutions and complex layers perform maximizations and sampling operations.

We propose a new approach, the *bag of max features* (BMAX) method which is based on HMAX to create the feature vector. The main difference from BOF is that BMAX find the closest local feature from each visual word and use their distance instead of occurrence counting to compose the feature vector. This method can handle better situations with less features used compared to BOF. This situation can be faced either by lack of available features, in case of small images classification, or by option, to improve the response time.

There are some works that also try to reduce the computational complexity of BOF in the classification step, without worrying with the training. Uijlings et al. [11] made changes in all steps: modification of SIFT and SURF to improve the extraction time using a regular grid, employment of a random forest to create the vocabulary and proceeded the dimensional reduction in the feature descriptors using principal component analysis. Galvez-Lopez and Tardos [12] proposed a BOF-based approach to detect closed loop in real time applied to robotics. They used the BRIEF binary descriptor and FAST key point detector to reduce the feature extraction time.

Our proposal reduces the computational complexity using less local features per image, bringing down the feature extraction time and assignment with the vocabulary.

This paper is organized as follows. In Section II we present the formal definitions and necessary background to follow the work. In Section III we formally present our approach describing the proposed method. In Section IV presents the test methodology and experimental results, and finally, Section V states the conclusion and points to future research directions.

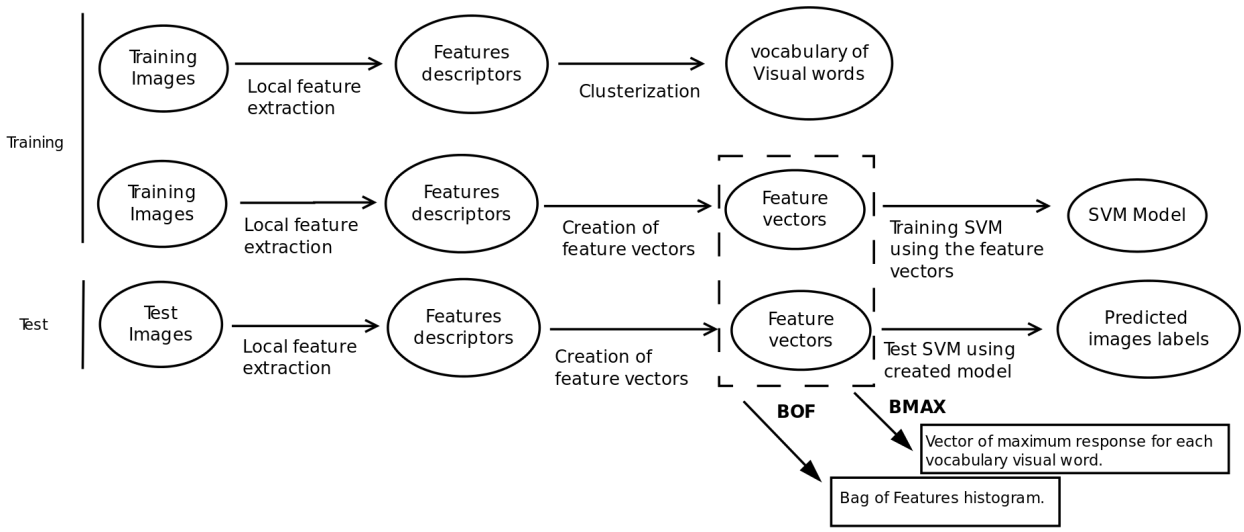

Fig. 1. Flowchart of BMAX and BOF algorithms.

II. Background and definitions

In this section we present the two methods used to compose our proposed method: *Bag of features* and *Hierarchical Model and X*. The first one is the base of our proposed method and the second is the inspiration of the process for image feature vector creation.

A. Bag of features

To perform the classification task, BOF model [6] employs features extracted from texture image and put them into a vocabulary to measure the occurrence of each word and to create a histogram of occurrences. The algorithm assumes that a class is defined by the occurrence of the visual words not considering the position of them. Fig.1 shows a flowchart of the proposed method.

In the local feature extraction process the keypoints (interest points) are identified and described. The identification of keypoints aims to find the relevant image regions. Sparse methods for keypoint detection are employed by [6]–[8], such as the difference of gaussians [13] that search for local extremum in the image. However, [14] and [15] report better results using a dense regular grid for keypoint detection. This grid has two parameters, the feature size(S) and the grid step(ds), which influence the amount of keypoints found. Fig. 2 depicts a sample of two regular grids and parameters influence in the feature detection process. We always use S as twice the value of ds.

For the feature description, the scale invariant feature transform (SIFT) [13] is used. In the original proposal, each point can be described in more than one orientation, but in this work only one orientation per keypoint is considered.

In the training process, the visual words vocabulary is created using a clusterization algorithm over the local feature descriptors extracted from the training images. In this work, we use Kmeans algorithm [16] that is widely used in BOF implementations. Each cluster found represents a visual word being the cluster center used as its reference.

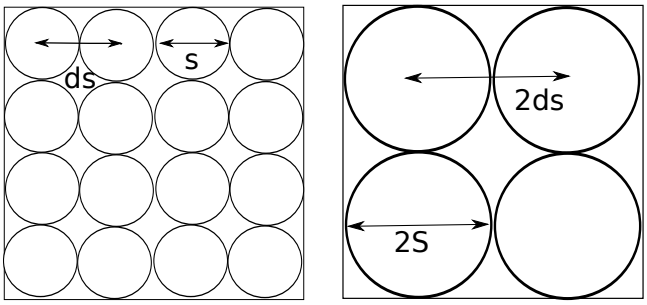

Fig. 2. Samples of regular grids where each circle is a detected feature. Keypoints are reduced by four when the grid step is doubled considering that grid fits perfectly the image.

The vocabulary is used to create the feature vector that is a representation of the image. For each image, its local features are extracted and assigned to the nearest visual word, using the euclidean distance to the visual word center as reference. Then, a histogram is built using the occurrence frequency of each visual word, which is normalized. This histogram is used as the image feature vector.

The feature vector of the training images is delivered to a support vector machine (SVM) [17]. A radial basis function is selected as the SVM kernel, and the one-against-all approach is applied for multi-class problems.

B. Hierarchical Model and X

The *Hierarchical Model and X* [9], [10] is based on the process of image recognition that occurs in the primate prefrontal cortex. This model is composed by four layers: S1, C1, S2, and C2.

S1 acts as the primate primary visual cortex. It applies a set of Gabor filters [18] with different orientations and scales in the input image. The result maps are grouped into bands, two

by two scales with all the orientations of each scale.

In C1, for each band, a maximum filter is applied over scales and positions for the same orientation and the result map is sub-sampled. We obtained one map for each orientation in each band. In the training process, random patches are extracted being used as references for visual features in a similar way of BOF visual words vocabulary.

In S2, the group of patches extracted from C1 layer during the training is matched with the C1 layer output, and the output of S2 layer is computed using the equation 1.

$$R(i,j) = exp(-\gamma||X - P_i||^2) \quad (1)$$

Where X represents the image patches for all positions, and Pi is each one of the extracted patches during training (for i ranging from 1 to the number of extracted patches). The parameter γ is the same used in the Gabor filter. This result represents the response of each image region to the visual characteristics represented by each patch.

In C2, the maximum response for each patch is found and used to compose the feature vector. It is based in the similarity of the nearest patch on the image from each patch of the group extracted in the training, representing the most similar region in the image to their visual characteristic. Then, the feature vector is delivered to a support vector machine for training using a radial basis function as kernel.

III. Proposed Method

The proposed approach is the *Bag of max features*(BMAX), that is similar to the BOF, but using the S2 and C2 layers of HMAX to generate the image feature vector.

BMAX employs a vector of best responses for each vocabulary visual word as the feature vector. The process to create this vector can be divided into two steps, inspired in S2 and C2 HMAX layers.

In the first step we compute the responses (R) from the vocabulary words (W) for each feature descriptor found (D) in a gaussian-like way, using the Equation 2.

$$R(i,j) = exp(-||W_i - D_j||^2/\alpha) \quad (2)$$

Where the difference between W and D vectors are their euclidean distances, and α defines the sharpness of the tuning being dependent of the descriptor nature. For SIFT descriptors, we found empirically a good value for α as around 100,000.

The second step is the global maximization. This process consists in finding the maximum response for each vocabulary visual word, while the other responses are discarded. The final result is a vector of size N, where N is the vocabulary size. The process to discover the best response for a visual word is illustrated in Fig. 3.

To classify an image from a trained model, we need four steps: obtain the image, extract the local features, construct the feature vector and classify using the SVM, as we can see in the test step presented in Fig. 1. The cost to capture the image is related to the camera and the drivers or the disk speed and the time spent in the classification with SVM showed to be

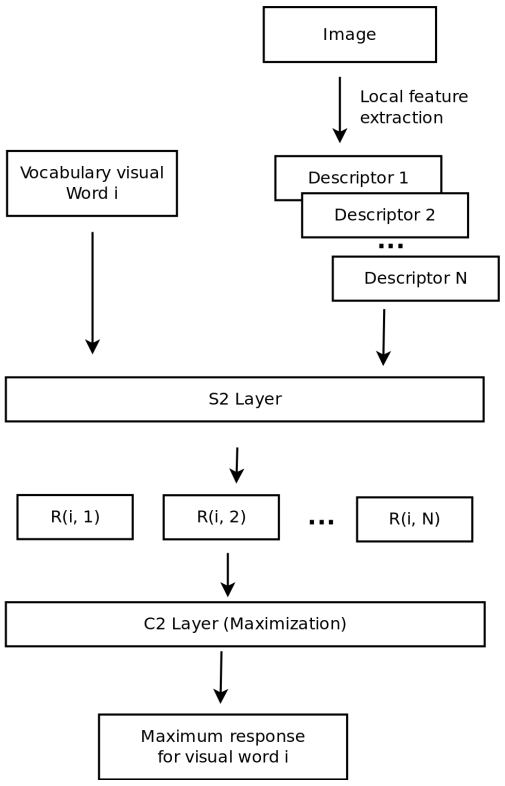

Fig. 3. Process to find the best stimulus for a vocabulary visual word. This process are repeated with each vocabulary word in order to create the feature vector.

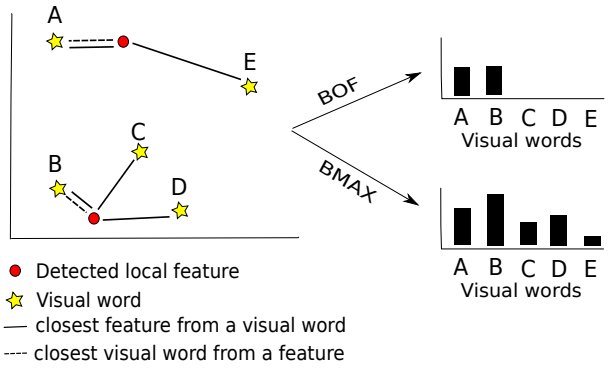

Fig. 4. Toy example of creating the image feature vector using BMAX and BOF in situations with few features available.

irrelevant in our test, but can be a threat to problems with more classes or major feature vectors. The main computational cost source to classify an image with this methods is the feature extraction and creation of the feature vector. Thus, the number of features used affects directly the run time methods.

Using this approach, we can aggregate more information to the image feature vector when few local feature are extracted. In the standard BOF model, each local feature can be assigned just to one visual word. Considering visual vocabulary with hundreds of words (from 400 to 800, like we had in our tests), a high number of local features is required to create a

histogram not composed basically of null values. Employing our approach, each local feature can be assigned to the best answer of all visual words without limitations. Therefore, all the values in the feature vector will have representative values, even if just one local feature is extracted (Fig. 4). Note that, with this characteristic, our proposed method can be used with less local features minimizing recognition rate drop when compared to BOF.

IV. Experimental Results

In the following we present the experimental results. We studied the behavior of BMAX method applied to three datasets, fifteen scene categories [19], Caltech-101 [20] and CIFAR10 [21]. For each test, we performed the algorithm ten times, varying the training and test images randomly. The results are represented by classifier accuracy for each class and for the final results we analyzed mean and a 95% confidence interval considering 10 executions. All the tests were carried out on grayscale images, being colored images converted to grayscale before feature detection.

A. CIFAR10

The CIFAR10 database [21] is composed of small images, 32x32 pixels, from 10 categories. This base contains 60,000 images, 6,000 per class. The difficulty of this base is the intra-class variance; for the bird class, the image can be of a distant flying bird or just his head. Sample images from CIFAR are displayed in Fig. 5.

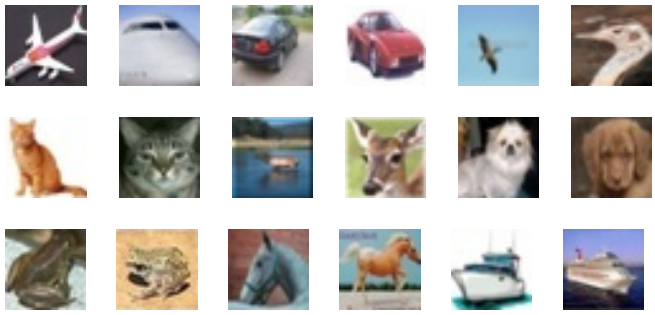

Fig. 5. Sample images from CIFAR10 database.

Originally, this base is used with predefined training (5,000 images per class) and test sets (1,000 images per class). However, in our tests we mixed the images and selected the training and test sets randomly for each experiment applying BOF and BMAX.

In Table I, the results are shown for classification experiments undertaken for CIFAR10. The first column presents the regular grid parameters, the second one the number of detected features used in this grid. In this case we selected 100 images for each class for training and 2,000 for testing, and vocabulary size of 400. For this set of images, in all the scenarios, our approach BMAX led to better classification performance when compared to BOF classification.

We can observe that with 36 features, BMAX presented a classification rate of 28.2% and BOF 25.9%. Comparing

TABLE I
Classification rate for CIFAR10 database.

Regular Grid	Number of Detected features	Classification Rate BMAX	BOF
$ds = 6, S = 12$	36	$\mathbf{28.2 \pm 0.3}$	25.9 ± 0.4
$ds = 8, S = 16$	16	$\mathbf{27.7 \pm 0.3}$	24.9 ± 0.8
$ds = 16, S = 32$	4	$\mathbf{26.1 \pm 0.2}$	21.8 ± 0.8

TABLE II
Classification result for each class for CIFAR10 database.

	$\mathbf{ds = 6, S = 12}$		$\mathbf{ds = 16, S = 32}$	
Class	**BOF**	**BMAX**	**BOF**	**BMAX**
airplane	18.7	**19.2**	**23.7**	20.7
automobile	28.9	**33.9**	22.3	**28.9**
bird	**21.9**	17.7	**15.8**	13.0
cat	12.4	**17.7**	15.2	**17.8**
deer	**19.1**	16.9	16.4	**22.8**
dog	32.9	**39.7**	28.1	**29.0**
frog	**36.2**	23.3	21.0	**35.9**
horse	21.2	**24.7**	**22.7**	19.7
ship	36.7	**52.3**	30.4	**42.2**
truck	31.3	**39.0**	23.0	**30.3**
General	25.9	**28.4**	21.9	**26.0**

the results from the methods with 4 detected features, BMAX obtained a gain of 20% over BOF. Using only 4 features, BMAX has a small advantage over BOF using nine times the number of features, showing that our proposal can led to good results in situations with less features available.

The Table II presents the classification results for each class in CIFAR10. In the two presented scenarios, BMAX got a better recognition rate on seven categories when compared to BOF. This difference is more evident in ship and truck classes.

For this database and considering the images size, the goal is to evaluate the two methods in situations when few features are available. Even with small images, BMAX presented a smaller classification drop when the number of features are reduced when compared to BOF. Comparing 4 and 36 features, BMAX had a classification loss around 8%, while BOF lost around 19%. For tasks where few features can be gathered, like classification of small image segments for image parsing, BMAX could have better results compared to BOF.

B. Scene Category

This database is composed of fifteen scenes categories, thirteen provided by [14] and two by [19]. Each class has 200 to 400 images with average size of 300 x 250 pixels. Sample images from natural scene category are displayed in Fig. 6.

The Table III presents the classification rate for BMAX and BOF using the 15 scenes database. For this tests we employ

TABLE III
Classification rate for 15 scenes database.

Regular Grid	Number of Detected features	Classification Rate BMAX	BOF
$ds = 8, S = 16$	1200	65.5 ± 0.6	$\mathbf{65.9 \pm 0.6}$
$ds = 16, S = 32$	300	$\mathbf{64.0 \pm 0.5}$	60.3 ± 0.6
$ds = 32, S = 64$	80	$\mathbf{63.1 \pm 0.4}$	57.2 ± 0.9
$ds = 64, S = 128$	20	$\mathbf{62.7 \pm 0.3}$	54.7 ± 0.6
$ds = 128, S = 256$	6	$\mathbf{59.8 \pm 0.4}$	51.9 ± 1.0

Fig. 6. Sample images from 15 scenes database.

TABLE IV
RUN TIME FOR PROCESS A IMAGE OF SIZE 300X250 PIXELS WITH BMAX RANGING THE REGULAR GRID PARAMETERS FOR A VOCABULARY OF SIZE 400.

Regular Grid	Number of Detected features	Run time
$ds = 8$	1200	600 ms
$ds = 16$	300	276 ms
$ds = 32$	80	117 ms
$ds = 64$	20	35 ms
$ds = 128$	6	12 ms

100 images per category for training and the remaining for test. The vocabulary size is 400, the stated number of detected features is acquired using the average image size of 300 x 250. Using the same regular grid employed in [19], $ds = 8$, $S = 16$, the BOF got a small advantage over BMAX. The difference between our BOF implementation and the one presented in [19] is the kernel of SVM, they use a histogram intersection while we use a radial basis function. And so, different values of recognition rate for the same database can be verified.

When we started to decrease the number of features used, BMAX got advantage against BOF. Using 300 features, our proposal obtained a gain of 6% over BOF using the same amount of features. Using 20 features, BMAX has lost just 5% in the recognition rate compared to BOF using 1200 features. For tasks that demand fast response, this penalty can be acceptable considering the reduction of 60 times the number of features to be processed and matched with the vocabulary.

The Table IV shows the run time to process a image of size 300x250. This process extracts the local features, compare them with the vocabulary to compose the feature vector and classification with SVM. The classification with SVM takes less than 1 ms using a feature vector with 400 elements and 15 classes with one-against-all approach. In our tests, it took around 2 ms to classify the feature vector of 100 images. For larger feature vectors and problems with more categories, the SVM classification cost can be a limitation. This times are acquired using a 3.3 GHz CPU and a single thread, while the file read time is disregarded. Reducing from 1200 to 20 features, the run time is decreased around 17 times.

The Table V presents the classification result for each class

TABLE V
CLASSIFICATION RESULT FOR EACH CLASS FOR 15 SCENES DATABASE.

	$\mathbf{ds = 8,\ S = 16}$		$\mathbf{ds = 64,\ S = 128}$	
Class	**BOF**	**BMAX**	**BOF**	**BMAX**
bedroom	**46.6**	**46.6**	22.4	**37.9**
CALsuburb	**89.4**	85.9	83.1	**92.9**
industrial	**47.9**	39.6	**34.9**	32.7
kitchen	**50.9**	43.6	**36.4**	30.0
living room	55.6	**63.2**	**45.3**	39.2
MITcoast	**80.8**	80.0	73.8	**78.1**
MITforest	87.7	**88.6**	78.1	**82.0**
MIThighway	**83.1**	75.0	51.3	**70.6**
MITinsidecity	**56.7**	50.9	68.3	**76.9**
MITmountain	**82.1**	76.6	54.7	**66.8**
MITopencountry	58.4	**68.4**	49.0	**58.1**
MITstreet	**82.8**	80.2	69.8	**76.0**
MITtallbuilding	70.7	**74.2**	71.1	**79.3**
PARoffice	**68.7**	46.6	55.2	**58.3**
store	58.1	**65.7**	45.4	**52.1**
General	**67.9**	65.7	55.9	**62.1**

in 15 scenes database. We can observe the difference from the methods when the number of features is reduced. Using a regular grid of $ds = 8$, BOF has an advantage in 9, disadvantage in 5, and parity in one class. In the test using less features, $ds = 64$, BMAX has advantage in 12 of the 15 classes.

C. Caltech-101

The caltech 101 database [20] is composed of 102 classes, from 31 to 800 images per class. The objects are centered and occupy most of the image. The majority of the images have medium resolution, around 300 x 300 pixels. It is widely used as benchmark for image classification algorithms and known for its intra-class variance.

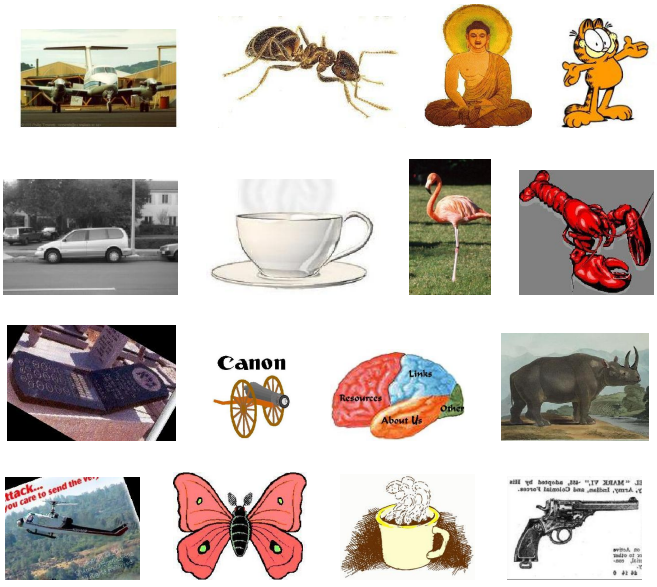

Fig. 7. Sample images from Caltech-101 database.

In our tests we used 30 images per class for training and the number of test images are limited to 70 for performance reason. For this database, we used a visual word vocabulary with

TABLE VI
CLASSIFICATION RATE FOR CALTECH 101 DATABASE.

Regular Grid	Number of Detected features	Classification Rate BMAX	BOF
$ds = 8$, $S = 16$	1450	46.6 ± 0.4	$\mathbf{47.7 \pm 0.6}$
$ds = 16$, $S = 32$	360	$\mathbf{49.2 \pm 0.7}$	45.1 ± 0.8
$ds = 32$, $S = 64$	100	$\mathbf{43.1 \pm 0.9}$	35.6 ± 1.2
$ds = 64$, $S = 128$	25	$\mathbf{35.5 \pm 0.5}$	25.9 ± 0.6
$ds = 128$, $S = 256$	9	$\mathbf{28.7 \pm 0.5}$	21.4 ± 0.6

TABLE VII
PUBLISHED CLASSIFICATION RESULTS FOR CALTECH 101 DATABASE.

Model	Recognition rate for 30 training images
BOF [19]	41
HMAX [10]	42
Holub *at al.* [22]	43
BOF (our implementation)	47
BMAX	49
Grauman & Darrell [23]	58
spatial pyramid matching [19]	**65**

800 words. Even though, [19] reports that they did not obtain any substantial benefit using more than 200 visual words in the vocabulary. Here, we observe considerable increase in the recognition rate using 800 words. Our test results are shown in the Table VI.

The results from altech 101 presented the same behavior of the 15 natural scenes database, BOF achieved a small advantage using the grid by [19] and BMAX obtained considerable advantage in the recognition rate increasing the space of the grid. Indeed, this is a most challenging database and both methods presented good results considering the decrease of number of local features.

Using 100 features, BMAX reached 43% of recognition rate, that is 21% better than BOF using the same amount of local features. Note that the classification result for BMAX was better using 360 features than using 1450 features, and also even better than BOF with 1450 features.

The Table VII presents some published results for caltech 101 database. In fact, we realize that our proposal BMAX is better than the two base models, BOF and HMAX.

The spatial pyramid matching [19] is a modification of standard BOF, that subdivide the image and compute the histogram of visual words for each part, concatenating them in order to create the feature vector. It encodes geometrical information with the histogram of visual words, and performs a considerable increase in the recognition rate. A similar approach can be used along with BMAX, and is one of our possible direction for further works.

V. DISCUSSION

This work has presented a new approach for image classification based on bag of features. The proposed approach uses the distance of the closest local feature for each visual word in the vocabulary, instead of occurrence count, to make the image feature vector. In the bag of max words, each local feature can be used for more than a visual word, unlike standard bag of features, and thereby aggregate more information to the image feature vector when few local feature are available.

We have conducted experiments in three different public images databases normally used for benchmark. The results showed that our approach can handle better situations where few features are available compared to standard bag of features. This characteristic enables our approach to derive a better cost benefit rate than standard BOF. In some cases, compared to BOF using 60 times more local features, that represents a raise of 17 times in the run time, BMAX has lost just 5% in the recognition rate.

In future works, we intend to test fastest feature extractors, like SURF [24] or ORB [25]. Currently, the slow feature extraction process of SIFT and its big descriptor vector are the main limitations to reduce the computational cost of our approach.

ACKNOWLEDGMENT

The authors are thankful to CAPES, Brazilian funding agency for the support to this work.

REFERENCES

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–511–I–518 vol.1.

[2] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 3642–3649.

[3] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 1058–1066.

[4] D. Claudiu Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep big simple neural nets excel on handwritten digit recognition," *arXiv preprint arXiv:1003.0358*, 2010.

[5] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[6] G. Csurka, C. Dance, J. Willamowski, L. Fan, and C. Bray, "Visual categorization with bags of keypoints," in *Proceedings of the Workshop on Statistical Learning in Computer Vision*, 2004, pp. 1–22.

[7] G. V. Pedrosa and A. J. Traina, "From bag-of-visual-words to bag-of-visual-phrases using n-grams," in *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI-Conference on*. IEEE, 2013, pp. 304–311.

[8] N. C. Batista, A. Lopes, and A. Albuquerque Araujo, "Detecting buildings in historical photographs using bag-of-keypoints," in *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing*, October 2009.

[9] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscience*, vol. 2, no. 11, pp. 1019–1025, 1999.

[10] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 3, pp. 411–426, March 2007.

[11] J. R. Uijlings, A. W. Smeulders, and R. J. Scha, "Real-time bag of words, approximately," in *Proceedings of the ACM international Conference on Image and Video Retrieval*. ACM, 2009, p. 6.

[12] D. Galvez-Lopez and J. D. Tardos, "Real-time loop detection with bags of binary words," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 51–58.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, June 2005, pp. 524–531 vol. 2.

[15] A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification via plsa," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 517–530.

[16] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[17] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[18] J. P. Jones and L. A. Palmer, "An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex," *Journal of neurophysiology*, vol. 58, no. 6, pp. 1233–1258, 1987.

[19] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 2169–2178.

[20] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, June 2004, pp. 178–178.

[21] A. Krizhevsky, "Learning multiple layers of features fromtiny images." Masters thesis, Computer Science Department, University of Toronto, 2009.

[22] A. Holub, M. Welling, and P. Perona, "Exploiting unlabelled data for hybrid object classification," *NIPS 2005 Workshop on Inter-Class Transfer*, 2005.

[23] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, Oct 2005, pp. 1458–1465 Vol. 2.

[24] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer vision–ECCV 2006*. Springer, 2006, pp. 404–417.

[25] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2564–2571.