Accurate Visual Word Construction using a Supervised Approach*

Basura Fernando[1], Elisa Fromont[2], Damien Muselet[2], Marc Sebban[2]

[1]CIMET, University Jean Monnet, F-42023, Saint Etienne France.
[2]Université de Lyon, F-42023, Saint-Etienne, France CNRS, UMR 5516, Laboratoire Hubert Curien, 42023, Saint-Etienne, France Université de Saint-Etienne, Jean Monnet, F-42023, Saint-Etienne, France.

Abstract

Most of the bag of visual words models are used to resorting to clustering techniques such as the K-means algorithm, to construct visual dictionaries. In order to improve their efficiency in the context of multi-class image classification tasks, we present in this paper a new incremental weighted average and gradient descent-based clustering algorithm which optimizes the visual word detection by the use of the class label of training examples. We show that this new supervised vector quantization allows us to better reveal concept or category-specific local feature distributions over the feature space. A large comparison with the standard K-means algorithm on the PASCAL VOC-2007 dataset is carried out. The results show that our visual word construction technique is much more suitable for learning efficient classifiers with Support Vector Machine and Random Forest algorithms.

Keywords: supervised vector quantization, bag of visual words, clustering

1 Introduction

Many of the object class recognition algorithms are based on aggregating the local visual information extracted from images to learn about object models and then to use them to classify images using supervised techniques. In this context, bag of visual words (BOVW) models have indisputably become a reference in scene classification [1]. In a BOVW model, local visual information is firstly extracted from training images. This often takes the form of salient local patches extracted from interest points, whose detection remains a crucial step. Recently, dense sampling strategy has been reported to be better than sparse sampling [2]. Then, a selection of descriptors is achieved to characterize each interest point (see [3] for an evaluation of color descriptors for object and scene classification). Once feature descriptors have been extracted from a training set of images, a clustering step is usually performed in order to obtain visual words or to create visual dictionaries. This step which creates the BOVW model is called vector quantization. Generally, cluster centers are considered as visual words. They are usually extracted by K-means based-algorithms [1, 3] even though other approaches have been applied, such as k-median clustering [4], mean-shift clustering [5], hierarchical K-means [6], agglomerative clustering [7], randomized trees [8], radius based-clustering [5, 9], or regular lattice-base strategies [10]. Thanks to the vector quantization, a given image can then be mapped into this new space of visual words leading to a bag of visual words, where each word can be weighted either according to its frequency or using more sophisticated techniques. A classifier can then be learned in this new vector space.

In this paper, we make a special focus on the crucial step of visual word determination. Cluster centers are supposed to reveal large variability of local image structures and capture parts that re-occur on many instances of similar concepts. Therefore, a good clustering approach should generate clusters that are discriminative enough to classify instances of various concepts with less ambiguity as well as general enough to represent an object model or part of it. We emphasize on accurate visual word representations that allow forming small yet powerful dictionaries that are discriminative and generalize well on large datasets.

It is important to note that much of the previously cited work on vector quantization uses unsupervised techniques. Therefore, clusters produced by unsupervised partitioning clustering tend to move towards dense areas and restrict the discriminative and generalization power of BOVW model. K-means and mean-sift-based clustering will keep

*This work is part of the ongoing ANR SATTIC 07-1_184534 research project.

adding feature vectors to highly populated centers even if the given feature vector is not semantically relevant in this cluster. For example, in an unsupervised clustering algorithm, a feature descriptor from a concept (say *cat*) may easily be assigned to a cluster whose majority represents a different concept (say a *dog*) due to moderate distance between cluster center and the feature descriptors. In these conventional clustering algorithms, there is no way to prevent semantically different feature descriptors from being assigned to the same cluster.

Only very few effort has been applied to use local statistical information to perform supervised visual word generation. For example, in [8], the authors present the Extremely Randomized Clustering Forests. This approach is first based on the supervised learning of small ensembles of trees which contain a lot of valuable information about locality in descriptor space. Then, ignoring the class labels, these trees are used as simple spatial partitioners that assign a distinct region label to each leaf. Semi-supervised clustering techniques [11] also take into account the labels but only part of the training sample is supposed to be labeled and this less complete information is used as strong constraints (*must-link* and *cannot-link* between data points) to create the clusters.

In this paper, we present another way to take into account class labels during the visual word construction process. The proposed method creates universal dictionaries based on supervised concept-specific local statistical data. We use conscious competitive learning with a supervised vector quantization that reveals concept or category specific local feature distributions with controlled cluster growing. Our algorithm is effective and very efficient in comparison with a standard K-means algorithm.

The rest of this paper is organized as follows: Section 2 is devoted to the presentation of our algorithm. In Section 3, we carry out a large experimental comparison on the PASCAL VOC-2007 dataset [12]. We conclude this paper in Section 4.

2 Proposed Method

Our proposed clustering algorithm tries to make use of available class labels during the visual word construction step. Let $C = \{C_1...C_n\}$ be the set of n classes or concepts in the image dataset. From each training image I, a set $X = \{X_1...X_m\}$ is extracted where X_k is a feature descriptor (for example a SIFT-128 feature descriptor) of local feature descriptors. Each feature descriptor X_k is assigned to a label C_j depending on the object class of the image from which the feature is generated. If there are multiple objects (which belong to multiple classes) in the image, the feature descriptors are extracted from bounding box images surrounding each object and having their own class label. Once this labeling process is done, the extracted feature descriptors are used in a supervised process to create the clusters. Each feature descriptor X_k ($X_k \in X$) mapped to a class label C_j ($C_j \in C$) is denoted by $X_k \rightarrow C_j$.

The common first step of our partitioning clustering algorithm consists in generating initial cluster representatives (see Algorithm 1). Our objective is to create a sufficient number (N) of clusters per class $C_j \in C$. These initial cluster representatives should be well separated from each other. We choose to create $N * n$ clusters in total. Let p_i^j be the i^{th} cluster representative of class C_j. P, the set of cluster representatives, is defined as follows:

$$P = \{p_i^j | 1 \leq i \leq N, 1 \leq j \leq n\} \quad (1)$$

Algorithm 1: Initial cluster representative selection
INPUT: Labeled data points set $X = \{X_1...X_m\}$
OUTPUT: Initial cluster representative set P.
METHOD:
1. Select randomly a class C_c and a first representative p_1^c s.t. $p_1^c \rightarrow C_c$; $P = \{p_1^c\}$
2. For i=1:N
3. For j=1:n | $j \neq c$ or $i \neq 1$
4. Select s such that $X_s \in (X - P)$ and $X_s \rightarrow C_j$ and

$$s \leftarrow argmax_{\{k|X_k \in (X-P) \wedge X_k \rightarrow C_j\}} \{min_{p \in P} D(p, X_k)\}$$

(*i.e.* X_s is the furthest point from all $p \in P$)
5. $p_i^j = X_s$, $P = P \cup \{p_i^j\}$
6. End
7. End

$D(p, X_k)$ is the Euclidean distance between the cluster representative p and the feature vector X_k. To illustrate this algorithm, consider the scenario presented in Fig.1. Suppose there are three classes ($n = 3$), namely *black, red* and *green* and 18 data points in total. Suppose we need to cluster this dataset with two cluster representatives per class ($N = 2$). Now suppose that the data point No.1 is randomly selected as the representative for the black cluster. Now we need to find a cluster representative for *green*. So we select the furthest green point from No.1 which is No.10. Then we need to find the red cluster representative. The furthest red data point from No.1 and No.10 is No.15. No.15 becomes the red cluster representative. The process continues to create a second cluster representative for each class. For *black*, the representative will be No.5, for *green* the representative will be No.8 and for *red* the representative will be No.18. So, the initial set of cluster

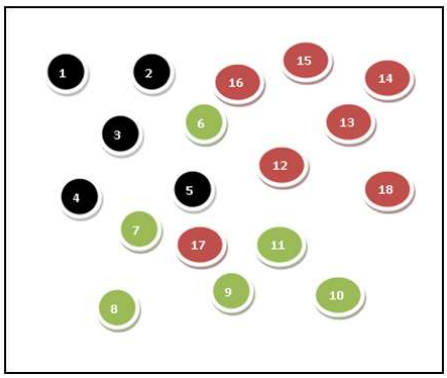

Figure 1: Dataset composed of 18 data points with black, green and red classes.

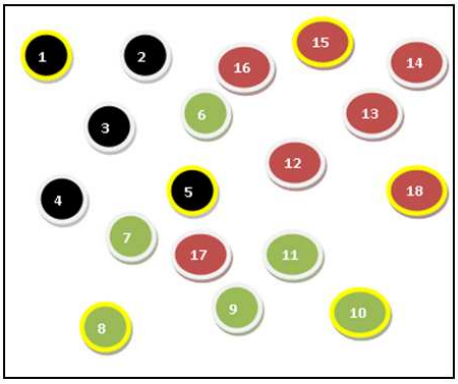

Figure 2: Initial cluster representatives at the end of initial centroids selection algorithm. Cluster representatives are shown in yellow color. Initial cluster representatives are well separated.

representatives is $\{1, 5, 8, 10, 15, 18\}$ where the set $\{1, 5\}$ represents the *black* class, $\{8, 10\}$ the *green* class and $\{15, 18\}$ the *red* one. As it can be seen from Fig. 2, this algorithm allows us to create well distributed and separated initial cluster representatives which belong to all the classes. This alleviates the usual strategy of partitioning algorithms such as K-means prone to initial random representative selection. Note that only the first representative has to be selected randomly by the user (step 1), the other representatives are selected by the algorithm. The complexity of this first selection step is $O(mTN^2n)$ where T is the total number of training images.

2.1 Supervised Clustering Step

After selecting the most interesting N first centroids for each class, we assign, as in the K-means algorithm, each point (*i.e.* a feature descriptor) of the training set to its nearest cluster. However, on the contrary of the K-means algorithm, (i) we take into account the labels of the training points both in the assignation phase and in the centroids computation phase and (ii) we recompute the centroids of the clusters incrementally after each assignment. The process is repeated until convergence of the centroids. Our new clustering algorithm is presented in Algorithm 2 whose core is the update rule of step 4. The current cluster representatives are updated iteratively using a supervised approach. For each cluster representative p_i^j, there is an associated weight counter f_i^j which represents in a way the density of this cluster. All f_i^j are initialized to one at the beginning of the clustering step. Each time a new instance is added to the cluster, the weight counter f_i^j is incremented by a value equal to $W_j = \frac{1}{|\{X_k | X_k \rightarrow C_j\}|}$ which represents the density of a feature descriptor belonging to class C_j. Note that c_i^j is the current size of the cluster whose center is p_i^j.

Algorithm 2: Extended Incremental K-means Clustering algorithm
INPUT:
(1) Set of data points excluding initial cluster representatives X = X - P
(2) Initial cluster representatives P from algo. (1)
OUTPUT:
Final cluster representatives set P
INITIALIZE:
$\forall q = 1 : N$, $\forall r = 1 : n$, $f_q^r = 1$, $c_q^r = 1$
t=0
Repeat
1. t=t+1
2. for each $X_k \in X$
3. $(i, j) \leftarrow argmin_{(q,r)|p_q^r \in P} D_f(p_q^r, X_k)$
4.
$p_i^j = \frac{p_i^j c_i^j + \eta X_k}{c_i^j + 1}$ if $X_k \rightarrow C_j$
$p_i^j = \frac{p_i^j c_i^j - (1-\eta)(X_k - p_i^j)}{c_i^j + 1}$ if $X_k \rightarrow C_q$ $(q \neq j)$
5. $c_i^j = c_i^j + 1$
6. $f_i^j = f_i^j + W_j$
7. End For
8. Until **Stopping Criteria** or $t = t_{max}$

The dissimilarity function $D_f(p_i^j, X_k)$ (used in step 3) which measures the dissimilarity between the cluster representative p_i^j and the feature descriptor X_k is given by eq. (2):

$$D_f(p_i^j, X_k) = \begin{cases} \|X_k - p_i^j\| \times f_i^j \times \alpha \text{ if } X_k \rightarrow C_j \\ \|X_k - p_i^j\| \times f_i^j \text{ if } X_k \rightarrow C_q,\ q \neq j, \end{cases} \quad (2)$$

where $\alpha \leq 1$. This parameter helps each feature descriptor to be assigned to a correct class cluster and we call it the semantic control parameter since it tries to prevent semantically different descriptors from being added to the same cluster. In other words, if X_k is at an about equal distance from 2 centers, using $\alpha < 1$ will lead to assign X_k to the center belonging to the same class. Note that fixing $\alpha = 1$ boils down to annihilating the supervision in the nearest-neighbor search process. The

dissimilarity measure $D_f(p_i^j, X_k)$ depends also on the weight counter f_i^j. If f_i^j is high for a particular cluster, then, the dissimilarity is comparatively higher for the pair (p_i^j, X_k). So, f_i^j controls the allocation of the feature descriptors to each cluster and controls the growth of the cluster population. Because of this condition, our algorithm will not allow data points being assigned to the same cluster over and over again. This allows a fair distribution of the feature descriptors assigned to each cluster and creates clusters which encapsulate local statistics over the feature space more accurately than most of other clustering algorithms. Since the class of the feature descriptors are taken into account during the clustering process, this algorithm will create more concept oriented clusters. Hence these clusters are good representatives of learned object class models and this helps to create more representative and informative feature vectors or bag of visual words.

In step 4, each cluster representative is updated based on the weighted mean of the cluster. The update rule makes sure that the mean vector p_i^j moves towards or away from the assigned data point X_k depending on the class j of p_i^j and on the own class of the data point X_k. Each time a new data point X_k is presented to Algorithm 2, all cluster representatives compete for this data point and the winner cluster representative is determined based on Eq. (2). η is a positive parameter $(0 < \eta < 1)$ which controls how much the winner cluster representative will move towards or move away from the assigned feature descriptor.

The stopping criterion for our clustering algorithm is given by eq. (3) as follows:

$$\forall\, p_i^j \in P \text{ if } \left\| p_i^{j^{(t)}} - p_i^{j^{(t+1)}} \right\| \leq \lambda \text{ then stop.} \quad (3)$$

According to eq. (3), if all cluster representatives move by a distance lower than λ in two successive iterations (t and $t+1$), the algorithm stops. In practice, the algorithm also stops when the number of iterations has reached the maximum value of t_{max}. Once the algorithm stops, the final cluster representatives P are used as visual words. The complexity of the learning step is $O(mNnDt_{max})$ where D is the dimensionality of the descriptors (*e.g.* 128 for SIFT descriptors [13]). The complexity of this clustering step is as low as for the simple K-means algorithm which is $O(mKDt_{max})$. For the K-means algorithm, K is the total number of clusters (in our case, $K = n \times N$). The next section describes the obtained experimental results.

3 Experimental Results

3.1 Experimental Protocol

We propose to assess the efficiency of our clustering algorithm in the context of object class recognition. Issues related to selecting a proper dataset in visual object class recognition problem is explained in [14]. In this experimental study, we use the PASCAL VOC-2007 dataset [12] which contains 9963 images, 5011 for training and 4952 for testing. There are 20 classes or concepts in the dataset. The objective is to classify each image of the test set to the appropriate class. In our experiments Harris-Laplace key point detector and dense sampling strategy have been used for feature vector construction with SIFT 128 descriptor. To construct visual dictionaries Harris-Laplace based keypoint detector with SIFT 128 feature descriptors are utilized. No dense sampling is used to create visual words. As pointed out in [3], orientation information is not significant for object class classification problem. Consequently, we neglect this information of key-points and set it to zero. 740,803 key points are utilized to create all visual dictionaries. A color descriptor software provided by [3, 15] is used to extract key-points and SIFT descriptors. A simple normalized term frequency [1] weighting scheme is used to construct bags of visual words. As done in [3], we use SVMs and Random Forest during the learning process. LibSVM [16] implementation is used with a kernel function which has been optimized in [3] for object class recognition problem. WEKA [17] implementation of Random Forests [18] with 10 random features and 20 trees is used. One difference from experiments done in [3] is that no spatial pyramids are used during any of our experiments. Several dictionaries with different word sizes are used to compare the performance of our algorithm. The precision and recall calculations come from the VOC 2009 competition. This competition also provides an algorithm to deduce the average precision which is the performance criterion we use in our experiments. For all our tests, the maximum number of iterations t_{max} is set to 1000 and λ is set to 0.5. We compare our algorithm with the K-means clustering algorithm which is the most common vector quantization method used in literature.

3.2 Parameter optimization

In our algorithm two parameters need to be optimized: α and η. Therefore, we propose to evaluate (using a validation set) the average precision of our clustering algorithm across different values of these two parameters. The results are plotted in Fig. 3. From this figure, we can see that the average precision is varying from 24% to 34% for

these ranges of values. Among all the tested values, $\alpha = 0.6$ and $\eta = 0.8$ provide the best results. For the following experiments, we have chosen these optimum values.

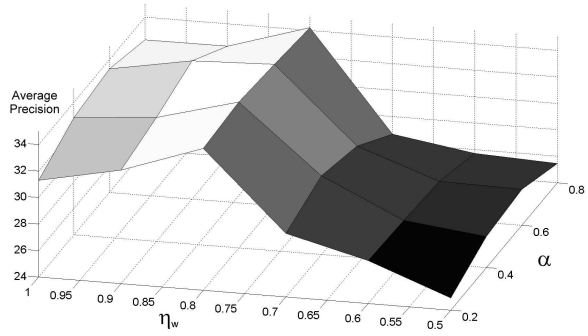

Figure 3: Average precision versus α and η.

3.3 Cluster features

The aim of our clustering algorithm is twofold. First, it controls the number of points in each cluster in such a way that all the points are well distributed among the clusters. Second, it creates clusters with high purity, *i.e.* so that the number of classes in each cluster is low. We propose to check these two cluster features in this section.

First, we propose to analyze and compare some statistics about the distribution of the descriptors over clusters formed by K-means and by our algorithm. The average, standard deviation, maximum and minimum number of descriptors assigned to each cluster are presented in tables 1 and 2 for different numbers of clusters. We have already underlined that the clusters produced by unsupervised partitioning clustering, such as the classical K-means tend to move toward dense areas. Indeed, we can see in table 1 that the standard-deviation and the difference between the maximum and the minimum numbers of data points in the clusters are very high. On the other hand, we can see in table 2 that the data points are better distributed among the clusters produced by our algorithm. And this is more obvious when the number of clusters increases. We will see in the next section that well distributed data points help to increase the discriminative power of the object recognition system.

Clusters	Avg.	Std.	Min	Max
200	3704	408	2217	6854
400	1852	243	922	4407
1000	741	133	282	3670
4000	185	48	6	1637

Table 1: Distribution of the data points among the clusters for the K-means algorithm.

Second, we propose to assess the purity of the clusters created by K-means or by our algorithm. Therefore, we present in tables 3 and 4 some statistics about the number of classes in each cluster. From table 3, we notice that the K-means clustering produces clusters with a high number of classes in each, i.e. clusters with semantically low purity. The problem of low-purity clusters is that a single visual word could represent many concepts which will not be discriminative enough during the learning step. In table 4, we can see that the clusters generated by our algorithm have a high purity comparing with those of K-means since the average numbers of classes per cluster are divided by almost 10 from K-means to our algorithm. Note that the average is not equal to 1 that means that the classes usually share some common features. Indeed, despite the fact our algorithm aims at reducing the impurity of the clusters, it allows us to keep some diversity to prevent the learning algorithm from learning by heart that would lead to an overfitting phenomenon. This potential to create category specific representatives is one of the advantages of our algorithm and we propose to assess the performance of our algorithm in term of average precision in the next section.

Clusters	Avg.	Std.	Min	Max
200	3704	323	2029	5839
400	1852	140	1013	3510
1000	741	21	405	1735
4000	185	11	101	672

Table 2: Distribution of the data points among the clusters for our algorithm.

Clusters	Avg.	Std.	Min	Max
200	20.00	0.000	10	20
400	19.99	0.086	19	20
1000	19.77	0.577	13	20
4000	15.79	2.969	1	20

Table 3: Distribution of the number of classes per cluster for the K-means algorithm.

Clusters	Avg.	Std.	Min	Max
200	2.70	1.616	1	9
400	2.22	1.147	1	11
1000	1.99	1.213	1	14
4000	1.23	0.887	1	15

Table 4: Distribution of the number of classes per cluster for our algorithm.

3.4 Classification results

Table 5 shows the average precisions provided by the two clustering methods for different sizes of visual word dictionaries and using either SVM classifier and Random Forest (RF) in the classification step. These results show that our algorithm significantly outperforms K-means algorithm for all dictionary sizes. The average precisions for

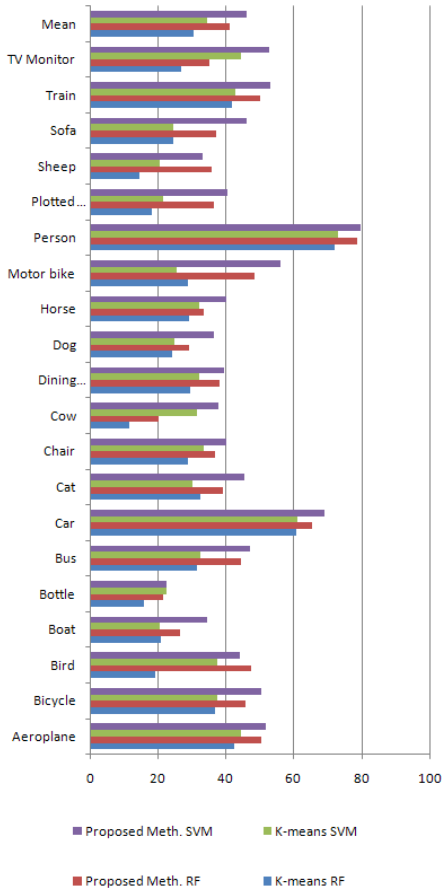

Figure 4: Average Precision × 100 for PASCAL dataset for our algorithm and K-means clustering using SVM or RF classifiers.

each class are presented in Fig. 4, for which 4000 visual words have been used. We notice that our algorithm performs well on almost all classes using both SVM classifier and Random Forest classifier.

Clusters	K-means SVM	K-means RF	SA SVM	SA RF
200	0.22	0.19	0.29	0.28
400	0.25	0.23	0.32	0.31
1000	0.27	0.26	0.35	0.33
4000	0.35	0.30	0.46	0.41
8000	0.35	0.32	0.47	0.43
Average	0.29	0.26	0.38	0.35

Table 5: Average Precision provided by K-means and our Supervised Algorithm (SA) with SVM or Random Forest (RF) classifiers.

4 Conclusion

In this paper, we introduced a novel visual word construction algorithm of low computational complexity. Our approach allows us to create well separated, evenly distributed, discriminative and concept specific clusters with controlled cluster growing. Our new gradient descent-based algorithm extends the standard K-means framework by using the class labels when assigning data points to clusters. Our algorithm leads to significant improvement over the standard K-means approach and we think that our new supervised vector quantization algorithm can be used to extend many other partitional clustering algorithms.

References

[1] J.Yang, Y. Jiang, A. Hauptmann, and C. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in Proceedings of the international workshop on Workshop on multimedia information retrieval, 2007, pp. 197–206.

[2] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in Proceedings of the European Conference on Computer Vision, 2006, pp. 490–503.

[3] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32(9), pp. 1582–1596, 2010.

[4] R. Cavet, S. Volmer, E. Leopold, J. Kindermann, and G. Paa, "Revealing the connoted visual code: a new approach to video classification," Computers & Graphics, vol. 28(3), pp. 361–369, 2004.

[5] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," in In Proceedings of the IEEE International Conference on Computer Vision, 2005.

[6] D. Nistr and H. Stewnius, "Scalable recognition with a vocabulary tree," in In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2161–2168.

[7] B. Leibe, K. Mikolajczyk, and B. Schiele, "Efficient clustering and matching for object class recognition," in In Proceedings of the British Machine Vision Conference, 2006.

[8] F. Moosmann, B. Triggs, and F. Jurie, "Randomized clustering forests for building fast and discriminative visual vocabularies," Neural Information Processing Systems, 2006.

[9] J. Gemert, C. Veenman, A. Smeulders, and J. Geusebroek, "Visual word ambiguity,"

IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32(7), pp. 1271–1283, 2010.

[10] T. Tuytelaars and C. Schmid, "Vector quantizing feature space with a regular lattice," in In Proceedings of the IEEE International Conference on Computer Vision, 2007.

[11] S. Basu, A. Banerjee, and R. Mooney, "Semi-supervised clustering by seeding," in In Proceedings of 19th International Conference on Machine Learning (ICML-2002, pp. 27–34.

[12] Pascal dataset web link. [Online]. Available: http://pascallin.ecs.soton.ac.uk/challenges/VOC/

[13] D. G. Lowe, "Object recognition from local scale-invariant features," in Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, vol. 2. IEEE Computer Society, August 1999, pp. 1150–1157 vol.2.

[14] J. Ponce, T. Berg, M. Everingham, D. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. Russell, A. Torralba, C. Williams, J. Zhang, and A. Zisserman, "Dataset issues in object recognition," Toward Category-Level Object Recognition, pp. 29–48, 2006.

[15] Color descriptor software from university of amsterdam. [Online]. Available: http://staff.science.uva.nl/~ksande/research/colordescriptors/

[16] Libsvm. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[17] Weka tool from university of waikato, new zealand. [Online]. Available: http://www.cs.waikato.ac.nz/~ml/weka/

[18] L. Breiman, "Random forests," Machine Learning, vol. 45(1), pp. 5–32.