

A ONE-PASS RESOURCE-ALLOCATING CODEBOOK FOR PATCH-BASED VISUAL OBJECT RECOGNITION

Amirthalingam Ramanan and Mahesan Niranjan

School of Electronics and Computer Science,
University of Southampton, Southampton, SO17 1BJ, UK.
{ar07r, mn}@ecs.soton.ac.uk

ABSTRACT

Frequencies of occurrence of low-level image features is the representation of choice in the design of state-of-the-art visual object recognition systems. A crucial step in this process is the construction of a codebook of visual features, which is usually done by cluster analysis of a large number of low-level image features detected as interest points. However, clustering is a process that retains regions of high density in a distribution and it follows that the resulting codebook need not have discriminant properties. Here we extend our recent work on constructing a one-pass discriminant codebook design procedure inspired by the resource allocating network model from the artificial neural networks literature. Unlike clustering, this approach retains data spread out more widely in the input space, thereby including rare low-level features in the codebook. It simultaneously achieves increased discrimination and a *drastic reduction* in the computational needs. We illustrate some properties of our method and compare it to a closely related approach.

1. INTRODUCTION

Each three-dimensional object in the real world can cast an infinite number of different two-dimensional images onto the retina. The important problem in computer vision is to determine the presence or absence of a specific object under a wide variety of conditions. Changes in pose, lighting, occlusion, clutter, intra-class differences, inner-class variances, deformations, as well as the background that varies relative to the viewer make the problem highly challenging.

The popular approach in visual object recognition is the use of local information extracted at several points or patches in the image. The underlying idea is that, in different images the distribution of the patches is different, which can be effectively used for recognition. In such a patch-based object recognition system, the key role of a visual codebook is to provide a way to map the low-level features (e.g. scale-invariant feature transform (SIFT) [1]) into a fixed-length vector in histogram space to which standard classifiers can be directly applied.

Several patch-based visual object recognition systems [2, 3, 4] fit into a general framework that is summarised in the following steps:

1. *Feature extraction*: Detection and description of image patches from the image corpus.
2. *Cluster analysis*: Constructing visual codebooks by means of clustering techniques. The codebook is the set of centres of the learnt clusters.
3. *Histogram generation*: Mapping the extracted image descriptors into a feature vector by computing the frequency histograms with the learnt clusters. This mapping produces a bag-of-features representation, similar to the bag-of-words representation that was originally used in text classification [5].
4. *Classification*: Classifying the test set to predict which category or categories to assign to the image.

This patch-based bag-of-features methodology has proved to yield state-of-the-art performance in large evaluations (e.g. the PASCAL Visual Object Classes (VOC) Challenges).

The classification performance of such a visual object recognition system strongly depends on the effectiveness of the visual codebook as it plays a central role that affects the model's complexity. Desirable properties of a codebook are compactness, low computational complexity, and high accuracy of subsequent categorisation. Discriminative power of a codebook determines the quality of the codebook model, whereas the size of a codebook controls the complexity of the model.

Usually the construction of a codebook is achieved by cluster analysis using K -means clustering [3, 4, 6, 7]. There are several known difficulties with the use of K -means clustering, including the choice of a suitable value for K , the computational cost of clustering when the dataset is large, and the convergence properties of the algorithm [8]. Clustering the visual keypoints using K -means, forces the relationship between different codes to be assigned to one of the K fixed clusters. This being a density preserving

algorithm formulates more clusters to high frequency areas of the descriptor space and less clusters for the remaining areas. There is no a priori reason to expect that cluster centres derived in this manner would form a discriminant codebook to separate object classes.

A common requirement amongst many existing clustering methods is that all pairwise distances between data must be computed in advance. This makes it computationally more expensive and difficult to cope with large scale data such as the Caltech¹ and PASCAL VOC Challenge² image datasets. The size of these image datasets nearly grows exponentially over the years. Table 1 shows the statistics of these benchmark datasets.

Table 1. Summary of the Caltech and PASCAL VOC Challenge datasets. There are actually 102 and 257 categories in the Caltech image datasets if the ‘clutter’ categories in each set are included. We provide data statistics of the PASCAL VOC Challenge datasets where there is an increase in object categories.

Dataset	Released	#Categories	#Images
Caltech-101	2003	102	9144
Caltech-256	2006	257	30607
PASCAL VOC’05	2005	4	1373
PASCAL VOC’06	2006	10	5304
PASCAL VOC’09	2009	20	13704

We refer to our algorithm as a resource-allocating codebook (RAC), that takes only one-pass through the entire data, inspired by the resource allocation network (RAN) algorithms [9, 10] developed in the artificial neural networks literature. The RAN can be interpreted from a function space approach to sequential learning. It either allocates a new unit, based on the novelty of a newly seen pattern, or adapts the network parameters by using the standard LMS gradient descent algorithm to fit that observation. Our goal is to construct a visual codebook by discarding visually similar keypoints at the nearest neighbours that of a fixed-radius hyperspheres. RAC carves the input space in a wider span than that would be found by K -means method.

The remainder of this paper is organised as follows. In section 2, we summarise various techniques that have been used in the literature in constructing a visual codebook for object categorisation. Section 3, presents the framework of the novel algorithm that we propose. Section 4, provides testing results demonstrating the better performance of RAC at a tiny fraction of computing cost. In section 5, we discuss various properties of RAC. Section 6 concludes our paper.

¹http://www.vision.caltech.edu/Image_Datasets/

²<http://pascallin.ecs.soton.ac.uk/challenges/VOC/>

2. PREVIOUS WORK

Several combinations of image patch detectors and descriptors, different features, matching strategies, various clustering methods and classification techniques have been proposed in the last few years for object recognition. The well-known framework in the literature uses the SIFT descriptors clustered using K -means algorithm, and images are encoded as a histogram of codewords as originally proposed in [2] and [7]. In [2], images are described by histograms over a selected size of codebook constructed by K -means clustering. The reported results were the clusters that gave them the lowest empirical risk in classification. They compare Naive Bayes and SVM classifiers in the learning task, and found that the SVM with linear kernel gives a significantly better performance.

Several other techniques have been proposed in constructing codebooks with different clustering techniques, such as hierarchical K -means [11], agglomerative clustering [12], randomised clustering forests [13], and Gaussian mixture models [14].

Recent studies [3, 15, 16, 17] have started to explore the construction of a codebook that considers both compactness and discriminative power. In [15], the method reduces the size of a codebook and enhances discriminative power by eliminating some visual codes using an entropy-based minimum description length criterion. However, there is a large amount of computation involved in constructing a codebook and the resized codebook is not optimally compact. In [3], they optimise codebooks by hierarchically merging codewords in a pair-wise manner using the information bottleneck principle from an initially constructed large codebook by K -means. Although the codebook is tailored, it would require fully retraining the framework on the arrival of new categories. In [16], codebook is constructed at a multi-resolution level using hierarchical clustering and then use a boosting feature selection method to select discriminant codewords. Although this method is effective, greater computation and the use of K -means make it difficult to cope with large scale data. In [17], the proposed method unifies a codebook generation with classifier training. Images are represented by means of visual bits associated with different categories. Similarly, in [18] an iterative non-redundant codebook construction process is done by means of a weighted voting scheme of the AdaBoost procedure integrated with classifier learning. The base codebook in [17] and [18] is constructed using K -means.

The RAC has strong similarities to the mean-shift based clustering technique of Jurie and Triggs [19]. They sub sample patches randomly from the feature set and allocate a new cluster centre for a fixed-radius hypersphere by running a mean-shift estimator. The mean-shift procedure is achieved by successively computing the mean-shift vector

of the sample keypoints and translating a Gaussian kernel on them. Subsequently, features that fall within the cluster are filtered out. This process is continued by monitoring the informativeness of the clusters or until a desired number of clusters is found. This method is computationally intensive in determining the cluster centroid by mean-shift iterations at each of the sub samples. The convergence of such a recursive mean-shift procedure greatly depends on the nearest stationary point of the underlying density function and its utility in detecting the modes of the density. This method requires sub sampling of visual keypoints with a regular grid and the selection of the bandwidth parameter. In contrast, our proposed approach that has a single threshold takes only one-pass through the entire data, making it computationally efficient. A specific experiment carried out with RAC for face recognition has been reported in [20].

3. METHODOLOGY

The RAC starts by arbitrarily assigning the first data item as an entry in the codebook. When a subsequent data item is processed, its minimum distance to all entries in the current codebook is computed, using an appropriate distance metric. If this distance is smaller than a predefined threshold, the current codebook is retained and no action is taken with respect to the processed data item. If the threshold is exceeded by the smallest distance to centroids, a new entry in the codebook is created by including the current data item as the additional entry. This process is continued until all data items are seen only once.

To make the RAC technique more informative to the order of presence of the data, we use an incremental update to shift the centroids by means of computing the weighted average of all its points. We define all visual descriptors as $X \in \mathbb{R}^d$ and the centres of the hyperspheres as $C \in \mathbb{R}^d$, where d is the dimension of the visual descriptors. We keep record of the weights (number of descriptors) of each hypersphere. Whenever a new descriptor falls into a hypersphere H_i then its centroid C_i is redefined by the weighted average of all its previous point and the new point. Thereafter the count of H_i is incremented by one. The pseudocode of this approach is given in Algorithm 1.

The RAC partitions the feature space into a set of overlapping hyperspheres when the distance metric used is the Euclidean norm. Local correlations between features could also be modelled in this framework by estimating covariance matrices associated with each vocabulary entry and using a Mahalanobis distance metric, similar to the sequential input space partitioning algorithm in [21], though for simplicity we restrict ourselves to Euclidean distance in this paper.

Algorithm 1 Resource-Allocating Codebook

Input: Visual descriptors (\mathbf{X}) and radius (r) of the hyperspheres.

Output: Visual codebook (\mathbf{C})

```

Step 1:  $C_1 \leftarrow X_1$  // initialise the codebook
        $n_1 \leftarrow 1$  // initialise the No. of descriptors in  $\mathbf{C}$ 
        $i \leftarrow 2$  // subsequent visual descriptor
        $j \leftarrow 1$  // size of the codebook
Step 2: Repeat steps 3 to 4 until  $i \leq \text{size}(\mathbf{X})$ 
Step 3: if  $\min \| \mathbf{X}_i - \mathbf{C}_j \|^2 > r^2$ 
       then create a new hypersphere of  $r$  such that,
        $j \leftarrow j + 1$ 
        $C_j \leftarrow X_i$ 
        $n_j \leftarrow 1$ 
       else  $\forall j$ , find the  $k^{th}$  hypersphere, such that,
        $\min \| \mathbf{X}_i - \mathbf{C}_j \|^2 \leq r^2$  and update its location
        $\mathbf{C}_k \leftarrow \frac{n_k \times C_k + X_i}{n_k + 1}$ 
        $n_k \leftarrow n_k + 1$ 
       endif
Step 4:  $i \leftarrow i + 1$ 
Step 5: return  $\mathbf{C}$ 

```

4. EXPERIMENTS AND RESULTS

We present our experimental results under two categories. In the first category, the performance of RAC is compared with K -means and mean-shift based method. We selected a set of binary classification tasks from the PASCAL VOC Challenge 2007 dataset³. In the second category, the performance of RAC is compared with K -means by performing a binary classification on a balanced set of the PASCAL07 dataset and a multi-class classification on the Xerox7 dataset [2].

In our first experiment, the selection of binary classes was based either on their appearance-based similarity (e.g. bicycle and motorbike) or on the number of images that are nearly equal from both of the object categories. We performed the classification task on the PASCAL07 for each of the twenty classes, predicting the presence or absence of an example of that class in the test image. In the second experiment, in order to perform the classification on a balanced set, we randomly selected 50 images per interest class and three images per other classes ($3 \times 19 = 57$). The multi-class classification was performed with 10-fold cross-validation.

Following the state-of-the-art approach [3, 17], we used SIFT features, which is a representation of keypoints extracted from an image in a 128 dimensional space. In all of our experiments performed with PASCAL07, the codebook construction involved the features extracted within the pro-

³<http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/>

Table 2. Comparison of three codebook generation methods tested on a selected binary classification tasks from the PASCAL VOC 2007 dataset. K -means (KM) with $K=1000$. Both mean-shift (MS) and RAC codebook sizes are slightly greater than 1000. The parameters of the mean-shift were $N=1000$ and $h=r=0.8$. The r of RAC was 0.8.

Objects	KM + SVM	MS + SVM	RAC + SVM
Aeroplane vs Car	0.7858 ± 0.0095	0.7805 ± 0.0049	0.7815 ± 0.0071
Bicycle vs Motorbike	0.7721 ± 0.0132	0.7715 ± 0.0024	0.7846 ± 0.0053
Bird vs Horse	0.7558 ± 0.0175	0.7579 ± 0.0175	0.7578 ± 0.0189
Boat vs TVmonitor	0.7536 ± 0.0181	0.7509 ± 0.0012	0.7584 ± 0.0012
Bottle vs Pottedplant	0.6817 ± 0.0034	0.6777 ± 0.0070	0.6832 ± 0.0024
Bus vs Train	0.7139 ± 0.0033	0.7191 ± 0.0069	0.7185 ± 0.0036
Chair vs Dog	0.8386 ± 0.0148	0.8218 ± 0.0048	0.8330 ± 0.0092
Cow vs Sheep	0.6755 ± 0.0152	0.6953 ± 0.0133	0.7056 ± 0.0105

vided bounding box information by ignoring objects marked as ‘difficult’. In Xerox7, features were extracted from the 9-folds of training set, and RAC method was applied on a randomly selected 15000 interest points from each class to construct a codebook.

In classification, the keypoints derived from an image is mapped into a histogram of nearest codeword of a codebook, and then support vector machines (SVMs) are applied. SVMs are quite naturally designed to perform classification in high dimensional spaces. Classification in this paper was performed using one-versus-all linear SVMs. For each problem, we estimate the regularisation parameter C of the binary SVMs by means of cross-validation on the training set.

In Table 2 and Table 3, we report the recognition results of three independent runs as the means of average precision (MAP) of our experiments performed with the PASCAL07 dataset. Each run is carried out by randomly shuffling the order of presence of the images to the clustering process. While we have included the standard deviation for completeness, we note that these are estimates of uncertainty for a very small number of trials.

In our experience, we were not able to extend K -means clustering on $1,696,797 \times \mathbb{R}^{128}$ SIFT descriptors that were extracted from the two dominant object classes (‘person’ and ‘car’) of PASCAL07 dataset, to construct the visual codebook owing to prohibitive execution times. Therefore, we used the RAC technique. The MAP of the Person vs Car classification was 0.8253 ± 0.0040 .

In Table 3, RAC is best for 15 out of 20 classes, whereas K -means is best for 5 out of 20 classes. We evaluated the experimental results based on the statistical test (1-tailed) and may conclude that RAC outperforms K -means at the level of significance $p = 0.0023$.

Table 4 shows the confusion matrix of the 10-fold cross validation performed on the Xerox7 dataset. Our overall error rate is 13.64% whereas the K -means method of [2] has an error rate of 15%. Also the error rate of faces in

Table 3. Recognition results as mean average precisions on a balanced subset of the PASCAL07 classification task. K -means with $K=1000$ and RAC with $r=0.8$.

Object	KM+SVM	RAC+SVM
Aeroplane	0.8643 ± 0.0215	0.8482 ± 0.0098
Bicycle	0.7185 ± 0.0166	0.7907 ± 0.0171
Bird	0.6528 ± 0.0095	0.6558 ± 0.0064
Boat	0.8015 ± 0.0185	0.7819 ± 0.0073
Bottle	0.7439 ± 0.0355	0.7172 ± 0.0355
Bus	0.6532 ± 0.0188	0.6594 ± 0.0182
Car	0.6933 ± 0.0355	0.7113 ± 0.0020
Cat	0.6615 ± 0.0219	0.7331 ± 0.0224
Chair	0.6416 ± 0.0127	0.6274 ± 0.0102
Cow	0.7084 ± 0.0245	0.8125 ± 0.0211
Diningtable	0.6790 ± 0.0130	0.6998 ± 0.0155
Dog	0.7432 ± 0.0176	0.7890 ± 0.0024
Horse	0.6830 ± 0.0190	0.7441 ± 0.0096
Motorbike	0.7345 ± 0.0073	0.7515 ± 0.0172
Person	0.6512 ± 0.0120	0.6894 ± 0.0083
Pottedplant	0.5915 ± 0.0013	0.6075 ± 0.0124
Sheep	0.7810 ± 0.0225	0.8259 ± 0.0200
Sofa	0.7373 ± 0.0156	0.7241 ± 0.0112
Train	0.7280 ± 0.0120	0.7582 ± 0.0282
TVmonitor	0.7377 ± 0.0150	0.7886 ± 0.0144
MAP	0.7103 ± 0.0175	0.7358 ± 0.0145

the Xerox7 dataset when we applied our approach is 1.52% whereas [2] has an error rate of 2%.

5. PROPERTIES OF RAC

5.1. The coverage of the feature space

To illustrate the difference between RAC and K -means clustering in partitioning the descriptor space, we consider four images each from two object categories, duck and horse (see Figure 1), that of the Amsterdam Library of Object

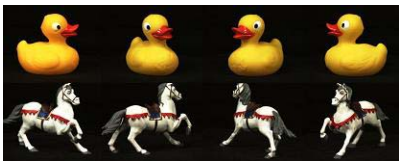


Fig. 1. Example images of the ALOI dataset

Images (ALOI) colour image collection⁴. The objects are subjective to pose, angle and illumination changes taken against a clear background. These 8 selected images result in $1559 \times \mathbb{R}^{128}$ SIFT descriptors that were clustered into 160 clusters using K -means and RAC techniques.

Figure 2 (a) shows a two dimensional projection of those 160 cluster centres projected on a plane defined by the first two principal components of the those clustered data. While projecting the SIFT based cluster centres from 128 to 2 dimensions masks much of the distribution, it can still be visualised that RAC gives codebook prototypes spanning in a wider range of space than K -means. Figure 2 (b) indicates the logarithmic count of the data points that fall in each bin of the clusters ordered descendingly. We used the duck and horse images shown in Figure 1. It can be seen that K -means has clusters with more equal data points that has a narrower span in the feature space, while RAC has more unequal points that spans more widely and capturing rare points in the feature space.

5.2. Computational savings

Our approach has far lower computational cost than K -means clustering. An illustrative example to support this claim would be the example of duck vs horse with the 1559 SIFT descriptors were clustered into 160 clusters using K -means in 19.79 seconds while RAC only needed 0.58 seconds to complete the one-pass execution on a desktop computer with an Intel Core 2 running at 2.4GHz and 4GB of RAM. Further, 105000 SIFT descriptors that were used in our multi-class classification of the Xerox7 dataset were clustered into 1000 clusters using K -means in an average time performing each fold of the 10-fold cross-validation in ≈ 149 hours while RAC only needed an average time of ≈ 19 minutes, showing the *drastic reduction* in the computational needs when using the RAC technique compared with the usage of the K -means clustering method. The time complexity of the latter method is $O(NDKm)$, where the symbols in parentheses represent number of data, dimensionality of features, the number of desired clusters and the number of iteration of the expectation-maximisation algorithm. The time complexity of RAC depends on the size of the candidate cluster set C , i.e., we compare each newly seen pattern to all existing clusters.

⁴<http://staff.science.uva.nl/~aloi/>

5.3. The hyper-parameter

The novelty threshold used in RAC is regarded as a hyper-parameter, and its choice has the same set of difficulties associated with the choice of K in K -means, and the size of sub samples N , radius r , and mean-shift radius h in the mean-shift based technique. Our approach to setting r is to take a small sample of the data, compute all pairwise distances between these samples and set the threshold, so that an approximate target codebook size is achieved. Another way to estimate r is cross-validating over the training set.

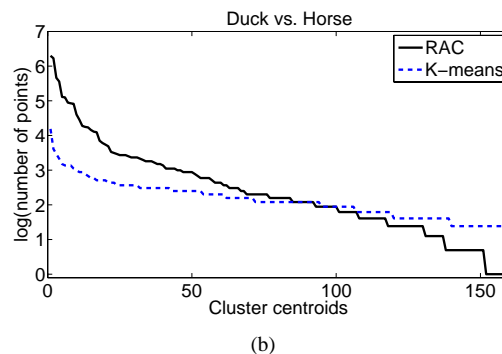
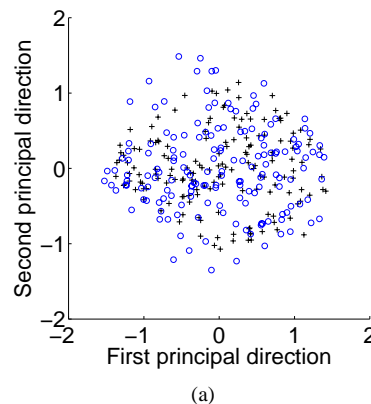


Fig. 2. (a) Projections of codewords along the first two principal components of the data. K -means ('+') yields cluster centres that preserves data density while RAC ('o') generates codebook that are more spaced out. (b) Plot of the logarithmic count of the data that falls in each histogram bin of the clusters generated by K -means (dotted line) and RAC (solid line) methods applied on duck vs horse task. We see that the K -means results in approximately the same number of data points associated with each cluster while RAC has clusters with a highly skewed distribution. Thus in RAC outlier (or less occurrence) data get included as part of the codebook.

Table 4. Recognition performance on the Xerox7 dataset. These results are very similar to (or slightly better than) [2] but achieved in a tiny fraction of computation time (see section 5.2).

True classes →	Bikes	Books	Buildings	Cars	Faces	Phones	Trees
Bikes	93.33	0.83	0	2.5	0	1.67	1.67
Books	0.71	72.86	7.14	5.0	6.43	7.14	0.71
Buildings	2.0	6.0	64.67	2.67	13.33	4.0	7.33
Cars	1.0	3.0	3.0	65.5	23.0	4.50	0
Faces	0.13	0.38	0.25	0.25	98.48	0.25	0.25
Phones	0.95	2.38	1.90	5.24	1.43	86.67	1.43
Trees	2.0	0	2.67	0	16.67	0	78.67

6. CONCLUSION

In this paper we have demonstrated a one-pass sequential strategy that can be used to construct visual codebooks with a computationally much simplified algorithm compared to what others have achieved. Indeed, the resource-allocation strategy we introduce slightly outperforms more traditional approaches due to its tendency to spread out the cluster centres over wider range of the feature space than density preserving clustering methods. In addition to computational and recognition performance, our approach is also fundamentally different from traditional approaches in that it is not the density of detected patches one needs to retain in the codebook but the coverage across the feature space.

Acknowledgment

AR is supported in parts by a grant from the University of Jaffna, Sri Lanka and the University of Southampton, UK.

7. REFERENCES

- [1] D. Lowe, “Distinctive Image Features from Scale-invariant Keypoints,” in *IJCV*, 2004, vol. 60, pp. 91–110.
- [2] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual Categorization with Bags of Keypoints,” in *Workshop on Statistical Learning in Computer Vision*, 2004, pp. 1–22.
- [3] J. Winn, A. Criminisi, and T. Minka, “Object Categorization by Learned Universal Visual Dictionary,” in *ICCV*, 2005, pp. 1800–1807.
- [4] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, “Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study,” in *IJCV*, 2007, pp. 213–238.
- [5] T. Joachims, “Text Categorization with Support Vector Machines: Learning with many relevant features,” in *ECML*, 1998, pp. 137–142.
- [6] L. Fei-Fei and P. Perona, “A Bayesian Hierarchical Model for Learning Natural Scene Categories,” in *CVPR*, 2005, pp. 524–531.
- [7] J. Sivic and A. Zisserman, “Video Google: A Text Retrieval Approach to Object Matching in Videos,” in *ICCV*, 2003, pp. 1470–1478.
- [8] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” in *ACM Computing Surveys*, 1999, pp. 264–323.
- [9] J. C. Platt, “A Resource-Allocating Network for Function Interpolation,” in *Neural Computation*, 1991, pp. 213–225.
- [10] V. Kadirkamanathan and M. Niranjan, “A Function Estimation Approach to Sequential Learning with Neural Networks,” in *Neural Computation*, 1993, vol. 5, pp. 954–975.
- [11] K. Mikolajczyk, B. Leibe, and B. Schiele, “Multiple Object Class Detection with a Generative Model,” in *CVPR*, 2006.
- [12] B. Leibe, A. Leonardis, and B. Schiele, “Robust Object Detection with Interleaved Categorization and Segmentation,” in *IJCV*, 2008, vol. 77, pp. 259–289.
- [13] F. Moosmann, B. Triggs, and F. Jurie, “Fast Discriminative Visual Codebooks using Randomized Clustering Forests,” in *NIPS*, 2007, pp. 985–992.
- [14] F. Perronnin, “Universal and Adapted Vocabularies for Generic Visual Categorization,” in *PAMI*, 2008.
- [15] S. Kim and I.-S. Kweon, “Object Categorization Robust to Surface Markings using Entropy-guided Codebook,” in *IEEE Workshop on Applications of Computer Vision*, 2007.
- [16] L. Wang, “Toward a Discriminative Codebook: Codeword Selection across Multi-resolution,” in *CVPR*, 2007, pp. 1–8.
- [17] L. Yang, R. Jin, R. Sukthankar, and F. Jurie, “Unifying Discriminative Visual Codebook Generation with Classifier Training for Object Category Recognition,” in *CVPR*, 2008, pp. 1–8.
- [18] W. Zhang, A. Surve, X. Fern, and T. Dietterich, “Learning Non-Redundant Codebooks for Classifying Complex Objects,” in *ICML*, 2009, pp. 1241–1248.
- [19] F. Jurie and B. Triggs, “Creating Efficient Codebooks for Visual Recognition,” in *ICCV*, 2005, vol. 1, pp. 604–610.
- [20] A. Ramanan and M. Niranjan, “Resource-Allocating Codebook for Patch-based Face Recognition,” in *ICHS*, 2009, pp. 268–271.
- [21] R. S. Shadafan and M. Niranjan, “A Dynamic Neural Network Architecture by Sequential Partitioning of the Input Space,” in *Neural Computation*, 1994, pp. 1202–1222.