Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 847608, 10 pages
http://dx.doi.org/10.1155/2014/847608

Research Article

Adaptive Aggregating Multiresolution Feature Coding for Image Classification

Honghong Liao,[1] Jinhai Xiang,[2] Weiping Sun,[1] and Shengsheng Yu[1]

[1] *School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China*
[2] *College of Informatics, Huazhong Agricultural University, Wuhan 430070, China*

Correspondence should be addressed to Weiping Sun; wpsun@hust.edu.cn

Received 12 July 2014; Accepted 3 September 2014; Published 6 November 2014

Academic Editor: Mohamed Djemai

The Bag of Visual Words (BoW) model is one of the most popular and effective image classification frameworks in the recent literature. The optimal formation of a visual vocabulary remains unclear, and the size of the vocabulary also affects the performance of image classification. Empirically, larger vocabulary leads to higher classification accuracy. However, larger vocabulary needs more memory and intensive computational resources. In this paper, we propose a multiresolution feature coding (MFC) framework via aggregating feature codings obtained from a set of small visual vocabularies with different sizes, where each vocabulary is obtained by a clustering algorithm, and different clustering algorithm discovers different aspect of image features. In MFC, feature codings from different visual vocabularies are aggregated adaptively by a modified Online Passive-Aggressive Algorithm under the histogram intersection kernel, which lead to a closed-form solution. Experiments demonstrate that the proposed method (1) obtains the same if not higher classification accuracy than the BoW model with a large visual vocabulary; and (2) needs much less memory and computational resources.

1. Introduction

Image classification is one of the most fundamental problems in computer vision and pattern recognition, which is to assign one or more category labels to an image. With the development of the Internet and multimedia technology nowadays, image classification has a wide range of applications, such as video surveillance, image and video retrieval, web content analysis, human-computer interaction, and biometrics, just to name a few. However, it is challenging on a number of fronts [1, 2]: (1) image classification performance would be affected by object viewpoint, illumination changes, partial occlusion, background clutter, and visual similarity between different classes; (2) large intraclass visual diversity and different instances of objects from the same category that exhibit significant variations in appearance also affect the performance; (3) in many cases appearance alone is ambiguous when considered in isolation, making it necessary to model not just the object class itself, but also its relationship to the scene context and priors on usual occurrences.

The Bag of Visual Words (BoW) image representation [3, 4], which is analogous to the bag-of-words representation of text documents [5] in terms of form and semantics, is one of the most popular and effective image classification framework in the recent literature. The essential idea behind this type of representation is to characterize an image by the histogram of its visual words, that is, vector-quantized local features. Popular candidates for these local features are local descriptors [6], such as SIFT [7] or SURF [8], that can be extracted as specific interest points, densely sampled over the image [9], or via a hybrid scheme called dense interest points [10]. Generally, sampling in a dense manner helps improve the image classification accuracy but requires more computational resources and storage usage.

The local descriptors have to be quantized, and there are very different clustering methods that can be used to obtain the vocabulary or dictionary. K-means and its variants [11, 12] are currently the most common methods, especially in large scale application. The visual "words" are the k cluster centers. Sparse coding methods [13, 14] have demonstrated the outstanding performance in image classification and object recognition; however they need to solve a ℓ_1-norm minimize optimization problem which requires solving

either an NP-hard problem or an alternative problem via costly iterative optimisation [15].

After obtaining the vocabulary, each local feature in an image is mapped to a "word" in order to represent any image as a histogram over the vocabulary. The BoW representation has been shown to characterize the images and objects within them in a robust yet descriptive manner, in spite of the fact that it ignores the spatial configuration between visual words [16]. Moreover, this approach has inspired a lot of research efforts, such as [17–19].

Notwithstanding its great success and wide adoption in BoW representation, the optimal formation of a visual vocabulary remains unclear; building one requires many choices on the part of the algorithm designer. Besides, the visual words will be affected by several factors, including the corpus of features used, the number of words selected, the quantization algorithm used, and the interest point or sampling mechanism chosen for feature extraction. Empirically, the larger the vocabulary, the more fine-gained the visual words, and the more discriminately the BoW histogram, leading to better performance. However, larger vocabulary needs more memory and intensive computational resources. Furthermore, with too large vocabularies, the quantization distances might be smaller than the fluctuations of visual descriptors under image distortions so that nearly identical fragments can be assigned to different visual words. While BoW model with small vocabulary usually lead to poor performance for its weak discrimination. As a result, there is a trade-off between the performance and computational resources required.

Visual vocabularies are usually constructed by using a single clustering algorithm (normally K-means algorithm). However, different clustering algorithms (or even the same clustering algorithm with different initialization) discover different aspects of image features; it is true that one particular quantization approach shall obtain a better solution than the others. If we were able to aggregate the BoW histograms from visual vocabularies constructed by a clustering algorithm with different initialization or even constructed by different clustering algorithms, we could integrate a generally more robust and more discriminative image representation. Furthermore, if these visual vocabularies are with small sizes, much memory usage and computational resources will be reduced compared to BoW model that usually needs a large visual vocabulary. How to reduce the computational resources and memory usage is a significant consideration in large scale image classification application.

In this paper, we propose a novel approach to aggregate the BoW histograms (feature codings) from different visual vocabularies in an adaptive manner. We first define the feature codings (BoW histogram) obtained from a set of visual vocabularies as multiresolution feature coding (MFC), which are for the abuse use of *Multiresolution Histograms* for image histogram in [20]. In MFC, feature codings from different visual vocabularies are weighted aggregated by modified Online Passive-Aggressive Algorithms [21, 22] under the histogram intersection kernel, which lead to a closed-form solution. Via our proposed approach we can achieve the state-of-the-art performance with a set of small vocabularies compared with other BoW based methods with a single large vocabulary, but with much lower memory usage and computational resources required.

The rest of this paper is organized as follows. Section 2 contains a review of the related work. We give the details of the proposed image representation approach in Section 3 and describe how to learn the weights for aggregating feature codings in Section 4. Experiments are described in Section 5, and Section 6 concludes this paper.

2. Related Work

Recognizing categories of objects and scenes is a fundamental human ability and an important, yet elusive, goal for computer vision research. One of the challenges is the semantic gap between the low-level image feature and high-level visual semantic [24]. Recently, more elaborated image representations, known as midlevel representations (i.e., richer representations of intermediate complexity), have been proposed to deal with the complexity of the image classification task, by aggregating hundreds and even thousands of low-level local descriptions about the image into a single feature vector.

The canonical midlevel model is the Bag of Visual Words (BoW) model. The basic BoW representation has important limitations; one of the notorious disadvantages of BoW is that it ignores the spatial relationships, which are very important in image representation. Several improvements have been suggested. To overcome the loss of spatial information, separate BoW can be computed in different subregions of the image, as in the spatial pyramid matching (SPM) scheme [17]. To attenuate the effect of coding errors induced by the descriptor space quantization, one can rely on soft assignment [18] or explicitly minimize reconstruction errors, such as sparse coding [13] and Local Linear Coding [19]. Finally, averaging local descriptor contributions (average pooling) can be reconsidered by studying alternative (more biologically plausible) pooling schemes, for example, max pooling [13].

The choice of dimension, selection, and weights of visual words (vocabulary) in BoW representation is crucial to the classification performance. Current prevailing dictionary learning approaches for BoW representation can be roughly categorized into three main types: universal learning, individual learning, and both. Lazebnik et al. [17] have learnt a universal vocabulary for spatial pyramid matching, and obtained promising performance for image classification. Because of diverse visual properties of images, such universal vocabulary may not be optimum for all the object classes and image concepts. With the increasing number of object classes and image concepts (which may have huge diversity of their visual properties), a universal vocabulary with larger size is needed to retain the performance. In [25, 26], a set of individual dictionaries have been learnt independently and have obtained excellent performance. Recently, Zhou and Fan [27] propose a new dictionary learning algorithm which explicitly separate the commonly shared visual words from the class-specific ones and jointly learn the common dictionary and interclass related dictionaries to enhance the discrimination.

Many works consider integrating multiple visual vocabularies to form a single one. In [28], the authors combine

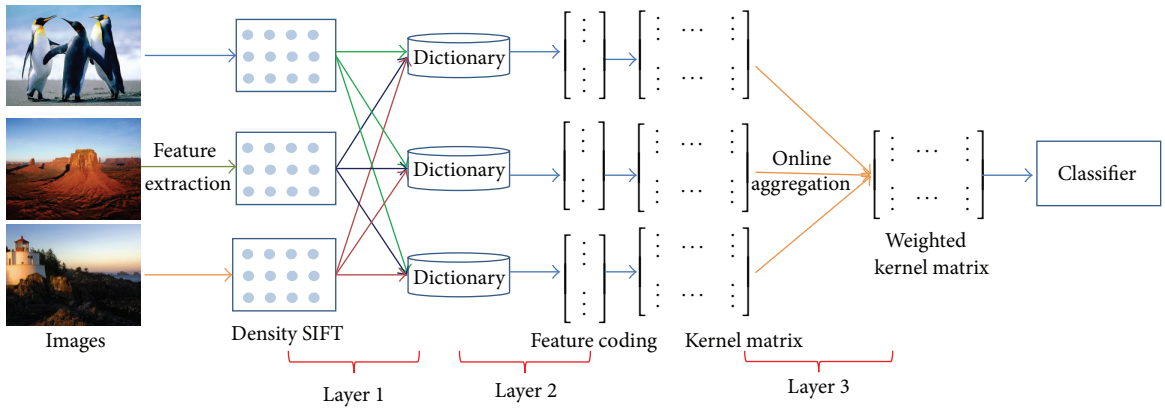

Figure 1: Overview of our proposed approach.

heterogeneous visual vocabularies via consensus clustering [29] and achieve a superior performance compared to traditional BoW approaches. Zheng et al. [30] propose a Bayes merging approach to multiple vocabularies for scalable image retrieval under the BoW model. Wang et al. [31] propose an algorithm to merge the visual words in a large vocabulary by maximally preserving class separability into a compact and discriminative one.

Khadem et al. [32] uses latent aspect models, such as LSA and pLSA, to embed visual words into a rich semantic space which named as concept space for action and scene recognition application. In their work, visual words are also weighted based on how they are present in the-same-category images, but their focus on the semantic relationship by constructing a discriminative and semantically meaningful vocabulary not on multiple visual vocabularies of smaller size. Śluzek [33] consider multiple features aggregated via Cartesian product to detect near-duplicate patches in images, and both visual and geometric characteristics of key point neighborhoods are represented by visual words.

The most related work to ours is Jégou and Chum [34], which emphasizes the benefit of PCA and feature whitening for image retrieval. In their work, the effects of merging vocabularies with different sizes on the performance of image retrieval by joint feature decorrelation are considered. Although image retrieval shares some techniques with image classification, their aims are different. Images that show the same scene with different objects may be considered similar in image retrieval but belong to different categories in image classification. At the same time, in [34] its main goal is to reduce the long feature vector to a small one, which is called small codes in [35]. The redundancy of feature vector should be reduced, while in image classification domain, these redundancies may be helpful. Regarding all the discussion above, the most important difference is the weights, which are fixed in their work and are learnt adaptively by a modified Online Passive-Aggressive Algorithms in ours.

3. The Proposed Method

Our proposed method can be divided into three layers. In the first layer, local image descriptors are extracted to generate a set of small visual dictionaries with different sizes; this is different from the traditional BoW model in which a single but usually very large dictionary is constructed. The second layer is the feature coding layer; each image is represented as a number of low-dimensional histograms, yielding multiple histograms with different resolution, which we name multiresolution feature coding. We use the hard-assignment (hard coding) as our default feature coding method in this paper, although it is to be easily extended to soft coding or sparse coding. After that, a set of kernels (we use the histogram intersection kernel in this paper, and χ^2 kernel can also be taken) are constructed by these low-dimensional histograms. In the third layer, these kernels are weighted aggregated to form a single kernel by a modified Online Passive-Aggressive Algorithms, which lead to a closed-form solution. Based on this image representation, image classification task can be easily done via using a simple classifier with kernel, such as support vector machines (SVM). We summarize our proposed image representation approach in Figure 1.

3.1. The First Layer: Dictionaries Generation. We extract dense-SIFT features in image I. For example, for every 3 pixels in row and column, we extract a 16×16 image patch. To describe every image patch, we build a histogram based on gradient of pixels following the way in [7] and denote the histograms by $I = \{x_1; \ldots; x_N\}$. These dense-SIFT features capture all local cues in image. To deal with variation of object scale, we can extract image patches in multiple scales through Gaussian smoothing with different covariance matrices.

After extracting the dense-SIFT features, the visual vocabularies are generated by k-means clustering algorithm. We take k-means algorithm into our consideration for it has several advantages [36]: (1) its time and storage complexity are both linear with respect to the data points; (2) it is guaranteed to converge at a quadratic rate; (3) it is invariant to data ordering. The time and storage complexity is the most fundamental factor that needs to be taken into consideration in image classification because thousands to millions of images are involved in the benchmark datasets or in the large scale application circumstances.

For dictionaries with different sizes, we can run the K-means algorithm multiple times, in each time, we set a different value of the number clusters K; and if the dictionaries with same size needed, we output the intermediate results of K-means clustering at different stages in the clustering process. In this paper, we construct dictionaries by running K-means algorithm multiple times with different values of K.

After this layer, a set of dictionaries is generated by K-means algorithm with different values of K; we call these dictionaries with different sizes as *multiresolution* visual dictionary for the abuse use of *Multiresolution Histograms* [20] for image histogram.

3.2. The Second Layer: Feature Coding. Feature coding plays a key role in image classification task; study in [37] has shown that, in image classification, feature coding strategy (vector quantization) is more important than learning a more discriminative dictionary (even the dictionaries are learnt by sparse coding) in large scale application. Although it is considered under the BoW model in this paper, feature coding is still a key factor.

Hard coding [3] assigns each local feature to its closest visual "word" in the visual dictionary, and the coding representation of a local feature x is

$$v(i) = \begin{cases} 1, & \text{if } i = \arg\min_j \left\| x - b_j \right\|_2 \\ 0, & \text{otherwise,} \end{cases} \qquad i = 1, 2, \ldots, M, \tag{1}$$

where b_j is the jth "word" in visual dictionary and M is the size of visual dictionary. After mapping each local feature to its closest visual "word," sum-pooling is used to add all the hard coding vectors to form a single histogram, where each bin reflects the occurrence frequency of visual words. The BoW histogram of an image is represented by normalizing the above histogram to be 1 by ℓ_2-norm.

The BoW model ignores the spatial relationships, which are very important in image representation. Spatial pyramid matching (SPM) [17], to some extent, makes up for the loss of spatial information and has made a remarkable success on a range of image classification benchmarks. The SPM strategy partitions an image into $2^\ell \times 2^\ell$ subregions in different scales $\ell = 0, 1, 2$, computes the BoW histograms within each of the 21 subregions, and finally concatenates all the histograms to form a vector representation of the image. In case where only the scale $\ell = 0$ is used, SPM reduces to traditional BoW model. We take the SPM strategy in our feature coding layer.

Suppose $\{\mathbf{h}^t\}_{t=1}^d$ is the histogram of an image set, where d is the number of dictionaries constructed in the first layer and $\{\mathbf{h}_i^t\}_{t=1}^d$ and $\{\mathbf{h}_j^t\}_{t=1}^d$ are the BoW histograms of image I_i and image I_j. We construct the histogram intersection kernel based on the BoW histogram obtained by the tth dictionary as

$$\mathbf{K}^t(i, j) = \sum_{k=1}^{m_t} \min\left(h_{ik}^t, h_{jk}^t\right), \tag{2}$$

where m_t is the size of the tth dictionary. Any other kernels based on histogram, such as the χ^2 kernel, can also be constructed in a similar manner.

As discussed above, the output of the feature coding layer can be either the BoW histograms $\{\mathbf{h}^t\}_{t=1}^d$ or the constructed kernels $\mathscr{K} = \{\mathbf{K}^t\}_{t=1}^d$, where d is the number of dictionaries. We call this representation as multiresolution feature coding. In this paper, the constructed kernels are used as the output for this layer.

3.3. The Third Layer: Weighted Aggregation. As discussed in latest subsection, a set of constructed kernels $\mathscr{K} = \{\mathbf{K}^t\}_{t=1}^d$ have been obtained. As different clustering algorithms discover different aspects of image features, each constructed kernel makes different contributions to the image classification task at hand, which leads to a weighted aggregation for the constructed kernels $\mathscr{K} = \{\mathbf{K}^t\}_{t=1}^d$:

$$\mathbf{K}_\eta(i, j) = \sum_{t=1}^{d} \eta_t \mathbf{K}^t(i, j) = \sum_{t=1}^{d} \eta_t \sum_{k=1}^{m_t} \min\left(h_{ik}, h_{jk}\right), \tag{3}$$

where $\sum_{t=1}^d \eta_t = 1$, d is the number of dictionaries and m_t is the size of the tth dictionary. $\mathbf{K}_\eta(i, j)$ is a similarity metric according to the definition in [38].

Our aims are to make sure the weighted similarity (3) of same labeled images is larger than that of different labeled images. Ideally, the learnt weight vector $\eta \in \mathscr{R}^d$ satisfies the constraint:

$$\mathbf{K}_\eta(i, p) \geq \mathbf{K}_\eta(i, n), \quad \forall (i, p, n) \in \Gamma; \tag{4}$$

that is

$$\sum_{t=1}^{d} \eta_t \mathbf{K}^t(i, p) \geq \sum_{t=1}^{d} \eta_t \mathbf{K}^t(i, n), \quad \forall (i, p, n) \in \Gamma, \tag{5}$$

where $\Gamma = \{(i, p, n) \mid y_i = y_p, y_i \neq y_n\}$ and y_i denotes the class label for image I_i.

It is impossible to fulfil these constraints for all triplets simultaneously. Hence, slack variables $\xi_{ipn} \geq 0$ are introduced, which is used to allow a small portion of images to violate the conditions, resulting in a soft constraint:

$$\mathbf{K}_\eta(i, p) - \mathbf{K}_\eta(i, n) \geq 1 - \xi_{ipn}, \quad \forall (i, p, n) \in \Gamma. \tag{6}$$

The ξ_{ipn} in (6) can be thought as a penalty or a loss for triplets violating the hard-constraints (4). Under the soft constraint (6), our aim is minimizing the loss over all triplets, which can be formulated as $\sum_{i,p,n} \xi_{ipn}$. So, similar to the formulation in SVM, we formulate the above loss function and its constraints as a quadratic optimization problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\eta\|^2 + C \sum_{i,p,n} \xi_{ipn} \\ \text{subject to} \quad & \begin{cases} \mathbf{K}_\eta(i, p) - \mathbf{K}_\eta(i, n) \geq 1 - \xi_{ipn}, \\ \qquad\qquad \forall (i, p, n) \in \Gamma, \\ \sum_{\ell=1}^{d} \eta_\ell = 1, \qquad \xi_{ipn} \geq 0, \ \forall (i, p, n) \in \Gamma, \\ \Gamma = \left\{(i, p, n) \mid y_i = y_p, y_i \neq y_n\right\}, \end{cases} \end{aligned} \tag{7}$$

where $\|\eta\|^2$ is the regularization term, which is used to avoid the overfitting of the loss function, and C is a trade-off between the loss function and the regularization term.

The quadratic optimization problem (7) can be solved directly by mathematical softwares, such as LIBSVM [39] and MOSEK ApS [40]. However, the quadratic optimization problem (7) has to pay a computational complexity $O(n^3)$, where n is the number of images. This is not appropriate for image classification applications when n is large. In the next section, we give a modified Online Passive-Aggressive Algorithms to learn the weights in an online manner.

4. Learning the Weights

To solve the quadratic optimization problem (7) efficiently with some standard optimization software, especially when the number of images or categories is large, and how to optimize the speed and memory usage have to be addressed. We tackle this issue by using the Online Passive-Aggressive Algorithms, which were proposed in [21]. The family of online Passive-Aggressive (PA) learning is formulated to trade off the objective of minimizing the distance between the learnt classifier and the previous classifier, and the objective of minimizing the loss of the learnt classifier suffered on the current instance. We define the following hinge loss function for the triplet (i, p, n):

$$L(\eta) = \max\left(0, 1 - \mathbf{K}_\eta(i,p) + \mathbf{K}_\eta(i,n)\right). \tag{8}$$

Our goal is to minimize the global loss $\sum_{(i,p,n)\in\Gamma} L(\eta)$ over all possible triplets in the training set. However, as discussed above, solving problem (7) is intractable when the number of images is large. Considering the complexity of exact solution, we give an approximate solution, by minimizing $L(\eta)$ for each triplet $(i, p, n) \in \Gamma$ instead of the global loss $\sum_{(i,p,n)\in\Gamma} L(\eta)$. In order to minimize this loss, we apply the Online Passive-Aggressive Algorithm iteratively over triplets to optimize η. Similar to the work in [22], we can rewrite (7) as

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\left\|\eta - \eta^{i-1}\right\|^2 + C\xi_{ipn} \\ &\text{subject to} \quad \begin{cases} L(\eta) \le \xi_{ipn}, \\ \xi_{ipn} \ge 0. \end{cases} \qquad (i,p,n) \in \Gamma. \end{aligned} \tag{9}$$

Therefore, at each iteration i, η^i is selected to optimize a trade-off between remaining close to the previous parameters η^{i-1} and minimizing the loss on the current triplet $L(\eta)$.

We follow Crammer et al. [21] to solve the problem in (9). When $L(\eta) = 0$, it is clear that $\eta = \eta^{i-1}$ satisfies the constrains directly. Otherwise, the Lagrangian is defined as

$$\begin{aligned} \mathscr{L}\left(\eta, \tau, \xi_{ipn}, \lambda\right) = {} & \frac{1}{2}\left\|\eta - \eta^{i-1}\right\|^2 + C\xi_{ipn} \\ & + \tau\left(1 - \mathbf{K}_\eta(i,p) + \mathbf{K}_\eta(i,n) - \xi_{ipn}\right) \\ & - \lambda\xi_{ipn}, \end{aligned} \tag{10}$$

where $\tau \ge 0$ and $\lambda \ge 0$ are Lagrange multipliers. The optimal solution is such that the gradient vanishes $(\partial\mathscr{L}(\eta, \tau, \xi_{ipn}, \lambda))/\partial\eta = 0$; hence

$$\frac{\partial\mathscr{L}\left(\eta, \tau, \xi_{ipn}, \lambda\right)}{\partial\eta} = \eta - \eta^{i-1} - \tau\left(\kappa(i,p) - \kappa(i,n)\right) = 0, \tag{11}$$

where $\kappa(i, p) = [\mathbf{K}^1(i,p), \ldots, \mathbf{K}^d(i,p)]^T \in R^d$ is a vector of corresponding elements of kernels.

The optimal new η is therefore

$$\eta = \eta^{i-1} - \tau\left(\kappa(i,p) - \kappa(i,n)\right). \tag{12}$$

We still need to estimate τ. Differentiating the Lagrangian with respect to ξ_{ipn} and setting it to zero also yield

$$\frac{\partial\mathscr{L}\left(\eta, \tau, \xi_{ipn}, \lambda\right)}{\partial\xi_{ipn}} = C - \tau - \lambda = 0, \tag{13}$$

which, knowing that $\lambda \ge 0$, means that $\tau \le C$. Plugging (12) and (13) back into the Lagrangian (10), we obtain

$$\begin{aligned} \mathscr{L}(\tau) = {} & -\frac{1}{2}\tau^2\left\|\kappa(i,p) - \kappa(i,n)\right\|^2 \\ & + \tau\left(1 - \left(\mathbf{K}_{\eta^{i-1}}(i,p) - \mathbf{K}_{\eta^{i-1}}(i,n)\right)\right). \end{aligned} \tag{14}$$

Taking the derivative of the second Lagrangian (14) with respect to τ and setting it to zero, we have

$$\begin{aligned} \frac{\partial\mathscr{L}(\tau)}{\partial\tau} = {} & -\tau\left\|\kappa(i,p) - \kappa(i,n)\right\|^2 \\ & + \left(1 - \left(\mathbf{K}_{\eta^{i-1}}(i,p) - \mathbf{K}_{\eta^{i-1}}(i,n)\right)\right) = 0, \end{aligned} \tag{15}$$

which yields

$$\tau = \frac{1 - \left(\mathbf{K}_{\eta^{i-1}}(i,p) - \mathbf{K}_{\eta^{i-1}}(i,n)\right)}{\left\|\kappa(i,p) - \kappa(i,n)\right\|^2} = \frac{L\left(\eta^{i-1}\right)}{\left\|\kappa(i,p) - \kappa(i,n)\right\|^2}. \tag{16}$$

Finally, since $\tau \le C$, we obtain

$$\tau = \min\left\{C, \frac{L\left(\eta^{i-1}\right)}{\left\|\kappa(i,p) - \kappa(i,n)\right\|^2}\right\}. \tag{17}$$

We summarize the discussion above and demonstrate the algorithm of optimizing (9) as follows (Algorithm 1). It should be noted that Algorithm 1 gives a closed-form solution for updating the weights, which is very efficient for image classification application as shown in our experiments.

5. Experiments

In this section, we evaluate our proposed approach for image classification on two public datasets: the Caltech-101 [41] and The Scene-15 dataset [17]. As for feature extraction for all

Initialize $\eta = [1, \ldots, 1]^T$.
repeat
(1) Sample a triplet (i, p, n).
(2) Update $\eta = \eta^{i-1} - \tau_i (\kappa(i, p) - \kappa(i, n))$,
where $\tau_i = \min\left\{C, L\left(\eta^{i-1}\right) / \left\|\kappa(i, p) - \kappa(i, n)\right\|^2\right\}$, and $\kappa(i, p) = \left[\mathbf{K}^1(i, p), \ldots, \mathbf{K}^d(i, p)\right]^T$.
Until stopping criterion is satisfied.

Algorithm 1: Learning weights via Online Passive-Aggressive Algorithim.

Table 1: Classification accuracy (%) comparison on Caltech-101.

Algorithms	15 training	30 training
BoW (400)	58.40 ± 0.34	72.02 ± 0.26
BoW (1000)	58.57 ± 0.35	70.11 ± 0.33
BoW (4000)	61.12 ± 0.33	71.24 ± 0.26
SPM [17]	56.4	64.4 ± 0.8
Soft-coding [18]	—	64.1 ± 1.2
LLC [23]	—	71.25 ± 0.98
Ours	**61.90** ± 0.40	**72.78** ± 0.32

The bold font refers to the highest score of the compared methods.

Table 2: Classification accuracy (%) comparison on Scene-15.

Algorithms	Classification rate (%)
BoW (400)	81.31 ± 0.43
BoW (1000)	80.91 ± 0.38
BoW (4000)	**82.53** ± 0.34
SPM [17]	81.4 ± 0.49
Soft-coding [18]	76.67 ± 0.39
LLC [23]	81.09 ± 0.43
Ours	82.33 ± 0.64

The bold font refers to the highest score of the compared methods.

datasets, we use the *VLFeat* [42] toolbox and LIBSVM [39] for training and testing the SVM classifier. We densely compute SIFT descriptors on overlapping 16×16 pixels with a step of 3 pixels. All images are processed in gray scale, and SPM [17] with $2^\ell \times 2^\ell$, $\ell = 0, 1, 2$, are used for each image. The number of dictionaries (Section 3.1) is set to 3; that is, for each dataset, we generate three small dictionaries with different sizes, and the sizes of these dictionaries are set to 50, 100, and 150. The number of triplets sampled to update the weights is set to 10^5. The size of dictionary in traditional BoW model [3, 4] with SPM [17] is set to 400, 1000, and 4000 for comparison. For fair comparison, all datasets are repeated for ten times and the *mean* and the *variance* are collected. We also compared our approach with original SPM method [17], soft-coding method [18], and LLC method [23], and results of these three methods are quoted directly from their papers.

5.1. Caltech-101 Dataset. The Caltech-101 dataset contains 101 classes (not including the *background* class) with high intraclass appearance shape variability. The number of images per category varies from 31 to 800 images and most of these images are medium resolution, that is, 300×300 pixels. The total number of images are 8,677. We randomly choose 15 and 30 images per category for training and the rest for testing. Detailed comparison results are shown in Table 1. As shown, our proposed method achieves the highest performance. Compared to the BoW method with a dictionary of size 4000, which achieves the classification rate of 71.24 percent for 30 training per category and 61.12 percent for 15 training per category, our proposed method just needs 50 + 100 + 150 = 300 visual "words" and achieves 72.78 percent for 30 training per category and 61.90 percent for 15 training per category. The memory has reduced more than 13 times, and computational complexity also reduces as shown in Table 1.

For BoW model, as shown, with the increase in the size of dictionary, the classification accuracy also increases. However, when the dictionary is with size 400, its performance is higher than that of dictionary with sizes 1000 and 4000 for 30 training per category; this may be due to the random choice of the initialization for K-means algorithm when we obtain the dictionaries.

Figure 2 shows a comparison of the average confusion matrix in the Caltech-101 dataset for 30 training per category. Figure 2(a) denotes the average confusion matrix by our proposed method, Figure 2(b) denotes the average confusion matrix by BoW model with dictionary size set to 4000, and Figure 2(c) shows the residual of confusion matrix in Figure 2(a) minus that in Figure 2(b). As shown in Figure 2(c), the number of elements on the diagonal that are larger than zero is 52 and their values are summed up to 1.5, which implies that our method classifies the images more accurately than BoW model with dictionary size set to 4000 in most image categories.

5.2. Scene-15 Dataset. The major sources of pictures in the Scene-15 dataset include the COREL collection, personal photographs, and Google Image Search. Each category has 200 to 400 images with the average image size of 300×250 pixels. The total image number is 4,485. The 15 scene categories are *CALsuburb, kitchen, living room, bedroom, store, industrial, MITcoast, MITforest, MIThighway, MITinsidecity, MITmountain, MITopencountry, MITstreet, MITtallbuilding*, and *PARoffice*. We followed the common experiment setup for Scene-15 dataset and randomly chose 100 images per category for training and the rest for testing.

Table 2 shows the comparison results in detail. In Scene-15 dataset, BoW model with dictionary size set to 4000 achieves the highest classification accuracy 82.53%, while our proposed method obtains the second highest score 82.33%.

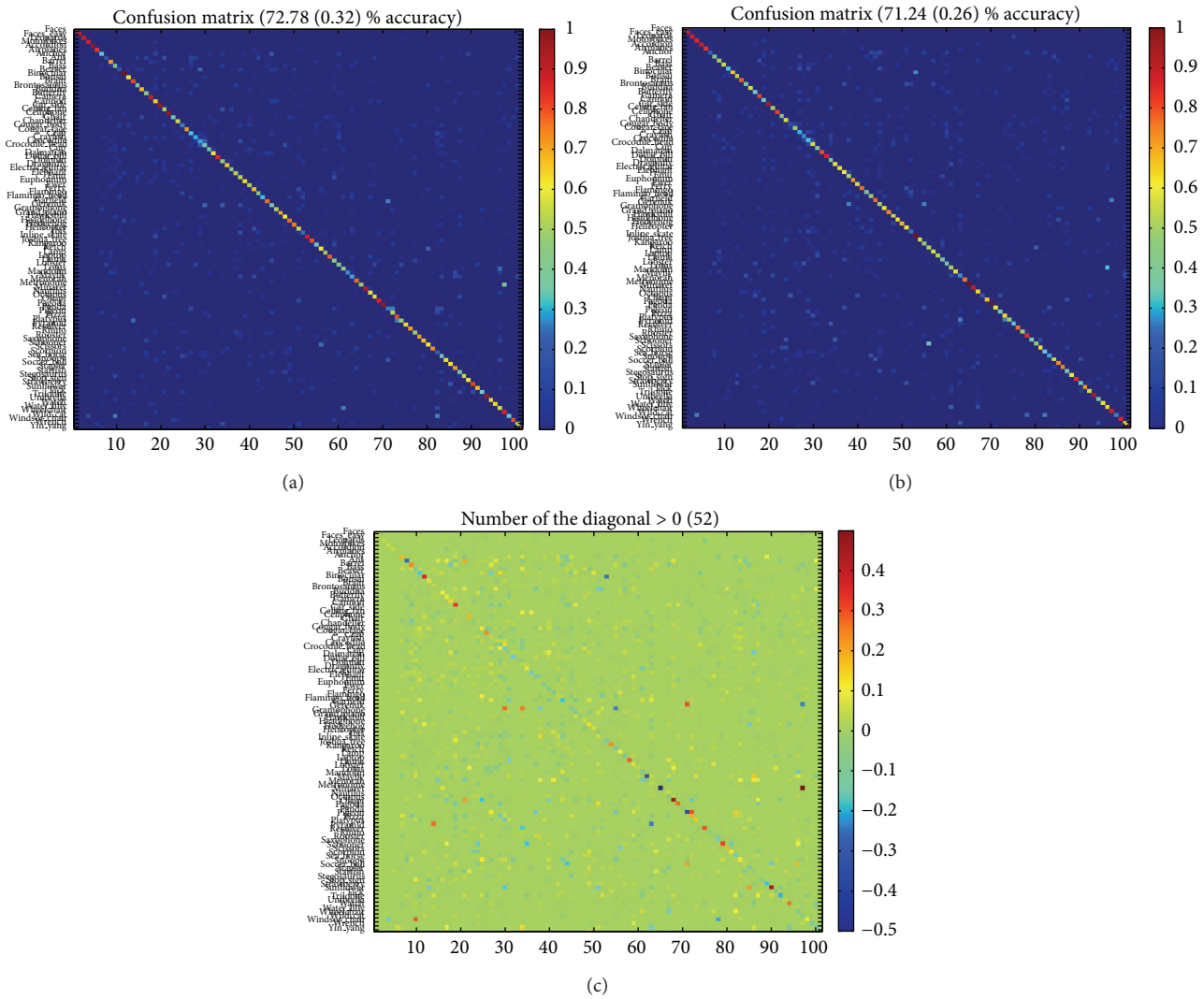

Figure 2: The average confusion matrices of the Caltech-101 dataset for 30 training per category. (a) Average confusion matrix by ours. (b) Average confusion matrix by BoW with dictionary size 4000. (c) Residual of (a) minus (b).

As similar to the Caltech-101 dataset, the memory usage in our method has reduced more than 13 times compared to BoW model with dictionary size set to 4000.

Figure 3 shows a comparison of the average confusion matrix in the Scene-15 dataset for 100 training per category. Figure 3(a) denotes the average confusion matrix by our proposed method, and Figure 3(b) denotes the average confusion matrix by BoW model with dictionary size set to 4000. In our results, the highest classification accuracy is 99.81% for *CALsuburb*, and the lowest is 72.62% compared to 61.61% in BoW with dictionary size set to 4000 for *industrial*.

5.3. Complexity Analysis. The computational complexity of BoW model can be decomposed into feature extraction, dictionary generation, feature coding, SVM training, and testing. Compared with BoW model, our method needs to run the K-means clustering algorithm multiple times for generating multiple dictionaries. As discussed in [36], the computational and storage complexity for K-means algorithm are both linear to the data samples; the computational complexity is $O(knd)$, where k is the number of clusters, n is the number of data samples, and d is the dimension of each data sample. In this paper, both BoW model and our method make use of the SIFT feature, so $d = 128$. If n and d are fixed, larger k needs more computational complexity, and the number of clusters in BoW model with dictionary size set to 4000 is 13 times larger than that of our method, so its running time for K-means is much longer than ours.

In feature coding stage, taking the hard-coding as the default coding strategy, the major computational complexity is to find the closest visual "word", where the Nearest Neighborhood (NN) method is used, makes the complexity be $O(knd)$.

We show the running time of dictionary generation (DG), feature coding (FC), and our modified Online Passive-Aggressive (OPA) Algorithms in Table 3 on Caltech-101 dataset. All results are obtained under a PC machine with

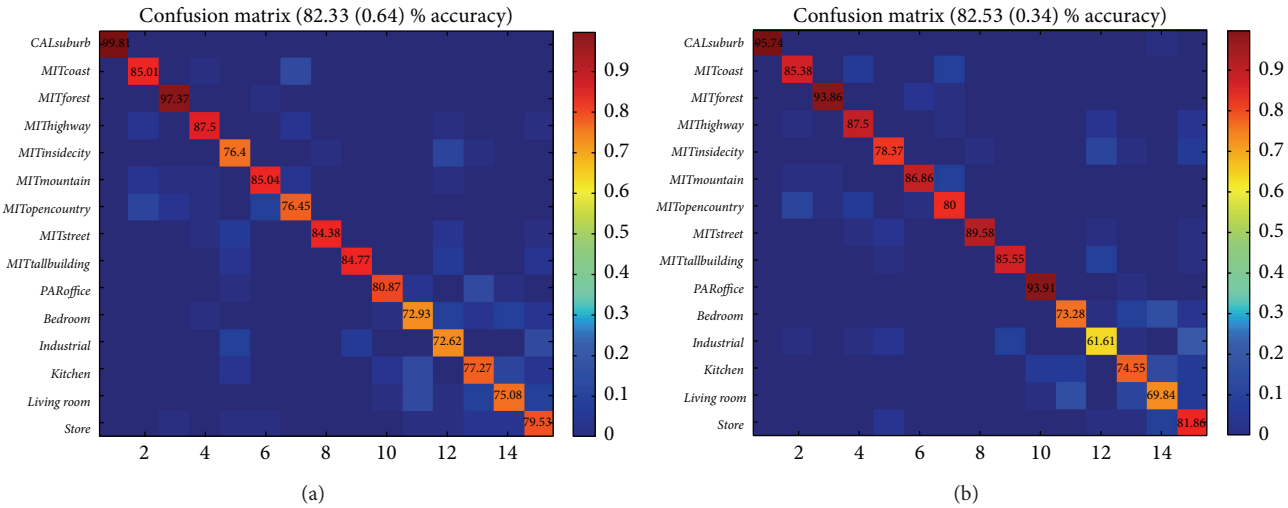

Figure 3: The average confusion matrices of the Scene-15 dataset for 100 training per category. (a) Average confusion matrix by ours. (b) Average confusion matrix by BoW with dictionary size 4000.

Table 3: Running time (second) comparison on Caltech-101.

Algorithms	DG	FC	OPA
BoW (400)	117.6	1774.0	—
BoW (1000)	333.9	2669.3	—
BoW (4000)	1050.4	8602.1	—
Ours	129.4	1916.6	13.58

single thread, the configurations of the PC are Intel Core i5 quad core CPU, and frequency is 3.4 GHZ, 8 GB RAM, Windows 7 64-bit operating system. All codes are written in Matlab.

As shown in Table 3, the running time of our method is slightly longer than that of BoW with dictionary size set to 400; this is because our method needs to run the K-means algorithm 3 times. And in feature coding, the time of our method is also longer than that of BoW with dictionary size set to 400; this may be because our method searches three times for each feature of an image. For BoW with dictionary size set to 4000, its running time is much longer than that of ours. It should be noticed that the running time of OPA only takes 13.58 s, which is less than 1% of feature coding time of our method (1916.6 s). This implies our modified Online Passive-Aggressive Algorithms is efficient and can be used in large scale application for learning the similarity between objects.

We ignore the analysis of memory usage of our method and BoW model; since the memory usage is also linear to the size of dictionary and feature dimension, larger dictionary needs more memory for storage.

6. Conclusion

Generally speaking, in image classification algorithm based on BoW model, the larger the dictionary, the more fine-gained the visual words, and the more discriminately the BoW histogram, leading to better performance. However, larger dictionary needs more memory usage and intensive computational resources. In this paper, we proposed a method to aggregate multiple feature coding adaptively in a weighted manner. The weights are learned by modified Online Passive-Aggressive Algorithms under the histogram intersection kernel, which lead to a closed-form solution. Extensive experimental results show that our proposed method obtains the same if not higher classification accuracy than the state-of-the-art method, but needs less memory and computing time, which verifies the effectiveness of our method. In the future, we will extend our method to soft-coding and sparse coding in the feature coding stage. Further, we would like to intergrate multiple features in order to improve the performance for image classification.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are grateful to Dr. Fuhao Zou and the anonymous referees for their valuable comments and suggestions. This work was supported in part by the National Natural Science Foundation of China (Grant no. 61300140).

References

[1] K. Grauman and B. Leibe, "Visual object recognition," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 5, no. 2, pp. 1–181, 2011.

[2] S. E. Fontes de Avila, *Extended bag-of-words formalism for image classification [Ph.D. thesis]*, Université Pierre et Marie Curie, Paris, France, 2013.

[3] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka, "Visual categorization with bags of keypoints," in *Proceedings of the Workshop on Statistical Learning in Computer Vision (ECCV '04)*, vol. 1, pp. 1–2, Prague, Czech Republic, 2004.

[4] J. Sivic and A. Zisserman, "Video google: a text retrieval approach to object matching in videos," in *Proceedings of 9th IEEE International Conference on Computer Vision*, pp. 1470–1477, IEEE, October 2003.

[5] T. Joachims, "Text categorization with suport vector machines: learning with many relevant features," in *Proceedings of the 10th European Conference on Machine Learning*, pp. 137–142, Springer, Chemnitz, Germany, April 1998.

[6] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

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

[8] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[9] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 1, pp. 604–610, October 2005.

[10] T. Tuytelaars, "Dense interest points," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2281–2288, IEEE, June 2010.

[11] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, IEEE, June 2007.

[12] H. Liao, J. Xiang, W. Sun, J. Dai, and S. Yu, "Adaptive initialization method based on spatial local information for k-means algorithm," *Mathematical Problems in Engineering*, vol. 2014, Article ID 761468, 11 pages, 2014.

[13] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 1794–1801, Miami, Fla, USA, June 2009.

[14] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.

[15] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[16] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: a comprehensive study," *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213–238, 2007.

[17] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 2, pp. 2169–2178, June 2006.

[18] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J.-M. Geusebroek, "Visual word ambiguity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1271–1283, 2010.

[19] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 3360–3367, IEEE, June 2010.

[20] E. Hadjidemetriou, M. D. Grossberg, and S. K. Nayar, "Multiresolution histograms and their use for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 831–847, 2004.

[21] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[22] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *Journal of Machine Learning Research*, vol. 11, pp. 1109–1135, 2010.

[23] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 2486–2493, IEEE, November 2011.

[24] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[25] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, vol. 2, pp. 1–8, IEEE, June 2008.

[26] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[27] N. Zhou and J. Fan, "Jointly learning visually correlated dictionaries for large-scale visual recognition applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 715–730, 2014.

[28] R. J. López-Sastre, J. Renes-Olalla, P. Gil-Jiménez, S. Maldonado-Bascón, and S. Lafuente-Arroyo, "Heterogeneous visual codebook integration via consensus clustering for visual categorization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 8, pp. 1358–1368, 2013.

[29] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, Article ID 1217303, 2007.

[30] L. Zheng, S. Wang, W. Zhou, and Q. Tian, "Bayes merging of multiple vocabularies for scalable image retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, p. 3, IEEE, 2014.

[31] L. Wang, L. Zhou, C. Shen, L. Liu, and H. Liu, "A hierarchical word-merging algorithm with class separability measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 417–435, 2014.

[32] B. S. Khadem, E. Farahzadeh, D. Rajan, and A. Sluzek, "Embedding visual words into concept space for action and scene recognition," in *Proceedings of the 21st British Machine Vision Conference (BMVC '10)*, pp. 15.1–15.11, BMVA Press, September 2010.

[33] A. Śluzek, "Large vocabularies for keypoint-based representation and matching of image patches," in *Proceedings of the 12th European Conference on Computer Vision (ECCV '12)*, pp. 229–238, Springer, October 2012.

[34] H. Jégou and O. Chum, "Negative evidences and co-occurences in image retrieval: the benefit of PCA and whitening," in *Proceedings of the 12th European Conference on Computer Vision*, pp. 774–787, Springer, Florence, Italy, October 2012.

[35] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, IEEE, June 2008.

[36] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Systems with Applications*, vol. 40, no. 1, pp. 200–210, 2013.

[37] A. Coates and A. Y. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *Proceedings of the 28th International Conference on Machine Learning (ICML '11)*, pp. 921–928, Bellevue, Wash, USA, July 2011.

[38] S. Chen, B. Ma, and K. Zhang, "On the similarity metric and the distance metric," *Theoretical Computer Science*, vol. 410, no. 24-25, pp. 2365–2376, 2009.

[39] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.

[40] MOSEK ApS, *The Mosek Optimization Toolbox for Matlab Manual, Version 7.0 (Revision 114)*, MOSEK ApS, Copenhagen, Denmark, 2014.

[41] 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," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[42] A. Vedaldi and B. Fulkerson, "Vlfeat: an open and portable library of computer vision algorithms," in *Proceedings of the International Conference on Multimedia (MM '10)*, pp. 1469–1472, October 2010.