Learning to detect concepts with Approximate Laplacian Eigenmaps in large-scale and online settings

Article *in* International Journal of Multimedia Information Retrieval · June 2015

DOI: 10.1007/s13735-015-0079-y

3 authors:

Eleni Mantziou
The Centre for Research and Technology, Hellas
5 PUBLICATIONS **24** CITATIONS

SEE PROFILE

Symeon Papadopoulos
The Centre for Research and Technology, Hellas
256 PUBLICATIONS **4,720** CITATIONS

SEE PROFILE

Ioannis (Yiannis) Kompatsiaris
The Centre for Research and Technology, Hellas
1,023 PUBLICATIONS **14,035** CITATIONS

SEE PROFILE

Noname manuscript No.
(will be inserted by the editor)

Learning to detect concepts with Approximate Laplacian Eigenmaps in large-scale and online settings

Eleni Mantziou · Symeon Papadopoulos · Yiannis Kompatsiaris

Received: date / Accepted: date

Abstract We present a versatile and effective manifold learning approach to tackle the concept detection problem in large scale and online settings. We demonstrate that Approximate Laplacian Eigenmaps (ALE), which constitutes a latent representation of the manifold underlying a set of images, offers a compact yet effective feature representation for the problem of concept detection. We expose the theoretical principles of the approach and present an extension that renders the approach applicable in online settings. We evaluate the approach on a number of well-known and two new datasets, coming from the social media domain, and demonstrate that it achieves equal or slightly better detection accuracy compared to supervised methods, while at the same time offering substantial speed up, enabling for instance the training of 10 concept detectors using 1.5M images in just 3 minutes on a commodity server. We also explore a number of factors that affect the detection accuracy of the proposed approach, including the size of training set, the role of unlabelled samples in semi-supervised learning settings, and the performance of the approach across different concepts.

Keywords Concept Detection · Semi-supervised Learning · Laplacian Eigenmaps · Online Learning

1 Introduction

Concept detection is a challenging computer vision problem that is typically defined as a multi-label classification problem, i.e. new images need to be classified to one or more concepts (also termed as categories or classes) out of a predefined concept set. Existing approaches typically focus on optimizing classification accuracy, often disregarding the training and data management costs involved in the process of creating new classification models. However, in modern image retrieval applications, there is often a need for rapidly building new classification models in order to accommodate the quickly changing information filtering needs of end users. In addition, there is often abundant unlabelled or weakly labelled multimedia content available that could be leveraged to improve the classification accuracy, typically with the use of semi-supervised learning (SSL) approaches. In a few cases, there are even large amounts of data available for training, yet using all of it for training is computationally very expensive, even though it appears that more training data leads to better classification results [29].

This paper focuses on SSL-based concept detection, which was found to lead to very competitive performance in standard datasets [10, 28] and has the potential of further improvements by leveraging additional unlabelled content in the training process. In particular, we focus on graph-based SSL approaches that have seen increasing adoption for concept detection in multimedia [6, 35–37] due to the popularity of graphs for modelling similarity in large image collections, and the intuitive idea that same-class images should cluster together (*cluster* assumption) and that images close on the graph should be more likely to be associated with the same concepts (*manifold* assumption).

This work was supported by the SocialSensor and REVEAL projects, partially funded by the European Commission, under the contract numbers FP7-287975 and FP7-610928 respectively.

E. Mantziou, S. Papadopoulos, Y. Kompatsiaris
Centre for Research and Technology Hellas (CERTH)
Information Technologies Institute (ITI)
Thessaloniki, 57001, Greece
E-mail: {lmantziou,papadop,ikom}@iti.gr

In graph-based SSL settings, a sparse $n \times n$ similarity graph is built to encode the visual similarities between images, and the resulting graph structure is leveraged to solve the concept detection problem. Two popular approaches to achieve this include the use of a) spectral graph analysis, e.g. using the graph Laplacian [5] as a new learning representation, and b) label propagation, utilizing a label diffusion process on the graph to spread the labels (concepts) from the known images (nodes) to the unknown ones. Both of these approaches, however, typically suffer from increased time complexity, which can be as high as $O(n^3)$. In addition, many of these approaches are *transductive* in nature, i.e. they are only applicable when both the training and testing images are available. Hence, both of these constraints render a large number of existing approaches impractical in realistic applications settings, which require fast training with a large number of samples, and real-time concept detection for newly arriving images.

Motivated by the above constraints, we present a scalable SSL approach that is also applicable in online settings. The proposed approach relies on the concept of Approximate Laplacian Eigenmaps (ALE) [9,23], and extends it to be applicable to online settings. The Inductive ALE (IALE) approach, presented in detail in Section 3.4, relies on a low-dimensional embedding of images computed from the training set to compute an embedding for the set of incoming images without the need to recompute the ALEs for the whole set of images. Furthermore, we conduct a comprehensive experimental study that delves into a number of previously unexplored aspects in large-scale SSL, including the role of the size and nature of the training set, the impact of feature dimensionality, and the performance across different concepts. More specifically, the paper makes the following contributions:

- We propose IALE, an inductive extension over ALE, that is easy to implement in online applications and also tackles the scalability and incremental computation issues, while maintaining high accuracy.
- We explore the trade-off between feature dimensionality and accuracy demonstrating exceptional gains in speed and scalability with marginal only decrease in detection accuracy.
- We compare with a supervised learning (SL) method in many datasets demonstrating that both ALE and IALE are very fast and robust methods compared with SL. The proposed methods were found to have competitive accuracy over a large range of concepts, including those arising in a social media context. In addition, we study the effect of the employed supervised learning step (SVM versus Smooth Functions) on the accuracy of the proposed approach.
- We explore the impact of adding more unlabelled images on the detection accuracy, demonstrating that for visually separable concepts, adding unlabelled data can moderately improve accuracy, while for overlapping concepts, adding unlabelled data can harm the detection accuracy.

To our knowledge, this is the first work that explores the use of a manifold learning approach on the problem of concept detection in large-scale and incremental settings and compares its performance to other semi-supervised and supervised learning approaches.

2 Related Work

SSL has received considerable attention in recent years due to its capability to use inexpensive unlabelled data. However, to make effective use of unlabelled data we need to make two strong assumptions, the cluster and the manifold assumption. Moreover, graph-based SSL algorithms suffer from two main problems: a) computationally intensive training step, typically involving the processing of large similarity matrices, b) applicability in online settings. In this section, we give a short overview of existing works on graph-based SSL and inductive graph-based SSL approaches. Extensive surveys on the topic are presented in [45] and [43].

2.1 Graph-based SSL

Given a set of training and a set of testing images, graph-based SSL proceed by building a graph that encodes the similarities among the images, and leverages the structure of the graph to perform the learning.

A large number of such approaches rely on the consistency of label predictions and the smoothness of the underlying graph. These algorithms are also known as *label propagation* algorithms, with *Gaussian Fields and Harmonic Functions* (GFHF) [44] and *Learning with Local and Global Consistency* (LLGC) [42] being two of the most popular approaches. These methods define a cost function to quantify the smoothness of the predicted labels over the distribution of the training data. Moreover, they converge to a solution in an iterative matter. The main drawbacks of these methods are that the results are sensitive to noise, and that they are inapplicable in large-scale online settings due to the need for iterative computations. A similar approach is grounded on the notions of hashing-based $l1$-graph construction and KL-based multi-label propagation [6], with the goal of handling large-scale datasets. However, the time complexity of this approach during inference still remains high, since it relies on a computational scheme that requires 50 iterations for convergence (as experimentally shown in [6]). Hence, its applicability in online settings is limited. Moreover, Wang et al. [36] proposed a bi-relational graph-based random walk approach, in which both the image graph and the label graph are used as subgraphs in a bipartite graph for image annotation. This approach is robust, but is transductive in nature and thus cannot be easily applied in online learning. In addition, it suffers from the same problem of constructing a huge similarity matrix when the training set is large.

Another class of graph-based SSL relies on *manifold learning*, which typically comprise a manifold encoding step combined with a supervised learning technique. LapSVM and regularized least squares (LapRLS) [4] constitute popular examples of such approaches. Another recent manifold-based approach is the Graph Structure Features (GSF) presented in [28]. GSF builds a sparse similarity graph and then computes the corresponding Laplacian Eigenmaps (LEs), which then uses as new features for learning the target concepts. Despite the obtained accuracy improvements, GSF cannot be used for large-scale learning due to its high complexity. In general, LE-based methods have to construct similarity matrices and subsequently the graph Laplacians, which have quadratic complexity to the size of labelled and unlabelled data. Thus, as the size of data increases, the use of such approaches becomes prohibitively expensive. Another problem in the construction of LEs is the diagonalization of a $n \times n$ data matrix which is impractical due to memory restrictions.

Motivated by the above issues, several methods were proposed to efficiently calculate the graph Laplacian. Some of them are based on building a smaller graph by randomly subsampling a subset of the points [34]. The drawback of these methods is that their output can change dramatically depending on the selected samples. In addition, the methods proposed in [34, 39] construct a large adjacency matrix implementing the Nyström method. Such approaches, however, cannot guarantee the graph Laplacian to be positive and semi-definite. In [38], the authors propose a variant, called clustered Nyström method. They construct approximate eigenfunctions to a kernel by choosing a subset (landmark points) of the entire data collection. The landmark points are determined as the cluster centers of a k-means sampling algorithm. This process is fast, though the main problem still remains, because the final eigenvectors are extracted by the sampled kernel matrix. In addition, Liu et al. [20] implement a method based on *anchor graphs* and Markov random walks between the data points and the anchors to produce an adjacency matrix that can guarantee the positiveness and semidefiniteness of graph Laplacians.

In [32], the authors propose an approach based on the cluster assumption for computing the adjacency matrix eigenfunctions. They assume that identifying the most important eigenfunction under high density areas corresponds to finding the most representative eigenvector. The drawback of this method is that if a data point belongs to multiple clusters, the representative eigenvector will fail to assign it correctly. Ji et al. [15] use the top eigenfunctions to build a prediction function by producing a Guassian kernel matrix. This approach ensures a better generalization error bound, but with scalability problems.

Recently, the ALE approach [23] was proposed to reduce the complexity by using the convergence of the eigenvectors of the normalized graph Laplacian to eigenfunctions using Markov Chains [9]. The framework presented here extends this approach to make it applicable in online settings.

2.2 Inductive graph-based SSL

Inductive approaches can detect concepts for newly arriving images without the need to perform global computations on the training set.

Jia et al. [16] propose a non-linear manifold learning method, in which they use sub-manifold analysis to derive the LE representation of a new point and to update the underlying manifold accordingly. Authors in [19] extend the seminal ISOMAP manifold learning algorithm [1] to online settings by using a nearest neighbour technique to describe the geometry of data and to compute the eigenvectors. Kouropteva et al. [18] propose an incremental extension of the LLE embedding algorithm [31], in which the nearest neighbours of new point are computed and then the neighbourhood is recalculated and the weights of the adjacency matrix are updated. Assuming the eigenvalues of the cost matrix remain the same when a new data point arrives, the minimization of cost matrix is solved by solving a $d \times d$ problem, where d is the number of eigenvectors.

Ning et al. [26] propose an incremental extension of spectral clustering by changing the eigenvalues of a dynamic system as new data points arrive. As the similarities among objects change, the algorithm updates the cluster labels. In [21], the authors propose Incremental LTSA, an incremental extension of Local Tangent Space Alignment (LTSA) [40] by using the geodesic structure of LTSA to compute the nearest neighbours of new data points and then to project the new point to the low-dimensional space close to its neighbours. Then, the ILTSA updates the low-dimensional coordinates of existing points. Another incremental SSL approach is presented in [17]. The algorithm clusters the training points using the LEs and generates cluster labels according to representative members. Then, for every new sample, the representative clusters are recomputed and the eigenvalues are updated to keep a newest set of representative points.

Zheng et al. [41] propose an alternative approach in online learning for large-scale datasets. They propose an online SVM method to update the model based on two prototypes, the Learning Prototypes and the Learning Support Vectors. The inductive extension of ALE described here was first presented in [24]; however, it was evaluated on a single dataset and in limited experimental settings. The present paper attempts a much more thorough presentation and experimental analysis of IALE in numerous datasets.

3 Approach description

The main motivation behind ALE is the efficient computation of a low-dimensional graph-based representation of images that are effective for the task of concept detection. Using the exactly computed LEs, as will be described in subsection 3.1, would require the construction of a $n \times n$ image similarity graph, the computation of its Laplacian, and more importantly the computation of the eigenvectors and eigenvalues of the graph Laplacian. As the number of images n grows, the approach becomes impractical due to the excessive computation costs. To this end, ALE proposes an approximate solution to the problem that is much faster to compute. This is described in detail in subsection 3.2. Similar to the original LE-based approach, ALE remains transductive, and hence is not suitable for application in online problem settings. To render the approach inductive, we present an extension of the approach in subsection 3.4, in which the features of the newly arriving images are projected to the existing LE vectors with an appropriate interpolation operation.

Overall, the inductive concept detection process can be summarized in the following steps: a) feature extraction, in which the feature vectors of choice are extracted from the images available at training time (those can be solely labelled images, or both labelled and unlabelled ones), b) dimensionality reduction through PCA, c) approximate computation of the eigenfunctions and eigenvectors of the image similarity matrix Laplacian without explicitly creating the graph, d) use the approximate LE vectors to create concept detectors using "standard" supervised learning techniques, e.g. SVM, e) project the PCA-reduced extracted features of the "new" images to the space of approximate LE vectors, f) use the trained concept detector to predict their labels. Step c) is detailed in subsections 3.1 and 3.2. Steps d) and f) are covered in subsection 3.3. Step e) is discussed in subsection 3.4. The impact of steps a) and b) is examined in Section 4. The notation used in the following is summarized in Table 1.

3.1 Background

Graph-based SSL leverages both labelled and unlabelled images by considering them as nodes (vertices) of a graph where edges (links) reflect the similarity between them. Given a set of K target concepts $\mathcal{Y} = \{Y_1 .... Y_K\}$ and a labelled set $\mathcal{L} = \{(\mathbf{x_i}, \mathbf{y_i})\}_{i=1}^{l}$ of training samples, where $\mathbf{x_i} \in \Re^D$ stands for the feature vector extracted from image i and $\mathbf{y_i} \in \{0,1\}^K$ for the corresponding concept indicator vector, a transductive learning algorithm attempts to predict concepts associated with a set of unlabelled items $\mathcal{U} = \{x_j\}_{j=l+1}^{l+u}$, by processing together sets $\mathcal{L}$ and $\mathcal{U}$. Based on the features of the input items, a graph $G = (V, E)$ is constructed that represents the similarities between all pairs of items. The nodes of the graph include the items of both sets $\mathcal{L}$ and $\mathcal{U}$, i.e. $V = V_L \cup V_U$ with $|V| = n$.

There are different options for constructing such a graph. A k-nn graph is created when an edge is inserted between items i and j as long as one of them belongs to the set of top-k most similar items of the other. Similarity between i and j can be computed by a $n \times n$ Gaussian kernel W (Heat Kernel):

$$w_{ij} = \exp\left(-\frac{|x_i - x_j|^2}{2t^2}\right) \qquad (1)$$

Symbol	description
$\mathcal{Y}$	Label matrix $y_i, ..., y_K$ of K concepts
L	Set of labelled images
$\mathcal{U}$	Set of unlabelled images
$X : x_i, x_j$	Feature matrix: x_i and x_j feature vectors with N dimensions
G	Image similarity graph
W	Heat Kernel
D	Diagonal matrix whose elements are $\Sigma w_{i,j}$
L	Graph Laplacian
$\tilde{L}$	Normalized Graph Laplacian
c_D	Number of eigenvectors
n	Number of images
B	Number of bins used to quantize the distribution
$p(x)$	Distribution over feature x discretized on bin values b
P	$B \times B$ diagonal matrix whose diagonal elements correspond to $p(x)$
$\tilde{W}$	$B \times B$ matrix expressing affinity between discrete points of density p
g	Eigenfunctions
σ, Σ	Eigenvalue, Eigenvalue matrix
ϕ, U	Eigenvector, Eigenvector matrix
$\tilde{D}$	Diagonal matrix equal to $\Sigma P \tilde{W} P$
$\hat{D}$	Diagonal matrix equal to $\Sigma P \tilde{W}$
f	Smooth Function
a	Smooth operator of f
Λ	Diagonal matrix, where $\Lambda_{ii} = \lambda$ if labelled, otherwise $\Lambda_{ii} = 0$

Table 1: Math notation

where t defines the strength of the affinity, how strongly connected the nodes are. Having constructed the similarity graph between the input items, we have to map the graph nodes to feature vectors that encode the position of nodes on the underlying manifold. This is achieved with the help of the graph Laplacian. Let D be the diagonal vertex degree matrix, defined as $D_{ii} = \sum_{j=1}^{n} w_{ij}$. Then, the graph Laplacian is defined as:

$$L = D - W \tag{2}$$

To extract the LE feature vectors, we must first construct the normalized Laplacian:

$$\tilde{L} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2} \tag{3}$$

Computing the eigenvectors of $\tilde{L}$ corresponding to the smallest non-zero eigenvalues of the matrix results in a set of c_D eigenvectors with n dimensions, which are then stacked to form the input matrix $S \in \Re^{nxc_D}$, each row of which is denoted as $S_i \in \Re^{c_D}$ and constitutes the LE feature vector for image i.

3.2 ALE construction

To compute the LEs exactly, we would need to build a $n \times n$ similarity graph between labelled and unlabelled images and then compute the eigenvalues and eigenvectors of its Laplacian. For large n, this is very costly. ALE tackles this problem based on an approximation of LEs by estimating a smaller covariance matrix, as suggested in [9], where it is hypothesized that the data $x_i \in \Re^d$ are samples from a distribution $p(x)$.

The key idea of the approach is that if the points are sampled uniformly at random from a manifold, then the eigenvectors of the corresponding graph Laplacian would converge to the eigenfunctions of the Laplace Beltrami operator. More specifically, as the number of samples goes to infinity we construct a set of approximate eigenfunctions to map the geometry of samples and to find the convergence of eigenfunctions over a diffusion map (a family of eigenvectors and eigenvalues defined on a low dimensional Euclidean space) using a discrete random walk [25]. To this end, for every dimension we construct Markov random walks along the images of the graph. For this, a transition probability matrix P is computed based on their pairwise similarities W (Equation 1). Specifically, we choose the Laplace-Beltrami operator to construct the random walk, and to compute the approximate eigenfunctions in the limit of infinite data, we make use of Equation 4. Therefore, under suitable convergence conditions we compute the eigenvectors from a small number of eigenfunctions (the ones with the smallest eigenvalues), which capture the geometrical and statistical properties of the data.

The ideal solution to derive the eigenfunctions would be that the density of features follows a known distribution (i.e Gaussian). The issue is that real data do not have such known distributions leading to the need for making certain assumptions to solve the problem. For every dimension of the N-dimensional feature vector of the images, a $B \times B$ matrix $\tilde{W}$ is derived, which expresses pairwise point affinities in the respective dimension, along with a diagonal matrix P, whose diagonal elements approximate the density $p(s)$ of the rotated data. Assuming that the N distributions are independent, the eigenfunctions of marginals are also eigenfunctions of the joint density. By building the histogram from discrete data, we create such marginal distributions. Therefore, using matrices $\tilde{W}$ and P, we numerically determine the eigenfunctions, and hence the eigenvalues, at a set of discrete points (the centres of histogram bins). Then, instead of computing the eigenfunctions of the similarity matrix between the original images, one can define eigenfunctions g corresponding to the eigenvalues σ of the rotated data, which can be seen as approximations of the LEs of the original data when $n \to \infty$. This is considerably faster, since typically $B \ll n$. These are recovered by solving the following equation:

$$\left(\tilde{D} - P\tilde{W}P\right) g = \sigma P \hat{D} g \tag{4}$$

$\tilde{D}$ is a diagonal matrix whose diagonal elements are the sum of the columns of $P\tilde{W}P$, and $\hat{D}$ is a diagonal matrix whose diagonal elements are the sum of the columns of $P\tilde{W}$. An example of this procedure is illustrated in Figure 1. The final step involves the interpolation of feature vectors to the target dimension c_D to derive the $U \in \Re^{n \times c_D}$ approximate LE vectors.

3.3 Model creation

We examine two supervised learning approaches for creating the concept models: SVMs and Smooth Functions. According to the first, a linear classifier is trained using the approximate vectors of the labelled items as input. In our implementation, we opted for the use of linear SVM to further increase the training speed. According to the second, we use the LEs to define a smoothness operator that takes into account the unlabelled data. The key idea is to find a set of functions f, which agree with the labelled data and are also smooth with respect to the graph. To this end, the following minimization problem needs to be solved:

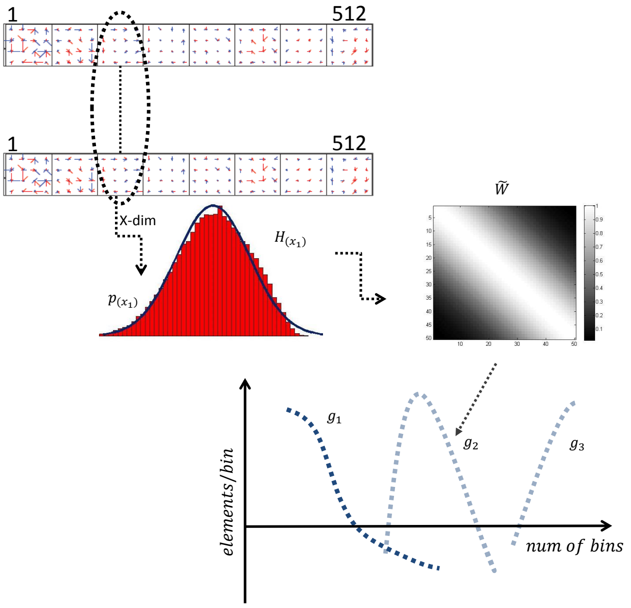


Fig. 1: Eigenfunctions computation. From every dimension of discrete data we create the marginal distribution $p(x_i)$, from which we derive the diagonal matrix P. We also construct the affinity matrix $\tilde{W}$ and use P and $\tilde{W}$ to solve for the values of g eigenfunctions according to Equation 4. According to these, we compute the smallest eigenvalues and then the final U eigenvectors.

$$arg \min_f \frac{1}{n^2} f^T L f = \frac{1}{2n^2} \sum_{ij} W_{ij}(f(i) - f(j))^2 \qquad (5)$$

where f denotes the value of label function F. That means that two points that are close to each other on the graph are more likely to share a label. In addition, the smoothness of any vector f can be defined as a linear combination of the eigenvectors with smallest eigenvalues [45]:

$$f = \Sigma_i \alpha \phi_i \qquad (6)$$

In our implementation, we define $f = U\alpha$. Thus, the minimization problem of Equation 5 reduces to the minimization of α through the following equation:

$$(\Sigma + U^T \Lambda U)a = U^T \Lambda y \qquad (7)$$

where Σ are the smallest eigenvalues (cf. Equation 4), U consists of the $n \times c_D$ approximate LE vectors, Λ is a diagonal matrix, whose diagonal elements are $\Lambda_{ii} = \lambda$ if i is labelled, otherwise $\lambda = 0$, and y are the labels.

3.4 Inductive ALE

ALE was originally proposed and tested in transductive learning settings [23]. There, the computation of the learning representation, i.e. the LE vectors U, happened simultaneously for the training and test images. Assuming that new test samples arrive in the system, one would need to recompute the LE vectors from scratch for the union of the training and test images. In contrast, inductive SSL approaches assume that the low-dimensional representations U_i of x_i are pre-computed for the training samples. When a new sample x_{n+1} is observed, the learning algorithm should project x_{n+1} to the space spanned by U so that the trained concept models can be applied.

To this end, in Inductive ALE (IALE) we reuse the $B \times B \times N$ eigenfunctions g and the $B \times N$ eigenvalues σ that were derived from the training data by numerically solving Equation 4. To derive each dimension $U_{n+1}(i)$ of sample x_{n+1}, we first identify the corresponding dimension $x_{n+1}(a_i)$ and the corresponding eigenfunction g_i and bins b_i, based on the ordering of the original eigenvalues σ (for instance, if the σ_3 was the smallest eigenvalue, then $a_1 = 3$). Having identified the bin values b_i^0, b_i^1, such that $b_i^0 \leq x_{n+1}(a_i) < b_i^1$, linear interpolation is used to derive the corresponding ALE value:

$$U_{n+1}(i) = g_i(b_i^0) + (g_i(b_i^1) - g_i(b_i^0)) \frac{x_{n+1}(a_i) - b_i^0}{b_i^1 - b_i^0} \qquad (8)$$

where $g(b)$ denotes the value of eigenfunction g at the center of bin b. Equation 8 is applied c_D times to derive the full ALE vector U_{n+1}.

3.5 Complexity analysis

Assuming n images available at training time, and the use of features with N dimensions, the training cost for extracting the ALE structure is $O(N \cdot B^2 \cdot n \cdot c_D)$. As will be seen in the experiments section, ALE performs particularly well with aggregated dimensionality reduced features, e.g. PCA-reduced VLAD with $N = 512$. Also, the approach is insensitive to B and hence low values are selected for efficiency reasons. In our experiments, we set $B = 50$. Finally, small to medium values are typically selected for c_D, e.g. $c_D = 500$. Given these assumptions, it becomes obvious that $N \cdot B^2 \cdot c_D$ can be considered as a constant factor, and hence the approach scales linearly to the number n of images available at training time. At prediction time, the computation of U_{n+1} based on Equation 8 requires only c_D linear interpolations, hence the prediction time is essentially equal to the time required to run the classification model.

4 Experimental Study

We performed a systematic evaluation of the proposed framework in a variety of datasets and settings. Section 4.1 describes the datasets and the evaluation criteria used in the study. Section 4.2 is divided in five parts. In the first, we explore the effect of different features on the accuracy of concept detection. In the second, we compare the two different learning methods of subsection 3.3, the linear SVM classifier and the smooth function. In the third part, we compare the proposed framework against a standard supervised learning method. Finally, we explore the IALE performance in large-scale settings, and the impact of adding new data to the training process in the fourth and fifth parts respectively.

4.1 Experimental Setup

4.1.1 Datasets and evaluation criteria

Most of the experiments were carried out on the following six datasets.

- **Flickr-25K**: The MIR-Flickr (MIRF) [12] dataset is associated with two different ground truth annotations. The first one has a set of 24 concepts [12] and the second is the ImageCLEF 2012 (ICLEF12) annotation [2] that has 94 concepts. Both contain annotations for all 25K images of MIRF. 15K images are used as labelled and the rest as unlabelled. Moreover, to study the learning stability of various algorithms, we vary the number of labelled images along 1K, 5K, 7.5K, 10K and 15K.
- **NUS-WIDE**: This contains 269,648 images and the associated 5,018 tags and 81 concepts [7]. 161,789 images are used as labelled and the rest as unlabelled. In the respective tests, we vary the number of labelled images along 10K, 20K, 50K, 90K, 120K and 162K.
- **Yahoo GC**[1]: This contains 2M images and 10 general concepts. Out of those, 1.5M images are used as labelled and the rest as unlabelled. The number of labelled images varies between 10K and 1.5M.
- **Flickr2013**[2]: This contains 22,142 images crawled from Flickr groups and about 10K images from the MIRFLICKR-1M dataset [12] that were added to the set of negative examples. It comprises 14 concepts, including concepts from the news domain, such as *demonstrations* and *Obama*. We use 14,762 images as labelled and the rest as unlabelled.
- **Twitter2013**[3]: This contains 12,635 images crawled from Twitter hashtags and users and about 10K images from the MIRFLICKR-1M dataset [12], again used to enrich the set of negative examples. It consists of five concepts: *Selfie*, *Porn*, *Messages*, *Memes*, *Keepcalm*. We use 8,424 images as labelled and the rest as unlabelled.

Evaluation criteria: To measure concept detection performance, we use the interpolated Average Precision (iAP) for each concept and the Mean interpolated Average Precision (MiAP) across all concepts [22, p.158][4]. We also use the Precision@100 score and the classification error to measure the performance of the framework against competing approaches. Each experiment is repeated 10 times and the MiAP averaged over the splits.

4.1.2 Framework Setup

Feature extraction: As state-of-the-art features, we test the $d = 128$ SIFT [30] computed using the vlfeat implementation[5]. As a fast alternative, we also test the Speeded-Up Robust Features (SURF, $d = 64$) [3] using the implementation of [33][6]. In both cases, we used a dense regular grid with a spacing of 6 pixels.

To aggregate the local descriptors into a single vector per image, we used the Vectors of Locally Aggregating Descriptors (VLAD). For SIFT, we performed K-means clustering with a vocabulary size of $K = 64$ centroids for better performance as proposed in [14] and for SURF with $k = 4 \times 128$ to apply multiple vocabulary aggregation [13]. The clustering was performed on an independent set of 10,000 images, randomly sampled from the MIRFLICKR-1M dataset [12]. The final VLAD vectors are power-and L2-normalized and then reduced to $D = \{512\}$ and L2-normalized again [33].

We also experimented with additional features to measure the impact of different representations in combination with ALEs. More specifically, we used the RGB-SIFT ($d = 384$) local descriptors and GIST [27] to measure the effect of a global descriptor combined with LEs. To measure the effectiveness of text-based features, we applied probabilistic Latent Semantic Analysis (pLSA) [11] on the tag bag-of-words vectors (with respect to the 1000 most frequent tags) using 100 latent topics.

[1] http://acmmm13.org/submissions/call-for-multimedia-grand-challenge-solutions/yahoo-large-scale-flickr-tag-image-classification-challenge/

[2] http://www.socialsensor.eu/datasets/mm-concept-detection-dataset-2013/mm-concept-detection-datasets.zip

[3] http://www.socialsensor.eu/datasets/mm-concept-detection-dataset-2013/mm-concept-detection-twitter2013-images.zip

[4] MiAP is also known as 11-points interpolated average precision. It is computed with the `vl_pr()` method of the `vlfeat` library, `http://www.vlfeat.org/matlab/vl_pr.html`

[5] http://www.vlfeat.org/

[6] https://github.com/socialsensor/multimedia-indexing

Competing approaches: Table 2 lists the competing concept detection systems. We compare both the transductive and inductive version of ALE combined with both SVM and smooth functions. We also compare to GSF, a recently proposed SSL approach [28], which is equivalent to using the LEs of an image similarity graph in combination with an SVM, and was found to exhibit highly competitive accuracy compared to alternative SSL approaches, such as the one based on Multiple Kernel Learning [10]. Due to its computational complexity, this approach was only tested on the Flickr-25K dataset (MIRF and ICLEF12). As a state-of-the-art baseline, we use SVM directly on the original feature representation.

Name	Methods
ALE-SVM	ALE with Linear SVM
IALE-SVM	IALE with Linear SVM
ALE-SF	ALE with Smooth Function
IALE-SF	IALE with Smooth Function
GSF [28]	Equivalent to exact LEs with SVM
BSVM	Baseline SVM

Table 2: The competing concept detection systems

Classification settings: We use linear SVM as baseline, in particular, the implementation of the liblinear library [8]. We set the SVM parameter $c = 5$ in all experiments.

ALE settings: In ALE, very limited parameter tuning was carried out: it was observed that different values of B did not considerably affect accuracy. Thus, we choose to set $B = 50$ for computational efficiency reasons. Regarding the number of eigenvectors, our preliminary experiments on the ICLEF12 dataset [2], reported in Table 3, indicated that a reasonable choice was $c_D = 500$, which we used across all experiments presented in the rest of this study. Later experiments on the NUS-WIDE dataset, reported in Table 4, indicate that further performance gains would be possible by dataset-specific tuning of the parameter. However, we opted for avoiding dataset-specific parameter tuning, since our interest has been to test the approach in real-time indexing settings, where there is no time available for offline processing operations (such as parameter tuning). For λ we observe in Figure 2 (also based on the ICLEF12 dataset) that after $\lambda = 10$ the MiAP stabilizes and we choose to set $\lambda = 100$.

IALE settings: Due to the optimized vector implementations of MATLAB, we performed the concept detection in batches of images (instead of per single image). In each batch we included 1000 images. This still simulates a realistic indexing scenario, when large amounts of images constantly arrive in the system.

Table 3: Dependence of MiAP on c_D in ICLEF12 (using SIFT and 15K training).

Method	$c_D = 50$	$c_D = 100$	$c_D = 200$	$c_D = 500$	$c_D = 1000$
ALE-SVM	21.81	23.22	24.66	24.90	**24.96**
ALE-SF	18.22	20.56	22.88	**24.57**	24.00

Table 4: Dependence of MiAP on c_D in NUS-WIDE (using SIFT and 160K training).

Method	$c_D = 100$	$c_D = 500$	$c_D = 1000$
ALE-SVM	16.26	22.06	**22.66**
ALE-SF	17.73	22.04	**23.93**

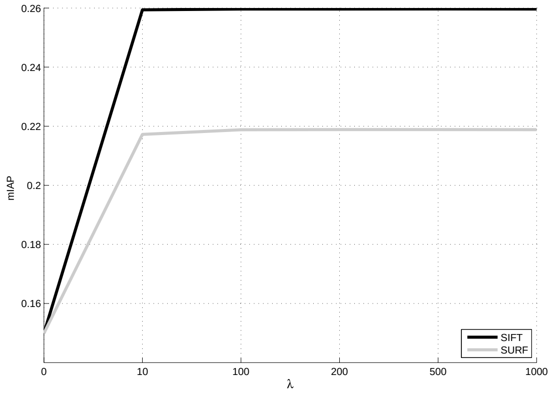


Fig. 2: Detection accuracy with respect to λ in ICLEF12

All experiments were implemented in MATLAB and executed on a 24-core (an Intel Xeon Q6600@2.0Ghz, 128G RAM) machine. The code is available on GitHub[7].

4.2 Results

4.2.1 PCA and features

In [23], we had concluded that the use of PCA on the VLAD vectors (for different variants of SIFT) offered substantial accuracy gains in terms of Mean interpolated Average Precision (MiAP) on concept detection using ALEs on the MIRF dataset. Here, we are interested in the impact that the extent of dimensionality reduction has on the detection accuracy over a large variety of datasets. Table 5 presents the effect of PCA in all datasets examined in this paper. The experiments are conducted for the transductive version of ALE and they are repeated for both the SVM and the smooth functions-based learning.

One may note that **performing more aggressive reduction** (i.e. to 512 dimensions instead of 1024)

[7] https://github.com/socialsensor/mm-concept-detection-experiments

MIRF					
Descriptor	K	D	#	1024	512
SIFT	64	128	ALE-SVM	**46.8**	46.6
			ALE-SF	46.7	46.6
SURF	4x128	64	ALE-SVM	**42.6**	**42.6**
			ALE-SF	41.8	41.9
RGB-SIFT	64	384	ALE-SVM	**49.4**	49.3
			ALE-SF	**49.4**	49.3

ICLEF12					
Descriptor	K	D	#	1024	512
SIFT	64	128	ALE-SVM	24.84	**24.9**
			ALE-SF	24.4	24.5
SURF	4x128	64	ALE-SVM	23.8	**23.9**
			ALE-SF	22.2	22.1
RGB-SIFT	64	384	ALE-SVM	**26.9**	26.8
			ALE-SF	26.3	26.3

Yahoo GC					
Descriptor	K	D	#	1024	512
SIFT	64	128	ALE-SVM	22.3	21.65
			ALE-SF	**40.9**	39.55
SURF	4x128	64	ALE-SVM	20.0	20.0
			ALE-SF	40.5	**40.92**
RGB-SIFT	64	384	ALE-SVM	22.65	21.46
			ALE-SF	**42.4**	41.21

NUS-WIDE					
Descriptor	K	D	#	1024	512
SIFT	64	128	ALE-SVM	20.8	19.8
			ALE-SF	**21.9**	20.2
SURF	4x128	64	ALE-SVM	15.3	15.4
			ALE-SF	**17.7**	17.6

Flickr2013					
Descriptor	K	D	#	1024	512
SIFT	64	128	ALE-SVM	**56.1**	52
			ALE-SF	55.44	52.02
SURF	4x128	64	ALE-SVM	45.4	**45.8**
			ALE-SF	45.03	44.66

Twitter2013					
Descriptor	K	D	#	1024	512
SIFT	64	128	ALE-SVM	77.7	73.8
			ALE-SF	**80.41**	72.00
SURF	4x128	64	ALE-SVM	72.6	**73.4**
			ALE-SF	71.26	71.74

Table 5: Accuracy of two learning configurations (ALE-SVM, ALE-SF) with respect to reduced size of VLAD.

affects only marginally the detection accuracy. This is important in case we are interested in storing the computed representation. More specifically, ALE-SVM in combination with SIFT 1024-d achieves a MiAP score of 46.8% on the MIRF dataset, while the use of SIFT 512-d slightly reduces accuracy to 46.6%. Also, ALE-SF in combination with SIFT 1024-d achieves a score of 46.7%, while with SIFT 512-d the score remains almost the same 46.6%. We also observe that SURF is more amenable to dimensionality reduction, since the performance difference there between 1024-d and 512-d is even smaller, and in some cases 512-d leads to higher accuracy. Hence, for the rest of the experiments we choose to use the smaller feature dimensionality to reduce the computational costs.

Tables 6-9 present the detection accuracy over different datasets (Flickr2013 and Twitter2013 are not included due to space limitations) and using different features. Tables 6 and 7 demonstrate that the vectors of aggregated local descriptors significantly outperform the global descriptors. In contrast, the tag-based pLSA features do not have consistent performance. In the case of MIRF and ICLEF12, they perform worse than the vectors of local descriptors, while the situation is reversed on the NUS-Wide dataset. Figure 3 illustrates several cases where the manifold assumption holds for SIFT and pLSA features, and others where it is violated. Naturally, in cases where the most similar images to the input image are largely irrelevant (in terms of the concept of interest) to the image, then the manifold assumption does not hold and consequently the concept detection fails.

Among local descriptors, RGB-SIFT is found to outperform both SIFT and SURF in all datasets where it was tested. Also, there is no considerable difference in the performance between SIFT and SURF in most of the datasets. In most of the datasets SIFT performs somewhat better, while in the case of the Yahoo GC dataset SURF achieves higher accuracy. The main conclusion out of these observations is that **the ALE framework can achieve highly competitive accuracy even with fast-to-compute and compact features** (SURF+VLAD).

4.2.2 SVM versus SF

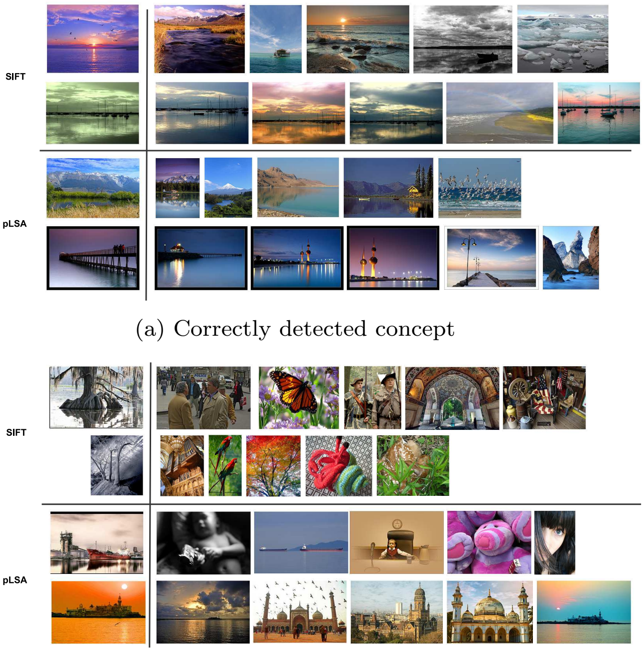
The results of Tables 6 - 9 also give insights into the role of the learning approach (SVM versus SF) employed in combination with the ALE (and IALE) representation. It appears that in the MIRF and ICLEF12 datasets, the use of SVM is clearly beneficial for the accuracy of the system. In the case of NUS-WIDE, the results are not conclusive, since there are settings where SF performs slightly better than SVM, and others where the reverse outcome is observed. Instead, in the case of Yahoo GC, the use of SF results in a clear improvement compared to SVMs. Another noteworthy observation is that SF tends to perform better in combination with the pLSA tag-based features. The difference in performance between the two approaches may be attributed to their different principles: SVMs try to maximize the inter-class separation of the training samples, hence its performance deteriorates in datasets and features where samples from different classes “mix” on the feature space. In contrast, SF attempts to enforce class

MIRF						
Descriptor	Method	1K	5K	7.5K	10K	15K
GIST	ALE-SVM	30.90	32.16	32.05	32.09	31.82
	IALE-SVM	30.31	33.64	33.75	33.23	32.26
	ALE-SF	29.27	32.94	33.47	34.07	34.72
	IALE-SF	25.61	32.06	33.02	33.79	34.63
	GSF [28]	**34.02**	**36.51**	**37.08**	**37.52**	**35.30**
	BSVM	30.33	31.01	31.88	32.57	33.13
SIFT	ALE-SVM	36.80	**43.74**	**45.05**	**45.80**	46.58
	IALE-SVM	33.20	43.17	44.76	45.60	46.51
	ALE-SF	32.86	43.03	44.63	45.50	**46.63**
	IALE-SF	31.50	42.86	44.46	45.36	46.40
	GSF [28]	**39.64**	43.72	44.70	45.26	46.05
	BSVM	35.80	42.04	43.76	44.82	46.24
RGB-SIFT	ALE-SVM	38.81	**46.27**	**47.56**	**48.35**	49.33
	IALE-SVM	35.04	45.64	47.33	48.32	49.35
	ALE-SF	34.97	45.97	47.50	48.31	**49.37**
	IALE-SF	33.46	45.70	47.36	48.15	49.24
	GSF [28]	**41.15**	45.80	47.05	47.77	48.40
	BSVM	38.01	44.45	46.35	47.57	49.01
SURF	ALE-SVM	32.57	39.45	40.84	41.70	42.88
	IALE-SVM	30.84	39.05	40.73	41.67	42.81
	ALE-SF	30.33	38.68	40.03	40.80	41.90
	IALE-SF	29.90	38.61	40.00	40.83	41.90
	GSF [28]	**37.43**	**42.31**	**43.42**	**44.28**	**45.26**
	BSVM	31.66	37.86	39.84	41.00	42.50
pLSA	ALE-SVM	23.17	23.16	23.17	23.17	23.20
	IALE-SVM	23.20	23.20	23.18	23.18	23.17
	ALE-SF	23.20	23.17	23.20	23.20	23.20
	IALE-SF	23.21	23.18	23.20	23.20	23.18
	GSF [28]	**23.55**	**23.51**	**23.52**	**23.51**	**23.58**
	BSVM	23.22	23.20	23.20	23.18	23.18

Table 6: MiAP results in MIRF

ICLEF12						
Descriptor	Method	1K	5K	7.5K	10K	15K
GIST	ALE-SVM	15.18	16.58	16.70	16.68	16.66
	IALE-SVM	14.52	16.87	17.25	17.43	17.24
	ALE-SF	13.79	15.87	16.36	16.88	17.29
	IALE-SF	12.46	15.31	16.00	16.56	17.11
	GSF [28]	**19.09**	**20.78**	**21.12**	**21.41**	**22.00**
	BSVM	13.35	14.87	15.32	15.70	16.18
SIFT	ALE-SVM	17.15	21.87	23.11	23.83	24.90
	IALE-SVM	15.77	21.26	22.64	23.57	24.51
	ALE-SF	15.11	21.20	22.48	23.28	24.57
	IALE-SF	14.64	21.09	22.31	23.04	24.25
	GSF [28]	**21.52**	**24.13**	**24.89**	**25.23**	**25.84**
	BSVM	17.21	20.48	21.50	22.21	23.57
RGB-SIFT	ALE-SVM	18.20	23.62	24.83	25.65	26.85
	IALE-SVM	16.53	23.00	24.42	25.31	26.73
	ALE-SF	16.09	22.89	24.24	25.01	26.36
	IALE-SF	15.51	22.62	24.03	24.83	26.36
	GSF [28]	**22.30**	**25.40**	**26.25**	**26.71**	**27.44**
	BSVM	18.29	22.02	23.22	24.11	25.50
SURF	ALE-SVM	17.90	21.52	22.36	23.03	23.93
	IALE-SVM	17.33	21.16	22.14	22.91	23.86
	ALE-SF	16.83	20.38	21.01	21.50	22.25
	IALE-SF	21.16	20.28	20.95	21.53	22.24
	GSF [28]	**20.78**	**23.72**	**24.40**	**24.98**	**25.55**
	BSVM	17.97	20.34	21.14	21.91	22.95
pLSA	ALE-SVM	11.71	11.83	11.73	11.78	11.61
	IALE-SVM	11.08	11.32	11.40	11.44	11.50
	ALE-SF	11.47	11.77	11.71	11.74	11.62
	IALE-SF	11.21	11.60	11.60	11.64	11.68
	GSF [28]	**15.11**	**15.07**	**15.08**	**15.08**	**15.07**
	BSVM	11.07	11.31	11.37	11.43	11.48

Table 7: MiAP results in ICLEF12



(a) Correctly detected concept

(b) Incorrectly detected concept

Fig. 3: Five most similar images using SIFT and pLSA for correct and incorrect detection of concept *lake* in NUS-WIDE.

NUS-WIDE							
Descriptor	Method	10K	20K	50K	80K	120K	160K
SIFT	ALE-SVM	17.43	18.51	19.61	20.08	20.11	22.06
	IALE-SVM	**17.80**	**20.00**	21.20	21.66	21.70	21.42
	ALE-SF	16.02	17.56	19.23	19.67	20.00	22.04
	IALE-SF	14.93	16.28	17.90	18.16	18.35	21.40
	BSVM	17.40	19.26	**21.87**	**23.01**	**23.85**	**24.26**
SURF	ALE-SVM	14.61	15.18	14.78	14.36	13.33	13.12
	IALE-SVM	15.20	**16.30**	**17.40**	17.09	16.14	16.01
	ALE-SF	**15.30**	16.15	17.13	17.44	17.63	17.69
	IALE-SF	14.51	15.96	17.08	17.35	17.44	17.50
	BSVM	14.80	15.86	17.25	**17.95**	**18.50**	**18.86**
pLSA	ALE-SVM	25.84	26.38	25.60	24.70	23.13	21.82
	IALE-SVM	27.90	28.83	28.24	27.64	26.03	24.37
	ALE-SF	**28.56**	**29.85**	**30.86**	**31.14**	**31.30**	**31.20**
	IALE-SF	24.27	26.12	27.30	27.82	27.95	28.14
	BSVM	15.22	16.07	16.66	16.80	16.87	16.93

Table 8: MiAP results in NUS-WIDE

smoothness over the samples with respect to the manifold imposed by the considered similarity graph; hence, it is mostly affected in cases where the manifold and clustering assumptions are violated.

4.2.3 ALE versus other methods and SL

One of the goals of this paper is to test whether there are performance benefits for SSL methods against standard supervised learning. The results of Tables 6-9 indicate a marginal improvement in the detection accuracy when ALE is used versus the baseline SVM. Interestingly, there are cases where the performance of the ALE systems exceeds the one of BSVM only for large train-

Yahoo GC								
Descriptor	Method	10K	50K	100K	200K	500K	1M	1.5M
SIFT	ALE-SVM	34.25	34.93	35.0	35.02	35.04	35.05	35.08
	IALE-SVM	34.16	34.94	34.93	35.03	35.08	35.04	35.06
	ALE-SF	**36.71**	**38.67**	**39.07**	**39.33**	**39.51**	**39.54**	**39.55**
	IALE-SF	35.37	37.90	38.50	38.80	38.98	39.07	38.93
	BSVM	35.43	38.02	38.31	38.50	38.32	38.39	38.44
RGB-SIFT	ALE-SVM	35.60	36.07	36.11	36.15	36.20	36.31	36.08
	IALE-SVM	35.61	36.12	36.18	36.30	36.24	36.30	36.34
	ALE-SF	**38.12**	**40.3**	**40.71**	**41.0**	**41.14**	**41.2**	**41.21**
	IALE-SF	37.12	39.70	40.05	40.33	40.38	40.45	40.40
	BSVM	37.60	39.82	40.32	40.43	40.78	40.84	40.90
SURF	ALE-SVM	32.63	33.35	33.22	33.10	33.08	33.05	33.04
	IALE-SVM	32.88	33.47	33.33	33.13	33.08	33.05	33.03
	ALE-SF	37.08	39.55	**40.37**	**40.67**	**40.84**	**40.9**	**40.92**
	IALE-SF	**37.25**	39.75	40.15	40.39	40.53	40.55	40.54
	BSVM	36.13	**39.82**	40.03	40.31	40.5	40.50	40.54

Table 9: MiAP results in Yahoo GC

ing sets, which indicates that such methods can better leverage the availability of large training sets. The performance comparison is complemented by the Table 10, which presents the precision@100 scores. Across almost all tested datasets and features, the best accuracy is attained by ALE-based systems, with the exception being NUS-WIDE with the use of SIFT and SURF, in which BSVM outperforms all ALE systems.

In addition, we compared ALE to competing SSL approaches. A comparison with the GSF approach [28] was already presented in Tables 6 and 7. GSF resulted in higher concept detection accuracy in combination with most features and training set sizes (it was surpassed by ALE-based methods only in the case of MIRF when SIFT or RGB-SIFT were used). However, its computational cost makes GSF highly impractical in realistic settings. For instance, the training step of GSF on MIRF took approximately 2 hours, while for ALE-SVM only 2.5 minutes. Furthermore, we compared ALE-SVM with the LSMP method [6] on the NUS-WIDE dataset using the same features as the ones reported in [6]. When using the full training set, LSMP achieved a MAP score of 0.193, while ALE-SVM scored 0.1866, which is somewhat lower compared to the one of LSMP, but still better than the rest of the four methods reported in [6]. Interestingly, while LSMP needed 31.4 hours for the training, ALE-SVM needed only 4 minutes. Finally, we included LapSVM [4] in the tests, measuring the classification error rate. As we can see from Table 11, ALE-SVM clearly outperforms both LapSVM and BSVM except in the cases where the complete training set (15K images) is given to the algorithms. This is justified by the fact that SSL methods leverage a relatively large test set (10K images), which, in the case of small training sets (1K-10K), greatly contributes to better capture the underlying feature structure. This advantage is mitigated when the size of the test set is smaller than the one of the training set.

MIRF				**ICLEF12**		
Descriptor	Method	P@100		Descriptor	Method	P@100
SIFT	ALE-SVM	0.7075		SIFT	ALE-SVM	**0.3313**
	IALE-SVM	**0.7125**			IALE-SVM	0.3293
	ALE-SF	0.6967			ALE-SF	0.3213
	IALE-SF	0.6913			IALE-SF	0.3205
	BSVM	0.7065			BSVM	0.3115
RGB-SIFT	ALE-SVM	**0.7492**		RGB-SIFT	ALE-SVM	0.3545
	IALE-SVM	0.7488			IALE-SVM	**0.3564**
	ALE-SF	0.7429			ALE-SF	0.3481
	IALE-SF	0.7360			IALE-SF	0.3473
	BSVM	0.7358			BSVM	0.3348
SURF	ALE-SVM	**0.6571**		SURF	ALE-SVM	**0.2812**
	IALE-SVM	0.6567			IALE-SVM	0.2790
	ALE-SF	0.6325			ALE-SF	0.2466
	IALE-SF	0.6338			IALE-SF	0.2482
	BSVM	0.6479			BSVM	0.2617

Yahoo GC				**NUS-WIDE**		
Descriptor	Method	P@100		Descriptor	Method	P@100
SIFT	ALE-SVM	0.5600		SIFT	ALE-SVM	0.3527
	IALE-SVM	0.5620			IALE-SVM	0.3715
	ALE-SF	**0.7490**			ALE-SF	0.3859
	IALE-SF	0.7350			IALE-SF	0.3472
	BSVM	0.6950			BSVM	**0.4611**
SURF	ALE-SVM	0.4930		SURF	ALE-SVM	0.2481
	IALE-SVM	0.5020			IALE-SVM	0.2714
	ALE-SF	0.7530			ALE-SF	0.2912
	IALE-SF	**0.7550**			IALE-SF	0.2849
	BSVM	0.7540			BSVM	**0.3222**
RGB-SIFT	ALE-SVM	0.5790		pLSA	ALE-SVM	0.3274
	IALE-SVM	0.5900			IALE-SVM	0.4355
	ALE-SF	**0.7600**			ALE-SF	**0.6077**
	IALE-SF	0.7560			IALE-SF	0.5653
	BSVM	0.7410			BSVM	0.1304

Flickr2013				**Twitter2013**		
Descriptor	Method	P@100		Descriptor	Method	P@100
SIFT	ALE-SVM	0.6757		SIFT	ALE-SVM	**0.7740**
	IALE-SVM	0.6200			IALE-SVM	0.7737
	ALE-SF	**0.6857**			ALE-SF	0.2278
	IALE-SF	0.6500			IALE-SF	0.7500
	BSVM	0.6500			BSVM	0.7640
SURF	ALE-SVM	**0.5971**		SURF	ALE-SVM	**0.7980**
	IALE-SVM	0.3007			IALE-SVM	0.7880
	ALE-SF	0.3007			ALE-SF	0.7780
	IALE-SF	0.5607			IALE-SF	0.7620
	BSVM	0.5829			BSVM	0.7580

Table 10: Precision at top-100 (P@100) results of ALE, IALE and BSVM when all training samples are used

MIRF						
Descriptor	Method	$1K$	5K	$7.5K$	10K	$15K$
SIFT	ALE-SVM	**13.6226**	**12.4455**	**12.2392**	**12.1486**	12.0629
	LapSVM	15.0545	13.6265	13.2378	12.9867	12.6296
	BSVM	15.332	12.8633	12.3503	12.134	**11.9088**

ICLEF12						
Descriptor	Method	1K	5K	$7.5K$	$10K$	$15K$
SIFT	ALE-SVM	**5.3211**	**5.1406**	**5.1089**	**5.0922**	5.0845
	LapSVM	5.4021	5.271	5.2287	5.1949	5.193
	BSVM	6.2781	5.4268	5.2097	5.1181	**5.0373**

Table 11: Error rate results in relation to the training set size. In all cases, 10K images are used for testing.

4.2.4 Large-scale experiments

One of the primary motivations behind the development and evaluation of the ALE framework was the possibility to train models with very large amounts of samples in very limited time. Here, we compare the performance of IALE with SVM from an efficiency point of view. As illustrated in Table 12, IALE achieves competitive classification accuracy, while achieving a much faster classification speed. More specifically, the execution time of IALE scales linearly to the training set. For example, if a 50K/concept training set is used, IALE

IALE-SF				BSVM			
	Train	*Test*	MiAP		*Train*	*Test*	MiAP
	$B \times B$, g, U	f			Model	Prediction	
1K	5.3 sec	10 mins	0.3537	1K	23 sec	2.5 sec	0.3543
50K	59 sec	10 mins	0.3898	50K	19 mins	2.5 sec	0.3832
150K	3 mins	10 mins	0.3893	150K	71 mins	2.5 sec	0.3844

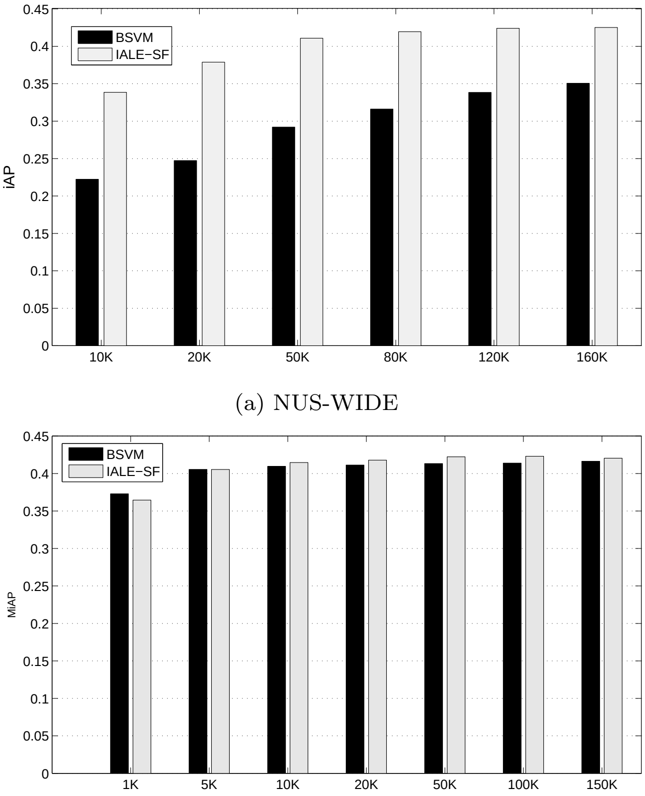
Table 12: Computational cost and accuracy of IALE-SF and linear SVM on Yahoo GC.

needs 59 secs for training and 10 mins to predict the 500K test set (in batches of 1000), while linear SVM needs 19 mins for training and 2.5 secs for prediction. When 150K images/concept are provided as input, ALE needs just 3 mins to compute the training variables ($B \times B$, g and U) and 10 mins for the prediction, while linear SVM needs about 71 mins to learn the model and 2.5 secs for prediction.

To further test the practical value of ALE in online settings, we focus on the performance of the inductive extension (IALE) versus the transductive one. It is observed that IALE results in competitive results in the majority of tests. The difference between the best performing configuration and the IALE ones is in all cases marginal, while there are cases where IALE systems yield the highest accuracy, e.g. in the case of NUS-WIDE with the use of SIFT, and in the case of MIRF and ICLEF12 with the use of GIST.

Fusion: Within IALE, fusion takes place at the level of the LE vectors. Compared to early fusion (i.e. at the level of VLAD vectors), this appears to lead to improved accuracy. As illustrated in Figure 4, IALE outperforms BSVM across all training ratios on NUS-WIDE. For 10K training images, the MiAP of IALE is 33.85%, while for BSVM it is 22.24%. For 160K training images, the MiAP of IALE is 42.52% considerably higher than the 35.07% score of BSVM. In the case of the Yahoo GC dataset, BSVM performs better than IALE for 1K training images/concept, but as the training set increases, IALE improves its accuracy and for 50K training images/concept, it outperforms BSVM.

Concept-level evaluation: Figures 5 and 6 illustrate the iAPs of concepts for NUS-WIDE and Yahoo GC datasets. NUS-WIDE includes 81 concepts. Despite the fact that BSVM outperforms IALE for the majority of concepts, it is interesting to see that IALE outperforms BSVM in many challenging: for instance, concepts *earthquake* and *flags* have a small number of samples (42 and 214 respectively), and the first of those exhibits high variability. We also observe that IALE can predict more accurately concepts containing objects that are conspicuous. For instance, in the concepts *military*, *cars* and *garden* IALE-SF achieves iAP scores of 18.63%, 12.92% and 18.05% respectively, while BSVM achieves 16.9%, 12.08% and 14.87%.



(a) NUS-WIDE

(b) Yahoo GC

Fig. 4: Comparison of MiAP between BSVM and IALE-SF in case of fusion in NUS-WIDE and Yahoo GC.

Figure 6 shows the obtained iAPs on the Yahoo GC dataset that contains 10 general concepts. For the majority of them, we note that the difference in accuracy between BSVM and IALE is only marginal. However, for concept *nature* and *2012*, IALE performs considerably better than BSVM. This is of particular value, since the *2012* concept was among the most challenging ones of the dataset (second lowest MiAP score).

Further insights can be gleaned by the scatter plots of Figures 7 and 8. The x-axis depicts the absolute difference in accuracy (in terms of iAP) when moving from the smallest training set size to the largest one (for IALE), i.e. it indicates the extent to which each concept benefits from the availability of a larger training set. The y-axis depicts the absolute difference in accuracy (in terms of iAP) between IALE and BSVM, i.e. to what extent a particular concept can be better detected by the proposed framework.

By inspecting the plots of the NUS-WIDE dataset, it appears that the large majority of concepts benefit from the availability of bigger training sets. Surprisingly, in the case of the concept *lake* the detection accuracy drops considerably. Furthermore, as mentioned

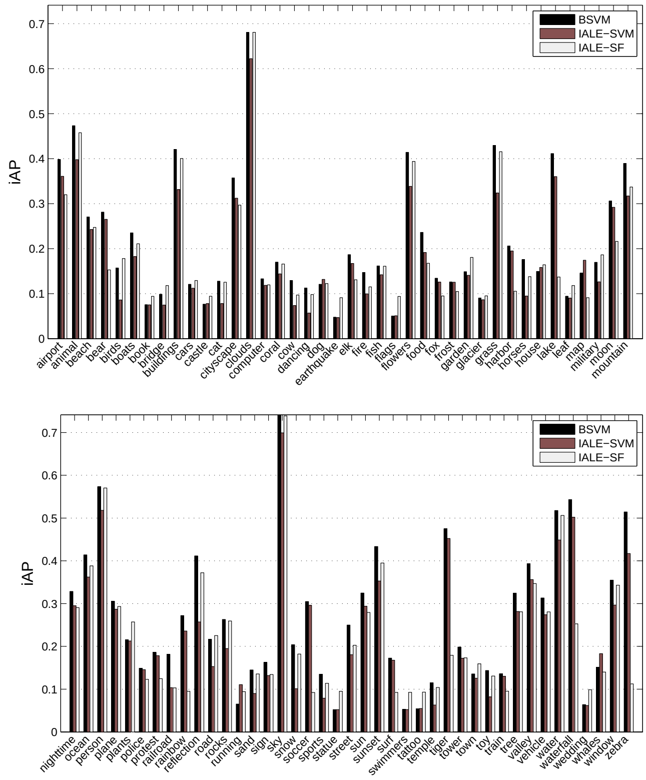


Fig. 5: Comparison of iAPs for the 81 concepts of NUS-WIDE using IALE-SF and BSVM.

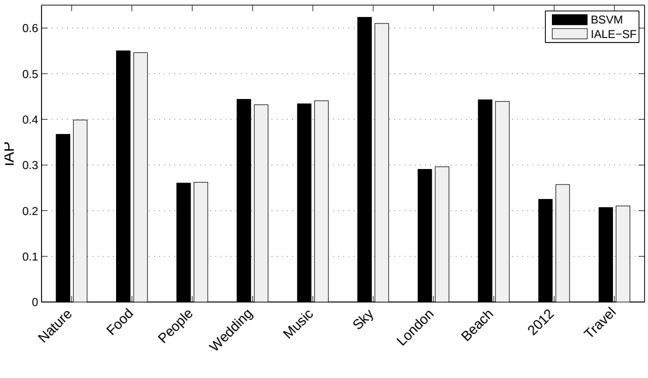


Fig. 6: Comparison of iAPs for the 10 concepts of Yahoo GC using IALE-SF and BSVM

above, for the majority of concepts, BSVM performs somewhat better than IALE. In contrast, inspection of the Yahoo GC plot reveals that IALE detects the majority of concepts with higher accuracy compared to BSVM and that all concepts benefit for the availability of a larger training set.

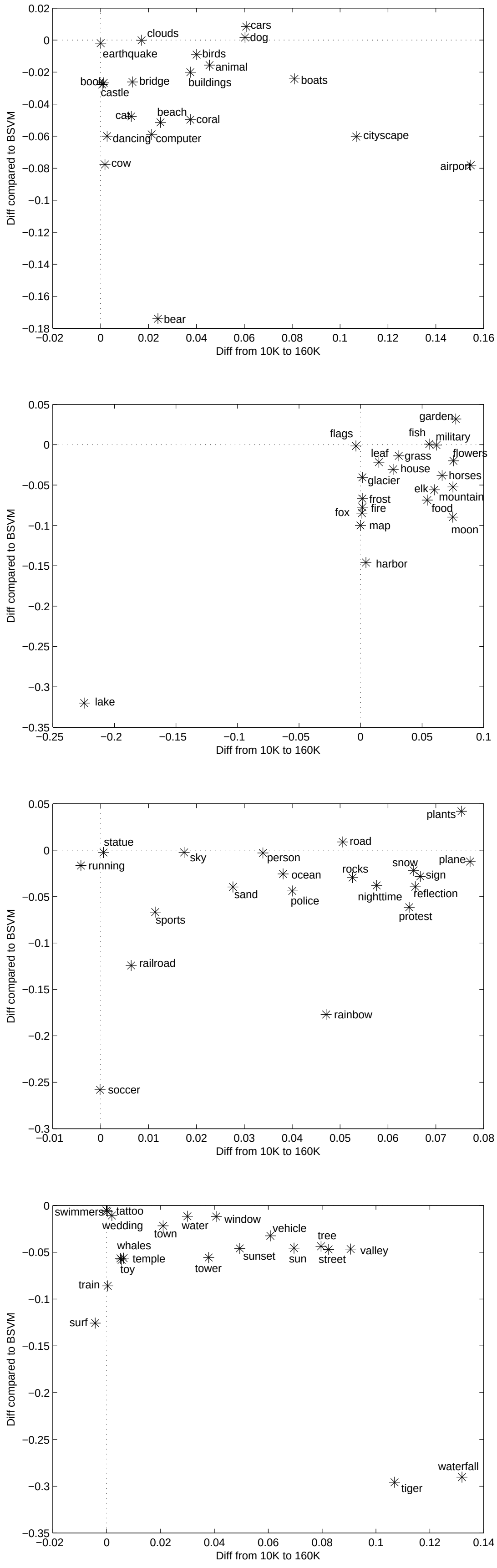


Fig. 7: Per concept analysis of training size impact on NUS-WIDE dataset (the figure is split in four subfigures for readability).

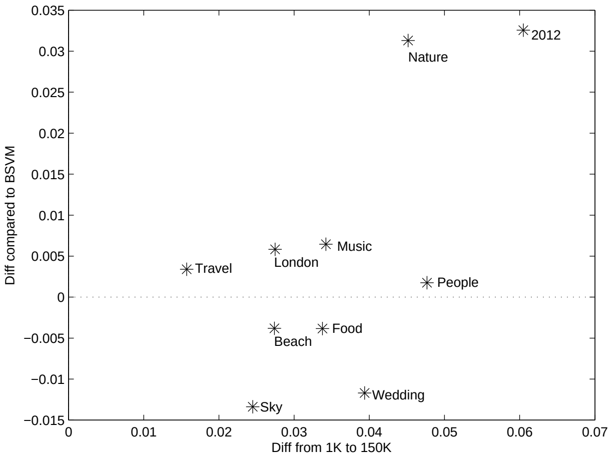
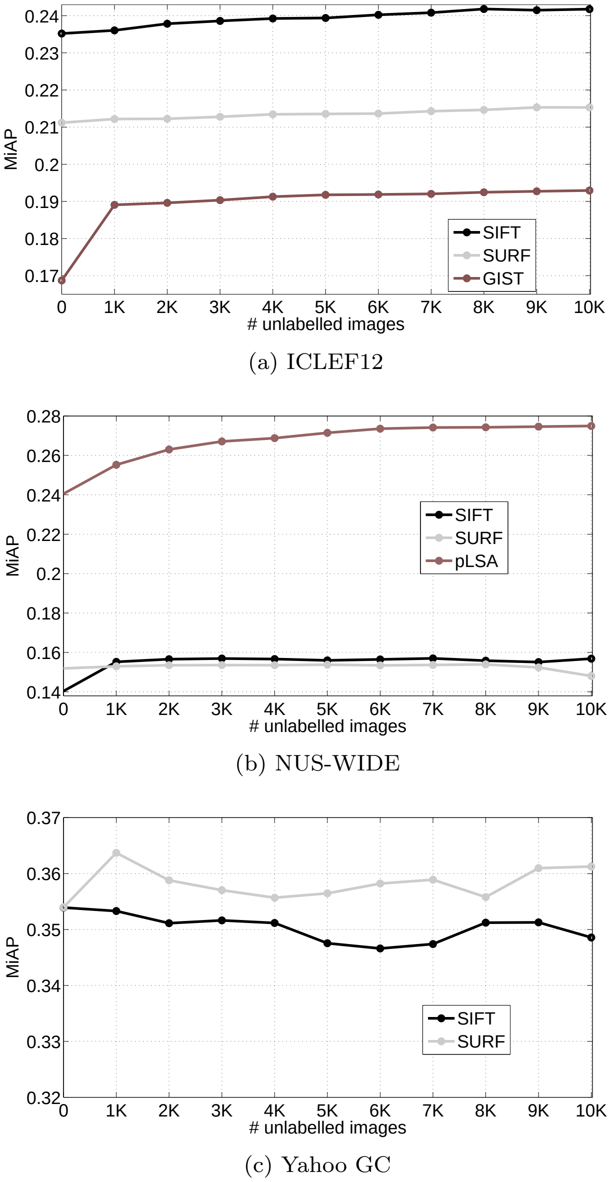


Fig. 8: Per concept analysis of training size impact on Yahoo GC dataset.

4.2.5 Impact of unlabelled data

So far, we have not accounted for the potential of SSL methods to further improve their performance with the inclusion of additional unlabelled images in the learning process. Figure 9 illustrates the gains reaped by IALE when more unlabelled samples are provided as input together with the labelled ones. In ICLEF12 out of the 25K images, 5K were reserved for training and 10K testing. The remaining 10K images were progressively added as unlabelled items. For NUS-WIDE from the 161,789 training images, 10K were reserved for training and the remaining 107,859 for testing. From the rest, we progressively added 1K as unalebelled items. The same pattern was also followed for Yahoo GC, where 10K were kept for training and 500K for testing.

In the case of ICLEF12, the performance benefits for SIFT and SURF are moderate (absolute difference of about 1% between using no unlabelled samples and using 10K samples). For GIST, there is considerable improvement already by adding 1K additional unlabelled samples. In the case of NUS-WIDE the gains are more significant. In particular, when using the tag-based pLSA features, the performance increases by almost 4% between having no unlabelled and adding 10K unlabelled in the learning process. The benefits are less pronounced for SIFT and SURF features. On the other hand, there is no consistent behaviour in the case of Yahoo GC. For a small number of additional unlabelled samples there is a small performance improvement, but when we add more the performance drops by little.



(a) ICLEF12

(b) NUS-WIDE

(c) Yahoo GC

Fig. 9: Impact of unlabelled data on IALE.

5 Conclusion

In this paper, we described a semi-supervised framework for large-scale learning based on the use of Approximate Laplacian Eigenmaps (ALE) in tandem with supervised learning approaches. The algorithm is applicable in both transductive and inductive settings, which makes it practical for applications where real-time indexing of the incoming media items is of importance.

Evaluation on a variety of real datasets shows that a significant speed up is achieved on the training process (compared to a standard supervised classification approach) without noticeable degradation of the detection accuracy. Given also the merits of the ALE representation in terms of storage cost and its amenability for fusing different features, we may conclude that the proposed framework offers an effective and extremely efficient concept detection solution for multimedia indexing and retrieval applications, with a particular focus on large scale settings.

References

1. Balasubramanian, M., Schwartz, E.L.: The isomap algorithm and topological stability. Science **295**(5552) (2002)
2. Bart, T., Adrian, P.: Overview of the clef 2012 flickr photo annotation and retrieval task. in the working notes for the clef 2012 labs and workshop. Rome, Italy (2012)
3. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
4. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research **7**, 2399–2434 (2006)
5. Bengio, Y., Delalleau, O., Roux, N., Paiement, J., Vincent, P., Ouimet, M.: Learning eigenfunctions links spectral embedding and kernel pca. Neural Computation **16**(10), 2197–2219 (2004)
6. Chen, X., Mu, Y., Yan, S., Chua, T.S.: Efficient large-scale image annotation by probabilistic collaborative multi-label propagation. In: Proceedings of the International Conference on Multimedia, MM '10, pp. 35–44. ACM, New York, NY, USA (2010)
7. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.T.: Nus-wide: A real-world web image database from national university of singapore. In: Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09). Santorini, Greece. (July 8-10, 2009)
8. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research **9**, 1871–1874 (2008)
9. Fergus, R., Weiss, Y., Torralba, A.: Semi-supervised learning in gigantic image collections. In: Advances in Neural Information Processing Systems 22, pp. 522–530 (2009)
10. Guillaumin, M., Verbeek, J., Schmid, C.: Multimodal semi-supervised learning for image classification. In: IEEE Conference on Computer Vision & Pattern Recognition, pp. 902 – 909 (2010)
11. Hofmann, T. (ed.): Probabilistic latent semantic analysis, in: Proc. of Uncertainty in Artificial Intelligence. UAI99, Stockholm (1999)
12. Huiskes, M.J., Lew, M.S.: The mir flickr retrieval evaluation. In: Proceedings of the 2008 ACM MIR '08:. ACM, New York, NY, USA
13. Jégou, H., Chum, O.: Negative evidences and co-occurrences in image retrieval: the benefit of PCA and whitening. URL `http://hal.inria.fr/hal-00722622`
14. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: IEEE Conf. on, CVPR, pp. 3304–3311 (2010)
15. Ji, M., Yang, T., Lin, B., Jin, R., Han, J.: A simple algorithm for semi-supervised learning with improved generalization error bound. arXiv:1206.6412 (2012)
16. Jia, P., Yin, J., Huang, X., Hu, D.: Incremental laplacian eigenmaps by preserving adjacent information between data points. Pattern Recognition Letters **30**(16), 1457–1463 (2009)
17. Kong, T., Tian, Y., Shen, H.: A fast incremental spectral clustering for large data sets. In: Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2011 12th International Conference on, pp. 1–5. IEEE (2011)
18. Kouropteva, O., Okun, O., Pietikäinen, M.: Incremental locally linear embedding. Pattern recognition **38**(10), 1764–1767 (2005)
19. Law, M.H., Jain, A.K.: Incremental nonlinear dimensionality reduction by manifold learning. Pattern Analysis and Machine Intelligence, IEEE Transactions on **28**(3), 377–391 (2006)
20. Liu, W., He, J., Chang, S.F.: Large graph construction for scalable semi-supervised learning. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 679–686. Omnipress, Haifa, Israel (2010)
21. Liu, X., Yin, J., Feng, Z., Dong, J.: Incremental manifold learning via tangent space alignment. In: Artificial Neural Networks in Pattern Recognition, pp. 107–121. Springer (2006)
22. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
23. Mantziou, E., Papadopoulos, S., Kompatsiaris, I.: Large-scale semi-supervised learning by approximate laplacian eigenmaps, VLAD and pyramids. In: WIAMIS (2013)
24. Mantziou, E., Papadopoulos, S., Kompatsiaris, Y.: Scalable training with approximate incremental laplacian eigenmaps and pca. In: Proceedings of the 21st ACM International Conference on Multimedia, MM '13, pp. 381–384. ACM, New York, NY, USA (2013)
25. Nadler, B., Lafon, S., Coifman, R.R., Kevrekidis, I.G.: Diffusion maps, spectral clustering and reaction. In: Applied and Computational Harmonic Analysis: Special issue on Diffusion Maps and Wavelets (2006)
26. Ning, H., Xu, W., Chi, Y., Gong, Y., Huang, T.S.: Incremental spectral clustering with application to monitoring of evolving blog communities. In: SDM. SIAM (2007)
27. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal of Computer Vision **42**, 145–175 (2001)
28. Papadopoulos, S., Sagonas, C., Kompatsiaris, I., Vakali, A.: Semi-supervised concept detection by learning the structure of similarity graphs. In: 19th Intern Conf. on MMM (2013)
29. Perronnin, F., Sánchez, J., Liu, Y.: Large-scale image categorization with explicit data embedding. In: CVPR, pp. 2297–2304 (2010)
30. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Empowering visual categorization with the gpu. IEEE Transactions on Multimedia **13**(1), 60–70 (2011)
31. Saul, L.K., Roweis, S.T.: Think globally, fit locally: unsupervised learning of low dimensional manifolds. The Journal of Machine Learning Research **4**, 119–155 (2003)
32. Sinha, K., Belkin, M.: Semi-supervised learning using sparse eigenfunction bases. In: Y. Bengio, D. Schuurmans, J. Lafferty, C.K.I. Williams, A. Culotta (eds.) Advances in Neural Information Processing Systems 22, pp. 1687–1695 (2009)
33. Spyromitros-Xioufis, E., Papadopoulos, S., Kompatsiaris, I., Tsoumakas, G., Vlahavas, I.: An empirical study on the combination of surf features with vlad vectors for image search. In: 13th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), pp. 1–4. IEEE (2012)

34. Talwalkar, A., Kumar, S., Rowley, H.: Large-scale manifold learning. In: IEEE CVPR, 2008., pp. 1–8
35. Tang, J., Yan, S., Hong, R., Qi, G.J., Chua, T.S.: Inferring semantic concepts from community-contributed images and noisy tags. In: Proceedings of the 17th ACM International Conference on Multimedia, MM '09, pp. 223–232. ACM, New York, NY, USA (2009)
36. Wang, H., Huang, H., Ding, C.H.Q.: Image annotation using bi-relational graph of images and semantic labels. In: CVPR, pp. 793–800. IEEE (2011)
37. Wang, M., Hua, X.S.: Beyond distance measurement: Constructing neighborhood similarity for video annotation. pp. 11(3):465–476 (2009)
38. Zhang, K., Kwok, J.T.: Clustered nyström method for large scale manifold learning and dimension reduction. IEEE Transactions on Neural Networks pp. 1576–1587 (2010)
39. Zhang, K., Kwok, J.T., Parvin, B.: Prototype vector machine for large scale semi-supervised learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, pp. 1233–1240. ACM, New York, NY, USA (2009)
40. Zhang, Z.y., Zha, H.y.: Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. Journal of Shanghai University (English Edition) **8**(4), 406–424 (2004)
41. Zheng, J., Yu, H., Shen, F., Zhao, J.: An online incremental learning support vector machine for large-scale data. In: International Conference on Artificial Neural Networks ICANN, Lecture Notes in Computer Science. Springer (2010)
42. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schlkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems 16, pp. 321–328. MIT Press (2004)
43. Zhu, X.: Semi-supervised learning literature survey. Tech. Rep. TR-1530, Computer Sciences, University of Wisconsin-Madison (2008)
44. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: IN ICML, pp. 912–919 (2003)
45. Zhu, X., Kandola, J., Laerty, J., Ghahramani, Z.: Graph Kernels by Spectral Transforms. MIT Press (2006). URL `http://pages.cs.wisc.edu/\~{}jerryzhu/pub/ssl-book.pdf`