A Hybrid Deep Learning Architecture for Latent Topic-based Image Retrieval

K. S. Arun¹ · V. K. Govindan¹

Received: 14 October 2017 / Revised: 15 January 2018 / Accepted: 17 March 2018 / Published online: 2 April 2018
© The Author(s) 2018

Abstract

Learning effective feature descriptors that bridge the semantic gap between low-level visual features directly extracted from image pixels and the corresponding high-level semantics perceived by humans is a challenging task in image retrieval. This paper proposes a hybrid deep learning architecture (HDLA) that generates sparse latent topic-based representation with the objective of minimizing the semantic gap problem in image retrieval. In fact, HDLA has a deep network structure with a constrained replicated Softmax Model in the lower layer and constrained restricted Boltzmann machines in the upper layers. The advantage of HDLA is that there exist nonnegativity restrictions on the model weights together with ℓ_1 -sparsity enforced over the activations of the hidden layer nodes of the network. This, in turn, enhances the modeling power of the network and leads to sparse, parts-based latent topic representation of images. Experimental results on various benchmark datasets show that the proposed model exhibits better generalization ability and the resulting high-level abstraction yields better retrieval performance as compared to state-of-the-art latent topic-based image representation schemes.

Keywords Image retrieval · Deep learning · Latent topics

1 Introduction

The rapid expansion of digital image repositories poses numerous challenges to computer vision research. Among them, the most important one is the development of accurate and efficient mechanisms to search and retrieve desired images from various digital image repositories. Making use of the feature vectors automatically extracted from image pixels together with a suitable similarity measure, Content-Based Image Retrieval (CBIR) systems enable the search and retrieval of images from large repositories that are identical to the given query image. In CBIR domain, the state-of-the-art approaches are based on BoVW model where images are represented as histograms of visual words. Even though the effectiveness of BoVW

model in image retrieval has been proved by many researchers, it still suffers from a major drawback, i.e., the resulting image representation is not as discriminative and descriptive as they are desired to be. This is mainly due to the loss of semantic information of visual words at each processing step of the BoVW model. Therefore, the semantic loss associated with BoVW-based image representation has to be minimized for better retrieval performance.

As the clustering operation in BoVW model often fails to take semantic information into account, there is a high probability that the generated visual dictionary contains many ambiguous visual words. These ambiguous visual words hinder the discriminative power of the BoVW-based image representation. The semantic loss in BoVW model can be reduced to a great extent by automatically grouping semantically similar visual words and then encoding images using these newly identified semantic structures. The work presented in this paper follows the above stated principle to derive a low dimensional but highly discriminative feature vector from the original BoVW-based representation for the task of image retrieval.

✉ K. S. Arun
iamarunks@gmail.com

V. K. Govindan
vkg@nitc.ac.in

¹ Department of Computer Science and Engineering, National Institute of Technology Calicut, Kozhikode 673601, India

It has been observed that visual polysemy and visual synonymy are the root causes behind the induction of ambiguous visual words in the traditional BoVW model. In general, polysemy and synonymy can be regarded as the representational uncertainty of visual information. Polysemy is the characteristic of a visual word that it corresponds to two or more semantic concepts, while synonymy is the characteristic of two or more visual words that they correspond to the same semantic concept. Polysemy originates as a consequence of the visual appearance diversity of different semantic concepts, and it often leads to low inter-semantic discrimination. On the other hand, synonymy arises due to the appearance-based diversity within a particular semantic concept. Thus, if two semantically dissimilar images have a set of polysemous visual words, then they are closer to each other in the visual word-based feature space. Similarly, synonymous visual words may cause images with same semantics to be far apart in the visual word-based feature space. Therefore, to minimize semantic loss and thus to improve the overall retrieval performance of BoVW-based image retrieval, the issue of polysemy and synonymy needs to be effectively tackled.

To mitigate the issue of polysemy and synonymy, researchers proposed to project image representation in the visual word space to an intermediate latent topic space. The underlying idea of latent topics is that not all visual words contain the same amount of information to describe the appearance of images. Therefore, to have better retrieval effectiveness, it is important to use very specific visual words with high discriminative power. This can be achieved by generalizing visual words which share similar meanings to a less specific latent topic. In this way, a set of latent topics $\mathbf{h} = \{h_1, h_2, \dots, h_N\}$ is defined such that a visual word can belong to none, one or several latent

topics. Figure 1 depicts the above-mentioned notion of latent topics in detail. In the end, images are characterized by the proportion of latent topics and this representation is found to be more reliable than the BoVW-based feature while calculating the similarity between images.

As latent topics are learned in a completely unsupervised manner, it is not possible to precisely associate a particular semantic concept to each latent topic. However, images with identical latent topic representations are assumed to contain same semantic concepts and are treated as semantically similar while measuring image similarity. Hence, the notion of latent topics considerably minimizes the semantic loss associated with BoVW model and thus increases the discriminative power of the resulting image representation.

Numerous latent topic-based image retrieval frameworks are available in the literature, and the majority of these approaches are based on graphical models. Approaches based on graphical models try to maximize the joint distribution of visual words and the latent topics to effectively capture the latent topic structures present in the visual word collection. In general, the joint distribution of visual words and latent topics is modeled using a graphical structure. Graphical model-based latent topic frameworks for image retrieval fall into two fundamental categories such as (i) directed topic models and (ii) undirected topic models. The former category involves models based on directed graphs and the most successful approaches toward this direction are Probabilistic Latent Semantic Analysis (PLSA) [1], Latent Dirichlet Allocation (LDA) [2], Correlated Topic Models (CTM) [3] and Pachinko Allocation Model (PAM) [4]. On the contrary, undirected topic modeling frameworks encode the joint distribution by means of undirected graphs. Recently, several undirected

Fig. 1 Pictorial representation of the notion of visual words, latent topics and their interrelationship

topic models have been proposed for image retrieval operation. The most popular among them are Rate Adapting Poisson model (RAP) [5] and Replicated Soft-max Model (RSM) [6].

The major drawback of directed topic modeling schemes is that exact inference is intractable, so they have to rely on approximation algorithms to compute the posterior distribution of latent topics. Another notable limitation is the disjunctive coding principle of directed topic models where they assume a visual word comes from a single latent topic resulting in a suboptimal representation of images. A more accurate latent topic-based image characterization can be obtained with undirected topic models. In general, undirected models are subjected to conjunctive coding principle, and they assume that a visual word always comes from a distribution influenced by all the latent topics. Moreover, accurate and efficient inference techniques have also been developed for undirected models. For these reasons, undirected topic models achieved state-of-the-art performance on large-scale image retrieval as compared to their directed counterparts.

This paper investigates the applicability of an undirected deep network for extracting latent topic-based feature descriptors from images to tackle the semantic loss associated with BoVW representation. To this end, an undirected topic modeling scheme named as Hybrid Deep Learning Architecture (HDLA) is proposed and the latent topic-based image representation obtained with the proposed model yields semantically similar images in response to a given query. In particular, this paper makes the following contributions:

1. A hybrid deep learning architecture which is able to model the higher-order correlations among visual words by employing multiple levels of nonlinear transformations.
2. A compact but discriminative image representation well suited for the retrieval task is obtained by directly imposing nonnegativity regulations on the network weights and ℓ_1 -sparseness constraint on the hidden layer activations.

The rest of this paper is organized as follows: Sect. 2 summarizes the related works in latent topics-based image retrieval. Section 3 presents the background study on Restricted Boltzmann Machine (RBM), Replicated Soft-max Model (RSM) and Deep Boltzmann Machine (DBM). Section 4 explains the proposed latent topic-based image retrieval framework in detail, including the formulation of the proposed HDLA model and the procedure used to obtain the parameters of HDLA. Section 5 delineates how latent topic-based representation is derived from a previously unseen image and the details of the distance metric used for similarity estimation. Section 6 presents the

empirical evaluation of the proposed HDLA model. Finally, the paper is concluded in Sect. 7 by highlighting the advantages of the proposed HDLA model.

2 Related Work

Topic models which automatically analyze and discover latent semantic structures from large image collections have been widely explored in image retrieval domain over the past few years. The basic idea behind topic modeling is the mapping of high-dimensional representation of images in the form of BoVW to a much lower-dimensional space defined by the latent topics. Loosely speaking, a latent topic can be viewed as a set of semantically related visual words. Thus, an image containing a large number of visual words can be concisely modeled using a smaller number of latent topics. This permits the easy estimation of semantic image similarity and consequently helps us to improve the overall retrieval effectiveness. A brief review of the most influential topic modeling schemes in image retrieval research is presented in the rest of this section.

Latent Semantic Analysis (LSA) [7] is regarded as the most primitive topic modeling scheme for semantics-based image retrieval. Pecunovic [8] introduced an LSA-based image modeling framework in which a visual word co-occurrence matrix is initially generated by accumulating the BoVW representation of all the images in the given collection. It is then decomposed into a set of orthogonal factors using Singular Value Decomposition (SVD) with the eigenvectors corresponding to the largest k eigenvalues constitutes the latent topics that represent the relevant semantic structures. When a query image is presented to the system, it is projected into the latent topic space and then the cosine similarity is computed between each indexed images to get a ranked retrieval list. Even though a competent approach, LSA is still computationally intensive. That is, singular value decomposition of the visual word co-occurrence matrix is not practically feasible for large-scale image databases.

Directed topic models have been developed to overcome the above-mentioned limitation of LSA. These models are based on the assumption that each image is a mixture of latent topics and each latent topic, in turn, is a distribution over the visual words. Directed topic models are generally represented with graphical structures comprising a set of random variables. The graphical representation mostly involves two different types of random variables: visible and hidden ones. The visible variables represent visual word count extracted from the given image collection, and the hidden variables capture the semantic structures (latent topics) embedded in these visual words. Then, the directed topic models find an optimal set of latent topics that best

explains the visual words found in the given images. Comprehensive evaluation of various directed topic modeling schemes on large-scale image data sets has shown promising results in terms of retrieval precision and recall.

The last decade has witnessed the emergence of a number of directed topic modeling schemes. The earliest effort in this direction is the Probabilistic Latent Semantic Analysis (PLSA) [1]. Using PLSA, Zhang et al. [9] encoded an image by a probability distribution over latent topics with only a few of them assigned with high probability values. The PLSA model presumes each image as a mixture of a finite number of latent topics. Then, the model fitting involves the estimation of topic specific visual word distributions and image specific latent topic distributions from the given database using Maximum Likelihood Estimation (MLE). Experimental results demonstrated the fact that PLSA-based image modeling schemes have shown to perform remarkably well in large-scale image mining operations.

In order to capture more accurate semantic structures, several research attempts have been made to enhance various aspects of the original PLSA model. With this objective, Lienhart et al. [10] proposed a multilayer PLSA architecture by incorporating not just a single layer of hidden variables, but multiple layers with a hierarchy of variables. Hence, information from various modalities can be efficiently integrated to form more meaningful abstractions. On the other hand, Li et al. [11] introduced correlated PLSA (c-PLSA) which tries to merge inter-image correlations into the basic PLSA formulation and reported promising results in image retrieval tasks. Later on, Chiang et al. [12] proposed Probabilistic Semantic Component Descriptor (PSCD) whereby the latent topics associated with local image regions are initially identified and then integrate this regional semantics together to form a final image descriptor.

However, in PLSA-based image modeling, it is not clear how to infer the topic proportions for an unseen image. That is, the entire model needs to be re-estimated when an image from outside the training dataset is presented as the query. Therefore, the PLSA model and its variants are not scalable. Moreover, the number of parameters to be estimated entirely depends on the size of the image dataset and hence the learned model often tends to overfit the training samples when the number of images in the collection increases linearly.

Later on, Blei et al. [2] formulated a more sophisticated directed topic modeling scheme called Latent Dirichlet Allocation (LDA). Similar to PLSA, LDA assumes that each image is represented by a mixture of fixed number of latent topics and each topic is a mixture over the set of all visual words in the dictionary. In contrast to PLSA, LDA further makes the assumption that these mixture

distributions are Dirichlet-distributed random variables whose parameters have to be estimated from the training data. Therefore, once the parameters of Dirichlet distributions are learned, the topic proportions for an unseen image can be predicted easily which is not the case with PLSA-based models. Moreover, the Dirichlet prior to the per-document topic distribution significantly reduces the effect of overfitting. Horster et al. [13] investigated the applicability of LDA in the context of semantic image modeling and demonstrated its effectiveness in query-by-example-based image retrieval settings.

Due to its good scalability, the LDA model is further extended by many researchers. One such simple extension is the Correlated Topic Model (CTM) [14]. It is similar to LDA except that instead of drawing topic mixture proportions from a Dirichlet distribution, it does so from a logistic normal distribution. Thus, the parameters of CTM involve a covariance matrix whose entries represent the correlation between all pair of latent topics. Greif et al. [14] adopted CTM to explicitly model topic correlation to derive a lower-dimensional latent topic vector and is found to be superior to LDA. As the pairwise correlation of latent topics are modeled by CTM, the number of parameters in the covariance matrix grows as the square of the number of latent topics. Recently, the Pachinko Allocation Model (PAM) [15] emerged as a flexible alternative to CTM. In PAM, the nested correlation among latent topics is efficiently modeled. It does so by extending the concept of latent topics to be distributions not only over the visual words but also over other latent topics. Using image data from large-scale databases, Boulemden and Tilli [4] reported improved performance of PAM-based latent topic representation in image retrieval operation.

It should be noted that inferring posterior distribution of latent topics in directed topic modeling schemes such as LDA and its extensions is typically intractable. In general, approximate inference techniques such as variational methods [16], expectation propagation [17] and Gibbs sampling [18] are utilized to solve this problem. However, these inference algorithms are computationally expensive and time-consuming especially for larger datasets.

Another alternative for topic modeling is the construction of undirected graphical models. As stated before, the visible nodes of undirected graph accept BoVW representation of input images and the hidden nodes indicate the latent topics learned from the given images. In fact, these nodes in undirected topic models are arranged in layers with the visible nodes constitute the first layer and the hidden nodes form the second layer. This layered architecture has an important characteristic that the nodes in one layer are conditionally independent given the values of the nodes in the opposite layer. With this type of architecture, the mapping from input space (i.e., visual words) to latent

topics can be done by a simple matrix multiplication. As a result, the overall retrieval performance, where speed is a primary concern, can be significantly improved. Additionally, undirected models generate distributed latent topic representation and are proven to be superior to the representations obtained with directed topic models for the task of image retrieval.

To date, only a handful of cases have been reported in image retrieval literature using undirected topic models. The Rate Adapting Poisson model (RAP) [5] is one of the earlier works in this direction. In this model, it is assumed that the distribution of the hidden nodes is Binomial and that of the visible nodes is Poisson. Even though RAP-based image retrieval framework performs well in terms of retrieval accuracy, the parameter learning process is unstable and hard. Recently, there has been great interest in using Replicated Softmax Model (RSM) [6] for large-scale image retrieval. It is basically a generalization of Restricted Boltzmann Machine (RBM) [19]. The advantage of using RSM over RAP for deriving high-level image abstractions is that parameter estimation is faster and stable. The Replicated Softmax Model is trained using a fairly efficient learning procedure known as the contrastive divergence algorithm [20]. More importantly, the generalization ability of RSM for unseen images is far better than other models and, this in turn, considerably enhances the overall retrieval performance.

More recently, high-level abstraction of text documents learned using a Deep Boltzmann Machine (DBM)-based formulation called Over Replicated Softmax Model (ORSM) [21] demonstrated promising results for the task of text document classification and retrieval. It has been observed that the high-level abstraction obtained with ORSM has better generalization performance on unseen data as compared to other topic modeling schemes. Encouraged by the recent success of ORSM in modeling text documents, this paper investigates the applicability of an undirected deep learning architecture for extracting efficient latent topic-based representations of images.

To summarize, the effectiveness of topic modeling schemes entirely depends on the quality of the latent topics discovered. It turns out that majority of the above-mentioned models still generate latent topics of inferior quality. This leads to a poor semantic characterization of images and hence degrades the overall retrieval performance. It has been observed that deep network models with many layers of latent topic variables can somehow solve the above-mentioned shortcoming. However, selecting an optimum value for the number of latent topics in each hidden layer is not a straightforward task in such deep models. That is, it should be large enough to fit the characteristics of the image data at hand and at the same time small enough to filter out the irrelevant representational

details. In this scenario, a sparse feature representation [22] where only a few latent topics describe the information that we are anticipating does the trick. Therefore, this paper investigates a hybrid deep learning architecture that generates sparse, parts-based characterization of images using latent topics and is found to be compatible for large-scale image retrieval.

3 Preliminaries

Before the proposed model is introduced, it is important to understand deep learning models which are in fact the stepping stone toward the newly proposed hybrid deep learning architecture. To keep things simple, this section provides a detailed overview of Restricted Boltzmann Machine (RBM) [19] and its special cases such as and Replicated Softmax Model (RSM) [6] and Deep Boltzmann Machine (DBM) [23]. To begin with, RBM is examined by elaborating the contrastive divergence algorithm [20] for deriving the model parameters. Then, the theory behind RSM is outlined, which is useful for modeling visual word-count vectors extracted from images. Finally, the working principle of DBM is explained along with the layer-by-layer training procedure to learn its model parameters.

Let us first introduce the main notations used in this paper. Some of them are used in this section, and the rest are used in subsequent section where the formulation of the proposed HDLA model is described. All these notations are summarized in Table 1.

Table 1 Symbols used in this paper

Symbol	Meaning
K	Visual dictionary size
$\mathbf{v}_{\text{test}}$	BoVW representation of test image
L	Number of hidden layers
T_L	Number of nodes in the L -th hidden layer
$\mathbf{u}$	Visible layer nodes
$\mathbf{h}$	Hidden layer nodes
$\mathbf{b}$	Visible layer bias
$\mathbf{a}$	Hidden layer bias
$\mathbf{W}$	Weight matrix
η	Learning rate
$\sigma(\cdot)$	Sigmoid function
$\mathbb{U}$	Visible layer nodes after weight sharing
$\mathcal{M}$	Number of epochs

3.1 Restricted Boltzmann Machine

A Restricted Boltzmann Machine (RBM) is an undirected probabilistic graphical model-based formulation with a bipartite structure. As depicted in Fig. 2, there exist two layers of binary stochastic units in RBM namely the visible layer $\mathbf{u} = [u_1, u_2, \dots, u_K]$ and the hidden layer $\mathbf{h} = [h_1, h_2, \dots, h_T]$. The visible layer nodes correspond to observed data, and the nodes in the hidden layer capture the dependencies among the observed data. There is a connection between each node in the visible layer to all the nodes in the hidden layer and vice versa. There is no link between the nodes within the same layer. In its standard form, the visible and hidden layer units of RBM are binary-valued. That is, the space of visible vectors for a binary RBM is $\mathbf{u} = \{0, 1\}^K$, while the space of hidden unit vectors is $\mathbf{h} = \{0, 1\}^T$. Associated with each nodes in the visible and the hidden layers, there exist bias units and the corresponding bias offsets are represented by $\mathbf{b} = [b_1, b_2, \dots, b_K]$ and $\mathbf{a} = [a_1, a_2, \dots, a_T]$.

The interaction between a visible layer node i and a hidden layer node j is quantified by a real-valued weight w_{ij} . The pairwise weights between all the elements of $\mathbf{u}$ and $\mathbf{h}$ are generally summarized by a symmetric weight matrix $\mathbf{W}$. It is important to note that RBMs are special cases of Energy-Based Models (EBM), in which the relationships among variables are modeled by assigning energy values to each of their joint configurations. Then, the model parameters of RBM are learned by minimizing the energy of all the desirable configurations of the state space vectors. The following function computes the energy value for the joint configuration of visible and hidden layer nodes $(\mathbf{u}, \mathbf{h})$:

$$\begin{aligned} \mathcal{E}(\mathbf{u}, \mathbf{h}; \Theta) &= -\mathbf{u}^T \mathbf{W} \mathbf{h} - \mathbf{b}^T \mathbf{u} - \mathbf{a}^T \mathbf{h} \\ &= -\sum_{i=1}^K \sum_{j=1}^T u_i h_j w_{ij} - \sum_{i=1}^K b_i u_i - \sum_{j=1}^T a_j h_j \end{aligned} \quad (1)$$

where $\Theta = [\mathbf{W}, \mathbf{b}, \mathbf{a}]$ is the model parameter vector. Based on this energy function, the model can further assign

Fig. 2 Restricted Boltzmann machine [19]

probabilities to every possible state vector pairs of visible and hidden units. This joint distribution is defined by:

$$p(\mathbf{u}, \mathbf{h}; \Theta) = \frac{1}{\mathcal{Z}(\Theta)} \exp(-\mathcal{E}(\mathbf{u}, \mathbf{h}; \Theta)) \quad (2)$$

where $\mathcal{Z}(\Theta)$ is a normalization constant also known as the partition function. The value of $\mathcal{Z}(\Theta)$ is computed as follows:

$$\mathcal{Z}(\Theta) = \sum_{\mathbf{u}} \sum_{\mathbf{h}} \exp(-\mathcal{E}(\mathbf{u}, \mathbf{h}; \Theta)) \quad (3)$$

Similarly, the model can assign probability to the visible vector $\mathbf{u}$ in the following fashion:

$$p(\mathbf{u}; \Theta) = \frac{1}{\mathcal{Z}(\Theta)} \sum_{\mathbf{h}} \exp(-\mathcal{E}(\mathbf{u}, \mathbf{h})) \quad (4)$$

Because of the bipartite structure of RBM, the conditional distribution over visible vector $\mathbf{u}$ and hidden units $\mathbf{h}$ can be easily derived from Eq. (2) and is given by:

$$p(\mathbf{u} | \mathbf{h}; \Theta) = \prod_{i=1}^K p(u_i | \mathbf{h}) \quad (5)$$

$$p(\mathbf{h} | \mathbf{u}; \Theta) = \prod_{j=1}^T p(h_j | \mathbf{u}) \quad (6)$$

where the individual activation probabilities $p(h_j | \mathbf{u})$, $p(u_i | \mathbf{h})$ are defined as follows:

$$p(u_i = 1 | \mathbf{h}) = \sigma(b_i + \sum_{j=1}^T w_{ij} h_j) \quad (7)$$

$$p(h_j = 1 | \mathbf{u}) = \sigma(a_j + \sum_{i=1}^K w_{ij} u_i) \quad (8)$$

where $\sigma(\cdot)$ is the logistic sigmoid function defined as $\sigma(y) = 1/(1 + \exp(-y))$.

Thus, RBM is a powerful generative model capable to capture the covariance structure present in the given input observations in a completely unsupervised fashion. This helps to group semantically similar visual words into a relatively small number of latent topics, and thus a more efficient latent topic-based image characterization can be derived with RBM-based image modeling. The next section provides a detailed description of the training procedure used to learn the model parameters of RBM.

3.1.1 Contrastive Divergence Algorithm

The Restricted Boltzmann Machine is trained in such a way that the obtained model parameter Θ should minimize the negative log-likelihood of the given training data set. Let $\mathcal{D} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N\}$ be the set of independent and

identically distributed training samples, then the log-likelihood of $\mathcal{S}$ is given by:

$$\ell(\mathcal{D}; \Theta) = \ln \prod_{i=1}^N p(\mathbf{u}_i; \Theta) = \sum_{i=1}^N \ln p(\mathbf{u}_i; \Theta) \quad (9)$$

The unknown parameter vector Θ of the RBM is then learned by solving the following optimization problem:

$$\underset{\Theta}{\operatorname{argmin}} \sum_{i=1}^N -\ln p(\mathbf{u}_i; \Theta) \quad (10)$$

The stochastic gradient descent procedure is then used to optimize the model parameter values. The gradient decent procedure updates the parameter vector Θ as:

$$\Theta^{(m+1)} = \Theta^{(m)} + \Delta\Theta \quad (11)$$

where m is the number of epoch, and it indicates the total presentations of the full training set to the learning algorithm. $\Delta\Theta$ is the change in the parameter vector Θ . In each epoch, $\Delta\Theta$ is initialized to zero and subsequently changed in a direction that minimizes the negative log-likelihood as shown below:

$$\Delta\Theta = -\eta \frac{\partial \ell(\Theta)}{\partial \Theta} \quad (12)$$

where η is the learning rate, and it indicates the relative size of the changes in the parameter vector Θ . For the model defined in Eq. (1), the gradient of the log-likelihood given a single training example $\mathbf{u}_s$ is given by:

$$\begin{aligned} \frac{\partial}{\partial \Theta} \ell(\mathbf{u}_s; \Theta) &= \frac{\partial}{\partial \Theta} \left[\ln p(\mathbf{u}_s; \Theta) \right] \\ &= \frac{\partial}{\partial \Theta} \left[\ln \frac{1}{Z(\Theta)} \sum_{\mathbf{h}} \exp(-\mathcal{E}(\mathbf{u}_s, \mathbf{h})) \right] \\ &= \frac{\partial}{\partial \Theta} \left[\ln \sum_{\mathbf{h}} \exp(-\mathcal{E}(\mathbf{u}_s, \mathbf{h})) \right] - \frac{\partial}{\partial \Theta} \left[\ln \sum_{\mathbf{u}} \sum_{\mathbf{h}} \exp(-\mathcal{E}(\mathbf{u}, \mathbf{h})) \right] \\ &= \frac{-1}{\sum_{\mathbf{h}} \exp(-\mathcal{E}(\mathbf{u}_s, \mathbf{h}))} \sum_{\mathbf{h}} \exp(-\mathcal{E}(\mathbf{u}_s, \mathbf{h})) \frac{\partial}{\partial \Theta} \mathcal{E}(\mathbf{u}_s, \mathbf{h}) \\ &\quad + \frac{1}{\sum_{\mathbf{u}} \sum_{\mathbf{h}} \exp(-\mathcal{E}(\mathbf{u}, \mathbf{h}))} \sum_{\mathbf{u}} \sum_{\mathbf{h}} \exp(-\mathcal{E}(\mathbf{u}, \mathbf{h})) \frac{\partial}{\partial \Theta} \mathcal{E}(\mathbf{u}, \mathbf{h}) \\ &= -\sum_{\mathbf{h}} p(\mathbf{h} | \mathbf{u}_s) \frac{\partial}{\partial \Theta} \mathcal{E}(\mathbf{u}_s, \mathbf{h}) \\ &\quad + \sum_{\mathbf{u}} \sum_{\mathbf{h}} p(\mathbf{u}, \mathbf{h}) \frac{\partial}{\partial \Theta} \mathcal{E}(\mathbf{u}, \mathbf{h}) \end{aligned} \quad (13)$$

Therefore, the gradient of the log-likelihood is the difference between two expectations. The first term of Eq. (13) is the expectation of the gradient of the energy function with respect to $p(\mathbf{h} | \mathbf{u}_s)$ and is termed as data-dependent expectation. Similarly, the second term is the expectation of the gradient of the energy function with respect to

$p(\mathbf{u}, \mathbf{h})$ and is known as model-dependent expectation. As both the terms involve expectations, the gradient of the log-likelihood can be rewritten as:

$$\frac{\partial}{\partial \Theta} \ell(\mathbf{u}_s; \Theta) = -\mathbb{E}_{p(\mathbf{h} | \mathbf{u}_s)} \left[\frac{\partial}{\partial \Theta} \mathcal{E}(\mathbf{u}_s, \mathbf{h}) \right] + \mathbb{E}_{p(\mathbf{u}, \mathbf{h})} \left[\frac{\partial}{\partial \Theta} \mathcal{E}(\mathbf{u}, \mathbf{h}) \right] \quad (14)$$

where the shorthand notation $\mathbb{E}_{p(\mathbf{h} | \mathbf{u}_s)}[\cdot]$ denotes the data-dependent expectation and $\mathbb{E}_{p(\mathbf{u}, \mathbf{h})}[\cdot]$ represents the model-dependent expectation. The derivative of the negative energy function with respect to all the model parameters $\Theta = [\mathbf{W}, \mathbf{b}, \mathbf{a}]$ can easily be computed as follows:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{W}} (-\mathcal{E}(\mathbf{u}, \mathbf{h})) &= \frac{\partial}{\partial \mathbf{W}} \mathbf{u}^T \mathbf{W} \mathbf{h} = \mathbf{u} \mathbf{h}^T \\ \frac{\partial}{\partial \mathbf{a}} (-\mathcal{E}(\mathbf{u}, \mathbf{h})) &= \frac{\partial}{\partial \mathbf{a}} \mathbf{a}^T \mathbf{h} = \mathbf{h} \\ \frac{\partial}{\partial \mathbf{b}} (-\mathcal{E}(\mathbf{u}, \mathbf{h})) &= \frac{\partial}{\partial \mathbf{b}} \mathbf{b}^T \mathbf{u} = \mathbf{u} \end{aligned} \quad (15)$$

Now the derivative of the log-likelihood of a given training pattern $\mathbf{u}_s$ with respect to the weights $\mathbf{W}$, visible layer bias $\mathbf{b}$ and hidden layer bias $\mathbf{a}$ becomes:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{W}} \ell(\mathbf{u}_s; \mathbf{W}) &= \mathbb{E}_{p(\mathbf{h} | \mathbf{u}_s)} [\mathbf{u}_s \mathbf{h}^T] - \mathbb{E}_{p(\mathbf{u}, \mathbf{h})} [\mathbf{u}_s \mathbf{h}^T] \\ \frac{\partial}{\partial \mathbf{a}} \ell(\mathbf{u}_s; \mathbf{a}) &= \mathbb{E}_{p(\mathbf{h} | \mathbf{u}_s)} [\mathbf{h}] - \mathbb{E}_{p(\mathbf{u}, \mathbf{h})} [\mathbf{h}] \\ \frac{\partial \ell(\mathbf{u}_s; \mathbf{b})}{\partial \mathbf{b}} &= \mathbb{E}_{p(\mathbf{h} | \mathbf{u}_s)} [\mathbf{u}_s] - \mathbb{E}_{p(\mathbf{u}, \mathbf{h})} [\mathbf{u}_s] \end{aligned} \quad (16)$$

The conditional independence property of RBM ensures an easy estimation of the data-dependent expectation. On the other hand, the model-dependent expectation involves a sum over all 2^K elements of $\mathbf{u}$ as well as the 2^T elements of $\mathbf{h}$. Therefore, exact computation of the data-dependent expectation is intractable because its complexity is exponential in the number of visible and hidden layer nodes. To avoid this computational burden, the data-dependent expectation can be approximated by generating a finite number of samples from the joint distribution $p(\mathbf{u}, \mathbf{h})$ using the Markov Chain Monte Carlo (MCMC) [24] technique.

The classical MCMC approach makes use of Gibbs sampling [18] to generate samples from a joint distribution of multiple random variables. The basic idea is to construct a Markov chain by updating each random variable based on its conditional distribution, given the state of the others. That is, to get a sample from a joint distribution $p(\dagger_1, \dagger_2, \dots, \dagger_c)$ of c random variables, Gibbs sampling performs a sequence of r sampling steps of the form $\dagger_i \sim P(\dagger_i | \dagger_{-i})$, where $\dagger_{-i}$ represents the ensemble of the $(c-1)$ random variables other than $\dagger_i$. Since an RBM consists of conditionally independent visible and hidden units, Gibbs sampling can be easily applied to get samples from the joint distribution $p(\mathbf{u}, \mathbf{h})$. The variables in the

hidden layer units are sampled simultaneously given fixed values for the variables in the visible layer. Similarly, visible layer variables are sampled simultaneously given the hidden layer variables. Thus, step (t) of the Gibbs sampling process for the RBM defined in Eq. (2) has the following two phases:

$$\begin{aligned} h_j^{(t)} &\sim p(h_j \mid \mathbf{u}^{(t-1)}) \\ u_i^{(t)} &\sim p(u_i \mid \mathbf{h}^{(t)}) \end{aligned} \quad (17)$$

where $\mathbf{h}^{(t)}$, $\mathbf{u}^{(t-1)}$ refers to the set of all hidden and visible layer units at steps (t) and $(t-1)$ of the Gibbs sampling procedure. Similarly, $h_j^{(t)}$, $u_i^{(t)}$ are the j -th hidden layer unit and the i -th visible layer unit of the model at step (t) of the Gibbs sampling procedure. It is assumed that as $t \rightarrow \infty$, *Gibbs sampling* is guaranteed to generate accurate samples of $p(\mathbf{u}, \mathbf{h})$.

Once sufficient number of samples are obtained with Gibbs sampling, the Monte Carlo approach can be used to approximate the model-dependent expectation specified in Eq. (14). Let $\{(\mathbf{u}^1, \mathbf{h}^1), (\mathbf{u}^2, \mathbf{h}^2), \dots, (\mathbf{u}^n, \mathbf{h}^n)\}$ be a set of samples drawn from $p(\mathbf{u}, \mathbf{h})$ using the above-mentioned

Gibbs sampling process, then the Monte Carlo approximation of $\mathbb{E}_{p(\mathbf{u}, \mathbf{h})} \left[\frac{\partial}{\partial \theta} \mathcal{E}(\mathbf{u}, \mathbf{h}) \right]$ is given by:

$$\mathbb{E}_{p(\mathbf{u}, \mathbf{h})} \left[\frac{\partial}{\partial \theta} \mathcal{E}(\mathbf{u}, \mathbf{h}) \right] \approx \frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial \theta} \mathcal{E}(\mathbf{u}^i, \mathbf{h}^i) \quad (18)$$

Consequently, the derivative of the log-likelihood for the given training sample $\mathbf{u}_s$ can be approximated by:

$$\frac{\partial}{\partial \theta} \ell(\mathbf{u}_s; \theta) \approx - \sum_{\mathbf{h}} p(\mathbf{h} \mid \mathbf{u}_s) \frac{\partial}{\partial \theta} \mathcal{E}(\mathbf{u}_s, \mathbf{h}) + \frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial \theta} \mathcal{E}(\mathbf{u}^i, \mathbf{h}^i) \quad (19)$$

However, obtaining unbiased samples from RBM distribution using MCMC method typically requires many sampling steps. As a result, the computation of log-likelihood remains intractable for large-scale image data sets. Recently, it has been shown that running the Markov chain for just a few steps is sufficient for estimating the log-likelihood gradient specified in Eq. (19). This leads to Contrastive Divergence (CD) algorithm [20] and is now the most commonly used method for RBM training.

Algorithm 1 Contrastive divergence learning procedure

Input: $\text{RBM}(u_1, u_2, \dots, u_K; h_1, h_2, \dots, h_T)$, training set $\mathcal{D} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N\}$ of input observation, total number of steps ℓ in Gibbs sampling, number of epochs $\mathcal{M}$, learning rate η .

Output: Model parameters of RBM.

```

1: Initialize  $w_{ij}, b_i, a_j$  with random values;  $\forall i \in \{1, 2, \dots, K\}$  and  $\forall j \in \{1, 2, \dots, T\}$ 
2:  $t=1$ ;
3: for  $m=1$  to  $\mathcal{M}$  do
4:   Initialize  $\Delta w_{ij} = 0; \Delta b_i = 0; \Delta a_j = 0; \forall i \in \{1, 2, \dots, K\}$  and  $\forall j \in \{1, 2, \dots, T\}$ 
5:   for all  $\mathbf{u}_s \in \mathcal{D}$  do
6:      $\mathbf{u}^{(0)} \leftarrow \mathbf{u}_s$ 
7:     for  $t=1$  to  $\ell$  do
8:       for  $j=1$  to  $T$  do
9:         Sample  $h_j^{(t)} \sim p(h_j | \mathbf{u}^{(t-1)})$ 
10:      end for
11:      for  $i=1$  to  $K$  do
12:        Sample  $u_i^{(t)} \sim p(u_i | \mathbf{h}^{(t)})$ 
13:      end for
14:    end for
15:    for  $i=1$  to  $K$  do
16:      for  $j=1$  to  $T$  do
17:         $\Delta w_{ij} \leftarrow \Delta w_{ij} + \eta \left( p(h_j = 1 | \mathbf{u}^{(0)}) u_i^{(0)} - p(h_j = 1 | \mathbf{u}^{(\ell)}) u_i^{(\ell)} \right)$ 
18:      end for
19:    end for
20:    for  $i=1$  to  $K$  do
21:       $\Delta b_i \leftarrow \Delta b_i + \eta \left( u_i^{(0)} - u_i^{(\ell)} \right)$ 
22:    end for
23:    for  $j=1$  to  $T$  do
24:       $\Delta a_j \leftarrow \Delta a_j + \eta \left( p(h_j = 1 | \mathbf{u}^{(0)}) - p(h_j = 1 | \mathbf{u}^{(\ell)}) \right)$ 
25:    end for
26:  end for
27:  for  $i=1$  to  $K$  do
28:    for  $j=1$  to  $T$  do
29:       $w_{ij} = w_{ij} + \Delta w_{ij}$ 
30:    end for
31:  end for
32:  for  $i=1$  to  $K$  do
33:     $b_i = b_i + \Delta b_i$ 
34:  end for
35:  for  $j=1$  to  $T$  do
36:     $a_j = a_j + \Delta a_j$ 
37:  end for
38: end for

```

Instead of waiting for the Gibbs chain to converge, the $\parallel$ -step Contrastive Divergence ($\text{CD}_{\parallel}$) algorithm runs the chain for only $\parallel$ steps. That is, the chain starts from an input observation $\mathbf{u}_s$ of the training set (i.e., $\mathbf{u}^{(0)} = \mathbf{u}_s$) and yields the sample $\mathbf{u}^{(\parallel)}$ by performing $\parallel$ steps of Gibbs sampling. Each step t of $\text{CD}_{\parallel}$ consists of sampling $\mathbf{h}^{(t)}$ from $p(\mathbf{h} | \mathbf{u}^{(t-1)})$ and then sampling $\mathbf{u}^{(t)}$ from $p(\mathbf{u} | \mathbf{h}^{(t)})$. Finally, the gradient in Eq. (19) can be written as:

$$\begin{aligned}
\text{CD}_{\parallel}(\Theta, \mathbf{u}^{(0)} = \mathbf{u}_s) &= - \sum_{\mathbf{h}} p(\mathbf{h} | \mathbf{u}^{(0)}) \frac{\partial}{\partial \Theta} \mathcal{E}(\mathbf{u}^{(0)}, \mathbf{h}) \\
&+ \sum_{\mathbf{h}} p(\mathbf{h} | \mathbf{u}^{(\parallel)}) \frac{\partial}{\partial \Theta} \mathcal{E}(\mathbf{u}^{(\parallel)}, \mathbf{h})
\end{aligned}
\tag{20}$$

Hinton et al. [20] empirically found that the learning algorithm converges closer to the exact maximum likelihood even for small values of $\parallel$ (often just one step). A batch-based version of $\text{CD}_{\parallel}$ has been presented in Algorithm 1. In batch-based training protocol, all input observations are presented to the model before the parameter update takes place. The algorithm makes several epochs through the training data so as to get a final estimate of the parameter vector Θ . For an input observation $\mathbf{u}_s$ of the training set (i.e., $\mathbf{u}^{(0)} = \mathbf{u}_s$), the following rules are used by the $\parallel$ -step Contrastive Divergence algorithm to update the weights and biases of the model.

$$\begin{aligned}
\Delta w_{ij} &= \Delta w_{ij} + \eta \left(p(h_j = 1 | \mathbf{u}^{(0)}) u_i^{(0)} - p(h_j = 1 | \mathbf{u}^{(\parallel)}) u_i^{(\parallel)} \right) \\
\Delta b_i &= \Delta b_i + \eta \left(u_i^{(0)} - u_i^{(\parallel)} \right) \\
\Delta a_j &= \Delta a_j + \eta \left(p(h_j = 1 | \mathbf{u}^{(0)}) - p(h_j = 1 | \mathbf{u}^{(\parallel)}) \right)
\end{aligned} \quad (21)$$

where $\eta > 0$ is the learning rate of RBM.

Once the unknown parameters are estimated, RBM generates a T -dimensional latent topic-based representation $p(\mathbf{h} | \mathbf{u}_{\text{new}})$ for an unseen input $\mathbf{u}_{\text{new}}$ supplied to the model. The newly generated feature vector provides a quantitative description of the latent topic structure associated with the unseen input $\mathbf{u}_{\text{new}}$. Moreover, the dimensionality of the obtained representation is considerably lower than that of the actual input. All these characteristics make RBM an ideal tool for latent topic-based image modeling.

3.2 Replicated Softmax Model

From the previous section, it is well understood that RBMs only deal with input observations from a Bernoulli distribution. While modeling an image characterized by a visual dictionary, we are interested in the occurrence frequency of visual words in the given image. However, the visual word-count vectors cannot be modeled by RBMs with binary-valued (*Bernoulli*) input units. Therefore, Salakhutdinov and Hinton [6] proposed Replicated Softmax Model (RSM) as a variant of RBM to model visual word-count data. The nodes in the visible layer are modeled as Softmax units and can have one of many different states. A graphical representation of the RSM framework is depicted in Fig. 3a. Let K be the size of the visual dictionary learned from a set of training images and N be the number of interest points detected in the given image, then the input data to the RSM

model is an $N \times K$ binary matrix $\mathbf{U}$ with $U_{ik} = 1$ if and only if the i -th interest point in the given image is assigned to the k -th visual word and is given by:

$$\mathbf{u} = \begin{bmatrix} U_{11} & U_{12} & U_{13} & \dots & U_{1K} \\ U_{21} & U_{22} & U_{23} & \dots & U_{2K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ U_{N1} & U_{N2} & U_{N3} & \dots & U_{NK} \end{bmatrix} \quad (22)$$

Let $\mathbf{h} \in \{0, 1\}^T$ be the binary stochastic latent topic feature, then the energy of the RSM model for the configuration $\{\mathbf{u}, \mathbf{h}\}$ is defined as:

$$\begin{aligned}
\mathcal{E}(\mathbf{u}, \mathbf{h}; \Theta) &= - \sum_{n=1}^N \sum_{j=1}^T \sum_{i=1}^K W_{ijn} h_j U_{ni} \\
&\quad - \sum_{n=1}^N \sum_{i=1}^K U_{ni} b_{ni} - \sum_{j=1}^T h_j a_j
\end{aligned} \quad (23)$$

where $\Theta = [\mathbf{W}, \mathbf{a}, \mathbf{b}]$ are the model parameters in which $\mathbf{W} = [W_{ijn}]$ denotes the connection strength between the i -th visible layer unit corresponding to the n th interest point in the given image and the j -th hidden layer unit. $\mathbf{b} = [b_{ni}]$ is the bias associated with the i th visible unit of the n th interest point in the given image and $\mathbf{a}$ is the bias of the hidden layer $\mathbf{h}$.

The concept of weight sharing simplifies the basic formulation of RSM specified in Eq. (23). Weight sharing ignores the sequence in which local descriptors occurs in the input image. That is, if the i th visible unit of the n th local image descriptor is forced to share its weight with the i th visible unit of all other local descriptors, then W_{ijn} can be simply redefined as W_{ij} . This procedure is illustrated in Fig. 3b. With this modification, the input binary matrix $\mathbf{U}$ of the RSM framework can be replaced with K visible layer nodes $\mathbb{U} = [\mathbb{u}_1, \mathbb{u}_2, \dots, \mathbb{u}_K]$ each of them corresponds to a distinct visual word in the learned dictionary. The nodes in the visible layer $\mathbb{U}$ are shown using concentric circles to indicate replication, i.e. the number of times each visual word occurs in the given image. The weight sharing operation brings down the total number of parameters to be learned from $(N \times T \times K)$ to $(T \times K)$, and it helps RSM to model images with a varying number of visual words. The energy of the configuration $\{\mathbb{U}, \mathbf{h}\}$ after weight sharing is then defined as:

$$\mathcal{E}(\mathbb{U}, \mathbf{h}; \Theta) = - \sum_{j=1}^T \sum_{i=1}^K W_{ij} h_j \hat{u}_i - \sum_{i=1}^K \hat{u}_i b_i - N \sum_{j=1}^T h_j a_j \quad (24)$$

Fig. 3 Graphical interpretation of RSM (a) without weight sharing (b) with weight sharing [6]

where $\hat{w}_i = \sum_{n=1}^N U_{ni}$ denotes the frequency with which the i -th visual word appears in the given image. It should be noted that the bias term for the hidden unit is scaled by the number of interest points N . This scaling is crucial as it allows hidden units to behave sensibly when dealing with documents of different lengths. Then, the probability that the model assigns to a visible binary matrix $\mathbb{U}$ is given by:

$$p(\mathbb{U}; \Theta) = \frac{1}{\mathcal{Z}(\Theta)} \sum_{\mathbf{h}} \exp(-\mathcal{E}(\mathbb{U}, \mathbf{h}; \Theta)) \quad (25)$$

where $\mathcal{Z}(\Theta)$ is known as the partition function and is defined as:

$$\mathcal{Z}(\Theta) = \sum_{\mathbb{U}} \sum_{\mathbf{h}} \exp(-\mathcal{E}(\mathbb{U}, \mathbf{h}; \Theta)) \quad (26)$$

The conditional probabilities of visual words and latent topics are expressed in terms of Softmax and logistic sigmoid functions defined as follows:

$$p(u_i = 1 | \mathbf{h}) = \frac{\exp(b_i + \sum_{j=1}^T W_{ij} h_j)}{\sum_{k=1}^K \exp(b_k + \sum_{j=1}^T W_{kj} h_j)} \quad (27)$$

$$p(h_j = 1 | \mathbf{u}) = \sigma\left(Na_j + \sum_{i=1}^K \hat{w}_i W_{ij}\right) \quad (28)$$

The major advantage of using Softmax units in RSM is that the principle behind parameter estimation remains the same as that of RBM. Thus, the weights and bias of RSM are optimized by applying the contrastive divergence algorithm to the log-likelihood gradient. By following the same conventions as used in RBM, the update rules for the model parameters of RSM can be derived as follows:

$$\begin{aligned} \Delta W_{ij} &= \Delta W_{ij} + \eta \left(p(h_j = 1 | \mathbb{U}^{(0)}) \hat{w}_i^{(0)} - p(h_j = 1 | \mathbb{U}^{(\parallel)}) \hat{w}_i^{(\parallel)} \right) \\ \Delta b_i &= \Delta b_i + \eta \left(\hat{w}_i^{(0)} - \hat{w}_i^{(\parallel)} \right) \\ \Delta a_j &= \Delta a_j + \eta \left(p(h_j = 1 | \mathbb{U}^{(0)}) - p(h_j = 1 | \mathbb{U}^{(\parallel)}) \right) \end{aligned} \quad (29)$$

where $\mathbb{U}^{(0)} = [\hat{w}_1^{(0)}, \hat{w}_2^{(0)}, \dots, \hat{w}_K^{(0)}]$ is an input observation from the training set from which the Gibbs chain starts and $\mathbb{U}^{(\parallel)}$ is the resulting sample after performing $\parallel$ -steps of Gibbs sampling.

3.3 Deep Boltzmann Machine

Similar to RBM, a Deep Boltzmann Machine (DBM) [23] is also an energy-based, undirected graphical model. It is a composite of a single visible layer and multiple hidden layers. It can be viewed as a number of RBMs that are stacked on top of each other. The detailed architecture of a Deep Boltzmann Machine with L hidden layers is shown in

Fig. 4 Graphical representation of deep Boltzmann machine with L hidden layers [23]

Fig. 4. There are connections only between adjacent hidden layer units as well as units in the visible layer and the first hidden layer. Because of the deep hierarchical structure, DBM has greater flexibility and good representation power while modeling complex data distributions. That is, DBM can generate more structured and abstract representations of input observations. Consider a Deep Boltzmann Machine with one input layer $\mathbf{u} = \{u_1, u_2, \dots, u_K\} \in \{0, 1\}^K$ and a series of L hidden layer units $\mathbf{h} = \{\mathbf{h}^{(1)} \in \{0, 1\}^{T_1}, \mathbf{h}^{(2)} \in \{0, 1\}^{T_2}, \dots, \mathbf{h}^{(L)} \in \{0, 1\}^{T_L}\}$. Then, the energy of the joint configuration $\{\mathbf{u}, \mathbf{h}\}$ is defined as:

$$\begin{aligned} \mathcal{E}(\mathbf{u}, \mathbf{h}; \Theta) &= - \sum_{i=1}^K \sum_{j=1}^{T_1} u_i h_j^{(1)} w_{ij} - \sum_{i=1}^K b_i u_i - \sum_{j=1}^{T_1} a_j^{(1)} h_j^{(1)} \\ &\quad + \sum_{\ell=1}^L \left(- \sum_{j=1}^{T_\ell} h_j^{(\ell)} a_j^{(\ell)} - \sum_{j=1}^{T_{\ell-1}} \sum_{k=1}^{T_\ell} h_j^{(\ell-1)} h_k^{(\ell)} w_{jk}^{(\ell)} \right) \end{aligned} \quad (30)$$

where $\mathbf{h}^{(\ell)} = [h_1^{(\ell)}, h_2^{(\ell)}, \dots, h_{T_\ell}^{(\ell)}]$ denotes the ℓ -th hidden layer of the DBM and it contains T_ℓ number of binary-valued hidden units. $\mathbf{W} = [w_{ij}]$ represents the weights between nodes in the visible layer and the nodes in the first hidden layer $\mathbf{h}^{(1)}$. b_i is the bias term associated with i -th visible layer node u_i . $\mathbf{W}^{(\ell)} = [w_{jk}^{(\ell)}]$ where $1 \leq \ell \leq L$ is the weight between the j -th node in the hidden layer $\mathbf{h}^{(\ell)}$ and the k -th node in the hidden layer $\mathbf{h}^{(\ell+1)}$. $a_j^{(\ell)}$ are the bias

terms associated with j -th node in the hidden layer $\mathbf{h}^{(\ell)}$. All these model parameters are represented by the vector Θ .

The probability that the model assigns to a visible vector $\mathbf{u}$ is then given by the Boltzmann distribution of the following form:

$$p(\mathbf{u}; \Theta) = \frac{1}{\mathcal{Z}(\Theta)} \sum_{\mathbf{h}} \exp(-\mathcal{E}(\mathbf{u}, \mathbf{h}; \Theta)) \quad (31)$$

Based on the above formulation the conditional distribution of each hidden layer ℓ , where $2 \leq \ell < L$, of the DBM can be expressed as:

$$\begin{aligned} p(h_j^{(\ell)} = 1 \mid \mathbf{h}^{(\ell-1)}, \mathbf{h}^{(\ell+1)}) \\ = \sigma \left(\sum_{k=1}^{T_{\ell-1}} h_k^{(\ell-1)} w_{kj}^{(\ell)} \right. \\ \left. + \sum_{i=1}^{T_{\ell+1}} h_i^{(\ell+1)} w_{ij}^{(\ell+1)} + a_j^{(\ell)} \right) \end{aligned} \quad (32)$$

The conditional distribution over the last hidden layer $\mathbf{h}^{(L)}$ is defined as:

$$p(h_j^{(L)} \mid \mathbf{h}^{(L-1)}) = \sigma \left(\sum_{k=1}^{T_L} w_{kj}^{(L)} h_k^{(L-1)} + a_j^{(L)} \right) \quad (33)$$

Similarly, the conditional distribution of the visible layer $\mathbf{u}$ and first hidden layer $\mathbf{h}^{(1)}$ is given by:

$$p(u_i = 1 \mid \mathbf{h}^{(1)}) = \sigma \left(\sum_{j=1}^{T_1} h_j^{(1)} w_{ij}^{(1)} + b_i \right) \quad (34)$$

$$p(h_j^{(1)} = 1 \mid \mathbf{u}, \mathbf{h}^{(2)}) = \sigma \left(\sum_{k=1}^{T_2} w_{kj}^{(2)} h_k^{(2)} + \sum_{i=1}^K w_{ij}^{(1)} u_i + a_j^{(1)} \right) \quad (35)$$

where $\sigma(\cdot)$ is the logistic sigmoid function defined as $\sigma(y) = 1/(1 + \exp(-y))$.

The previously mentioned maximum-likelihood learning procedure is also applicable to estimate the model parameters of DBM. However, it should be noted that this algorithm is rather slow, especially for deep architectures with multiple layers of hidden units where the upper layers are quite remote from the visible layer. This limitation can be effectively resolved using the greedy layer-wise learning strategy [25] and is briefly reviewed in the following subsection. This layer-wise training strategy is extended by the proposed HDLA to learn the model parameters.

3.3.1 The Layer-Wise Training Strategy for DBM

Parameter learning in DBM is performed using an unsupervised layer-wise training procedure. In this approach, the layers of DBM are grouped pairwise to form a

sequence of RBMs. Then, the RBMs in the stack are trained independently in a bottom-up fashion such that successive RBMs use the samples drawn from the joint distribution of the visible and hidden layers of the previous RBM in the hierarchy as their input data. The entire learning procedure for a DBM with L hidden layers is summarized in Algorithm 2.

In layer-by-layer training procedure, the first RBM in the hierarchy is trained to model the given input observation. That is, the visible layer $\mathbf{u}$ of the first RBM accepts the input observations and models it using the $\parallel$ -step contrastive divergence algorithm. After training the first RBM, a sufficiently large number of samples are generated from the joint distribution $p(\mathbf{u} \mid \mathbf{h})$ as the input data for the next RBM in hierarchy (step 3 of Algorithm 2).

While training the remaining portion of the DBM, only two layers $\mathbf{h}^{(\ell-1)}$ and $\mathbf{h}^{(\ell)}$ of the network are considered at a time with the assumption that $\mathbf{h}^{(\ell-1)}$ is known and fixed. Then, the joint distribution $p(\mathbf{h}^{(\ell-1)}, \mathbf{h}^{(\ell)})$ of these two layers is approximated as if they constitute an isolated Restricted Boltzmann Machine and its parameters are learned by maximizing the likelihood $p(\mathbf{h}^{(\ell-1)})$. The $\parallel$ -step contrastive divergence learning procedure mentioned in Algorithm 1 is used for this purpose.

Since all the edges are undirected, each hidden layer nodes except those in the last hidden layer of the DBM accept signals from the upper and the lower layer nodes as indicated in Eq. (32). Hence, the training algorithm must account for the top-down and the bottom-up interaction terms while learning the parameters of DBM. With this objective, Salakhutdinov and Hinton [25] modified the structure of the RBMs in the entire stack before the actual training begins. For instance, the following changes have been made to the structure of RBMs while training a DBM with three hidden layers as shown in Fig. 5b. Initially, the first layer RBM is altered to have two copies of visible layer nodes along with tied weights. The newly added visible layer nodes compensate for the lack of top-down interaction terms from the second layer. Similarly, the structure of the third layer RBM is modified in such a way that it involves two copies of hidden layer units $\mathbf{h}^{(3)}$ and the respective weight matrix $\mathbf{W}^{(3)}$ to compensate for the lack of bottom-up interactions from RBM-2. For the intermediate layer, the RBM is restructured such that only the connection strengths $\mathbf{W}^{(2)}$ are doubled. Salakhutdinov and Hinton [25] were able to show that the layer-wise training of DBM with this type of structural modification is guaranteed to yield optimal values for the model parameters.

Fig. 5 Greedy learning strategy for DBM with three hidden layers [25]

Algorithm 2 Layer-by-layer training procedure for DBM with L hidden layers

Input: DBM $(\mathbf{u}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \dots, \mathbf{h}^{(L)})$, training set $\mathcal{D} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N\}$ of input observation.

Output: Model parameters of DBM.

- 1: **for all** $\mathbf{u}_s \in \mathcal{D}$ **do**
 - 2: Replicate the visible layer units and the connection strengths $\mathbf{W}^{(1)}$ of the first level RBM. Then, optimize the parameters of RBM-1 using 1-step contrastive divergence algorithm.
 - 3: Sample new instances $\{\hat{h}^{(i)}\}_{i=1}^n$ from $p(\mathbf{h}^{(1)}, \mathbf{u}_s; 2\mathbf{W}^{(1)})$ and use them as the training data for RBM-2.
 - 4: Optimize the parameters of the second level RBM using 1-step contrastive divergence algorithm.
 - 5: Sample new instances $\{\hat{h}^{(i)}\}_{i=1}^n$ from $p(\mathbf{h}^{(2)}, \mathbf{u}_s; 2\mathbf{W}^{(1)}, 2\mathbf{W}^{(2)})$ and use them as inputs to train RBM-3.
 - 6: Optimize the parameters of RBM-3 using 1-step contrastive divergence algorithm.
 - 7: Proceed recursively up to RBM- $(L - 1)$
 - 8: Replicate the nodes in the hidden layer $\mathbf{h}^{(L)}$ and the connection weights $\mathbf{W}^{(L)}$. Then, learn the parameters of RBM- L using 1-step contrastive divergence algorithm.
 - 9: **end for**
 - 10: Return $\{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}, \dots, \mathbf{W}^{(L)}\}$.
-

4 The Proposed Image Retrieval Framework

The proposed HDLA model for latent topic-based image retrieval mainly involves two processing steps. The first step is fitting the HDLA model to the entire training images. In this step, the parameters of the HDLA model are learned from the training images, and it proceeds in three stages: (i) visual dictionary learning (ii) generating Bag of Visual Word (BoVW) representation of the training images and (iii) layer-by-layer training of the HDLA model in an unsupervised fashion. The second processing step is testing the learned HDLA model and thereby inferring latent topic-based representation of previously unseen images for the task of CBIR.

To obtain the visual dictionary, each image in the training collection is decomposed into non-overlapping, fixed size local image blocks. Then, scattering transform coefficients [26] are extracted from all these local image patches to form the feature space. Finally, the local image feature space is quantized into a predefined number (K) of clusters using the K -means algorithm. Each of the resulting cluster center is termed as a visual word and the set of all visual words thus obtained are termed as a visual dictionary.

The BoVW representation of the images in the training collection is generated by decomposing each of the images into local patches and are then represented by means of scattering transform coefficients. The local image descriptors thus obtained are then mapped to the nearest visual word in the initially constructed visual dictionary. Finally, the number of occurrence of each visual word over the entire image is computed to form a K -dimensional feature vector popularly known as BoVW representation.

The HDLA model has a layered hierarchical structure where the processing elements are called nodes. There is one layer of visible nodes and multiple layers of hidden nodes stacked on top of one another to constitute the HDLA model. The nodes of any two adjacent layers are bidirectionally connected through weights, and it serves as the model parameters. Each layer of the HDLA model generates activation probability conditioned on the corresponding inputs, and it mainly depends on the model weights.

As the visible layer accepts the visual word count in the form of BoVW representation of training images, the lowest level in the HDLA model is an RSM with additional constraints on its weights and activation probabilities. The upper hidden layers of the HDLA model are paired together to form a hierarchy of Restricted Boltzmann Machines. The hidden layer nodes in HDLA capture the higher-order correlation among visual words and thereby group semantically identical visual words together to form latent topics. The output of the topmost hidden layer will be the latent topic distribution of the given image and is employed for the task of image retrieval.

We use a greedy layer-wise training strategy to learn the parameters of the proposed HDLA model, and it leads to iterative update rules for the parameters of individual layers. The basic idea of the layer-wise training strategy is to train the HDLA model one layer at a time, starting from the first layer. The principle of maximum likelihood is employed to learn the parameters of individual layer in the HDLA model. Thus, for a given collection of training images, the parameters of individual layers are learned in such a way that gives the highest possible probability to the given training data.

Given a previously unseen image (I_{test}) in the testing phase of the proposed HDLA model, its BoVW representation ($\mathbf{v}_{\text{test}}$) is obtained based on the initially created visual dictionary and this BoVW representation is then presented as input to the visible layer of the HDLA model. The latent topic distribution of the test image is then computed as the activation probability $p(\mathbf{h}^L | \mathbf{v}_{\text{test}})$ of the topmost hidden layer in the HDLA model conditioned on the BoVW representation of the given test image. A ranked list of database images is then prepared on the basis of this latent topic features. Figure 6 shows graphically the process for both training and testing the proposed HDLA for the task of image retrieval. The rest of this section provides the implementation details of the proposed HDLA model.

4.1 The Hybrid Deep Learning Architecture

As mentioned earlier, latent topic representation obtained with Deep Boltzmann Machine-based architecture possesses good generalization ability. Deep Boltzmann Machine has multiple layers of processing modules stacked on top of one another, and each unsupervised module in this hierarchy is provided with the representation vectors from the lower level module. Thus, the latent topic vector in the upper-layer capture the high-level dependencies among input variables and thereby improve the ability of the system to learn complex distributions present in the input data.

However, the fully distributed representation yielded by DBM often fails to capture the constituent parts or factors of the input observations. In other words, the high-level abstraction generated by DBM often lacks the inherent meaning of adding parts to form a whole. In fact, “part-based” representation [27] ensures non-subtractive combinations of components to form the given input. Therefore, by restricting the network weights of DBM to nonnegative values yield a “part-based” representation of input data and it possibly enhances the expressive power of the basic DBM model. Another possibility for improving the performance of DBM is the incorporation of sparsity into the learned representation. In sparse feature coding [28], the final representation is forced to have only a few non-zero components, and most of the remaining entries

Fig. 6 The proposed HDLA model-based image retrieval framework

are zero. Hence, sparsity is an effective constraint for performance enhancement where there is no intimation about the required number of hidden layers in DBM and the amount of hidden units required in successive layers while creating an optimal deep network that efficiently discovers interesting structures embedded in the input data.

Considering the above-mentioned factors, this paper proposes a Hybrid Deep Learning Architecture (HDLA) which uses a Constrained Replicated Softmax Model (CRSM) in the lowest level together with Constrained Restricted Boltzmann Machines (CRBMs) in the higher layers. The proposed architecture integrates a quadratic barrier function [29] into the objective function of both CRSM and CRBM so that learning is skewed toward nonnegative weights. With this formulation, the contribution of lower layer units toward each unit in the next higher layer becomes additive in nature. In addition to this, ℓ_1 -regularization term is also added to the objective functions of RSM and RBM to enforce sparseness of the final representation. The basic architecture of the proposed model is shown in Fig. 7.

The following subsections provide a detailed description of the Constrained Replicated Softmax Model (CRSM) and the Constrained Restricted Boltzmann Machine (CRBM)

which add up to form the proposed HDLA model to infer latent topic-based image representation applicable for the retrieval operation.

4.1.1 Constrained Replicated Softmax Model

This section presents a modified version of the Replicated Softmax Model (RSM) named CRSM which serves as the base-level processing module in the proposed HDLA model. Let $\mathbb{U} = (\mathbb{u}_1, \mathbb{u}_2, \dots, \mathbb{u}_K) \in \{1, 2, \dots, P\}^K$ denote the set of visible variables and $\mathbf{h} = (h_1^{(1)}, h_2^{(1)}, \dots, h_{T_1}^{(1)}) \in \{0, 1\}^{T_1}$ indicate the set of hidden nodes of CRSM. The input to the visible units of CRSM is the visual word-count vectors and to learn an optimum fitting distribution for any given set of m data samples $\{\mathbb{U}_1, \mathbb{U}_2, \dots, \mathbb{U}_m\}$ CRSM attempt to solve the following minimization problem.

$$J_1(\theta_1) = \min_{\theta_1} - \sum_{s=1}^m \ln [p(\mathbb{U}_s; \theta_1)] + \beta_1 \sum_{i=1}^K \sum_{j=1}^{T_1} f(\mathbb{u}_{ij}) + \gamma_1 \sum_{s=1}^m f(p(\mathbf{h}^{(1)} | \mathbb{U}_s)) \quad (36)$$

Fig. 7 Graphical representation of the proposed hybrid deep learning architecture

where $\Theta_1 = [\mathbb{W}, \mathbf{a}, \mathbf{b}]$ is the model parameter vector in which $\mathbf{a} = [a_1, a_2, \dots, a_{T_1}]$ and $\mathbf{b} = [b_1, b_2, \dots, b_K]$ represent the bias of hidden layer $\mathbf{h}^{(1)}$ and visible layer $\mathbb{U}$, respectively, $\mathbb{W} = [\mathbb{w}_{ij}]$ denote the weight between the i -th visible layer node and the j -th hidden layer unit. $\ln [p(\mathbb{U}_s; \theta_1)]$ is the log-likelihood of the training sample $\mathbb{U}_s$ and is computed by taking the logarithm of the probability value defined in Eq. (25). $f(\mathbb{w}_{ij})$ is the quadratic barrier function which enforces nonnegativity restriction on the model weights, $f(p(\mathbf{h}^{(1)} | \mathbb{U}_s))$ is the ℓ_1 -regularization term which is used to enforce sparsity on the latent topic representation learned by CRSM. β_1, γ_1 are the weight penalty term and the sparse hyper-parameter of CRSM. They, respectively, control the level of nonnegativity of connection weight matrix $\mathbb{W}$ and the sparsity of hidden layer activation $p(\mathbf{h}^{(1)} | \mathbb{U}_s)$.

The quadratic barrier function is defined as follows:

$$f(\mathbb{w}_{ij}) = \begin{cases} \mathbb{w}_{ij}^2, & \mathbb{w}_{ij} < 0 \\ 0, & \mathbb{w}_{ij} \geq 0 \end{cases} \quad (37)$$

The sparse regularization term which makes the hidden activation of CRSM to be sparse is written as:

$$f(p(\mathbf{h}^{(1)} | \mathbb{U}_s)) = \sum_{k=1}^{T_1} |f(p(h_k^{(1)} = 1 | \mathbb{U}_s))| \quad (38)$$

Thus, the objective function is the sum of a log-likelihood term and two regularization terms. To estimate the model parameters, the stochastic gradient descent procedure is used. Then, the derivative of Eq. (36) with respect to the model parameter Θ_1 for a given sample $\mathbb{U}_s$ consists of three terms as shown below:

$$\begin{aligned} \frac{\partial}{\partial \Theta_1} J_1(\mathbb{U}_s; \Theta_1) = & - \frac{\partial}{\partial \Theta_1} [\ln p(\mathbb{U}_s; \Theta_1)] \\ & + \alpha \frac{\partial}{\partial \Theta_1} \left[\sum_{i=1}^K \sum_{j=1}^{T_1} f(\mathbb{w}_{ij}) \right] \\ & + \beta \frac{\partial}{\partial \Theta_1} [f(p(\mathbf{h}^{(1)} | \mathbb{U}_s))] \end{aligned} \quad (39)$$

In fact, the contrastive divergence learning procedure provides an efficient approximation to the gradient of the log-likelihood term present in Eq. (39). Hence on every iteration, the contrastive divergence algorithm is applied followed by one step of gradient descent using the derivative of the two regularization terms. Thus, for an input observation $\mathbb{U}_s$ of the training set (i.e., $\mathbb{U}^0 = \mathbb{U}_s$) the parameters of CRSM are updated as follows:

$$\begin{aligned} \mathbb{w}_{ij} = & \mathbb{w}_{ij} + \eta \left(\left(p(h_j^{(1)} = 1 | \mathbb{U}^0) \mathbb{w}_{ij}^0 - p(h_j^{(1)} = 1 | \mathbb{U}^k) \mathbb{w}_{ij}^k \right) \right. \\ & \left. + \beta_1 \lceil \lceil \mathbb{w}_{ij} \rceil \rceil^- + \gamma_1 \Delta \mathbb{w}_{ij} \right) \end{aligned} \quad (40)$$

$$a_j = a_j + \eta \left(\left(p(h_j^{(1)} = 1 | \mathbb{U}^0) - p(h_j^{(1)} = 1 | \mathbb{U}^k) \right) + \gamma_1 \Delta a_j \right) \quad (41)$$

$$b_i = b_i + \eta \left(\mathbb{w}_i^0 - \mathbb{w}_i^k \right) \quad (42)$$

where the complete description of the terms $\lceil \lceil \mathbb{w}_{ij} \rceil \rceil^-$, $\Delta \mathbb{w}_{ij}$ and Δa_j are provided in “Appendix A”.

4.1.2 Constrained Restricted Boltzmann Machine

The higher-level processing modules of the proposed HDLA formulation are termed as Constrained Restricted Boltzmann Machines (CRBMs). There are L CRBM modules in the proposed HDLA model. This section explains the formulation of the ℓ -th CRBM (i.e., CRBM- ℓ) where $1 \leq \ell \leq L$ and the basic theory remains the same for all other CRBMs in the hierarchy. More formally, CRBM- ℓ involve two sets of binary stochastic hidden layers $\mathbf{h}^{(\ell)} = (h_1^{(\ell)}, h_2^{(\ell)}, \dots, h_{T_\ell}^{(\ell)})$ and $\mathbf{h}^{(\ell+1)} = (h_1^{(\ell+1)}, h_2^{(\ell+1)}, \dots, h_{T_{\ell+1}}^{(\ell+1)})$. Then, CRBM- ℓ can model any distribution on $\{0, 1\}^{T_\ell}$ by

learning appropriate model parameter values that minimizes the following optimization problem for a given set of m training samples $\{\mathbf{h}_1^{(\ell)}, \mathbf{h}_2^{(\ell)}, \dots, \mathbf{h}_m^{(\ell)}\}$

$$J(\Theta_\ell) = \min_{\Theta_\ell} - \sum_{s=1}^m \ln [p(\mathbf{h}_s^{(\ell)}; \Theta_\ell)] + \beta_\ell \sum_{i=1}^{T_\ell} \sum_{j=1}^{T_{\ell+1}} f(w_{ij}^{(\ell)}) + \gamma_\ell \sum_{s=1}^m f(p(\mathbf{h}^{(\ell+1)} | \mathbf{h}_s^{(\ell)})) \quad (43)$$

where $\Theta_\ell = [\mathbf{W}^{(\ell)}, \mathbf{a}^{(\ell)}]$ indicates the parameters of CRBM- ℓ among which $\mathbf{W}^{(\ell)} = [w_{ij}^{(\ell)}]$ represent the interaction between i -th unit in the hidden layer $\mathbf{h}^{(\ell)}$ and j -th unit in the hidden layer $\mathbf{h}^{(\ell+1)}$, $\mathbf{a}^{(\ell)}$ is the bias associated with hidden layer units in $\mathbf{h}^{(\ell+1)}$. $\ln [p(\mathbf{h}_s^{(\ell)}; \Theta_\ell)]$ is the log-likelihood of the given sample $\mathbf{h}_s^{(\ell)}$ and is expressed as the logarithm of the probability value defined in Eq. (25). $f(w_{ij}^{(\ell)})$ is the quadratic barrier function to ensure nonnegativity restriction on the network weights of CRBM- ℓ . $f(p(\mathbf{h}^{(\ell+1)} | \mathbf{h}_s^{(\ell)}))$ is the ℓ_1 -regularization term for the sparse activation of the output hidden layer units of CRBM- ℓ . β_ℓ, γ_ℓ are the weight penalty term and the sparse hyper-parameter of CRBM- ℓ . These parameters are defined in the same way as it was done before in the case of CRSM.

The stochastic gradient descent procedure is then applied to estimate the parameters of CRBM- ℓ . The derivative of Eq. (43) with respect to the model parameters Θ_ℓ for a given input sample $\mathbf{h}_s^{(\ell)}$ is given by:

$$\begin{aligned} \frac{\partial}{\partial \Theta_\ell} J_\ell(\mathbf{h}_s^{(\ell)}; \Theta_\ell) &= -\frac{\partial}{\partial \Theta_\ell} [\ln p(\mathbf{h}_s^{(\ell)}; \Theta_\ell)] \\ &+ \alpha_\ell \frac{\partial}{\partial \Theta_\ell} \left[\sum_{i=1}^{T_\ell} \sum_{j=1}^{T_{\ell+1}} f(w_{ij}^{(\ell)}) \right] \\ &+ \beta_\ell \frac{\partial}{\partial \Theta_\ell} [f(p(\mathbf{h}^{(\ell+1)} | \mathbf{h}_s^{(\ell)}))] \end{aligned} \quad (44)$$

Similar to CRSM, the parameter estimation of the CRBM- ℓ is obtained by applying the contrastive divergence learning rule followed by a gradient descent step based on the derivative of the sparse regularization term and nonnegativity constraint (refer “Appendix B” for more details).

Then, for an input sample $\mathbf{h}_s^{(\ell)}$ from the training set (i.e., $\mathbf{h}^0 = \mathbf{h}_s^{(\ell)}$) the parameter update rules of CRBM- ℓ becomes:

$$\begin{aligned} w_{ij}^{(\ell)} &= w_{ij}^{(\ell)} + \eta \left((p(h_j^{(\ell+1)} = 1 | \mathbf{h}^0) h_i^0 - p(h_j^{(\ell+1)} = 1 | \mathbf{h}^k) h_i^k) \right. \\ &\quad \left. + \beta_\ell [\|w_{ij}^{(\ell)}\|^- + \gamma_\ell \nabla w_{ij}^{(\ell)}] \right) \end{aligned} \quad (45)$$

$$a_j^{(\ell)} = a_j^{(\ell)} + \eta \left((p(h_j^{(\ell+1)} = 1 | \mathbf{h}^0) - p(h_j^{(\ell+1)} = 1 | \mathbf{h}^k)) + \gamma_\ell \nabla a_j^{(\ell)} \right) \quad (46)$$

where the complete description of the terms $\|w_{ij}^{(\ell)}\|^-$, $\nabla w_{ij}^{(\ell)}$ and $\nabla a_j^{(\ell)}$ are provided in “Appendix B”.

Algorithm 3 Layer-by-layer training procedure for the proposed HDLA model

Input: HDLA $(\mathbb{U}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \dots, \mathbf{h}^{(L)})$, training set $\mathcal{D} = \{\mathbb{U}_1, \mathbb{U}_2, \dots, \mathbb{U}_N\}$ of input observation.

Output: Model parameters of DBM.

- 1: **for all** $\mathbf{v}_{test}^{(1)} \in \mathcal{D}$ **do**
 - 2: Make two copies of visible layer nodes of the Constrained Replicated Softmax (CRSM) model.
 - 3: Fit the parameters of CRSM by means of one step contrastive divergence algorithm with the update rules as specified in Equations (40) - (42).
 - 4: Freeze $W^{(1)}$ that defines the output of CRSM and obtain samples $\{h^{(i)}\}_{i=1}^n$ from $p(\mathbf{h}^{(1)}, \mathbf{v}_{test}^{(i)}; 2\mathbf{W}^{(1)})$ as the training data for CRBM-1.
 - 5: Fit the parameters of CRBM-1 by running one step contrastive divergence algorithm using the update rules specified in Equations (45) and (46).
 - 6: Freeze $W^{(2)}$ that defines the output of CRBM-1 and obtain samples $\{h^{(i)}\}_{i=1}^n$ from $p(\mathbf{h}^{(2)}, \mathbf{v}_{test}^{(i)}; 2\mathbf{W}^{(1)}, 2\mathbf{W}^{(2)})$ as the training data for RBM-2.
 - 7: Recursively proceed up to CRBM- $(L-1)$
 - 8: Fit the parameters of CRBM- L by replicating the processing units in the hidden layer $\mathbf{h}^{(L)}$ and the bidirectional connection weights $\mathbf{W}^{(L)}$.
 - 9: **end for**
 - 10: Return $\{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}, \dots, \mathbf{W}^{(L)}\}$.
-

4.1.3 HDLA Model Training

The layer-wise learning procedure already mentioned in Algorithm 2 is extended to learn the parameters of the proposed HDLA model. By using the layer-wise strategy, the learning process of the proposed HDLA model is broken down into a number of related sub-tasks such that all of them can be completed in a stage-by-stage fashion. The basic idea here is to gradually present input observations to the HDLA model so that at the early stages of training the coarse-scale properties of input observations are captured while the fine-scale characteristics are learned in later stages. After training each layer, its output is considered as the input for training the next layer. That is, the output of each layer serves as a prior for learning the parameters of the next higher layer. The entire procedure for training the proposed HDLA model is summarized in Algorithm 3.

Initially, the parameters of CRSM module which takes the BoVW representation of each training image as input are optimized using one-step contrastive divergence algorithm with the update rules specified in Eqs. (40)–(42). Then, we freeze the obtained parameters of CRSM and its hidden layer configuration for the given input observations is inferred. These inferred values then act as the input data for CRBM-1 in the next higher level of the hybrid deep learning architecture. Again, the one-step contrastive divergence algorithm with the value $\ell = 1$ and the modified update rules specified in Eqs. (45) and (46) are used to derive the parameters of CRBM-1. This procedure is repeated until the parameters of CRBM- L in the hierarchy are learned. To account for the top-down and bottom-up interaction terms, the structure of the HDLA model is altered while training according to the strategy already illustrated in Sect. 3.3.1. Finally, these parameters are composed together to form the required HDLA model.

5 HDLA-Based Image Representation

This section describes how to learn a latent topic-based representation suitable for image retrieval from the trained HDLA model. Furthermore, the distance metric used to estimate the semantic similarity between images is also discussed.

5.1 Encoding of Previously Unseen Images

Once the model parameters of HDLA are learned from an appropriate set of training samples, the given query and the database images can be mapped into the latent topic space for the purpose of image retrieval. The conceived HDLA model with L hidden layers generates a latent topic-based representation $p(\mathbf{h}^L | \mathbf{v}_{\text{test}})$ for every input image whose

BoVW representation is $\mathbf{v}_{\text{test}}$. The activation $p(\mathbf{h}^L | \mathbf{v}_{\text{test}})$ of the topmost hidden layer of HDLA denotes the latent topic structure of the given image and is taken as the feature vector for the desired retrieval operation.

5.2 Image Similarity Measure

To use the features generated by the proposed hybrid deep learning architecture for image retrieval, an appropriate similarity measure has to be defined which efficiently estimates the correspondence between images characterized by their latent topic distribution. In recent years, deep learning-based models for document classification and retrieval use Jensen–Shannon (JS) divergence as the similarity metric, and found to yield good performance in terms of classification and retrieval accuracy [21]. This motivates the use of JS divergence as the similarity metric in the proposed work. Given the query $\mathcal{J}_q$ and the database image $\mathcal{J}_d$, let the K -dimensional latent topic-based representation obtained with the proposed HDLA model is denoted by $\mathbb{f}_q$ and $\mathbb{f}_d$. Then, the Jensen–Shannon divergence similarity measure $JS(\mathbb{f}_q, \mathbb{f}_d)$ for estimating the similarity between two latent topic-based distributions $\mathbb{f}_q$ and $\mathbb{f}_d$ and is formally defined as follows:

$$JS(\mathbb{f}_q, \mathbb{f}_d) = \frac{1}{2} \left(\text{KL} \left(\mathbb{f}_q, \frac{\mathbb{f}_q + \mathbb{f}_d}{2} \right) + \text{KL} \left(\mathbb{f}_d, \frac{\mathbb{f}_q + \mathbb{f}_d}{2} \right) \right) \quad (47)$$

where $\text{KL}(\mathbb{f}_q, \mathbb{f}_d)$ is expressed as:

$$\text{KL}(\mathbb{f}_q, \mathbb{f}_d) = \sum_{i=1}^K f_q^i \log \left(\frac{f_q^i}{f_d^i} \right) \quad (48)$$

where f_q^i and f_d^i , respectively, denote the i -th bin of the feature vectors $\mathbb{f}_q$ and $\mathbb{f}_d$.

6 Performance Evaluation and Discussion

The experimental validation of the formulated model is demonstrated in this section. Firstly, a short description of the datasets used for evaluation is provided. Then, the quantitative evaluation of the proposed HDLA model in terms of its generalization ability is carried out. Finally, the retrieval efficiency of the latent topic-based image representation obtained with the proposed HDLA model is compared with state-of-the-art approaches.

6.1 Datasets Used

In the past, a number of benchmark datasets having ground truth images for a set of predefined queries have been introduced for evaluating different CBIR frameworks.

Among them, six image collections with contrasting characteristics are selected to use in our retrieval experiments, and this section provides a detailed description of all these image collections.

INRIA Holiday dataset [30] It involves 1491 high-resolution images of various places situated all over the universe. Images in this collection have a resolution of either 570×760 or 1020×760 and it mainly includes natural scene types. Among them, 500 images are reserved as queries and there exist predefined retrieval lists for each of the queries.

Scene-15 dataset [31] There are mainly 4485 images in this collection and are grouped into 15 concept categories. In total, 210 to 410 images are there in each category and all of them have a fixed resolution equal to 250×300 pixels. Most of the images in the Scene-15 collection have distinguishing background and foreground context. Therefore, this image collection serves as a good choice for evaluating context-aware semantic image modeling schemes for the task of CBIR.

Oxford dataset [32] This benchmark dataset comprises 5062 building images located at 11 various landmarks of the Oxford city, and it is difficult to distinguish similar building facades from one another. All images in the collection have a fixed resolution of 1020×760 . The ground truth includes five images from each of the 11 landmarks and their corresponding search results. That is, 55 queries are there to evaluate the effectiveness of any retrieval system.

GHIM-10K dataset [33] There are 10,000 images in the GHIM-10K dataset which spread over 20 concept categories. Each category contains 500 color images in JPEG format with a resolution of 300×400 or 400×300 . Those images in the search result that belongs to the semantic category similar to the given query are judged as relevant. That is, a randomly selected image from any of these 20 concept classes can act as the query and there are exactly 499 relevant images in the collection.

IAPR TC-12 dataset [34] Another widely used image collection selected for retrieval evaluation is the IAPR TC-12 dataset. It involves 20,000 images collected from various locations around the globe comprising different types of natural scene images. All images in this collection are in JPEG format with a fixed size of 360×480 pixels. An interesting property of this image collection is that there are many images having identical visual content; however, they differ in background, lighting conditions and viewing position.

MIRFLICKR-40K dataset [35] The final image collection selected for evaluation is the MIRFLICKR-40K dataset and is a subset of the MIRFLICKR-1M collection. This dataset comprises 40,000 images and all of them have a fixed resolution of 720×480 . The notable characteristic

of this image collection is that it exhibits semantic diversity by having images belonging to multiple categories and varying appearance. Thus, the MIRFLICKR-40K dataset provides an in-depth analysis of any image retrieval framework due to its moderate size and heterogeneous content.

6.2 Quantitative Evaluation of the HDLA Model

An ideal topic modeling scheme should adequately model the given data samples and at the same time has the potential to yield semantically coherent latent topics. Therefore, it is necessary to analyze these two aspects of the proposed model while judging its competence. To do so, two sets of experiments are carried out using the proposed model. The first one is the generalization test on unseen data samples, and the other one is the evaluation of reconstruction error for a standard handwritten image collection. In all these experiments, the performance of the proposed model is compared with the following baseline approaches such as Over Replicated Softmax Model (ORMS) [21], Replicated Softmax Model (RSM) [6] and Rate Adapting Poisson model (RAP) [5].

6.2.1 Experimental Setup

The hardware platform for simulating the proposed HDLA model is an Intel Core i7-4570 machine equipped with 3.4 GHz CPU and 16 GB of RAM. The HDLA model is coded in MATLAB R2016b(9.1) environment under Unix operating system. For all the experiments presented in this paper, the proposed HDLA model is trained for 200 epochs with a learning rate $\eta = 0.2$. The visible and hidden layer biases are initialized with small random values, and the model weights are randomly chosen from positive values in the range $[0, 1]$. It is found that $\|\cdot\|_1 = 1$ is sufficient for the contrastive divergence algorithm to generate good latent topic-based features.

6.2.2 Generalization Performance on Unseen Samples

Since topic models are trained in a completely unsupervised fashion, it is difficult to evaluate the competence of one model over the other. In practice, the performance of topic models is evaluated using their generalization ability on unseen data sample. More specifically, estimating the likelihood of a held-out data set provides a clear, interpretable metric for evaluating the performance of topic models relative to other existing models.

The log-likelihood and the perplexity scores are the commonly used metrics to quantify the generalization ability of topic models. Let $\mathbf{v}_{\text{test}}$ denote the BoVW-based representation of an input image, then the HDLA model

assign a probability $p(\mathbf{v}_{\text{test}}) = \sum_{\mathbf{h}} P(\mathbf{v}_{\text{test}}, \mathbf{h})$ to the visible vector $\mathbf{v}_{\text{test}}$. However, in practice, it is computationally intractable because of the sum of an exponential number of terms. Therefore, we rely on sampling to compute the log-likelihood values as follows:

$$\log [p(\mathbf{v}_{\text{test}})] = \log \left[\frac{1}{n} \sum_{i=1}^n p(\mathbf{v}_{\text{test}} | \mathbf{h}^{(i)}) \right] \quad (49)$$

where $\{\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \dots, \mathbf{h}^{(n)}\}$ is a set of n samples drawn from $P(\mathbf{v}_{\text{test}}, \mathbf{h})$ by means of Gibbs sampling. Then, the average test perplexity value is computed as:

$$\text{perplexity}(\mathcal{J}_{\text{test}}) = \exp \left(-\frac{1}{|D|} \sum_{i=1}^{|D|} \frac{1}{N_i} \log p(\mathbf{v}_{\text{test}}^{(i)}) \right) \quad (50)$$

where $\mathcal{J}_{\text{test}}$ is the given collection of test images, $|D|$ is the number of images in the collection $\mathcal{J}_{\text{test}}$, N_i and $\mathbf{v}_{\text{test}}^{(i)}$, respectively, denotes the number of interest points and the visual word-count vector for the i -th image in the collection $\mathcal{J}_{\text{test}}$. From this definition, one can see that a low perplexity score always indicates a better generalization performance.

We conducted log-likelihood and perplexity analysis by experimenting on all the six data sets considered for evaluation. HDLA model with three hidden layer units (i.e., $L = 3$) is used in this experiment. The visible layer of the proposed model accepts BoVW-based representation of input images and then maps the input to latent topic space. The log-likelihood and perplexity values are calculated by running the Gibbs sampler three times each with 1000 iterations and then by taking the average of these three scores. Tenfold cross-validation is performed in all the six datasets considered for evaluation. That is, images in the individual dataset are grouped into tenfolds of approximately equal sizes. Special care has been taken to ensure that there is no overlap between images belonging to each fold. Then, in each run of the experiment, ninefolds are used for model training, and the remaining onefold is used for testing the model. For different sizes (K) of the visual dictionary, the total log-likelihood values obtained for each of the compared models by varying the number of latent topics ($T_{L=3}$) are summarized in Table 2. From these results, it can be concluded that the proposed model outperforms other existing models in terms of its generalization performance.

Next, the convergence property of the hybrid deep learning model is analyzed. To this end, a series of experiments have been carried out to see whether the proposed topic modeling scheme converges at a rate faster than state-of-the-art approaches. Figure 8 depicts the perplexity values of individual models as a function of the number of iterations when applied to all the six image data sets. In all these experiments, K and $T_{L=3}$ values are

selected in such a way that gives better generalization performance. The obtained results revealed the fact that the perplexity values of the formulated model consistently decrease in successive iterations and it achieves a faster rate of convergence as compared to other models.

In conclusion, the effectiveness of a given topic modeling scheme entirely depends on its generalization ability and which in turn directly related to the number of training iterations. There is always an upper limit beyond which an increase in the number of iteration has no effect on the model's generalization power. It is evident from the above results that the generalization power of the existing models is not up to the mark even for a substantially large number of training iterations. However, the proposed HDLA model outperforms the widely used baseline models in terms of its generalization ability and convergence rate. That is, HDLA model attains better generalization power within a lesser number of training iterations. Therefore, the HDLA-based formulation is capable of yielding a semantic-based image representation having more discriminative power.

6.2.3 Reconstruction Performance

To further evaluate the effectiveness of the obtained latent topic-based representation, the hybrid deep learning architecture is applied to model images of handwritten digits. The performance of the proposed model is then measured in terms of reconstruction error, which is defined as the average pixel differences between the original and reconstructed images. The Reconstruction Error (RE) for a given image $\mathcal{J}$ is calculated as follows:

$$\text{RE}(\mathcal{J}) = \frac{1}{d} \sum_{j=1}^d (\mathcal{J}_j \neq \tilde{\mathcal{J}}_j) \quad (51)$$

where d the dimensionality of the vectorized version of each input image $\mathcal{J}$ and $\tilde{\mathcal{J}}$ is the reconstructed value of $\mathcal{J}$ by the learned model. The MNIST handwritten digit dataset [36] is used as the benchmark for experimental evaluation. This dataset contains 60,000 training and 10,000 test images for each of the 10 (0 to 9) digits. Each handwritten digit is a 28×28 -pixel gray level image. Hence, the visible layer of the proposed model contains $28 \times 28 = 784$ nodes.

Initially, the pixel values (0-255) of all input images are mapped to 0 or 1. For this, a threshold value of 30 is selected, and pixel values greater than or equal to 30 are set to 1 while values less than 30 are set to 0. A given image in its vectorized binary form is reconstructed by sampling the top most hidden layer vector from the latent model under evaluation followed by sampling the visible vector based on the generated hidden vector. The resulting visible vector is multiplied by 255 and is then binarized by the same

Table 2 Quantitative evaluation of proposed HDLA model based on total log-probability ($\sum \log p(v_{\text{test}})$) scores calculated over the test images of individual data sets

Dictionary size	$T_L=3$	Holiday dataset				Scene-15 dataset				Oxford dataset			
		ORSM		RAP		ORSM		RAP		ORSM		RAP	
		HDLA	RSM	HDLA	RAP	HDLA	RSM	HDLA	RAP	HDLA	RSM	HDLA	RAP
250	50	-69.23	-84.20	-89.25	-94.02	-70.79	-82.25	-82.78	-86.73	-88.32	-100.02	-110.52	-116.35
	75	-59.82	-73.26	-78.84	-93.85	-65.29	-73.38	-77.83	-82.16	-76.71	-87.49	-96.38	-102.58
	100	-46.65	-60.69	-68.32	-74.82	-60.96	-68.49	-73.48	-78.83	-63.56	-74.96	-83.23	-89.38
	125	-41.81	-55.42	-61.09	-66.38	-58.20	-65.31	-69.80	-74.56	-60.12	-71.67	-81.49	-87.63
500	100	-60.26	-74.86	-79.46	-84.68	-60.19	-69.29	-75.98	-81.46	-82.49	-93.16	-101.28	-108.58
	125	-50.75	-65.99	-70.43	-75.18	-57.59	-65.44	-71.84	-76.61	-68.65	-79.26	-87.82	-95.68
	150	-44.73	-57.81	-62.33	-68.49	-52.34	-61.02	-67.18	-72.29	-57.88	-68.49	-76.19	-84.42
	175	-39.11	-53.27	-58.09	-73.14	-49.87	-57.48	-63.49	-69.18	-54.74	-65.20	-72.39	-78.56
750	150	-53.49	-67.18	-73.51	-78.32	-56.12	-64.32	-69.44	-74.14	-70.97	-81.46	-88.13	-94.88
	175	-45.63	-59.47	-64.36	-69.48	-52.67	-60.44	-65.90	-71.41	-62.39	-73.43	-80.38	-86.36
	200	-40.16	-54.32	-59.07	-64.54	-48.29	-56.82	-62.76	-68.79	-54.29	-65.38	-72.19	-78.37
	225	-36.79	-50.71	-55.86	-60.48	-43.50	-52.28	-59.34	-65.43	-51.66	-62.29	-69.93	-74.86
Dictionary Size	$T_L=3$	GHIM-10K dataset				IAPR TC-12 dataset				MIRFlickr-40K dataset			
		ORSM		RAP		ORSM		RAP		ORSM		RAP	
		HDLA	RSM	HDLA	RAP	HDLA	RSM	HDLA	RAP	HDLA	RSM	HDLA	RAP
250	50	-86.11	-92.55	-96.41	-101.23	-71.72	-87.51	-93.38	-98.14	-60.27	-72.93	-77.02	-82.54
	75	-75.28	-81.82	-85.29	-90.28	-61.42	-75.80	-81.16	-86.28	-49.36	-60.15	-65.47	-71.41
	100	-67.33	-73.62	-79.39	-84.08	-48.91	-62.67	-68.33	-73.45	-45.36	-56.23	-60.31	-65.38
	125	-64.53	-70.27	-76.38	-81.22	-43.17	-57.23	-73.49	-79.07	-43.76	-54.13	-58.22	-64.14
500	100	-76.37	-82.48	-87.69	-92.49	-58.45	-73.43	-79.28	-85.66	-55.37	-65.24	-69.06	-73.38
	125	-71.85	-76.31	-80.88	-85.59	-55.34	-70.56	-76.22	-82.72	-44.39	-55.18	-59.73	-64.18
	150	-65.77	-70.11	-74.29	-79.71	-44.32	-59.85	-65.20	-70.15	-40.83	-51.49	-55.12	-60.66
	175	-61.28	-67.48	-71.29	-76.38	-40.39	-55.77	-62.54	-68.16	-37.34	-48.64	-52.23	-56.11
750	150	-69.86	-75.36	-80.34	-85.43	-45.74	-60.42	-67.53	-72.34	-45.91	-56.14	-61.22	-66.18
	175	-65.39	-71.82	-76.29	-81.81	-47.58	-62.17	-68.34	-74.04	-40.92	-51.87	-56.23	-60.63
	200	-62.78	-68.53	-74.62	-79.33	-41.76	-56.26	-63.52	-68.85	-35.11	-46.88	-50.03	-55.18
	225	-57.41	-63.34	-69.93	-74.19	-37.30	-52.65	-59.47	-65.13	-33.11	-44.33	-48.12	-53.74

The table summarizes the total log-probability values of tenfold cross-validation. The log-probability values of the proposed HDLA model are shown in boldface letters

Fig. 8 Test perplexity values versus iteration count of the proposed HDLA model in comparison with state-of-the-art latent topic modeling approaches. **a** Holiday dataset. **b** Scene-15 dataset. **c** Oxford dataset. **d** GHIM-10K dataset. **e** IAPR TC-12 dataset. **f** MIRFLICKR-40K dataset

Table 3 Evaluation of the reconstruction performance of the proposed HDLA model

Number of RBM units	No of training samples	Model configuration	Reconstruction error (%)	
			ORSM	HDLA
3	30,000	(784-500-150)	21.38	14.16
		(784-300-100)	19.66	11.48
		(784-200-50)	17.43	10.61
	60,000	(784-500-150)	18.49	11.82
		(784-300-100)	15.57	8.94
		(784-200-50)	14.22	7.42
4	30,000	(784-550-350-200)	20.86	13.92
		(784-450-250-150)	17.79	10.29
		(784-350-150-75)	16.84	9.26
	60,000	(784-550-350-200)	17.53	10.56
		(784-450-250-150)	14.19	7.71
		(784-350-150-75)	13.63	6.18

procedure described above. To deal with binary inputs, the RSM unit in the first layer of the proposed HDLA model shown in Fig. 7 is replaced with an RBM unit.

In our experiments, different configurations of the proposed HDLA model are trained for the purpose of reconstructing MNIST handwritten digit images. The performance of the proposed HDLA model is then evaluated in comparison with Over Replicated Softmax Model (ORSM). Instead of directly using the actual training and test sets, the entire data set is pooled into ten equal-sized subsets. One of this subset is then used for model evaluation, and the remaining nine subsets are used for model training. This process is repeated ten times rotating through all the subsets which lead to tenfold cross-validation results. The obtained values are summarized in Table 3. From these results, it is evident that HDLA is a good generative model and it can significantly minimize the reconstruction error as compared to the ORSM-based formulation. Another factor to take into account is the impact of the number of training samples on the performance of HDLA and ORSM. Therefore, experiments are conducted by varying the number of training samples for each configuration of HDLA and ORSM. In all such cases, it seems that the proposed HDLA framework exhibits better reconstruction performance and is less sensitive to the size of training set as compared to ORSM.

6.3 Evaluation of HDLA-Based Image Search

This section evaluates the retrieval effectiveness of the proposed HDLA model in comparison with other latent topic-based approaches. The following subsections delineate the performance measures employed to judge the retrieval results, the procedure used to select appropriate values for the model parameters of HDLA in connection

with effective image retrieval and the search results of the retrieval experiments carried out in various datasets.

6.3.1 Evaluation Metrics

The primary objective of a typical CBIR system is to generate a ranked list of top k images from the given dataset in response to a submitted query. The rank of an image is determined by its relevance to the query at hand. To be able to compare various image retrieval models, first a set of performance measures are to be identified. When the ground truth of the data set is available, the system's performance is generally measured in terms of quantitative metrics such as precision and recall. The precision of a retrieval system measures the percentage of relevant images in the ranked retrieval list and the recall denotes the percentage of relevant images retrieved by the system. These two metrics are defined as follows:

$$\text{Precision} = \frac{\text{Number of relevant images retrieved}}{\text{Total number of images retrieved}} \quad (52)$$

$$\text{Recall} = \frac{\text{Number of relevant images retrieved}}{\text{Total number of relevant images in the set}} \quad (53)$$

Precision and recall do not take into account the order in which relevant images appear in the ranked retrieval list. When two retrieval systems have the same precision and recall values, the system that ranks relevant images higher is mostly preferred. In order to solve this issue, measures like Precision at rank position k ($p@k$) and R-precision are introduced. $p@k$ is the value of precision calculated using the first k documents in the retrieval list. Similarly, R -Precision for a given query is defined to be the precision after retrieving R images from the image data base and is expressed as:

$$R - \text{Precision} = \frac{1}{R} \sum_{j=1}^R \text{Rel}(j) \quad (54)$$

where R is the total number of relevant images in the database for the given query and $\text{Rel}(j)$ is an indicator function which returns the value 1 when the image present at the j -th location of the retrieval list is relevant with respect to the given query.

Moreover, precision can be expressed as a function of recall. The interpolated precision recall graph plots precision as a function of recall and can be used to assess the overall performance of the retrieval framework. The interpolated precision p_{int} at a recall level r_i is calculated as the largest observed precision for any recall value r between r_i and r_{i+1} :

$$P_{\text{int}}(r_i) = \max_{r_i \leq r \leq r_{i+1}} \text{Precision}(r) \quad (55)$$

An alternative single valued evaluation metric is the mean average precision (mAP). For a set of m query images the Mean Average Precision is defined as:

$$\text{Mean Average Precision (mAP)} = \frac{1}{m} \sum_{i=1}^m \text{AP}(q) \quad (56)$$

where $\text{AP}(q)$ is the average precision for a given query q and is defined as the ratio of the sum of precision values from rank positions where a relevant image is found in the retrieval result to the total number of relevant images in the database.

One last metric is the Average Retrieval Rate (ARR) which is defined as:

$$\text{Average Retrieval Rate(ARR)} = \frac{1}{N_Q} \sum_{q=1}^{N_Q} \text{RR}(q) \quad (57)$$

where N_Q represents the number of queries used for evaluating the retrieval system. $\text{RR}(q)$ is the retrieval rate for a single query q and is calculated as:

$$\text{RR}(q) = \frac{N_R(\alpha, q)}{N_G(q)} \quad (58)$$

where $N_G(q)$ is the number of ground truth images of a query q and $N_R(\alpha, q)$ indicates the number of relevant images found in the first $\alpha \times N_G(q)$ images. The value of α should be greater than or equal to 1. Selecting larger α values would be less discriminative between very good retrieval results and those retrieval results that are not so good ones. Hence, in this work the value of α is fixed as 1.5.

Another important metric to evaluate the quality of a retrieval result is the normalized Discounted Cumulative Gain (nDCG). The intuition underlying nDCG is that an end user mainly interested in the top positions of the

retrieval list and less likely to explore the lower-ranked images. To incorporate this notion in the evaluation metric, nDCG follows a graded correctness score. The correctness cr_i of an image i in the retrieval list varies within the range 0–3 according to user judgement, where 0 corresponds to irrelevant images and 3 corresponds to the most relevant image. Based on the correctness score, the usefulness or gain of each image with respect to its position p in the retrieval list is estimated and is then accumulated to compute the nDCG value as follows:

$$\text{nDCG}_p = \frac{\text{DCG}_p}{\text{IDCG}_p} \quad (59)$$

where DCG_p is the discounted cumulative gain and IDCG_p is the ideal DCG value at rank list position p and are, respectively, defined as follows:

$$\text{DCG}_p = \sum_{i=1}^p \frac{2^{cr_i} - 1}{\log_2(i + 1)} \quad (60)$$

$$\text{IDCG}_p = \sum_{i=1}^{|R_N|} \frac{2^{cr_i} - 1}{\log_2(i + 1)} \quad (61)$$

where $|R_N|$ is the number of images in the retrieval list R_q sorted in descending order of their correctness score up to rank position p . The logarithmic factor in the denominator is a penalty term by which a discount is made to the correctness value of highly relevant images appearing at the bottom position of the search result. Finally, the nDCG values of all the queries are averaged to get the overall performance of the retrieval system.

6.3.2 Parameter Selection

In the context of image retrieval, it is important to select appropriate values for the parameters of HDLA model. More specifically, the parameters such as visual dictionary size (K), the number of hidden layers and the number of nodes in each hidden layers need to be tuned for good retrieval performance. For individual image collection, this is done by calculating the average retrieval rates for each query set by varying the visual dictionary size and the number of nodes in each hidden layers of HDLA. Figure 9 depicts the average retrieval rates obtained by different image collections while changing the number of hidden layer units along with visual dictionary size. It is now easy to fix reasonable values for the model parameters by analyzing the results shown in Fig. 9. Once the proper estimates of these parameters have been obtained, they can be frozen and used for subsequent retrieval experiments. To avoid computational bottlenecks, HDLA model with three layers of hidden units are considered in our retrieval experiments. It is empirically found that HDLA model with

Fig. 9 Average retrieval rate obtained by the proposed HDLA model for various combinations of visual dictionary sizes (K) and hidden layer configurations. **a** Holiday dataset. **b** Scene-15 dataset. **c** Oxford dataset. **d** GHIM-10K dataset. **e** IAPR TC-12 dataset. **f** MIRFLICKR-40K dataset

Table 4 Comparative evaluation of the proposed HDLA model based on mean average precision (mAP), average R-precision and normalized discounted cumulative gain ($nDCG$) calculated at rank position $p=10$

Dataset used	mAP					
	HDLA (proposed)	ORSM	RSM	RAP	PAM	LDA
Holiday dataset	0.758	0.695	0.663	0.631	0.613	0.597
Scene-15 dataset	0.725	0.668	0.635	0.607	0.584	0.564
Oxford dataset	0.700	0.647	0.613	0.586	0.561	0.543
GHIM-10K dataset	0.703	0.640	0.611	0.579	0.565	0.544
IAPR TC-12 dataset	0.743	0.681	0.654	0.621	0.606	0.581
MIRFLICKR-40K dataset	0.764	0.702	0.678	0.644	0.626	0.602
Dataset used	Average R-Precision					
	HDLA (proposed)	ORSM	RSM	RAP	PAM	LDA
Holiday dataset	0.768	0.702	0.672	0.644	0.627	0.602
Scene-15 dataset	0.754	0.693	0.661	0.647	0.625	0.600
Oxford dataset	0.725	0.668	0.664	0.632	0.614	0.596
GHIM-10K dataset	0.737	0.679	0.647	0.619	0.595	0.575
IAPR TC-12 dataset	0.759	0.694	0.665	0.660	0.639	0.619
MIRFLICKR-40K dataset	0.786	0.727	0.694	0.665	0.642	0.626
Dataset used	$nDCG_{p=10}$					
	HDLA (proposed)	ORSM	RSM	RAP	PAM	LDA
Holiday dataset	0.837	0.776	0.748	0.716	0.694	0.675
Scene-15 dataset	0.819	0.751	0.729	0.695	0.672	0.653
Oxford dataset	0.774	0.717	0.687	0.653	0.634	0.610
GHIM-10K dataset	0.792	0.734	0.706	0.675	0.653	0.632
IAPR TC-12 dataset	0.825	0.763	0.737	0.702	0.688	0.663
MIRFLICKR-40K dataset	0.853	0.794	0.760	0.733	0.700	0.683

three layers of hidden units is good enough to generate latent topic-based image representation having more discriminative power and retrieval accuracy than the existing topic modeling schemes. The next subsection summarizes the comparative evaluation of various image retrieval experiments.

6.3.3 Retrieval Results and Discussion

This section verifies the retrieval efficiency of the proposed scheme in comparison with state-of-the-art models. In this regard, the following retrieval frameworks have been selected for comparison purpose, namely, Over Replicated Softmax Model (ORSM) [21], Replicated Softmax model (RSM) [6], Rate Adapting Poisson model (RAP) [5], Pachinko Allocation Model (PAM) [15] and Latent Dirichlet Allocation (LDA) [2].

The retrieval effectiveness of the proposed HDLA model is initially evaluated on the basis of mAP , average R-Precision and $nDCG_{p=10}$ values. The comparison of the proposed model and the already existing methods is

provided in Table 4. On average, the HDLA model achieves 6% improvement in the values of mAP , average R-Precision and $nDCG_{p=10}$ as compared to the best performing approach in the literature. From these statistics, it is evident that the proposed HDLA model is promising and it gives better retrieval results compared to state-of-the-art methods.

Figure 10 shows the 11-point interpolated average precision values obtained for the proposed HDLA-based image search in comparison with other latent topic-based retrieval strategies. From these results, it can be concluded that the precision achieved with the proposed HDLA-based image representation is obviously better than the existing models across all values of recall for all image collections selected for evaluation.

To further validate the effectiveness of the proposed HDLA model, its performance is compared with other existing models in terms of the average precision values at selected rank thresholds of 10, 20 and 30 (i.e., $p@10$, $p@20$ and $p@30$). The average precision values of the retrieval experiments carried out in all the benchmark datasets are

Fig. 10 Evaluation of the proposed HDLA-based image retrieval framework in comparison with state-of-the-art approaches based on 11-point interpolated average precision. **a** Holiday dataset. **b** Scene-15

dataset. **c** Oxford dataset. **d** GHIM-10K dataset. **e** IAPR TC-12 dataset. **f** MIRFLICKR-40K dataset

Table 5 Comparative evaluation of the proposed HDLA model based on precision values calculated at cut-off levels 10, 20 and 30

Evaluation metric	Holiday dataset						Scene-15 dataset					
	HDLA	ORSM	RSM	RAP	PAM	LDA	HDLA	ORSM	RSM	RAP	PAM	LDA
p@10	0.868	0.807	0.765	0.742	0.733	0.721	0.833	0.774	0.732	0.710	0.703	0.698
p@20	0.835	0.785	0.733	0.716	0.702	0.696	0.804	0.746	0.701	0.687	0.675	0.664
p@30	0.816	0.757	0.717	0.684	0.687	0.675	0.776	0.712	0.677	0.653	0.642	0.633
Evaluation metric	Oxford dataset						GHIM-10K dataset					
	HDLA	ORSM	RSM	RAP	PAM	LDA	HDLA	ORSM	RSM	RAP	PAM	LDA
p@10	0.800	0.743	0.716	0.693	0.684	0.671	0.813	0.754	0.711	0.697	0.680	0.678
p@20	0.779	0.712	0.687	0.661	0.653	0.646	0.787	0.716	0.689	0.664	0.655	0.647
p@30	0.747	0.682	0.654	0.637	0.625	0.611	0.745	0.674	0.644	0.623	0.616	0.605
Evaluation metric	IAPR TC-12 dataset						MIRFLICKR-40K dataset					
	HDLA	ORSM	RSM	RAP	PAM	LDA	HDLA	ORSM	RSM	RAP	PAM	LDA
p@10	0.853	0.795	0.760	0.748	0.733	0.726	0.874	0.812	0.686	0.642	0.633	0.620
p@20	0.821	0.766	0.735	0.711	0.700	0.692	0.851	0.787	0.651	0.623	0.614	0.601
p@30	0.797	0.738	0.704	0.687	0.675	0.664	0.824	0.764	0.637	0.605	0.598	0.587

The performance figures of the proposed HDLA model are shown in boldface letters

presented in Table 5. When an end user is interested in viewing only the top 10, 20 and 30 results returned by the retrieval model, then 6% improvement on average is achieved with the proposed HDLA-based formulation.

To conclude, the hybrid deep learning architecture proposed in this paper yields compact but discriminative image representation well suited for the retrieval operation. All the retrieval experiments substantiate the ability of the proposed HDLA model in discovering latent topics by grouping semantically similar visual words to characterize images at a much higher-level of abstraction. The above-mentioned experimental results validate the potential of HDLA-based formulation to bridge the semantic gap in image understanding and retrieval.

7 Conclusion

In this paper, a new class of topic modeling scheme called hybrid deep learning architecture is proposed for semantic image modeling and retrieval. The proposed architecture is a composite of Replicated Softmax Model and Restricted Boltzmann Machines with nonnegativity restriction on the network weights and ℓ_1 -sparseness constraint on the hidden layer activations. As part of image modeling, the formulated architecture infers a hierarchical nonlinear mapping function in a completely unsupervised fashion that projects the original BoVW-based representation on to a latent topic-based semantic concept space. Thus, the hybrid deep

learning architecture can capture semantic correlation among visual words and consequently minimizes the semantic loss associated with BoVW-based image retrieval. Based on the experimental evaluations it can be concluded that the image representation yielded by the proposed HDLA model significantly improves the retrieval performance as compared to state-of-the-art latent topic-based image retrieval systems.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no competing interests.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix A Gradient of the Regularization Terms for CRSM

The gradient of the quadratic barrier function with respect to the network weights of CRSM is computed as follows:

$$\frac{\partial}{\partial w_{ij}} f(w_{ij}) = \llbracket w_{ij} \rrbracket^- = \begin{cases} 2w_{ij}, & w_{ij} < 0 \\ 0, & w_{ij} \geq 0 \end{cases} \quad (62)$$

the gradient exists only for negative weights and in all other cases its value is zero. As the definition of the quadratic barrier function is free from the bias terms, its gradient with respect to $\mathbf{b}$ and $\mathbf{a}^{(1)}$ will be zero.

Similarly, the gradient of the sparse regularization term with respect to the model parameters of CRSM is computed as follows:

$$\begin{aligned} \frac{\partial}{\partial w_{ij}} \left(f(p(\mathbf{h}^{(1)} | \mathbb{U}_s)) \right) &= \frac{\partial}{\partial w_{ij}} \left(\frac{1}{1 + \exp\left(-\sum_{i=1}^K w_{ij} \mathbb{U}_i - Na_j\right)} \right) \\ &= \frac{\mathbb{U}_i \times \exp\left(-\sum_{i=1}^K w_{ij} \mathbb{U}_i - Na_j\right)}{\left| 1 + \exp\left(-\sum_{i=1}^K w_{ij} \mathbb{U}_i - Na_j\right) \right|^2} \\ &= \mathbb{U}_i \times \left(p(h_j^{(1)} = 1 | \mathbb{U}_s) - p^2(h_j^{(1)} = 1 | \mathbb{U}_s) \right) \\ &= \Delta w_{ij} \end{aligned} \quad (63)$$

$$\begin{aligned} \frac{\partial}{\partial a_j} \left(f(p(\mathbf{h}^{(1)} | \mathbb{U}_s)) \right) &= \frac{\partial}{\partial a_j} \left(\frac{1}{1 + \exp\left(-\sum_{i=1}^K w_{ij} \mathbb{U}_i - Na_j\right)} \right) \\ &= \frac{N \times \exp\left(-\sum_{i=1}^K w_{ij} \mathbb{U}_i - Na_j\right)}{\left| 1 + \exp\left(-\sum_{i=1}^K w_{ij} \mathbb{U}_i - Na_j\right) \right|^2} \\ &= N \times \left(p(h_j^{(1)} = 1 | \mathbb{U}_s) - p^2(h_j^{(1)} = 1 | \mathbb{U}_s) \right) \\ &= \Delta a_j \end{aligned} \quad (64)$$

Since the activation probability of the hidden units in CRSM is independent of the visible layer bias term $\mathbf{b}$, the gradient of the sparse regularization part with respect to $\mathbf{b}$ will be zero.

Appendix B Gradient of the Regularization Terms for CRBM- ℓ

The gradient of the quadratic barrier function with respect to the network weights of CRBM- ℓ is computed as follows:

$$\frac{\partial}{\partial w_{ij}^{(\ell)}} f(w_{ij}^{(\ell)}) = \llbracket w_{ij}^{(\ell)} \rrbracket^- = \begin{cases} 2w_{ij}^{(\ell)}, & w_{ij}^{(\ell)} < 0 \\ 0, & w_{ij}^{(\ell)} \geq 0 \end{cases} \quad (65)$$

The gradient of the quadratic barrier function with respect to the parameter $\mathbf{a}^{(\ell)}$ is zero because the definition of the nonnegativity constraint does not involve any bias terms. Similarly, the gradient of the sparse regularization term

with respect to the model parameters of CRBM- ℓ is computed as:

$$\begin{aligned} \frac{\partial}{\partial w_{ij}^{(\ell)}} \left(f(p(\mathbf{h}^{(\ell+1)} | \mathbf{h}_s^{(\ell)})) \right) &= \frac{\partial}{\partial w_{ij}^{(\ell)}} \left(\frac{1}{1 + \exp\left(-\sum_{i=1}^{T_\ell} w_{ij}^{(\ell)} h_i^{(\ell)} - a_j^{(\ell)}\right)} \right) \\ &= \frac{h_j^{(\ell)} \times \exp\left(-\sum_{i=1}^{T_\ell} w_{ij}^{(\ell)} h_i^{(\ell)} - a_j^{(\ell)}\right)}{\left| 1 + \exp\left(-\sum_{i=1}^{T_\ell} w_{ij}^{(\ell)} h_i^{(\ell)} - a_j^{(\ell)}\right) \right|^2} \\ &= h_j^{(\ell)} \times \left(p(h_j^{(\ell+1)} = 1 | \mathbf{h}_s^{(\ell)}) - p^2(h_j^{(\ell+1)} = 1 | \mathbf{h}_s^{(\ell)}) \right) \\ &= \nabla w_{ij}^{(\ell)} \end{aligned} \quad (66)$$

$$\begin{aligned} \frac{\partial}{\partial a_j^{(\ell)}} \left(f(p(\mathbf{h}^{(\ell+1)} | \mathbf{h}_s^{(\ell)})) \right) &= \frac{\partial}{\partial a_j^{(\ell)}} \left(\frac{1}{1 + \exp\left(-\sum_{i=1}^{T_\ell} w_{ij}^{(\ell)} h_i^{(\ell)} - a_j^{(\ell)}\right)} \right) \\ &= \frac{-\exp\left(-\sum_{i=1}^{T_\ell} w_{ij}^{(\ell)} h_i^{(\ell)} - a_j^{(\ell)}\right)}{\left| 1 + \exp\left(-\sum_{i=1}^{T_\ell} w_{ij}^{(\ell)} h_i^{(\ell)} - a_j^{(\ell)}\right) \right|^2} \\ &= -\left(p(h_j^{(\ell+1)} = 1 | \mathbf{h}_s^{(\ell)}) - p^2(h_j^{(\ell+1)} = 1 | \mathbf{h}_s^{(\ell)}) \right) \\ &= \nabla a_j^{(\ell)} \end{aligned} \quad (67)$$

References

1. Hofmann T (2001) Unsupervised learning by probabilistic latent semantic analysis. *Mach Learn* 42(1):177
2. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. *J Mach Learn Res* 3(January):993
3. Blei DM, Lafferty JD (2005) Correlated topic models. In: *Proceedings of the 18th international conference on neural information processing systems*, MIT Press, Cambridge, MA, pp 147–154
4. Boulemden A, Tlili Y (2012) Image indexing and retrieval with pachinko allocation model: application on local and global features. In: *Proceedings of the 12th pacific rim conference on knowledge management and acquisition for intelligent systems*. Springer, Berlin, pp 140–146
5. Gehler PV, Holub AD, Welling M (2006) The rate adapting poisson model for information retrieval and object recognition. In: *Proceedings of the 23rd international conference on machine learning*. ACM, New York, pp 337–344
6. Salakhutdinov R, Hinton G (2009) Replicated softmax: an undirected topic model. In: *Proceedings of the 22nd international conference on neural information processing systems*. Curran Associates Inc., USA, pp 1607–1614
7. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. *J Am Soc Inf Sci* 41(6):391
8. Pecenov Z (1997) Intelligent image retrieval using latent semantic indexing. Master's thesis, Swiss Federal Institute of Technology
9. Zhang R, Zhang Z (2007) Effective image retrieval based on hidden concept discovery in image database. *IEEE Trans Image Process* 16(2):562

10. Lienhart R, Romberg S, Hörster E (2009) Multilayer pLSA for multimodal image retrieval. In: Proceedings of the ACM international conference on image and video retrieval. ACM, New York
11. Li P, Cheng J, Li Z, Lu H (2011) Correlated PLSA for image clustering. In: Advances in multimedia modeling, pp 307–316
12. Chiang CC, Wu JW, Lee GC (2012) Probabilistic semantic component descriptor. *Multimed Tools Appl* 59(2):629
13. Hörster E, Lienhart R, Slaney M (2007) In: Proceedings of the 6th ACM international conference on image and video retrieval. ACM, New York, pp 17–24
14. Greif T, Hörster E, Lienhart R (2008) Correlated topic models for image retrieval. University of Augsburg, Germany, July, Tech. rep
15. Li W, McCallum A (2006) Pachinko allocation: DAG-structured mixture models of topic correlations. In: Proceedings of the 23rd international conference on machine learning, ACM, New York, pp 577–584
16. Andrieu C, De Freitas N, Doucet A, Jordan MI (2003) An introduction to MCMC for machine learning. *Mach Learn* 50(1–2):5
17. Minka T, Lafferty J (2002) Expectation-propagation for the generative aspect model. In: Proceedings of the eighteenth conference on uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., pp 352–359
18. Casella G, George EI (1992) Explaining the Gibbs sampler. *Am Stat* 46(3):167
19. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *Science* 313(5786):504
20. Hinton G (2010) A practical guide to training restricted Boltzmann machines. *Momentum* 9(1):926
21. Srivastava N, Salakhutdinov R, Hinton G (2013) Modeling documents with a deep boltzmann machine. In: Proceedings of the twenty-ninth conference on uncertainty in artificial intelligence. AUAI Press, Arlington, Virginia, pp 616–624
22. Olshausen BA, Field DJ (2004) Sparse coding of sensory inputs. *Curr Opin Neurobiol* 14(4):481
23. Salakhutdinov R, Hinton G (2009) Deep boltzmann machines. In: Proceedings of the twelfth international conference on artificial intelligence and statistics, Clearwater Beach, Florida, pp 448–455
24. Brooks S, Gelman A, Jones GL, Meng XL (2011) Handbook of markov chain monte carlo. CRC Press, Boca Raton
25. Hinton GE, Salakhutdinov RR (2012) A better way to pretrain deep boltzmann machines. In: Proceedings of the 26th annual conference on neural information processing systems. Lake Tahoe, Nevada, pp 2447–2455
26. Bruna J, Mallat S (2013) Invariant scattering convolution networks. *IEEE Trans Pattern Anal Machine Intelligence* 35(8):1872
27. Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. *Nature* 401(6755):788
28. Poggio T, Girosi F (1998) A sparse representation for function approximation. *Neural Comput* 10(6):1445
29. Nguyen TD, Tran T, Phung DQ, Venkatesh S (2013) Learning parts-based representations with nonnegative restricted boltzmann machine. In: Proceedings of the Asian conference on machine learning. ACT, Canberra, pp 133–148
30. Jegou H, Douze M, Schmid C (2008) Hamming embedding and weak geometric consistency for large scale image search. In: Proceedings of the 10th European conference on computer vision: Part I. Springer, Berlin, pp 304–317
31. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the 2006 IEEE computer society conference on computer vision and pattern recognition. Vol 2, IEEE Computer Society, Washington, DC, pp 2169–2178
32. Philbin J, Chum O, Isard M, Sivic J, Zisserman A (2007) Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, New York, pp 1–8
33. Liu GH, Yang JY, Li Z (2015) Content-based image retrieval using computational visual attention model. *Pattern Recogn* 48(8):2554
34. Grubinger M, Clough P, Müller H, Deselaers T (2006) The iaprtc-12 benchmark: a new evaluation resource for visual information systems. In: Proceedings of international conference on language resources and evaluation. vol 5, ELRA, 2006, vol 5, p 10
35. Huiskes MJ, Thomee B, Lew MS (2010) New Trends and ideas in visual concept detection: the MIR Flickr retrieval evaluation initiative. In: Proceedings of international conference on multimedia information retrieval. ACM, New ork, pp 527–536
36. Deng L (2012) The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process Mag* 29(6):141