Deep Networks for Image Retrieval on Large-Scale Databases

Eva Hörster
Multimedia Computing Lab
University of Augsburg
Augsburg, Germany
hoerster@informatik.uni-augsburg.de

Rainer Lienhart
Multimedia Computing Lab
University of Augsburg
Augsburg, Germany
lienhart@informatik.uni-augsburg.de

ABSTRACT

Currently there are hundreds of millions (high-quality) images in online image repositories such as Flickr. This makes is necessary to develop new algorithms that allow for searching and browsing in those large-scale databases. In this work we explore deep networks for deriving a low-dimensional image representation appropriate for image retrieval. A deep network consisting of multiple layers of features aims to capture higher order correlations between basic image features. We will evaluate our approach on a real world large-scale image database and compare it to image representations based on topic models. Our results show the suitability of the approach for very large databases.

Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Applications; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms

Algorithms, Experimentation

1. INTRODUCTION

Large-scale community image databases such as Flickr are growing fast and contain hundreds of millions of images. They are currently mostly indexed and searched based on manually entered tags. However, the tags – if present at all – are provided by the creator of the picture and do not necessarily refer to the content shown. In this paper we concentrate on developing a retrieval approach which explores the usage of image features. Specifically we are interested in deriving a low-dimensional image description that enables fast retrieval of images of similar content.

Motivated by the promising results achieved by deep networks in information retrieval [10], we will exploit those models for deriving a low-dimensional description of the coarse image content. A deep network (DN) consists of multiple, non-linear layers each capturing the strong correlations of the feature activations in the level below. This way we compute a multi-level representation of each image; by decreasing the number of units in each higher layer the dimensionality of our input vector is reduced. By using only the low-dimensional image representation of the highest layer, we are able to perform fast and accurate retrieval as our experimental results show.

The works which are most similar to our approach are the application of DNs to information retrieval [10] and the very recent work of Torralba et al. [12] where a deep network is used for performing image recognition. In [12] the authors use a global image description in contrast to our local features. Also they apply the model in a different context: it is applied to a labeled image database as well as to a web database with images of size 40×40 pixels, containing mostly only one object. Other approaches for low-dimensional image representations in the context of large-scale image retrieval include topic-models, e.g. [8] used probabilistic Latent Semantic Analysis (pLSA) [6] based models, [7] applied Latent Dirichlet Allocation (LDA) [2] to derive a topic representation and [3] adopted the Correlated Topic Model (CTM) [1]. In the experimental evaluation we compare our approach with those models.

The paper is organized as follows. First we describe the computation of the basic image representation followed by an introduction to DNs in Sec. 2 and 3, respectively. In Sec. 4 we present our image retrieval approach and evaluate it experimentally in Sec. 5. Sec. 6 concludes the paper.

2. BASIC IMAGE REPRESENTATION

The first step in applying a deep network model to our image database is to obtain a basic representation for each image. Therefore we compute a visual word co-occurrence vector for each image. This kind of representation is very flexible and has shown good performance in various image analysis tasks. A co-occurrence vector for a specific image is derived by counting the number of occurrences of so-called visual words from a fixed sized visual vocabulary in that image. Note that the geometric relations between the different words and thus between the extracted visual features are completely ignored.

Visual words are derived by vector-quantizing local feature descriptors that are extracted at predefined locations and scales. There exist various techniques for learning visual words from local image features. In this work we derive the vocabulary by merging the results of multiple k-means clusterings on non-overlapping feature subsets. Therefore relatively small sets of features are selected randomly from all features and k-means clustering is applied to each subset.

MM'08, October 26–31, 2008, Vancouver, British Columbia, Canada. https://doi.org/10.1145/1459359.1459449

The means of each cluster are kept as visual words. Finally, the derived visual words of each subset are amalgamated into the vocabulary [8].

There are various possibilities for finding locations and scales of interest as well as for describing these regions of interest by features. In this work we will consider two different possibilities of defining interest points and scales for feature extraction: sparse features extracted at extrema in the difference of Gaussian pyramid [9] and dense features which are extracted at regular grid points at various scales. Furthermore we consider two different kinds of local features: SIFT features [9] and self-similarity features [11]. Both feature types have shown good performance in related image analysis tasks.

Given the vocabulary, features are extracted from each image first. Then, each detected feature vector is replaced by its most similar visual word defined as the closest word in the high-dimensional feature space. Finally, counting the specific word occurrences for each image results in the co-occurrence vectors: one for each image. These vectors are usually of high dimension.

3. DEEP NETWORKS

Having computed a basic visual representation for each image, we now apply a deep network model to derive a low-dimensional image representation. The applied deep network uses multiple, non-linear hidden layers and was introduced by Hinton et al. in [5] and [10]. It will be described in the following. The learning procedure for such a deep model consists of two stages. In the first stage, the pretraining, an initialization based on restricted Boltzmann machines (RBM) is computed. In the second stage it is refined by using backpropagation.

RBMs provide a simple way to learn a layer of hidden features without any supervision. They consists of a layer of visible units which are connected to hidden units using symmetrically weighted connections. Note that a RBM does not have any visible-visible or hidden-hidden connections (Fig. 1). Assuming binary vectors as our input, the energy of the joint configuration of visible, stochastic, binary units $\mathbf{v}$ and hidden, stochastic, binary units $\mathbf{h}$ is given by [5]:

$$E(\mathbf{v},\mathbf{h}) = -\sum_i b_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i h_j w_{ij} \qquad (1)$$

where v_i and h_j are the binary states of the visible and hidden units respectively, b_i and b_j their biases and w_{ij} the symmetric weights. The probability of a visible vector $\mathbf{v}$ given this model can be computed as follows:

$$p(\mathbf{v}) = \frac{\sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h}))}{\sum_{\mathbf{u},\mathbf{g}} \exp(-E(\mathbf{u},\mathbf{g}))}. \qquad (2)$$

Given the states of the visible units, the probability that a hidden unit h_j is activated on is:

$$p(h_j = 1|\mathbf{v}) = \sigma(b_j + \sum_i v_i w_{ij}) \qquad (3)$$

where $\sigma(x)$ denotes the logistic function. Similarly it holds:

$$p(v_i = 1|\mathbf{h}) = \sigma(b_i + \sum_j h_j w_{ij}) \qquad (4)$$

In order to learn the variables, i.e. the weights w_{ij} and the biases b_i, b_j, we apply one step contrastive divergence [4]:

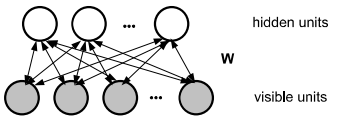

Figure 1: Restricted Boltzmann machine

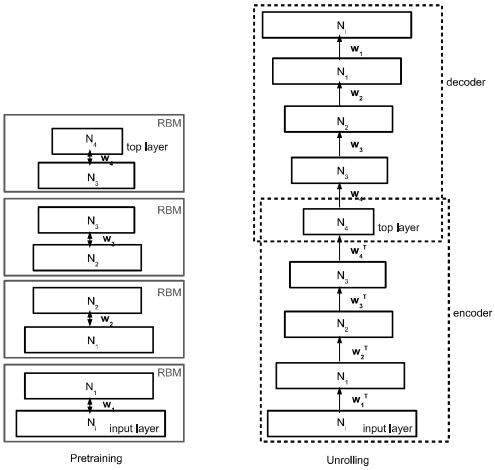

Figure 2: DN models: layer-by-layer pretraining (left); unrolling and fine-tuning (right)

$\Delta w_{ij} = \epsilon(< v_i h_j >_{data} - < v_i h_j >_{recon})$. A similar update equation is used for learning the biases.

To construct a deep network, Hinton et al. [5] propose to learn additional layers of features by treating the hidden states (or activation probabilities) of the lower level RBM as the visible data for training a higher level RBM, that learns the next layer of features (see Fig. 2 left). By repeating this greedy layer-by-layer training several times, we can learn a deep model which is able to capture higher order correlations between the input units.

After having greedily pretrained all layers, the parameters of the deep model are further refined. This is done by replacing the stochastic activities of the binary features by deterministic real-valued probabilities and unrolling the layers to create an autoencoder as proposed in [5] (see Fig. 2 right). Using the pretrained biases and weights to initialize the backpropagation algorithm, backpropagation is used to fine-tune the parameters for optimal reconstructing the input data.

4. IMAGE RETRIEVAL

In this work our aim is to study the representation of images by deep networks. When applying the DN model described in the previous section to our image data, some modifications to that model are necessary as the input vector at the lowest layer is a co-occurrence vector and not binary. We first divide each entry of the respective vector by the total number of visual words detected in the specific image. This creates a discrete probability distribution over the visual vocabulary for each image document. According to [5], the probabilities of the visible units given the hidden

ones can be modeled by a so called 'softmax' unit:

$$p(v_i = 1|\mathbf{h}) = \frac{\exp(b_i + \sum_j h_j w_{ij})}{\sum_k \exp(b_k + \sum_j h_j w_{kj})} \quad (5)$$

The learning rules for the weights are not affected by the usage of softmax units. However, the weights w_{ij} from visible unit i to hidden unit j are multiplied by the number of detected features N_d in image d, whereas the weights from hidden units to visible units remain w_{ij}. This is done to account for the fact that each image d may contain a different number of visual words depending on its size in case of densely extracted features or size and image structure in case of sparsely extracted features.

It should be noted that there are different possibilities to choose the type of unit at the top level of the network. In this work we will evaluate two different types of units: logistic units and linear units.

After pretraining the layers of the deep network, an autoencoder is created as described in the previous section. The parameters of the autoencoder are initialized with the pretrained biases and weights and refined using the backpropagation algorithm. For backpropagation the multi-class cross-entropy error function is used:

$$e = -\sum_i v_i \log(\hat{v}_i) \quad (6)$$

where $\hat{v}_i$ denotes the reconstruction of v_i by the autoencoder and v_i is the i-th component of the normalized input vector.

For image retrieval we apply the learned deep model to each image in the database and use its top-level unit values as its low-dimensional description. It should be noted that the mapping from the co-occurrence vector, i.e. the basic image description, to the high level representation only consists of a single matrix multiplication and single squashing function per network unit. Given a query image, we then retrieve images of similar content by comparing the high level image representations based on some distance metric. In this work we use the simple L1 distance metric.

5. EXPERIMENTAL EVALUATION

Dataset and implementation details: All experiments are performed on a real world database consisting of 246,348 images. The images were selected from all public Flickr images uploaded prior to Sep. 2006 and labeled as *geotagged* together with one of the following tags: *sanfancisco, beach* and *tokyo*. Of these images only images having at least one of the following tags were kept: *wildlife, animal, animals, cat, cats, dog, dogs, bird, birds, flower, flowers, graffiti, sign, signs, surf, surfing, night, food, building, buildings, goldengate, goldengatebridge, baseball*. Note that the tags are only needed for pre-selecting a subset of images from the entire Flickr database and are not used at any stage in our retrieval approach. We computed the visual vocabulary for each feature type from 12 randomly selected non-overlapping subsets, deriving a total vocabulary size of 2400 visual words. The trained DNs consisted of four hidden layers with a 2400-1000-500-250-50 structure. Thus, we obtain a 50-dimensional image description for image retrieval. We used 50,000 images for training, 25 iterations for pretraining each layer and 50 iterations to optimize the autoencoder.

Performance metric: To evaluate our approach, we judge its performance by users in a query-by-example task.

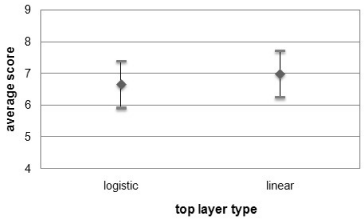

Figure 3: Avg. # of correctly retrieved images using DN-based image models with two different types of top layer units: logistic and linear.

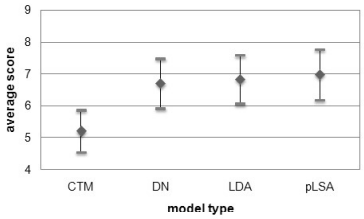

Figure 4: Avg. # of correctly retrieved images using different image models.

Here the objective is to obtain images with similar content to the given query image. We selected a total of 60 test query images in a random fashion. For each query image the 20 most similar images (including the query image) according to the L1 distance measure are returned to the users. For each experiment we asked ten users to judge the retrieval results by counting how many of the retrieved images show content similar to the query image. As the query image is counted too, the lowest number of correctly retrieved images is one and the largest 20. The average number of similar images over all queries for one retrieval technique is computed for each user to give the final score. Note that the user's judgment is subjective. As our test users varied for each experiment, we may derive different average results in different experiments even for the same approach and parameters. Therefore we will also show the standard deviation in addition to the average number of correctly retrieved images.

Experiments: First we examine the influence of the top layer type on our retrieval results. We compare the logistic unit type with the linear unit type using sparsely extracted SIFT features. As can be seen from the results in Fig. 3, the performances of both types differ only slightly with a small edge for the linear units. Thus we will use the linear units for our subsequent experiments.

Our second experiment compares the results of the DN-based image representation to other proposed approaches. For comparison we use three recently proposed approaches described in [8], [7] and [3] which have shown good performance in large-scale image retrieval tasks. The approaches are based on different topic models: the CTM, LDA and pLSA. All three models derive a low-dimensional topic-based representation from the original co-occurrence vectors. We train each model with 50,000 images to derive 50 topics, resulting in a 50-dimensional topic vector for image representation. Fig. 4 displays the average number of correctly retrieved images for each approach. Again we have assumed

Figure 6: Retrieval results for different local features; each top left image depicts the query image.

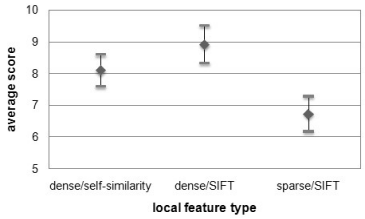

Figure 5: Avg. # of correctly retrieved images using DN-based image models with different local image detectors and descriptors.

sparsely extracted SIFT as the basic image feature. Clearly, CTM shows the worst performance. Further, it can be observed that DN, LDA and pLSA perform almost equally well with a very slight advantage for the pLSA. However DN has the advantage of modeling each image by multiple layers of feature activations. Here we only used the highest level in the model to represent an image. Nevertheless, there are possibilities of extending the approach by using multi-level representations. Further, the mapping from the high-dimensional word count vector to the low-dimensional representation is much faster for the DN model compared to inference in the LDA and pLSA model. As inference in those models requires multiple iterations of the (variational) EM algorithm, it is more costly than the feed forward structure of the DN, requiring only a matrix multiplication followed by a non-linearity per unit for each layer.

In our last experiment we compare three different types of visual features as the basic building block: sparse SIFT, dense SIFT and dense self-similarity features. The result is depicted in Fig. 5. It can be observed that dense feature extraction outperforms the sparse extraction. Furthermore the dense SIFT descriptor shows slightly better results than the densely extracted self-similarity features. One should note that the SIFT descriptor has 128 dimensions whereas the self-similarity feature consists of only 80 dimensions, which results in a faster vocabulary and co-occurrence vector computation. Finally we show some retrieval examples for different types of features in Fig. 6.

6. CONCLUSION

In this work we have proposed a novel approach for image search in very large databases, which applies a deep network to derive a low-dimensional image representation. We have evaluated our system experimentally. User studies have shown that the DN based image representation is suitable for retrieval in large, real world databases and that our system performs as well as other state of the art algorithms. Future work will consist of a more extensive evaluation. Also we want to extend our system to multiple modalities, i.e. including the noisy tags or other available image information.

7. REFERENCES

[1] D. Blei and J. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems 18*, pages 147–154. 2006.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[3] T. Greif, E. Hörster, and R. Lienhart. Correlated topic models for image retrieval. In *Technical Report TR2008-09, University of Augsburg, 2008.*

[4] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800, 2002.

[5] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[6] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.*, 42(1-2):177–196, 2001.

[7] E. Hörster, R. Lienhart, and M. Slaney. Image retrieval on large-scale image databases. In *Proc. ACM CIVR'07*, pages 17–24, 2007.

[8] R. Lienhart and M. Slaney. pLSA on large scale image databases. In *IEEE ICASSP*, 2007.

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[10] R. R. Salakhutdinov and G. E. Hinton. Semantic hashing. In *Proc. SIGIR Workshop on Information Retrieval and Applications of Graphical Models*, 2007.

[11] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *IEEE CVPR*, 2007.

[12] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large databases for recognition. In *IEEE CVPR*, 2008.