Similar and Class Based Image Retrieval Using Hash Code

B.Bharathi [1], Nagarjuna Reddy Akkim[2]
Faculty of computing, Sathyabama University, Chennai, India

Abstract
Image retrieval based on visual similarity has become an active research in recent years. For each and every image we create hash code by extracting the features of the image, here search is performed based on hamming distance. Unlike Euclidean metrics that offer continuous distances, the hamming distances are separate integer values. As a result, they are often many images that share equal hamming distance to a query, where there is a chance of error occurrence so we go for ranking of image. This paper introduces an new approach where similar and class based images can be retrieved based on hamming distance. This is the first time where the concept of data mining is being introduced into image searching, which will reduce the amount of time taken to retrieve similar images.

Index Terms
similar image retrieval, hash codes, hamming distance.

I. Introduction

Content based image retrieval is an lively research in recent years. While the traditional search heavily relay on keyword associated to images, while content-based search retrieves images based on visual similarity, so this is receiving increasing significance.
Generally a large-scale image search system consists of two key components—an valuable image feature representation and an competent search mechanism. It is known that the excellence of search results relies heavily on the representation power of image features. The latter, an efficient search mechanism, is critical since image are of high dimensions and current image databases are huge, on top of which comparing a query with every database sample is computationally prohibitive.
This work represents images using the popular bag-of-visual- words (BoW) structure, where local invariant image descriptors (e.g., SIFT)are extracted. The BoW features are then inserted into compact hash codes for efficient search. For this, we used state-of-the-art techniques including semi-supervised hashing and semantic hashing with deep belief networks. Hashing is better over tree-based indexing structures because it requires less memory and works better on high-dimensional images with the hash codes, similar images can be efficiently measured in hamming space by hamming distance.

II. RELATED ARTICLES

There are many surveys on general image retrieval task See Smeulders *et al.* for works from the Datta *et al.* for those from the past decade. Many people has been using simple features such as color and texture in systems developed in the early years.
Inverted index was originally proposed and is still very popular for document retrieval in the informational retrieval community. It is now being used in the field of image retrieval a BoW are very analogous to the BoW representation of textual credentials. In this structure, a list of references to each image for each visual word is created so that relevant images can be quickly located given a query with several words. A key difference of document retrieval from visual search, however, is that the textual queries usually contain very few words. For instance, on average there are merely 4 words per query in Google web search.2 while in the BoW representation, a single image may contain hundreds of visual words, resulting in a large number of candidate images.
Kulis and Grauman proposed unsupervised hashing. Among them, one of the most well-known hashing methods is Locality Sensitive Hashing (LSH). Recently, Kulis and Grauman extended LSH to work in arbitrary kernel space.
Chum *et al.* proposed min-Hashing to expand LSH for sets of features. Since these LSH-based methods use indiscriminate projections, when the dimension of the input space is high, many more bits are needed to achieve suitable performance. Motivated by this mainly,
Weiss *et al.* proposed a spectral hashing (SH) method which hashes the input space based on data division. SH also ensures that the projections are orthogonal and sample number is being balanced across diverse buckets. Although SH can achieve better performance than LSH with a fewer number of bits, it is important to underline that unsupervised hashing techniques are not better enough for similar image search.

III. SYSTEM FRAMEWORK

The proposed similar and class based image retrieval system is depicted in Fig. 2. To reach the goal of similar

Manuscript received March 5, 2015
Manuscript revised March 20, 2015

and class based image retrieval, we strap up a set of semantic concept classes, each with a cluster of representative images as shown on the left of the figure. Bag of visual words features (BoW) of all the images are inserted into hash codes, we calculate bitwise weights for each of the semantic concepts separately. The weight calculation process is done by an algorithm that lies in the very heart of our approach.

The flow chat on the right of Fig. 2 explains the process of online search. We first generate hash code of the query image, which is used to search images against in the predefined semantic classes. From there we group a large set of images which are near to the query in Hamming space, and use them to predict bitwise weights for the query. One hypothesis made here is that images around the query in Hamming space, together, should be able to infer query semantics. Finally, with the query-adaptive weights, images from the target database can be rapidly ranked by weighted Hamming distance to the query.

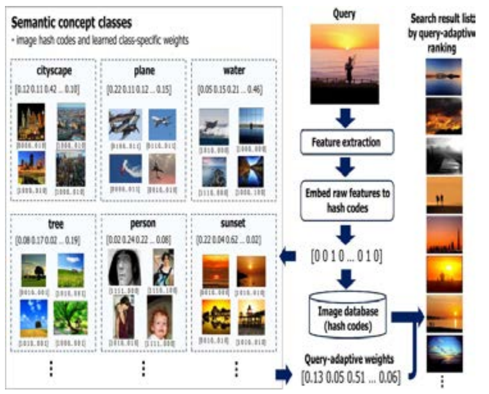

Fig.1. Framework of similar and class based image retrieval with hash codes

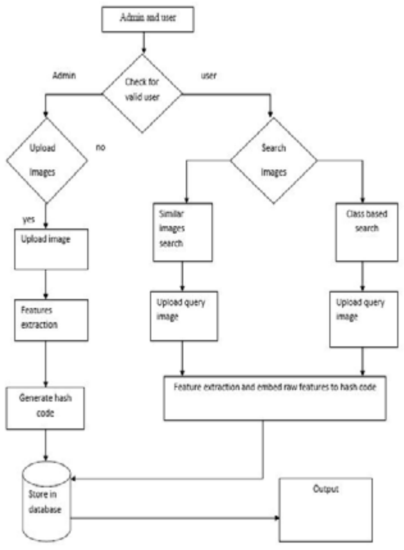

Fig.2. Functional Architecture

IV. HASHING

In this paper two state-of-the-art hashing techniques are adopted, *semi-supervised hashing* and semantic hashing with deep belief networks.

A. *Semi-Supervised Hashing*

Semi-Supervised Hashing (SSH) is an algorithm that leverages semantic similarities among labeled data while remains robust to over appropriate. The objective function of SSH consists of two major mechanism, supervised empirical fitness and unsupervised information theoretic regularization. Moreover, on one hand, the supervised part tries to minimize an empirical error on a small amount of labeled data. The unsupervised term, on the other hand, provides effective regularization by maximizing desirable properties like variance and independence of individual bits. Mathematically

$$J(\mathrm{H}) = \sum_{k=1}^{d} \left\{ \sum_{(\mathbf{v}_i,\mathbf{v}_j)\in\mathcal{M}} h_k(\mathbf{v}_i)h_k(\mathbf{v}_j) - \sum_{(\mathbf{v}_i,\mathbf{v}_j)\in\mathcal{C}} h_k(\mathbf{v}_i)h_k(\mathbf{v}_j) \right\} + \mu \sum_{k=1}^{d} \mathrm{var}[h_k(\mathbf{v})],$$

B. *Semantic Hashing With Deep Belief Networks*

Deep belief networks (DBN) was initially proposed for dimensionality reduction. It was recently being used for semantic hashing in large-scale search applications. Like SSH, to produce hash codes. DBN also requires image labels during exercise phase, so that images with the same label can be hashed into the same bucket. Since the DBN structure gradually reduces the number of units in each layer, the high-dimensional input of original image features can be projected into a compact Hamming space. In general DBN is a directed acyclic graph, where each node represents a stochastic variable. There are two main steps in using DBN for hash code generation, the learning of interactions between variables and the inference of Observations from inputs. The learning of a DBN with multiple layers is very hard since it usually requires estimating millions of parameters.

As shown by Hinton *et al.*in that the training process can be much more efficient if a DBN is specifically planned based on the RBMs. Each single RBM has two layers containing output visible units and hidden units, and multiple RBMs can be stacked to form a deep belief net. Starting from the input layer with dimension, the

network can be specifically designed to reduce the number of units, and finally output compact – dimensional hash codes. To obtain optimal weights in the entire network, the training process of a DBN has two critical stages: unsupervised pre-training and supervised fine-tuning.
The pre-training phase is progressively executed layer by layer from input to output, aiming to place the network weights to suitable neighborhoods in parameter space. After achieving union of the parameters of one layer via Contrastive Divergence, the outputs of this layer are fixed and treated as inputs to drive the training of the next layer.

V. QUERY-ADAPTIVE SEARCH

Scalable image search can be performed in Hamming space using Hamming distance. By finding, the Hamming distance between two hash codes is the total number of bits at which the binary values are different. Specific indices locations of the bits with different values are not considered For example, given three hash codes **X**=1100 ,**Y**=1111,and **Z**=0000, the Hamming distance of **X** and **Z** is equal to that of and , regardless of the fact **Z** that differs from **X** in the first two bits **Y** differs in the last two bits. Due to this nature of the Hamming distance, practically there can be hundreds or even thousands of samples sharing the same distance to a query. Going back to the example, suppose we knew that the first two bits are more important (discriminative) for **X**, then **Y** should be ranked higher than **Z** if was the query.

A. *Learning Class-Specific Bitwise Weights*

A class-specific bitwise weight is used for to calculate the query-adaptive weights for a number of semantic concept classes (e.g., scenes and objects). Imagine that we have a dataset of semantic classes, each with a set of representative images. We learn bitwise weights separately for each class, with an objective of maximizing intra-class similarity as well as maintaining inter-class relationship. Properly, for two hash codes and in classes and correspondingly, their proximity is measured by weighted Hamming distance

B. Computing Query-Adaptive Bitwise Weights

As described in Fig.1, images and the learned weights of the predefined semantic concept classes form a semantic database for fast computation of the query-adaptive bitwise weights. Given a query image and its hash codes, the objective here is to adaptively determine a suitable weight vector, such that weighted Hamming distance (WHD) can be applied to compare x_q with each hash code X_t in the target database

$$\mathrm{WHD}(\mathbf{x}_q, \mathbf{x}_t) = \|\mathbf{a}_q \circ \mathbf{x}_q - \mathbf{a}_q \circ \mathbf{x}_t\|^2.$$

Algorithm 1 Learning class-specific bitwise weights.

```
 1: INPUT:
    Hash codes $\mathcal{X}$;
    Class similarity $s_{ij}$ in original image feature space,
    $i, j = 1, \dots, k$.
 2: Compute $\mathbf{c}^{(i)}$ using (2);
 3: Initialize $\mathbf{a}_j = 1/d$, $j = 1, \dots, k$;
 4: Repeat
 5:   For $i = 1, \dots, k$
 6:     Compute $Q_i$, $\mathbf{p}_i$, and $t_i$ using (11)–(13);
 7:     Solve the following QP problem:

        $\mathbf{a}_i^* = \arg\min_{\mathbf{a}_i} \frac{1}{2}\mathbf{a}_i^\top Q_i \mathbf{a}_i + \mathbf{p}_i^\top \mathbf{a}_i + t_i$
              s.t. $\mathbf{a}_i^\top 1 = 1$ and $\mathbf{a}_i \geq \mathbf{0}$;

 8:     Set $\mathbf{a}_i = \mathbf{a}_i^*$;
 9:   End for
10: Until convergence
11: OUTPUT:
    Class-specific bitwise weights $\mathbf{a}_j$, $j = 1, \dots, k$.
```

VI. RESULTS

Let us first generate hamming code to the number these total number of images in our database so that we can calculate hamming distance to the query image to the database images. The images that are having les hamming Distance are said to be most relevant image.

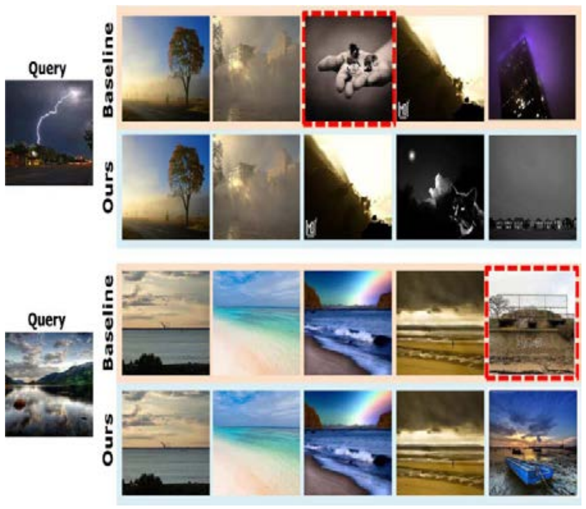

Fig 3. Top 5 returned images of two example queries, using the 48-bit DBN codes.

Hash codes will reduce the amount of time taken to search an image, because the content of image is stored in the form of hash code.

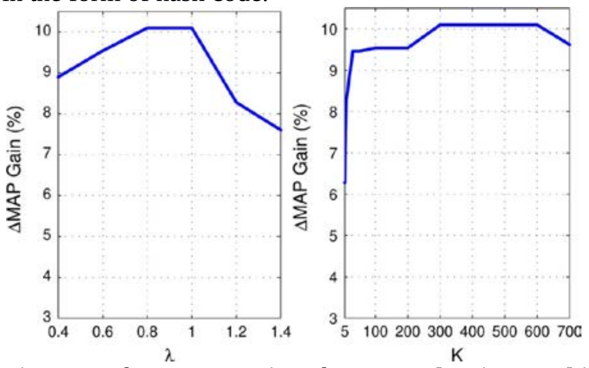

Fig.4. Performance gain of query-adaptive ranking versus parameters -k (left) and k (right), using DBN codes.

VII. CONCLUSION

The similar and class based image retrieval using hash code main aim is to provide better search of images using image a query. Hash code for image searching is being used for the first time in this paper.

VIII. ACKNOWLEDGEMENT

I would like to thank Dr.B.Bharathi, Head of the Department, Department of Computer Science and Engineering for her encouragement and support

REFERENCES

[1] Yu-Gang Jiang, Jun Wang, Xiangyang Xue “Query-adaptive image search with hash codes” IEEE transactions on multimedia, vol. 15, no. 2, February 2013

[2] A. Torralba, R. Fergus, and W. Freeman, “80 million tiny images: A large data set for nonparametric object and scene ecognition,” *IEEE Trans. Pattern Anal.Mach. Intell.*, vol. 30, no. 11, pp. 1958–1970, Nov.2011

[3] J. Wang, S. Kumar, and S.-F. Chang, “Sequential projection learning for hashing with compact codes,” in *Proc. Int Conf.Machine Learning*,2010.

[4] R. Datta, D. Joshi, J. Li, and J. Z.Wang, “Image retrieval: Ideas, influences, and trends of the new age,” *ACM Comput. Surveys*, vol. 40, no. 2, 2008.

[5] R.-S. Lin, D. A. Ross, and J. Yagnik, “Spec hashing: Similarity preserving algorithm for entropy-based coding,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010.

[6] Y. Mu, J. Shen, and S. Yan, “Weakly-supervised hashing in kernel space,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*,2010.

[7] W. Liu, J. Wang, S. Kumar, and S.-F. Chang, “Hashing with graphs,” in *Proc. Int. Conf. Machine Learning*, 2011.

[8] H. Xu, J. Wang, Z. Li, G. Zeng, S. Li, and N. Yu, “Complementary hashing for approximate nearest neighbor search,” in *Proc. IEEE Int.Conf. Computer Vision*, 2011.

[9] E. Horster and R. Lienhart, “Deep networks for image retrieval onlarge-scale databases,” in *Proc. ACM Int. Conf. Multimedia*, 2008.

[10] G. Shakhnarovich, P. Viola, and T. Darrell, “Fast pose estimation with parameter-sensitive hashing,” in *Proc. IEEE Int. Conf. Computer Vision*, 2003.