Efficiently Retrieving Images that We Perceived as Similar

Abstract

Despite growing interest in using sparse coding based methods for image classification and retrieval, progress in this direction has been limited by the high computational cost for generating each image's sparse representation. To overcome this problem, we leverage sparsity-based dictionary learning and hash-based feature selection to build a novel unsupervised way to efficiently pick out a query image's most important high-level features that can determine to which group we would visually perceived as similar. Moreover, the method is adaptive to current retrieval database. The preliminary results based on L1 feature map show the method's efficiency and accuracy from the visual cognitive perspective.

Motivation and Introduction

As the amount of digital data grows in unprecedented speed, new opportunities come with new challenges. The real value of big data lies in the ability to extract from it meaningful, even insightful information, rather than the "big" itself. Furthermore, many applications also require information to be retrieved *fast*. *Efficient* similar image retrieval thus becomes an important problem in the field of artificial intelligence with many real-world applications. The task is closely related to the nature of *human cognition* since any definition of *similarity* is meaningful only when it coincides with human feeling. Though the similar-or-not decision comes intuitively in no time for human, to find a well-defined decision guideline for computers is extremely hard. To resolve this stark discrepency that can inhibit human-machine cooperation, in this paper, we propose a novel method that emulates several important aspects of actual neurophysiological mechanisms, including *sparse coding* in primary visual cortex (V1), *synaptic plasticity*, and *mutual inhibition* between neurons. The mechanisms are not chosen randomly, instead, they complement each other's weak points for overall improvements without sacrificing efficiency.

Given unlabeled data, *sparse coding* provides a class of algorithms capable of extracting higher-level features that are actually more cognitively effective than hand-picked ones by emulating partial activity of neurons. The features can be regarded as the most representative building blocks by which the input data can be reconstructed most *efficiently* – highest accuracy with fewest elements used. The features form the *overcomplete* bases that resemble the *receptive fields* of neurons of in the visual cortex, making sparse coding a more appropriate medium than other widely-used computer vision features such as SIFT (Lowe 1999), GIST (Oliva et al. 2001), HOG (Dalal et al. 2005) etc., to bridge human cognition and algorithmic way of learning.

Unfortunately, though there are works (Ge et al. 2013) that tackle the image search problem by first representing query images as their sparse codes, the high computational cost involved in sparse code calculation renders it infeasible for (near) realtime retrieval tasks. Thus, we proposed in this paper a novel approach to overcome the performance bottleneck by a change of perspective: equipped with already-learned features stored as a *dictionary*, can we filter out the dominant features *directly* from an image, without resorting to the *complete* set of its sparse code? By regarding those selected features collectively as the image's *hash value*, the method actually transforms the sparse coding based approach to one more similar to efficient hash-based methods. In the following, we would use basis and feature interchangeably, and refer to the group of importatnt features learned as the *dictionary*.

Considering the fact that sparse representation is based on an *overcomplete* basis (Olshausen and Field 1996), we use concepts similar to that of inner-products and orthogonal decompositions to aid our feature selection. Overcomplete basis emerges as a natural result of the *sparsity* constraint. Sparsity requirement goes against using only independent basis to assemble other important features, instead, if a feature is popular and representative enough, it would be more beneficial to add it into the dictionary to improve sparsity. As a result, apart from the robustness it provides, the seemingly redundant overcompleteness implies the vectors' independent importance in representing different types of data. This forms the reason behind choosing similar features only exclusively, i.e. similar features are mutually inhibitive.

By combining the inhibition-like decomposition scheme with an additional mechanism that emulates synaptic plasticity to add adaptiveness given a static pre-learned dictionary, we are able to come up with a retrieval system with hash-based-like efficiency, and performance comparable to

GIST-based approach. Experiments based on *SUN397 scene benchmark* confirmed the effectiveness and efficiency of our method.

Proposed Retrieval System

System framework

A complete view of our proposed system is shown here for easier reference. Each step will be discussed in more detail in the following sections.

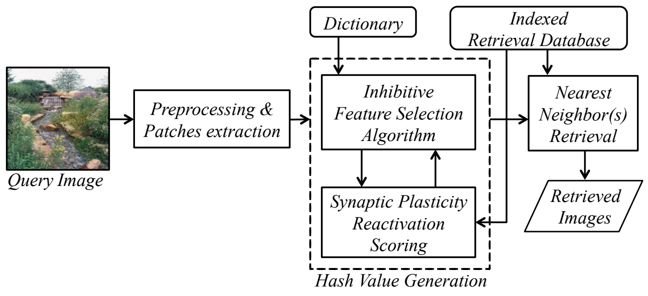

Dictionary
Indexed Retrieval Database
Query Image
Preprocessing & Patches extraction
Inhibitive Feature Selection Algorithm
Synaptic Plasticity Reactivation Scoring
Hash Value Generation
Nearest Neighbor(s) Retrieval
Retrieved Images

Figure 1: Proposed system. Given a query image, the system extracts from the database the required number of similar images in real time. Note that the dictionary is trained in advance using a *different* set of images, though of the same categories.

Image Preprocessing

Before being fed into the retrieval system or used for dictionary training, all images are first *whitened* to remove information redundancy that lies in correlations between neighboring pixels. This step is very fast while allowing significant improvement in feature learning quality, as the features to be learned would become less correlated and of the same variance.

Offline Unsupervised Dictionary Learning

Sparse-based dictionary learning has been proved to be especially effective in capturing discriminative features of scene images. We use the *Efficient sparse coding algorithms* proposed in (Lee et al. 2007) to build the dictionary of features from a learning set of 1000 images of size 200×200 pixels, the result is shown in Fig. 2. The Achilles' heels of sparse code dependent methods are rarely their performance, but the *time* needed for image encoding. In our method, sparse code calculation is done *once and for all* during dictionary learning, off-line and in advance. The calculation can be done on an arbitrary set of images by some powerful computer not contained in the retrieval system itself. Only the resulted dictionary is required.

Hash Value Generation

Inspired by the efficient *localitive sensitive hashing (LSH)* proposed by (Andoni et al. 2008), where high dimensional data can be projected to lower dimensional space with similarity preserving promise, this hash value generation step plays central role in our proposed method. Instead of determining two images' similarity by directly computing the difference between their sparse codes, we use the *projection* of patches onto dictionary as an emulation of image-triggered neuron excitation: bases with longest projected *length* correspond to the most responsive neurons. We proposed two neural-inspired procedures that when used cooperatively, can generate each image's hash value adaptive to current retrieval database. For each image, this step can efficiently select from the dictionary a group of (non-redundant) features that together captures the image's characteristic. "Which features are chosen from the static dictionary" can be regarded as a form of hash value. If our method can effectively select features in a way like how we recognize similarity, the cognitively similar images would have similar hash values, i.e. have similar features selected. This is indeed the case as will be shown in the experiment.

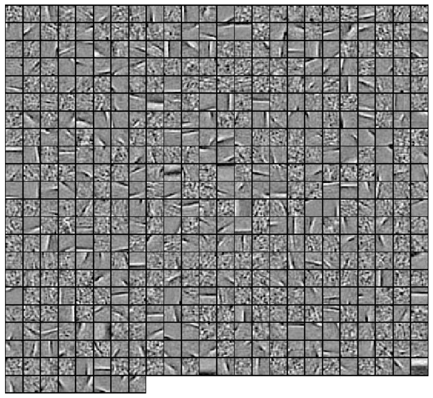


Figure 2: The dictionary learned off-line.

The two neural-inspired procedures: *inhibitive feature selection*, and *synaptic plasticity reactivation scoring* focus on the mutual inhibition of neurons and the connection weight tuning based on experience, respectively. As also will be shown in the experiment, the two mechanisms reinforce each other and leads to much better result when used together.

Inhibitive Feature Selection (IFS) Algorithm Dictionary as an overcomplete basis comes as a natural result under sparsity constraints. Given a training set of n input image patches $\overrightarrow{i^{(1)}}, \overrightarrow{i^{(2)}}, ..., \overrightarrow{i^{(n)}}$, assume that their sparse coefficients given the to-be-learned basis are $\overrightarrow{c^{(1)}}, \overrightarrow{c^{(2)}}, ..., \overrightarrow{c^{(n)}}$, the basis $\overrightarrow{b^{(1)}}, \overrightarrow{b^{(2)}}, ..., \overrightarrow{b^{(m)}}$ that form the learned dictionary is the solution to the optimization problem:

$$\text{minimize} \sum_{j=1}^{n} \|\overrightarrow{i^{(j)}} - \sum_{k=1}^{m} \overrightarrow{b^{(k)}} c_k^{(j)}\|^2 + \beta \sum_{j=1}^{n}\sum_{k=1}^{m} \phi(c_k^{(j)})$$

$$\text{subject to} \qquad \|\overrightarrow{b^{(k)}}\|^2 \le \delta,\ \forall k = 1, ..., m$$

Sparsity constraint (the second term) makes it more optimal to reconstruct each training image within certain accuracy using as few features from the dictionary as possible, thus effectively extracting the most representative features, not orthogonal features, out of the training database. This constraint is reasonable because features actually reside in very-high-dimensional vector space(s); what we try to capture in the dictionary are already their projections. A projected feature is not necessarily independent from other features' projections. In other words, features that are only slightly different in the relatively low dimensional vector space may be the main cause of their *non*-similarity.

As we use projection as our efficient way of gauging each basis's relevance with the (image) patch at hand, we are also confronted with the problem that projecting onto similar bases lead to similar strength of response. Considering the theory of sparse coding, only one of them should be selected. We choose the one with the highest response $\widehat{b}$ (i.e. with the longest projected length) *and subtract this component from the input vector* $\overrightarrow{i}$, to form the vector $\overrightarrow{i'}$ used in the next projection onto the dictionary. i.e. $\overrightarrow{i'} = \overrightarrow{i} - \frac{\overrightarrow{i} \cdot \widehat{b}}{\|\widehat{b}\|^2} \widehat{b}$. Iteratively, we can filter out the top n strong responsive features as our hash value for each patch vector, where n is flexible to the application at hand. Though we came to the *subtraction* from a mathematical viewpoint, this coincides well with the inhibitive nature between neurons with similar receptive field.

Algorithm 1 IFS Algorithm

```
($\tau$: How many basis vectors to be selected.)
$k$ = Number of patches;
$m$ = Dimension of patches vectors and basis;
$n$ = Number of basis in the dictionary;
$Data \in R^{k \times m}$ // each image patch is a row vector;
$Dictionary \in R^{m \times n}$ // each basis forms a column vector;
$Projection \leftarrow Data \cdot Dictionary$;
for $i = 1; i < \tau; i++$ do
    for all $j,\ 1 < j < k$ do
        $value^{(j)}_{max} \leftarrow max(j^{th}$row of $Projection)$
        $index^{(j)}_{max} \leftarrow position(max(j^{th}$row of $Projection))$
        $(p_j \equiv j^{th}$ row of $Data)$
        $p_j \leftarrow p_j - value^{(j)}_{max} \cdot$ (basis at $index^{(j)}_{max})^T$;
    end for
    $Projection = Data \cdot Dictionary$
end for
```

Synaptic Plasticity Reactivation Scoring Recall that the dictionary is trained beforehand on a different database and will stay static throughout the query process, it will be beneficial to make the retrieval process *adaptive* to current retrieval database without changing the dictionary itself. Actually, this is what the nerons are doing everyday – the synaptic weights are altered to adapt to the survival tasks confronted. We incoporate this weight tuning scheme into our hash value generation by *repeating a query image's stimulus* using multiple groups of patches. Given an input image, we extract from it multiple sets of patches. Each set of patches goes through the *IFS* step and selects its set of important features. Everytime a set finishes its selection, the *weights* of those images *in retrieval database* that share the most of this iteration's selected features are scaled up while all other weights are scaled down. In the end, the images with the highest *weights* are retrieved as answer. The designed repeated stimulation from the same query image, and the corresponding weight updates not only average out possible noise, but also take current task condition into account.

Algorithm 2 SPRS Algorithm

```
1: $r_{inc} > 1; 0 < r_{dec} < 1$ // scaling parameters
2: $averageNumber \in N$
3: $reentrantNumber \in N$
4: $n =$ Total Number of Images in Retrieval Database.
5: $scoring = [1]_{1 \times n}$
6: $distance \in R^{1 \times n}$ // the ouput from the distance metric
7: for $i = 1; i < averageNumber; i++$ do
8:     $distance \leftarrow$ the metric result from IFS Algorithm
9:     for $j = 1; i < reentrantNumber; i++$ do
10:        $index_{closest} \leftarrow sort(distancd(:), j)$
11:        $scoring(index_{closest}) = scoring(index_{closest}) \cdot c_{enlarge}$
12:    end for
13:    $scoring = scoring \cdot c_{shorten}$
14: end for
```

Experimental results

We evaluate our proposed system on a subset of *SUN397 scene benchmark* dataset and it contains 10 categories from all three top-level partitions (indoor, outdoor natural, outdoor man-made): bamboo forest, beach, botanical garden, corridor, cottage garden, hayfield, mountain snowy, waterfall block, wheat field, and wine cellar barrel storage. To accelerate training as well as later query stage, we randomly extract 50 patches of 14×14 pixels from each 200×200 pixels image as our input. The results are shown in Fig. 3 and Fig. 4

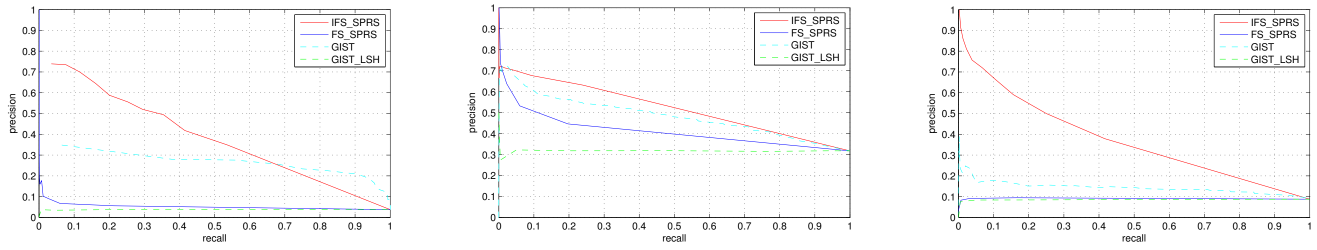
The average runtime for one query that retrieves top 8 results from a retrieval database of 3853 images is *10 seconds* on a *MacBook Air with 4 GB memory.*

Improvement with Inhibition Considered

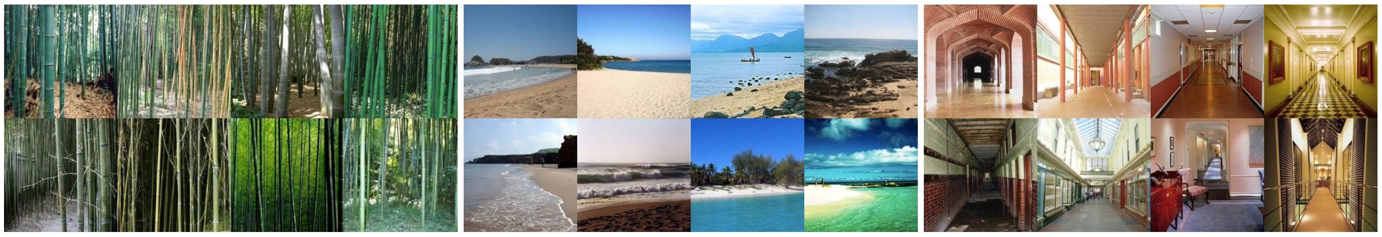
Under our sparse coding dictionary framework, we implemented *two* encoding method. One is a direct transplant of localitive sensitive hashing: reducing the high dimensional data into lower dimension by projecting onto the dictionary, without considering inhibitions between similar features, although the reactivation scoring is included. In fact, this is our first proof-of-concept method to explore the effectiveness of the sparse coding based framework *with adaptiveness*; we will refer to this first method as *feature selection algorithm*, denoted as *FS* in the figures. Another one is our novel *inhibitive feature selection algorithm* with both inhibition and adaptiveness considered, denoted as *IFS_SPRS* in the figures. As shown in Fig. 3(a), adding inhibition into the flow can lead to significant performance improvement.

Comparison with GIST-based methods

As we use a *scene* image database, to be fair, we compare our results with GIST-based methods, which is well-suited for capturing scene related features. GIST features are extracted directly from all 200×200 images using methods



(a)



(b)

Figure 3: (a) Precision& Recall curves with 10 queries of each category; our proposed method (IFS_ SPRS) outperforms the baselines in different image categories. (b) An example of top 8 retrieved results from one of the 10 random queries. (Categories from left to right: bamboo forest, beach, corridor.)

proposed by (Oliva el al. 2001). The first baseline method *GIST* simply extracts 128-bits GIST feature of the query image and finds the ones with nearest GIST features from the retrieval database. Another one, *GIST_LSH* accelerates the retrieval by generating hash values through random projection of images' GIST features. Experiments show that our method performs better than GIST-based ones for most cases (Fig. 3 and Fig. 4).

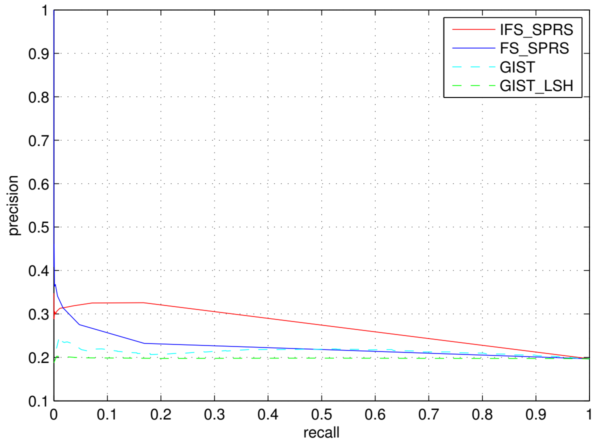


Figure 4: Precision & Recall curve for 100 random queries. Note that our methods' effectiveness is even more apparent than the results with 10 queries shown in Fig. 3(a).

Conclusion

In this paper, we proposed a real-time similar image retrieval system that based its similarity judgement on sparse coding based features, which are closer to human perception. Apart from maintaining a pre-trained dictionary as a feature storage, we overcome the obstacles of sparse coding's high computational complexity by neural-inspired strategies, including *inhibitive feature selection* that improves the quality of extracted features for a query image without computing its sparse code. Instead, projection similar to those used in localitive sensitive hashing is adopted for efficiency reason. Another strategy emulates *synaptic plasticity* to bring adaptivity into the static dictionary. Experimental results based on a retrieval database with 3853 images demonstrate the effectiveness of our methods, and show the large improvement from introducing inhibition between similar features.

References

Ge, Tiezheng, Qifa Ke, and Jian Sun. ”Sparse-Coded Features for Image Retrieval.” (2013).

Wright, John, et al. ”Sparse representation for computer vision and pattern recognition.” Proceedings of the IEEE 98.6 (2010): 1031-1044.

Sivaram, Garimella SVS, et al. ”Sparse coding for speech recognition.”Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on. IEEE, 2010.

Olshausen, Bruno A., and David J. Field. ”Sparse coding with an overcomplete basis set: A strategy employed by V1?.” Vision research 37.23 (1997): 3311-3325.

Lee, Honglak, et al. ”Efficient sparse coding algorithms.” Advances in neural information processing systems 19 (2007): 801.

Lowe, David G. ”Object recognition from local scale-invariant features.”Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Vol. 2. Ieee, 1999.

Oliva, Aude, and Antonio Torralba. ”Modeling the shape of the scene: A holistic representation of the spatial envelope.” International journal of computer vision42.3 (2001): 145-175.

Dalal, Navneet, and Bill Triggs. ”Histograms of oriented gradients for human detection.” Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.

CACM survey of LSH (2008): ”Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions” (by Alexandr Andoni and Piotr Indyk).Communications of the ACM, vol. 51, no. 1, 2008, pp. 117-122. directly from CACM