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. The preliminary results based on L1 feature map show the method's efficiency and high accuracy from the visual cognitive perspective. Finally, we consider a more general problem of how to make the pre-learned dictionary to adaptively refine the features contained according to past queries.

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, 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.

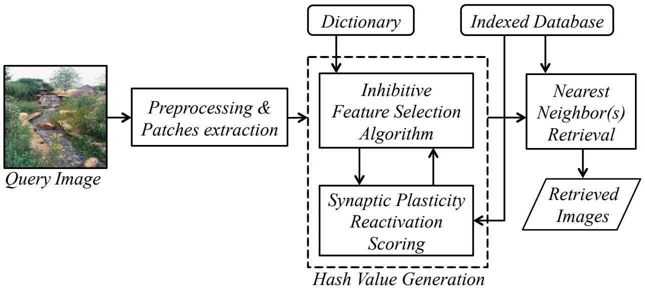

Dictionary
Indexed Database
Inhibitive Feature Selection Algorithm
Preprocessing & Patches extraction
Nearest Neighbor(s) Retrieval
Query Image
Retrieved Images
Synaptic Plasticity Reactivation Scoring
Hash Value Generation

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 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

This is the main step of our proposed method. Instead of determining two images' similarity by directly computing the difference between their sparse codes, we use the *projection* of an image onto dictionary basis as an emulation of image-triggered neuron excitation. We proposed two neural-inspired procedures that when used cooperatively, can generate each image's hash value. 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.

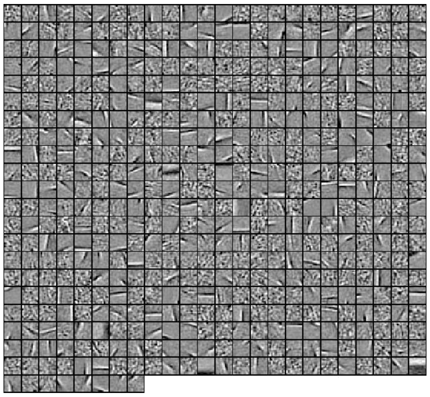


Figure 2: The dictionary learned off-line.

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

$$\text{minimize} \sum_{i=1}^{m}$$

Sparsity constraint 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.

-Robustness vs entropy: when two elements are selected together with high probability, one of them becomes redundant as it provides merely any extra information. But this should not be the case. Since our retrieval framework encode the image pattern of natural images into sparsity-based dictionary, we are motivated to select effective feature, especially those have high response to patches of natural images. Inspired by localitive sensitive hashing proposed by (Andoni et al. 2008), where high dimensional data can be projected to lower dimensional space with similarity preserving promise, we propose our novel algorithm to find out the atom of feature pattern in the dictionary to perform our hash-based dimensional reduction.

Firstly, we project our patches vector onto the atom of dictionary to get the highest values of the result for each vector of patches and have another zero array with the same size. We call those atoms strong responsive to the corresponding patches vector. Then, we set the value of each patches vector at corresponding atom of dictionary to be one.

Secondly, we substract the strong responsive atom from the corresponding patches vector in order to select second strong responsive atom with respect to the corresponding patches vector.

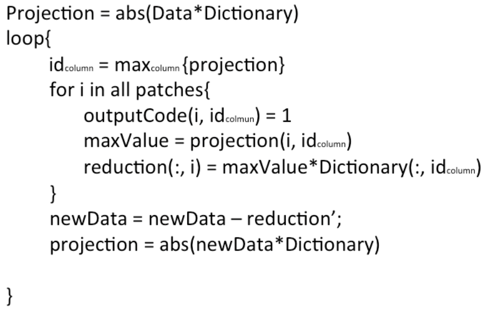
Iteratively, we will rank out the top n strong responsive atoms as our output for each patch vector. By this way, we can encode the raw data directly by the ranking of the response of corresponding atom based on sparsity based dictionary and we will show that the result has some effects consistent with our visual system.

Algorithm 1 IFS Algorithm

```
$Data \in k \times m;\ image\ patches$
$Dictionary \in m \times n;\ learned\ dictionary$
$Reduction \in m \times n;\ inhibitive\ parameter$
$Projection = Data \bullet Dictionary \in k \times n$
for $i = 1; i < n; i++$ do
    Get the maximum column value and ID for each row $\in Projection$
    for all $p \in image\ patches$ do
        $V_p \in \max \{\ basis\ w.r.t.\ image\ patch\ \}$
        $Reduction(:, p) = V_p \bullet Dictionary(:, ID_{max}(p))$
    end for
    $Data = Data - Reduction^T$
    $Projection = Data \bullet Dictionary$
end for
```

Algorithm 2 SPRS Algorithm

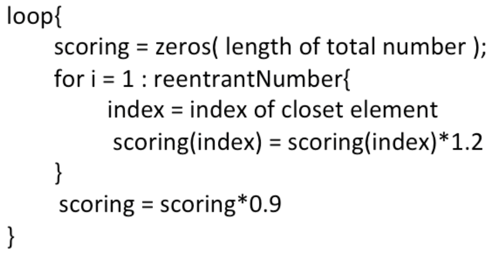
```
for $i = 1; i < 10; i++$ do$scoring \in 1 \times (the\ number\ of\ images\ in\ database)$ ;
    for $j = 1; i < reentrant\ number; i++$ do
        Get the index of the closet number
        $scoring(index) = scoring \times 1.2$
    end for
    $scoring = scoring \times 0.9$
end for
```



```
Projection = abs(Data*Dictionary)
loop{
      id_column = max_column{projection}
      for i in all patches{
            outputCode(i, id_colmun) = 1
            maxValue = projection(i, id_column)
            reduction(:, i) = maxValue*Dictionary(:, id_column)
      }
      newData = newData – reduction';
      projection = abs(newData*Dictionary)

}
```

Figure 3: Auto-learned feature selection algorithm (ALFSA)



```
loop{
      scoring = zeros( length of total number );
      for i = 1 : reentrantNumber{
            index = index of closet element
             scoring(index) = scoring(index)*1.2
      }
       scoring = scoring*0.9
}
```

Figure 4: Response Reactivation Scoring

Synaptic Plasticity Reactivation Scoring

Experimental results

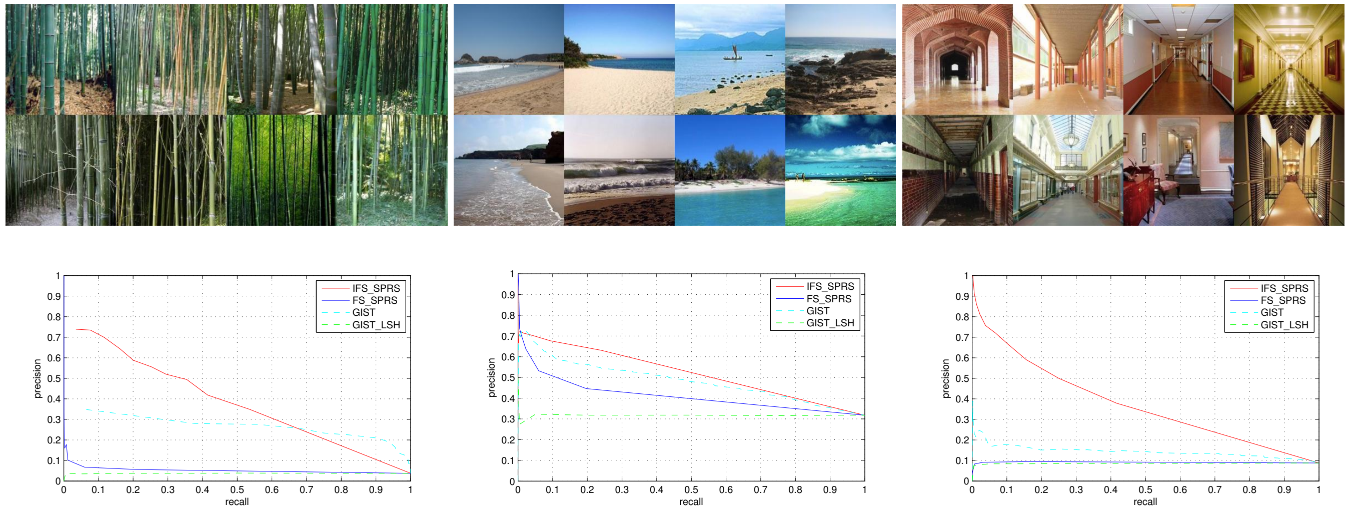
We evaluate our proposed system on a subset of *SUN397 scene benchmark* dataset and it contains 10 categories from all three top-level partitions (indoor, outdoor natural, outdoor man-made): bamboo forest, beach, botanical garden, corridor, cottage garden, hayfield, mountain snowy, waterfallBlock, wheat field, and wine cellearBarrelStorage. To accelerate training as well as later query stage, rather than directly representing the whole image by human-tuned feature sets, we randomly extract 50 14×14 pixels patches from each images.

Improvement under sparsity dictioanry framework

Under our sparsity dictioanry framework inspired from neuron activity, we implemented two encoding method: one is inspired from localitive sensitive hashing method, reducing the high dimensional data into lower dimension with near neighbors collide in the high probility. This is our first basic method to explore the effectiveness of image retrieval under such a sparsity dictionary framework and we call it feature selection algorithm, FS in the figures; Another one is our novel Inhibitive Feature Selection Algorithm algorithm called IFS_SPRS in out figures. We largely improved the original performance in FS_SPRS and we can see the outperformance in the Fig 5.

Comparison with Human-tuned feature methods

Due to our scene dataset, to be fair, we employ GIST features to do the evaluation. For comparison, we extract



(b)

Figure 5: Precision & Recall curve with 10 random queries for each different category. In each case, the set of images are top 8 retrieved results given a certain query image. Our proposed algorithm outperforms the baselines in different categories

GIST features proposed by (Oliva el al. 2001) directly from 200x200 pixels images Due to the state of the art working under different framework from us, we compare our ALFS method with LSH-based scheme under our spetial sparse coding framework as the baseline and we will show how much we have improved under this novel framework for image retrieval as an example.

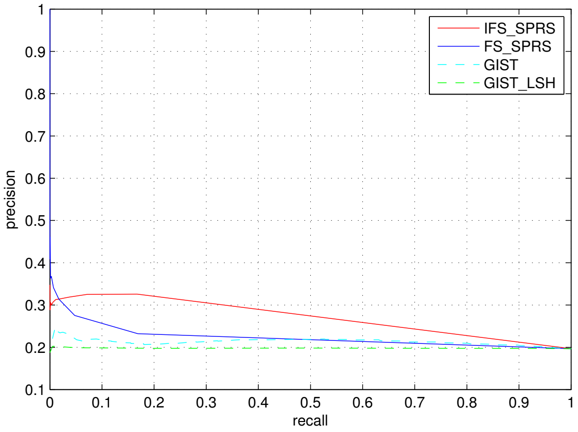


Figure 6: Precision & Recall curve for 100 random queries.

Conclusion

Rather than traditional human-turned feature extraction our cognitive system based on sparse coding successfully combine proposed novel auto-learned feature selection algorithm with sparsity-based dictionary to create ouw own discriminative code to retrieve natural images with high performance. The sparsity-based dictionary which capture basic elements consisting a natural image is a well learned structure to encode images. Although it needs more powerful algorithm and research in large-scale image retrieval or other big data, this is the promising direction of relative application.

Discussion

How to work with big data?

When the world is filled with big data, effective approach is needed to deal with such a challenge. Large-scale image with effective and reliable performance is one of examples. Recently, we are attempting to address an open question if there is new approach based our framework to handle this old but not well-solved problem. Our work lies in how we design the connection between visual neuron encoding simulation and image retrieval problem and how we investigate an effective large-sale image retrieval new candidate.

Dictionary is trained off-line, and can take full advantage of the large amount of data.

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