Robust Content Based Image Retrieval using Hierarchical Temporal Memory with Query by example

Dipan Kumar Pal, S.S.Tripathy, Viresh Ranjan, Avinash Das
Dept. of Electronics and Communication Engineering
Birla Institute of Technology, Mesra
Ranchi, India
Email: dipan2912@gmail.com, sstripathy@bitmesra.ac.in, vireshranjan@yahoo.co.in, avinashdas.bitmesra@gmail.com

***Abstract*— Content based image retrieval systems are now becoming important to efficiently store visual information. These systems employ techniques from artificial intelligence, machine learning and image processing to build a model of the image database. The model must grasp the semantics hidden within each image and only then robust querying by example is possible. We present a system in which the image is first classified to reduce the sample space for the actual retrieval. This is achieved by using a Hierarchical Temporal Memory (HTM) which models the mechanism of the human neocortex to store and process information. The color histogram technique is used for retrieving the final images post classification of the query image. The results obtained are very encouraging and serve to show that HTMs can be a very accurate and efficient solution to the image classification problem and be of significant use in image retrieval and related applications.**

Keywords-Hierarchical Temporal Memory; image retrieval; HTM; Content Based Image Retrieval

I. INTRODUCTION

The amount of visual information contained within a comprehensible image is much more than can be inferred from using superficial processing of color, shape and texture. Content Based Image Retrieval (CBIR) systems have emerged which aim to utilize the semantics and knowledge of the content of the image to develop a database to which queries can be made to be more complex so as to retrieve images of a more specific meaning or concept. The vast database of visual content available online for example, demands the need for a system which models and "understands" the images and what they represent and helps in efficient image and information search and inference. Content Based Image Retrieval (CBIR) systems utilize concepts and techniques from artificial intelligence, machine learning, pattern recognition and computer vision to search large image databases.

Some CBIR systems employ processing over text annotations and tags of the images whereas others employ image processing techniques to try and model the visual information encoded within each image.

Cox, Miller, Minka, Papathomas, Yianilos [5] present a Bayesian approach for CBIR. Their prototype system PicHunter takes into account a set of actions taken by the user and uses Bayes rule to get the output image. For image retrieval and classification, Malik, Frome and Singer [6] suggest a method based on local perceptual distance functions. Jain and Vailaya [7] present an approach which uses 3D color histogram and shape histogram. They have used a similarity measure based on Euclidean distance between the histograms. There also have been applications of HTMs in CBIR by Bobier and Wirth [8]. But their work only used binary images and no further retrieval algorithms were employed.

In this paper, we use the Hierarchical Temporal Memory implemented using the Vision Framework of the Numenta Intelligent Platform for Computing (NuPIC) [1],[3],[4]. The HTM network is trained on 10 categories of images of the 384x256 RGB Corel 1000 image database [10]. Very positive results have been obtained with training only on 3 images per category with highly accurate classification on the rest 970 test images. The 970 test images simulate the images which would be added to an image database of 30 initial images. Since an image database in real world applications will likely have more images added into it, this robust classification shows that the image database can be expanded while minimizing the risk of misclassification. A color histogram of each image in the database is also maintained for the actual retrieval after the initial classification of the query image.

This approach might be unique in the way that we use the HTM classifier to drastically reduce the input for the actual retrieval process. This is shown to improve the efficiency and effectiveness of image retrieval using the color histogram method.

II. HIERARCHICAL TEMPORAL MEMORY

The Hierarchical Temporal Memory (HTM) is an algorithm which tries to capture the data modeling and processing capabilities of the human neocortex [2]. HTM is similar to Bayesian networks which use belief propagation, but they are self-training and are easier to handle. The algorithm essentially uses clustering mechanisms to achieve invariance in output when an input belonging to a particular class is presented to the network. It does this by forming a spatial temporal correlation between low level input patterns which appear to the network. Thus knowledge and

understanding about the HTM environment is only gained with what the HTM perceives as input.

HTMs in general are a tree structured multi-leveled hierarchy with each level consisting of a region of nodes. A typical 3 level HTM is shown in Fig. 1. An HTM can consist of any number of levels, but for most applications a 2 or 3 level node network suffices. Each level consists of a fixed number of nodes all of which perform the same algorithm. The bottom most level of the HTM is fed with the raw input data, which in this case is the output of a Gabor filter fed with a RGB color image. Each node performs clustering in overall three dimensions and it does this in two stages. The first stage is called the spatial pooler and the second one is the temporal pooler.

As the name suggests the spatial pooler pools or clusters data in the spatial dimension. Each pattern appearing at the input during learning of the spatial pooler is compared with the database of other patterns, if the distance between the input pattern and each is less than the maxDistance [3] parameter, then the input pattern is considered same as the corresponding existing pattern, termed as a coincidence. If the previous condition does not satisfy, then the input pattern is "memorized" as a new coincidence. Thus the spatial pooler quantizes the input space but only remembers the patterns which appear.

The temporal pooler performs clustering over time and forms temporal groups of coincidence patterns. These groups are formed on the basis of the statistical behavior of the input data, which is captured using a Markov graph whose nodes are the coincidence patterns learned previously. Hence, the members of a temporal group are likely to follow one another. After training, a vector of probabilities of membership of the input pattern to each of the temporal groups is the input to the next level of nodes. Therefore, the overall effect of this approach causes the lower level nodes to remember and recognize patterns of lower complexities such as a line or corner. As we ascend the hierarchy, we find that the coincidences represent combinations of patterns of lower complexities. This increases the variance and complexity of data represented at higher levels. But in spite of the seemingly large input space at higher levels, the spatial pooler at higher levels only remembers patterns it encounters thereby improving efficiency.

HTM levels can individually run in two modes, the learning mode and the inference mode. In the learning mode, a level tries to find new coincidences and keeps updating the Markov graph time progresses. In inference mode, the probability distribution of the membership of the input pattern is outputted to the next higher level. The learning mode provides no such output. During training of a particular level, all levels below it are run in inference mode and it itself is run in the learning mode.

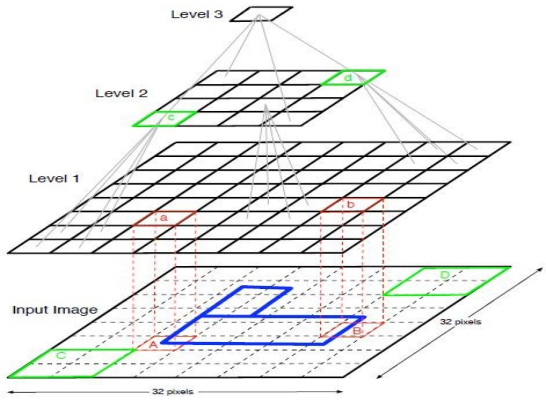

Figure 1: HTM with 3 levels

For intelligent recognition, a system must be able to identify a known pattern from within novel inputs. We have seen how the HTM works to manage huge input spaces and has the ability to cluster various complex patterns under a single category. This property of the HTM allows it achieve very good classification accuracy, since various images of the same object with different orientations, noise, brightness, etc., all get classified under the same object. We must understand that this important characteristic of invariance of class at higher levels given a changing noisy pattern of the same class at lower level is achieved by clustering or quantizing and mapping the input space which results in robustness.

HTM is thus very apt for applications such as image classification, since it quite successfully captures the semantic of visual data. The HTM doesn't "understand" the image as we do on a much higher level i.e. it doesn't have any other knowledge about the object other than the very high level features it contains. But even so, the knowledge of membership such high level features in each object is enough for accurate classification. Our approach not only uses the trained HTM for class identification of the new query image, it also is used to maintain the integrity of the database while adding new images into it. Also we present a method to use the HTM classifier to increase retrieval performance as explained in the next section.

III. Our Approach to CBIR using HTM

In this section, the structure and functioning of the overall CBIR system is explained. The HTM used was implemented using the Vision Framework of the Numenta Platform for Intelligent Computing. The CBIR system developed accepts query only by image, since such an interface was found to have the capability to encode the subtle semantics of the query as opposed to a query by text. Both the structure and functioning of our CBIR system can be considered to be broadly divided into two parts. We first

explain the structure of the system followed by the functioning.

A. Structure

Our CBIR system has two parts, first of which is the HTM and the other is the stage post classification which retrieves the final images using image processing techniques more specifically the color histogram similarity. The HTM consisted of 5 levels. The first level consists of a Gabor filter with a 384x256 receptive field which receives the input image from the image sensor provided with the Numenta Vision Framework. The Gabor filter can be said to work as a spatial pooler since it too quantizes the input space and reduces it. Its output is sent to a 29x29 temporal pooler region. The spatial pooler and temporal pooler are considered to be different levels in the convention of the Vision Framework. The output of the temporal pooler is sent again to a 26x26 spatial pooler region followed by a 16x16 temporal pooler region up the hierarchy. This completes 4 levels of the hierarchy. But effectively the hierarchy forms only a 2 level HTM network. The top level is the classifier node which outputs a probability distribution of membership of the image in each of the categories.

The second part of our CBIR consists of an image processing stage after the query image has been assigned a known class. In that stage, a 3 dimensional color histogram of the RGB image is computed. This histogram is prepared for each image during training and is done by counting the number of pixels whose value falls within a certain range. The 256 possible intensity levels are split into 16 bins and this is done for each of the three colors. Thus for each image a 3x16 matrix is computed. The image database is always kept organized according to category along with the histogram matrix even when adding new images with the help of the trained HTM. Thus, for each entry in the database, we have a classified image and its corresponding color histogram matrix computed as mentioned.

B. Functioning

The CBIR system after training consists as mentioned earlier, of a trained HTM and a database complete with the images and their histogram matrices. The query image is presented to the image sensor of the HTM which is run in inference mode. The output of the HTM gives us a probability distribution over the various categories the HTM has been trained on. The difference between the maximum probability value and the other values gives us an idea of the confidence level of the HTM. This completes the first stage of the retrieval process.

In the second stage, the color histogram of the query image is computed and it is then compared with the histogram of the images of the same category as the query image. The comparison is done using a similarity measure [7] which utilizes the Euclidean distance between the two histograms and is defined as

$$S_c^{ED}(I,Q) = 1.0 - \sqrt{\frac{\sum_r (I_R(r) - Q_R(r))^2 + \sum_g (I_G(g) - Q_G(g))^2 + \sum_b (I_B(b) - Q_B(b))^2}{2*3}}$$

where I is the database image, Q is the query image, I_R, I_G, I_B, Q_R, Q_G, Q_B are the normalized color histograms and $S_c^{ED}(I,Q)$ is the measure of similarity between I and Q.
This process will create a system in which the images in the winning category are ranked according to the distance between the query image's histogram and the same of the corresponding image in the database. A shorter distance will denote a stronger match. This also enables us to retrieve any desired number of images according to this ranked sequence. The image retrieval process is thus non-iterative.

The CBIR system we developed utilizes the 1000 image database as the complete set. A query image may be chosen external to the database but the images returned will only be from the database. The system is robust to intra category variation of images i.e. if we have a category of elephants, then the system would show invariance to the size, number, pose, color and orientation of the elephants. More focus was given to this kind of robustness since HTM can be easily trained and tweaked to still give good results in case noise, blur, occlusion etc. as can be seen in cases of the previous work of Bobier and Wirth [8].

IV. Experimental Details and Results

All experiments were conducted as mentioned earlier on the 384x256 Corel 1000 image database having 10 categories of 100 images each [10]. The CBIR system we designed uses the HTM to initially classify the query image and uses the category information to run the color histogram technique on the 100 images of the category. The query images were chosen at random from the database and the first result for the final image retrieval is always omitted since it is always the query image itself. The CBIR system can be effectively split into two separate sub-systems which can be independently studied, the first is the trained HTM which solves the image classification problem and the second is the image processing stage which uses the color histogram technique for the final image retrieval. The combination of these two results gives us the performance of the entire CBIR system.

A. Image Classification

Major parameters of the HTM in the Numenta Vision Framework are shown in Table I. Multi sweep was the training algorithm chosen for the level 2 and level 4 temporal pooler nodes, whereas the default training algorithm was found to give good results for the level 3 spatial pooler nodes.

TABLE I. MAJOR PARAMETER CHANGES FOR THE HTM USING NUMENTA VISION FRAMEWORK

HTM Parameters	Values
numCategories	10
seed	24
midLevelPatches	160
gaborNumOrients	(is varied)
gaborPhaseMode	'single'
gaborCenterSurroud	False
spatialPoolerAlgorithm	kthroot_product
maxDistance	0.3
temporalPoolerAlgorithm	maxProp
spatialPoolerTraining Algorithm	RandomFlash
temporalPoolerTrainingAlgorithm	MultiSweep

Table II shows the accuracies obtained with increasing the number of training images per category, whereas Table III shows the accuracies and other features when a key parameter of the Gabor filter the gaborNumOrients which controls the number of orientations of the Gabor filter was varied. We find that upon increasing the number of images the accuracy doesn't increase much. In fact on a few occasions increasing the number of training images was seen to slightly reduce accuracy. But this is simply because the statistics of the data changed with changing the number of testing images. We must understand that the performance of the HTM remained the same because the number of coincidences observed and groups found were constant. The parameter which controls the number of groups formed in the level 2 temporal pooler nodes was the gaborNumOrients which was then varied to show the importance of number of groups allowed to be formed. More groups allow the HTM to learn and classify more varied coincidence patterns which has a direct effect on recognition performance. Hence we can infer that the number of learnt coincidences is directly related to the amount of knowledge stored in the HTM. This in the end enables us to drastically reduce the number of training images required for good accuracy, and this fact is a big advantage in scalability. The system when scaled up to larger datasets with many more categories would require only relatively small increase in training images.

Also, the confidence level of the HTM in most of the classification was quite high. This also might indicate that the image has components of only a single category. If for example, an image was presented which had components of 2 or more distinct categories, then the HTM would classify it in the class with the highest component participation but with lower confidence. This allows for using techniques for retrieval post classification which utilize high membership values for multiple classes. Thus more specific retrieval would be possible.

B. Image Retrieval

When a query in the form of an image is submitted to the system, the HTM classifies it and this information about its category is utilized by the second stage. This stage has to compute the similarity measure between the query image and each of the 99 other images in the category. Since the color histogram was computed for each image in the database during training, the ranking of the images becomes a sorting problem. The sorting takes place till only the requested number of images is obtained. The system was tested with a variety of queries from the database and was found to return satisfactory results. A number of images within each of the categories were queried during the test and the most of the queries did return very similar images. A few of the queries are illustrated below.

2(a) 2(b) 2(c) 2(d) 2(e) 2(f)

Figure 2(a): Query Image, Figures 2(b) to 2(f): Retrieved Images


3 (a)


3 (b)


3(c)


3(d)


3(e)


3(f)

Figure 3(a): Query Image, Figures 3(b) to 3(f): Retrieved Images

V. Conclusion and Future Work

This paper describes the implementation of a Content Based Image Retrieval system using the Hierarchical Temporal Memory algorithm. The system was found to give very promising results overall. The use of HTM to reduce input space for the color histogram technique provides much better results than without using it. Other techniques could be used to make the actual retrieval process more refined.

The HTM algorithm demonstrated very robust classification with very little training. The neurologically inspired computing method is very appropriate for applications in which a human currently exceeds machine capabilities. The image classification problem was a subjective problem requiring a system to "understand" the semantics within each image. HTMs were observed to model visual data very effectively in this implementation.

This experiment also shows the potential of HTMs to far exceed human capabilities since attaining enough knowledge to achieve 99.8 percent accuracy on 970 images with only 30 images for training requires deep understanding of the mechanics of human learning and when implemented in hardware, the system would have a huge performance boost. One might argue that a human with even fewer training examples could correctly classify an even larger dataset, but we must remember that the HTM had no knowledge about the world prior to training unlike a human who has been using and developing visual recognition systems since birth. But this difference is only because a biological system is much slower than conventional hardware.

The CBIR system can be scaled up to larger databases if the image processing techniques are optimized which opens up further research options. Also other distance measures might be used which might improve performance by incorporating more low level features such as shape for final retrieval post classification. A drawback of our implementation is that if the query image is misclassified, then the retrieved images will have absolutely no high level similarity. But this can be avoided by keeping the classification accuracy of the HTM high while adding new images to the database. Numenta is working on newer algorithms for HTMs which can be hoped to improve performance on many accounts once they are released. This provides yet another opportunity for research.

Another fact that makes HTMs very useful for CBIR systems is that it outputs the membership probabilities of the query image to each class. Our approach currently used only the highest value, but the idea of using the other values for more specific and refined retrieval opens up yet more research opportunities.

The implementation of the CBIR system is simple yet very effective and can easily be expanded to include more categories. The distance between the semantics of two categories can also be reduced with proper tweaking of the HTM classifier. The overall system if hardware implemented will have a significant performance boost and will be suitable for very large scale applications such as on the internet image database. But further research might improve retrieval performance if techniques are developed specifically to retrieve similar images from within a single category.

References

[1] D. George and B. Jaros, "The HTM Learning Algorithms", Numenta Incorprated, March 1, 2007. [Online]. Available: http://www.numenta.com/htm-overview/education.php

[2] J. Hawkins and D. George, "Hierarchical Temporal Memory: Concepts, Theory, and Terminology", Numenta Incorprated , March 27, 2007. [Online]. Available: http://www.numenta.com/htm-overview/education.php

[3] "VisionFrameworkGuide", Numenta Incorprated, August 2009.

[4] "nupic_gettingstarted", Numenta Incorprated, September 2008.

[5] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos, 2000: "The Bayesian image retrieval system, PicHunter: theory, implementation and psychophysical experiments." IEEE Transactions on Image Processing, 9, 20–37.

[6] A. Frome, Y. Singer, and J. Malik. "Image retrieval and classi-fication using local distance functions." In Advances in Neural Information Processing Systems (NIPS), volume 19, pages417–424, 2007.

[7] A. K. Jain, A. Vailaya :"Image retrieval using color and shape." Pattern Recognition 29(8): 1233-1244 (1996)

[8] B. A. Bobier , M. Wirth, "Content-based image retrieval using hierarchical temporal memory", Proceeding of the 16th ACM international conference on Multimedia, October 26-31, 2008, Vancouver, British Columbia, Canada [doi>10.1145/1459359.1459523]

[9] J. Thornton, T. Gustafsson, M. Blumenstein, T. Hine (2006) "Robust character recognition using a hierarchical bayesian network." AI 2006: Advances in Articial Intelligence : 12591264.

[10] Corel 1000 image database [Online]. Available: http://wang.ist.psu.edu/docs/related/

TABLE II. VARIATION OF ACCURACY AND COINCIDENCE COUNT WITH NUMBER OF TRAINING IMAGES PER CATEGORY KEEPING NUMBER OF ORIENTATIONS OF THE GABOR FILTER = 2

Number of training images per category	**Accuracy in %**	**Level 2 coincidences (temporal)**	**Level 2 groups (temporal)**	**Level 3 coincidences (spatial)**	**Level 4 coincidences (temporal)**	**Level 4 groups (temporal)**
3	98.6	936	2	160	640	160
6	98.9	936	2	160	640	160
9	99	936	2	160	640	160
12	97.3	936	2	160	640	160
15	98.1	936	2	160	640	160
20	98.4	936	2	160	640	160

TABLE III. VARIATION OF ACCURACY AND COINCIDENCE COUNT WITH NUMBER OF ORIENTATIONS OF GABOR FILTER KEEPING NUMBER OF TRAINING IMAGES PER CATEOGRY = 3

gaborNumOrients	**Accuracy in %**	**Level 2 coincidences (temporal)**	**Level 2 groups (temporal)**	**Level 3 coincidences (spatial)**	**Level 4 coincidences (temporal)**	**Level 4 groups (temporal)**
2	98.6	936	2	160	640	160
5	99.6	2340	5	160	637	93
10	99.2	4680	10	160	636	160
15	99.8	7018	15	160	639	149
20	99.9	9358	20	160	638	155