A Cortex-like Model for Rapid Object Recognition Using Feature-Selective Hashing

Yu-Ju Lee, Chuan-Yung Tsai, and Liang-Gee Chen
Graduate Institute of Electronics Engineering and Department of Electrical Engineering,
National Taiwan University, Taipei, Taiwan

Abstract— **Building models by mimicking the structures and functions of visual cortex has always been a major approach to implement a human-like intelligent visual system. Several feedforward hierarchical models have been proposed and perform well on invariant feature extraction. However, less attention has been given to the biologically plausible feature matching model which mimics higher levels of the ventral stream. In this work, with the inspirations from both neuroscience and computer science, we propose a framework for rapid object recognition and present the feature-selective hashing scheme to model the memory association in inferior temporal cortex. The experimental results on 1000-class ALOI dataset demonstrate its efficiency and scalability of learning on feature matching. We also discuss the biological plausibility of our framework and present a bio-plausible network mapping of the feature-selective hashing scheme.**

I. Introduction

CAPABILITY of the primate visual system outperforms the best computer vision systems in every aspect such as speed, generalization ability, scalability and plasticity. Therefore, in order to achieve the ultimate goal of developing a human-like visual machine, building a cortex-like model by mimicking the processing flow and network structure of visual cortex has always been a major approach. Several cortex-inspired models have been proposed and applied successfully to high-level visual tasks, like face, object, action recognition and localization [1]-[5]. The core of these models is to learn invariant and discriminative features and describe the complex and variant visual inputs with compact representations. Although there have been several works about using cortex-like models to get the invariant feature representations, which performs well, less attention has been given to a biologically plausible and efficient feature matching model which mimics the memory association in higher levels of the ventral stream. In previous works, the most common ways to do feature matching for classification are nearest-neighbors (NN) search [5], support vector machine (SVM) [1] and artificial neural network [3]. All of them suffer from long matching or training time when handling large database. However, recent read-out experiments have shown that the time course in monkey inferior temporal (IT) cortex, which is considered to be handling object categorization, can be just as short as 12.5 milliseconds [6]. This implies primates and humans can read out the object identity in an extremely effective way.

Learning and memorizing objects via categorization is a common scheme for avoiding exhaustive feature matching. Within computer science, in order to quickly find the nearest neighbors in a large database, one can categorize the data by building trees [7][8] or hash tables [9]-[11] and then search fewer candidates only, instead of exhaustive search. These methods provide theoretical guarantee on the search quality and efficiency with some proper constraints. Similarly, within neuroscience, it has been found that the neural activity in IT cortex shows its object selectivity in alignment with cortical columnar organization, which means groups of neurons with similar response properties for specific objects are clustered together in localized cortical regions [12]. Another study about thalamocortical regions suggests that stored memories are organized into similarity-based hierarchies via hash-like storage, which is sparse and enables large amounts of data to be stored in a compact space [13]. We believe that such mechanisms are the keys for the rapid feature matching in primate visual system.

In this paper, we present a visual recognition framework based on the feedforward hierarchical model with the proposed scheme of memory association – *feature-selective hashing* (FSH), which provides the matching efficiency, scalability and plasticity of learning at the same time. We demonstrate its capability by testing it on multi-class object recognition. Furthermore, we compare the performance between utilizing local and holistic features for the FSH.

The remainder of this paper is organized as follows. In Section II, we briefly introduce the background knowledge of the feedforward hierarchical model and the hashing scheme on which our work is based. Section III presents the proposed framework and shows some experimental results. Discussions about the biological plausibility are given in Section IV, and Section V concludes this paper.

II. Background

In this section, we briefly introduce the concept and processing flow of two previous works, which are the bases of the proposed framework. The first one is a feedforward hierarchical model called HMAX, which extracts invariant features and provides good object selectivity. The second one is locality-sensitive hashing (LSH), which aims to index data points into codes by pre-defined LSH functions and construct the hash tables for looking up NN efficiently.

A. HMAX Model

HMAX is a hierarchical processing model for visual feature extraction. It was first proposed by Riesenhuber and

Poggio [14], extended by Serre [1], and further improved by Mutch [15]. HMAX mimics the hierarchical structure and tuning properties of visual cortex in the feedforward path of ventral stream, which starts from V1 (primary visual cortex), through V2, and V4 to IT (inferior temporal cortex). The studies of visual cortical hierarchy showed that V1 is tuned for simple features like oriented lines [16], V4 for features of intermediate complexity like geometric shapes [17], and IT for complex object features like faces [18]. It implies that, through the hierarchy, neurons at lower level detect the low-level features and fire to the next level, while neurons at higher level receive and combine the stimulus from lower level to detect more complex features. Moreover, the invariance capability of detected features as well as receptive fields increases along the hierarchy from bottom to top.

The two computation units in HMAX, simple and complex units, are defined according to the tuning properties of simple and complex cells found in V1 [16]. The simple units are designed for responding to certain input patterns at specific position in the receptive fields. Thus, they receive the afferent inputs and match them with learned prototypes to detect corresponding features. For example, in Serre's HMAX implementation, Gabor filters [19] are used at bottom-level as prototypes, while at higher level, prototypes are learned by randomly sampling patches from training images. The complex units are designed for pooling the stimulus from afferent simple units and providing the scale and location invariance. In HMAX, they are generally modeled by max operations in the corresponding receptive fields.

Through interleaving the simple and complex units hierarchically, one can build the HMAX model for extracting the invariant features of an input image into a feature vector. Each component of a feature vector represents the strength of response to the corresponding complex patterns learned at higher level. Our framework is based on Mutch's improved HMAX model – FHLib [15], which stacks two levels of simple and complex cells to construct five-layer hierarchy: Image-S1-C1-S2-C2.

B. Locality-Sensitive Hashing

Locality-sensitive hashing is an indexing scheme proposed by Indyk et al. [9]. Through hashing data points into corresponding buckets, it can categorize large amount of data into several subsets based on their locality. It is very useful to reduce the query time of finding NN, because one can only search the data collided with query in the same bucket instead of exhaustive search. However the trade-off is one may not find the exact NN (said r^*). Fortunately, using LSH functions, it can be guaranteed that distance of the returned approximate NN (said r) to the query q satisfies: $d(r,q) = (1+\varepsilon)\cdot d(r^*,q)$, with a small error $\varepsilon > 0$. Therefore, utilizing LSH for Approximate Nearest Neighbor (ANN) search can both ease the curse of dimensionality and improve the query time from $O(dn)$ to $O(dn^{1/(1+\varepsilon)})$, where d is dimension and n is number of data points. The improvement becomes significant especially for large-scale datasets with high dimensionality.

The essence of LSH is, through choosing the proper hash functions, hashing close points in feature space into same buckets of hash tables with high probability, and distant points into different buckets. More precisely, the hash functions $h(\cdot)$ from the locality-sensitive family $\mathcal{H}$ is defined such that for any points $\mathbf{p}$ and $\mathbf{q}$ in $\mathbb{R}^D$, the probability of collision in the same bucket satisfy the following conditions:

$$\begin{cases} P[h(\mathbf{p})=h(\mathbf{q})] \geq P_1, & \text{if } d(\mathbf{p},\mathbf{q}) \leq R; \\ P[h(\mathbf{p})=h(\mathbf{q})] \geq P_2, & \text{if } d(\mathbf{p},\mathbf{q}) \geq cR; \end{cases} \tag{1}$$

where $P_1 > P_2$ and $c > 0$. A generic LSH function is defined as follows. Given a data point $\mathbf{x}$, we first project $\mathbf{x}$ onto a 1D vector $\mathbf{w}$, then set the threshold as b and the quantization step as s. The resulting hash function is given by

$$h(\mathbf{x}) = \left\lfloor \frac{\langle \mathbf{x}|\mathbf{w}\rangle + b}{s} \right\rfloor \tag{2}$$

where $\langle\cdot|\cdot\rangle$ is the projection operation and $\lfloor\cdot\rfloor$ is the floor operation. To increase P_1/P_2, it is general to construct L hash tables by concatenating K different hash functions h as its code. The formulation of code H for the l^{th} hash table is $H_l = [h_{1l}, h_{2l}, \cdots, h_{Kl}]$, where $l = 1,\ldots,L$. The returned NN candidates of query are the union of collided data from all hash tables, so using more hash tables (larger L) causes more returned candidates, which can improve accuracy but also extend the query time. Usually, L should be large enough to guarantee $P_1 \to 1$. K is another factor for controlling the trade-off between accuracy and query time because larger K causes more buckets per hash table, which can lower P_2 effectively.

In order to get compact and efficient codes, many works were proposed to design the hash functions heuristically by defining the projecting vector $\mathbf{w}$ and the method of scalar projection $\langle\cdot|\cdot\rangle$ [20]. In our approach we adopt the simplest hash function by random projection using dot-product. The random vector $\mathbf{w}$ is usually constructed by sampling each component randomly from a normal distribution. For simplicity, we set $\mathbf{w}$ as a standard vector $\mathbf{e}_d$ (i.e. a vector with d^{th} component equal to one and other components equal to zero), where d is randomly chosen from 1 to D. Furthermore, we binarize the returned value of hash functions as

$$h(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{w}^T\mathbf{x} + b \geq 0; \\ 0, & \text{otherwise}; \end{cases} \tag{3}$$

which is also a typical response modeled in many neural networks. Thus, H_j can be represented as a K-bit binary code.

III. Proposed Framework using Feature-Selective Hashing

The proposed framework builds on the FHLib and attempts to mimic the property of object selective organization in IT by incorporating the feature-selective hashing scheme. We test this approach with objects recognition tasks and the

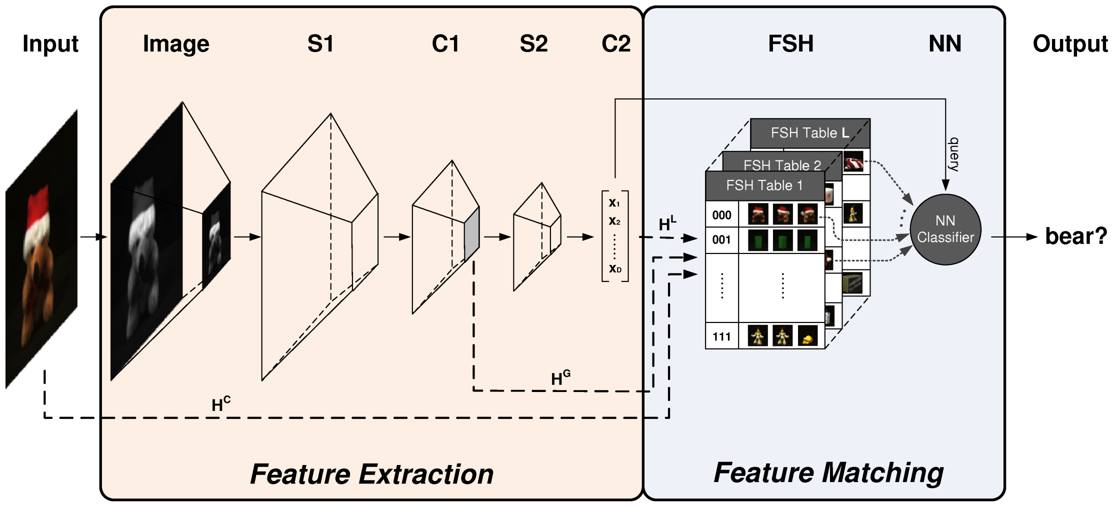


Fig. 1. Overview of the proposed framework. The framework consists of two stages: feature extraction and feature matching. In the stage of feature extraction, features are computed in 5-layer hierarchy: Image-S1-C1-S2-C2 [15]. Through this processing, input image is represented as a D-dimensional vector and then fed into NN classifier as query. In the stage of feature matching, first we use one of the features (local features, gist-like features and color features) as the input of hash functions H, and lookup the similar object candidates in L hash tables. Then, these candidates are all fed into NN classifier and matched with the query. Last, we output the category label of returned NN as answer.

results show improvements on matching speed and learning scalability, while maintaining the classification accuracy. We first give an overview of our framework, then describe the implementation details of FSH with local and holistic features respectively, and finally evaluate their performance in object recognition tasks.

A. Framework

The overall framework (shown in Fig. 1) can be divided into two stages: feature extraction and feature matching. In the stage of feature extraction, images are reduced to feature vectors, which are computed through the five-layer hierarchy of FHLib described briefly as follows. Image layer is an image pyramid with multi-scale resolution. S1 layer is the response of matching different orientation Gabor filters with the image layer. C1 layer is the subsample of S1 layer using local max operations across scale and position. S2 layer is the response of matching pre-sampled C1 local patches. C2 layer is computed by global max operation, which reduces the set of S2 responses into a D-dimensional vector and each component is the maximum response to one of the D pre-sampled C1 patches.

In the stage of feature matching for classification, we use the C2 feature vectors generated by FHLib as query to find NN in the database. Instead of exhaustive NN search, we search the candidates returned by FSH tables only. Thus, this stage can be further divided into two parts: FSH and NN search. The first part is to find the candidates with similar features to test images' via FSH. *The difference between FSH and LSH lies in the input of hash functions*. For LSH, it directly uses the query for NN search as input of hash functions, which aims to preserve the locality of query into hash codes. While in FSH, we can choose different features other than query as input, which provides a flexible and biologically inspired way to hash by using other features like color, texture, shape or context. Here we use two kinds of features, local and holistic features, as input of FSH functions. The second part is using the C2 vectors as query to do the NN search among the returned candidates. The categories of returned NN are the answers of classification. In the next two subsections, we mainly focus on the two different implementations of FSH.

B. Local Feature-Selective Hashing

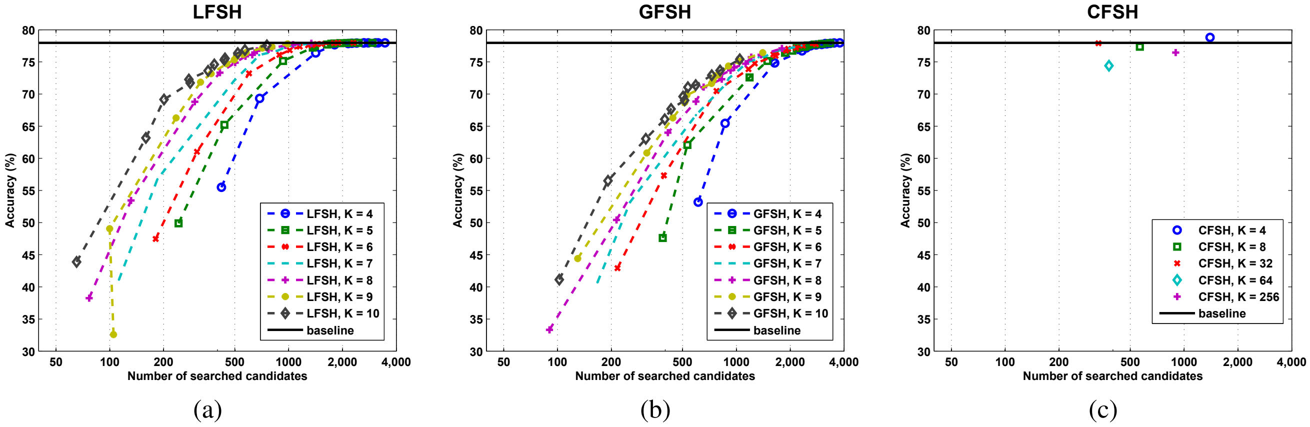
The main idea of local feature-selective hashing (LFSH) is to use the similarity of local features to define the LSH functions, where the local features are the components of C2 vectors outputted by FHLib. The formulation of LFSH is defined as follows. Given a training dataset containing n C2 feature vectors, $\mathcal{X} = \{\mathbf{x}_i\}$, $i = 1, \ldots, n$ and $\mathbf{x}_i \in \mathbb{R}^D$, we aim to define the LFSH functions h via random projection described in equation (3). Since we set $\mathbf{w}$ as the standard vector $\mathbf{e}_d$, the dot-product operation $\mathbf{w}^T\mathbf{x}_i$ can be reduced to the d^{th} component of $\mathbf{x}_i$ denoted by x_{id}. Thus, to construct a K-bit code of L LFSH tables, the k^{th} hash function is defined as

$$h_k^L(\mathbf{x}_i) = \begin{cases} 1, & \text{if } x_{id_k} + b_k \geq 0; \\ 0, & \text{otherwise}; \end{cases} \tag{4}$$

where $1 \leq d_k \leq D$ and b_k can be just set to zero because each dimension of C2 vectors is normalized to zero mean and unit variance in FHLib.

C. Holistic Feature-Selective Hashing

Similarly, holistic feature selective hashing (HFSH) aims to use the similarity of holistic features to define the hash functions. Here we adopt the color and gist information, which are known as two major categories of pre-attentive



(a) (b) (c)

Fig. 2. The classification accuracy on variant viewpoints test set for different number of searched candidates (controlled by different L) with three kinds of FSH: (a) LFSH (b) GFSH (c) CFSH. The baseline represents the accuracy of exhaustive search. For each curve in (a) and (b), data points from left to right are obtained using $L = 1, 2, 5, 8, 10, 12, 14, 16, 18, 20, 25, 30$ and 50, respectively.

features of primates, as the holistic features. The studies of pre-attentive processing have shown that the basic features or coarse gist information detected from the early stage of ventral pathway can carve the complex visual input into candidates rapidly and also guide the attention to help objects detection [21]. A successful work for object and scene recognition and detection using the holistic features is gist descriptor proposed by Oliva and Torralba [22]. The gist descriptor describe the whole image as a vector without detecting any interest point, and it performs well even for the low-resolution images [23]. Inspired by these, we propose two kinds of HFSH described as follows.

For gist-like HFSH (GFSH), based on the concept of gist, we use the low-level features extracted from the lowest scale in C1 layer of FHLib. More precisely, given the lowest scale in C1 layer containing M-by-O responses, where M is the number of locations and O is the number of Gabor filter orientations, we can concatenate them to get a gist-like feature vector of size $M \cdot O$ as the input of hash functions. The formulation is similar to LFSH. Given the training set containing n gist-like feature vectors $\mathcal{G} = \{\mathbf{g}_i\}$, $i = 1, \ldots, n$ and $\mathbf{g}_i \in \mathbb{R}^{M \cdot O}$, we define the k^{th} GFSH function as follows:

$$h_k^G(\mathbf{g}_i) = \begin{cases} 1, & \text{if } g_{id_k} + b_k \geq 0; \\ 0, & \text{otherwise}; \end{cases} \tag{5}$$

where $1 \leq d_k \leq M \cdot O$ and b_k can be just set to zero as well because each component of gist-like features is normalized.

For color-based HFSH (CFSH), inspired from the response properties of bipolar cells, which have the preference for either red-green or blue-yellow opponency [24], we accumulate the color histogram of input images on the opponent color space:

$$\begin{pmatrix} O1 \\ O2 \\ O3 \end{pmatrix} = \begin{pmatrix} (R-G)/\sqrt{2} \\ (R+G-2B)/\sqrt{6} \\ (R+G+B)/\sqrt{3} \end{pmatrix} \tag{6}$$

where the intensity is represented in $O3$ and the color information is in $O1$ and $O2$. To construct the opponent histogram, we divide the opponent color space into S bins and count the number of pixels for each bin to get a color histogram of size S as input of hash functions. In addition, for simplicity, we only construct one hash table instead of L and let $K = S$. Even when adopting such simplification, this method still performs well in retrieving objects with similar color to query's. For detailed formulation, given the training set containing n color histograms $\mathcal{C} = \{\mathbf{c}_i\}$, $i = 1, \ldots, n$ and $\mathbf{c}_i \in \mathbb{R}^S$, we define the k^{th} CFSH function as following equation:

$$h_k^C(\mathbf{c}_i) = \begin{cases} 1, & \text{if } c_{ik} + b_k \geq 0; \\ 0, & \text{otherwise}; \end{cases} \tag{7}$$

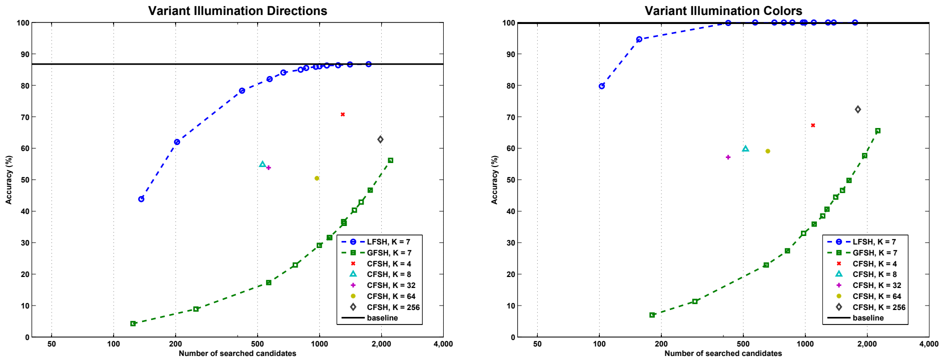
where b_k is heuristically set to 100 in the following experiments.

D. Experiments

To evaluate the performance of the proposed framework, we test it on the Amsterdam Library of Object Images (ALOI) dataset [25]. The ALOI dataset consists of 110,250 images comprising 1,000 different object categories. Each object category is collected under different viewpoints, illumination colors and illumination directions. It is suitable for evaluating the efficiency and scalability of FSH because ALOI contains a large number of objects with variations.

In our setting of FHLib model, we follow the parameter settings described in [15]. For training, we choose four viewpoints $(0°, 90°, 180°, 270°)$ in each category as training images and randomly sample one C1 patch per image, while the remaining images compose the testing set. Thus, training a full ALOI dataset using FHLib will sample 4×1000 C1 patches and form a 4000-dimension C2 vector per image.

Firstly, to test the efficiency of FSH, we train our model on the full ALOI dataset with different settings of L and K, which results in the trade-off between searching range and classification accuracy, and test it on different viewpoint images. The results are averaged over 5 runs and shown in Fig. 2. From this result, FSH shows its efficiency on reducing the searching range while maintaining the accuracy. In Fig.



(a) (b)

Fig. 3. The classification accuracy with three kinds of FSH for different number of searched candidates on two variant illumination test sets: (a) variant illumination colors test set (b) variant illumination directions test set. The baseline represents the accuracy of exhaustive search.

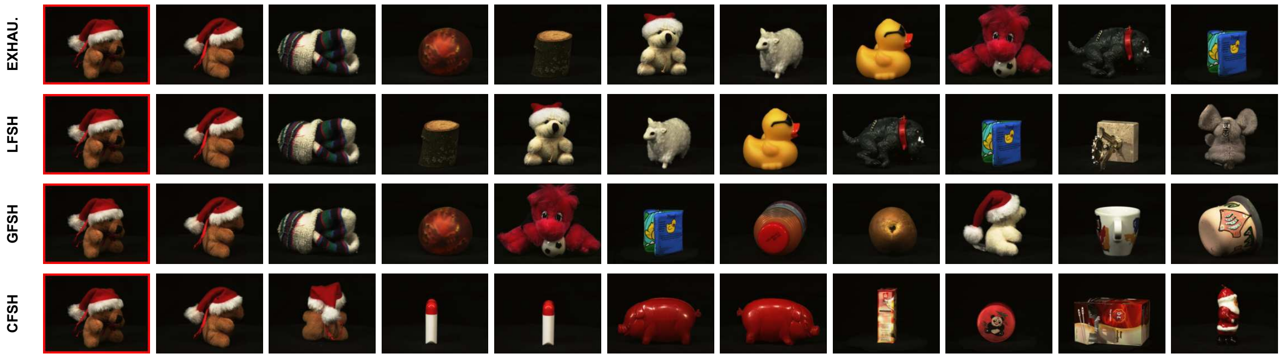
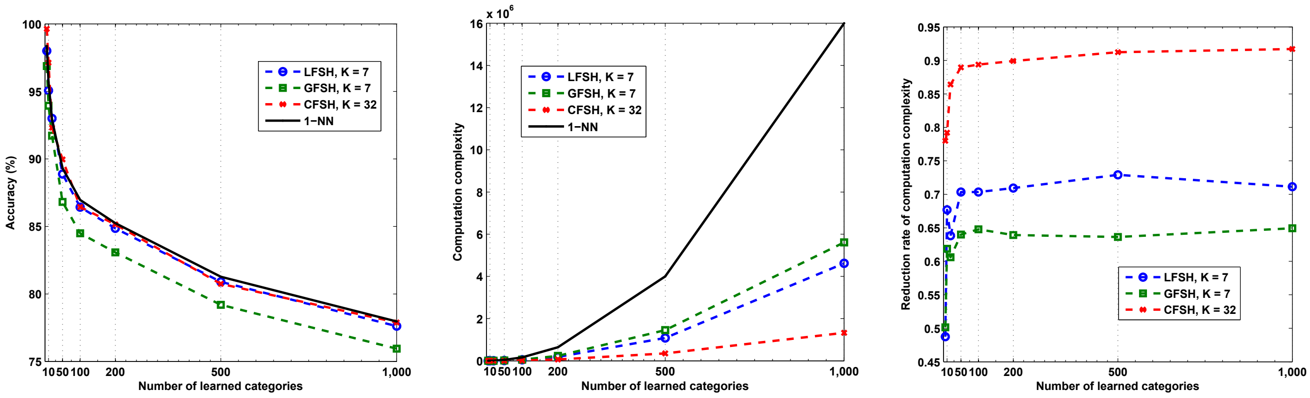


Fig. 4. The top 10 nearest neighbors returned by exhaustive search and three kinds of FSH. The left most image with red bounding box is the query and its nearest neighbors are ordered from left to right.



(a) (b) (c)

Fig. 5. Experimental results with three kinds of FSH for different number of learned categories: (a) The classification accuracy (b) The computation complexity (c) The reduction rate of computation complexity.

2 (a)(b), the efficiency of LFSH and GFSH increases with K, but at the same time we need larger L to provide enough accuracy. Thus, it shows the trade-off between performance and memory requirement. Among these three approaches, CFSH has superior performance over others even using only one hash table because the test set has large color inter-variation and low color intra-variation. However, if we test it on the different illumination color and direction cases (shown in Fig. 3), CFSH shows its weakness for color variations. These results indicate that utilizing color features is rather effective in simple cases, while local features can provide stable performance in general cases but hardly achieve superior performance. On the contrary, GFSH performs worse than others in every test case because gist-like features are less invariant to viewpoint and illumination variation. As a result, how to make CFSH and GFSH more tolerant to color and shape variations is an important issue of our future work. To visualize the quality of FSH, we show top 10 neighbors

returned by different approaches with an example image, as shown in Fig. 4. CFSH tends to return objects with similar color and thus rule out the confusing candidates that are similar in shape or local features.

Secondly, we test the scalability of FSH by varying the number of training categories, and both the curve of accuracy and computation complexity are shown in Fig. 5 (a)(b). Here we choose $K = 7$, $L = 20$ for LFSH and GFSH, and $B = 32$ for CFSH. The score of computation complexity is computed by $D \times N^*$, where D is the length of C2 feature vectors and N^* is the number of searched candidates. These results show that FSH largely reduces the computation complexity with only a little drop in accuracy, and thus provides much better scalability, with an acceptable overhead of extra memory usage. In addition, the curve of complexity reduction rate is shown in Fig. 5 (c). The reduction rate can reach 90% but tends to saturate when learning more than 200 categories. It might be a problem when facing larger datasets, like 10,000 to 30,000 categories learned during a human's whole life. We suggest that adopting top-down feature selection scheme may be the key for further improving the reduction rate to fight against this problem.

IV. Discussions

In this paper, we have shown that using FSH to emulate the memory association in IT can not only improve the efficiency of feature matching but also provide the scalability of learning. Even though we adopt a relatively simple way to determine the hash function, FSH still performs well in our experiments. To further enhance the performance, one may replace the hashing scheme with other more complex state-of-the-art techniques like Spectral Hashing [10] or Semantic Hashing [11]. In our implementation, we utilize three features (local C2, gist-like and color-based features) to index the hash tables, but it is open to use other features or information like context, texture or prior knowledge. Another important issue of the future work is how to fuse these features properly in order to provide more efficient and accurate looking up. Interestingly, using hash to activate the relevant data and gate the irrelevant also provides the sparsity in memory association. It has been widely found that the sparsity constitutes a general principle of sensory coding in the nervous system [26]. Utilizing sparse coding in our brain provides several advantages like increasing the capacity of memory, making read-out easier and saving energy, and FSH also provide similar advantages since benefiting from the sparsity.

To categorize our memories in higher brain areas, tree-based method is another possible approach and also has been widely used in computer science. We didn't choose tree-based method in our framework due to its poorer biological plausibility. A tree-based memory structure grows when learning more data and increases its depth as well. Therefore, many decisions need to be made when descending the tree to find the answer. Furthermore, we usually have to trace back the route when descending the tree to verify the answer getting from the leaf node, which makes the number of decision makings even larger. We argue that it is unreasonable for too many IT neurons to fire sequentially (for making decisions) and then finding answers within such a small time interval of about 12.5ms. A study has showed that the firing rates of neurons are barely above 100Hz in the visual system [27], which implies a neuron probably at most fires once during the feedforward ventral stream (only as short as 80-100ms). On the contrary, hash-based structure is more biologically plausible because it provides more parallelism and is also in agreement with the cortical columnar organization in IT geometrically (see later paragraph). As a result, we choose the hash-based structure rather than tree-based.

Although we can investigate the brain functions through techniques like functional magnetic resonance imaging (fMRI), it is still uncertain how the visual cortex implements the mechanisms of memory organization, association and retrieval. Therefore, to provide an answer to this question, we propose a biologically plausible network mapping from our framework as shown in Fig. 6. In the feature extraction stage of the proposed framework, HMAX mimics the structure and tuning properties of visual cortex, and it corresponds to the visual cortex hierarchy from V1, V4 to anterior inferior temporal cortex (AIT). The C2 feature vectors which are outputted from HMAX represent the matching scores of learned prototypes, so each component can be mapped to V4/AIT neurons (called C2 units) tuned for complex features like shape. In the feature matching stage, we assume FSH tables for organizing object-level memory and providing efficient lookup operations are both functionally and structurally in agreement with the property of cortical columnar regions in IT. We map FSH tables to clusters of cortical columnar regions (called CC units), and each CC unit organizes the object-level memory into several CC regions as buckets. Similarly, only a few CC regions are activated by the stimulus from C2 units. Finally, the NN classifier responsible for similarity-based classification corresponds to neurons of prefrontal cortex (PFC), which function as Winner-Take-All (WTA) units to get the most possible object identity. It is conjectured that the derived network accounting for memory association are fundamental building blocks in the region of IT and PFC, and we will seek for more behavioral and electrophysiological evidences to support our model in the future.

V. Conclusions

In this paper, we have proposed a cortex-like framework consisting of HMAX and feature-selective hashing scheme to extract invariant features and then efficiently match features for object recognition tasks. The proposed hashing scheme utilizes the local features and holistic features to find fewer candidates which are similar to query. We test the proposed framework on 1000-class ALOI dataset and the results show that FSH provides good efficiency and scalability at the same time (best case is up to 90% computation complexity reduction). Finally, we discuss the biological plausibility of proposed FSH and provide its network mapping.

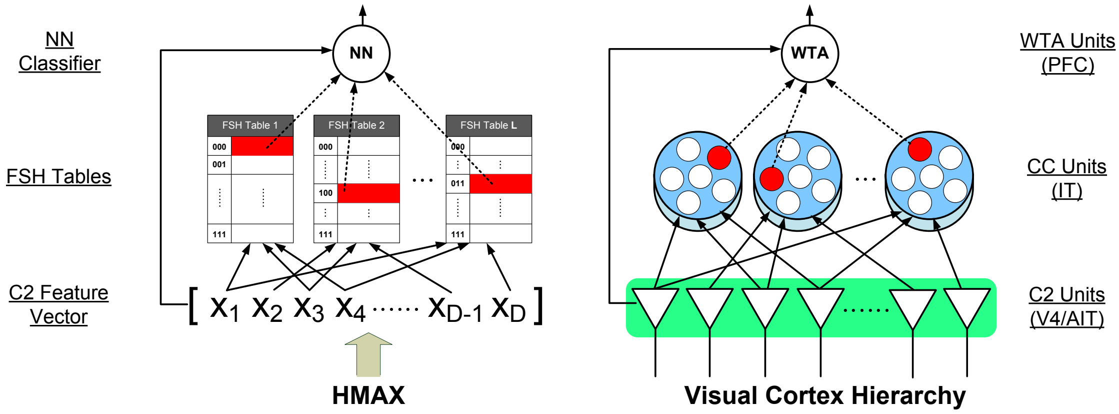


Fig. 6. The biologically plausible network mapping from the proposed framework. (Left panel) The illustration of proposed local feature-selective hashing. (Right panel) The schematic drawing of biologically plausible network. C2 units consists of D neurons, each of which is tuned for specific complex prototypes. CC units consists of many clusters of cortical columnar regions (shown as blue cylinders), each of which have several columnar regions (shown as circle inside the CC units) for memory association. After the CC units receive the stimulus from C2 units, only a few columnar regions will be activated (shown as red circle) while others are inhibited. Thus, the columnar regions in CC units fire sparsely to WTA units. WTA units match the overall C2 unit response with its associated memory from CC units and then get the answer. The tentative corresponding brain areas of various units are specified in the brackets.

Acknowledgments

This project is supported by National Science Council, Taiwan, under Grant 97-2221-E-002-238-MY3. The authors also would like to thank Prof. M.D. Fon-Jou Hsieh for advising this project, and anonymous reviewers for their extremely helpful comments/suggestions.

References

[1] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 411-426, March. 2007.

[2] H. Jhuang, T. Serre, L. Wolf and T. Poggio, "A biologically inspired system for action recognition," *IEEE 11th International Conference on Computer Vision*, pp. 1-8, Oct. 2007.

[3] Y. LeCun, K. Kavukcuoglu and C. Farabet, "Convolutional networks and applications in vision," *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pp. 253-256, June. 2010.

[4] H. Lee, R. Grosse, R. Ranganath and A.Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 609-616, 2009.

[5] D. George and J. Hawkins, "Towards a mathematical theory of cortical micro-circuits," *PLoS Computational Biology*, vol. 5, no. 10, p. e1000532, 2009.

[6] C.P. Hung, G. Kreiman, T. Poggio and J.J. DiCarlo, "Fast readout of object identity from macaque inferior temporal cortex," *Science*, vol. 310, no. 5749, pp. 863-866, Nov. 2005.

[7] J.S. Beis and D.G. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1000-1006, June. 1997.

[8] M. Muja and D.G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," *International Conference on Computer Vision Theory and Applications*, vol. 340, pp. 331-340, 2009.

[9] A. Gionis, P. Indyk and R. Motwani, "Similarity search in high dimensions via hashing," *Proceedings of the 25th International Conference on Very Large Data Bases*, pp. 518-529, 1999.

[10] Y. Weiss, A. Torralba and R. Fergus, "Spectral hashing," *Advances in Neural Information Processing Systems*, vol. 21, pp. 1753-1760, 2009.

[11] R. Salakhutdinov and G. Hinton, "Semantic hashing," *International Journal of Approximate Reasoning*, vol. 50, no. 7, pp. 969-978, Jul. 2009.

[12] T. Sato, G. Uchida and M. Tanifuji, "Cortical columnar organization is reconsidered in inferior temporal cortex," *Cerebral Cortex*, vol. 19, no. 8, pp. 1870-1888, 2009.

[13] A. Rodriguez, J. Whitson and R. Granger, "Derivation and analysis of basic computational operations of thalamocortical circuits," *Journal of Cognitive Neuroscience*, vol. 16, no. 5, pp. 856-877, Jun. 2004.

[14] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscience*, vol. 2, pp. 1019-1025, 1999.

[15] J. Mutch and D.G. Lowe, "Object class recognition and localization using sparse features with limited receptive fields," *International Journal of Computer Vision*, vol. 80, no. 1, pp. 45-57, Oct. 2008.

[16] D.H. Hubel and T.N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, no. 1, pp. 106-154, 1962.

[17] D.I. Perrett and M.W. Oram, "Neurophysiology of shape processing," *Image and Vision Computing*, vol. 11, no. 6, pp. 317-333, Jul. 1993.

[18] R. Desimone, "Face-selective cells in the temporal cortex of monkeys," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 1-8, 1991.

[19] J.P. Jones and L.A. Palmer, "An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex," *Journal of Neurophysiology*, vol. 58, no. 6, pp. 1233-1258, Dec. 1987.

[20] L. Paulevé, H. Jégou and L. Amsaleg, "Locality sensitive hashing: a comparison of hash function types and querying mechanisms," *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1348-1358, Aug. 2010.

[21] J.M. Wolfe and S.C. Bennett, "Preattentive object files: Shapeless bundles of basic features," *Vision Research*, vol. 37, no. 1, pp. 25-43, Jan. 1997.

[22] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145-175, 2001.

[23] A. Torralba, R. Fergus and Y. Weiss, "Small codes and large image databases for recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, Jun. 2008.

[24] D.M. Dacey, "Primate retina: cell types, circuits and color opponency," *Progress in Retinal and Eye Research*, vol. 18, no. 6, pp. 737-763, Nov. 1999.

[25] J.M. Geusebroek, G.J. Burghouts and A.W.M. Smeulders, "The Amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103-112, 2005.

[26] B.A. Olshausen and D.J. Field, "Sparse coding of sensory inputs," *Current Opinion in Neurobiology*, vol. 14, no. 4, pp. 481-487, Aug. 2004.

[27] S.J. Thorpe and M. Imbert, "Biological constraints on connectionist modelling," *Connectionism in Perspective*, pp. 63-92, 1989.