Cognitive Science Online, Vol.3.2, pp.1-14, 2005 http://cogsci-online.ucsd.edu

Analysis of a Biologically-Inspired System for Real-time Object Recognition

Erik Murphy-Chutorian[1,*], Sarah Aboutalib[2] & Jochen Triesch[2,3]

[1]Dept. of Electrical and Computer Engineering, and
[2]Dept. of Cognitive Science
University of California, San Diego
9500 Gilman Drive
La Jolla, CA 92093-0515

[3]Frankfurt Institute for Advanced Studies
Frankfurt, Germany

*Corresponding author. *E-mail address:* erikmc@ucsd.edu

Abstract

We present a biologically-inspired system for real-time, feed-forward object recognition in cluttered scenes. Our system utilizes a vocabulary of very sparse features that are shared between and within different object models. To detect objects in a novel scene, these features are located in the image, and each detected feature votes for all objects that are consistent with its presence. Due to the sharing of features between object models our approach is more scalable to large object databases than traditional methods. To demonstrate the utility of this approach, we train our system to recognize any of 50 objects in everyday cluttered scenes with substantial occlusion. Without further optimization we also demonstrate near-perfect recognition on a standard 3-D recognition problem. Our system has an interpretation as a sparsely connected feed-forward neural network, making it a viable model for fast, feed-forward object recognition in the primate visual system.

Introduction

Efficient detection of multiple objects in real-world scenes is a challenging problem for object recognition systems[1]. Natural scenes can contain background clutter, occlusion, and object transformations which make reliable recognition very difficult. In this work we develop a system that efficiently and accurately recognizes partially occluded objects despite position, scale, and lighting changes in cluttered real-world scenes.

Most modern recognition approaches represent specific views of objects as constellations of localized image features. The Scale Invariant Feature Transform, SIFT, is a well-known example (Lowe, 2004). In this approach, gradient histogram-

[1] Authors occasionally make a distinction between *recognition* (what is this object?), *detection* (e.g., is a face somewhere in this image?), and *multiple object detection* (is any of a set of known objects in this image?). In this paper, we use the generic term recognition to refer to all of these problems.

based SIFT descriptors are computed at Difference-of-Gaussian keypoints and stored along with a record of the key-point's 2D location, scale, and orientation relative to the training image. To detect an object in a new image, an approximate nearest neighbor search matches SIFT descriptors extracted from the image, and a Hough Transform detects and roughly localizes the object.

To improve performance for multiple objects, similar approaches have employed a quantized feature vocabulary[2], such that the set of features is shared across different object models (Murphy-Chutorian & Triesch, 2005). In this approach, every extracted local feature is compared to a much smaller set of vocabulary features by a fast nearest neighbor search, and a reference is stored with the key-point's 2D location relative to the location of the object. As the number of objects increases, the number of shared features need not grow proportionally. This benefit from shared features has been corroborated in a boosting framework (Torralba, Murphy, & Freeman, 2004). These authors demonstrated that by allowing only a fixed number of total features, using such shared features greatly outperforms a set of classifiers learned independently for each object class. Vocabulary-based recognition systems have also been proposed for single object recognition and image retrieval (Agarwal, Awan, & Roth, 2004; Leibe & Schiele, 2004; Sivic & Zisserman, 2003). This paper presents a novel framework for sharing multiple feature types, such as texture and color features, within and between different object representations. We learn probabilistic weights for the associations between features and objects so that any feature, regardless of type, can contribute to the recognition in a unified framework.

An interesting debate regarding the aforementioned recognition approaches is the question of how invariance to transformations (position, scale, rotation in plane, rotation in depth) should be achieved. On one end of the spectrum are approaches that try to hard-wire such invariance into the system by using invariant features. At the other end are approaches that try to learn certain invariance directly from training data. Our approach takes an intermediate stance, where position invariance is built into the system, and invariance to scale and pose are learned from training data.

System Overview

In brief, our system works as follows. During training, it creates a set of weighted associations between a learned set of vocabulary features and the set of objects to be recognized. During recognition, vocabulary features that are detected at interest points in the image cast weighted votes for the presence of all associated objects at corresponding locations, and the system detects objects whenever this consensus exceeds a learned threshold. In the following sections we describe these steps in more detail.

Feature Vocabulary

The recognition system uses a vocabulary of local features that quantize a potentially high-dimensional feature space. Our implementation uses color and texture feature

[2] The term *vocabulary* is analogous to the *feature dictionary* used in previous work (Murphy-Chutorian & Triesch, 2005).

vocabularies. The color features are represented as 2D hue saturation vectors, corresponding to the local average of 5x5 pixel windows. The Euclidean distance in polar hue-saturation coordinate space provides the basis for comparing color features. To learn the color feature vocabulary, we extract color features at the locations of objects in a large training set of images and cluster them with a standard K-means algorithm to arrive at our 500 entry color feature vocabulary.

The texture features are 40-dimensional Gabor jets (Lades et al., 1993) comprised of the magnitude responses of Gabor wavelets with 5 scales and 8 orientations, for details see (Murphy-Chutorian & Triesch, 2005). For vertically or horizontally oriented Gabor jets, the necessary convolutions can be efficiently calculated with separable filters. For all other orientations, the image can be first rotated and then processed with the same filters. Our implementation processes all 40 convolutions in approximately 200ms on a 2.8Ghz computer. To compare two Gabor jets, x and y, we use the normalized inner product,

$$S(\mathrm{x}, \mathrm{y}) = \frac{\mathrm{x}^T \mathrm{y}}{\| \mathrm{x} \| \| \mathrm{y} \|}, \tag{1}$$

which is robust to changes in brightness and contrast. By normalizing the vectors and computing only the inner product at runtime, the calculations are reduced. To learn the Gabor feature vocabulary we extract many Gabor jets at interest point locations from around the objects in a large set of training images. As an interest point operator we choose the Harris corner point detector which is highly stable over multiple views of an object (Harris & Stephens, 1988). We use a modified K-means clustering to compute a 4000 entry Gabor jet vocabulary. The modification of the K-means clustering consists of normalizing the jets to unit magnitude following each iteration of the algorithm.

Given either feature type, finding the nearest vocabulary features that best represent it requires a nearest neighbor search in a 2-, or 40-dimensional space, respectively. An approximate kd-tree algorithm accomplishes this efficiently (Mount & Arya, 2005). We have found that the system performs optimally if we use the six nearest Gabor jets and the single nearest color-jet for each respective vocabulary query. As a consequence, our initial encoding of the image in terms of its features is extremely sparse with only 6 out of 4000 Gabor features or 1 out of 500 color features being activated at a given interest point location.

Transform Space

A 2D-Hough transform space (Ballard, 1981; Lowe, 2004) partitions the image space into a set of regions or bins for each object. During recognition, the detected vocabulary features cast weighted votes for the presence of an object in a specific bin, storing the consensus for classification[3]. The optimal size of the bins will be discussed in its own section.

[3] To avoid the problem of boundary effects from the discrete Hough bins, each feature votes for a bin and its 8 neighboring bins.

Feature Associations

Initially, we develop a sparse set of associations between the features and objects. If an object and feature are both present in a training image, the system creates an association between the two. This association is labeled with the distance vector between the location of the feature and the center of a manually drawn bounding box around the object, discretized at the level of the bin spacing. Duplicate associations, (i.e. same feature, same object, same displacement) are disallowed. Once all of the training images have been processed in this way, the system begins a second pass through the training images to learn a weight for each of the associations. Assuming conditional independence between the inputs given the outputs, Bayesian probability theory dictates the optimum weights are given by the log-likelihood ratios,

$$w_{fm\vec{d}} \equiv \ln P\left(X_f = 1 \mid Y_{m\vec{d}} = 1\right) - \ln P\left(X_f = 1 \mid Y_{m\vec{d}} = 0\right), \tag{2}$$

where X_f is a Bernoulli random variable describing the presence ($X_f = 1$) or absence ($X_f = 0$) of feature f in the scene, and $Y_{m\vec{d}}$ is another Bernoulli random variable indicating the presence or absence of object m at a discretized spatial offset $\vec{d}$ from feature f[4]. Figure 1 shows the distribution of the log-likelihood weights for the color and the texture features. Not surprisingly, the higher-dimensional texture features tend to be more discriminative as they have a higher average log-likelihood weight.

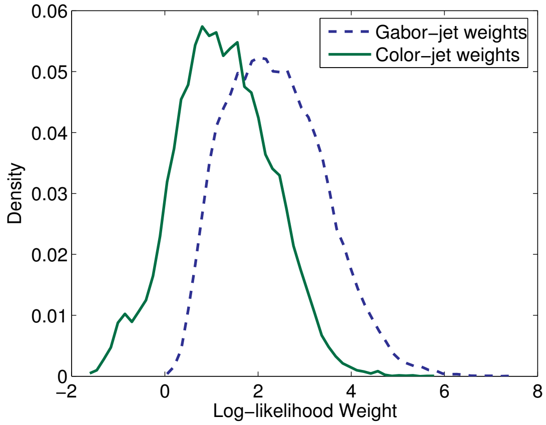


Figure 1. Distribution of Log-Likelihood Weights for each feature type

Optimum Detection Thresholds

During recognition, all of the detected features cast weighted votes to determine the presence of the objects. If any Hough transform bin receives enough activation, this suggests the presence of the object. To determine a detection criterion, we develop optimum thresholds from the maximum a posteriori (MAP) estimator under Gaussian

[4] It may seem at this point that a naive Bayes rule expansion could be applied with these log-likelihood ratios and known priors to obtain the posterior probability than an object is present, but the underlying conditional independence assumption is highly erroneous in our case and leads to rather poor performance.

assumptions. Let Y$_m$ be a Bernoulli random variable describing the presence of the object m and let tm be a continuous random variable corresponding to the maximum bin value in the Hough parameter space of m. The MAP estimator, ŷ$_m$, describes the most likely value for y$_m$ given the value of t$_m$:

$$\hat{y}_m = \arg\max_{y_m} p\left(t_m \mid Y_m = y_m\right)\mathrm{P}\left(Y_m = y_m\right) \tag{3}$$

$$= \begin{cases} 0 : \mathrm{p}\left(t_m \mid 0\right)\mathrm{P}\left(0\right) \geq \mathrm{p}\left(t_m \mid 1\right)\mathrm{P}(1) \\ 1 : \mathrm{p}\left(t_m \mid 0\right)\mathrm{P}\left(0\right) < \mathrm{p}\left(t_m \mid 1\right)\mathrm{P}(1) \end{cases}, \tag{4}$$

where P(y_m) is the prior probability that Y$_\mathrm{m}$ = my_m, and p(t_m| y_m) is the conditional pdf of t$_m$ given Y$_\mathrm{m}$ = y_m. We then define the optimum threshold, θ_m, as the value of t$_m$ which satisfies

$$\mathrm{p}\left(t_m \mid Y_m = 0\right)\mathrm{P}\left(Y_m = 1\right) = \mathrm{p}\left(t_m \mid Y_m = 1\right)\mathrm{P}\left(Y_m = 1\right) \tag{5}$$

For t$_m$ > θ_m it is more probable that the object is present in the scene, and for t$_m$ < θ_m it is more probable that the object is absent. Assuming that p(t$_m$ | y$_m$) is a Gaussian distribution, we can fully determine p(t$_m$ | Y$_m$ = y$_m$) knowing only the first and second order moments, μ$_{ml}$ and σ^2_{ml}, where l = 1 if the object is present. We estimate the moments from the training data and find θ_m by solving the quadratic equation:

$$\frac{1-p}{\sqrt{2\pi\sigma^2_{m0}}}\exp^{-\frac{(\theta_m-\mu_{m0})^2}{2\sigma^2_{m0}}} = \frac{p}{\sqrt{2\pi\sigma^2_{m1}}}\exp^{-\frac{(\theta_m-\mu_{m1})^2}{2\sigma^2_{m1}}} \tag{6}$$

where p = P(Y$_m$=1). Assuming $\mu_{m1} > \mu_{m0}$ and $\sigma^2_{ml} > \sigma^2_{m0}$ as is always the case for our data, the solution is given as

$$\theta_m = \frac{-b-\sqrt{b^2-4ac}}{2a}, \tag{7}$$

with

$$a = \sigma^2_{m0} - \sigma^2_{m1}$$
$$b = 2\left(\mu_{m0}\sigma^2_{m0} - \mu_{m1}\sigma^2_{m0}\right)$$
$$c = \mu_{m1}\sigma^2_{m0} - \mu_{m0}\sigma^2_{m1} + 2\sigma^2_{m0} - \sigma^2_{m1}\ln\left(\frac{(1-p)\sigma_{m1}}{p\sigma_{m0}}\right).$$

Experiments and Results

The CSCLAB cluttered scenes database was used to test the performance of our system (Murphy-Chutorian & Triesch, 2005). It consists of 500 scenes of 50 everyday objects against cluttered, real-world backgrounds with significant occlusion. Each scene contains 3 to 7 objects as shown in Figure 2. The objects are presented at roughly the same viewpoint in every scene, but there remains differences in depth, position, rotation, and lighting. The depth changes cause considerable scale variation among the object classes, which vary by a factor of two on the average. The system learns scale-invariant representations by building a conglomerate set of associations from training images of objects at representative scales. Alternatively, it could be trained with fewer scenes, explicitly presented at multiple scales (Burt & Adelson, 1983). In addition, the database contains scenes of all ten backgrounds without objects, as well as scenes of every background with each object by itself. All of the scenes have associated XML files that store the manually-labeled bounding boxes and names of the objects for supervised training and evaluation.

The dataset was split into three sets. The first set contained 100 multiple object scenes which were used to create the feature dictionary. The second set contained 100 additional multiple-object scenes and all of the individual object scenes. This set provided the training data for learning associations between vocabulary features and objects and the corresponding weights. The third set, containing the remaining 200 multiple-object scenes, was presented to the system for recognition.

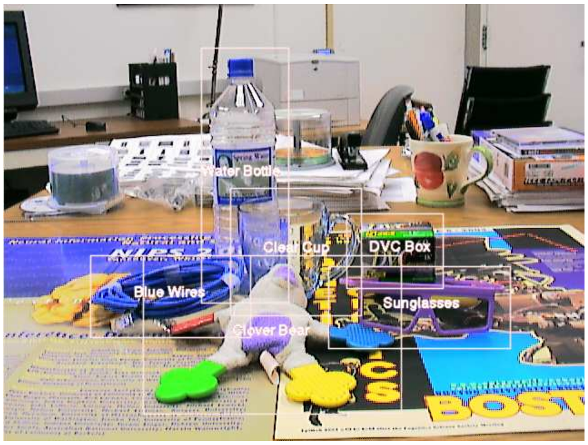
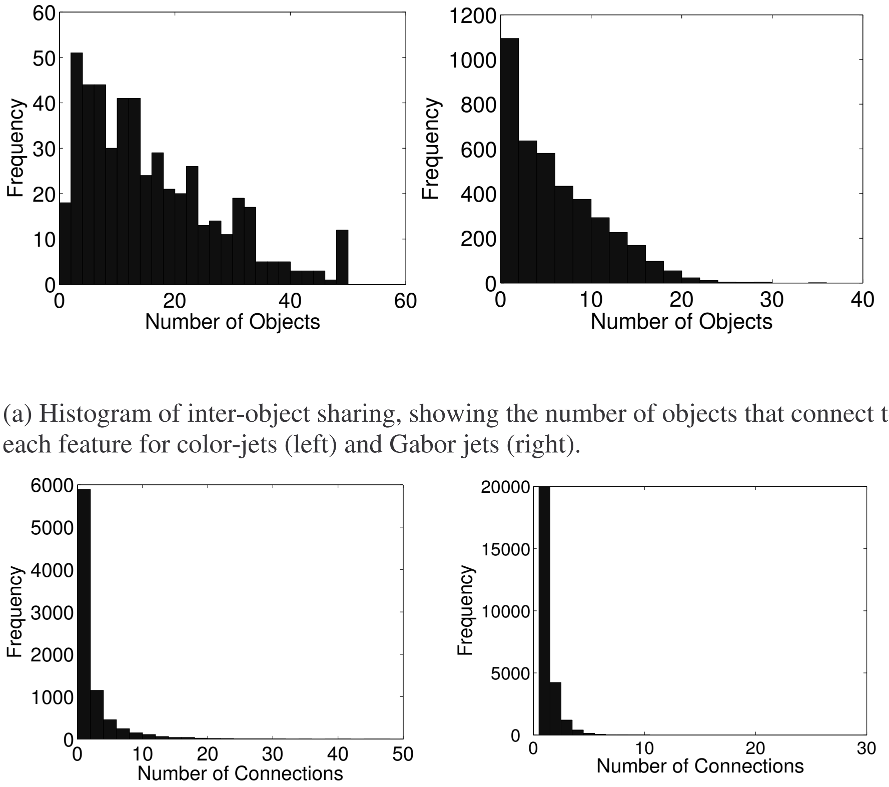


Figure 2. Labeled Example Scene from the CSCLAB dataset

Feature Sharing

Figure 3 demonstrates the amount of feature sharing in the learned representations for the 50 objects from the CSCLAB data base. In Figure 3(a) we show histograms of how frequently a feature is shared between representations of different objects. Interestingly, there is a sizable fraction of features that are shared by many objects, and only few features are not shared at all, i.e. they are specific to one object only. Figure 3(b) shows how often a feature is shared within one or multiple views of a single object. Noting that there are no duplicate associations, this denotes the number of associations between this feature and the object with different discretized displacements. One can see that this intra-object sharing is happening less often than the inter-object sharing, but this is a meaningless ratio, since it directly depends on our choice of the Hough bin size and number of objects.



(a) Histogram of inter-object sharing, showing the number of objects that connect to each feature for color-jets (left) and Gabor jets (right).

(b) Histogram of intra-object sharing, showing the number of times a feature connects to the same object for color-jets (left) and Gabor jets (right).

Figure 3. Feature Sharing

Optimal Bin Size

The optimal size of the Hough transform bins is determined by a trade-off between two competing factors. If the bin size is too small, votes from the same object may fall into different bins because of variations in object appearance such as scale or rotation. Larger bins, however, increase the risk of a spurious accumulation of votes from background clutter or unrelated objects into a single bin, which can lead to a false positive detection. Because of this trade-off, there exists an intermediate bin size that yields optimal performance (Aboutalib, 2005). We investigated this effect by systematically varying the bin size[5]. Figure 4 shows the result. The tradeoff favoring intermediate bin sizes is clearly visible. Based on this result, we use 16x16 pixel Hough transform bins to maximize recognition.

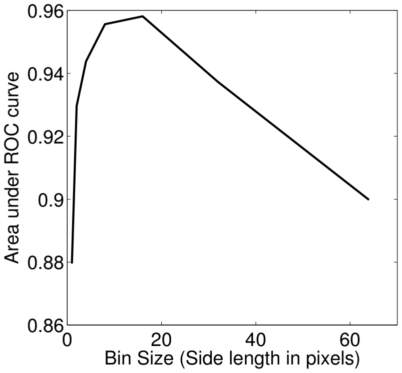


Figure 4. Area under the averaged ROC curves for various bin sizes

Recognition Performance

Figure 5 and Figure 6 show histograms of the detection rates and false positive rates for the 50 objects in the CSCLAB dataset. The detection rate is defined as the fraction of objects that were successfully detected, and the false positive rate is the fraction of images in which an object is incorrectly detected. In this application, the system is able to detect most of the objects more than 80% of the time while maintaining less than a 5% false positive rate. The system has the most difficulty with the objects that lack sufficient texture, or have significant transparencies. Performance examples are shown as ROC curves for the best, median, and worst individual ROC curves are given in Figure 7. Figure 8 shows an average of the spline-interpolated ROC curves for all of the objects. In the course of the experiment, 10 of the 50 objects were perfectly recognized with a 100% detection rate and no false positives. Figure 9 provides examples of the system's recognition ability. On a 2.8Ghz personal computer, our system requires approximately one second to recognize all of the objects in a 640x480 pixel image.

[5] In this experiment we kept the bin size fixed for every object, but an object specific selection of the bin size may further improve performance.

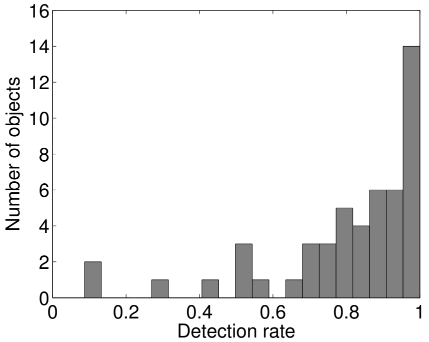


Figure 5. Histogram of the individual object detection rates at optimum thresholds

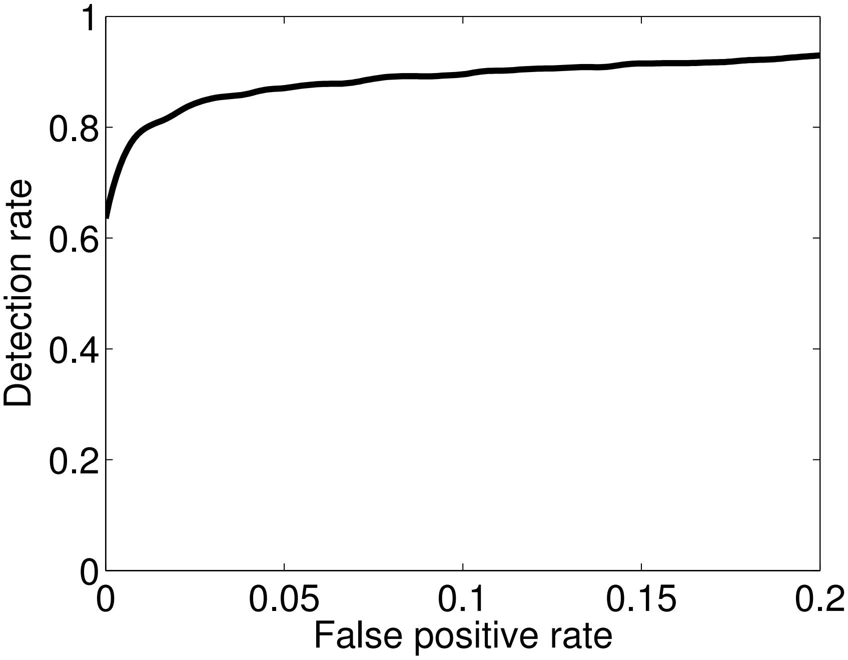


Figure 6. Histogram of the individual object per-image false positive rates at optimum thresholds

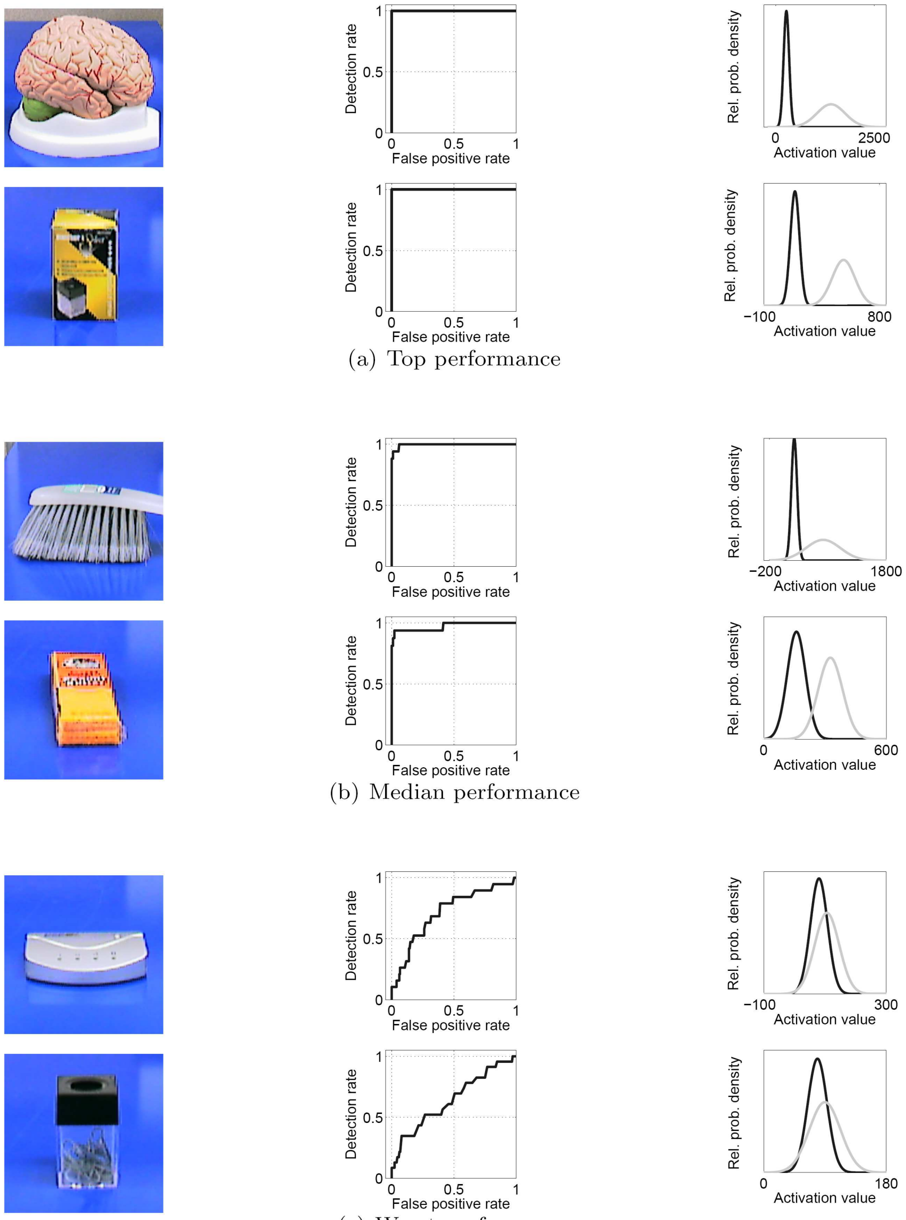
Neural Network Interpretation and Relation to Models of Biological Object Recognition

It is frequently argued that the remarkable speed of primate object recognition suggests a processing architecture that is essentially feed-forward in nature, and prominent models of biological object recognition are feed-forward processes (Fukushima, Miyake, & Ito, 1983; Riesenhuber & Poggio, 1999). Feed-forward models are unlikely to be able to account for all aspects of primate object recognition, but they may be a reasonable approximation in many situations.

We can interpret our system as a simple feed-forward neural network. In this case, the input layer consists of the vocabulary features at every possible discretized location. The output layer consists of the objects at every possible Hough bin. The activation of an output node, $y_j = y(m_j, q_j)$, is given by the linear summation,

$$\begin{aligned} y_j &= \sum_i w_{ij} x_i, \\ x_i &\equiv x_{(f_i, p_i)}, \\ w_{ij} &\equiv w_{f_i m_j (q_j - p_i)}, \end{aligned} \tag{9}$$

and the weights of the network are the log-likelihood ratios of the features mentioned earlier. In this context, x_i is the *ith* binary input node that “fires” whenever the shared vocabulary feature, f_i, is found anywhere inside the unit’s “receptive field,” and *w*ij is the weighted connection between y_j and *x*i. p_i is the location of input x_i, and q_j is the location of y_j . We assume $w_{ij} = 0$ whenever *y*j and x_i are not connected. Although the weighted connections are learned from the relative displacement between the input and output nodes, this can be interpreted as *weight sharing* in a neural network with connections based on absolute displacements.



(a) Top performance

(b) Median performance

(c) Worst performance

Figure 7. ROC curves and estimated conditional pdfs (black: object absent, gray: object present) for individual object examples

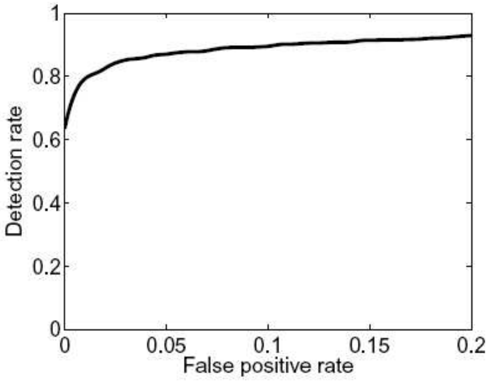


Figure 8. Averaged ROC curve for all 50 objects

The feed-forward neural network interpretation of our system suggests that one could view it as an abstract model of primate object recognition. In fact the introduction of Gabor wavelet features into computer vision systems was inspired by biological findings. In this context, our shared Gabor jet features loosely correspond to shape selective cells in area V4. Compared to the other models mentioned above, the binning operation inherent in the Hough transform mechanism corresponds to non-linear operations that introduce a degree of shift invariance in the above models. The sparseness of connections from these features to object detectors (corresponding to populations of cells in inferotemporal cortex) is also in line with biological considerations, but in stark contrast to many previous models of biological object recognition, we obtain excellent performance on a difficult real-world recognition problem. To do so in real-time paves the way for the development of more elaborate models of visual cognition that model object recognition and learning in the context of ongoing behavior.

Discussion

We presented a new framework for multiple-object detection with a vocabulary of shared features. Using multiple feature types and sparse, weighted associations between vocabulary features and objects, we demonstrated object detection in cluttered real-world scenes despite significant scale variation and occlusion in real-time. Since the system can be interpreted as a feed-forward neural network, it may be viewed as an abstract model of object recognition in the primate visual system, although this was not the main focus of this research.

In a full 3-D recognition task on the otherwise much simpler COIL database, our system showed excellent performance. Evaluating our system on a full 3-D recognition problem that also includes clutter, occlusions, and lighting variations remains a topic for future research. At present, there are no available benchmark databases of this kind. Performance gains could be achieved by the addition of other feature types. Transparent objects and objects lacking unique texture and color were the most difficult to detect, and this could be remedied by the addition of features that could detect these objects by their characteristic shape. The framework presented in this paper easily accommodates additional features. A further avenue for future research is the incorporation of stereo information and the explicit modeling of object occlusions (Eckes, Triesch, & Malsburg, 2005).

We would also like to investigate the ability to learn objects with only minimal supervision, since hand-labeled training data as we have used here is tedious to create. Recent pilot work has demonstrated this system's potential for learning object representations in a semi-autonomous fashion through online demonstration, where objects are simply shown to the system for an extended period of time as they undergo scale and pose changes and the system detects, tracks, segments, and learns to recognize these objects without additional human intervention (Murphy-Chutorian, Kim, Chen, & Triesch, 2005).

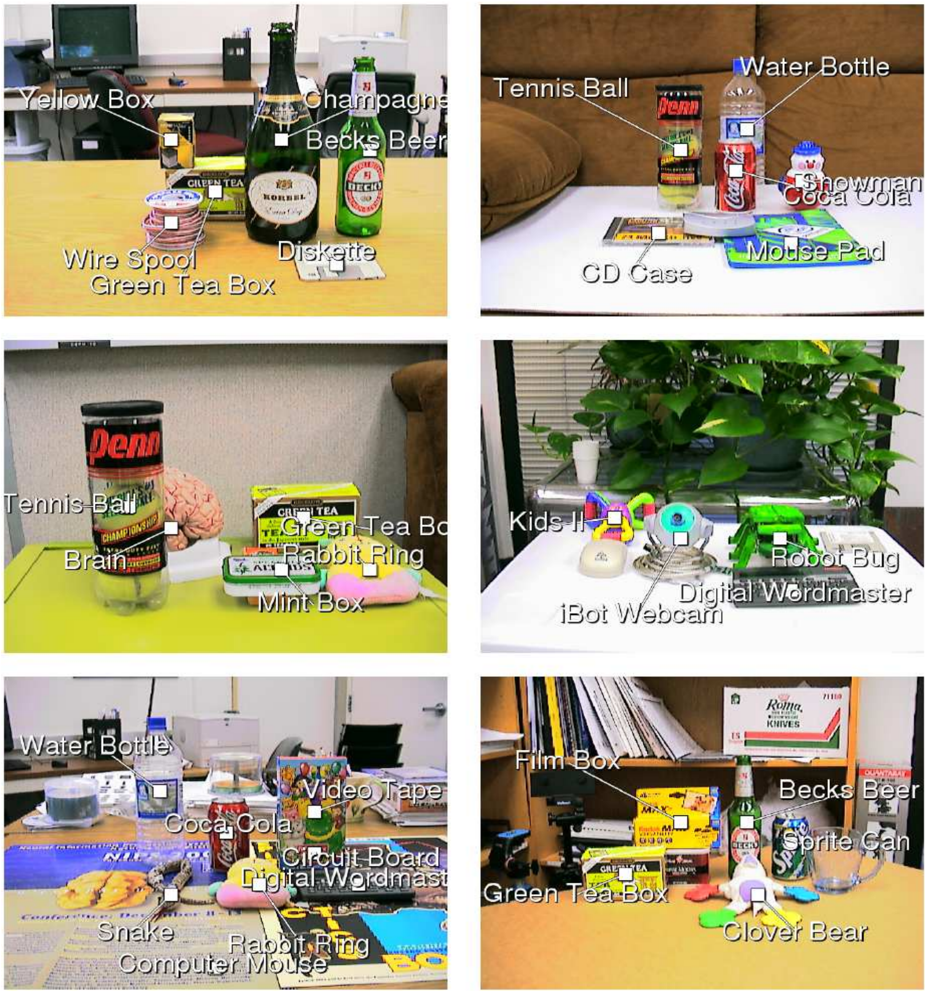


Figure 9. Example Recognition Results (squares indicate the estimated object center)

Acknowledgments

This work was supported by the National Science Foundation under grants IIS-0208451 and IIS-0233899. We would like to thank Hyundo Kim, Huei-Ju Chen, Emily Ashbaugh and Nathan Rosencrans for their help in creating the cluttered scenes database. We would also like to thank Boris Lau, Alan Robinson and Tom Sullivan for their helpful suggestions and programming contributions.

References

Aboutalib, S. (2005). *Position invariance in a view-based object recognition system.* Unpublished honor's thesis, University of California, San Diego.

Agarwal, S., Awan, A., & Roth, D. (2004, November). Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26* (11), 1475–1490.

Ballard, D. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition, 13* (2), 111–122.

Burt, P. J., & Adelson, E. H. (1983). The laplacian pyramid as a compact image code. *IEEE Transactions on Communications, COM-31*,4, 532–540.

Eckes, C., Triesch, J., & Malsburg, C. von der. (in press). Analysis of cluttered scenes using an elastic matching approach for stereo images. *Neural Computation*.

Fukushima, K., Miyake, S., & Ito, T. (1983). Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics, SMC–13* (5), 826–834.

Harris, C., & Stephens, M. (1988). A combined corner and edge detector. *In Proceedings of the Alvey Vision Conference* (pp. 147–151). Manchester.

Lades, M., Vorbr¨uggen, J. C., Buhmann, J., Lange, J., Malsburg, C. von der, W¨urtz, R. P., et al. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers, 42*, 300–311.

Leibe, B., & Schiele, B. (2004). Scale invariant object categorization using a scale-adaptive mean-shift search. *In Proc. deutsche arbeitsge- meinschaft fr mustererkennung pattern recognition symposium*. Tuebingen, Germany.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60* (2), 91–110.

Mount, D., & Arya, S. (2005, May). ANN: A library for approximate near- 12 est neighbor searching, version 1.1. (http://www.cs.umd.edu/_mount/ ANN/).

Murphy-Chutorian, E., Kim, H., Chen, H.-J., & Triesch, J. (2005). Scalable object recognition and object learning with an anthropomorphic robot head. *In*

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. San Diego, CA.

Murphy-Chutorian, E., & Triesch, J. (2005, January). Shared features for scalable appearance-based object recognition. *In Proceedings of the IEEE Workshop on Applications of Computer Vision*. Breckenridge, CO, USA.

Nene, S., Nayar, S., & Murase, H. (1996, February). Columbia object image library (COIL-100) (Tech. Rep.). Columbia University.

Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience, 2*, 1019–1025.

Sivic, J., & Zisserman, A. (2003, October). Video google: A text retrieval approach to object matching in videos. *In Proceedings of the IEEE International Conference on Computer Vision*. Nice, France.

Torralba, A., Murphy, K., & Freeman, W. (2004). Sharing features: efficient boosting procedures for multiclass object detection. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.