

Sharing Resources: Buy Attention, Get Object Recognition

Vidhya Navalpakkam
University of Southern California
Computer Science Department
Los Angeles, CA 90089 - USA
navalpak@usc.edu

Laurent Itti
University of Southern California
Computer Science Department
Los Angeles, CA 90089 - USA
itti@usc.edu

Abstract

Inspired by nature's policy of sharing resources, we have enhanced our attention model with minimal extra hardware to enable the twin powers of object detection and recognition. With just the elementary information available at the preattentive stage in the form of low-level feature maps tuned to color, intensity and orientation, our model learns representations of objects in diverse, complex backgrounds. The representation starts with simple vectors of low-level feature values computed at different locations on the object (views of the object). We then recursively combine these views to form instances, in turn combined into simple objects, composite objects, and so on, taking into account feature values and their variance. Given any new scene, our model uses the learnt representation of the target object to perform top-down biasing on the attention system such as to render this object more salient by enhancing those features which are characteristic of the object. Experimental results verified the null hypothesis that the enhanced model would take half the number of fixations as that taken by the naive bottom-up model to detect all targets in a scene. Our model is also able to recognize a wide variety of simple objects ranging from geometrical objects to soda cans, handicap signs, and many others under noisy conditions. There are few false negatives and false positives. The good performance of our lightweight model suggests that the human visual system may indeed be sharing resources extensively and attention and object recognition may be so intimately related that if we buy attention, we might get the other with minimal effort!

1. Introduction

In our everyday life, each of our retinas is bombarded with enormous amounts of information, and transmits on the order of 10^8 bits/s to the thalamus for central processing. Yet, our visual system keeps us largely unaware of

the complexity of this incoming information, and helps us localize and detect the objects that are relevant to our current behavior, and recognize the fixations in a fairly effortless manner. While enormous progress has been made in recent years towards understanding the basic principles of information processing in visual cortex, we currently lack a solid computational understanding of how high-level task demands may allow us to filter out large amounts of irrelevant information while focusing our attention on those visual elements that are salient and relevant to the current task.

Previous work in our lab focussed on how human attention is deployed under free-examination of any visual scene [4, 3]. We have developed a bottom-up salience model that mimics the human attentional system by pruning the search space and attending to the interesting scene locations. A natural question to ask next is "what is this interesting scene location/fixation entity? If it is indeed interesting, how can we learn it so that we can detect it faster and recognize it in any future scene?" Since nature and biology generally exhibit sharing of resources and reuse of hardware, we are particularly interested in knowing what the minimal extra hardware is that would be required to enhance the human attentional system with the twin powers of object detection and recognition. Understanding the extent to which attention and object recognition share resources can also give us insight into the internal representations of objects.

Apart from the desire to understand the way the visual brain works, our motivation also stems from the enormous potential applications of such a model. Applications range from simple search tasks requiring single object detection to more complex search tasks involving multiple object detections and recognition such as in machine assembly and video surveillance.

We have enhanced our bottom-up salience model to yield a simple, yet powerful architecture to learn target objects from training images containing the targets in diverse, complex backgrounds. Our enhanced model shows promising results and detects the target objects in half the number of fixations as that taken by our earlier model that was purely

bottom-up and had no bias towards finding specific objects. Further, there are few false negatives and false positives during object recognition.

2. Related Work

In an earlier work [5], we learnt the absolute features (local color, intensity and orientation information, at nine spatial scales) of simple target objects, and biased the bottom-up salience model with the learnt weights. There are significant differences between our current model and that previous model. Earlier, we learnt absolute features whereas now we learn the center-surround features that are more robust in the presence of noise or changes in absolute illumination, since they only consider the properties of a location relative to its neighborhood. Earlier, the model did not learn any object hierarchy and could not generalize. Instead, it could only detect the target instances that it had learnt. On the other hand, our current model can generalize by combining object classes into a more general super-class (e.g: combining individual views into instances, instances into simple objects, and so on). Finally, our current model implements a new object recognition model, which uses the learnt features to recognize the fixations, unlike the earlier model.

Among the well known models of visual search is the guided search model [13]. The author provides a good discussion on top-down biasing of visual search. In [14] the authors discuss preattentive object files as bundles of basic shapeless features. This seems to suggest that object recognition may be difficult at this stage. Interestingly, our model also uses the information available at the preattentive stage, but, despite the lack of explicit encoding of shape, our model still performs good detection and recognition of a wide range of geometrical objects varying in shapes and size.

Another interesting attentional model for visual search can be found in [10]. The authors use iconic scene representations to guide attention during visual search. But their model does not consider variability in the object appearance and hence cannot generalize.

There are several models of object recognition but most of them do not use the attention system or use non-preattentive information to perform recognition [9, 11]. However, we use only the elementary information available at the preattentive stage and reuse the attention system to perform object detection/recognition.

To summarize, we are not aware of any complete implementation that integrates attention and object detection/recognition systems.

3. Architecture

The general architecture of our model is as shown in figure 1. We maintain 7 types of features, for which wide evidence exists in the mammalian visual systems. One feature type encodes for on/off image intensity contrast [6], two encode for red/green and blue/yellow double-opponent channels [7, 2], and four encode for local orientation contrast [1, 12]. Each feature is computed in a center-surround structure akin to visual receptive fields and we compute six feature maps for each type of feature[3].

3.1. Top-down Biasing for detecting the target objects

Attention is deployed according to the bottom-up salience model [3]. During the learning phase, the model is guided by a binary target mask that highlights the targets in the input image. When the model hits the target, a few locations are chosen around the salient location using covert attention. Each chosen location is called a view. For each view, we learn the center-surround features (after a normalization) at multiple scales. Specifically, a 42-component feature vector represents a view (six center-surround scale pairs, for four orientation, two color opponent, and one intensity feature types). Thus, we obtain a “bunch” of views (each view being represented by a feature vector) contained in the current instance of the target. Next, we gain a single, general representation of the instance by combining the views in a manner discussed below (assuming that each view is independent and equally likely). We repeat this process to gain a single, general representation of the object by combining the various instances (assuming each instance is independent and equally likely). In general, we have the following rules for combination of several classes z_i to form a general representation of the super-class Z . Let each class be a random variable with normal distribution $N(\mu_i, \Sigma_i)$ where i denotes the i^{th} class; μ_i denotes a vector of the mean of the sub-class features and Σ_i denotes the covariance matrix whose diagonals are the variance of sub-class features. Due to our assumption that the different features are mutually independent, Σ_i is a diagonal matrix. We define Z such that an observation x comes from Z if it comes from any of its sub-classes z_i . Using Bayes rule, we have:

$$\begin{aligned} P(X|Z) &= \sum_i P(X|z_i)P(z_i) \\ &= \sum_i P(X|z_i) * w_i \end{aligned}$$

where

$$w_i = 1/n$$

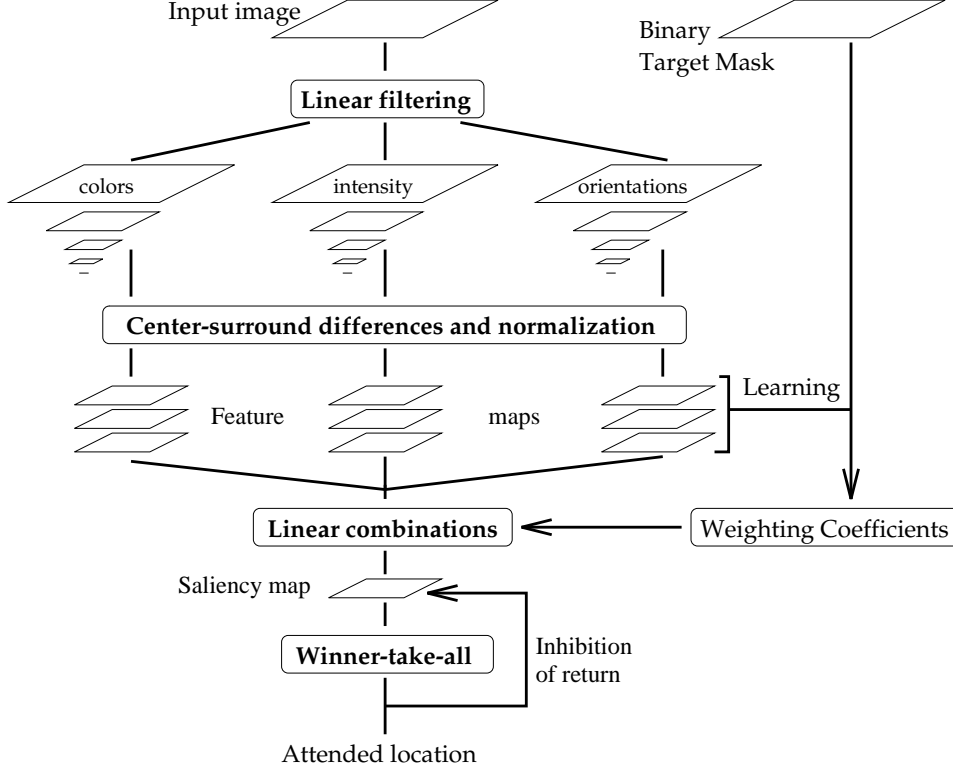


Figure 1. General architecture of the visual attention system studied here. Early visual features are extracted in parallel in several multi-scale “feature maps”, which represent the entire visual scene. Such feature extraction is achieved through linear filtering for a given feature type (e.g., intensity, color or orientation), followed by a center-surround operation which extracts local spatial discontinuities for each feature type. If the target object’s feature weights are already learnt, then they are used to weight the feature maps that are subsequently combined into a unique “saliency map.” After such combination is computed, a maximum detector selects the most salient location in the saliency map and shifts attention towards it. This location is subsequently suppressed (inhibited), to allow the system to focus onto the next most salient location.

where n is the number of object classes.

$$P(X|z_i) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_i|}} e^{-\frac{(X-\mu_i)^T \Sigma_i^{-1} (X-\mu_i)}{2}}$$

The mean and variance of Z are as follows (the proof is not shown here but is easy to demonstrate):

$$\mu = \sum_i w_i * \mu_i \quad (1)$$

The variance of the j^{th} feature is:

$$\Sigma(j, j) = \sum_i w_i * (\Sigma_i(j, j) + \mu_i(j)^2) - \frac{\mu(j)^2}{\sum w_i} \quad (2)$$

In general, Z has a multi-modal distribution. But as a first approximation and to achieve recursiveness in our implementation, we approximate this multi-modal distribution

by a normal distribution $N(\mu, \Sigma)$. That is, we consider only up to the second moment and discard information contained in higher moments.

By running the model on several images containing different instances of an object, we can learn the representation of the instances and combine them to form a representation of an object, and so on. Thus we have a general rule for combining representations at any stage of the object hierarchy where views are the leaves (level 0), instances are level 1, simple objects are level 2, composite objects are level 3, etc.

When we want to detect a specific target object (say Z) in any scene, we bias the visual cortex with the learnt features. A feature is considered to be relevant if its mean feature value is high and its feature variance is low. Hence, we determine the feature weight as $\frac{\mu}{1+\Sigma}$. In order to promote the target in all feature channels, each channel promotes itself proportionally to the maximum feature weight or value of its subchannels. For instance, if the target has a high

value of redness at some scale, then the weight of the red channel increases and so does the weight of the color channel. Hence, those channels that are irrelevant for this target are weighted down or not considered while contributing to the salience (e.g., for detecting a red object, the orientation of edges is irrelevant; so the orientation channel’s weights are decreased so as to promote only color). The weighted feature maps are then combined to form conspicuity maps that are in turn combined in a weighted manner to form the salience map. In the salience map thus formed, all scene locations whose local features are the same as the target’s relevant features become more salient. Thus, the target is more likely to draw attention. However, it should be noted that all scene locations whose relevant features are a superset of the target’s relevant features also become salient (e.g., a red ellipse also becomes salient if we are searching for a red circle). This generates some false positives, which can be removed at the recognition stage.

3.2. Object Recognition

We allow the model to freely examine the scene and fixate on the salient locations in the scene. At each fixation, we have the normalized center-surround features at that location. It is interesting to ask how far we can go towards object recognition using this elementary information. We propose a simple, recursive architecture for object recognition that shares most of its resources with the attentional system and utilizes minimal extra resources.

As described in the earlier section, we first learn the object hierarchy and store it in a knowledge base, which is just a collection of all known objects. We recognize each fixation by proceeding from coarser to finer granularity while finding the object class that gives the maximum likelihood estimate, i.e. find the object o that maximises the chances of occurrence of the fixation f . We recursively compute $Match(f, x)$ that gives the best match among all objects from the root upto level x in the object hierarchy. Initially, x is the highest level in our object hierarchy. There are two important reasons for doing this. The highest level nodes in the hierarchy have the most coarse, general levels of representation of their finer descendent nodes; and there are fewer nodes in the higher levels than the lower levels. Hence, we can prune our search space by first matching the fixation with the higher level nodes and pruning the subtrees rooted at those nodes that do not match.

We compare the likelihood estimates of the nodes upto level x in an attempt to find a unique maximum which is significantly higher than the other likelihood estimates. If we find a unique maximum, then we declare a unique object match. Else, we declare ambiguity. Accordingly, we have the following two cases.

Case 1: Unique match upto level x

Since the object’s representation at level x is a coarse generalization of its representation at lower levels, we proceed deeper into the object hierarchy to investigate if the fixation indeed matches any instance or view that we have seen in the past. Hence, we evaluate $Match(f, x-1)$ by comparing the likelihood estimates of nodes at the next lower level $x-1$. We recurse until we hit the leaves (views); or until the parent node provides a better match than its children nodes (in which case we prune the sub-tree rooted at the parent node). Thus, the algorithm is guaranteed to terminate. Let’s suppose that we obtain a unique match at termination. Now, there are several interesting possibilities. How should the model react if it finds a unique match but the likelihood estimate is low. Or how should the model react if the parent node matches but none of the children nodes match?

Suppose we find a unique match whose likelihood estimate is low, we declare failure to recognize the fixation. As a future extension, we can prompt the user to reveal the identity of the fixation entity and add it to our knowledge base. Suppose that at the end of the recursion, we find that there is a unique match at some object level y that is higher than the level of views or instances, then we declare that the fixation is recognized. Since none of the children, i.e., no instance or view matches, as a future extension, we can seek the user’s guidance to tell us if this fixation entity should be added to our knowledge base.

Case 2: Ambiguity upto level x

Due to the lack of certainty upto this level, we proceed to the next lower level, seeking better matches that will help us resolve ambiguity. We evaluate $Match(f, x-1)$ and so on until we hit the leaf or until the parent node provides a better match than its children nodes. At termination, if we cannot find a unique match at any level, we declare failure to recognize. As a future extension, we intend to seek the user’s guidance to know the identity of the unknown object and add it to our knowledge base.

4. Experimental Results

We ran our model on databases of training images that ranged from artificial images of simple geometrical objects like squares, circles at different orientations and sizes to natural images of more complex objects like coke cans, traffic signs, handicap signs in diverse backgrounds. The model learnt the features of the different training objects and organized the information in a hierarchical manner with objects composed of instances that are in turn composed of views.

Next, we tested our model’s ability to search/detect a single learnt object in a new scene. To verify our results, we

postulated 3 hypotheses. The null hypothesis H_0 predicted that the enhanced model would take half the number the fixations taken by the naive bottom-up model to detect the targets. The alternative hypothesis H_1 postulated that the enhanced model would take more than half the number of fixations as the naive and the last alternative hypothesis H_2 predicted that the enhanced model would take less than half the number of fixations as the naive model. We considered the significance scores for a significance level of 0.05. Results confirmed the null hypothesis in most cases. In a few other cases, results supported H_2 that indicated even better performance. An example of a comparison between the number of fixations taken by the two models are shown in figure 2. Our model efficiently detected the target even in scenes with poor resolution. In all of nearly 300 test images, our model detected the targets efficiently.

To test the ability of our model to recognize arbitrary fixations, we ran it on the same test images mentioned previously. But this time, the model was task-free and fixated according to the bottom-up saliency of locations in the scene. There were few false negatives and false positives (see figure 3).

5. Discussion

Our main contribution in this paper is to highlight the extent to which attention and object recognition might share resources. We are already able to achieve substantial performance in object recognition even with the elementary information available at the preattentive stage.

With minimal extra hardware, we have enhanced our existing architecture for deploying attention to perform object detection and recognition of simple objects in complex naturally cluttered scenes. Our model can learn representations of the object from different locations (views) within an instance, combine these views to form representations of object instances, and further combine these instances to form the general object representation. Thus, we have a simple recursive scheme to obtain the general representation of a super-class as a combination of representations of classes that are composed of sub-classes and so on. Our simple, biologically plausible architecture shows a remarkable improvement in performance during object detection in diverse complex scenes when compared to the naive bottom-up model. Detection of single targets is real time and the model can recognize objects ranging from simple geometrical objects like squares, triangles to more complex ones like coke cans, handicap signs, traffic signs etc.

The good performance of this lightweight model suggests that we may not require any sophisticated machinery demanding complex dedicated filters or explicit encoding of shape in order to provide a first rapid detection and some recognition of even fairly complex objects in cluttered

scenes. Sharing of resources and reuse of hardware seems to be the guide to understanding the way the visual brain works.

The potential applications of our model range from tasks demanding visual search for single targets to sequential searches for multiple targets. For instance, our model can be used in a task like machine assembly where the target components are to be detected in a strict sequence. A potential application in daily life is by running the model on mobile robots that can serve their masters better by finding the tooth-brush and tooth-paste in the morning, followed by the coffee mug and the dress and shoes and briefcase before the master leaves for his work. In video surveillance, our model can serve to detect single suspicious targets in video sequences.

Apart from commercial applications, the model can also be used to compare performance against human observers to give us insight into how the human visual system might detect and recognize objects.

6. Future work and Conclusion

We have proposed an architecture to achieve fast, reliable localization, detection and recognition of simple targets. We would like to extend this to enable detection and recognition of more complex targets that are composed of several simple targets. In particular, we would like to see how to combine distinct simple objects into a complex composite object such that the resulting representation does not possess a high feature variance. This will take us from the current unimodal distributions to multimodal distributions for each feature.

We would like to make the knowledge base learnable where the model fixates and if the fixation entity is new, it learns the features and automatically classifies the fixation entity into some object category and updates the knowledge base. While searching for multiple targets in any video sequence, we would like to know the optimal biasing strategy that will enable efficient detection of all targets. This is a difficult task since the biasing strategy can depend on the targets (while looking for a white and black object, it is better to bias sequentially in some order than to combine and look for gray!) or the nature of task (in machine assembly, the strict sequence might force a sequential search for target components; finding target1 AND target2 versus finding target1 OR target2) or more complex factors such as difficulty in finding the targets, probability of occurrence of target etc.

We would also like to study how attention and recognition of parts can guide us towards recognition of the whole object. Since, our ultimate aim is to study how task influences attention, we would like to find the task-relevant objects in any scene and bias for them - detect and recog-

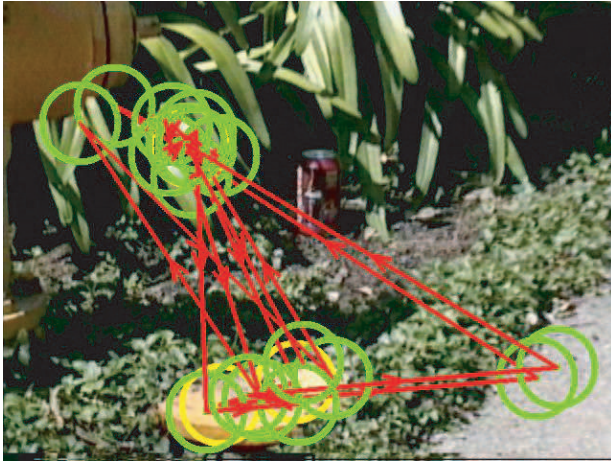


Figure 2. The example on the left shows the attentional trajectory during free examination of this scene by our bottom-up saliency model. Even after 20 fixations, the model did not attend to the coke can, simply because its saliency was very low compared to that of other conspicuous objects in the scene. Displayed on the right is the attentional trajectory after top-down biasing for the coke can object class (built from instances and views of the coke can from other photographs containing the can in various settings). Our model detected the target as early as the third fixation.

nize them in real time [8]. Towards this end, we would like to conduct psychophysics experiments and compare eye movements generated by the model versus human observers to further our understanding of the human attentional system.

7. Acknowledgements

This work is supported the National Science Foundation, the National Eye Institute, the National Imagery and Mapping Agency, the Zumberge Research and Innovation Fund and the Charles Lee Powell Foundation.

References

- [1] R. L. DeValois, D. G. Albrecht, and L. G. Thorell. Spatial-frequency selectivity of cells in macaque visual cortex. *Vision Research*, 22:545–559, 1982.
- [2] S. Engel, X. Zhang, and B. Wandell. Colour tuning in human visual cortex measured with functional magnetic resonance imaging [see comments]. *Nature*, 388(6637):68–71, Jul 1997.
- [3] L. Itti and C. Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40(10-12):1489–1506, May 2000.
- [4] L. Itti and C. Koch. Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, Mar 2001.
- [5] L. Itti and C. Koch. Feature combination strategies for saliency-based visual attention systems. *Journal of Electronic Imaging*, 10(1):161–169, Jan 2001.
- [6] A. G. Leventhal. *The Neural Basis of Visual Function (Vision and Visual Dysfunction Vol. 4)*. CRC Press, Boca Raton, FL, 1991.
- [7] A. Luschow and H. C. Nothdurft. Pop-out of orientation but no pop-out of motion at isoluminance. *Vision Research*, 33(1):91–104, 1993.
- [8] V. Navalpakkam and L. Itti. A goal oriented attention guidance model. *Proc. 2nd Workshop on Biologically Motivated Computer Vision (BMCV’02), Tuebingen, Germany, November 2002*.
- [9] R. P. Rao and D. H. Ballard. Dynamic model of visual recognition predicts neural response properties in the visual cortex. *Neural Computation*, 9(4):721–63, May 1997.
- [10] R. P. Rao, G. Zelinsky, M. Hayhoe, and D. H. Ballard. Eye movements in iconic visual search. *Vision Research*, 42(11):1447–1463, Nov 2002.
- [11] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, Nov 1999.
- [12] R. B. Tootell, M. S. Silverman, S. L. Hamilton, R. L. De Valois, and E. Switkes. Functional anatomy of macaque striate cortex. iii. color. *Journal of Neuroscience*, 8(5):1569–93, May 1988.
- [13] J. Wolfe. Visual search in continuous naturalistic stimuli. *Vision Research*, 34:1187–1195, 1994.
- [14] J. M. Wolfe and S. C. Bennett. Preattentive object files: shapeless bundles of basic features. *Vision Research*, 37(1):25–43, Jan 1997.




















Target	Mean	Standard deviation	Supporting hypothesis	False positives	False Negatives
	1.11348	1.2115	H_1	9.38	4.17
	1.76267	2.96217	H_0	10.37	3.05
	1.03295	1.62655	H_1	0	57.14
	2	0.00001	H_0	0	10.00
	3.9	3.31111	H_1	0	27.59
	0.307739	0.271301	H_1	6.25	10.94
	2	0.00001	H_0	0	20.00
	1.18122	0.762739	H_1	0	5.85
	2	0.00001	H_0	0	90.00
	16.4	19.0568	H_2	0.65	5.84
	10.9	14.0263	H_2	0	10.00
	2.4	1.17724	H_0	7.14	21.43
	2.2	1.11907	H_0	0	41.67
	2	0.00001	H_0	0	50.00
	21	0.00001	H_2	0	5.00
	1.83333	0.593847	H_0	10.00	10.00
	2.27778	1.10383	H_0	0	7.00
	3.71429	3.55834	H_0	3.70	33.33
	3.15	18.5185	H_0	23.11	8.49

Figure 3. This table shows the top-down biasing and recognition statistics of our model for a sample from our database of objects. The first column is the target object that we biased the model for, the second column shows the mean factor of improvement (number of fixations taken by the naive bottom-up model to detect the target over the number of fixations taken by the biased model); the third column shows the standard deviation of the factor of improvement; the fourth column shows the supporting hypothesis for a significance level of 0.05. While the maximum factor of improvement was as high as 20, the average improvement was around 2.5. The null hypothesis H_0 (mean factor of improvement = 2.0) was supported by majority of the target objects. The subsequent columns show the statistics for the hierarchical recognition of arbitrary fixations. As an initial implementation, we considered a simple object hierarchy with just 3 main levels (all objects, their instances and their views) and at the 4th level was the root that was a general class combining all the objects. We allowed the model to freely examine the scene and recognize the fixations until it hit all the target objects. The fifth column shows the percentage of false positives (non-target locations that were falsely recognized as the target) and the sixth column shows the percentage of false negatives (target locations that were not recognized as the target). Despite the simplicity of the model, there were few false positives. Since we are attempting to recognize fixations by looking at just one location on the object, we generated some false negatives.