CHAPTER 7: OBJECT AND SCENE PERCEPTION
COVER SHEET

Authors: Owen Lewis and Tomaso Poggio

Affiliations (for both):
Center for Biological and Computational Learning, MIT
McGovern Institute for Brain Research MIT
Department of Brain and Cognitive Science MIT

For correspondence: Owen Lewis,
olewis@mit.edu
(303) 349 1106

Chapter length: 10495 words, including figure captions. 6 figures.

Code: HMAX, deep belief networks/RBMs, GIST features, Saliency maps

Exercises: Experiments with code above

Lecture/slide topics:

Lectures 1 and 2:

- Basic anatomical and functional structure of the feed-forward visual pathway
- HMAX

Lecture 3:

- Deep belief networks
- Scene recognition and gist features

Lecture 4:

-Saliency maps: top-down, bottom-up and contextually guided models
-Use of context for object detection
-Future directions

CHAPTER 7: OBJECT AND SCENE PERCEPTION

7.1 Introduction

The problem of recognizing objects and scenes is a fundamental one for any organism that interacts with the world visually, and humans and other animals are able to solve it remarkably well. We can recognize a huge variety of very complicated stimuli, often after being exposed to them only very briefly. Further, this recognition capability is degraded only slightly by changes in viewpoint and lighting, partial occlusion, and the presence of distracting clutter. The ability of humans to recognize objects and scenes is made all the more remarkable by the difficulty of replicating it in a computer. Computers today can compete with humans on a few tightly circumscribed visual tasks such as detecting faces or cars, but on the general vision problem of learning to recognize arbitrary objects and scenes in arbitrary conditions, they lag far behind.

The difficulty of the computer vision problem is one reason that computational visual neuroscience is a challenging field: if "no holds barred" computer vision has trouble recognizing objects and scenes, then doing so while at the same time accounting for neuroscientific and psychophysical data should be doubly difficult. But the connection to computer vision also makes computational visual neuroscience exciting; successful computer vision approaches can yield hypotheses about how the biological visual systems might work, and neuroscientific breakthroughs can suggest new approaches in computer vision.

It is not always easy to cleanly separate computer vision algorithms from computational models of the human visual system, and the models of scene and object recognition presented in this chapter will draw on both traditions. Reflecting this, the models in this chapter will look to differing degrees to the experimental literature. In some areas, such as the basic function and architecture of the feed-forward part of the ventral visual stream, there is a fairly well established body of experimental facts on which models can build, and for which they should be able to account. In other areas, though, such as the role of feed-back projections in the visual stream, or calculation of detailed contextual relationships between different objects, neuroscience has, as yet, found relatively little. Models of these functions, therefore, of necessity have somewhat more the character of computer vision techniques.

Outline of this Chapter

We have not tried to give an encyclopedic review of the state of the art in object and scene perception. Rather, we have focused on a relatively small number of key models that we believe are illustrative of the range of goals and techniques in the field. We start with hierarchical models of the ventral stream of the visual pathway. These models attempt to explain how visual information is transformed as it passes through the visual cortex, and how the transformations it undergoes eventually represent it in a way that allows objects to be recognized robustly.

The second part of the chapter focuses on scene perception. The term ‘scene perception’ can refer to a variety of processes, from those that deal with an image globally, bypassing object recognition altogether, to those that incorporate both object recognition and contextual reasoning. We begin with the first kind of processes, discussing models of fast scene recognition that represent an image as a collection of global features that capture its rough structure without any explicit representation of individual objects. We then move to models of attention, which are designed to replicate human judgments about which parts of an image are important for various tasks. Last, we present models of how contextual information about the relations between objects and between objects and scene information can be used for object identification. We end by discussing visual routines, which are able to support a variety of visual tasks that go beyond simple object and scene detection and recognition, and which we believe will play an important part in the future of the field.

7.2 Hierarchical Models of The Visual Pathway

In order for a computational or biological visual system to accurately recognize or otherwise process scenes and objects, it must represent its input in the right way. For example, any computer vision system that operates directly on images represented as arrays of pixel intensities will fall prey to the fact that simple transformations like shifts render images unrecognizable. In general, a visual system needs a representation that can cut through superficial factors such as lighting, view-point, and clutter to expose meaningful content like object identities and scene characteristics. This section is concerned with how the brain may build the representation that it uses to process and categorize visual input.

The models in this section are based on the theory that this representation is built up sequentially: after visual information enters the eye, it passes through a number of areas, specifically V1, V2, V4 and IT (figure 1a), each of which transforms the output of the last. After the final transformation, the information is in a form that is useful for recognition and other tasks.

The idea of this hierarchy of transformation areas, and many ideas regarding the details of its workings, comes from a series of classic studies of the cat visual system conducted by David Hubel and Torsten Wiesel in the 1960s. We will briefly discuss what these researchers found in their experiments before presenting the theoretical picture they developed. Later, we will show how Hubel and Wiesel's qualitative model can be specified quantitatively and implemented on a computer.

The first experiment of relevance to this section was conducted in 1962. Hubel and Wiesel recorded the activities of different cells in the area V1 of an anesthetized cat as they projected different stimuli onto a screen [1]. Hubel and Wiesel discovered several distinct kinds of V1 cells, of which two are especially important for our discussion. The first of these types, called *simple*, was maximally responsive to a bar-like stimulus with a particular orientation positioned at a particular location in the visual field. The second type, called *complex*, was also maximally responsive to bars of a particular orientation, but the activities of these cells were more or less invariant to small changes in the bar's position.

In a later experiment, Hubel and Wiesel examined higher visual areas in the cat [2]. In these areas, cells are responsive to stimuli that are more complicated than oriented bars, but Hubel and Wiesel again found cells that had simple and complex attributes, i.e. cells that were responsive to the same sorts of features but that had different levels of tolerance to transformations.

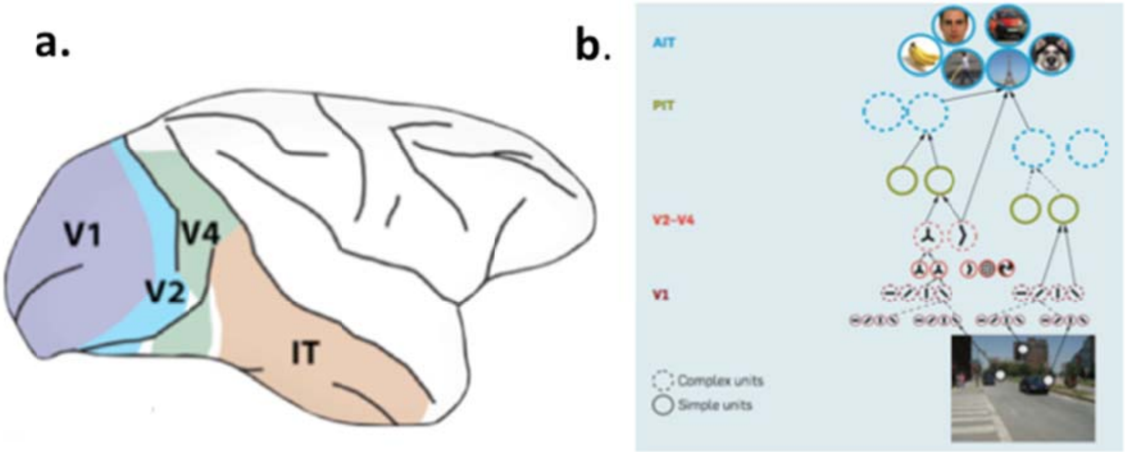


Figure 1. **(**a**)** After visual information leaves the retina and LGN, it passes through areas V1, V2, V4 and IT. Compared to cells in earlier areas, cells in the later areas of this pathway are selective for more complex stimuli, and have response patterns that are more robust to stimulus transformations. (b) A schematic representation of the HMAX model, with rough correspondences to brain regions indicated. Reprinted from [3]; permission not yet obtained.

The description of visual processing that Hubel and Wiesel formulated to account for their findings posits that simple and complex cells have fundamentally different and complementary functional roles, and that these different roles are supported by different methods of processing incoming signals from afferent cells.

We will discuss complex cells first. In the Hubel and Wiesel description, the role of complex cells is to build up *invariance* to transformations of visual stimuli. On a small scale, this invariance enables the complex cells in V1 to respond in the same way to bars of the correct orientation even if these bars are presented at different positions. On a large scale, this invariance accounts for our ability to recognize an object even when the object appears in a position in the visual field in which we have not encountered it before, or when it is rotated or otherwise transformed.

Hubel and Wiesel give a physiological account of how such invariance is achieved. Consider a complex cell in V1 selective for bars oriented at 45 degrees. The theory predicts that the cell receives input from a number of simple cells also responsive to bars at 45 degrees, but with receptive fields at slightly different positions in the visual field. The connections between these simple cells and the complex cell are such that the complex cell will become active even if only a small number of its afferent simple cells are active. Thus, if a bar at 45 degrees is present anywhere in the region covered by the simple cells' receptive fields, it activates at least one of these cells, which in turn activates the complex cell. The same mechanism can also account for invariance under other kinds of transformation: a complex cell may be invariant to any transformation, such as scaling or rotation, as long as it integrates the activities of simple cells tuned to a variety of transformed versions of the stimulus in the way described above.

Hubel and Wiesel had an additional finding that lends support to their hypothesis about the wiring of the complex cell. V1 simple cells' response to shifted versions of the bars at the same orientation are clustered together spatially, making connection to a single complex cell economical.

Now consider simple cells. Higher visual areas are responsive to more and more complicated and specific stimuli, and indeed, at some point in the visual pathway, cells must become responsive to highly specific stimuli such as faces and other objects that we encounter in the real world [4]. The hypothesized role of simple cells is to build this specificity. Simple cells, the theory goes, work in the opposite way from complex cells: their afferents are tuned to a range of different preferred stimuli, and a larger number of these afferents must be active in order for the simple cell to become activated. For example, a simple cell whose afferents are responsive to line segments at a variety of different orientations will become activated only by a stimulus with segments at a large number of these required orientations. Thus, a simple cell can become selective for relatively complex patterns.

Building on Hubel and Wiesel's experimental findings, one may posit that cells of the simple and complex types exist throughout the visual pathway, and that information is processed in an alternating simple-complex-simple order. After several iterations, this process produces a representation that is specific enough to recognize detailed and complicated objects as real-world objects, and robust enough to preserve this recognition from disruption by irrelevant transformations.

Before continuing, we pause to note that descriptions like this one only account for so-called *feed-forward* processing, which occurs as information passes forward through the visual pathway. They ignore processes like selective attention, eye-movements, and other aspects of higher-level vision, which are thought to rely on *feed-back* signals from higher visual areas to lower ones.

7.2.1 The HMAX Model

Hubel and Wiesel's ideas elegantly account for many of their, and other researchers', experimental findings, but lack quantitative detail. In particular, it would be desirable to have a version of the model that is sufficiently tightly specified to implement on a computer. A number of computational models inspired by Hubel and Wiesel's work have appeared over the years; Fukushima's Neocognitron is a pioneering early example [5]. We present here the HMAX model first proposed by Reisenhuber and Poggio in 1999 [6].

Consistent with the Hubel and Wiesel picture, HMAX consists of alternating layers of simple and complex cells, as shown in figure 1b. The model uses 4 layers, corresponding to the four visual areas V1, V2, V4 and IT. As in the Hubel and Wiesel model, each layer except the last has simple and complex cells; the final layer has only simple cells. We will write *Sn* and *Cn* for the collections of simple and complex cells in layer *n.*

The mathematical heart of HMAX is in its specification of the mechanisms that simple and complex cells use to process their inputs. Recall that a complex cell should be activated even if only one of its afferents is active. In HMAX, this is achieved by defining the activity of a complex cell to be the maximum activity of its afferents. Introducing some notation, we can write:

$$c_i = max_{j \in N(i)} S_j$$

where c_i is the activity of a complex cell, and $s_j, j \in N(i)$ are the activities of the simple cells that connect to it.

The specificity-building connections are encoded by a linear summation of the form

$$s_i = \exp \frac{-1}{2\sigma^2} \sum_{j \in N(i)} (w_j - c_j)^2$$

Here, s_i is the activity of a simple cell, $c_j, j \in N(i)$ are the activities of its afferent complex cells and the w_j are synaptic weights. This equation can be thought of as encoding a template-matching operation: the weights *w* specify a preferred prototype or template stimulus, and the simple cell's firing depends on how similar the activities of its afferents are to this prototype. In a process described in more detail below, the weights are learned in an unsupervised way from a database of training images.

HMAX Details

An actual implementation of the HMAX model requires a number of additional choices about the structure of the model and the values of its parameters. Fortunately, many of these choices can be made in a principled way on the basis of experimental data. We give details layer by layer. In order to balance completeness and simplicity, the model we describe combines features of the simple model described in [7] and the more updated and detailed model described in [8].

Layer 1.

S1 Units: The *S1* units are the first to process the image. *S1* is the only group of simple cells that does not use the transfer function in equation (2). Rather, *S1* cells are hand-tuned to replicate the properties of V1 cells. Like V1 cells, *S1* units respond to linear stimuli, and their response properties are dictated by the orientation of these stimuli. In addition, each *S1* unit has a scale parameter that determines the size of its receptive field and the size of the stimuli that will activate optimally. Recent versions of HMAX also use phase information, but for expositional simplicity, we ignore this here. There are 16 different scales, from 7 x 7 pixels to 37 x 37 pixels, and four different orientations, 0°, 45°, 90°, and 135°. This gives a total of 64 different types of *S1* units. For a fixed scale, *S1* units pack the visual field as densely as possible; *S1* RFs overlap maximally. Thus, there are more units at smaller scales.

As discussed, *C1* cells detect a stimulus of a particular orientation at a range of positions and scales. Thus, the activity of a *C1* cell is determined by the maximum activity of a collection of *S1* cells, all with the same preferred orientation, and with similar but not identical preferred positions and scales. In [7] *C1* units pool over two adjacent scales; the *C1* units pool over different numbers of *S1* units depending on the two scales for which they are selective. In [7], *C1* units pool over 8 x 8 to 22 x 22 grids of *S1* cells.

Unlike *S1* cells whose receptive fields overlap maximally, *C1* cells only overlap to the extent that a given pixel in the middle of the image is covered by the receptive fields of two *C1* cells. This is actually a key property of the model that is replicated at other layers: C cells subsample S layers, so that at a given layer there are many fewer C cells than S cells. This keeps the size of the network from exploding as layers are added, and is part of the reason that HMAX uses an alternating C and S architecture. From a computational point of view, this architecture might seem odd since it builds up invariance at the level of features or parts of objects rather than at the level of whole objects; it might seem desirable to have many levels of S cells with C cells used only at the top of the network. But in this architecture, the number of units would grow exponentially with the number of simple cell layers used; C layers must be interleaved to prevent this from happening. Put another way, top-layer units should be tuned to things like big pieces of real world objects. There are many more of these stimuli than there are of the simple oriented line segments to which lower-level cells respond, and the number of neurons required to recognize them at all positions and scales would be prohibitively large. Therefore, invariance must be built in earlier, via intermediate layers of C cells.

Layer 2.

S2: *S2* cells are the first layer to use the pooling transfer function defined by equation (2). The afferents of an *S2* cell are a cluster of *C1* cells that are tuned to different orientations; [7] used 10 *C1* cells. Importantly, the weights in equation (2) that define the templates that *S2* cells prefer are learned in an unsupervised way from a database of natural images. The intuition is that during development a visual system should become tuned in such a way that it adapts to the statistics of the visual world: it should develop detectors that it encounters frequently. In HMAX, this goal is achieved via an extremely simple sampling procedure: images from a database are presented sequentially, and at each presentation one *S2* unit is selected at random and its weights are set to the responses of its afferents to the current image; these responses define the template to which it is tuned. In order to preserve the structure of the model, the number of templates and scales (and hence positions) to be used is decided ahead of time, and each time a template imprints a unit, it also imprints a complete set of other units at all positions and at all decided-upon scales.

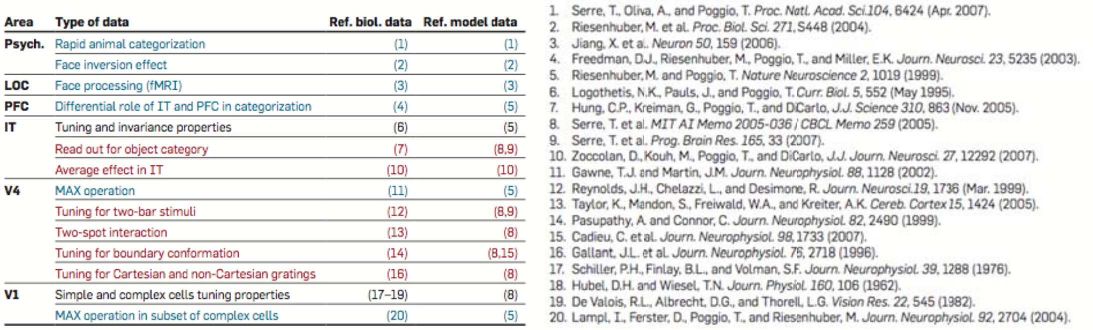
C2: *C2* units work the same way as *C1* units. The only difference is that they have larger receptive fields, that is, they pool over more S2 positions.

Layers 3 and 4.

Conceptually, *C3*, *S3* and *S4* units work the same way as analogous units in the previous layer. The learning for the S units is similar too. In order to make the learning procedure well defined, it is assumed to occur sequentially; the weights of earlier S units are assumed to be set first.

Experimental Comparisons

One of the main reasons that computational models like HMAX are useful is that they allow us to compare theories about the visual system with quantitative experimental data. A number of studies have compared HMAX with experiments; see table 1. Here we briefly discuss two studies, one showing that a computer vision system that uses the representation computed by HMAX has similar performance to humans on a rapid recognition task, and one giving experimental evidence that complex cells do in fact integrate their inputs using the max operation.



Area	Type of data	Ref. biol. data	Ref. model data
Psych.	Rapid animal categorization	(1)	(1)
	Face inversion effect	(2)	(2)
LOC	Face processing (fMRI)	(3)	(3)
PFC	Differential role of IT and PFC in categorization	(4)	(5)
IT	Tuning and invariance properties	(6)	(5)
	Read out for object category	(7)	(8,9)
	Average effect in IT	(10)	(10)
V4	MAX operation	(11)	(5)
	Tuning for two-bar stimuli	(12)	(8,9)
	Two-spot interaction	(13)	(8)
	Tuning for boundary conformation	(14)	(8,15)
	Tuning for Cartesian and non-Cartesian gratings	(16)	(8)
V1	Simple and complex cells tuning properties	(17–19)	(8)
	MAX operation in subset of complex cells	(20)	(5)

1. Serre, T., Oliva, A., and Poggio, T. *Proc. Natl. Acad. Sci.104*, 6424 (Apr. 2007).
2. Riesenhuber, M. et al. *Proc. Biol. Sci. 271*, S448 (2004).
3. Jiang, X. et al.. *Neuron 50*, 159 (2006).
4. Freedman, D.J., Riesenhuber, M., Poggio, T., and Miller, E.K. *Journ. Neurosci. 23*, 5235 (2003).
5. Riesenhuber, M. and Poggio, T. *Nature Neuroscience 2*, 1019 (1999).
6. Logothetis, N.K., Pauls, J., and Poggio, T. *Curr. Biol. 5*, 552 (May 1995).
7. Hung, C.P., Kreiman, G., Poggio, T., and DiCarlo, J.J. *Science 310*, 863 (Nov. 2005).
8. Serre, T. et al. *MIT AI Memo 2005-036 / CBCL Memo 259* (2005).
9. Serre, T. et al. *Prog. Brain Res. 165*, 33 (2007).
10. Zoccolan, D., Kouh, M., Poggio, T., and DiCarlo, J.J. *Journ. Neurosci. 27*, 12292 (2007).
11. Gawne, T.J. and Martin, J.M. *Journ. Neurophysiol. 88*, 1128 (2002).
12. Reynolds, J.H., Chelazzi, L., and Desimone, R. *Journ. Neurosci.19*, 1736 (Mar. 1999).
13. Taylor, K., Mandon, S., Freiwald, W.A., and Kreiter, A.K. *Cereb. Cortex 15*, 1424 (2005).
14. Pasupathy, A. and Connor, C. *Journ. Neurophysiol. 82*, 2490 (1999).
15. Cadieu, C. et al. *Journ. Neurophysiol. 98*, 1733 (2007).
16. Gallant, J.L. et al. *Journ. Neurophysiol. 76*, 2718 (1996).
17. Schiller, P.H., Finlay, B.L., and Volman, S.F. *Journ. Neurophysiol. 39*, 1288 (1976).
18. Hubel, D.H. and Wiesel, T.N. *Journ. Physiol. 160*, 106 (1962).
19. De Valois, R.L., Albrecht, D.G., and Thorell, L.G. *Vision Res. 22*, 545 (1982).
20. Lampl, I., Ferster, D., Poggio, T., and Riesenhuber, M. *Journ. Neurophysiol. 92*, 2704 (2004).

Table 1: A compilation of experimental findings pertaining to HMAX. The studies in blue experimentally verified predictions made by the model, the studies in red are consistent with the model, and the studies in black were used to set the model’s parameters. Reprinted from [3]; permission not yet obtained.

HMAX recognition

It is natural to try to validate models like HMAX by testing whether they can be used in a computer vision system to deliver human-level performance at recognition tasks. The main problem with tests like this is that, as we have discussed, HMAX only models feed-forward processing, while humans can use feed-back mechanisms as well. To overcome this problem, Serre et al. [8] used a recognition task – determining whether or not an animal was present in a picture – with extremely short stimulus presentations of only 20ms. Previous experiments [9] have shown that feed-forward processing occurs before feed-back processing begins, so that given limited time, only feed-forward mechanisms are used. In the Serre et al. experiment, a random noise “mask” stimulus was shown after the image, since masks are thought to disrupt any feed-back processing that might occur after the stimulus is removed. The results closely matched human performance, both in terms of overall accuracy in the kinds of images that were easy and difficult. See [8] for details.

The max operation.

Given that the max operation predicted by HMAX for complex cells differs from the standard integrate-and-fire neuron model, it is worth briefly presenting some experimental evidence for it. In [10], Lampl et al. tested the responses of V1 complex cells to stimuli presented in pairs and one at a time. The authors found that the complex cells’ behavior was consistent with a max rule – the magnitude of their responses to a pair of stimuli was very close to the larger of the two responses to the individual stimuli. In addition, [11] proposes biophysically plausible circuits that may implement the max operation.

7.2.2 Deep Belief Networks

Deep belief networks (DBNs) [12] [13] are another hierarchical neural network model that seeks to find good representations for object recognition. DBNs as presented here are a comparatively young family of models, and to date there has not been an extensive effort to work out the details of how they might be implemented in the brain. Therefore, compared to, for example, HMAX, they make less contact with the neurophysiological and psychophysical literature. Nevertheless, they raise several interesting issues, for example about what exactly a visual system should try to learn during development, and the role of feedback connections in the visual pathway that are important to consider when thinking out computation in the visual system. In addition, they have demonstrated impressive performance on a number of visual tasks. A successful model of the visual system has to accomplish the twin goals of effectively processing objects and scenes and accounting for experimental findings, and the ability to accomplish the first of these is not to be discounted.

As we discussed above, the purpose of hierarchical models of the visual system is to compute a representation of their input that will support tasks like recognition. Another way of saying this is that hierarchical models try to discover a collection of latent factors, represented by their top-level units, which capture the relevant structure of the images in their training set. One way to determine whether a model has extracted the right latent features is to see if the features it produces give good results when fed to a classification algorithm; this is a *discriminative* criterion. DBNs are based on the idea that a better criterion is *generative*: a set of latent features is good if it can be used to generate plausible members of the training set. In this sense, DBNs are similar to algorithms like PCA that seek to find an economical representation of a dataset from which the dataset can be reconstructed with minimal error. Unlike PCA, though, DBNs are capable of discovering latent factors that are related in very complicated and non-linear ways to their input data.

There are several motivations for the generative approach. First, class labels typically carry very little information. For example, the labels in a binary classification task only contain one bit. Images themselves, though, are very rich in information and are thus much better able to constrain the parameters in a complicated model. Second, discriminative training seems biologically improbable: animals are not given databases of labeled images. Last, generative training is advantageous from a practical point of view; the unlabeled data needed to train a generative model are much easier to obtain than are large sets of labeled data.

DBNs, then, are neural networks that learn a set of synaptic weights that enable them to generate probable members of the set of images on which they are trained. This generation is a top-down processing. In the context of our earlier discussion, though, it is not clear that such a network would be useful for the recognition tasks that we are interested in, given that these tasks require bottom-up processing. However, we might hypothesize that the series of transformations that take a set of latent feature values to a realistic training image are symmetric to the set of transformations that transform an image into its latent representation, so that top-down and bottom-up mechanisms can be learned simultaneously. With this hypothesis in mind, we learn a network that can do both top-down generation and bottom-up analysis, but require it to use the same weights for both forms of processing, thereby enforcing symmetry. Such a network is shown in figure 2a. It turns out that the hypothesis that the top-down and bottom-up processes should be exactly symmetric is not quite true, but the defects in the resulting model are small and can be corrected with a post-processing step that separates the top-down and bottom-up weights and fine-tunes them independently of each other.

How is such a network to be learned? First consider a simple bipartite network with one visible layer and one hidden layer, like the one shown in figure 2b. Such a network is called a *restricted Boltzmann machine (RBM)*. Each of the units in an RBM is binary and stochastic: the probability of a unit turning on (i.e. taking the value 1) is determined by the number of its afferents that are on, according to the following rule:

$$P(u_i = 1) = \frac{1}{1 + \exp(-b_i - \sum_{j \in N(i)} w_{ij} u_j)}$$

where w_{ij} is the synaptic weight between units i and j and b_i is a so-called bias term. The goal of learning will be to find good values for the weights and biases.

Roughly, the learning will proceed by initializing the weights with some values, using them to reconstruct a training image, adjusting the parameters to make the reconstruction and the original more similar and then repeating this process iteratively. So the first step is to explain how an RBM can reconstruct an image. First, note that equation (3) defines *an energy function* for the RBM that assigns high energy to states that are improbable:

$$E = -\sum_i b_i u_i - \sum_{i,j} w_{i,j} u_i u_j$$

The reconstruction, then, can be accomplished by encoding an image in the visible units of the RBM (as with HMAX, the visible units correspond to pixels) and then letting neurons interact with each other according to equation (3) until the network has settled into a low-energy state. At this point, the reconstruction is simply given by the activities of the visible units. The idea here is similar to the dynamics of a Hopfield network (chapter xx).

This reconstruction procedure is correct, but it is also very time-consuming because of the time required for the network to reach a low-energy state. Therefore, in practice, the reconstruction is usually done with only a single up-down-up pass, that is by inputting a training image, using it to set the activities of the hidden units according to equation (3), resetting visible units using these activities, and then resetting the hidden units once more.

Repeating this process for a set of training images gives a collection of image-reconstruction pairs that can be used to update the weights of the network. This update is done according the following rule:

$$w_{ij} \leftarrow w_{ij} + \epsilon(< u_i u_j >_{original} - < u_i u_j >_{recon})$$

Here, is the fraction over the set of training images used that units *i* and *j* are on together, and is the corresponding quantity for the reconstruction; is a learning rate.

This rule can be argued for formally, but we will just motivate it intuitively. If w_{ij} is large, it will encourage units *i* and *j* to take the same value. So if they take the same value more often in the reconstruction than in the original image, the weight should be decreased to bring the reconstruction and original image closer together. Similarly, the weight between two units should be increased if the units take the same value too *in*frequently in the reconstruction. The overall training process for an RBM consists of dividing the training set into several batches, and using the batches sequentially to update the RBM weights according to equation (4).

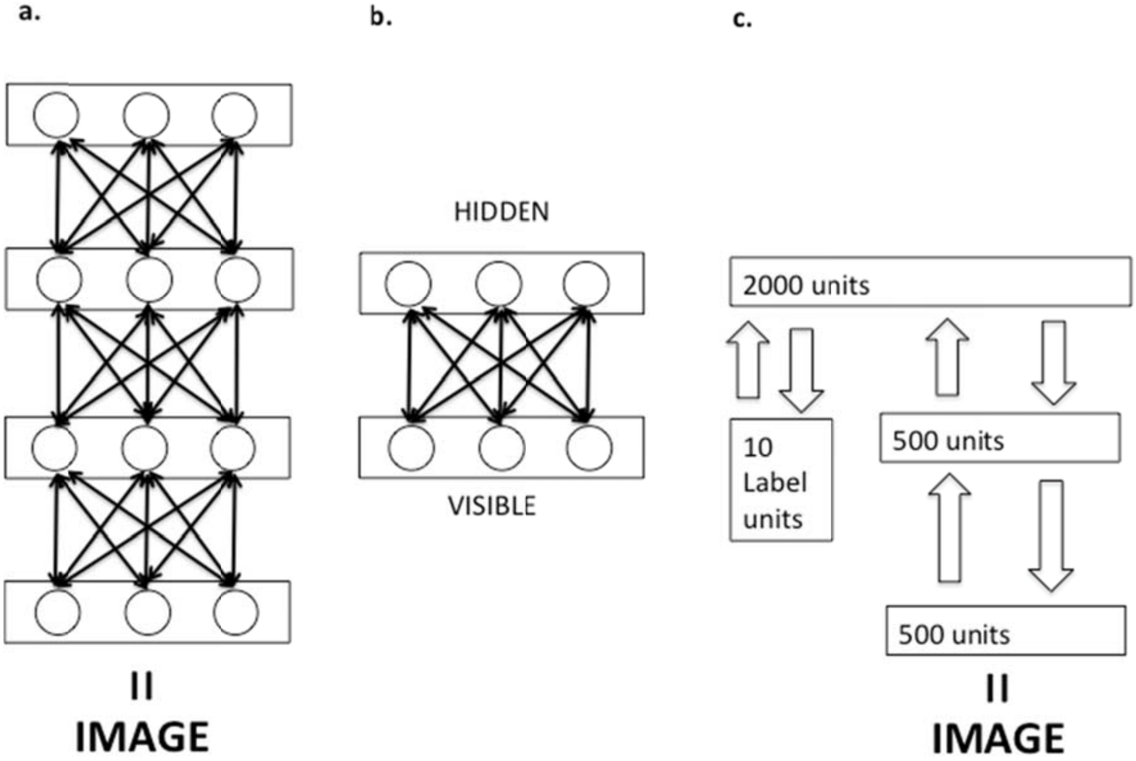


Figure 2: (a) A deep belief network with four layers. The bidirectional arrows indicate that the same weights are used for top-down and bottom-up processing. (b) A restricted Boltzmann machine (RBM). (c) The architecture used for handwritten digit classification. The top-down and bottom-up weights have been fine-tuned separately and are no longer equal.

We now consider training a deeper network, with more than two layers. It turns out that the best way to make this fairly daunting task manageable is to modularize the larger network, treating it as a stack of two-layer networks and repeating the simple training procedure above. This is done in a greedy way, working from the bottom of the network to the top. First, the bottom two layers of the network are trained as an RBM in the way described, yielding a set of weights between these two layers that will remain fixed for the remainder of the training process. The weights between the second and third layers are treated next. We cannot use exactly the same procedure for these weights; since layer two is not adjacent to the input layer, training images can't be fed in directly. Rather, training images are fed through the first RBM using the weights that have already been learned, and the output of this process is used as the input for the new RBM. This process can then be iterated to learn as many layers as are desired.

The output of this training procedure is a network of some chosen depth that can pass information either top-down or bottom-up using the same set of weights for each task. To correct for the fact that the optimal top-down and bottom-up weights are not exactly the same, the two sets of weights can be untied and fine-tuned separately. The bottom-up weights can be fine-tuned for discrimination using back-propagation, and the top-down weights can be fine-tuned for generation using the wake-sleep algorithm [14].

Experimental Results

An early successful application of DBNs was the use of the architecture in figure 2c to improve upon state-of-the-art results on a hand-written digit recognition task [13]. One of the appealing features of DBNs is that they are not limited to processing visual data. For example, they have successfully been applied to problems in audition[13] and document compression [14], among others. As for neuroscience, it has been argued that a variant of DBNs that in the activities in each layer are encouraged to be sparse replicates properties of V2 cells [17]. Also, a class of models called convolutional deep belief networks incorporate into DBNs some features of models like HMAX which allow them to display invariance to image transformations.

7.2.3 Other models

Of the large number of other models of the visual pathway and object recognition, we briefly mention here two. In [18], Lee and Mumford hypothesize that the visual cortex does a Bayesian inference process, in which higher and lower visual areas use feed-forward and feed-back connections to adjust each other's estimates of stimulus properties. In [19] Irving Biederman proposes a very different picture of object perception in which objects are recognized by inferring their composition in terms of a collection of primitive geometric parts called geons.

7.3 Scene Perception

It is perhaps natural to assume that a scene is just a collection of objects, and that perception or recognition of a scene is nothing other than the perception or recognition of these objects. This naïve view is far from the truth; scene perception can involve processes that are both less and more detailed than repeated object perception. First, there is evidence that humans can recognize many important properties of a scene very quickly and without explicitly recognizing many of the objects present; section 7.3.1 presents a model of this phenomenon. Second, not all of the objects or regions in an image of a scene are equally important for a given task, and in order to avoid wasteful computation, a visual system needs mechanisms to find and focus on those areas of an image that are most salient. In 7.3.2, we review three computational models of how this may be accomplished. Next, even viewed as a collection objects, natural scenes are at least *organized* collections; objects occur in them in fairly predictable patterns. In section 7.3.3 we show how these contextual regularities can be used to improve object detection. Last, section 7.3.4 finishes by speculating that future models of scene perception will benefit from incorporating sequences of visual actions called visual routines.

7.3.1 Fast Recognition and the gist of a scene

It has been known since at least the 1970s that people can extract a great deal of semantic information about a scene over the course of an extremely brief viewing period. In [15], Molly Potter showed subjects a sequence of images, and found that they were able to identify pictures matching a previously given title (e.g. girl holding a pie) even when the viewing period was as short as 125 ms. Further, subjects were also able to quickly identify pictures *not* matching a given semantic description, which argues that the first result is not simply an effect of priming. As discussed above, it is believed that perception on very short timescales uses feed-forward mechanisms almost exclusively [9]. Thus, Potter's subjects were not able to deploy eye movements or shifts in selective attention, so their scene identifications were mostly informed by *global* information about the whole image, rather than more detailed information about attended-to parts.

The speed of processing also implies that the subjects were probably not using detailed object recognition. This is not inconsistent with their high recognition performance: experiments done by Schyns and Oliva and others show that humans are able to accurately recognize scenes even from images that have been blurred to the extent that objects appear only as oriented blobs [20].

These findings suggest that a global, object-free representation could also be used in computational scene recognition models, and several approaches along these lines have been developed. One particularly well-known model, developed by Oliva and Torralba in [21], and reviewed in an accessible way in [22], also has the advantage that it is computed with neurologically plausible V1-like mechanisms. This is the so-called *gist feature* model.

Like the *S1* activations in the HMAX model, gist features are computed by applying filters responsive to linear stimuli at a range of scales and orientations at a dense collection of locations throughout the input image. The output of this collection of filters is extremely high dimensional, so to make further processing tractable it undergoes two dimensionality reductions steps. First, the image is divided into an N x N grid, and for each scale and orientation, the filter outputs are averaged within grid cells. This leaves a total of N x N x S x R filter outputs, where S is the number of scales used and R is the number of orientations. The second dimensionality reduction step is just PCA: the downsampled filters are applied to a large collection of images and the first principle components are retained as a basis onto which the outputs for a new image can be projected.

Although representation by gist features entails some loss of information, the information retained contains the kind of large-scale structure that is important for scene-level analysis and recognition. In one study, a classifier operating on gist features was able to identify scenes of 15 possible classes with 75% accuracy [23]. Further, gist features have been successfully used to incorporate scene-level information into vision algorithms for a variety of tasks, such as object detection [24], and attention modeling [25]. We will discuss some of these models in later sections.

An additional way to get insight into what information gist features carry is to specify a target image and then coerce a noise image to share its gist features, as is done in figure 3. As the figure shows, the amount of information that the gist features carry depends on the size of the grid used to downsample the filter outputs, but even for fairly coarse grids, gist can capture much of the target image's key properties.

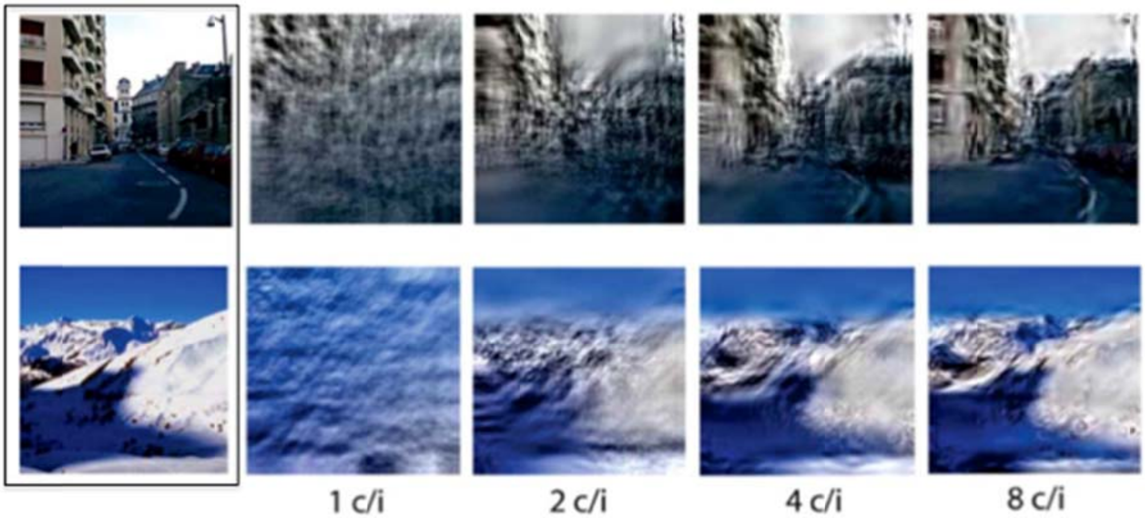


Figure 3: The images on the right were produced by iteratively modifying a noise image until it came to have the same gist features as a target image. Moving from left to right, the gist features were obtained by averaging the filter outputs over finer grids, from 2 x 2 cells to 16 x 16 cells. Reprinted from [22]; permission not yet obtained.

The techniques available for scene-level image modeling are not limited to gist. We do not have space to review other models here, but see [26] for a comparison of a number of approaches.

7.3.2 Modeling Attention and Saliency

In the previous section, we argued that scene perception on very fast timescales can be accomplished with global features that do not preferentially process specific objects or regions. While global models perform tasks like scene recognition well, they are clearly not suited for many other tasks, such as locating a small target object. In general, it is extremely inefficient to do the kind of detailed processing required for these tasks on the image as a whole. Rather, detailed processing should be preferentially deployed on regions that are likely to be relevant. These regions are called *salient*, and finding salient parts of an image is the job of visual *attention*.

In this section, we review several models of visual attention. The output of all of these models is a *saliency map*, which assigns to each point or region of an image an estimate of its saliency; see figure 4 for an example. Whether or not such saliency maps are explicitly represented in the brain is a matter of debate, but models like the ones in this section can still be compared to human data in a number of ways. For example, they can be used to predict where in an image human subjects will look, and these predictions can be compared to data acquired with an eye tracker.

The first model we present attempts to predict attention patterns in the absence of any explicit task specification. In effect, this model just finds regions that stand out from their surroundings. It turns out, though, that doing this effectively is a more complicated business than it may first appear to be. The second and third models assume a search task in which a target object is to be located among distractor objects or in a natural scene. The second model assumes that the appearance of the target object or of the distractors is known ahead of time, while the third model assumes that only the class (e.g. “car” or “painting”) of the target is known. This third model uses scene-level information in the form of gist features to suggest probable target locations.

Bottom-Up Attention.

The idea of this first and most basic model is that in the absence of any specific task specification, people tend to look at parts of an image that differ significantly from their neighboring parts. For example, people will look at an airplane rather than at a patch of the uniform blue sky that surrounds it. This idea has been the basis for a number of models, starting with the 1985 model of Koch and Ullman [27]. These models are called “bottom up” attention models, because attention is guided exclusively by the stimulus: no task-specific control is used.

The Koch and Ullman model assumes that saliency arises from local differences in various low-level features, such as color, luminance, and orientation.

The model’s first processing step, therefore, is to extract these local differences by using arrays of units with center-surround antagonism at a number of scales. The units

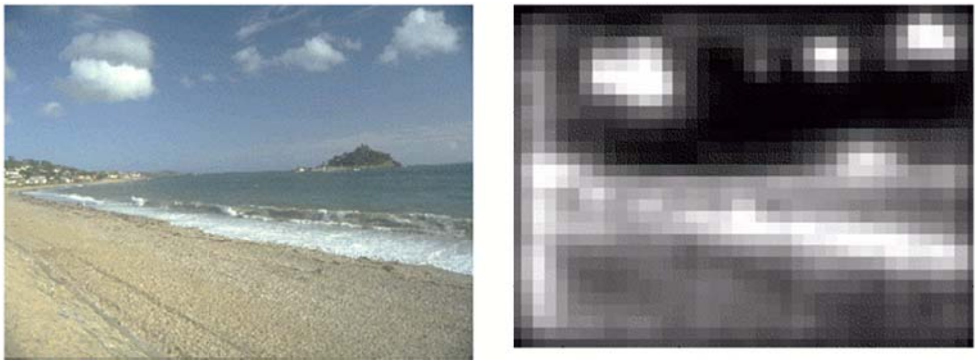


Figure 4: An example of a saliency map. Light colored regions in the right-hand image indicate salient regions in the left-hand image. Note that salient regions differ from their surroundings, and that uniform regions like the sky are not salient. Reprinted; permission not yet obtained.

corresponding to a particular feature are organized retinotopically and are said to constitute a *conspicuity map*. A cell at a particular point in, say, the luminance map will be active if the luminance changes at the corresponding point in the image. All the conspicuity maps are then combined to form an overall saliency map for the whole image. Like the conspicuity maps, the saliency map is instantiated as a collection of retinotopically organized neurons; a cell at a given location in the saliency map receives inputs from the cells at the corresponding point in all the conspicuity maps. Thus, a cell at a given location in the saliency map will be active if that point is sufficiently conspicuous in the maps for all the low-level features.

The last component of the model is a mechanism to translate a saliency map into a pattern of saccades. The idea is fairly simple. First, the output of the saliency map is passed to a winner-take-all network that finds its most active neuron, corresponding to the most salient location. The first saccade is then targeted at this location. After the first saccade, an inhibition of return mechanism is applied to reduce the activity at its target point, ensuring that this target is not selected again for subsequent saccades. Later saccades are chosen in the same way.

The main point at which this simple model calls for elaboration concerns the way in which the conspicuity maps are combined to form the saliency map. In particular, there is no principled way to weight the contributions of different features to overall saliency. One would not want to say, for example, that a change in luminance of a given magnitude is three times as important as a change of the same size in color. Further, since the low-level features are not measured on the same scale, it does not make sense to simply assign them all equal importance. These considerations would seem to rule out any scheme in which a unit in a saliency map combines its inputs linearly. Perhaps the most obvious way to overcome this problem is to normalize the activities in all conspicuity maps, bringing them all to the same dynamic range. However, this simple scheme has been shown not to work well in practice [28], and an alternative solution that is both more effective and more neurologically well-motivated was proposed by Itti and Koch in [29].

To see the intuition behind this approach, consider the stimuli in figure 5a. The fact that the central circle stands out as salient in the left image but not in the right one illustrates the general principle that features that change abruptly at many points in the image are less able to confer saliency than ones that change only rarely. This suggests a scheme in which activity at one point in a conspicuity map can inhibit to some extent activity at nearby points. This inhibition would produce an overall suppression of activity in conspicuity maps with many active regions, such as the red map in the right image in figure 5a, while leaving more or less unchanged maps that have a few isolated points of activity, such as the red map in the left image in figure 5a.

The Itti and Koch model implements this idea by first normalizing each conspicuity map as a preprocessing step, and then applying ten iterations of the following three steps to each of the conspicuity maps.

(1) The map is added to its convolution with a two-dimensional difference of Gaussians (DoG) filter, like the one shown in figure 5b. This filter implements a kind of center-surround antagonism: the broad inhibitory region causes any peak of the conspicuity map that neighbors another peak to be inhibited. This step corresponds to an experimentally

observed phenomenon called non-classical inhibition in which the firing of a cell can be altered by stimuli that do not actually lie in its receptive field.

(2) If there are regions of the conspicuity map in which firing is more or less constant, the excitation and inhibition from the DoG filter will more or less cancel, and little or no overall suppression will occur. To account for the intuition that constant regions should not be salient, a small constant term is subtracted from the output of step one.

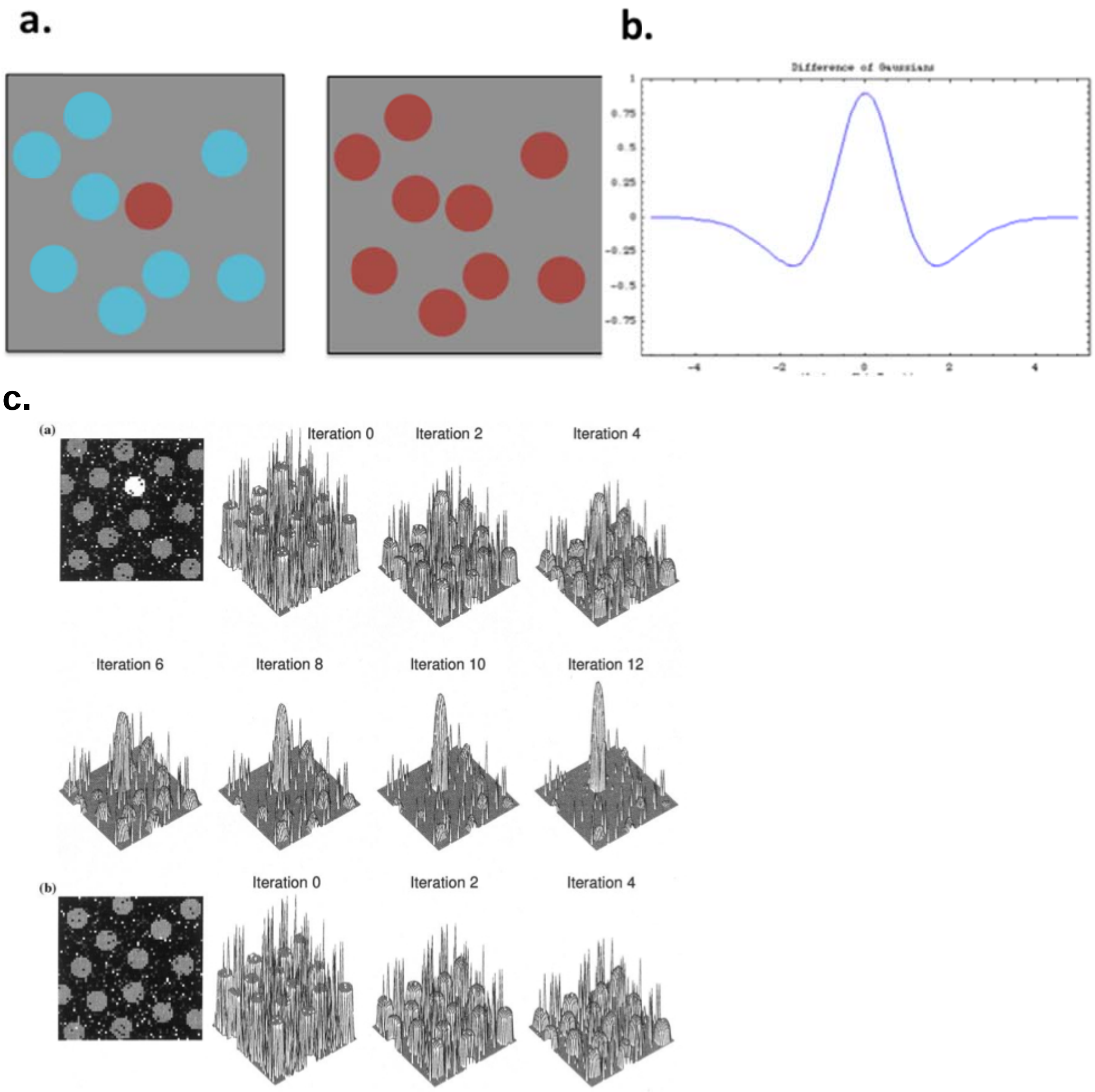


Figure 5: (a) The central circle is salient when it is the only region that has a large local change in the red channel. (b) A difference of Gaussians (DoG) filter. (c) Several iterations of the Itti and Koch nonlinear normalization process. Part (c) is reprinted from [29]; permission not yet obtained.

(3) Since the values of the maps are represented by neural firing rates, negative values are nonsensical. So if any of the values resulting from the first two steps are negatives, these values are set to zero.

An example of this iterative process is shown in figure 5c.

After the ten iterations of these steps have been completed, the resulting conspicuity maps are simply summed to form the final saliency map. Subsequently, saccades can be generated with the winner-take-all and inhibition of return mechanisms, as described earlier.

Itti and Koch test their model in two ways. First, they show that it reproduces perhaps the best known experimental finding about visual search, namely that so-called pop-out search in which a single attribute distinguishes the target from the distractors (e.g. “find the red bar”) can be accomplished in an amount of time that is more or less independent of the number of distractors in the image, while the time it takes to find targets distinguished by a conjunction of features (e.g. “find the red vertical bar”) is proportional to the number of distracting objects.

Second, they use their system to find small target objects in high-resolution images of complex real world scenes. Specifically, the system tries to locate military vehicles against natural backgrounds. Since the system is not given any knowledge of the target’s appearance, it cannot identify the target explicitly. Rather, the system chooses a succession of regions on which to fixate, and it is said to have found the target if the target is in one of these fixated regions. Especially given that it had no model of the target’s appearance, the system was quite successful: it found the target on the first fixation in 93% percent of images, and found it in the first twenty fixations in 68%.

Using accepted values for the length of time it takes a human to make a new fixation, Itti and Koch were able to estimate how long it would take a human to find the target using eye movements controlled by the model. Interestingly, they found that these estimates were considerably faster than the actual human search times. They hypothesized that this discrepancy was due to human’s use of scene-level assumptions, such as that a vehicle is likely to be found on a road, that were often wrong in the experimental dataset. We will describe later how scene-level information can be incorporated into computational attention models.

Top-Down Influences on Attention

Consider again the task of finding a vehicle in a natural scene that Koch and Itti used to evaluate their model. As discussed, this system simply looks for regions that contrast with their neighbors and does not use any knowledge about what the vehicle looks like. This would seem to be a deficit compared to human vision; this section shows how this deficit can be corrected by incorporating prior knowledge of the appearance of the target or distractors.

There are a number of models that do this; we describe a model by Navalpakkam and Itti [30]; see [31] for an alternative point of view. In the Navalpakkam and Itti model, the final saliency map is the product of two stages of integration. First, features of the same type are integrated into *feature dimensions,* which are used to create conspicuity maps in the same way as above. For example, the red and green features would both be part of the "color" feature dimension. Second, the conspicuity maps are integrated to form the saliency map.

The Navalpakkam and Itti model allows the features or feature dimensions that are known a priori to be diagnostic of the target to be weighted to make a larger contribution to conspicuity or saliency. For example, if the target is known to be red, it makes sense for local changes in the red channel to influence saliency disproportionally. The weights that accomplish this are called *top-down gains*.

The first step in the model is to define what we can call *sub-conspicuity maps* σ_{ij} , where j specifies a low-level feature and i specifies a feature dimension. For each location (x,y), $\sigma_{ij}(x,y)$ is how conspicuous the location (x,y) is according to feature j alone. These sub-conspicuity maps are formed using the same iterative nonlinear normalization process that was used in the Itti and Koch model.

Next, top-down gains are used to form conspicuity maps as weighted sums of sub-conspicuity maps. For each feature dimension i, the conspicuity map is defined to be

$$s_i(x,y) = \sum_j g_{ij}\sigma_{ij}(x,y)$$

The weights g_{ij} are called *low-level gains*. Next, the overall saliency map is formed analogously, using a set of *high-level gains*:

$$S(x,y) \ = \sum_i G_i s_i(x,y)$$

The main machinery of the model is devoted to determining values for these two sets of gains to make optimal use of prior knowledge to increase search speed. To formalize this problem, suppose we are given probability distributions over the sets of features that define the appearances of the target and distractor objects. This information might come from explicit verbal instruction, or, if the search task is repeated, it might be learned from experience. Then define three *signal to noise ratios (SNRs).* For the jth feature in the ith feature dimension, SNR_{ij} is the ratio of the expected saliency of the target and the expected saliency of the distractors, when only feature j is considered. Similarly, for each feature dimension i, SNR_i is the ratio of the expected saliency of the target and distractors, according to just feature dimension i. Last, the overall signal to noise ratio SNR is the ratio of the expected saliency and distractors when all features and all dimensions are used. All expectations are with respect to the distributions that define the features of the target and distractors, and to random neural noise.

The goal, then, is to find the gains that maximize the overall SNR. If some assumptions are made, for example that the gains for all features must sum to a constant, then the optimal gains can be found analytically. However, the result is quite intuitive, so we skip the derivation. The optimal low-level gains are given by

$$g_{ij} = \frac{SNR_{ij}}{\frac{1}{n}\sum_{k=1}^{n} SNR_{jk}}$$

Thus, large low-level gains are assigned to features that have better-than-average ability to distinguish between the target and the distractors. Similarly, the optimal high-level gains are

$$G_i = \frac{SNR_i}{\frac{1}{N}\sum_{k=1}^{N} SNR_k}$$

As was done for the previous model, Navalpakkam and Itti first validated their model by replicating results from the psychophysics literature. For example, they found that pop-out search can be speeded up by priming, i.e. pop-out search is faster when the feature that distinguishes the target from the distractors is known ahead of time. For an account of several other psychophysics-style experiments, see the paper [30].

To test the performance of the model on natural scenes, Navalpakkam and Itti, used ten training images in which a target object and a set of distractor objects appeared in different configurations to learn the required appearance distributions. These distributions were used to compute the top-down gains using the two equations given above, and the model with these gains was applied to test images. The results compared quite favorably to a base model like Itti and Koch's that does not use top-down information.

Incorporating Scene Information into Visual Search

The model in the previous section shows how knowledge of the visual features of the target and distractors can guide visual search. It also seems reasonable for search in natural images to be guided by scene-level information. For example, if we are looking for a car, we should focus our attention on areas where it is reasonable for a car to be, say on a road rather than in the sky. In addition, as we saw in section 7.3.1, scene-level information can be extracted very quickly in a bottom-up way, so it is physiologically reasonable to assume that this knowledge is incorporated into subsequent top-down attentional control.

One formal model of how to use global information to improve saliency estimates is given in [16]. In this model, the saliency of a location is determined by both local and global factors. Although formalized differently, the local saliency model is philosophically quite similar to the Itti and Koch model, so we do not discuss it here. Although this model does not assume that the target's visual features are known ahead of time, it does assume knowledge of the class of the target object (e.g. "car" or "painting"). Because objects of certain classes occur in predictable locations in given scenes, gist features can be used to assign higher saliency to regions that are likely to contain a target. Formally, the contribution of the gist features G to that saliency of a location X is given by $P(Y|O = 1, G)$. Here, $O = 1$ simply encodes the assumption that a target object is present somewhere in the image; we will drop this notation in the future. There is one important caveat to mention before moving on: it turns out that global information is only informative about the vertical locations of the objects within a scene, so here the location Y will only consist of a y-coordinate. The model is shown schematically in figure 6.

This conditional distribution can be derived by starting with the joint distribution $P(Y, G)$. The relationship between global features and locations is mediated by a collection of *scene prototypes* r_t, $t = 1 \ldots M$. Heuristically, the idea is that the raw gist features of an image can be used to define a distribution over scene categories, which although not actually named can be thought of as things like street scenes, mountain scenes, etc., and then these categories define a distribution over the target location. The model makes two mathematical assumptions:

(1) Within each prototype, the distribution of gist features is Gaussian

(2) The distribution over target locations given gist features is again Gaussian, with a mean that depends linearly on the gist features.

These assumptions give the following decomposition:

$$P(X,G) = \sum_{t=1}^{M} P(r_t)P(X|G,r_t)P(G|r_t)$$

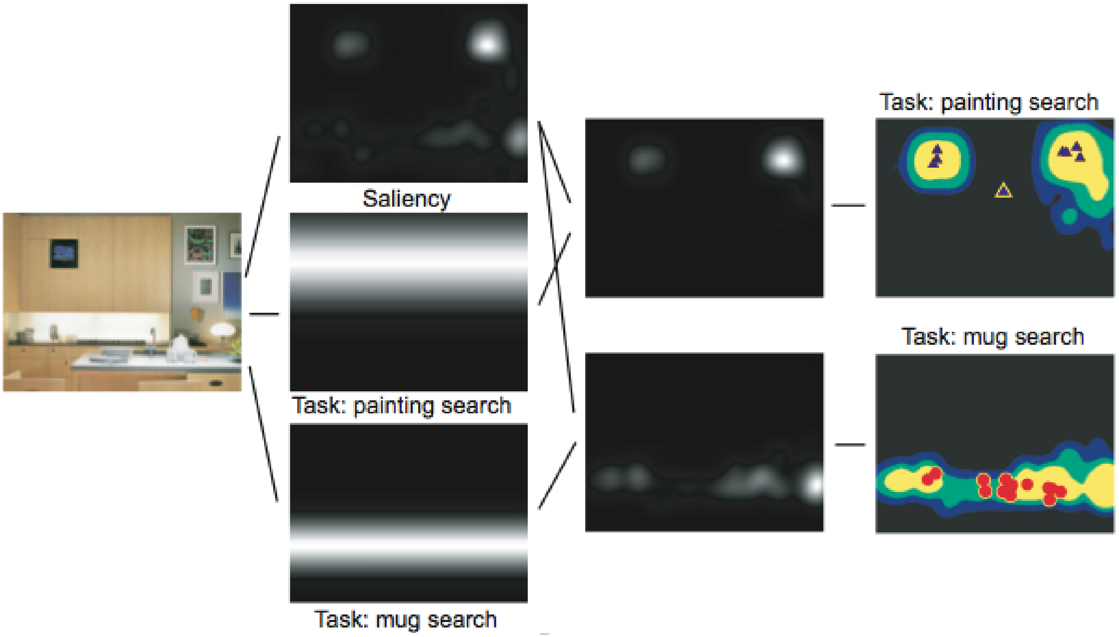


Figure 6. Scene information in the form of gist features can be used to suggest probable vertical locations for target objects, which results in more accurate salient map. Reprinted from [25]; permission not yet obtained.

$$= \sum_{t=1}^{M} \alpha_t N(X;W_t G, \Lambda_t)N(G;\mu_t,\Upsilon_t)$$

This equation has a number of parameters, but given a training set of images in which the target objects are labeled, they can be estimated using standard methods for fitting Gaussian mixtures. We won’t detail this training process here.

Having calculated the joint distribution, the desired conditional is just:

$$P(X|G) = \frac{P(X,G)}{P(G)} = \frac{\sum_{t=1}^{M} P(r_t)P(X|G,r_t)P(G|r_t)}{\sum_{t=1}^{M} P(r_t)P(G|r_t)}$$

This then gives the contribution of global features to the saliency of the location *X*. The overall saliency is given by

$$S(X) = L(X)^{-\gamma}P(X|G)$$

Here the parameter γ controls the relative contributions of local and global factors to the overall saliency. Interestingly the best value for this exponent as found by cross validation is quite small, 0.05, indicating the large relative importance of global contextual information.

Torralba et al. evaluated the contributions of local and global evidence, comparing models with and without global factors to human eye movement data. For both models, they defined a preferred region that consisted of the most salient 20% percent of the image, and counted the saccades that were contained in this region. To obtain an upper bound on the performance that could be expected of the model, they also defined a preferred region for a model that predicted the fixations using the fixations of all the others. The results show that using global information significantly improved performance over local saliency alone.

Other models

We have only been able to scratch the surface of the attention modeling literature, but we can mention here several further models that are particularly relevant. [33] unites models of three types presented here by placing them in a control space that allows them to be treated in a unified way. [34] is a model of attention that explicitly attempts to account for physiological data and for results at the single-cell level. Last, [35] is an intricate model that attempts to replicate a very wide array of experimental findings about the psychophysics of visual search.

7.3.3 Using Context for Object Detection.

A substantial body of experimental literature indicates that context plays an important role in human object detection. For example, [36] found that the ability of human subjects to recognize objects was impaired when the objects appeared outside of their natural or expected contexts. See [37] for a review of other contextual effects on human object perception. In addition, as illustrated by the last model in the previous section, context information can be used to improve the performance of computer vision models. Here we briefly present two more applications of this idea, showing how context can help with object detection.

A standard approach to object detection is to use a training set to learn the features of objects in the target class, and then to process a test image by using a sliding window to look for these features at a number of positions and scales. There are two glaring problems with the approach. First, it assumes that detections at different locations are independent; having found a target in one part of the image does not change the probability of finding another one in a different part. This is a problem because, ideally, the total number of detections should be reasonable given the scene-level content of the image. It seems unlikely, for example, that there would be a hundred cars in a mountain scene, or for that matter, a kitchen scene. The second problem is that the local detectors do not use location information; local features aside, they are just as likely to detect a car in the sky as on a road.

The model [24] uses scene-level information in the form of gist features to correct these problems. The gist features are used to estimate the total number of targets that are likely to be in an image given its scene type, and to predict the probable vertical locations of these targets. This information is then used to adjust the outputs of the local object detector. In many ways, this model is similar to the scene-based attention model of the previous section. Indeed, it can roughly be thought of as augmenting this model by adding the count estimates and incorporating information about the appearance of the target objects.

Another model [38] takes contextual reasoning further, incorporating relationships between object classes as well as between objects and scenes. This is a logical step, since the presence and position of objects in one class are often predictable given the presence and position of objects in another class. If we a see a computer monitor, for example, it is fairly safe to guess that there is keyboard in front of it. The model uses a database of labeled training images to learn this kind of statistical regularity, encoding the information it extracts in a tree-structured graphical model. In this graph, the nodes represent object classes and the edges represent statistical dependencies, Gaussian for position and Bernoulli for presence. As with the previous model, this information is used to correct the output of local object detectors.

7.3.4 Future Directions: Visual routines for scene perception

Although contextual information in the form of statistics about the co-occurrences and likely relative positions of objects is important, there are clearly problems that require the computation of much more detailed relations. For example, consider the problem of finding the object that a person in an image is pointing to. The most intuitively natural solution to this problem will require a fairly complex sequence of operations such as locating several features on the person's hand and arm and tracing a ray outwards. Simple schemes like using a database to learn a Gaussian relationship between the locations of the hand and the target are clearly inadequate, especially since the target's appearance cannot be used to pin down it's location. Similar considerations arise in action recognition. Imagine, for example, trying to distinguish images of a person catching a ball from images of a person throwing a ball.

Some of the reasoning also applies to a range of quite different visual tasks, such as finding the largest object in an image, or counting the objects present in an image. It should be noted that good solutions to all of the problems mentioned can be hand-coded, but getting a computer to *learn* to solve them is much more difficult, and will require a different set of tools than the ones described in the earlier sections of this chapter.

Although originally proposed as a model for intermediate-level vision, Shimon Ullman's visual routines constitute one framework that may be used for tasks like the ones given above. Ullman proposes that certain tasks that the visual system does – determining whether a point is inside a curve, for example – are best understood as being done by small visual programs that are composed of certain primitive visual actions. Ullman proposes the following five primitive actions, but the overall framework is robust to the individual choices:

(1) Shifting processing focus

(2) Finding salient (pop-out) locations
(3) Bounded activation, or the "coloring in" of a region enclosed by a boundary
(4) Edge tracing
(5) Marking a location as already processed

Tasks like finding the object that someone is pointing at are quite naturally thought of in routine-like terms, involving as they do operations like line tracing [39]. It therefore seems reasonable to formulate the problem of learning to accomplish them as a decision process in which an agent chooses primitive actions sequentially. An advantage of this formulation is that it allows to us draw on techniques from the reinforcement learning literature. In one possible setup, for example, we could imagine that an agent is given a collection of training images from which to learn a policy that can be applied on an unseen test image. Ideas in this direction have been explored, for example in [40] and [41], but most prior work has limited either the actions available or the class of problems to be solved. Overall, approaches like this have received much less attention than the other techniques covered in this chapter, and we believe that they will be a fertile area of future research.

Bibliography

[1] D. Hubel and T. Wiesel, "Recptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex," *Journal of Physiology*, vol. 160, pp. 106-154, 1962.

[2] D. Hubel and T. Wiesel, "Recptive Fields and Functional Architecture in Two Nonstraite Visual Areas (18 and 19) of the Cat," *Journal of neurophysiologyhysiology*, vol. 28, no. 2, pp. 229-289, 1965.

[3] T. Serre and T. Poggio, "A neuromorphic approach to computer vision," *Communications of the ACM*, vol. 53, no. 10, p. 54, Oct. 2010.

[4] E. Kobatake and K. Tanaka, "Neuronal selectivities to complex object features in the ventral visual pathway of the macaque cerebral cortex.," *Journal of neurophysiology*, vol. 71, no. 3, pp. 856-67, Mar. 1994.

[5] K. Fukushima, "Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193-202, 1980.

[6] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex.," *Nature neuroscience*, vol. 2, no. 11, pp. 1019-25, Nov. 1999.

[7] 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. 411-26, Mar. 2007.

[8] T. Serre, A. Oliva, and T. Poggio, “A feedforward architecture accounts for rapid categorization.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 15, pp. 6424-9, Apr. 2007.

[9] S. Thorpe, D. Fize, and C. Marlot, “Speed of processing in the human visual system.,” *nature*, vol. 381, no. 6582, pp. 520–522, 1996.

[10] I. Lampl, D. Ferster, T. Poggio, and M. Riesenhuber, “Intracellular measurements of spatial integration and the MAX operation in complex cells of the cat primary visual cortex.,” *Journal of neurophysiology*, vol. 92, no. 5, pp. 2704-13, Nov. 2004.

[11] U. Knoblich, “Biophysical Models of Neural Computation: Max and Tuning Circuits.” 06-Jun-2007.

[12] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks.,” *Science (New York, N.Y.)*, vol. 313, no. 5786, pp. 504-7, Jul. 2006.

[13] G. E. Hinton, “Learning multiple layers of representation.,” *Trends in cognitive sciences*, vol. 11, no. 10, pp. 428-34, Oct. 2007.

[14] G. E. Hinton, S. Osindero, and Y. W. Teh, “A Fast Learning Algorithm for Deep Belief Nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[15] G. E. Hinton, P. Dayan, B. Frey, and R. Neal, “The wake-sleep algorithm for unsupervised neural networks,” *Science*, vol. 268, no. 5214, pp. 1158-1161, 1995.

[16] A.-rahman Mohamed, G. Dahl, and G. Hinton, “Acoustic modeling using deep belief networks,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 99, pp. 1–1, 2011.

[17] H. Lee and A. Y. Ng, “Sparse deep belief net model for visual area V2,” *Energy*, pp. 1-8.

[18] T. S. Lee and D. Mumford, “Hierarchical Bayesian inference in the visual cortex.,” *Journal of the Optical Society of America. A, Optics, image science, and vision*, vol. 20, no. 7, pp. 1434-48, Jul. 2003.

[19] I. Biederman, “Recognition-by-components: a theory of human image understanding.,” *Psychological review*, vol. 94, no. 2, pp. 115-47, Apr. 1987.

[20] P. G. Schyns and A. Oliva, “FROM BLOBS TO BOUNDARY EDGES:. Evidence for Time- and Spatial-Scale-Dependent Scene Recognition,” *Psychological Science*, vol. 5, no. 4, pp. 195-200, Jul. 1994.

[21] 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.

[22] A. Oliva and A. Torralba, “Building the gist of a scene: the role of global image features in recognition.,” *Progress in brain research*, vol. 155, pp. 23-36, Jan. 2006.

[23] A. Quattoni and A. Torralba, “Recognizing Indoor Scenes,” in *Computer Vision and Pattern Recogntion*, 2009, pp. 413-420.

[24] B. A. Torralba, K. P. Murphy, and W. T. Freeman, “Using the Forest to See the Trees : Exploiting Context for Visual Object Detection and Localization,” *Processing*, pp. 107-114, 2003.

[25] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson, “Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search.,” *Psychological review*, vol. 113, no. 4, pp. 766-86, Oct. 2006.

[26] J. Xiao, J. Hays, K. A. Ehinger, and A. Torralba, “SUN Database : Large-scale Scene Recognition from Abbey to Zoo,” *Computer*.

[27] C. Koch and S. Ullman, “Shifts in selective visual attention: towards the underlying neural circuitry,” *Human Neurobiology*, vol. 4, no. 4, pp. 219-227, 1985.

[28] L. Itti and C. Koch, “A Comparison of Feature Combination Strategies for Saliency-Based Visual Attention Systems,” in *SPIE human vision and electronic imaging*, 1999.

[29] L. Itti and C. Koch, “A saliency-based search mechanism for overt and covert shifts of visual attention.,” *Vision research*, vol. 40, no. 10-12, pp. 1489-506, Jan. 2000.

[30] V. Navalpakkam and L. Itti, “An Integrated Model of Top-down and Bottom-up Attention for Optimizing Detection Speed,” in *Computer Vision and Pattern Recogntion*, 2006, pp. 2049 - 2056.

[31] M. C. Mozer and D. Baldwin, “Experience-Guided Search : A Theory of Attentional Control,” *Search*, pp. 1-8.

[32] M. C. Potter, "Journal of Experimental Psychology : Human Learning and Memory Short-Term Conceptual Memory for Pictures," vol. 2, no. 5, 1976.

[33] M. H. Wilder, M. C. Mozer, and C. D. Wickens, "An integrative , experience-based theory of attentional control," *Journal of Vision*, vol. 11, pp. 1-30, 2011.

[34] S. Chikkerur, T. Serre, C. Tan, and T. Poggio, "What and where: a Bayesian inference theory of attention.," *Vision research*, vol. 50, no. 22, pp. 2233-47, Oct. 2010.

[35] J. M. Wolfe, "Guided Search 4 . 0 Current Progress With a Model of Visual Search," *Search*, pp. 99-120, 2006.

[36] S. E. Palmer, "The effects of contextual scenes on the identification of objects," *Memory and Cognition*, vol. 3, no. 5, pp. 519-526, 1975.

[37] A. Oliva and A. Torralba, "The role of context in object recognition.," *Trends in cognitive sciences*, vol. 11, no. 12, pp. 520-7, Dec. 2007.

[38] M. Choi, J. Lim, and A. Torralba, "Exploiting hierarchical context on a large database of object categories," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 129-136.

[39] S. Rao, "Visual Routines and Attention," MIT, 1998.

[40] J. Vogel and N. de Freitas, "Target-directed attention: Sequential decision-making for gaze planning," *2008 IEEE International Conference on Robotics and Automation*, pp. 2372-2379, May. 2008.

[41] T. Darrell, "Reinforcement Learning of Active Recognition Behaviors," *Advances in Neural Information Processing Systems*, pp. 73-80, 1995.