Chapter VIII. Towards a biologically plausible computational model of ventral visual cortex

We have now come a long way since our initial steps towards defining the basic properties of vision in Chapter I. We started with characterizing the spatial and temporal statistics of natural images (Chapter II). We summarized visual behavior, how observers perceive the images around them (Chapter III). We explored how neurons along ventral visual cortex respond to a variety of different stimulus conditions (Chapters II, V, and VI). We described the recognition impairments that arise through cortical lesions and the consequences of injecting electrical currents in cortex (Chapter IV). We would like to put all of these separate bits and pieces of data into a coherent framework to understand how neuronal circuits help us see and interpret the world around us.

8.1. Quick recap and definitions

We start by defining what needs to be explained and the necessary constraints to solve the problem of vision. A theory of vision, implemented by a computational model, should have the following properties and be able to explain the following phenomena:

1. **Selectivity.** The visual system shows a remarkable degree of selectivity and can differentiate among shapes that appear to be very similar at the pixel level (e.g. arbitrary 3D shapes created from paperclips, different letters or faces, etc.). A model should be able to discriminate among physically similar but distinct shapes.

2. **Transformation tolerance.** A trivial solution to achieve high selectivity would be to memorize all the pixels in an image, a pure template matching algorithm. The problem with this type of algorithm is that it would not tolerate any changes in the image. An object can cast an infinite number of projections onto the retina. These changes arise due to changes in object position with respect to fixation, object scale, plane or depth rotation, changes in contrast or illumination, color, occlusion and others. The importance of combining selectivity and tolerance has been emphasized by many investigators (e.g. (Rolls, 1991; Olshausen et al., 1993; Logothetis and Sheinberg, 1996; Riesenhuber and Poggio, 1999; Deco and Rolls, 2004b; Serre et al., 2007b) among others).

3. **Speed.** Vision is very fast, as emphasized by many psychophysical investigations (Potter and Levy, 1969; Kirchner and Thorpe, 2006; Serre et al., 2007a), scalp EEG measurements (Thorpe et al., 1996) and neurophysiological recordings in humans (Liu et al., 2009) and monkeys (e.g. (Richmond et al., 1983; Keysers et al., 2001; Hung et al., 2005), among others). In about 150 milliseconds, we can decipher a lot of information and get a rather good first impression of what is happening in an image. This speed imposes an important constraint to the number of computational steps that the visual system can use for pattern recognition (Rolls, 1991; Serre et al., 2007b).
4. **Generic.** We can recognize a large variety of objects and shapes. Estimates about the exact number of objects or object categories that primates can discriminate vary widely depending on several assumptions and extrapolations (e.g. (Standing, 1973; Biederman, 1987; Abbott et al., 1996; Brady et al., 2008)). Certain types of shapes may be particularly interesting, they may have more cortical real estate associated with them, they could be processed faster and could be independently impaired. Yet, independently of precise figures about the number of shapes that primates can discriminate and independently also of whether natural objects are special or not, it is clear that there exists a generic system capable of discriminating among multiple arbitrary shapes. In fact, we can even discriminate between shapes that we see for the first time.

5. **Implementable in an image computable algorithm.** A successful theory of vision needs to be described in sufficient detail to be implemented through computational algorithms. This requirement is important because the computational implementation allows us to run simulations and hence to quantitatively compare the performance of the model against behavioral metrics. The simulations also lend themselves to a direct comparison between the model’s computational steps and neurophysiological responses at different stages of the visual processing circuitry. The algorithmic implementation forces us to rigorously state the assumptions and formalize the computational steps; in this way, computational models can be more readily compared than “armchair” or language-based conceptualizations. The implementation can also help us debug the theory by discovering hidden assumptions, bottlenecks and challenges that the algorithms cannot solve or where performance is poor. There are multiple fascinating ideas and theories about vision that have not been implemented through computational algorithms. These ideas can be extremely useful and helpful for the field and can inspire the development of computational models. Yet, we emphasize that we cannot easily compare theories that can be and have been implemented as image computable models against other ones that have not.

6. **Restricted to primates.** For simplicity, here we restrict the discussion to primate vision. There are strong similarities in vision at the behavioral and neurophysiological levels between macaque monkeys (one of the prime species for neurophysiological studies) and humans (e.g. (Myerson et al., 1981; Logothetis and Sheinberg, 1996; Orban, 2004; Nielsen et al., 2006; Kriegeskorte et al., 2008; Liu et al., 2009). Some of these models may well apply to other species (e.g., cats and rodents). Some aspects of the model may require refinement and modification for other species. When thinking about invertebrate vision (e.g. flies), there may need to be drastic changes to the overall model.

7. **Biophysically plausible.** If we are interested in linking the theory to actual brains, this implementation should be based on neural networks, it needs to be able to explain how those computations take place in terms of the basic units of computation in neural circuits, that is, neurons. Therefore, we restrict ourselves to models that are biophysically plausible. In doing so; thereby skipping a vast literature in Computer Vision where investigators are trying to solve similar problems without direct reference to the cortical circuitry. These engineering
approaches are extremely interesting and useful from a practical viewpoint. Ultimately, in the same way that computers can become quite successful at playing chess without any direct connection to the way humans play chess, computer vision approaches can achieve high performance without mimicking neuronal circuits. The insistence on biological plausibility should not be taken to imply that the operations or architectures in current networks are necessary, let alone sufficient, it seems very likely that we will make major changes and improvements to neural networks but the ultimate flavor of the implementation needs to be map to biological hardware.

8. Restricted to the visual system. The visual system is not isolated from the rest of the brain and there are plenty of connections between visual cortex and other sensory cortices, between visual cortex and memory systems in the medial temporal lobe and between the visual cortex and frontal cortex. It is likely that these connections also play an important role in the process of visual recognition, particularly through feedback signals that incorporate expectations (e.g. the probability that there is a lion in an office setting is very small), prior knowledge and experience (e.g. the object appears similar to another object that we are familiar with), cross-modal information (e.g. the object is likely to be a musical instrument because of the sound). To begin with and to simplify the problem, we restrict the discussion to the visual system.

8.2. Modeling the ventral visual stream – Common themes

Several investigators have proposed computational models that aim to capture some of the essential principles behind the transformations along the primate ventral visual stream. Before discussing some of those models in more detail, we start by providing some common themes that are shared by many of these models.

The input to the models is typically an image, defined by a matrix that contains the color of each pixel. Typically, this is a 3D matrix with the Red, Green, Blue intensities at each pixel. Some models also work in grayscale space. Object shapes can be discriminated quite well in grayscale images. Because the focus is often on the computational properties of ventral visual cortex, several investigators often ignore the complexities of modeling the computations in the retina and LGN; the pixels are meant to coarsely represent the output of retinal ganglion cells or LGN cells. This is of course one of the many oversimplifications in several computation models given that we know that images go through a number of transformations before retinal ganglion cells convey information to the LGN and on to cortex (Meister, 1996).

Most models have a hierarchical and deep structure that aims to mimic the approximately hierarchical architecture of ventral visual cortex (Felleman and Van Essen, 1991; Maunsell, 1995). The properties of deep networks has received considerable attention in the computational world, even if the mathematics of learning in deep networks that include non-linear responses is far less understood than their shallow counterparts (Poggio and Smale, 2003).
Neocortex and computer modelers have adopted a *Divide and Conquer* strategy whereby a complex problem is divided into many simpler tasks.

Most computational models assume, explicitly or implicitly, that cortex is cortex, and hence that there exist canonical microcircuits and computations that are repeated over and over throughout the hierarchy (Riesenhuber and Poggio, 1999; Douglas and Martin, 2004; Serre et al., 2007b). Thus, visual processing is approximated by a hierarchy of computations, each one of which is rather simple in nature and encompasses basic computations that are biophysically plausible such as computing dot products, applying a non-linear transformation to the integrated activity in the neuronal soma, and normalizing the outputs.

As we ascend through the hierarchical structure of the model, units in higher levels typically have larger receptive fields, respond to more complex visual features and show an increased degree of tolerance to transformations of their preferred features.

### 8.3. A panoply of models

We summarize here a few important ideas that have been developed to describe visual processing. The presentation here is neither an exhaustive list nor a thorough discussion of each of these approaches. For a more detailed discussion of several of these approaches, see (Ullman, 1996; LeCun et al., 1998; Riesenhuber and Poggio, 2002; Deco and Rolls, 2004a; Serre et al., 2005b).

Straightforward template matching does not work for pattern recognition. Even shifting a pattern by one pixel would pose significant challenges for an algorithm that merely compares the input with a stored pattern on a pixel-by-pixel fashion. As noted at the beginning of this chapter, a key challenge to recognition is that an object can lead to infinite number of retinal images depending on its size, position, illumination, etc. If all objects were always presented in a standardized position, scale, rotation and illumination, recognition would be considerably easier (DiCarlo and Cox, 2007; Serre et al., 2007b). Based on this notion, several approaches are based on trying to transform an incoming object into a canonical prototypical format by shifting, scaling and rotating objects (e.g. (Ullman, 1996)). The type of transformations required is usually rather complex, particularly for non-affine transformations. While some of these problems can be overcome by ingenious computational strategies, it is not entirely clear how the brain would implement such complex calculations nor is there currently any clear link to the type of neurophysiological responses observed in ventral visual cortex.

A number of approaches are based on decomposing an object into its component parts and their interactions. The idea behind this notion is that there could be a small dictionary of object parts and a small set of possible interactions that act as building blocks of all objects. Several of these ideas can be traced back to the prominent work of David Marr (Marr and Nishihara, 1978; Marr, 1982) where those constituent parts were based on generalized cone shapes. The artificial intelligence community also embraced the notion of structural descriptions (Winston, 1975). In the same way that a mathematical function can
be decomposed into a sum over a certain basis set (e.g. polynomials or sine and cosine functions), the idea of thinking about objects as a sum over parts is attractive because it may be possible and easier to detect these parts in a transformation-invariant manner (Biederman, 1987; Mel, 1997). In the simplest instantiations, these models are based on merely detecting a conjunction of object parts, an approach that suffers from the fact that part rearrangements would not impair recognition but they should (e.g. a house with a garage on the roof and the chimney on the floor). More elaborate versions include part interactions and relative positions. Yet, this approach seems to convert the problem of object recognition to the problem of object part recognition plus the problem of object parts interaction recognition. It is not entirely obvious that object part recognition would be a trivial problem in itself nor is it obvious that any object can be uniquely and succinctly described by a universal and small dictionary of simpler parts. It is not entirely trivial how recognition of complex shapes (e.g. consider discriminating between two faces) can be easily described in terms of a structural description of parts and their interactions. Computational implementations of these structural descriptions have been sparse (see however (Hummel and Biederman, 1992)). More importantly, it is not entirely apparent how these structural descriptions relate to the neurophysiology of the ventral visual cortex (see however (Vogels et al., 2001)).

A series of computational algorithms, typically rooted in the neural network literature (Hinton, 1992), attempt to build deep structures where inputs can be reconstructed (for a recent version of this, see e.g. (Hinton and Salakhutdinov, 2006). In an extreme version of this approach, there is no information loss along the deep hierarchy and backward signals carry information capable of re-creating arbitrary inputs in lower visual areas. There are a number of interesting applications for such “auto-encoder” deep networks such as the possibility of performing dimensionality reduction. From a neurophysiological viewpoint, however, it seems that the purpose of cortex is precisely the opposite, namely, to lose information in biologically interesting ways. It is not clear why one build an entire network to copy the input (possibly with fewer units). In other words, as emphasized at the beginning of this chapter, it seems that a key goal of ventral visual cortex is to be able to extract biologically relevant information (e.g. object identity) in spite of changes in the input at the pixel level.

Particularly within the neurophysiology community, there exist several “metric” approaches where investigators attempt to parametrically define a space of shapes and then record the activity of neurons along the ventral visual stream in response to these shapes (Tanaka, 1996; Brincat and Connor, 2004; Connor et al., 2007). This dictionary of shapes can be more or less quantitatively defined. For example, in some cases, investigators start by presenting different shapes in search of a stimulus that elicits strong responses. Subsequently, they manipulate the “preferred” stimulus by removing different parts and evaluating how the neuronal responses are modified by these transformations. While interesting, these approaches suffer from the difficulties inherent in considering arbitrary shapes that may or may not constitute truly “preferred” stimuli. Additionally, in some cases, the transformations examined only reveal anthropomorphic biases
about what features could be relevant. Another approach is to define shapes parametrically. For example, Brincat and colleagues considered a family of curvatures and modeled responses in a six-dimensional space defined by a sum of Gaussians with parameters given by the curvature, orientation, relative position and absolute position of the contour elements in the display. This approach is intriguing because it has the attractive property of allowing investigators to plot "tuning curves" similar to the ones used to represent the activity of units in earlier visual areas. Yet, it also makes strong assumptions about the type of shapes preferred by the units.

8.4. The convolution operation

One of the key computational ingredients of visual processing is that the same operation is typically repeated throughout the visual field. For example, we find neurons in primary visual cortex that show orientation tuning with receptive fields tiling the entire visual field. Thus, a computational operation that filters the image to yield units with this type of tuning, needs to be repeated over and over throughout the image. This type of operation is readily implemented through the convolution operation.

Figure 8.1. Basic operations in neural networks. A grayscale image (14 x 14 pixel image representing number 3) is convolved with three different filters (b, e, h). In this case, each of the filters is 3 x 3 pixels and for simplicity the values are only 0s and 1s. The convolution operation here has a "stride" of 2, meaning the filter skips through one pixel as it slides through the image. The green (blue) location in the image (a) yields the output highlighted in green (blue) after convolution in (c, f, i). A pooling operation takes the output of the convolution and extracts the maximum in blocks of size 2x2, also with a stride of 2. The yellow location after convolution corresponds to the yellow location in the final output in d, g, j.
**convolutus**, to roll together) is defined as the integral of one signal being reflected, shifted, and multiplied by the other:

\[
f(t) \ast g(t) = \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau
\]

In image processing, the process of convolution refers to shifting a certain filter throughout the entire image (or the entire previous layer), and returning the output at each location. An example of this process is illustrated in Figure 8.1. For simplicity, here the input image is a grayscale version of a handwritten version of the number 3, reduced to 14 x 14 pixels. Each pixel has an intensity between 0 (black) and 255 (white). In general, the numbers do not need to be integers and the input image could have three colors. We consider three possible feature filters shown in Figure 8.1c (vertical), e (horizontal) and h (diagonal). To simplify the numbers, here the 3x3 pixel filter weights consist of zeros and ones, but again, in general, these are real values. The filter is placed at each location in the image and the values in the filter are multiplied by the corresponding values in the image. For example, consider the yellow square in the image containing the values 3, 6, 5 in the first row, 0, 0, 0 in the second row, and 58, 127, 130 in the third row. We get 0 x 3 + 1 x 6 + 0 x 5 = 6 in the first row, 0 x 0 + 1 x 0 + 0 x 0 = 0 in the second row, and 0 x 58 + 1 x 127 + 0 x 130 = 127 in the third row, yielding a total of 133 in Figure 8.1c. The same process is repeated throughout the entire image to yield the matrix in Figure 8.1c. Because the filter resembles a vertical line, the operation highlights regions of the input image that contain pixels that look like short vertical lines. Similarly, the filter in Figure 8.1e highlights horizontal edges and the one in Figure 8.1h highlights diagonal ones.

After convolution, there is a pooling operation that combines multiple values within a window. This has the effect of increasing the receptive field size. A typical pooling operation is to take a maximum. For example, consider the yellow square in Figure 8.1c consisting a 2 x 2 matrix with values 229, 262 in the first row and 391, 467 in the second row. These four numbers are combined to yield 467 in Figure 8.1d. The max pooling operation provides position invariance, by allowing high activity in this case in any of the four locations. This pooling operation is reminiscent of the transformation between simple cells and complex cells in primary visual cortex (Chapter V).

### 8.5. Bottom-up hierarchical models of the ventral visual stream

A hierarchical network model can be described by a series of layers \( l=0, 1, \ldots, N \). Each layer contains \( n_l \times n_l \) units arranged in a matrix, each unit having a circumscribed receptive field and therefore being activated by a certain location in the image. Additionally, there may be multiple different filter features \( K_l \) (sometimes referred to as channels or kernels) at each location, creating a collection of such matrices; for example, the input image may have \( K=3 \) colors.
Such a collection of matrices in each layer is called a tensor. The activity of each unit in each layer can be represented by the value $x_l (x \in \mathbb{R}^{n_l \times n_l \times K_1})$. Each entry in $x$ is typically represented by a scalar value, which can be coarsely interpreted as the firing rate of the unit. The initial layer is defined as the input image; $x_0$ represents the pixel values in the image.

From one layer to the next, the matrices are transformed through convolution operations, through non-linearities like the ReLU operation, and through pooling operations like max pooling. In the most typical scenario, these operations take place between layer $l$ and layer $l+1$. In other words, this assumes that the activity in a given layer only depends on the activity pattern in the previous layer. This assumption implies that at least three types of connections are ignored: (i) connections that “skip” a layer in the hierarchy (e.g. synapses from V1 directly onto V4 skipping V2); (ii) top-down connections (e.g. synapses from V2 to V1 (Virga, 1989)) and (iii) connections within a layer (e.g. horizontal connections between neurons with similar preferences in V1 (Callaway, 1998)).

It is tacitly assumed in most models that there exist general rules, often summarized in the epithet “cortex is cortex”, such that only a few types of transformations are allowed in the computations from one layer to the next. One of the early models that aimed to describe object recognition, inspired by the neurophysiological findings of Hubel and Wiesel, was the neocognitron (Fukushima, 1980) (see also earlier theoretical ideas in (Sutherland, 1968)). This model had two possible operations, a linear tuning function (performed by “simple” cells) and a non-linear OR operation (performed by “complex” cells). These two operations were alternated and repeated through the multiple layers in the deep hierarchy. This model demonstrated the feasibility of such linear/non-linear cascades in achieving scale and position tolerance in a letter recognition task. Several subsequent efforts (Olshausen et al., 1993; Wallis and Rolls, 1997; LeCun et al., 1998; Riesenhuber and Poggio, 1999; Amit and Mascaro, 2003; Deco and Rolls, 2004b) were inspired by the Neocognitron architecture.

One such effort to expand on the computational abilities of the Neocognitron in the computational model developed in the Poggio group (Riesenhuber and Poggio, 1999; Serre et al., 2005b; Serre et al., 2007b). This model is characterized by a purely feed-forward and hierarchical architecture. An image, represented by grayscale values, is convolved with Gabor filters at multiple scales and positions to mimic the responses of simple cells in primary visual cortex. Like other efforts, the model consists of a cascade of linear and non-linear operations. These operations come in only two flavors in the model: a tuning operation and max pooling operation.

This and similar architectures have been subjected to several tests including comparison with psychophysical measurements (e.g. (Serre et al., 2007a)), comparison with neurophysiological responses (e.g. (Deco and Rolls, 2004b; Lampl et al., 2004; Hung et al., 2005; Serre et al., 2005b; Cadieu et al.,
and quantitative evaluation of performance in computer vision recognition tasks (e.g. (LeCun et al., 1998; Serre et al., 2005a; Mutch and Lowe, 2006)).

8.6. Learning the weights

In the Neocognitron and initial implementation of the HMAX architecture, the weights between layers are hand-crafted and fixed. In current implementations of image computable models, the weights are typically learned while the algorithm is trained.

Let us consider the problem of object categorization where the goal is to take an image and assign a label to it (for example, to detect whether the image contains a handwritten 0, 1, 2, …, or 9 as in the MNIST database, Chapter VII, or to identify a face). In a large number of computer vision studies, investigators tried to come up with better and better features to extract from the image. These features were then submitted to a suitable classifier. All the task-dependent learning happened at the level of the classifier. A lot of features were found to be generally useful for object classification tasks including extraction of edges, colors, principal components, shift-invariant feature transformations (SIFT), corners, spatial frequency decomposition, and many others. A classifier, such as a support vector machine, was then in charge of learning the map between those features and the corresponding image labels.

In stark contrast, the bulk of the work nowadays is in building end-to-end trainable systems, typically with a fixed architecture, typically with randomly initialized weights, and where all the weights are plastic and can be modified to achieve the best possible classification accuracy. The data (images and labels) are divided into a training set and a test set (cross-validation, Chapter VII). The weights can be learnt in a supervised manner or in an unsupervised manner (Chapter VII).

One of the most successful ways of adjusting the weights via supervised learning is backpropagation where the difference between the target outputs and current outputs is calculated and propagated back via gradient descent throughout the entire network.

8.7. Computer vision everywhere

The success of such end-to-end training algorithms, combined with the rapid increase in computational power and the availability of large amounts of data has given rise to wide applicability of computer vision pattern recognition algorithms.

Computer vision algorithms have been applied to a large number of image classification tasks where the input is an image and the output is a label.
Supervised learning algorithms for deep convolutional networks with randomly initialized weights are typically data hungry. At least partly, progress in image classification research can be attributed to the enormous number of pictures that people have uploaded on the web and persistent efforts to annotate those images. Computer scientists love when users tag the images they upload. An example database that has played an important role in computer vision over the last years is ImageNet consists of approximately $10^6$ images divided into 1,000 categories. These images were taken from the web, and the categories include chairs, tables, a wide variety of animals, different types of flowers and many more. Some of these categories are rather specialized and perhaps even a bit esoteric, like stinkhorn. Some types of objects have multiple categories, like multiple different types of cats, and an even greater number of dog-related categories. Yet, regardless of these considerations, a database of this size was not available before, and therefore this provided a nice playground to develop, refine, and build more complex deep convolutional neural networks.

Another great aspect of ImageNet and similar datasets is that it empowered direct comparison and benchmarking of different algorithms. Comparing how algorithm X processes a dataset Ix to how an algorithm Y processes a different dataset Iy is rather difficult. Although this is a rather simple concept, benchmarks based on common datasets are not common in other domains. For example, in Neuroscience, almost every lab creates its own tasks, using their custom-made images, rendering results difficult to integrate and compare.

Success in labeling ImageNet images inspired and triggered a large number of efforts in image classification in other domains. Clinical diagnosis based on images can sometimes be simplified into a visual pattern recognition on images. For example, a radiologist can examine a mammogram to determine whether it contains a breast tumor or not. A database consisting of many such images annotated by experts can be readily used to train computer vision algorithms. Computer vision algorithms have thrived in a wide variety of image diagnosis efforts. Clinicians do not always agree with each other on the diagnosis of a given image. Furthermore, clinicians sometimes do not even agree with themselves when repeatedly tested on the same images. In breast tumor detection, computational algorithms are now indistinguishable from expert clinical diagnosis. Another interesting was the development of computational algorithms to diagnose diabetic retinopathy based on fundus photographs. Not surprisingly, computer vision algorithms quickly learned to recognize the right patterns and achieved clinician-level accuracy. In a surprising twist, computer scientists then asked whether they could extract other types of information from the fundus photographs. For example, instead of learning yes/no labels for diabetic retinopathy, they trained the same algorithms to predict the subject's gender, or separately, the subject's age. The algorithms were able to predict gender and age quite accurately, which was quite surprising to clinicians who were unable to do so. In other words, computer vision could answer new questions and discover...
new patterns that clinicians had not even thought about. Gender and age are perhaps not that interesting. After all, the ophthalmologist has full access to the patient and always knows his/her age and gender, without the need to decipher this information from fundus photographs. Yet, there was yet another interesting development to this line of work. Computer scientists discovered that they were able to predict cardiovascular disease from the fundus photographs. This is quite remarkable because this is a question that ophthalmologists had never thought about, it is a question that is extremely relevant from a clinical standpoint, and it is additional information that comes for free from the fundus photograph without any additional clinical testing. What is perhaps even more remarkable is that computer vision was able to predict cardiovascular disease better than the Framingham score, which is considered to be one of the best indicators of cardiovascular risk.

Image classification extended well beyond clinical diagnosis. Computer vision has shed light on the gargantuan task of classifying galaxies from astronomy images. Computer vision has been used to classify flora and fauna, quickly surpassing any naïve observer and becoming the envy of expert biologists. Another image classification problem that has been radically transformed by computer vision is face identification. Many smart phones these days have algorithms that use faces to login, which used to be the domain of science fiction movies not too long ago. Some ATM machines are testing face recognition algorithms as well and it seems likely that it won’t take long before this becomes standard. Quantitative studies of face identification have shown that computer vision systems are better than forensic experts and better than so-called superrecognizers, people with extraordinary capacity to remember and recognize faces. In many developed countries, it may be difficult to remain anonymous in the near future and any device may be able to identify a person quite accurately.

In addition to image classification, algorithms have been developed for object detection. In these tasks, the goal is to place a bounding box around the object of interest in an image. Progress in object localization rapidly accelerated with the development of the MSCOCO dataset which contains detailed tracing contours around objects from 80 different common categories. One example of object detection is the ability to put a box around a face in an image, which is routinely used nowadays for digital cameras to focus on faces. Current algorithms can detect and place bounding boxes around multiple objects in an image. This type of effort has provided a large boost to the possibility of developing self-driving cars, which are equipped with sensors to detect other cars, to detect pedestrians, car lanes, and many other objects of interest.

Another question of interest for computer vision is object segmentation where the goal is to trace the contour of a given object. Instance segmentation refers to separating and labeling every single pixel in an image.
In addition to labeling and detecting individual objects, many applications require the ability to identify actions in an image. Action recognition has so far been characterized by smaller training datasets, but the field is progressing quickly. Action recognition can be based on individual images but it has also triggered the development of databases based on videos. It is likely that we will soon have systems that can detect and track players in sport videos and automatically analyze the games in excruciating detail. Action recognition has also become widespread among biologists studying animal behavior. Traditionally, this is a very hard task: a graduate student interested in mouse behavior may easily mount a camera to record hours and hours of behavioral data. Analyzing those data typically involved very long hours of watching those videos and subjectively annotating the observations. Nowadays, there are many systems that can objectively and reliably perform this type of annotations. Face identification as well as action recognition are linked to one important application for automatic computer vision: surveillance.

The list of computer vision applications is so large and grows so rapidly that it is likely that by the time the reader has access to these lines, there will already be a plethora of impressive new feats. Interestingly, the same algorithms developed for computer vision problems quickly found applications outside vision. Speech recognition, systems that suggest automatic replies to emails, attempts at predicting the weather or the stock market, predicting the results of elections, consumer behavior, and many other questions have now been revolutionized by deep convolutional networks originally developed to label images.

8.8. A cautionary note: lots of parameters!

An intriguing aspect of the procedure outlined in Section 8.6 to learn from examples is that adjusting the weights involves an enormous number of parameters. Consider an image of 256 x 256 pixels with 3 colors, that amounts to 196,608 inputs. If there are 1,000 possible categories (ImageNet) and in the simplest scenario of mapping the inputs directly to the outputs, we would have about 200 x 10^6 parameters. And about 10^6 training examples (ImageNet).

If you have a system of equations with 10 unknowns and 4 equations, you are in trouble. In general, there are infinite possible solutions. More parameters than constraints is generally a bad idea. Similarly, if you are trying to fit a curve to data and your curve has 10 free parameters and you only have 4 data points, it is very likely that you will overfit.

When training deep convolutional networks, having more free parameters than training data is actually rather common. Therefore, it is quite puzzling when these models perform well in cross-validated data.

8.9. Top-down signals in visual recognition
In spite of the multiple simplifications, the success of bottom-up architecture in describing a large number of visual recognition phenomena suggest that they may not be a bad first cut. As emphasized above, bottom-up architectures constitute only an approximation to the complexities and wonders of neocortical computation. One of the several simplifications in bottom-up models is the lack of top-down signals. We know that there are abundant back-projections in neocortex (e.g. (Felleman and Van Essen, 1991; Callaway, 2004; Douglas and Martin, 2004)). The functions of top-down connections have been less studied at the neurophysiological level but there is no shortage of computational models illustrating the rich array of computations that emerge with such connectivity. Several models have used top-down connections to guide attention to specific locations or specific features within the image (e.g. (Olshausen et al., 1993; Itti and Koch, 2001)(Tsotsos, 1990; Deco and Rolls, 2005; Rao, 2005; Compte and Wang, 2006; Chikkerur et al., 2009)). The allocation of attention to specific parts of an image can significantly enhance recognition performance by alleviating the problems associated with image segmentation and with clutter.

Top-down signals can also play an important role in recognition of occluded objects. When only partial object information is available, the system must be able to perform object completion and interpret the image based on prior knowledge. Attractor networks have been shown to be able to retrieve the identity of stored memories from partial information (e.g. (Hopfield, 1982)). Some computational models have combined bottom-up architectures with attractor networks at the top of the hierarchy (e.g. (Deco and Rolls, 2004b)). During object completion, top-down signals could play an important role by providing prior stored information that influences the bottom-up sensory responses. Several proposals have argued that visual recognition can be formulated as a Bayesian inference problem (Mumford, 1992; Rao et al., 2002; Lee and Mumford, 2003; Rao, 2004; Yuille and Kersten, 2006; Chikkerur et al., 2009). Considering three layers of the visual cascade (e.g. LGN, V1 and higher areas), and denoting activity in those layers as $x_0$, $x_1$ and $x_h$ respectively, then the probability of obtaining a given response pattern in V1 depends both on the sensory input as well as feedback from higher areas:

$$P(x_1|x_0) = \frac{P(x_0|x_1)P(x_1|x_h)}{P(x_0|x_h)}$$

Equation 11.8

where $P(x_1|x_h)$ represents the feedback biases conveying prior information. An intriguing idea proposed by Rao and Ballard argues that top-down connections provide predictive signals whereas bottom-up signals convey the difference between the sensory input and the top-down predictions (Rao and Ballard, 1999).

8.7. Applications
There is no shortage of applications where automatic or semi-automatic algorithms are being explored in computer vision. Here are a few examples:

(A) Intelligent content-based search. Searching for images in the web by content will open the doors to a large number of applications. Facebook users can already experiment with prototypes that let them search for people. One may be able to look for images that are similar to a search query in terms of content. Blind people may be able to point their phones and find out where they are and how to navigate.

(B) Prototype cars that can navigate automatically rely heavily on algorithms to detect pedestrians, other cars, other objects and road conditions.

(C) ATM machines may be able to recognize their customers. Cars and houses may recognize their owners.

(D) Security screening in places like airports may benefit from automatic recognition systems.

(E) Several clinical problems are based on pattern recognition and computers may soon help doctors to make informed decisions based on their understanding of patterns.

8.8. Computer vision tasks

Algorithms have been developed to address several interrelated problems in machine vision. While some of the boundaries are blurred in several applications, it is useful to think about the following tasks:

(A) Object detection. For example, a digital camera may require detecting the presence and location of a face in an image for focusing. Face detection may thrive without solving the problem of recognition.

(B) Object segmentation. In natural images, it may be of interest to separate an object from the background. For example, it may be important to detect the location of a tumor in an image. Or to detect the presence of a tank in a camouflaged image.

(C) Object recognition. Recognizing objects can often be thought of as associating the image with labels. These labels may refer to the identity of the object (e.g. given a face, who is it?) or the object's category (e.g. is there an animal in this image?).

(D) Object verification. In some cases, it may be of interest to evaluate whether two images are the same or not.

8.9. Object segmentation

Given a natural scene, humans (and other species) are quite good at being able to characterize and localize different objects embedded in complex backgrounds. The fact that this is not a trivial problem is highlighted by the ubiquitous use of camouflage in the animal world. Particularly for objects that are not moving, matching colors, contrast and textures can help animals avoid predators or at least buy sufficient time for escape. Basic aspects of segmentation may depend on adequately detecting edges. However, more
complex problems often involve a deeper understanding of the interrelationships among different object parts. A typical case involves recognizing a zebra as a whole animal as opposed to thinking of each stripe as a separate object.


Some algorithms require recognition prior to segmentation while other algorithms use segmentation to guide recognition in complex scenes. To avoid this chicken-and-egg dilemma, it is tempting to speculate that certain aspects of bottom-up recognition and segmentation could occur (or at least) start independently of each other, using overlapping neuronal circuits. Top-down signals may then combine segmentation and recognition in synergistic fashion. For examples of object segmentation algorithms see (Borenstein et al., 2004).

8.10. A general scheme for object recognition

Figure 12.1 illustrates a typical approach in computer vision efforts. Consider a series of N labeled images \((x_i, y_i)\) where \(i=1,...,N\), \(x\) is a matrix representing the image and \(y\) is a label (e.g. face present or not). A set of features \(f\) is extracted from the images: \(f_i = g(x_i)\). Those features may include properties such as edges, principal components, etc. How those features are chosen is one of the key aspects that differentiates computer vision algorithms. A supervised learning scheme is then used to learn the map between those features and labels (Poggio and Smale, 2003; Meyers and Kreiman, 2011; Singer and Kreiman, 2011). For example, a support vector machine (SVM) classifier with a linear kernel may be used to learn the structure of the data and labels. A cross-validation procedure is followed by separating the data into a training set and a test set to avoid overfitting. After training, the algorithm is evaluated with the images in the test set. By using different algorithms applied to the same data, the merits of alternative approaches can be quantitatively compared.

8.11. A successful example: digit recognition

Recognizing hand-written digits constitutes an example where computers have reached high accuracy, almost comparable to human levels (e.g. (LeCun et al., 1998)). Figure 12.2 shows an example of the errors made by an early attempt
at recognizing hand-written digits. The overall error rate of this algorithm was <2%. Several of those errors are not trivial to recognize and humans could make mistakes as well.

8.12. Image recognition competitions

There are several computer vision competitions with large data sets consisting of labeled images. One such competition is called the Imagenet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2014). The 2014 instantiation of the object classification part of this challenge included 1000 object classes, 1,281,413 images for training (732-1300 images per class) and 100,000 images for testing (100 images per class). This competition also includes other tasks beyond classification including object detection and localization. To give an idea of performance, the winning team in the object classification part of the challenge achieved an error rate slightly above 6%. This is quite impressive considering that there were 1000 classes. It should be noted, though, that the results of these competitions are often reported in a somewhat strange way by allowing the models 5 changes to get it right and reporting the results as correct if any of those 5 predictions are correct. This makes it a bit more difficult to directly compare against human performance (Borji and Itti, 2014). Another aspect of machine vision that has also been highlighted is the difficulty in interpreting how the machine classifies objects and investigators have reported puzzling examples where minimal changes to an image drastically change the predicted class (Szegedy et al., 2014). With that said, the results are still quite remarkable and they show rapid progress in teaching machines to recognize objects.

8.10. Visual recognition goes beyond identifying objects in single images

We emphasize that visual recognition is far more complex than the identification of specific objects. Under natural viewing conditions, objects are embedded in complex scenes and need to be separated from their background. How this segmentation occurs constitutes an important challenge in itself. Segmentation depends on a variety of cues including sharp edges, texture changes and object motion among others. Some object recognition models assume that segmentation must occur prior to recognition. There is no clear biological evidence for segmentation prior to recognition and therefore this constitutes a weakness in such approaches. We
do not discuss segmentation here (see (Borenstein et al., 2004; Sharon et al., 2006) for recent examples of segmentation algorithms).

Most object recognition models are based on studying static images. Under natural viewing conditions, there are important cues that depend on the temporal integration of information. These dynamic cues can significantly enhance recognition. Yet, it is clear that we can recognize objects in static images and therefore many models focus on the reduced version the pattern recognition problem using static objects. Here we also focus on static images.

We can perform a variety of complex tasks that rely on visual information that are different from identification. For example, we can put together images of snakes, lions and dolphins and categorize them as animals. Categorization is a very important problem in vision research and it also constitutes a formidable challenge for computer-based approaches. Here we focus on the question of object identification.

8.11. The road ahead

If you can do it, a computer can do it too. Significant progress has been made over the last decade in teaching computers to perform multiple tasks that were traditionally thought to be the domain of humans. Any desktop computer can play chess competitively and the best computers can beat the world’s chess champion. IBM’s Watson has thrived in the trivia-like game of Jeopardy. And while imperfect, Siri and related systems are making enormous strides in becoming the world’s best assistants.

In the domain of vision, computational algorithms are already able to perform certain tasks such as recognizing digits in a fully automatic fashion at human performance level and demonstrate reasonable performance in other tasks such as detecting faces for focusing on in digital cameras. In several other tasks, humans still outperform the most sophisticated current algorithms but the gap between machines and humans in vision tasks is closing rapidly. Here we provide an overview of several computer vision systems, particularly in the context of pattern recognition problems and describe what machine vision systems can and cannot do,

Significant progress has been made towards describing visual object recognition in a principled and theoretically sound fashion. Yet, the lacunas in our understanding of the functional and computational architecture of ventral visual cortex are not small. The preliminary steps have distilled important principles of neocortical computation including deep networks that can divide and conquer complex tasks, bottom-up circuits that perform rapid computations, gradual increases in selectivity and tolerance to object transformation.

In stark contrast with the pathway from the retina to primary visual cortex, we do not have a quantitative description of the feature preferences of neurons along the ventral visual pathway. And several computational models do not make
clear, concrete and testable predictions towards systematically characterizing ventral visual cortex at the physiological levels. Computational models can perform several complex recognition tasks and compete against non-biological computer vision approaches. Yet, for the vast majority of recognition tasks, they still fall significantly below human performance.

The next several years are likely to bring many new surprises in the field. We will be able to characterize the system at unprecedented resolution at the experimental level and we will be able to evaluate sophisticated and computationally intensive theories in realistic times. In the same way that the younger generations are not surprised by machines that can play chess quite competitively, the next generation may not be surprised by intelligent devices that can "see" like we do.

### 8.13. References

Biological and Computer Vision

Chapter 8


