9 Toward a World with Intelligent Machines That Can Interpret the Visual World

Supplementary content at http://bit.ly/2t53QRd

In the previous chapter, we introduced the idea of directly comparing computational models versus human behavior in visual tasks. For example, we assess how models classify an image versus how humans classify the same image. In some tasks, the types of errors made by computational models can be similar to human mistakes. Here we will dig deeper into what current computer vision algorithms can and cannot do. We will highlight the enormous power of current computational models, while at the same time emphasizing some of their limitations and the exciting work ahead of us to build better models.

There are many visual problems where computers are already significantly better than humans. A simple example is the ability to read bar codes, such as the ones used in a supermarket to label each product. Even if humans could, in principle, go through enormous training to read bar codes, it would be extremely challenging to achieve machine-level performance in this task. In most supermarkets, there is still a need for a human to turn the product, locate the bar code, and position the bar code in such a way that the scanner can process it. This level of human intervention will probably vanish soon, yet in some sense, it is interesting to note that localizing the bar code and adequately positioning it is still easier for humans than machines.

There is a double dissociation here in terms of which tasks humans find easy (locating a bar code and positioning the product the right way) and which tasks are easy for machines (deciphering the bar code). The task may seem somewhat limited: it all comes down to measuring bar widths and distances. The human solves the challenging invariance problem (recognition of an image at different scales, positions, and angles, as in Figure 3.6) by positioning the object in the right place. A similar case can be made for reading quick response (QR) codes. As we will discuss soon, there are many other visual tasks where computers already match or outperform humans. There are also many visual tasks where machines still have a long way to go to reach human performance levels. Hans Moravec, Rodney Brooks, and Marvin Minsky articulated this dissociation between machine and human performance in Moravec's paradox. The paradox states that it is relatively easy to endow computers with adult-level performance on traditional intelligence tests and incredibly challenging to give machines the skills of a one-year-old in terms of perception and mobility.

What would it mean for computational algorithms to match or outperform humans in every possible visual task? Imagine a world where machines can truly see and interpret the visual world around us – a world where machines can pass the *Turing test for vision*.

9.1 The Turing Test for Vision

Alan Turing (1912–1954) was one of the great minds of the twentieth century and pioneered the development of the theory of computer science. In his seminal 1950 paper, he proposed the "Imitation Game," whereby a series of questions is posed both to a human and to a computer. Turing proposed that if we cannot distinguish which answers came from the human and which ones came from the computer, then we should call that computer intelligent.

The term intelligence is ill defined and used in many different ways. Furthermore, the notion of machine intelligence is often a moving target: once computers can solve a given task (such as beating world champions at the game of chess or Go), then critics invariably argue that such a feat is *not* an actual demonstration of intelligence (even though the same experts claimed otherwise before computers beat humans). Those people often have in mind a useless definition of intelligence: intelligence is whatever computers cannot do! To avoid such tautologies, the Turing test has become the standard goal to assess intelligence.

We can define a specialized version of the Turing test for visual intelligence (Figure 9.1). Suppose that we present a human or computer with an image (or a video without sound). It is important that there are no restrictions on the image: it can be a frame extracted from a Disney movie, a Kandinsky, or a photograph like the one in Figure 9.1. We are allowed to ask *any* question about the image. For example, we can ask whether it contains a tree, how many cars there are, whether any person is wearing a

Figure 9.1 Turing test for vision. Given an arbitrary image and any question about the image, if we cannot distinguish whether the answers come from a human or a computational algorithm, we say that the algorithm has passed the Turing test for vision.

hat, whether the person wearing a hat is closer to the viewer than the tree, whether our friend John is in the picture, whether John looks happy in that picture, whether the picture is funny or sad, how many people are riding a bicycle, and so on. If we cannot distinguish whether the answers come from the human or the computer, we can claim victory. We claim that, from a behavioral standpoint, humans interpret images in the same way as the computer vision algorithm.

A few clarifications and further specifications are pertinent here. If someone asked me questions about an image, and the questions were posed in Chinese, I would not be able to answer the questions. This is not a failure of my visual system; this merely shows that I cannot speak Chinese. I would pass the Turing test for vision, but I would not pass a Turing test for Chinese! Therefore, the definition of the Turing test for vision assumes that we have some way of encoding the questions and answers in a format that the computer understands. For example, if we ask whether John appears to be happy or not, the computer needs to be able to interpret what "happy" means. We seek to circumscribe the Turing test strictly to visual processing and dissociate it from language understanding.

Language is, of course, another fascinating aspect of cognition, and we want computers to be able to use language too. One could even extend the Turing test to include both vision and language. For example, we will briefly discuss later in this chapter the task of image captioning – that is, coming up with a short description for an image. However, the main concern in this chapter is to pass the test of visual processing. Therefore, we define the Turing test strictly in the domain of vision. We still want the computer to be able to answer *any* question, but we are not going to be concerned with whether the computer knows the words and the grammar in the question or not.

For a computer to answer whether John appears to be happy or not, one would need to train the computer with pictures rendering happy people and pictures rendering people who do not look happy. Alternatively, we could figure out some other ways to educate the computer about what happy people look like. This training to interpret the task holds for all other questions as well. If we want to know whether a woman is riding a blue bicycle, the computer needs to understand what woman, riding, blue, and bicycle mean. Of course, the same holds for human vision, even though we tend to take this for granted and underestimate this obvious point. In the same way that I would fail in answering questions in Chinese, if we ask a human whether there is a *beldam* in the picture, the person will not be able to answer unless they understand what the word beldam means (beldam is an archaic noun meaning an old woman).

It is important in this definition that the number of questions remains infinite. For example, one could build a computational model that excels at recognizing whether our friend John is in the picture or not; that is, a perfect John detector that can recognize John even better than we do. Such a computational model would be quite nice, but it would not pass the Turing test for vision. Similarly, one could build a model that can label every pixel in the image (this pixel is part of a tree; this pixel is part of a red car; this pixel is part of John). Such a model would be even more impressive, but it would not be able to answer any arbitrary question about the image, such as whether John is happy or not, and therefore, the model would not pass the Turing test for vision either.

While the Turing test, as defined thus far, focuses on *human* vision, we can also define a Turing test for rat vision, meaning an algorithm that is indistinguishable from a rat's behavior in visual tasks. We can also define a Turing test for visual processing of a one-year-old infant, meaning an algorithm that is indistinguishable from the behavior of a one-year-old human infant. Similarly, some people may possess rather specialized knowledge, like a bird watcher who can classify different types of birds or a doctor who can diagnose certain conditions based on clinical images. One could define restricted versions of the Turing test for those cases, such as a machine that cannot be distinguished from a world expert bird watcher in terms of classifying birds from images.

9.2 Computer Vision Everywhere

Despite enormous progress in computational modeling of visual processing, we are still far from being able to build algorithms that can pass the Turing test for vision. Most computer vision studies focus on specific sets of questions or tasks en route toward building systems that can pass the general Turing test. Many exciting algorithms have been developed to address several interrelated problems in computer vision (Figure 9.2).

One of the most common tasks is *object classification* (Figure 9.2A): the computer is presented with an image, and it has to produce one of a fixed number of possible labels. For example, does the image contain a tree [yes | no]? Which of the following objects is in the image: [people | tree | building | flower]? Another instance of object classification is the task of clinical diagnosis based on images; for example, does the mammogram image

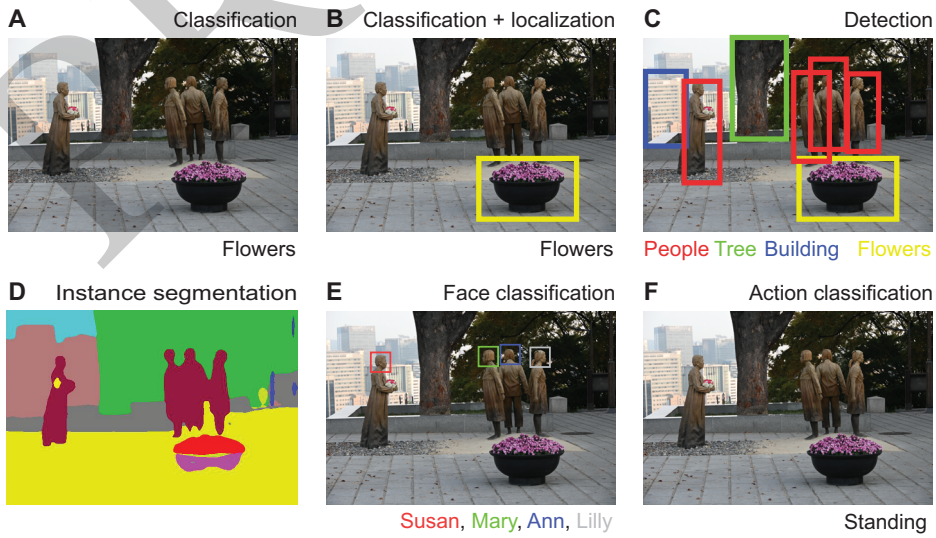

Figure 9.2 Typical computer vision tasks. (**A**) Object classification. (**B**) Object classification and localization. (**C**) Object detection. (**D**) Instance segmentation. (**E**) Face classification. (**F**) Action classification.

contain a tumor [yes | no]? Yet another instance of object classification is the task of face recognition (Figure 9.2E); for example, is [Susan | Mary | Ann | Lilly] in this image?

When assigning a label to an image, those labels could be nested into structures and hierarchies. For example, some psychologists refer to object *categorization* (does the image contain a car or a face?), as distinct from object *identification* (which particular car is it, which particular face is it?). From a computational standpoint, these are essentially the same problem, and it is possible to design hierarchical algorithms that will answer these questions sequentially or in parallel.

An intriguing and ubiquitous aspect of human language is the definition of categorical distinctions that transcend the exact visual features in the image; the notion of semantic categories was discussed in Section 6.8 (Figure 6.5). For example, we can put together images of ants, snakes, lions, birds, and dolphins and categorize them as animals. If we train a computer vision that excels at recognizing ants and snakes, *exclusively* ants and snakes, the algorithm may not be able to understand that a bird is another type of animal. This failure to extrapolate to another animal may seem like a significant problem for computer vision: of course, essentially any human can tell that a bird is an animal. However, it is unclear whether humans could succeed in this same task, with the same type of training that the computers were subject to. Imagine a person who is an expert in ants and snakes but has never seen *any* other animal. Given a picture of a bird (without movement, without contextual information, or any other cue; remember that we want to match the human task to the computer task as closely as possible; otherwise, humans have an unfair advantage), would the person be able to understand that the bird is another type of animal? One may think that the answer is yes. However, it is difficult to imagine what his or her understanding of "animalness" would be if their entire visual expertise were restricted exclusively to static pictures of ants and snakes. We often tend to underestimate the amount of visual experience that we have.

Another version of object classification is the problem of *object verification*: given two (or more) images, the task is to determine whether the images correspond to the same object or not. For example, the airport security officer may examine a passport and the person in front of him or her, and assess whether the person matches the picture or not. Yet another related problem is that of image retrieval; given an image, retrieve all instances of similar images from a dataset. For example, one may want to retrieve all the images on the web that are visually similar to a given picture.

Extending the task of object classification, algorithms have been developed for *object detection* or *object localization* (Figure 9.2B and C). In these tasks, the goal is to place a bounding box around the object of interest in an image. For example, "locate all the pedestrians in the image." Progress in object localization rapidly accelerated with the development of the MSCOCO dataset, which contains detailed tracing contours around objects from 80 common categories. One example of object detection is the ability to put a box around a face in an image (face detection), 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 tremendous boost to the possibility of developing self-driving cars, which are equipped with sensors to detect other cars, pedestrians, car lanes, and many other objects of interest.

Related to the problem of object detection is the question of *object segmentation,* where the goal is to trace the contour of a given object (Figure 9.2D). An initial map of segmented objects in an image can be extracted by adequately detecting edges. However, more complex problems often involve a deeper understanding of the interrelationships among different object parts. An example of a challenging problem for object segmentation is the case of a zebra: the algorithm should separate the zebra as a whole, rather than marking every stripe as a separate object. Another typical challenge in segmentation arises when there is occlusion. For example, consider the rotated B letters in Figure 3.8: the object segmentation algorithm should isolate every letter rather than merely mark each letter fragment as a separate object. Investigators may be interested in algorithms to segment all the objects in an image rather than localizing every single object of a specific class. *Semantic edge detection* refers to drawing the outlines of objects in an image without labeling edges that do not separate objects.

There has been extensive discussion in the literature about the chicken-and-egg problem of whether segmentation comes before recognition or whether recognition comes first. When there are depth boundaries defined by stereo and motion discontinuities, segmentation may occur early, prior to recognition. However, when the only cues are based on luminance, there is no clear biological evidence for segmentation taking place prior to recognition or vice versa. It is likely that both computations happen in parallel. In many practical applications, object classification, detection, and segmentation are often combined.

An example application combining all three tasks involves analyzing microscopy images in cellular biology. Biologists are interested in an algorithm that can automatically detect cells with a given shape, mark them with a given color, and count them. A particularly difficult and exciting challenge along these lines was advanced by a community of researchers working toward mapping connectivity in the nervous system based on electron microscopy images (Figure 9.3). These images consist of section after section of high-resolution rendering of the inner structure of nervous tissue; the goal is to automatically trace the connectivity of every neuron from these images. *Instance segmentation* refers to separating and labeling every pixel in an image. For example, we want to label every neuronal dendrite, soma, axon, glial cell, and other cell types in the electron microscopy images. We especially want to follow dendrites and axons across multiple sections to map where they originate and where they synapse onto another neuron.

Action recognition refers to the ability to identify actions in an image or video (Figure 9.2F, Figure 9.4). Is a person playing soccer [yes | no]? Which of these actions is the person performing [playing cello | brushing teeth | bowling | soccer juggling]? Action recognition can be based on individual images, but it has also triggered the development of databases based on videos. In sports, people are interested in building computer vision systems that can automatically analyze the game in excruciating detail, including detecting individual players, tracking them, and identifying what they are doing (e.g., running with the ball, passing the ball, dodging the opponent, or shooting).

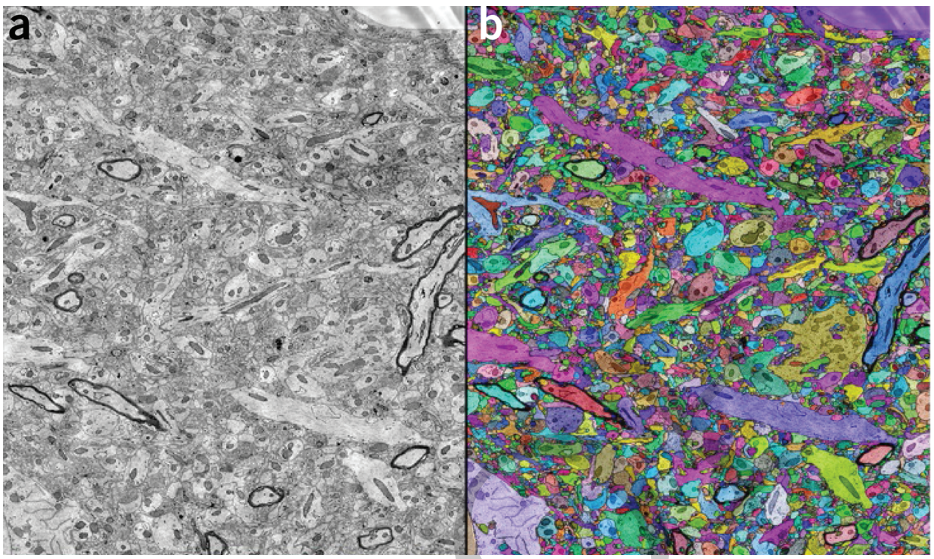

Figure 9.3 Image segmentation algorithms can help map neuronal connections. (**A**) Electron micrograph from a 40 × 20μm section of mouse cerebral cortex. (B) Automatic computer segmentation, where each cellular object is shown as a separate color overlaid on the original image. Reproduced from Lichtman et al. 2014

Action recognition and tracking are examples where many of the computer vision tasks defined earlier are intertwined and need to be combined. Action recognition applications have also become widespread among biologists studying animal behavior. Traditionally, quantifying animal behavior has been a tedious and time-consuming 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 long hours of scrutinizing those videos and subjectively describing the animal's behavior. Nowadays, some systems can objectively and reliably perform these types of annotations: computer vision approaches can automatically analyze the videos, quantify the amount of time spent in different behaviors, and describe the sequence of different types of movements. Yet another widespread application for action recognition systems is surveillance. One may be interested in detecting "anomalous" behavior near a house, at an airport, or at a crowded concert. Computer vision scientists refer to this problem as *anomaly detection*.

Action recognition is a good example to illustrate how experimental design and databases can make tasks easy or hard. Distinguishing whether someone is playing the cello or juggling a soccer ball based on the types of images shown in Figure 9.4A can be easy. However, determining whether a person is reading or not based on the types of images shown in Figure 9.4C can be substantially harder. We will discuss this point again in Section 9.10.

The list of computer vision applications is so extensive 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.

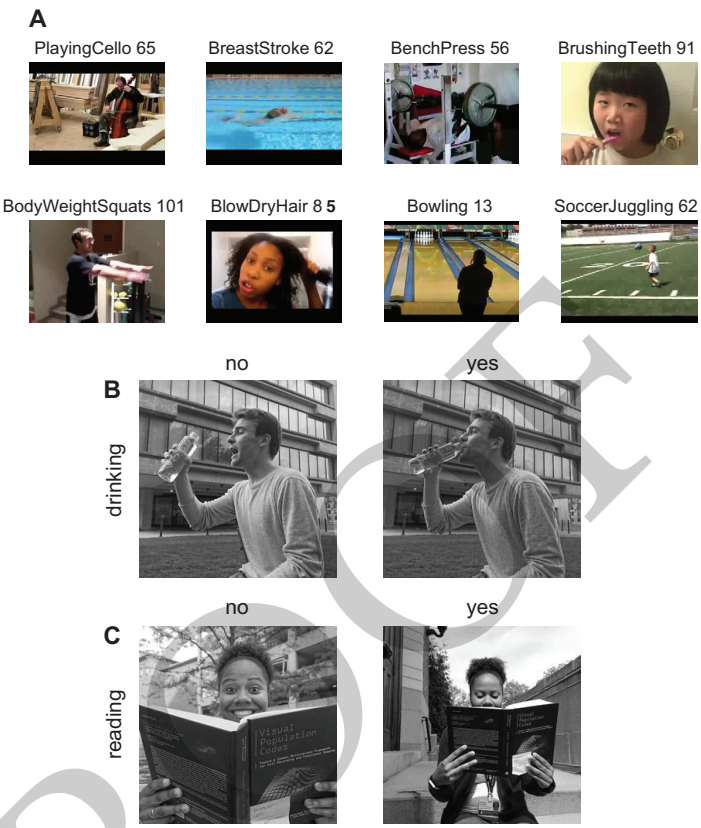

Figure 9.4 Dataset design can make problems easy or hard in action recognition. (**A**) UCF-101 dataset of videos with labeled actions (Soomro et al. 2012). The first frame in eight examples out of the 101 action categories are shown here. Titles indicate the category number and description. (**B**)–(**C**) A challenging dataset for action recognition where subjects need to indicate in a binary fashion whether a subject is drinking or not (**B**), or reading or not (**C**).

9.3 Incorporating Temporal Information Using Videos

Historically, many computer vision studies have been restricted to analyzing static images. In part, work has focused on static images because both humans and machines can recognize objects in images quite well. The focus on static images is also partly a historical accident: it was easier to create databases with static images, images occupy less hard drive space, and they require fewer computational resources to process. These practical restrictions are less relevant today.

Under natural viewing conditions, there are several cues that depend on integrating information over time. These dynamic cues can significantly enhance object classification. A paradigmatic case where temporal integration can be essential is action recognition. Although it is possible to recognize actions purely from static images (e.g., Figure 9.4), it is generally significantly easier to do so using videos both for

computers and for humans. For example, it can be difficult to discern whether a person is talking or not using only a static image. Modern models for action recognition from spatiotemporal input based on deep convolutional neural network architectures can be partitioned into three groups: (i) networks with three-dimensional convolutional filters, where spatial and temporal features are processed together via three-dimensional convolutions; (ii) two-stream networks where one stream processes spatial information and another stream obtains optical flow from consecutive frames, and the two streams are merged at a late stage for classification; (iii) networks that feed onto a recurrent architecture such as a long short-term memory (LSTM) (Section 8.17) that integrates spatial features over time.

Temporal information is relevant for many other tasks beyond action recognition. Object segmentation generally becomes significantly easier with video data. The importance of temporal change for segmentation has been exploited by the ubiquitous use of camouflage in the animal world. In the absence of movement, matching colors, contrast, and textures can help animals avoid predators, or at least buy sufficient time to escape. It is particularly challenging to segment objects in the visual periphery, yet neurons with receptive fields located at large eccentricities remain highly sensitive to visual motion. Furthermore, motion is one of the most robust bottom-up saliency cues.

Temporal information can also play a critical role in visual learning. In an elegant experiment, cats were reared under stroboscopic lighting conditions – that is, with flashes of lights turning on and off like those used at a disco, which prevent seeing continuous motion. The development of the primary visual cortex in those cats was abnormal in terms of orientation selectivity, binocular integration, motion detection, and receptive field sizes. These results further corroborate the discussion in Section 2.2 about natural stimulus statistics governing the tuning properties of neurons in the visual system.

Additionally, because objects do not just simply vanish instantaneously, using video data can naturally help humans and models learn to recognize objects from multiple viewpoints. Video sequences automatically provide a biologically plausible way to perform "data augmentation" by getting many similar images of an object from a single label (Section 8.9). Another example of how temporal information can be used for visual learning is the case of self-supervised learning to predict future events, discussed in the PredNet algorithm in Section 8.17 (Figure 8.11).

9.4 Major Milestones in Object Classification

In Section 8.7, we introduced several image databases, such as ImageNet, which have played an essential role in the development of computational models of visual recognition (Figure 8.4). These databases were created for large-scale visual recognition challenges where investigators compete to get low classification errors.

A good way to report performance in these competitions is to cite top-1 classification accuracy where the model produces a single label per image, and the result is either right or wrong. Many computer vision applications have reported a more lenient and more confusing metric: top-5 classification accuracy, where the model is allowed to produce

five different labels for each image, and the result is considered to be correct if any of these labels is correct. One excuse for considering the top-5 metric is that some natural images extracted from the web contain multiple objects. An image may contain both a dog *and* a tree; the association between that image and a label of tree is therefore arbitrary. The same image could have easily been labeled dog as well. While this makes sense, reporting top-5 accuracies exaggerates the accuracy of the algorithms and makes it more difficult to directly compare against human performance. For example, consider an image from the ImageNet dataset (where there are 1,000 possible labels) showing exclusively a tree in the street. The image label is "tree." A computational algorithm may provide the following five labels, sorted in decreasing probability order given by the numbers in parenthesis: elephant (probability = 0.62), refrigerator (0.31), car (0.02), tree (0.02), ice (0.01). These probabilities add up to 0.98 and not 1 because the remaining 1,000 − 5 = 995 categories add up to 0.02. These five labels would be considered a correct answer according to the top-5 accuracy measure, yet they are somewhat strange. Humans would not say that the image has 0.62 probability of containing an elephant and 0.31 probability of containing a refrigerator! Other databases like MSCOCO label multiple objects per image, and therefore, it is possible to check the accuracy of multiple labels.

Figure 9.5 shows top-1 performance in ImageNet for several computational models, many of which have won object classification competitions over the last decade, and some of which were already mentioned in Chapter 8. Current top-1 performance is slightly greater than 80 percent, and current top-5 performance is almost 95 percent. These metrics are quite impressive, considering that there are 1,000 classes and, hence, chance level is 0.1 percent. It is not easy to directly compare these performance metrics with humans, particularly top-5 measures, given the arguments in the previous paragraph. Humans are not very good at 1,000-way classification: it is hard to remember those 1,000 labels, and

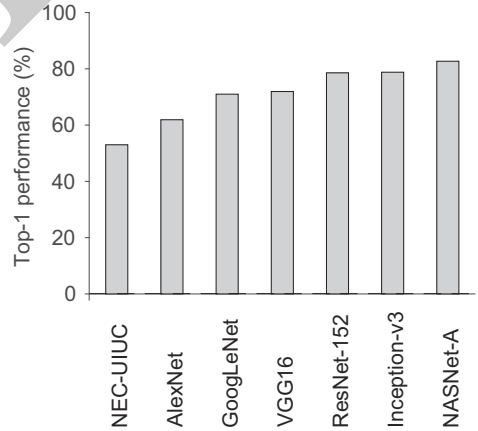

Figure 9.5 Evolution of performance on the ImageNet dataset. Top-1 classification performance in object classification based on the ImageNet dataset. Each column refers to a different computational algorithm. Chance = 0.1 percent.

humans may have lots of biases toward remembering and using some labels more than others. Additionally, as we discussed in Section 8.7, some of the image categories in ImageNet are somewhat esoteric (how many times have you seen an isopod, a jetty, or a cuirass?). Humans could potentially be trained in the same way that the algorithms in Figure 9.5 have been trained to become experts at distinguishing an isopod, a jetty, a cuirass, or any of the other 997 labels. Regardless of these considerations, informal measures of human performance in this dataset yield accuracy rates that are between 90 and 95 percent. Hence, even with all their limitations, current algorithms can perform object classification on ImageNet images as well as or even better than humans.

It should be noted that top-1 performance is not always a great metric. For example, in the next section, we will consider the problem of analyzing clinical images. Consider a particular disease that is present in one out of 10,000 people. Suppose that we train an algorithm, and the algorithm achieves 99.99 percent performance. At first glance, this performance seems quite impressive. However, it is easy to achieve 99.99 percent performance by simply indicating that all the images do not show evidence for the disease! Trivially, such an algorithm would not be useful at all. The algorithm would have 9,999 true negatives, 0 true positives, 1 false negative, and 0 false positives. Particularly in situations where there is a difference between the number of images with each label (an imbalanced classification problem), it is useful to define two metrics, precision and recall:

$$\text{recall} = \text{true positives}/(\text{true positives} + \text{false negatives})$$

$$\text{precision} = \text{true positives}/(\text{true positives} + \text{false positives})$$

. An algorithm stating that none of the images show the disease has zero recall and zero precision, even though it reached 99.99 percent accuracy. Conversely, consider another algorithm that is also *not* useful, which labels all the images as showing evidence for the disease. This algorithm would have 0 true negatives, 1 true positive, 0 false negatives, and 9,999 false positives. The recall would be 1 – which may seem quite nice, except that the precision would be very low, despite the high recall. The same ideas are often discussed in statistics classes as Type I error (false positives) and Type II error (false negatives). For the aficionados, some investigators also use another metric called the F_1 score, which is the harmonic mean of the precision and recall: $F_1 = 2\frac{precision \cdot recall}{precision + recall} = \frac{TruePositives}{TruePositives + 0{:}5(FalsePositives + FalseNegatives)}$.

Depending on the nature of the problem and the consequences of errors, false positives could be much worse than false negatives, or vice versa. It is possible to assign weights in loss functions to differentially penalize the different types of errors. For example, if recall is considered to be β times as important as precision, one can define $F_\beta = \left(1 + \beta^2\right)\frac{precision \cdot recall}{\left(\beta^2 \cdot precision + recall\right)}$ (which is equivalent to F_1 when $\beta = 1$).

Independently of the specific metrics, it is clear that there has been notable progress in object classification tasks (Figure 9.5). AlexNet itself showed a substantial boost with respect to all its predecessors, giving rise to a rapid exploration of deeper and more complex architectures that have boosted performance by more than 20 percent in less

than a decade. This notable improvement in academic competitions attracted the attention of many people looking to solve pattern-recognition applications.

9.5 Real-World Applications of Computer Vision Algorithms for Object Classification

Success in image labeling competitions inspired a large number of efforts in image classification across many domains. One of the earliest real-world applications was optical character recognition (OCR), which rapidly became mainstream in sorting mail based on the handwritten zip codes. Now, there are even neat applications that can translate handwritten traces into mathematical formulae. On the one hand, some mathematical symbols are relatively simple; on the other hand, mathematical symbols are probably less stereotyped, and there is less training data than in other OCR applications. Computer vision algorithms have already made rapid progress in a wide array of exciting applications; we discuss next only a few examples.

A field that is rapidly being transformed by computer vision is clinical image analysis. Clinical diagnosis based on images can *sometimes* be simplified into a visual pattern-recognition problem. Clinicians may combine information from image-based diagnosis with a wealth of other information – including medical history, genetic information, symptoms, and more. How to combine these different sources of information into automatic diagnosis methods is an interesting problem in and of itself, but this is beyond the scope of our current discussion. Here we restrict the problem of diagnosis strictly to image analysis. For example, a radiologist can examine a mammogram to determine whether it contains a breast tumor or not (Figure 9.6). A database consisting of many mammogram images annotated by experts can be readily used to train computer vision algorithms. The American Cancer Society recommends obtaining a mammogram, generally consisting of two X-ray images of each breast, to all women

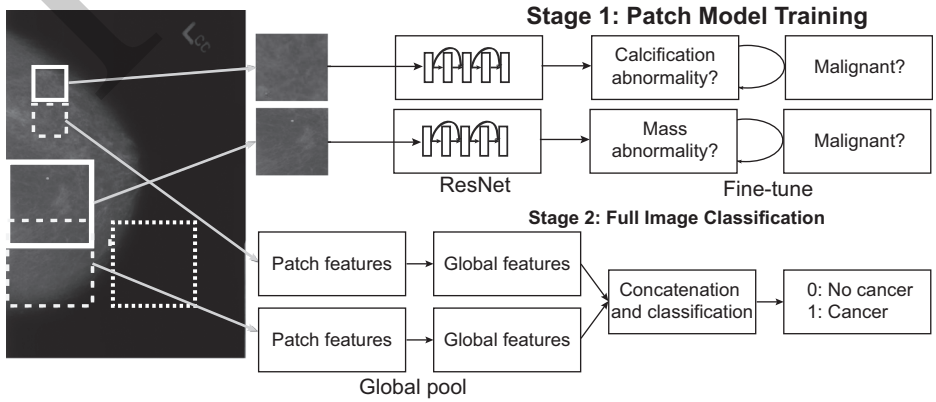

Figure 9.6 Computer vision can help clinical diagnosis based on images. Example algorithm to detect cancer in breast mammograms. Modified from Lotter 2018

once or twice a year, depending on age. This number of mammograms leads to a lot of images (about 40 million images a year in the United States alone). The problem is important because early diagnosis can have a critical impact on deciding the course of action. It is estimated that radiologists read on the order of 10,000 cases per year; a radiologist with three decades of experience may have seen 300,000 cases. Nowadays, a computer vision algorithm can be trained with many more examples than a human clinician can see in his/her lifetime.

Computer vision algorithms have thrived in a wide variety of image diagnosis efforts. To train and test these computer vision algorithms, ground truth labels provided by clinicians are needed. It should be noted that humans are capricious creatures. Clinicians do not always agree with each other on the diagnosis of a given image (between-expert variability). Furthermore, clinicians sometimes do not even agree with themselves when repeatedly tested on the same images (within-expert variability)! In the case of breast tumor detection, computational algorithms are now on par or even better than human clinicians. In other words, the differences between a state-of-the-art computer vision algorithm and a human expert are the same as the within-expert and between-expert variability. Future generations may regard humans trying to diagnose images in the same way that we now regard a human trying to interpret a bar code in the supermarket or trying to compute the square root of 17 by hand.

While the presence or absence of a tumor is the central question of interest in the vast majority of breast exams, occasionally, there may be other relevant questions clinicians may want to ask about an image. For example, sometimes there are incidental findings where a person is scanned to diagnose a given condition X (e.g., breast cancer), the scan does not reveal any finding regarding X, but the radiologist detects other anomalies that lead to a different diagnosis Y. Such incidental findings may be challenging for current computer vision algorithms because they may be extremely infrequent. The algorithms are ultra-specialized and outperform radiologists in detecting condition X but were never trained in detecting the rare condition Y. One possible compromise as an initial solution for this challenge would be for computer vision systems to flag such images as anomalous and route them back to a human for further inspection.

Incidental findings represent one arena where humans may still surpass machines in clinical image diagnosis, where humans can find patterns that computers miss. The reverse is also true: machines may be able to discover novel patterns that were not previously found by humans in clinical images. An intriguing example of this phenomenon arose when investigators were developing computer vision approaches examining retinal fundus photographs to diagnose a condition known as diabetic retinopathy (Figure 9.7). Diabetic retinopathy is a condition that may arise in diabetic patients when high blood sugar levels cause blood vessels in the retina to swell and leak. These blood vessels can be examined in fundus photographs, which are images of the back of the eye, used by ophthalmologists to diagnose the disease. After collecting hundreds of thousands of labeled images, a deep learning computer vision algorithm quickly learned to match clinicians in diagnosis, a feat that comes as no surprise at this stage.

The diagnosis label is only one of the questions that one can pose about those images. The investigators decided to turn their machine learning algorithms to other questions

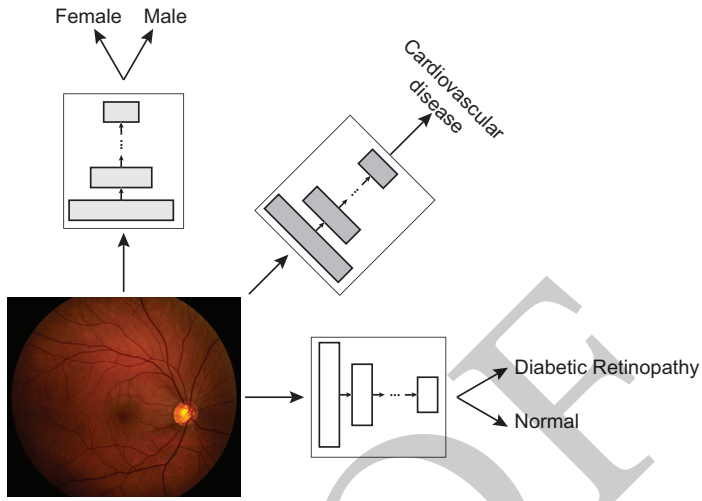

Figure 9.7 Computational algorithms can make new observations. Example clinical application of computer vision, taking a photograph of the back of the eye (fundus photograph) and using a deep convolutional network to diagnose diabetic retinopathy (Poplin et al. 2018). In addition, computer vision algorithms can be trained to ask other questions from the same image, including predicting the subject's gender or even the risk of cardiovascular disease.

on the same images. In a surprising twist, computer scientists 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 age. The algorithms were able to predict age quite accurately, with an absolute error of less than 3.5 years. Next, the investigators assessed whether they could predict the subject's gender. Surprisingly, they were able to do so exceptionally well, with an area under the receiver operating characteristic (ROC) curve of 0.97. The ROC curve is a plot of the probability of correct detection versus the probability of false alarm. It is trivial to achieve high detection rates at the expense of high false alarm rates (by claiming that every image shows disease; see previous section) or low false alarm rates without any correct detection (by claiming that no image shows disease). A good algorithm will have a low false alarm rate and high probability of detection. The best that an algorithm could achieve is an area of 1.0; chance levels would yield an area of 0.5. Trained ophthalmologists had never been able to estimate somebody's gender or age from fundus photographs. Perhaps they never cared to ask that question; after all, the clinicians will have the subjects and their records right in front of them. However, even after telling clinicians that the gender and age information was present in these images and asking doctors to infer the gender or age, they were unable to do so. It is not entirely clear what exact image features the algorithm uses to discriminate gender or age. One could hypothesize that perhaps doctors, both male and female, might position the apparatus to take fundus photographs slightly closer to female patients than to male patients, on average, when acquiring these images. The algorithms could well capture such a slight unconscious bias. Alternatively, perhaps there exist real subtle differences

between female and male blood vessels in the retina. Regardless of whether this explanation holds, this example shows that computer vision can discover image features that are not apparent even to experts in the field.

Estimating a subject's age and gender from fundus photographs is perhaps not particularly exciting from a practical standpoint. The most enigmatic finding emerged when the investigators decided to ask an even more daring question: would it be possible to predict the risk of cardiovascular disease from fundus photographs? Computer scientists discovered that they were able to predict cardiovascular disease from the fundus photographs with an area under the ROC curve of 0.7. This result is quite remarkable because this is a question that ophthalmologists had not thought about, it is a question that is extremely relevant from a clinical standpoint, and the computational analyses constitute additional information that comes for free from the fundus photograph without any additional clinical testing. What is perhaps even more remarkable is that the computer vision algorithm was able to predict cardiovascular disease better than the Framingham Risk Score, which is considered to be one of the best indicators of cardiovascular risk based on decades of clinical work. Computer vision algorithms can not only learn to diagnose images like doctors, but they can also teach us novel things about those images.

There are several situations where there is an enormous number of images (or videos) that needs to be classified. Automatic image classification has found applications well beyond clinical diagnosis. For example, computer vision has shed light on the gargantuan task of classifying galaxies and exoplanets from telescope images. There are vast amounts of imagery to help us understand the shape of galaxies and characterize planets outside the solar system, but we do not have enough astrophysicists to classify all those images. Astrophysicists turned to crowd-sourcing by engaging the public in looking at images and learning to categorize galaxies. This is an ideal setting to apply pattern-recognition techniques from computer vision: the last few years have seen many exciting discoveries made by machine learning algorithms. A conceptually similar example is the categorization of plants and animals. 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. There is a wide variety of applications for automatic face-recognition algorithms. Many smartphones have algorithms that use faces to log in, which used to be the domain of science fiction movies not too long ago. Facebook can now search for photos that include a particular person when that person is not tagged. Quantitative studies of face identification have shown that computer vision systems are better than forensic experts and also better than so-called superrecognizers, people with an extraordinary capacity to recognize and remember faces. There is also a growing industry of security applications based on facial recognition capabilities. Security applications in the near future may also rely on action recognition classification algorithms. Concomitant with advances in face recognition, there are vigorous and timely discussions about issues of privacy. It is quite likely that, very soon, it will be rather challenging to walk down the street without being recognized.

George Orwell's Big Brother scenario with cameras that can recognize people is now technically feasible.

The exciting progress in self-driving cars has also been fueled by progress in computer vision – with tasks such as localizing pedestrians, cars, brake lights, traffic lights, other signs, lanes, the sidewalk, and even animals, bicycles, or anomalous objects on the road. While the majority of computer vision applications rely on video or camera feeds from regular cameras, images do not have to be restricted to such sensors. For example, self-driving cars can simultaneously use information from multiple cameras and many other sensors. There has been so much progress in terms of computer vision that most engineers trying to build self-driving cars think that the main challenges ahead transcend vision and involve decision making, legal issues, and vulnerability.

Other applications of computer vision algorithms are still under development but will be ready quite soon. For example, there is much interest in intelligent content-based image or video search (referred to as image retrieval in the computer vision literature). Searching the web by content (as opposed to searching for the word "dog" and using the label to search for text or images with a dog tag) opens the doors to a whole set of applications. Initial prototypes of these types of searches are already in place.

The previous section introduced advances in face identification. These algorithms will allow searching for people from photographs, which may have a lot of exciting applications such as searching for missing people or finding a friend from long ago. Progress in face identification may soon lead to ATMs that can recognize customers. Cars and houses may also soon recognize their owners from their faces. Progress in person recognition and action recognition may radically transform security screening in crowded environments, including airports, stadiums, and perhaps every street in large cities. Efforts in computer vision applications for security screening, and perhaps other purposes, are already ongoing in several major cities.

9.6 Computer Vision to Help People with Visual Disabilities

A particularly exciting application of computer vision systems is to help people with visual deficits, particularly the blind (Figure 9.8). In the United States alone, there are approximately one million people who are legally blind and about 3.25 million people with visual impairment. Combined with high-quality and relatively inexpensive cameras, computer vision algorithms can help digest the output of digital cameras to convey information to the blind. Most phones these days can determine a person's location by using GPS coordinates, yet one may soon be able to get even more precise information by pointing the phone and having it determine the direction of certain shops, bus stops, or landmarks. Phones can also help read signs and restaurant menus. However, blind people need and deserve much more.

An interesting application of computer vision would be to restore visual functionality to people with severe visual impairment. By restoring "visual functionality," we do not necessarily mean getting a blind person to *see* in the same way that a sighted

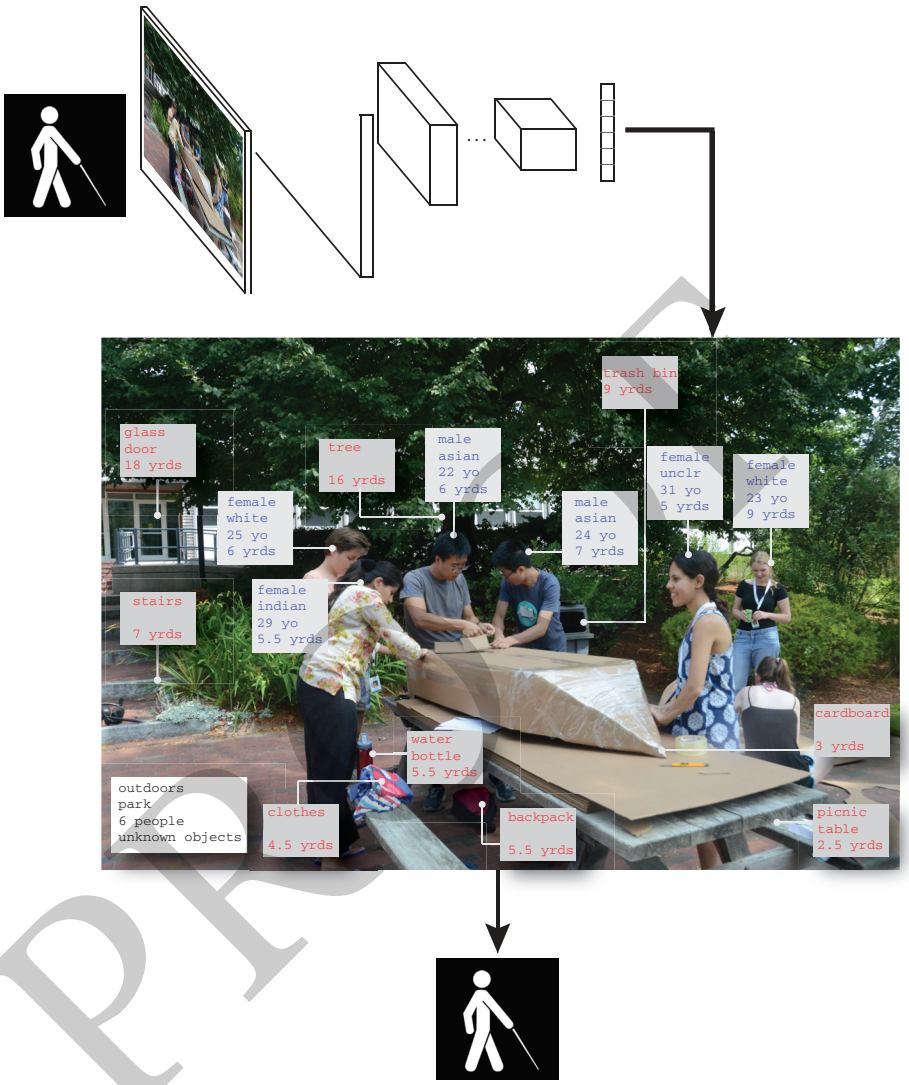

Figure 9.8 Computer vision could help visually impaired people. Example potential approach to use computer vision to help people with visual impairment. A blind person may carry a camera that connects to a computer vision algorithm and that can interpret the surrounding scene. The computer vision algorithm can deliver information about people, objects, distances, and relative locations in real time.

person does. Instead, visual functionality refers to the ability to rapidly and accurately convey information that blind people can use. A blind person could easily wear a camera on their forehead, or in a pendant. Imagine an algorithm that can label every object in an image (instance segmentation). How can we convey such rich information to a blind person? An image is worth a thousand words. In a glimpse, we get a rich representation of our surroundings, which is quite different from labeling every object.

This representation highlights certain aspects of the image while ignoring other, less relevant information. For example, we may not be interested in the shape of every branch in a nearby tree, though we could access that information by attending to it if we wanted to. Instead, we may be more interested in whether a bicycle is coming toward us at full speed. In a glimpse, we can discern distances, relationships between objects, and even actions and intentions. Even if we could accurately label all the objects in an image, there is much more to visual understanding, a theme that we will come back to at the end of this chapter. The main challenge in helping the blind is to provide *relevant* information in real time.

As a side note, we could easily extend these ideas to enhancing the visual capabilities of sighted people as well. It would be easy to wear a camera that would give us immediate access to a 360-degree view of the world, or grant us access to other parts of the light spectrum that our eyes are not sensitive to, such as infrared. We are all "blind" in the infrared and ultraviolet frequency bands, or behind our heads, but we have instruments that can detect those signals. Computer vision systems could help us parse and interpret those images. Of note, the basic operations of convolution, normalization, pooling, and rectification (Section 8.5) do not depend on whether the signals come from the visible part of the spectrum or infrared, ultraviolet, or other sources. In sum, computer vision could help restore, and perhaps even augment, human vision.

9.7 Deep Convolutional Neural Networks Work Outside of Vision Too

The same mathematical operations used to analyze images taken from photographs can be extended to non-visible parts of the spectrum. Furthermore, there is no reason to restrict ourselves to light patterns. Although our focus is the discussion of computer vision systems, it is interesting to point out that the same mathematics, the same types of architectures, and the same types of training algorithms have extended well beyond vision.

Vision has led the way to success in a wide variety of other problems. For example, systems for speech recognition; systems that suggest automatic replies to emails; systems to predict the weather, the stock market, or consumer behavior; and many other questions have now been revolutionized by deep convolutional neural networks originally developed to label images. Each of these domains requires training with different types of data, changing the inputs, and, in some cases, also making adjustments to the architectures themselves. However, at the heart of these domains outside of vision is a similar mathematical problem: training a neural network to learn to extract adequate features from the data and then classifying the resulting features. What changes is the input: instead of using pixels in RGB space, in the case of speech recognition, one can use a spectrogram of the frequencies of sound as a function of time to process sounds. However, the subsequent processing steps and the procedure to train those algorithms are remarkably similar, if not exactly the same, in many applications.

In neuroscience, the idea that similar computational principles can be used for different problems is sometimes phrased as "cortex is cortex" (Section 8.2), alluding

to the conjecture that the same basic architectural principles are followed in the visual, auditory, and tactile systems. Without a doubt, there are important differences across modalities, and engineers will also fine-tune their algorithms for each application. However, as a first approximation, some of the primary ingredients seem to hold across multiple seemingly distinct tasks.

9.8 Image Generators and GANs

The basic paradigm in most of the computer vision applications that we have discussed thus far follows the structure shown in Figure 8.2. An image is processed through a neural network that learns to extract features for the task at hand. Another remarkable development from deep convolutional neural networks has been the idea of turning this process in reverse and using features to *generate* images. The computational models discussed so far are discriminative algorithms that assign descriptive labels to images or parts thereof. In contrast, the goal of generative algorithms is not to assign a label but rather to create a new sample from a given distribution. In the context of vision, this typically amounts to creating novel images or videos. A particularly successful approach to generating images is the use of generative adversarial networks (GANs, Figure 9.9).

GANs consist of two main components: an image generator, and an image discriminator. The image generator can be thought of as an inverted deep convolutional neural network. In a typical deep convolutional neural network, the input is an image, and the output is a series of features. In an image generator, the input is a series of features, and the output is an image. For example, using random initial inputs, the goal may be to create images of realistic faces. The image discriminator takes as input both real images and images created by the generator; the task of the discriminator is to ascertain whether

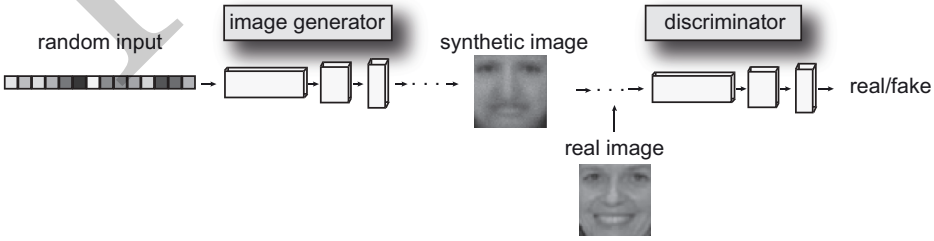

Figure 9.9 Generative adversarial networks (GAN) play police-versus-thief games. A generative adversarial network is an algorithm that creates new samples from a given distribution – for example, generating new images. The algorithm consists of two main components: an image generator and an image discriminator. The generator can be thought of as an inverted deep convolutional neural network, using features as inputs and creating images as output. The discriminator takes samples from the generator and real images and determines whether the generated images are real or fake.

an image is real or fake. The two components are jointly trained – the generator trying to fool the discriminator and the discriminator trying to catch the impostor generator.

Such image generators have found fun applications in several domains. One of these domains is style transfer. One can take an arbitrary picture and re-render it according to the style of a famous painting. One can use a GAN to merge different faces, to make a face look like a celebrity, or to visualize how a given person might look like when he or she gets older. Another application is to create graphic art. Recently, an image generated by a GAN, *The Portrait of Edmond Belamy*, was sold by Christie's for the sizable prize of $432,500.

Other GANs have focused on trying to create realistic-looking photographs. In fact, to the naïve eye, it can be difficult to distinguish a fake from a real photograph. Beyond Hollywood, these algorithms raise a lot of interesting questions. The notion that "seeing is believing" may require some serious revision in the era of sophisticated digital fakes.

9.9 DeepDream and XDream: Elucidating the Tuning Properties of Computational Units and Biological Neurons

A particularly exciting use of image generators is to help address the curse of dimensionality when studying the tuning properties of neurons in visual cortex (Figure 5.10). A family of techniques initially referred to under the poetic name of *DeepDream* was introduced by computer scientists to visualize the types of images preferred by units in deep convolutional neural networks. When considering these neural networks, we know the architecture and all the weights; in other words, we can mathematically define perfectly well the activation of every unit. Under these conditions, we can reverse the process to ask what types of images will yield high activation for a given unit. Here the "loss function" is the unit activation (which is to be maximized), and we can still apply the gradient descent algorithm introduced in Section 8.6, except that we calculate derivatives with respect to the image itself instead of changing the network weights.

Now imagine that we want to generate images that will maximally activate a neuron in the brain rather than a unit in a neural network. The situation is far more complicated when it comes to the neural networks in biological brains, where we do not know the architecture, let alone the weights. To circumvent these challenges, Will Xiao and colleagues developed the XDream algorithm (e**X**tending **D**eepDream with **r**eal-time **e**volution for **a**ctivation **m**aximization, Figure 9.10), which was briefly introduced in Section 6.4. The algorithm consists of three components: (i) an image generator, (ii) a mechanism to assess the fitness of each image, and (iii) a search method to create the next set of images (Figure 9.10A). The image generator is an inverted deep convolutional neural network along the lines of the algorithms introduced in the previous section. The image generator takes a set of features as input and creates a color image. The initial conditions are random images. Next, the algorithm evaluates the images created by the generator and rank orders them according to a fitness function defined by what we want to maximize. For example, the algorithm may maximize the activation of

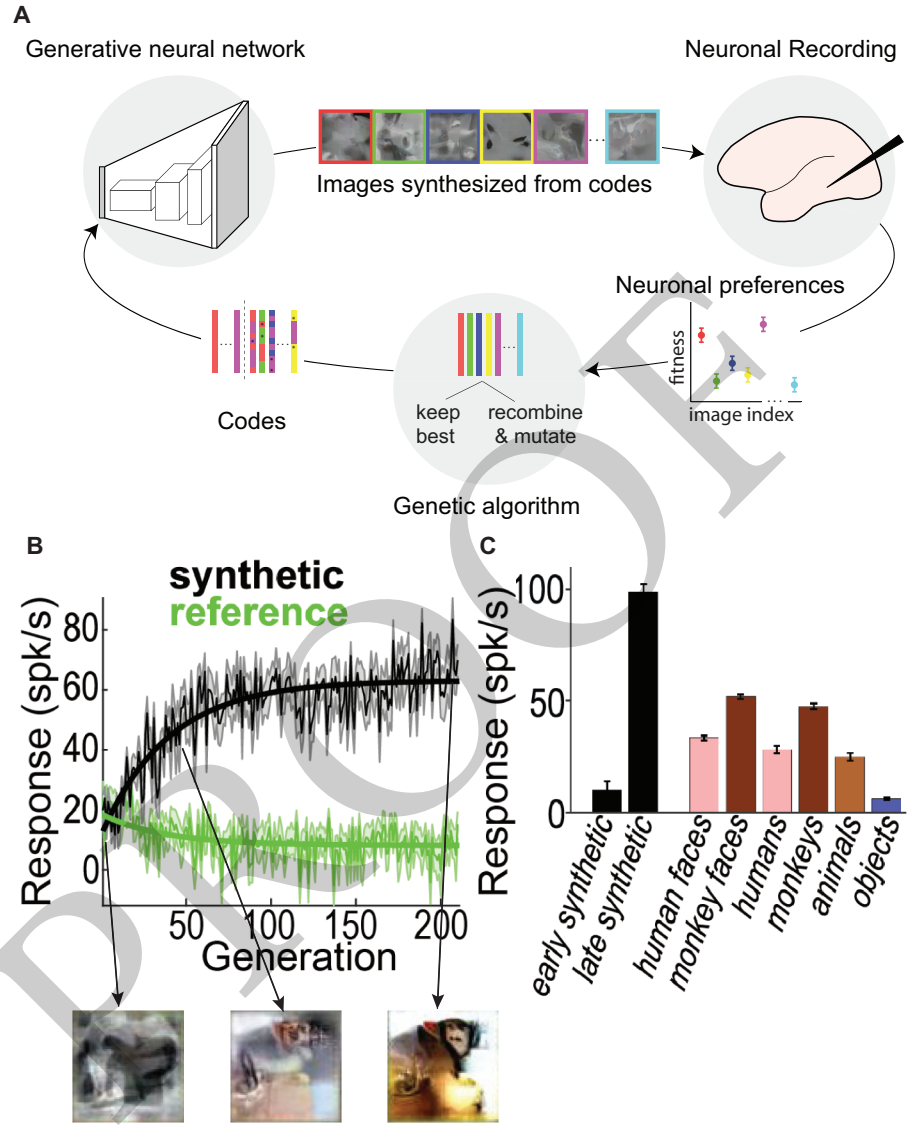

Figure 9.10 Image generators can help probe neuronal tuning in an unbiased manner. **(A)** A promising recent application of image generators is the development of closed-loop algorithms to investigate neuronal tuning. Schematic of the XDream algorithm consisting of an image generator, neuronal recordings, and a genetic algorithm. (**B**) The firing rate of an inferior temporal cortex neuron increases with each iteration of the XDream algorithm (synthetic images, black), creating images that are better than reference natural images. (**C**) While the average responses of this neuron to natural images may lead some investigators to infer tuning for faces, the synthetic images trigger even higher firing rates.

a particular unit in the network, the average activity of all units in a given layer, or the standard deviation of the activity of units within a layer. In neuroscience, the fitness function could be the firing rate of a given neuron in response to the images (as shown in multiple examples in Chapters 5 and 6). After ranking the images based on the fitness function, XDream uses a genetic search algorithm to select, delete, and recombine the initial set of features to create a new round of images. Importantly, XDream does not make any a priori assumptions about neuronal tuning, nor does it require any knowledge about the architecture or weights in the neural network or brain; the algorithm only requires a way to evaluate fitness values for each image.

XDream can visualize the features preferred by units in neural networks. It can discover images that trigger high activation – extrapolating across different layers, different architectures, and even different training regimes. Remarkably, XDream is also very effective in discovering images that trigger high activation in real biological neurons (Section 6.4). Without any assumption about cortical connectivity or preconceptions about neuronal preferences, and within the constraints introduced by biological recordings, the algorithm generates images that trigger high firing rates (Figure 9.10B). These synthetic images turn out to be as effective as – or, in several cases, more effective than – the types of random natural images that have been used in neuroscience for decades (Figure 9.10C).

9.10 Reflections on Cross-Validation and Extrapolation

In this chapter, we have highlighted some of the remarkable achievements of computer vision algorithms. We shift gears now to emphasize some of the critical challenges for current algorithms and some of the exciting opportunities ahead. Let us start with the critical question of generalization. In Section 8.8, we introduced the concept of cross-validation. To reduce the risk of overfitting and deluding ourselves into thinking our algorithms are better than they actually are, it is critical to separate the data into a training set and an independent test set.

What is not well defined in most computer vision applications is how different the test set should be from the training set. In most typical scenarios, we have a large dataset, and we randomly select some images for training and the rest for testing. How excited we should be about the results depends critically on how distinct the test set really is. In a trivial example, we alluded earlier to the potential problem of duplicate images in datasets (Section 8.8). Suppose that image 5,000 and image 8,000 are actually identical, and suppose that the random selection assigns image 5,000 to the training set and image 8,000 to the test set. Of course, this is not real cross-validation, and correctly classifying image 8,000 should not be considered to be an achievement of the algorithm. In a barely more complex example, suppose now that image 8,000 is identical to image 5,000 except for one pixel, or that image 8,000 is a slightly cropped version of image 5,000. Although we can follow all the rules of cross-validation and adequately separate images into an independent test set, adequately assessing performance is problematic if the test images are very similar to those in the training set.

There are more subtle and pernicious versions of this problem. Many databases are based on pictures from the web. There may be strong biases and spurious correlations in the types of pictures that people upload on the web. For example, imagine that we want to build an algorithm to recognize the Tower of Pisa in Italy. Tourists who visit the Leaning Tower of Pisa tend to take pictures of the famous tower and upload those pictures on the web. There are only so many positions from which one can take a picture of the Tower of Pisa, and there are many, many tourists (about 10^6 tourists every year). There may be many biases in the locations from which people take those pictures. For example, people may tend to approach the tower from certain streets, there may be specific locations where people tend to sit, and few people use drones to take aerial pictures. There may be biases also in terms of what exactly the pictures show (for example, most people photograph the entire tower as opposed to parts of it; most pictures may contain much of the surrounding grass area around the tower). There may even be general biases in the color of the sky surrounding the tower (for example, there may be many more pictures on a sunny day and very few pictures during a thunderstorm). Collecting all the Leaning Tower of Pisa pictures and performing adequate cross-validation to ensure that the test images are not too similar to those in the training set is difficult. Unless cross-validation is done extremely carefully, an algorithm might achieve high accuracy in recognizing the Tower of Pisa yet fail miserably with an unusual picture taken from a drone on a rainy day. In other words, it is easy for the algorithm to overfit to the training data, despite our best intentions and best efforts to separate the training and test datasets.

This problem is not restricted to famous landmarks. For example, many people are fond of showing off the food that they prepared by uploading pictures to social media. Consider all the pictures of omelets on the web. Are they mostly taken from the same angle? Are the omelets typically on a plate? Is the plate white in many pictures? Are most of the pictures taken with more or less uniform kitchen illumination? Do some of them also contain forks and knives? How many pictures of an omelet hanging from a tree branch in the park on a rainy day are there on the web?

Yet another example of this family of problems can be gleaned from the action recognition task illustrated in Figure 9.4. The frames in Figure 9.4A are taken from a well-known video database for action recognition, UCF101. Without any sophisticated processing, using only single frames and pixel-level information, one can infer that if the image contains many blue pixels, it is likely to correspond to "breaststroke," whereas if the image contains many green pixels, it is likely to correspond to "soccer juggling." Other actions also contain a lot of blue or green, but it is nonetheless possible to get well above chance performance in this task without any acute understanding of the images, let alone any comprehension of what the action labels mean. In contrast, the controlled datasets shown in Figure 9.4B are significantly harder: here, the task is to determine whether the person is drinking or not. There are lots of different ways of drinking (from a cup, from a bottle, using a straw, using hands as a vessel, from a drinking fountain). A true action classifier capable of discriminating pictures showing drinking should be able to generalize to all of these

conditions. We cannot get significantly above chance performance in the task in Figure 9.4B by merely considering the number of blue pixels. Above chance performance in pixel-level classification is a good indication that the task is too easy, that there are strong similarities between training and test images, and that there could be a significant degree of overfitting.

Because of these types of correlations in the images within a dataset, contextual information tends to play a prominent role in computer vision algorithms. Algorithms can adequately infer the right label even if the object itself is completely occluded, purely based on the statistics of contextual information. For example, pictures of traffic lights tend to be in a street environment, and the traffic light tends to be positioned in the upper part of the picture. While this may be seen as favorable capitalization on image statistics, the converse is also true: neural networks can misclassify an object placed out of context. Contextual information can help humans too (Section 3.7); however, humans tend to be more immune to image manipulations like placing objects out of context.

Not all real-world applications depend on generalization. For example, if Facebook wishes to automatically tag the Tower of Pisa in pictures uploaded by its users, Facebook may be satisfied with achieving 99 percent accuracy and miss those few instances of an aerial picture during a thunderstorm. Other applications may critically require preparing for the unexpected. We want self-driving cars to be able to detect a cow crossing the highway, even if this is a rare circumstance.

The problem of cross-validation is related to the question of bias in training datasets (referred to as dataset bias in the computer vision community). For example, suppose that we build an algorithm to detect breast tumors using mammograms from white women between 50 and 60 years old who live in California. Will the algorithm work with similarly aged white women from Massachusetts? And from Europe? Would the algorithm work with African American or Asian women? Would the algorithm work with women in their thirties or their eighties? The issue of biases in training data has recently been highlighted in the news for the task of face identification systems that performed better for certain ethnic groups than for others.

Of note, the problem of biases is not unique to computer vision. Visual recognition biases are prevalent in human vision too. Radiologists trained to recognize breast tumors in mammograms from white women in their fifties may also fail when tested with mammograms from other groups of women. In the case of face identification, there are well-known human biases based on where people grow up and the amount of exposure they have had to faces from different ethnic groups.

Generalization is an essential and desirable property for computational algorithms. The ability to generalize from cross-validated data is not well defined and depends on how distinct the test set is. One way to attempt to quantify this problem is to distinguish between interpolation (within-distribution generalization) and extrapolation (out-of-distribution generalization). Again, precisely what is meant by distribution is not well defined, but at least this provides a way to begin to quantify the ability of algorithms to extrapolate beyond their training set.

9.11 Adversarial Images

We have highlighted some of the exciting advances in how computational algorithms process images and how machine vision can match or even surpass human performance in many applications. However, caution should be exercised before thinking that machines might be about to pass the general Turing test for vision. There are still many visual tasks that machines cannot solve. Furthermore, it is relatively easy to fool machines in visual tasks (e.g., Figure 9.4).

One example of perplexing behavior by deep convolutional neural networks is the case of *adversarial images*, whereby minimal changes to an image drastically change the predicted class (Figure 9.11). Adversarial images appear similar, almost identical, to humans, yet they receive different labels by a computer vision system. For example, the two images in Figure 9.11 are virtually indistinguishable to human observers, yet a deep convolutional network correctly classified the one on the left as "corn," and incorrectly labeled the one on the right as "snorkel." Given an algorithm that is forced to assign a binary label to an image, A versus B, it is inevitable that there will be a boundary where we can move from A to B with small image changes. The separation between two labels in image space is akin to standing in the often-arbitrary border between two states or trying to define precisely where the rain starts when it is raining in location A and not B.

These adversarial images are typically created by using knowledge about the categorical boundaries and astutely changing a few pixels to push the image into the opposite side of the label. As in the DeepDream algorithm introduced in Section 9.10, the process of creating adversarial images involves gradient descent on the pixels of the image itself.

What is intriguing about the adversarial examples is the profound difference between machine and human perception. In many real-world applications, seeing the world the way humans do may be quite relevant. In fact, there has been a whole industry of

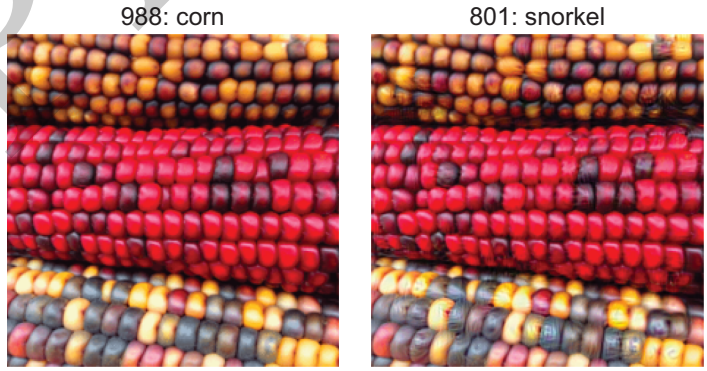

Figure 9.11 Adversarial examples are misclassified by computational algorithms, yet they seem indistinguishable to the human brain. The two images appear to be indistinguishable to humans. However, state-of-the-art computer algorithms classify the one on the left as "corn" and the one on the right as "snorkel." The image on the right was created by introducing small amounts of noise to the image on the left, along specific directions.

investigators designing "adversarial attacks" to confuse computer vision systems, together with a similarly vigorous community of defenses against such adversarial attacks. For example, one may ask whether the image on the right in Figure 9.11 would revert back to corn upon scaling it, changing its color, using different versions of the same network (e.g., starting from different random initial conditions), or using different architectures. These examples clearly illustrate that, even when current algorithms can correctly label many images, state-of-the-art deep convolutional neural networks do not necessarily see the world the way humans do.

Adversarial examples are not unique to the field of computer vision. Humans also suffer from such adversarial examples; it is just much harder to generate such examples for humans because we cannot compute gradients on biological networks as we do with artificial neural networks. Even without such gradients, psychologists have discovered many images that confuse humans. Humans are fallible in many visual illusions that deceive us into seeing things that do not exist (Chapter 3).

In sum, humans and state-of-the-art computer vision systems make similar mistakes in object classification tasks (Section 8.12). However, many images can trick computer vision systems and not humans, and vice versa. These results show that even our best computer vision systems still do not fully account for human visual recognition capabilities. Because it is possible to find such double dissociations between machine and human vision, these results also show that current deep convolutional neural networks still cannot pass the Turing test. We can easily tell a machine from a human by showing the image on the right in Figure 9.11.

9.12 Deceptively Simple Tasks That Challenge Computer Vision Algorithms

Adversarial examples are especially constructed to fool computational algorithms. It is also possible to challenge computational algorithms in basic visual tasks that are not designed with the specific purpose of moving images across categorical boundaries. While there are many visual questions where computers outperform humans, such as bar code reading, there are also many common visual questions where it is easy to trick computers (Figure 9.11).

Many visual questions that are simple for humans represent a formidable challenge for current architectures. Consider the examples in Figure 9.12, taken from a set of 23 visual reasoning tasks introduced by Don Geman's group. Given a set of positive (top row) and negative (bottom row) examples, we need to figure out what the rule is to be able to classify novel images. Humans quickly realize that the rule is "same or different" except for translation for the two shapes in Figure 9.12A, "inside or outside" in Figure 9.12B, and whether the largest of the three shapes is in between the other two or not in Figure 9.12C. Even if humans have never seen these particular examples and tasks before, they can quickly infer what the rules are. Humans can then use those rules to reason about new examples. Thomas Serre's group has shown that current computer vision models struggle with these tasks despite extensive training with up to a million examples.

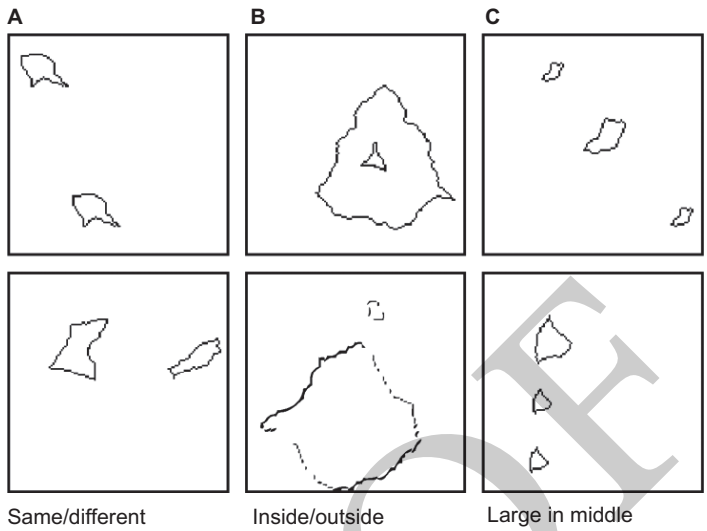

Figure 9.12 Some apparently simple tasks pose a challenge to current algorithms. The task involves learning to classify images into two groups according to certain fixed but unknown rules. Here are shown three types of rules: (**A**) same or different, (**B**) inside/outside, (**C**) large object in the middle. Positive examples are shown on the top row and negative examples on the bottom row. Reproduced from Fleuret et al. 2011

A related example is the CLEVR dataset consisting of images containing multiple geometrical shapes like spheres, cubes, and cylinders of varying sizes, colors, and material properties. The task involves answering questions such as whether the red cylinder to the left of the blue cube is larger than the red cylinder to the right of the blue cube or whether the number of large objects is the same as the number of metallic objects. Current networks appear to adequately learn to answer these questions when trained and tested on the same combinations of shapes and color properties. However, when tested on novel combinations of shapes and colors (e.g., when the network has never encountered a blue cylinder during training even though it has seen lots of blue cubes and lots of red cylinders), the networks failed to generalize.

9.13 Challenges Ahead

There has been significant progress in teaching computers how to see. We are already surrounded by machines that can successfully use automatic vision algorithms in real-world applications. The exhilarating progress in computer vision may lead us to think that we have almost solved the problem of vision. Indeed, prominent newspapers proposed headlines with statements hinting that vision has almost been solved. However, I would argue that we are still extremely far from passing the general Turing test for vision and that the best is yet to come.

In addition to some of the challenges discussed in the previous sections (adversarial images, generalization, visual reasoning in simple tasks), an area that is advancing rapidly and highlights progress and challenges is image captioning (also related to question-answering systems on images). Given an image, the goal is to provide a brief and "relevant" description. In contrast to categorization tasks, it is more challenging to quantitatively evaluate the results. Furthermore, these tasks may confound vision and language, as articulated at the beginning of this chapter. However, captioning algorithms provide a good summary to close this chapter while highlighting the exciting challenges ahead of us in the field.

An example of the state-of-the-art in image captioning is shown in Figure 9.13, which is based on results obtained using a caption bot (www.captionbot.ai, circa November 2018). It is important to emphasize the date because I suspect that we will see a major improvement in the years to come. The captions provided by this algorithm are quite

Figure 9.13 Successes and challenges in image captioning. Four example results from the www .captionbot.ai image captioning system

impressive. The system is good at detecting people, even quantifying whether the image contains one person (Figure 9.13A) or multiple people (Figure 9.13D). The system can also detect the gender in Figure 9.13A, and it makes a reasonable guess about whether people are happy in Figure 9.13D (I am in that picture, and I can attest that I was very happy; I suspect that most people visiting the Tower of Pisa are). The system also correctly infers that the person is sitting in Figure 9.13A and standing in Figure 9.13D. Furthermore, the system also detects other aspects of the scene, including the presence of a table in Figure 9.13A, water in Figure 9.13B, and a building in Figure 9.13D. Many other objects are not described, which is perhaps reasonable, given that the goal is to caption and not to mention every single object. Another caveat of using image captioning as a test-bed for vision is that we do not know whether particular objects are not mentioned because they were not detected or because the algorithm deemed those objects not to be too relevant.

It is a bit surprising that the system does not describe the Tower of Pisa in Figure 9.13D, given that such monuments have an exorbitant amount of training data. Perhaps even more surprisingly, there is a rather salient spoon in Figure 9.13A that was not described. It also seems likely that many humans would describe the bride in Figure 9.13B. The system is not able to deal with line drawings (Figure 9.13C), but it is nice that the algorithm was able to realize its limitations and admit that it cannot describe line drawings. Differentiating line drawings from photographs is perhaps not too difficult, particularly if the image has a considerable number of white pixels, a few black pixels, and essentially no textures. It is relatively easy for humans to recognize that there are three people in the drawing in Figure 9.13C, though it is not clear exactly how this deduction happens. Current algorithms such as the image captioning one illustrated here probably have minimal, if any, training with drawings. In contrast, most humans have had exposure to the underlying symbolism behind line drawings.

One easy way to break these captioning systems is to scramble the image. For example, we can divide the image into four quadrants and rearrange the quadrants randomly. The image mostly loses its meaning, yet the caption remains largely unchanged. If we present the fundus photograph from Figure 9.7 (only the fundus photograph, without the rest of the figure), the system responds with "I can't really describe the picture but I do see light, sitting, lamp." It is commendable that the system realizes that it cannot quite describe the image – that the system realizes that the image is different from its training set. There is indeed a light in the image. The system probably saw many examples where the word "light" correlated with the word "lamp," throwing it into the description.

It is a bit harder to deduce where the word "sitting" comes from in this example. The challenge in explaining where the labels come from is a characteristic of deep neural networks that many people have criticized. Given the large number of parameters in the system, it is not always easy to put into words why the system produces a given output. Humans can come up with post hoc explanations, but it is not always easy to evaluate those explanations. Radiologists do not tend to explain much about how they make their diagnoses, and they certainly are not required to come up with an explanation at the

level of what neurons in their brains do. Humans would struggle to provide a mechanistic explanation of why they think that they see a tree in Figure 8.1.

Of note, the same type of architectures used in image captioning can be trained to outperform doctors in interpreting the same fundus photographs. The same architectures can be trained to detect the Tower of Pisa. Each one of these questions requires separate training steps. In contrast, a doctor can evaluate fundus photographs *and* also understand what is happening in Figure 9.13, whereas many current deep convolutional networks are ultra-specialized for specific tasks, and it is not easy to train the neural networks to perform multiple tasks.

Passing the Turing test requires being able to answer *any* question about an image, not just being trained to answer a single type of question. It is clear that one can ask many questions about the images in Figure 9.13. As impressive as those captions are, they do not come even close to solving the Turing test for vision. The captions completely fail to grasp fundamental aspects of the scene, what is happening, and who is doing what to whom and why. Humans can look at these images and understand the relationships between the different objects, their relative positions, and why they are where they are and even make inferences about what happened before or what may happen next.

Even more intriguingly, all these images are meant to be somewhat curious or funny. To end on a light note, I would like to highlight an example problem that I consider to be extremely challenging: understanding the human sense of humor based on images. Of course, even though the concept of funny is subjective and depends on age, gender, and cultural background, there are still strong correlations between different humans in what is funny or not.

Let us consider Figure 9.13C as an example. What is funny about this image? To grasp what is happening in the image, we need to incorporate not merely pixel-level information, not just labels of specific objects, but also their symbolism and relative interactions. The scale, together with the few traces that represent the attire of the person in the center, plus his relative position with respect to the other people, leads us to think that he is a judge. Note that it is the combination of many of these labels and their interactions that lead us to this understanding. Each one piece of information on its own would not necessarily be sufficient. The person sitting below the judge is probably the accused (or, less likely, a witness). This inference is partly based on the person's shirt with horizontal stripes but mostly based on his relative position and an understanding of the arrangement of the judge and the accused in a court of law. We can infer that the third person is a policeman, which is consistent with his outfit but also with the fact that he is standing and that he is behind the accused.

After deciphering that the person in the center is a judge, we realize that he is holding a gavel, he is shouting, and he is hitting the table with his gavel. The accused is also angry, making eye contact with the judge. Curiously, the accused also seems to be holding a gavel. This observation strikes us as unusual: the accused is not supposed to hold a gavel, let alone use it. The deviation from the norm is the essence of why the image is funny: it portrays an unexpected scenario. If we take out the few pixels that represent the accused's gavel, the image immediately becomes less appealing. Of course, humor is subjective and may vary from human to human.

Even if people do not find Figure 9.13C to be funny, they may still understand all the symbolism, the actions, who the people are, and how they relate to each other. Regardless of whether a particular image is funny or not, humans can interpret what is happening in Figure 9.13C the first time they see this image. Humans do not need extensive training with black-and-white drawings of people in a court of law to understand this image. There is a substantial amount of world knowledge that we need to have to be able to understand and interpret Figure 9.13C. Predicting whether an image is funny or not is further complicated by the fact that, even if we trained an algorithm to understand all the symbolism in Figure 9.13C, that would be of no help whatsoever to understand why Figure 9.13A is intriguing, nor to deduce what probably happened in Figure 9.13B.

There are trivial, brute-force, and ultimately uninteresting solutions that could yield above-chance performance in a funny versus not-funny discrimination task. Throwing lots of images like the ones in Figure 9.13 into a deep convolutional network trained via supervised learning could lead to some ability to decipher funny or not more than 50 percent of the time. For example, a lot of funny images are cartoons or drawings. A system could quickly learn to differentiate drawings from real photographs. If drawings are correlated with more "funny" labels, then the system might appear to perform quite well. However, in reality, the model would know absolutely nothing about humor. Removing the gavel from the accused in Figure 9.13C would not change the label for this type of model, even though this simple manipulation radically changes how funny the image is. This image manipulation is but another example of the problems with overfitting and biases elaborated upon in Section 9.11. A well-controlled visual task should ensure that the labels are not correlated with any other properties beyond the ones under study.

Determining whether an image is funny or not illustrates current challenges to incorporate additional knowledge into visual processing. However, it is worth pointing out that there is no physical limit to what computers can do. If we 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 exclusively the domain of humans. Any desktop computer can play chess competitively, and the best computers can beat the world's chess champions. IBM's Watson has thrived in the trivia-like game of Jeopardy. Even more, 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 the level of human performance, separating images from the web into 1,000 different categories, detecting faces to take pictures, recognizing faces to log in to a smartphone, or analyzing clinical images, galaxies, and much more. While humans still outperform the most sophisticated current algorithms in the majority of visual tasks, the gap between machines and human vision tasks is closing rapidly.

Significant progress has been made toward describing visual object recognition in a principled and theoretically sound fashion. However, the lacunas in our understanding of the functional and computational architecture of the 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 and bottom-up circuits that perform rapid computations through gradual increases in selectivity and tolerance to object transformations. In stark contrast with the pathway from the retina to the primary visual cortex, we do not have a quantitative description of the feature preferences of neurons along the ventral visual pathway. Furthermore, several computational models do not make clear, concrete, and testable predictions toward systematically characterizing the ventral visual cortex at the physiological level. Computational models can perform several complex recognition tasks. However, for the vast majority of recognition tasks, machine vision still falls 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 visual cortex circuitry at an 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 competitively, the next generation may not be surprised by intelligent devices that can see the world as we do.

9.14 Summary

- A machine would pass the Turing test for vision if we cannot distinguish its answers from human answers in response to any arbitrary question about any image.
- Computer vision has shown remarkable success in a variety of tasks – including object classification, object detection, segmenting objects in an image, and action classification.
- Success in visual tasks has given rise to a plethora of real-world applications – including face recognition, visual interpretation of a scene for self-driving cars, analyses of clinical images, classification of galaxies from astronomy images, and many more.
- Inverting convolutional networks opened the doors to algorithms that generate synthetic images. One of the applications of image generators is to systematically study the tuning properties of neurons along ventral visual cortex.
- Despite rapid progress, computer vision applications remain fragile. Algorithms can be fooled relatively easily, and there are many tasks that are simple for humans yet very challenging for machines, such as determining whether a shape is inside or outside of another one.
- Due to the large number of parameters, it is often unclear how well current computer vision algorithms can extrapolate to novel scenarios as opposed to merely interpolating between training samples. Generalization is an essential requirement for future computational algorithms in vision.
- Many exciting challenges remain to teach computers to see and interpret the world the way humans do. As an example of a formidable challenge, training computer vision systems to determine whether an image is funny or not seems to be well beyond the capabilities of current systems.

Further Reading

See more references at http://bit.ly/2t53QRd

- Lotter, W.; Kreiman, G.; and Cox, D. (2020). A neural network trained for prediction mimics diverse features of biological neurons and perception. *Nature Machine Learning*. 2:210–219.
- Poplin, R.; Varadarajan, A.; Blumer, K.; et al. (2018). Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning. *Nature Biomedical Engineering* 2:158–164.
- Russakovsky, O.; Deng, J.; Su, H., et al. (2014). ImageNet Large Scale Visual Recognition Challenge. In: CVPR: 1409.0575.
- Szegedy, C.; Zaremba, W.; Sutskever, I.; et al. (2014). Intriguing properties of neural networks. In: International Conference on Learning Representations.
- Turing, A. (1950). Computing machinery and intelligence. *Mind* LIX:433–460.