Why is detection and localization important? One very useful localization task is detecting pedestrians in images taken from a self-driving car. Needless to say, it's extremely important that a self-driving car be able to identify all nearby pedestrians. Other applications of object detection could be used to find all instances of friends in photos uploaded to a social network. Yet another application could be to identify potential collision dangers from a drone.

This wealth of applications has made detection and localization the focus of tremendous amounts of research activity. The ILSVRC challenge mentioned multiple times in this book focused on detecting and localizing objects found in the ImagetNet collection.

Image Segmentation

Image segmentation is the task of labeling each pixel in an image with the object it belongs to. Segmentation is related to object localization, but is significantly harder since it requires precisely understanding the boundaries between objects in images. Until recently, image segmentation was often done with graphical models, an alternate form of machine learning (as opposed to deep networks), but recently convolutional segmentations have risen to prominence and allowed image segmentation algorithms to achieve new accuracy and speed records. Figure 6-10 displays an example of image segmentation applied to data for self-driving car imagery.

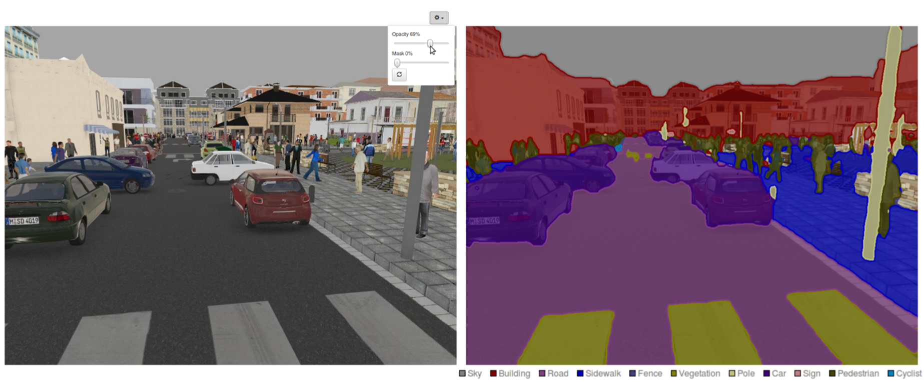

Figure 6-10. Objects in an image are "segmented" into various categories. Image segmentation is expected to prove very useful for applications such as self-driving cars and robotics since it will enable fine-grained scene understanding.

Graph Convolutions

The convolutional algorithms we've shown you thus far expect rectangular tensors as their inputs. Such inputs could come in the form of images, videos, or even sentences. Is it possible to generalize convolutions to apply to irregular inputs?

The fundamental idea underlying convolutional layers is the notion of a local receptive field. Each neuron computes upon the inputs in its local receptive field, which typically constitute adjacent pixels in an image input. For irregular inputs, such as the undirected graph in Figure 6-11, this simple notion of a local receptive field doesn't make sense; there are no adjacent pixels. If we can define a more general local receptive field for an undirected graph, it stands to reason that we should be able to define convolutional layers that accept undirected graphs.

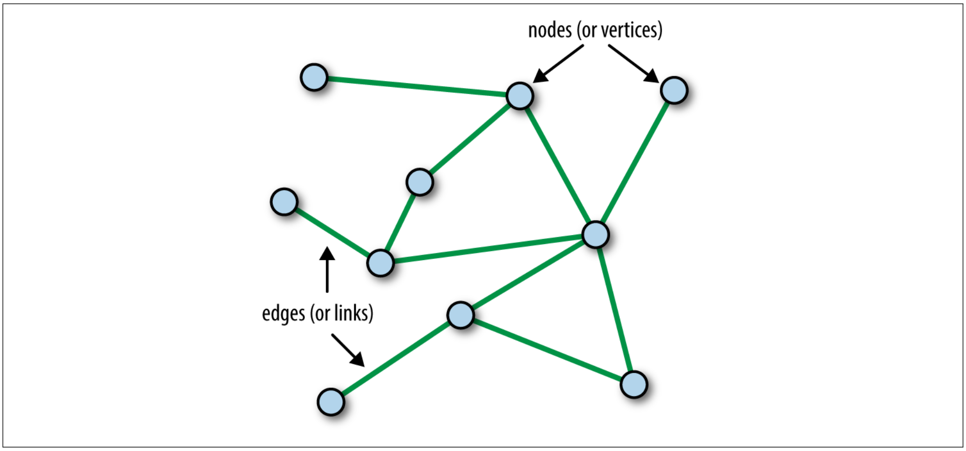

Figure 6-11. An illustration of an undirected graph consisting of nodes connected by edges.

As Figure 6-11 shows, a graph is made up of a collection of nodes connected by edges. One potential definition of a local receptive field might be to define it to constitute a node and its collection of neighbors (where two nodes are considered neighbors if they are connected by an edge). Using this definition of local receptive fields, it's possible to define generalized notions of convolutional and pooling layers. These layers can be assembled into graph convolutional architectures.

Where might such graph convolutional architectures prove useful? In chemistry, it turns out molecules can be modeled as undirected graphs where atoms form nodes and chemical bonds form edges. As a result, graph convolutional architectures are particularly useful in chemical machine learning. For example, Figure 6-12 demonstrates how graph convolutional architectures can be applied to process molecular inputs.