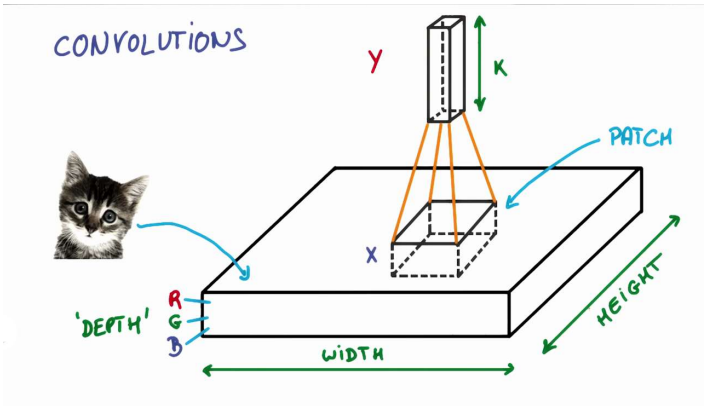
Breaking up an Image

The first step for a CNN is to break up the image into smaller pieces. We do this by selecting a width and height that defines a filter.

The filter looks at small pieces, or patches, of the image. These patches are the same size as the filter.

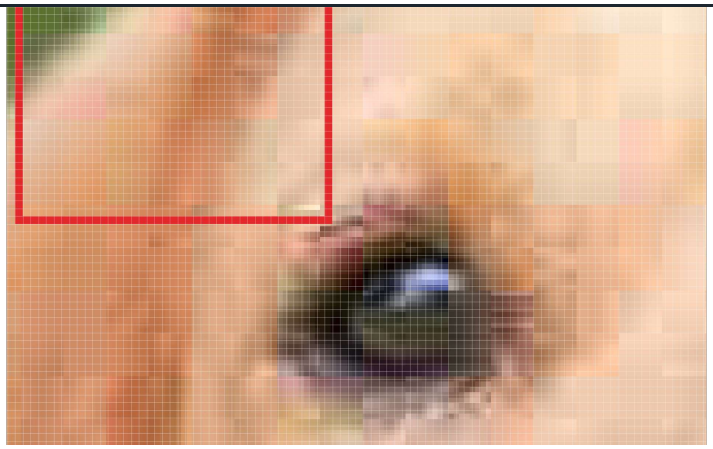


As shown in the previous video, a CNN uses filters to split an image into smaller patches. The size of these patches matches the filter size.

We then simply slide this filter horizontally or vertically to focus on a different piece of the image.

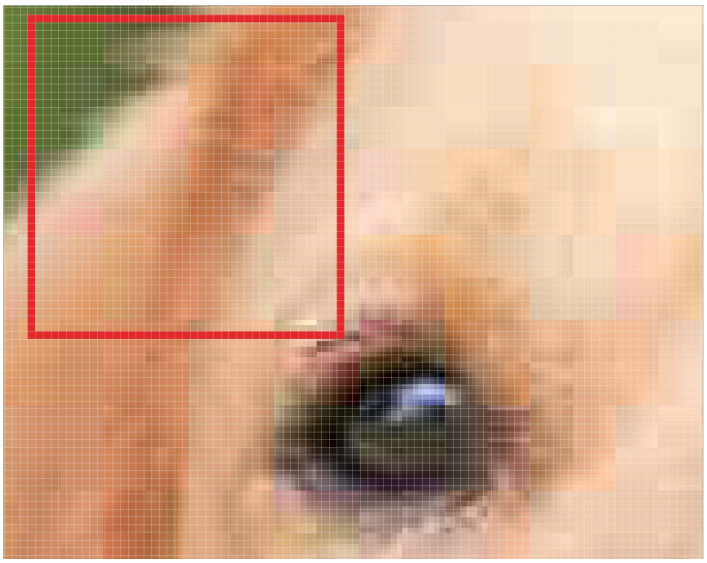
The amount by which the filter slides is referred to as the 'stride'. The stride is a hyperparameter which you, the engineer, can tune. Increasing the stride reduces the size of your model by reducing the number of total patches each layer observes. However, this usually comes with a reduction in accuracy.

Let's look at an example. In this zoomed in image of the dog, we first start with the patch outlined in red. The width and height of our filter define the size of this square.



One patch of the Golden Retriever image.

We then move the square over to the right by a given stride (2 in this case) to get another patch.



We move our square to the right by two pixels to create another patch.

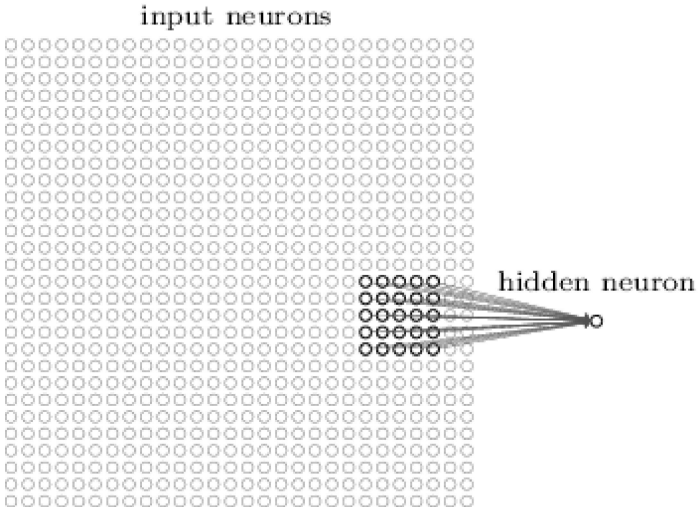
What's important here is that we are **grouping together adjacent pixels** and treating them as a collective.

In a normal, non-convolutional neural network, we would have ignored this adjacency. In a normal network, we would have connected every pixel in the input image to a neuron in the next layer. In doing so, we would not have taken advantage of the fact that

By taking advantage of this local structure, our CNN learns to classify local patterns, like shapes and objects, in an image.

Filter Depth

It's common to have more than one filter. Different filters pick up different qualities of a patch. For example, one filter might look for a particular color, while another might look for a kind of object of a specific shape. The amount of filters in a convolutional layer is called the *filter depth*.

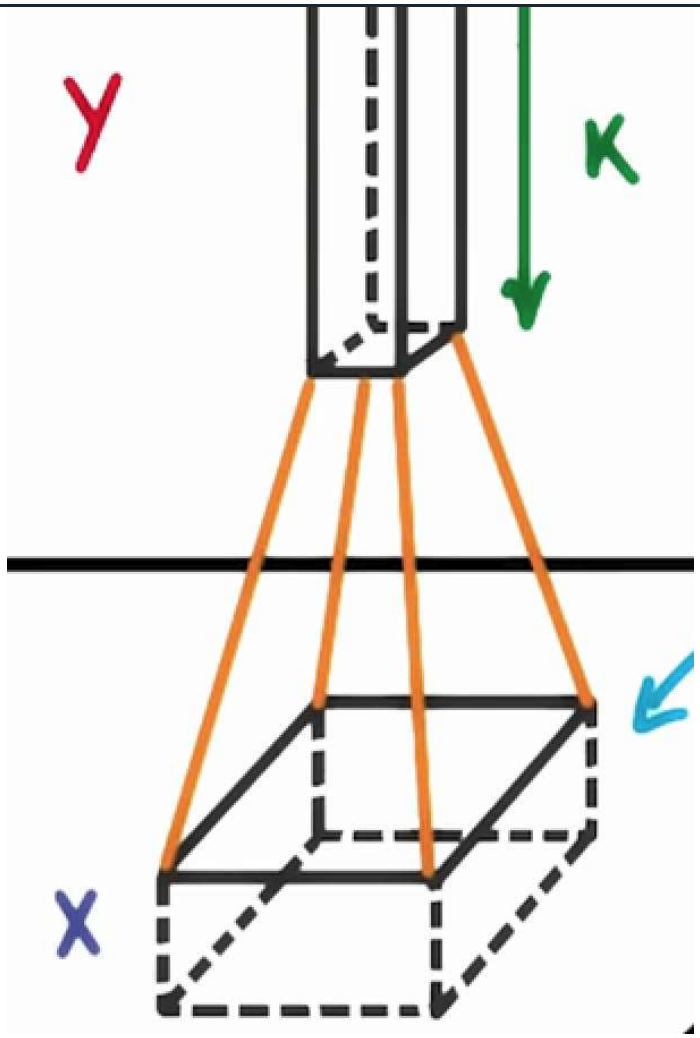


In the above example, a patch is connected to a neuron in the next layer. Source: MIchael Nielsen.

How many neurons does each patch connect to?

That's dependent on our filter depth. If we have a depth of k, we connect each patch of pixels to k neurons in the next layer. This gives us the height of k in the next layer, as shown below. In practice, k is a hyperparameter we tune, and most CNNs tend to pick the same starting values.

Filters



Choosing a filter depth of k connects each patch to k neurons in the next layer.

But why connect a single patch to multiple neurons in the next layer? Isn't one neuron good enough?

Multiple neurons can be useful because a patch can have multiple interesting characteristics that we want to capture.

For example, one patch might include some white teeth, some blonde whiskers, and part of a red tongue. In that case, we might want a filter depth of at least three - one for each of teeth, whiskers, and tongue.

Lesson 4:
Convolutional Networks

12. Quiz: Number of Parameters

13. Solution: Number of Parameters

14. Quiz: Parameter Sharing

Live Help

Get immediate help

Filters

This patch of the dog has many interesting features we may want to capture. These include the presence of teeth, the presence of whiskers, and the pink color of the tongue.

Having multiple neurons for a given patch ensures that our CNN can learn to capture whatever characteristics the CNN learns are important.

Remember that the CNN isn't "programmed" to look for certain characteristics. Rather, it learns **on its own** which characteristics to notice.

NEXT