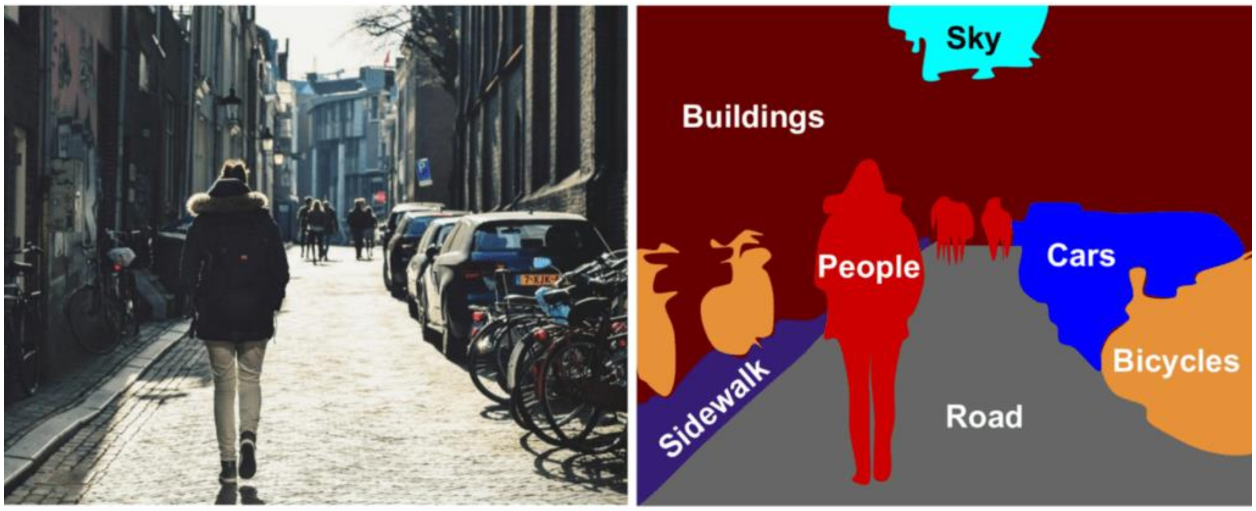
Image Segmentation?

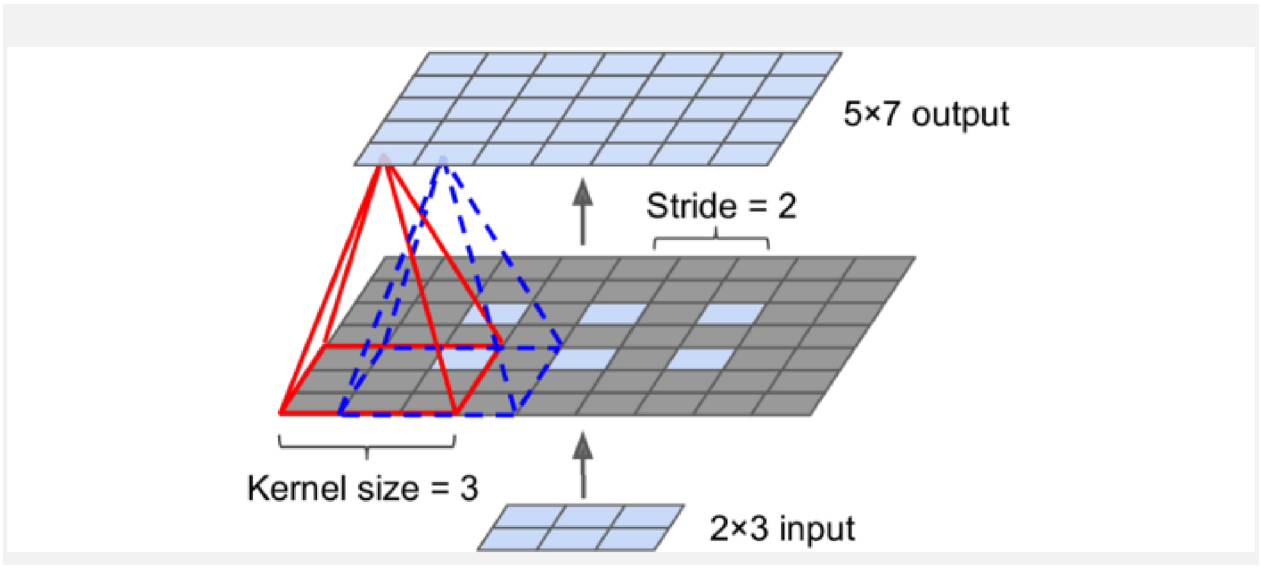


In Image segmentation, each pixel is classified according to the class of the object it belongs to (e.g., road, car, pedestrian, building, etc.), as shown in the figure above. Note that different objects of the same class are not distinguished. For example, all the bicycles on the right side of the segmented image end up as one big lump of pixels. The main difficulty in this task is that when images go through a regular CNN, they gradually lose their spatial resolution (due to the layers with strides higher than); so, a regular CNN may end up knowing that there's a person somewhere in the bottom left of the image, but it will not be much more precise than that.

Just like object detection, there are many different approaches to tackle this problem, some quite complex. However, a reasonably simple solution was proposed in the 2015 paper by Jonathan Long et al. The author starts by taking a pre-trained CNN and turning it into an FCN. The CNN applies an overall stride of 32 to the input image, meaning the last layer outputs feature maps 32 times smaller than the input image. This is too coarse, so they add a single up sampling layer that multiplies the resolution by 32.

By Waqas Ali Munawar

There are several solutions available for up sampling (increasing the size of an image), such as bilinear interpolation, but that only works reasonably well up to * 4 or *8 instead, they use a transposed convolutional layer. It is equivalent to first stretching the image by inserting empty rows and columns (full of zeros) and performing a regular convolution.



Alternatively, some people prefer to think of it as a consistent convolutional layer that can be initialized to perform something close to linear interpolation. Still, since it is a trainable layer, it will learn to do better during training. In **tf.keras**, we can use the **conv2DTranspose** layer.

By Waqas Ali Munawar