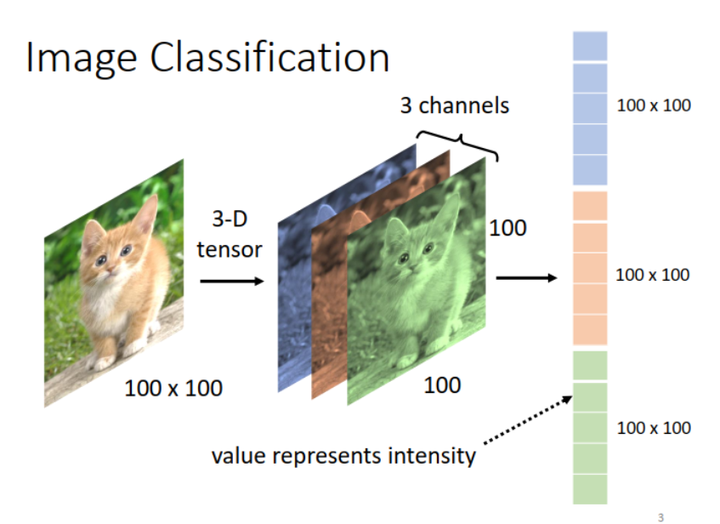
**CNN**

专用于影像分类，在进行分类前需假设图片大小一致。一般图像分为三维，即高，宽和channel数。输入数据需将三维数据合并为一维，图像是怎么表示的呢？在计算机中用RGB三色表示图像。例如，下图中大小为100x100的图片，在RGB这三个channels上各有一个100x100 的矩阵，其中的数值表示强度(intensity)。

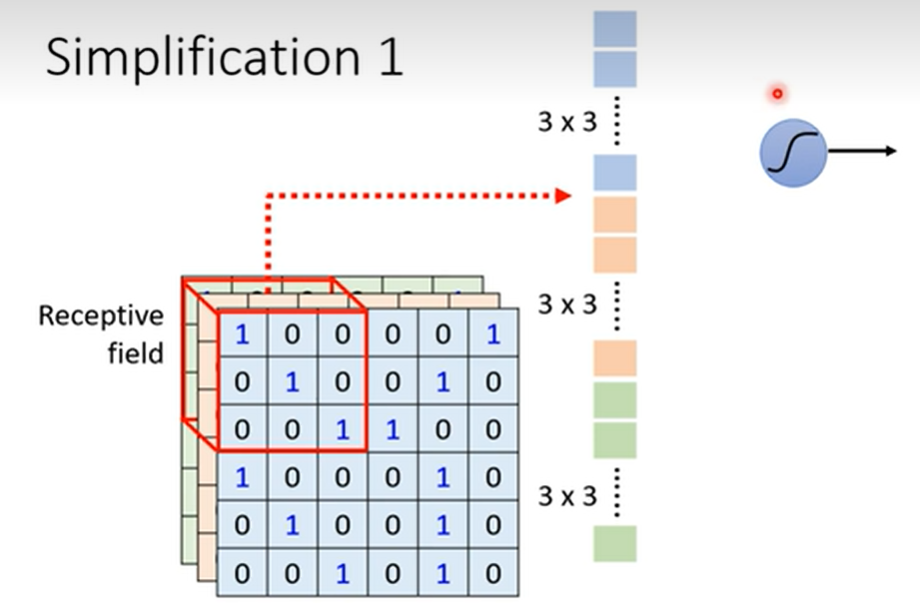


然后生成的向量作为Network的输入。

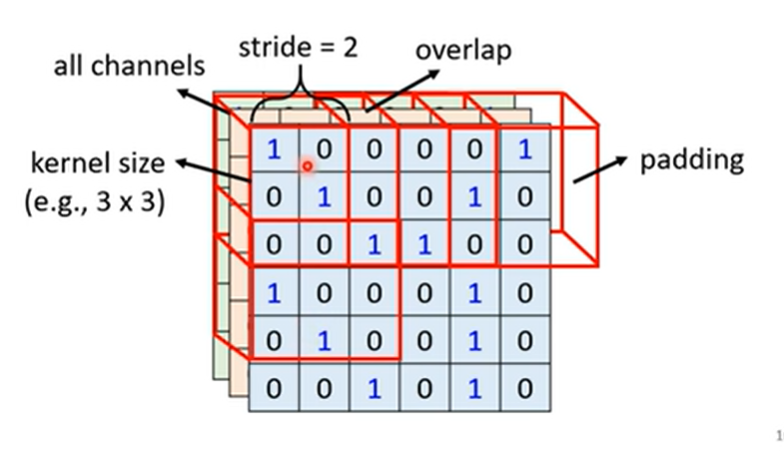
由此我们有一个问题即是有必要把每个像素点都当作一个输入吗？对于大小为100 x 100 的彩色图像，所需的输入神经元达到3\*104之多。如果输出神经元有1000个，那么参数量达到3\*107。模型的复杂度高，而且运算量巨大。

然后根据特征分类的特点我们可以得出不需要将整个图片作为输入，因此我们对输入进行化简。

1.例如，设定一个Receptive field，每个network只需要关心自己Receptive field中发生的事情就好。



以下是一种典型的Receptive field设定方式：

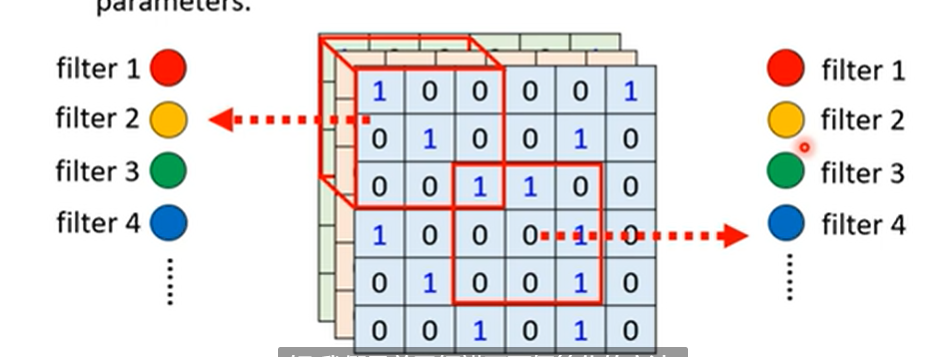


kernel size:指receptive field的宽和高大小。receptive field的depth不需设定，和图像的depth相同。

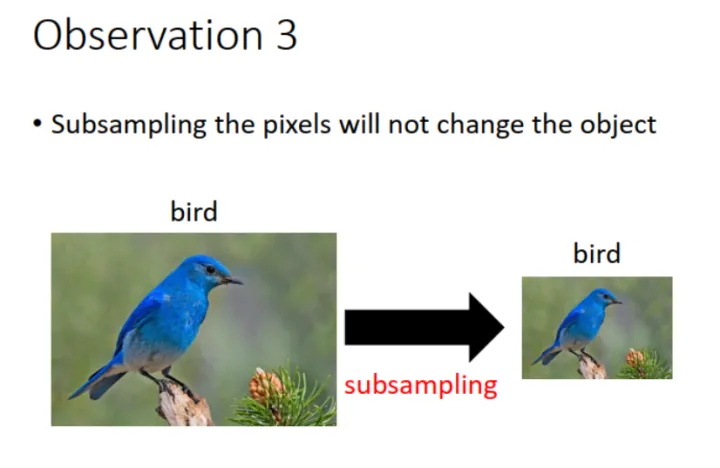
stride:指每次receptive field移动多少像素点。一般stride<kernel size，因为要让相邻receptive fields有重叠(overlap)。

padding:为了充分识别到边缘的pattern，在边缘外补值（比如0）。

参数共用：同样的pattern可能出现在图像的不同区域。比如，下图中的鸟嘴。这样，对于不同的Receptive Fields，可以使用相同的神经元来探测同一个pattern。



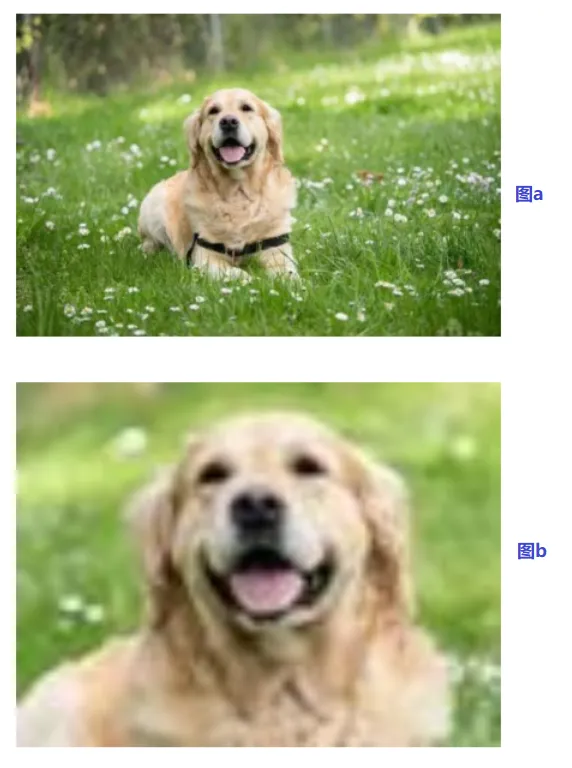
Pooling：我们都有这样的体验，网上下载图片时，有不同的尺寸，比如1920x1440, 1280x960,640x480等，这些图片只是分辨率不同，都可以辨认出里面的图象。如下图所示，对一张鸟儿的图片做subsampling，比如只保留偶数行和偶数列，得到小图片，其中的鸟儿图像虽然更模糊一点，但是不影响辨认。



因此，可以对 Convolutional Layer的输出做subsampling操作，降低运算量。这里叫做Pooling，例如：Max Pooling，取一定大小区域内（比如：2x2）的最大值。

**CNN 的不足之处**

如果对一张图片的部分区域放大，得到的图片无法识别。比如，用类似下图 a 的图片训练模型，识别不了图 b 这种类型。是不是挺奇怪的？这明明就是同一只狗的图片，图 b 只是把图 a 部分放大到同样尺寸而已。



**架构**

卷积神经网络(CNN)的基本架构如下图所示，一般在Convolution之后做Pooling（Pooling 可有可无），这样的操作可以叠几层，然后把输出做flatten变成向量，最后输入到Fully Connected Layer，当作一个分类问题处理。

