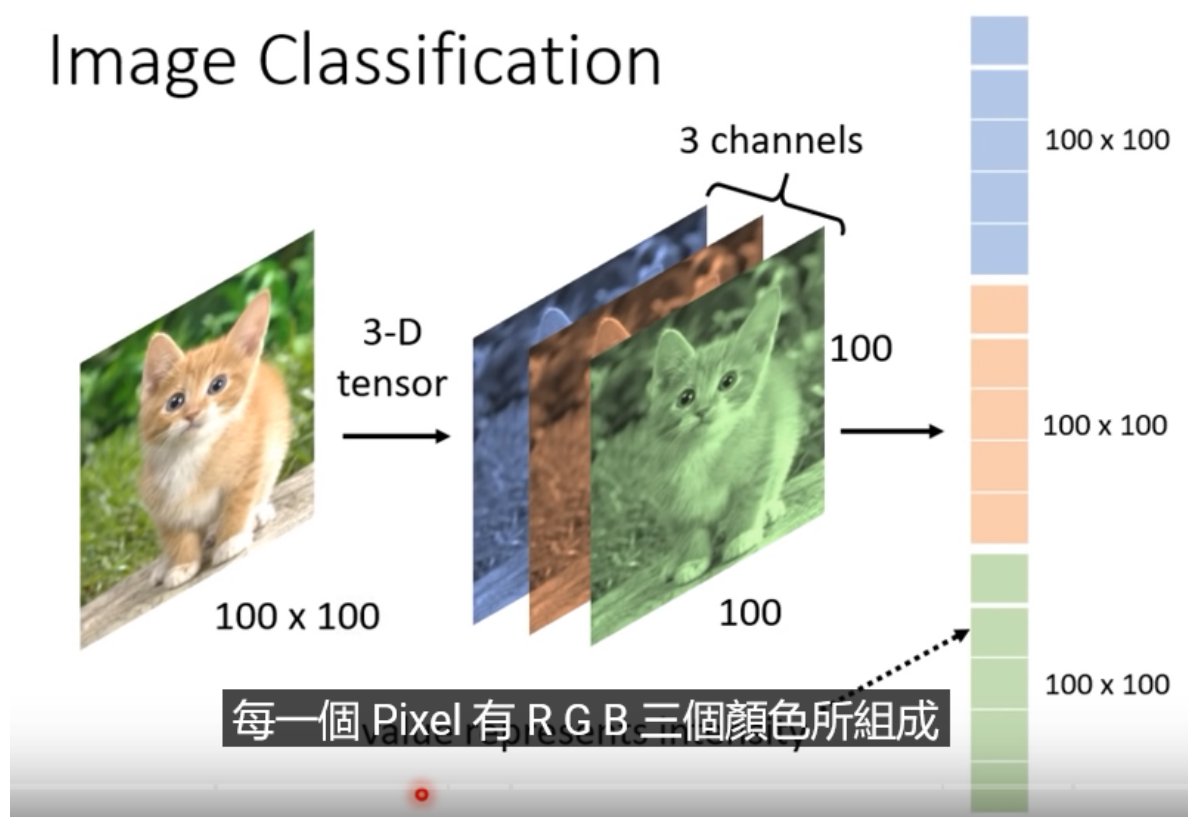


CNN(Convolutional Neural Networks)

1.图像基础

Image Classification



长、宽、channels

计算机处理图片需要将图片转换为向量，转换为3个Pixel，每一个里面都是代表一个channel的RGB像素情况。

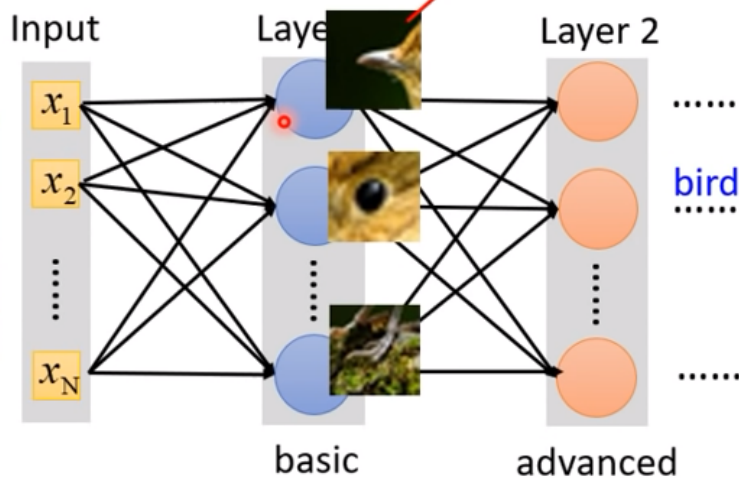
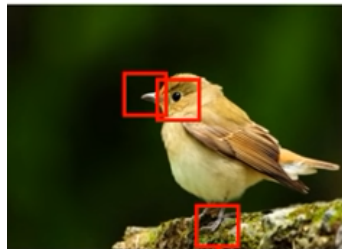
2、网络结构构建

网络结构越复杂越容易过拟合并且需要的计算资源越多，而且不需要看完整张图片其实就可以对图像做出了解

Observation 1

A neuron does not have to see the whole image.

Need to see the whole image?

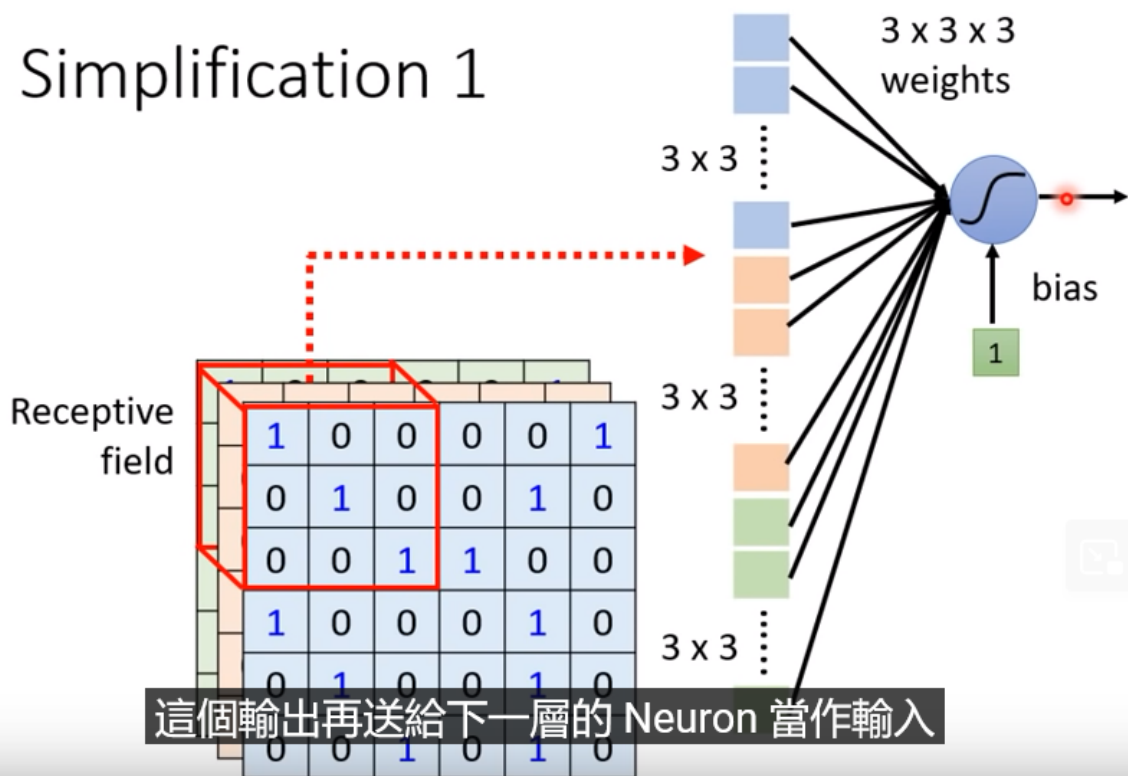


就足以讓它們偵測某些特別關鍵的 Pattern

Some patterns are much smaller than the whole image.

每一个神经元只需要关注图像中的某一部分就行

Simplification 1

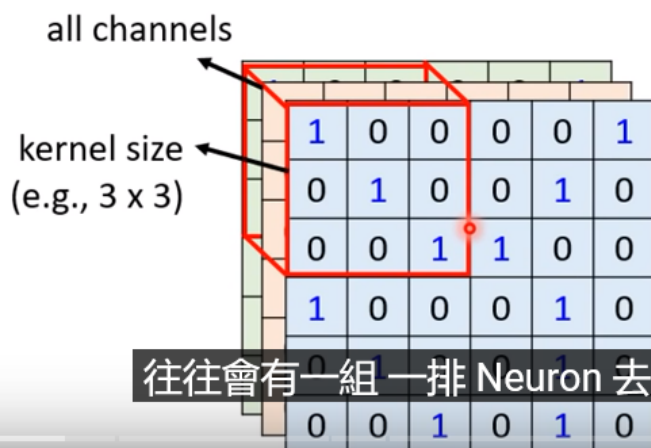


不同神经元关注可以重叠，甚至可以一模一样。

神经元关注的策略有很多种，接下来介绍一些常见的组合：

Simplification 1 – Typical Setting

Each receptive field has a set of neurons (e.g., 64 neurons).



往往會有一組一排 Neuron 去守備它

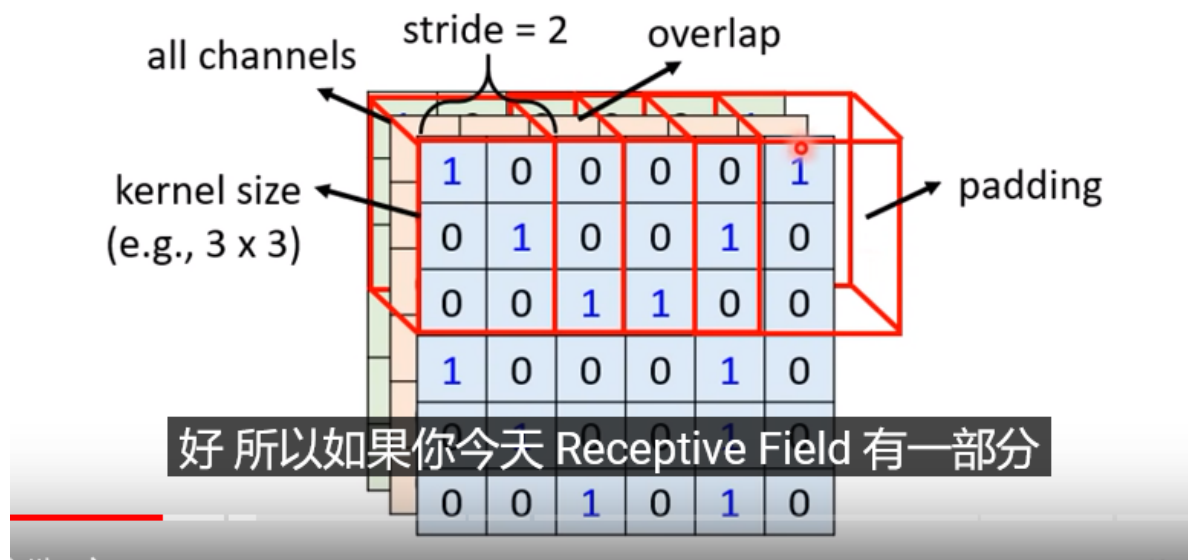
默认全通道，kernel size 一般为 3x3

stride: "Stride" (步长) 是指在图像处理和卷积运算中，滑动窗口或卷积核在输入数据上移动的步幅大小。它决定了滑动窗口在每个维度上跳过的像素数目。

padding: "Padding" (填充) 是指在图像处理和卷积运算中，在输入数据的边缘周围添加额外的像素或值。填充可以在卷积操作中保留输入数据的空间维度，并控制输出特征图的尺寸。

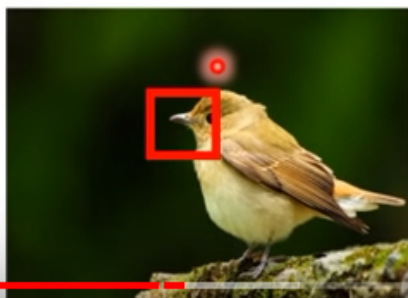
Simplification 1 – Typical Setting

Each receptive field has a set of neurons (e.g., 64 neurons).



Observation 2

- The same patterns appear in different regions.



共享参数:

Simplification 2

Diagram illustrating the simplification of a 3D convolution operation into a 2D matrix multiplication.

On the left, a 3D volume (input tensor) is shown, with a 3x3x3 kernel highlighted in red. This kernel is flattened into a 27x1 vector.

On the right, a 2D matrix (kernel tensor) is shown, with the same 3x3x3 kernel highlighted in red. This matrix is multiplied by a 6x1 vector of input features (x_1, x_2, \dots, x_6) to produce a single output value.

The diagram shows the weights w_1 and w_2 , and the bias term.

所以上面這個 Neuron

Each receptive field has a set of neurons (e.g., 64 neurons).

[illegible]

使用filter计算：（卷积神经网络中的滤波器：在卷积神经网络（CNN）中，滤波器通常被用于卷积层进行特征提取。滤波器在卷积过程中通过与输入数据进行卷积运算，可以检测输入数据中的不同特征，如边缘、纹理等。CNN 中的滤波器也被称为卷积核（Convolutional Kernel）或特征映射（Feature Map））

Convolutional Layer

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

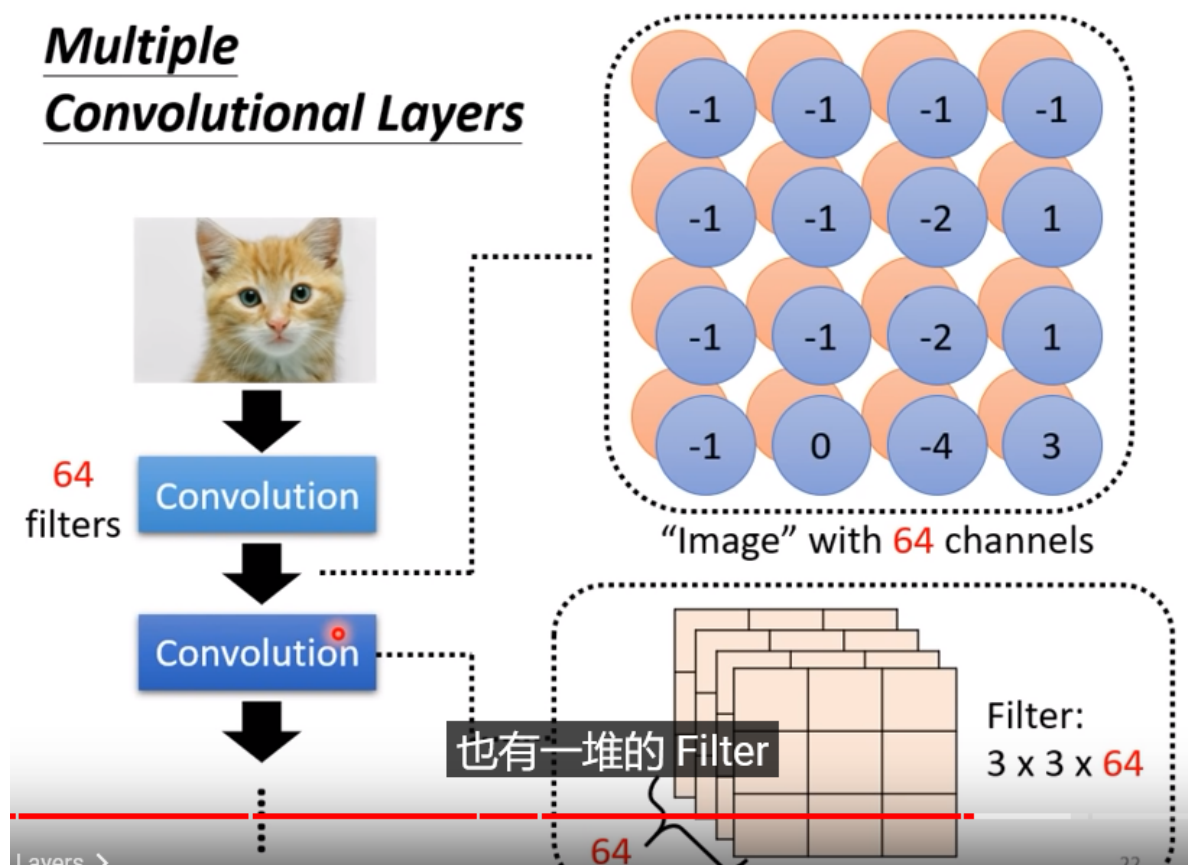
stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

Multiple Convolutional Layers



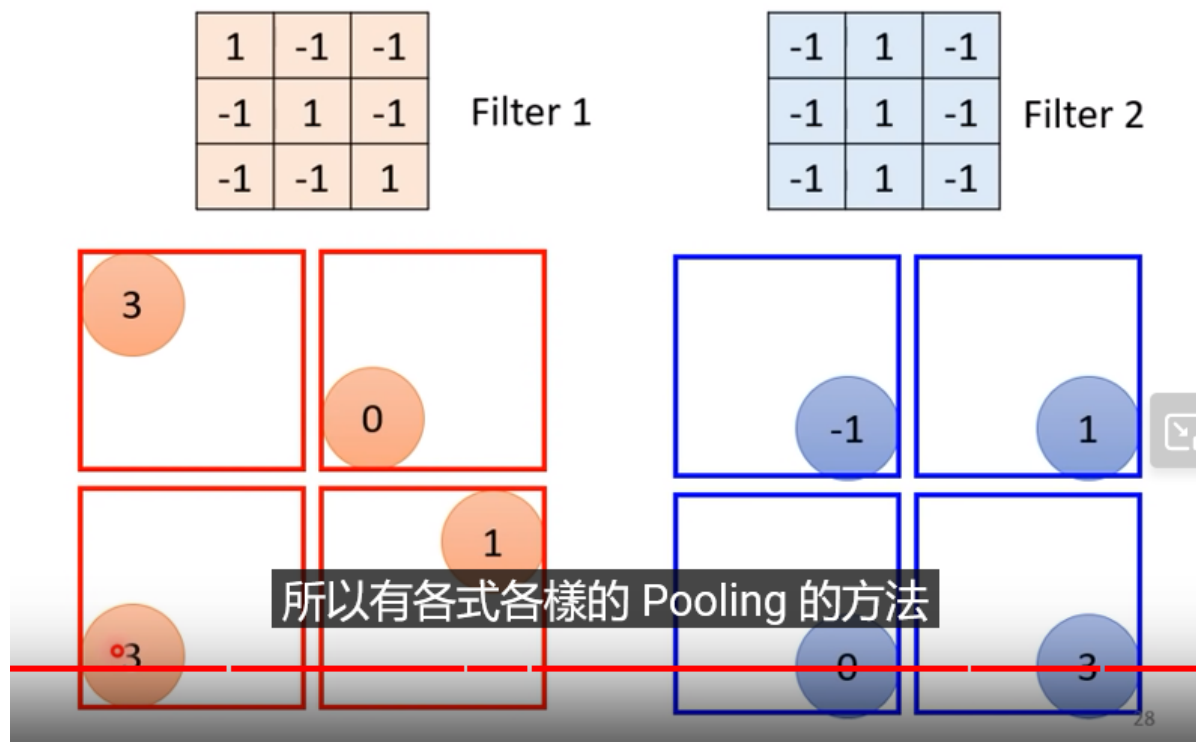
使用多个filters去做，得到 feature map image，再对feature map image使用filter去做，不断迭代，filter的高度与图片的channels相同。

Pooling (max pooling):

Pooling（池化）是深度学习中常用的一种操作，用于减少特征图的空间尺寸、降低计算复杂度，并提取特征的空间层级信息。在卷积神经网络（CNN）中，池化层通常紧跟在卷积层之后，对卷积层的输出进行下采样。池化操作通过在输入特征图上滑动一个固定大小的窗口，并对窗口内的数值进行汇聚（如最大值、平均值等），从而生成池化后的输出特征图。

常见的池化操作包括最大池化（Max Pooling）和平均池化（Average Pooling）。

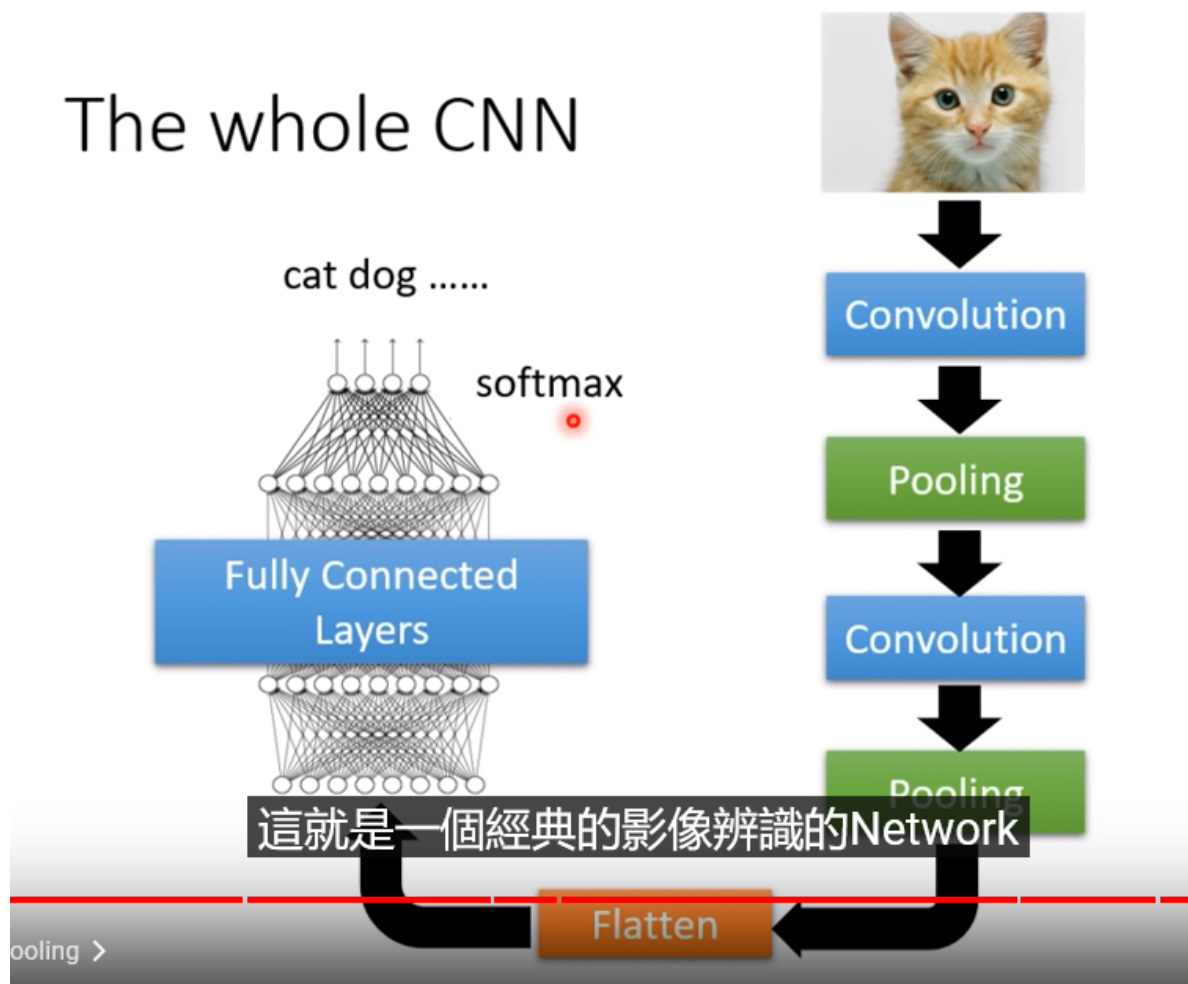
Pooling – Max Pooling



当做完convolution后我们会做一个pooling工作，一般的策略为convolution和pooling交替使用。

（一般计算资源充足的情况下不做pooling）

The whole CNN



"Flatten"（扁平化）是深度学习中常用的一种操作，用于将多维的输入数据转换为一维的向量形式。

在神经网络中，通常使用多维的输入数据，如图像数据可以表示为三维的张量（宽度 × 高度 × 通道）。然而，某些层（如全连接层）需要接收一维的输入数据，因此需要将多维数据转换为一维向量，这个过程就被称为扁平化。