



深度學習與電腦視覺 學習馬拉松

cupay 陪跑專家：楊哲寧



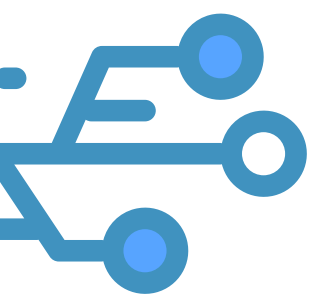
深度學習理論與實作

CNN原理：卷積、步長、填充

重要知識點



- 步長(Strides)與填充(Padding)的原理
- 如何透過步長(Strides)與填充(Padding) 控制卷積大小
- 輸出 Feature map 大小的計算

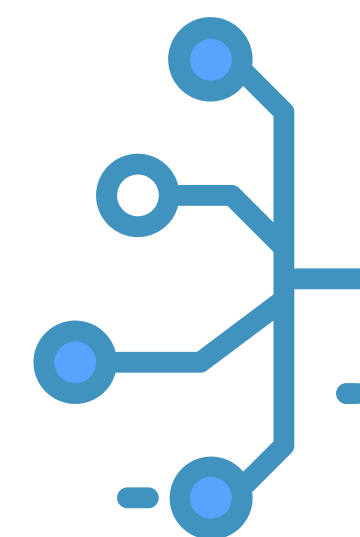
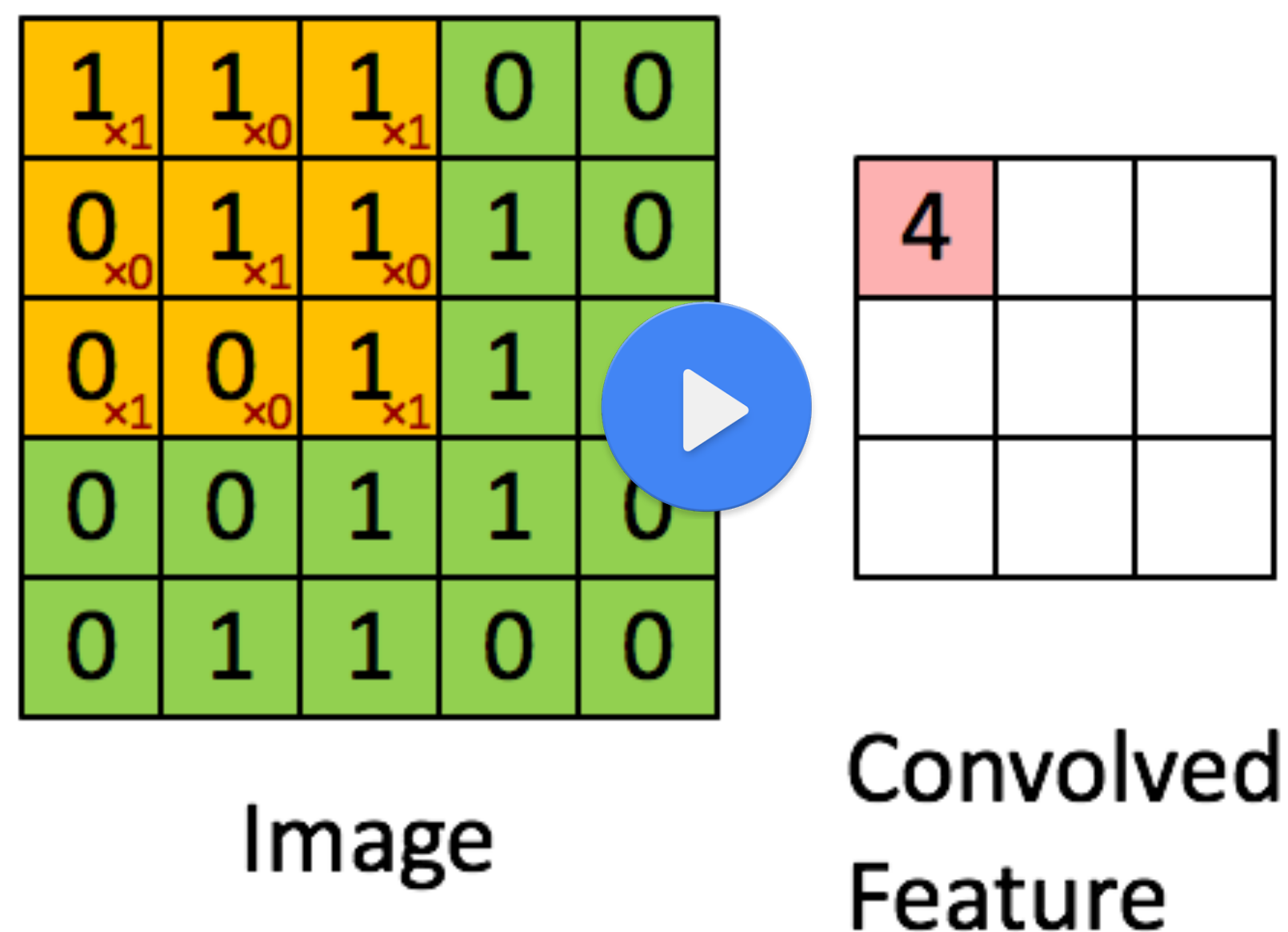


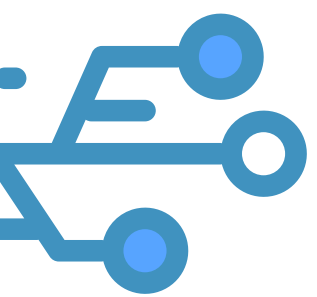
步長與填充

我們先來看看下圖：

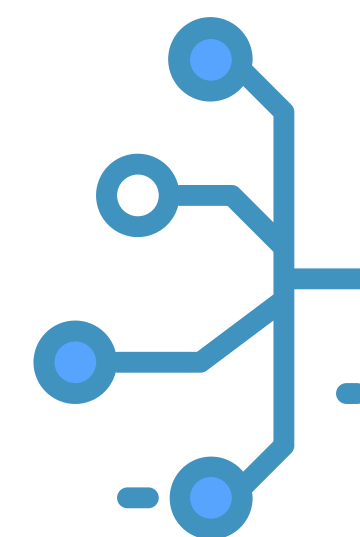
步長(Strides)：圖中 Kernel 的步長 (Strides) 在 height、width 均為 1，可以看到黃色 Kernel 往右、下都是一格像素。

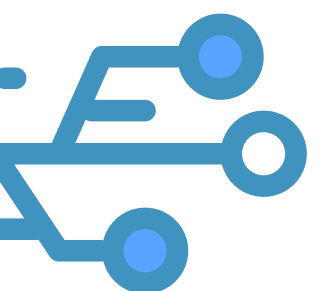
填充 (Padding)：圖中並沒有使用任何 Padding，因此可以看到原圖周圍並沒有補 0 的像素，而輸出的 Feature map 長寬也下降。





那步長與填充的運作原理是什麼呢？





步長(Strides)

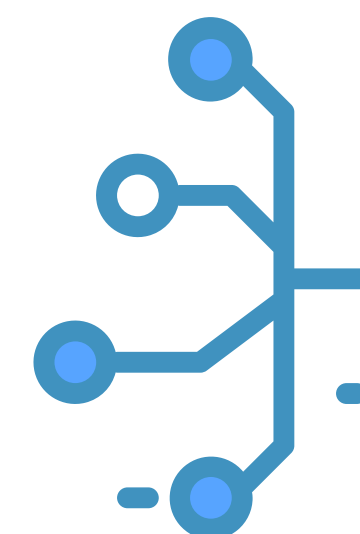


- **Strides**：控制 Kernel 在圖像不同 Dimention 上移動的距離。
- **Keras Convolution2D**：其中有一個可調參數為 **Strides**，可以針對 **Height**、**Width** 賦予不同的值，藉此控制輸出 **Feature map** 高、寬的尺度變化。

```
Init signature: Convolution2D(filters, kernel_size, strides=(1, 1), padding='valid', data_format=None, dilation_rate=(1, 1), activation=None, use_bias=True, kernel_initializer='glorot_uniform', bias_initializer='zeros', kernel_regularizer=None, bias_regularizer=None, activity_regularizer=None, kernel_constraint=None, bias_constraint=None, **kwargs)
```

Docstring:

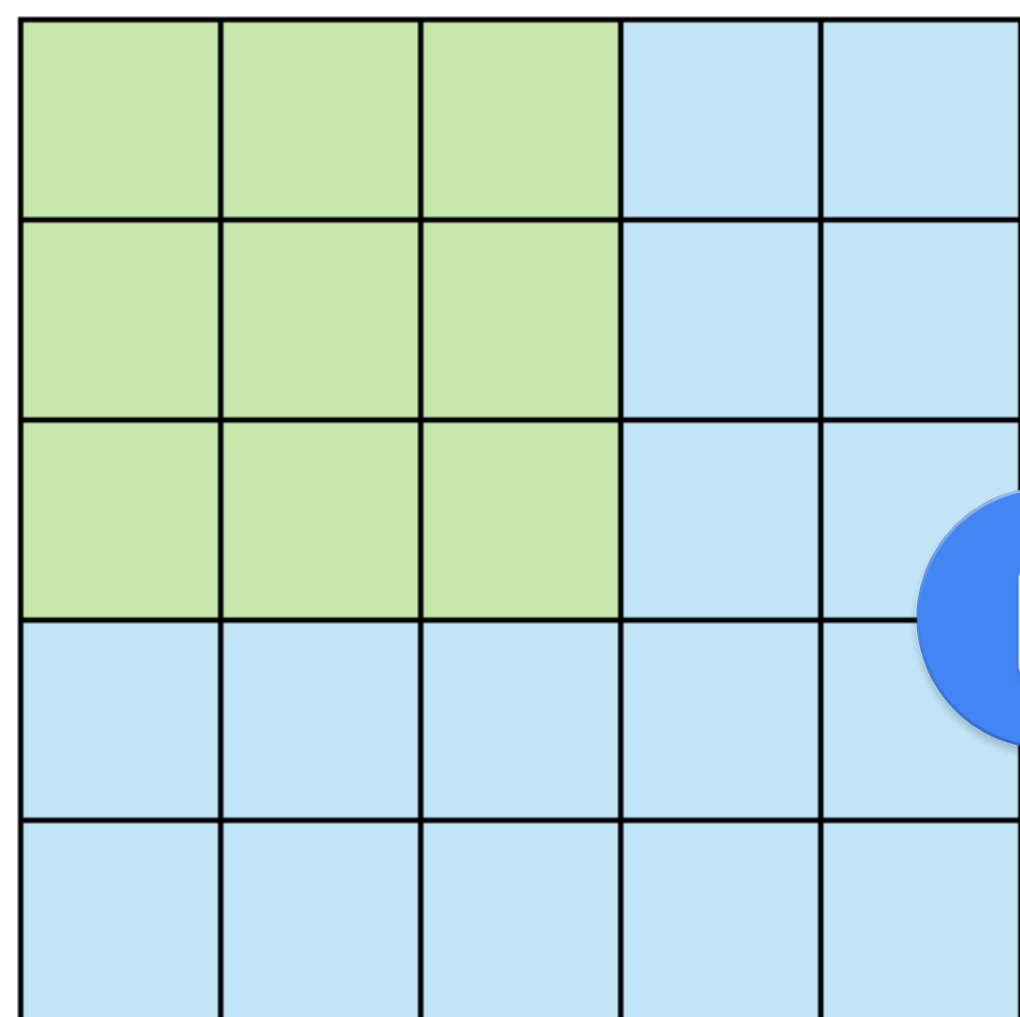
2D convolution layer (e.g. spatial convolution over images).



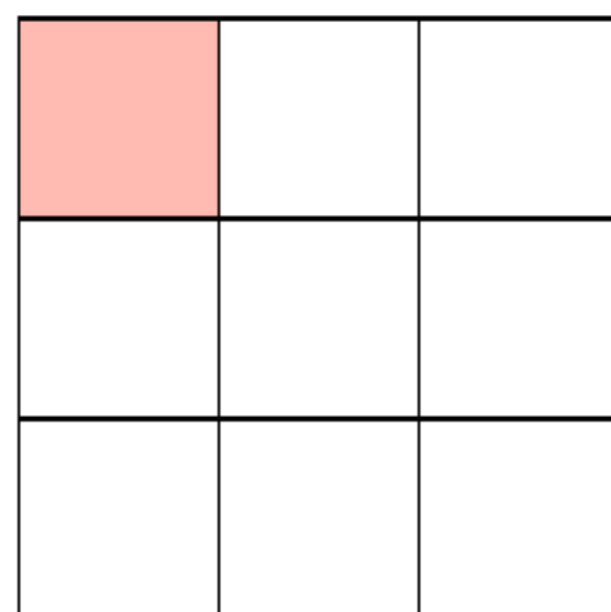


步長(Strides)

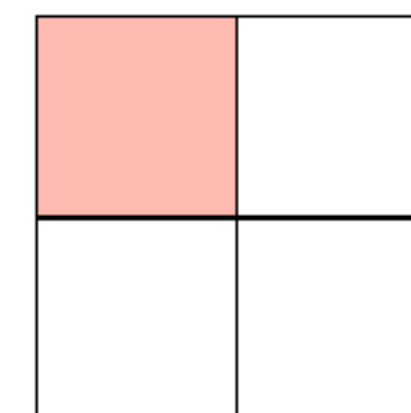
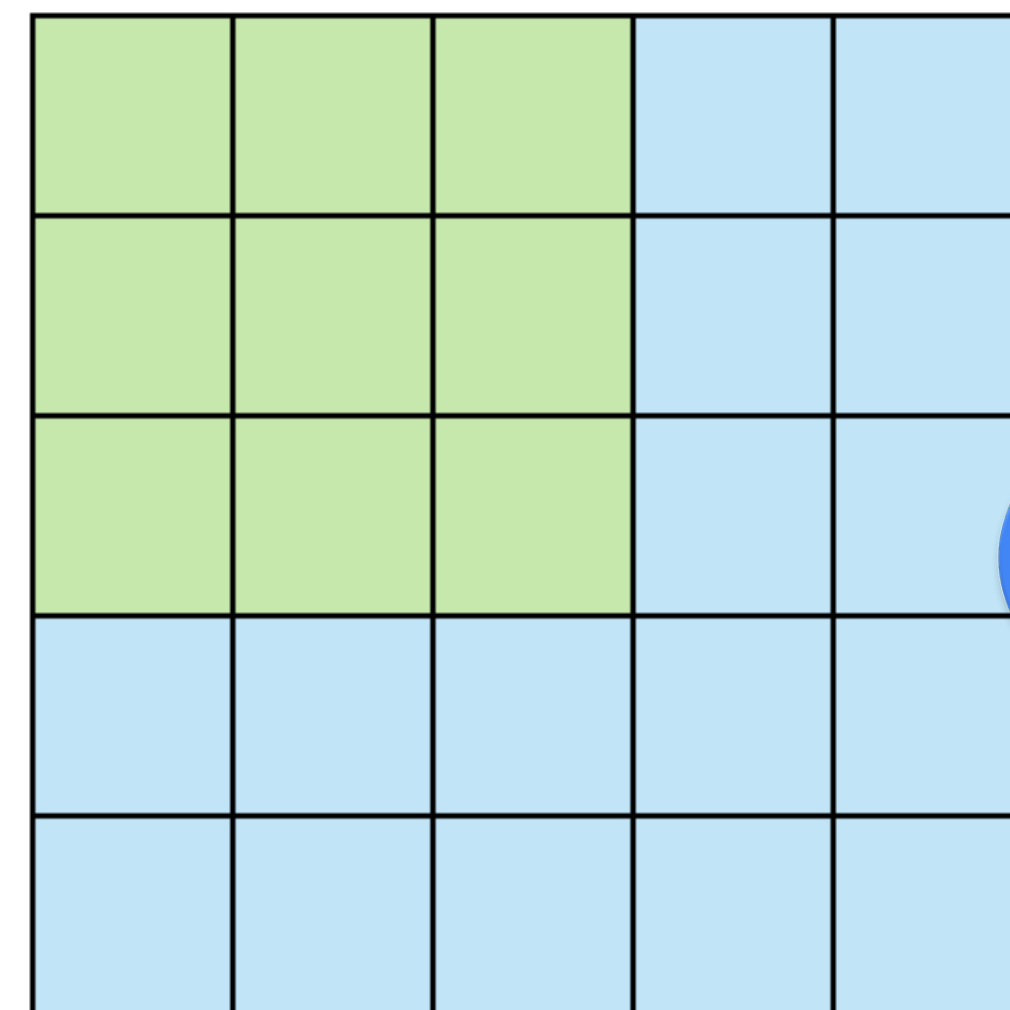
下圖中運用不同的步長(1,1)與(2,2)，可以發現輸出的 Feature map 尺寸也有所不同。



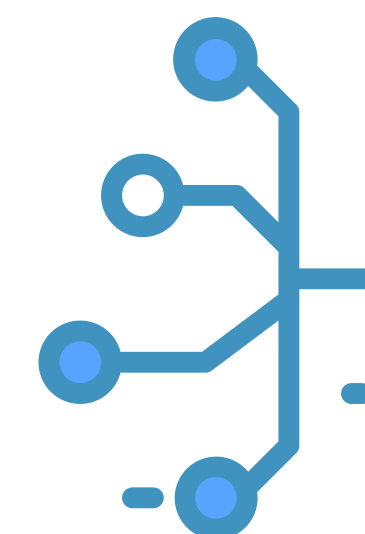
Stride 1

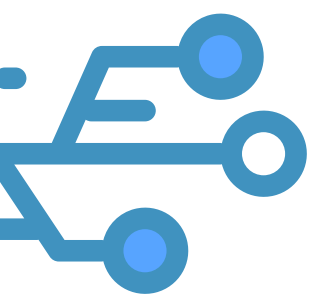


Feature Map



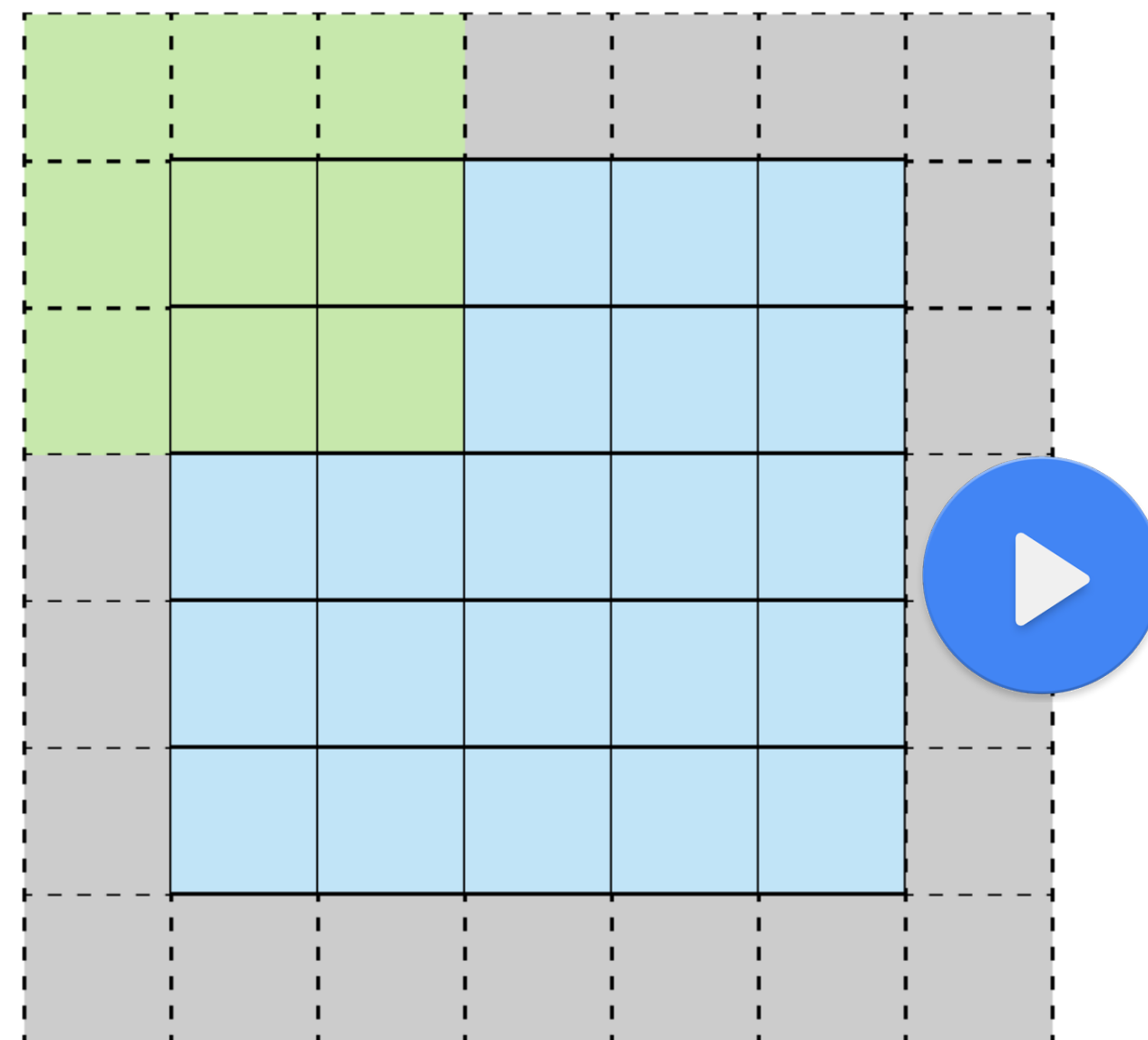
Feature Map



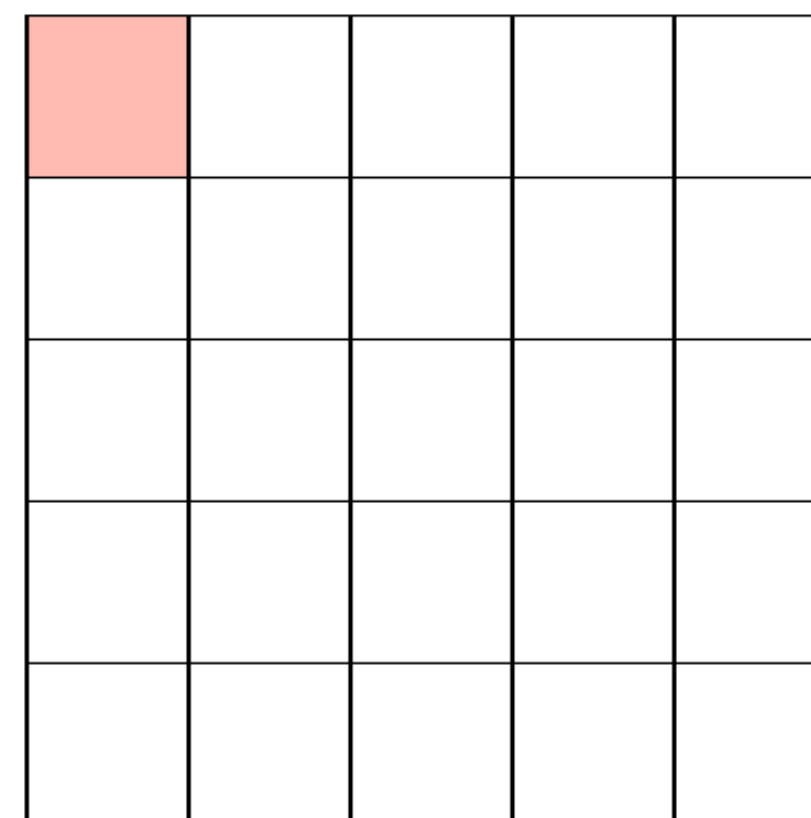


填充 (Padding)

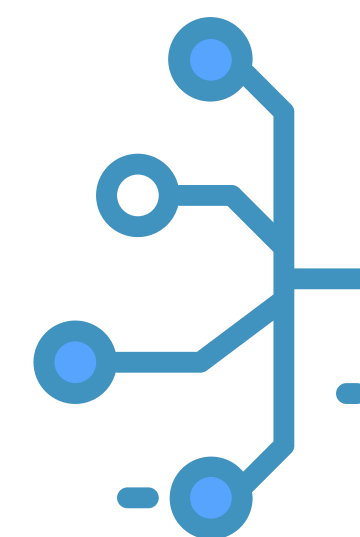
Padding：Padding 的用途主要在於避免圖像尺寸下降，而為了避免干擾圖像資訊，通常Padding為補 0 的像素，而 **Padding=1** 就是在圖像周圍補一圈值為 0 的像素，也就是圖中灰色的區域。



Stride 1 with Padding



Feature Map

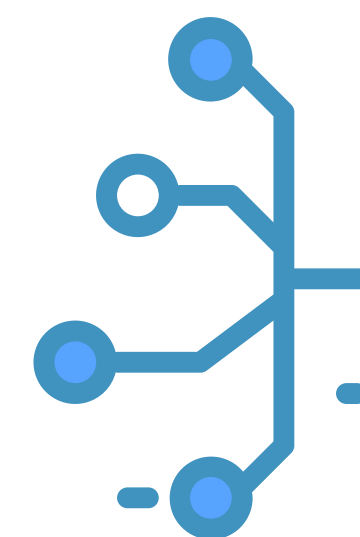




填充 (Padding)



而我們在操作 Keras 的 Convolution2D 時，會發現默認為『**Valid**』而一
多數人會用『**SAME**』，這是什麼意思呢？



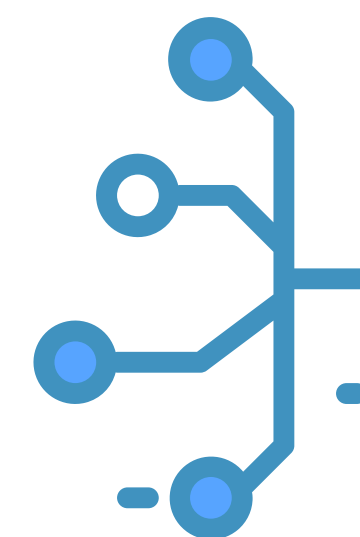
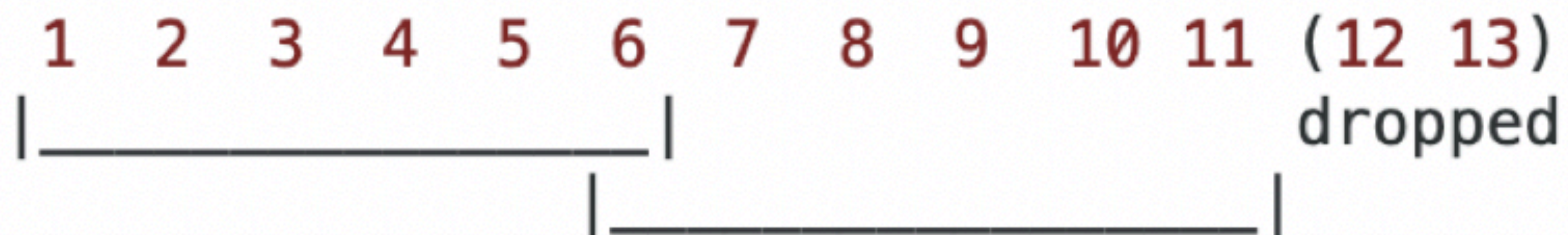


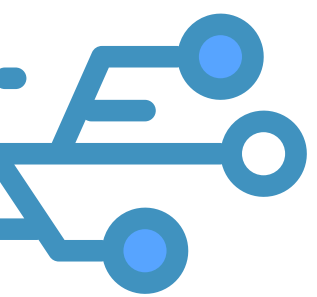
填充 (Padding)

- **Valid**：就是不去 Padding，多的像素直接捨去。像是下圖中，可以想像原圖為 13×13 ，kernel 大小為 6×6 ，步長為 $(5, 5)$ ，當Kernel要跨出第二步時，只剩下 **2** 個像素(12、13)，a就直接捨去。

- "VALID" = without padding:

inputs:

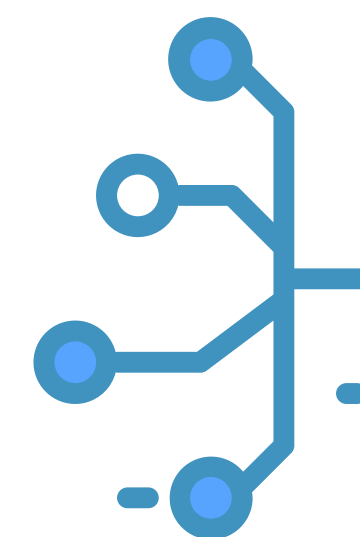




填充 (Padding)



SAME：透過補邊讓輸出長寬==原圖長寬/Strides，什麼意思呢？假如我們使用Strides=(1,1)，那麼不管使用多大的Kernel，輸出 Feature map 的寬、高等於輸入影像的寬高。





填充 (Padding)



公式化如下：

『**SAME**』



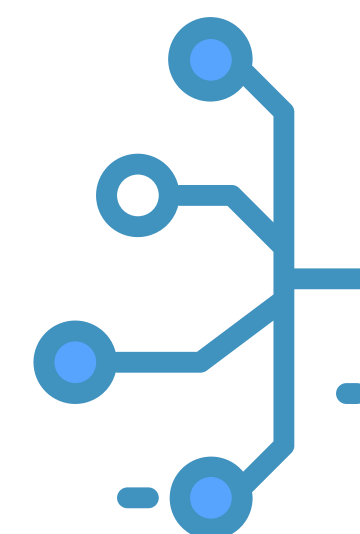
$\text{out_height} = \text{ceil}(\text{float}(\text{in_height}) / \text{float}(\text{strides}[\textcolor{red}{1}])))$

$\text{out_width} = \text{ceil}(\text{float}(\text{in_width}) / \text{float}(\text{strides}[\textcolor{red}{2}])))$

『**Valid**』

$\text{out_height} = \text{ceil}(\text{float}(\text{in_height} - \text{filter_height} + 1) / \text{float}(\text{strides}[\textcolor{red}{1}])))$

$\text{out_width} = \text{ceil}(\text{float}(\text{in_width} - \text{filter_width} + 1) / \text{float}(\text{strides}[\textcolor{red}{2}])))$

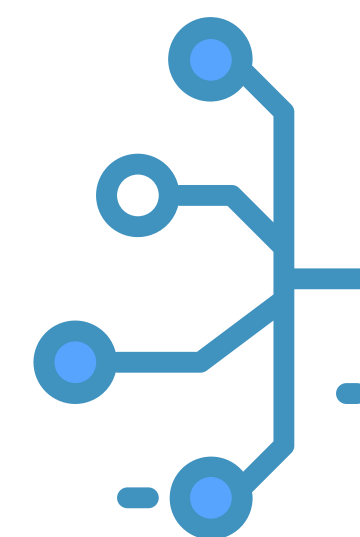


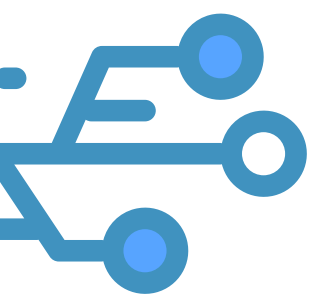


輸出Feature map尺寸



$$\text{Output} = (\text{Input} + 2 * \text{padding} - \text{Kernel_Size}) / \text{Stride} + 1$$





推薦延伸閱讀



DeepLearning.ai- C4W1L01

吳恩達課程

[PPT連結](#)

[Youtube影片連結](#)



deeplearning.ai

Stanford-CS231n

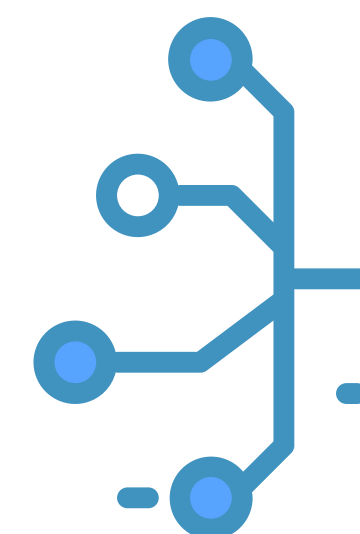
史丹福大學課程

[PDF連結](#)

[Youtube影片連結](#)



兩個課程中皆介紹許多DeepLearning CV領域相關知識，有些部分超過課程進度，學員們可以斟酌觀看，不懂的部分也可以直接在共學社團上詢問。



解題時間 Let's Crack It



請跳出 PDF 至官網 Sample Code & 作業開始解題