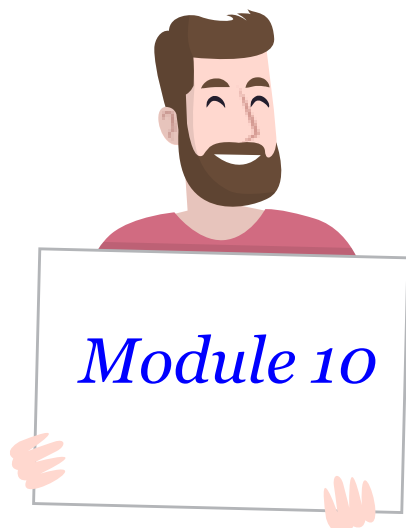




CNN神經網路介紹



designed by  freepik

Estimated time:
45 min.



資訊工業策進會 Institute for Information Industry

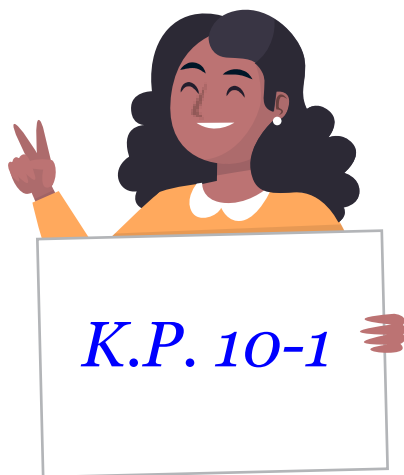
學習目標

- 10-1: CNN神經網路的優勢
- 10-2: Convolution
- 10-3: Maxpooling



10-1: CNN神經網路的優勢

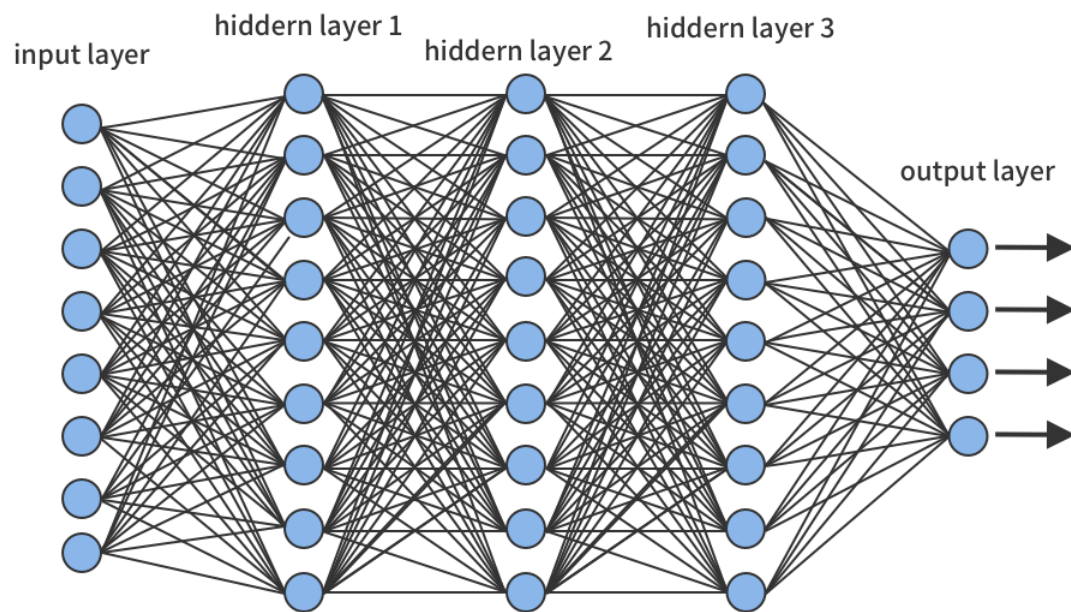
- DNN網路的缺點
- CNN網路介紹
- CNN網路架構



designed by freepik

DNN網路的缺點

- 缺點1：DNN網路隨著層數的增加，網路參數的數量也會增加非常快
 - 參數越多代表需要越多記憶體才能跑得動



DNN網路的缺點

- 缺點2：DNN網路在做圖片工作的時候，常常需要把一張圖表示成一個向量，才能輸入網路當中
 - 這會破壞圖片裡物件的空間性，對於影像工作來說會受到不小影響



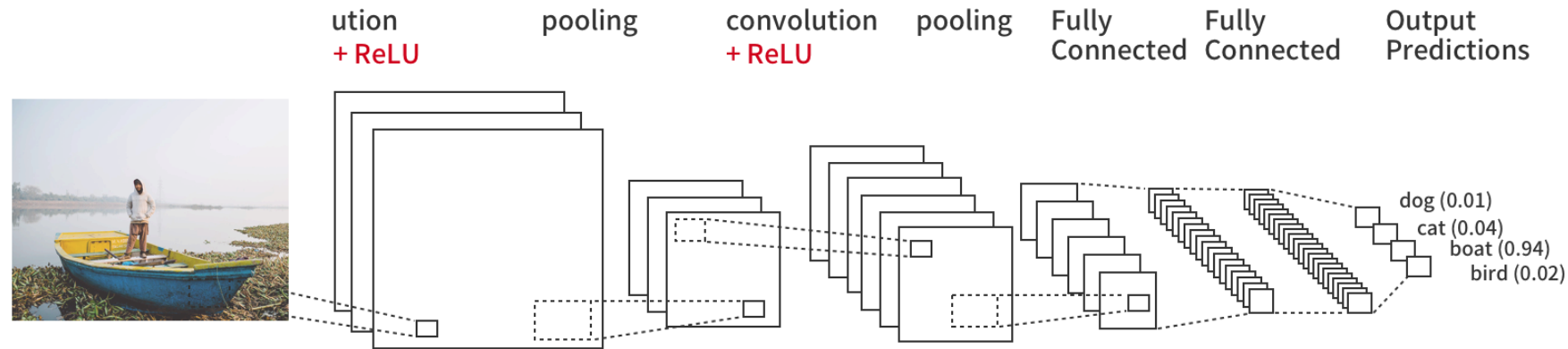
同一個物件在照片裡的像素彼此距離應該很近

CNN網路的特性

- **Convolutional Neural Networks(CNN)網路為一種非常著名的神經網路**
 - 常被用來處理與圖像有關之工作
 - 此網路包含了很多卷積層以及池化層的運算

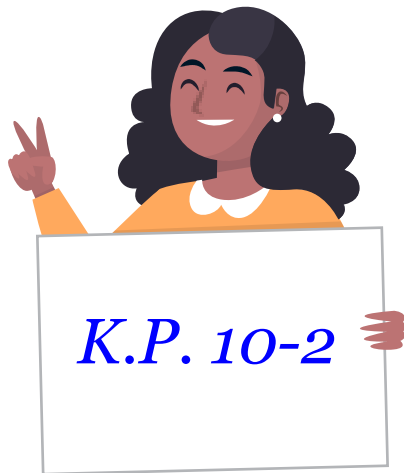
CNN網路架構

- CNN網路在經過多次的卷積層以及池化層後，最後會送入全連接層做分類
 - 卷積層裡面包含了許多卷積的運算，用來提取圖片特徵
 - 池化層裡面包含了許多池化的運算，用來壓所圖片特徵



10-2: Convolution

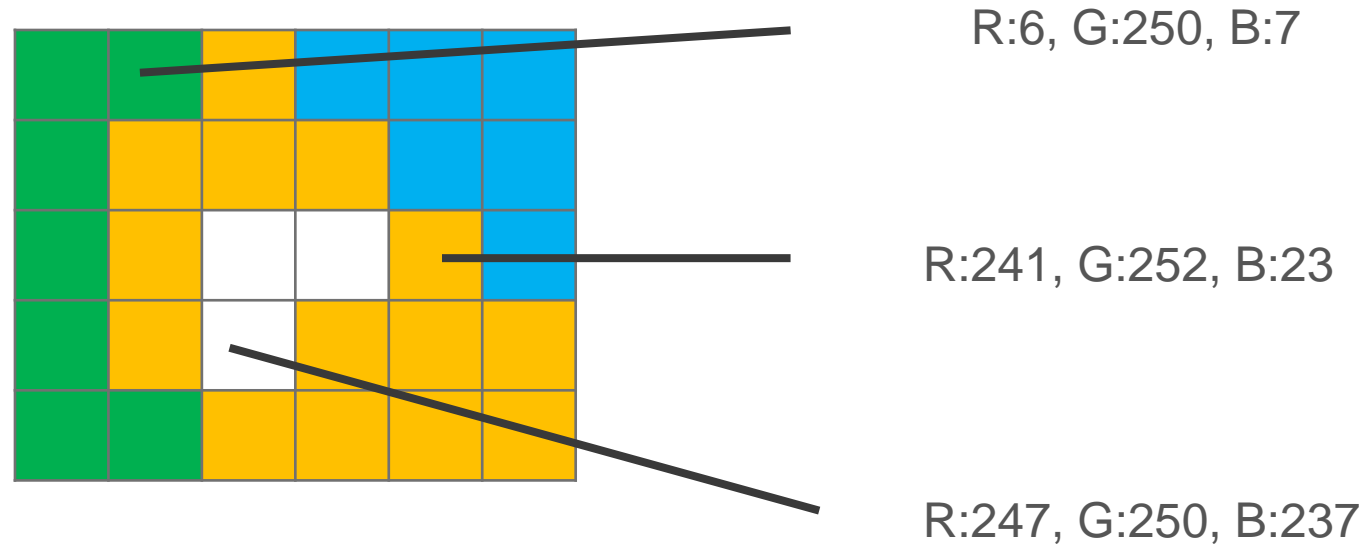
- 圖片組成方式
- Convolution運算方式
- Convolution物理意義



designed by freepik

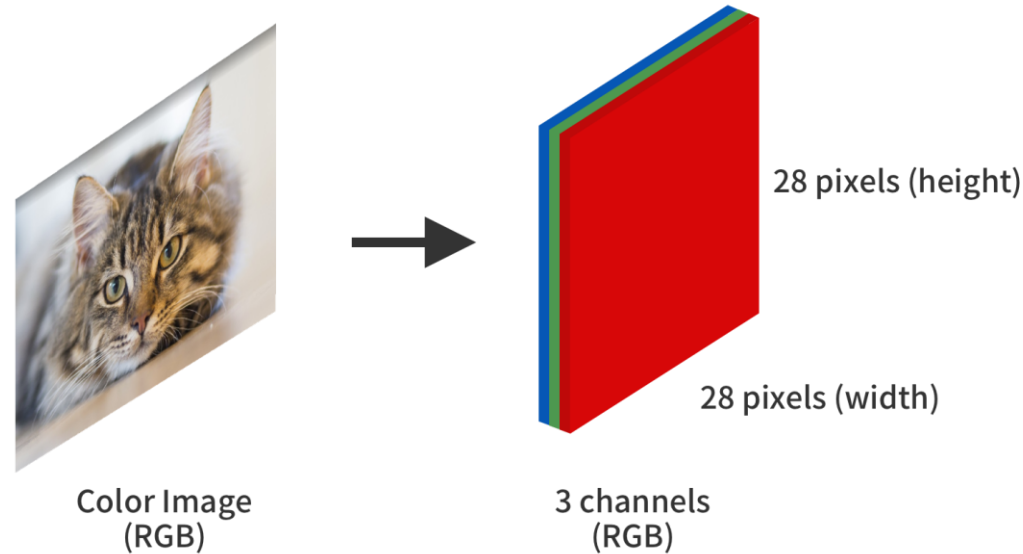
圖片組成方式

- 一張照片是由多個像素所組成
 - 如果編碼方式是以RGB編碼的話，每個像素包含了RGB三個數值
- 每個RGB編碼數值介於0~255之間



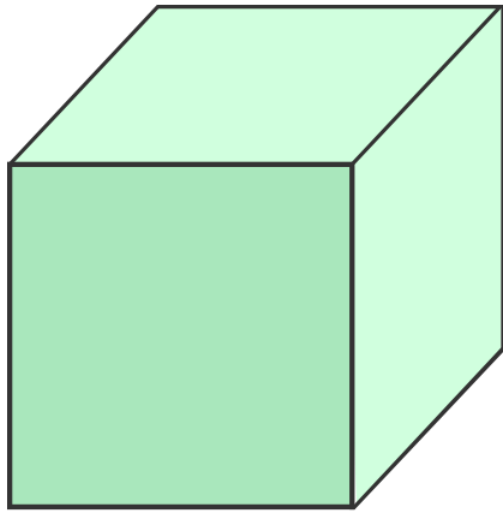
圖片組成方式

- 一張RGB彩色照片可以視為是一個三維度的數據組
 - 照片長*照片寬*照片深度
 - 或也可以想像成是3D張量



圖片組成方式

- 在**CNN**網路裡，其輸入的照片格式需為**3D**的張量
- 如果輸入是一個批次的照片，那麼在**CNN**網路裡，則被視為**4D**張量



3d-tensor

[照片長, 照片寬, 照片深度]

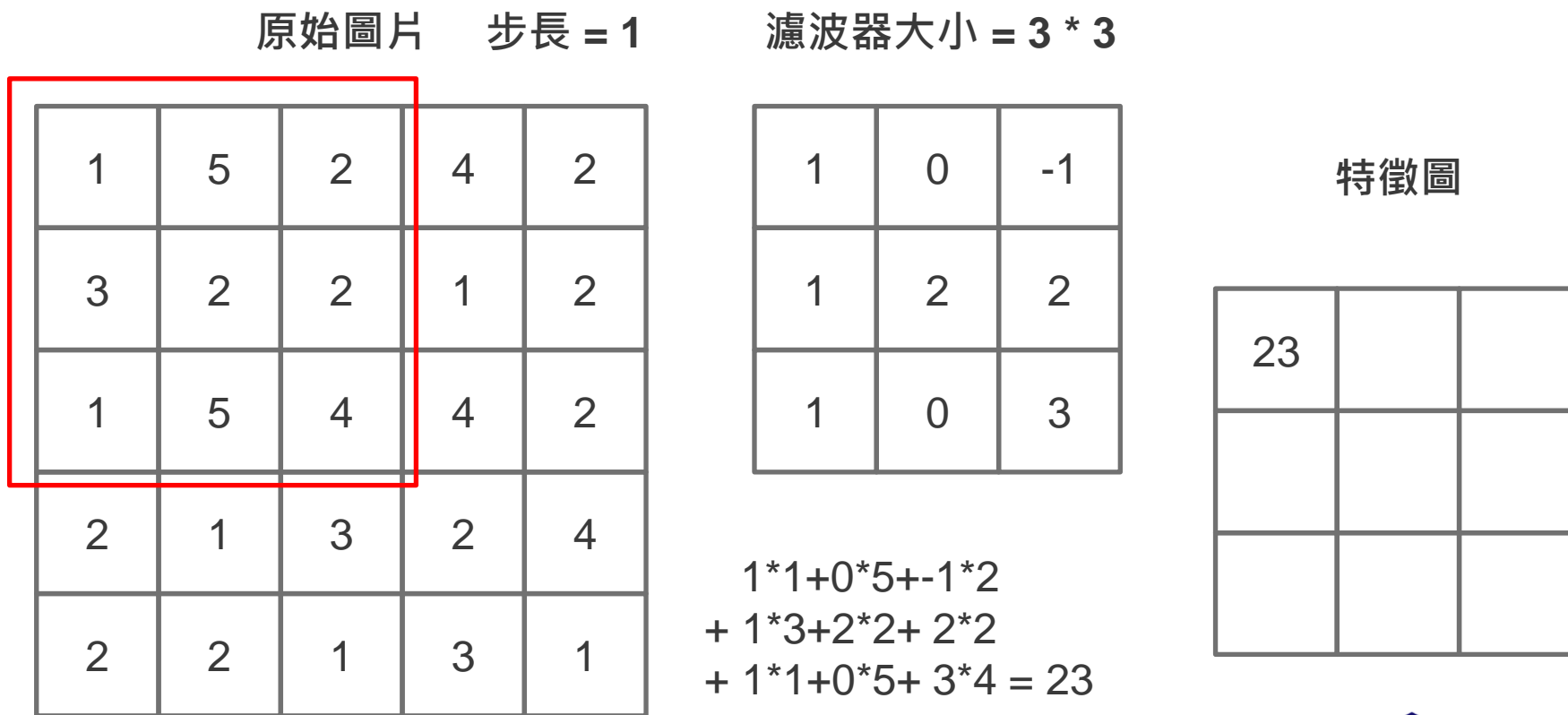


4d-tensor

[批次大小, 照片長, 照片寬, 照片深度]

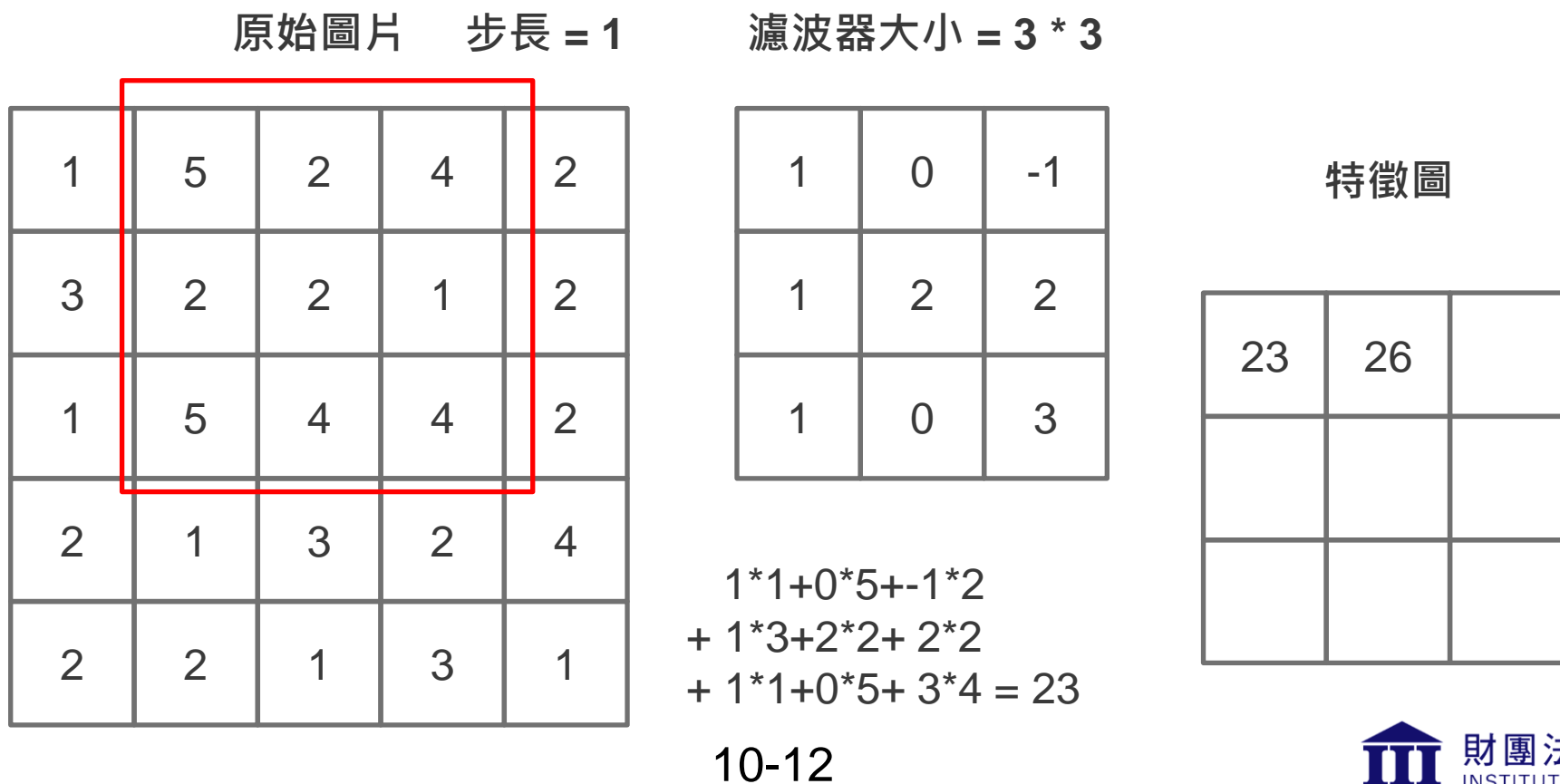
Convolution 運算方式

- 假設有一個單通道的圖片，我們可以設定一個濾波器的大小、步長來做來做Convolution
 - 傳統做Convolution時，濾波器裡面的數字通常都是人為設定好



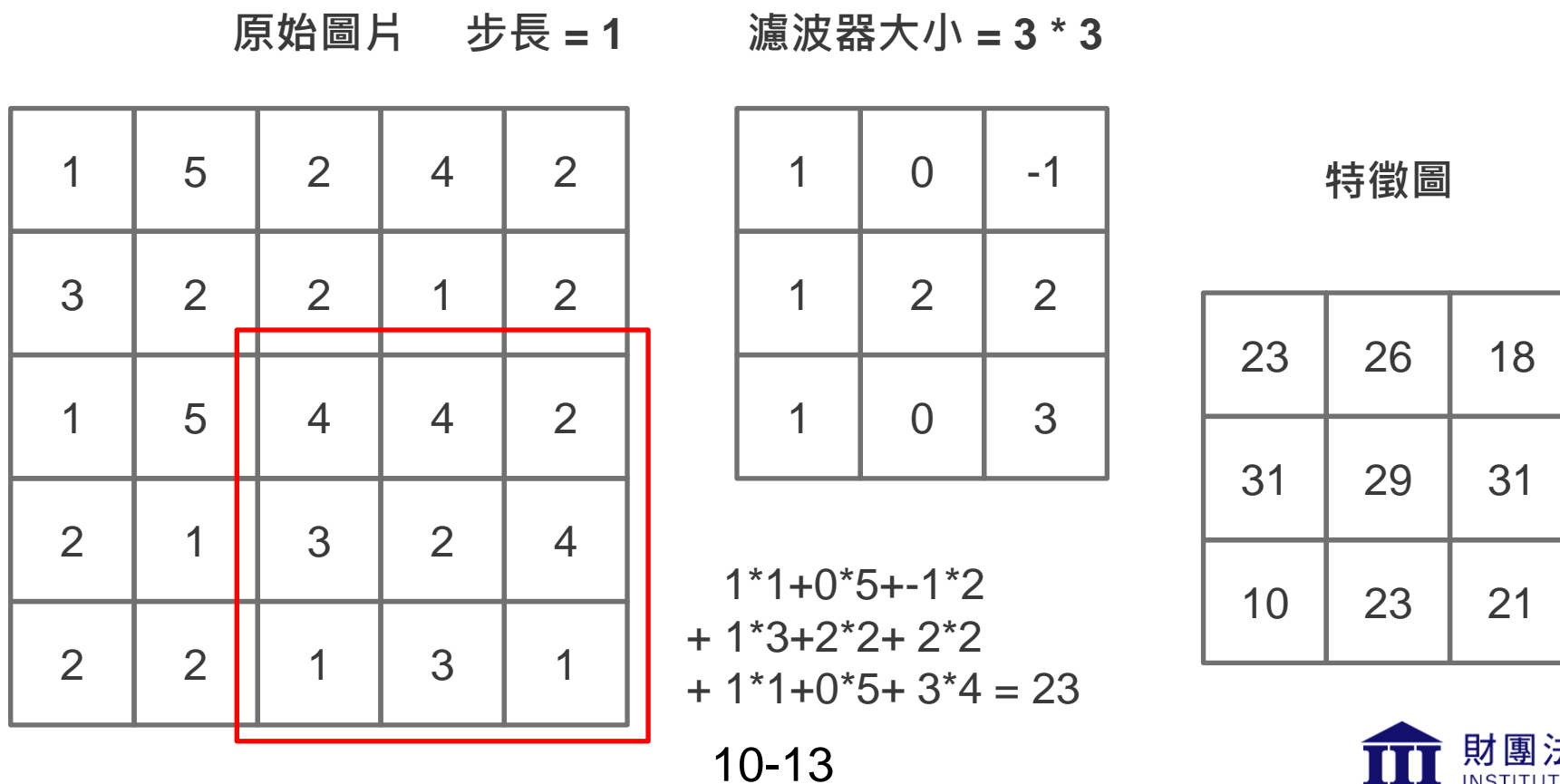
Convolution 運算方式

- 做完一次計算以後，依據步長的大小往右/往下滑動，再做一次計算



Convolution 運算方式

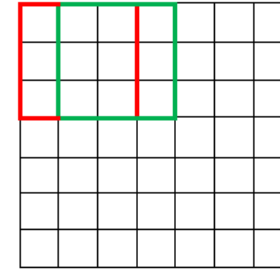
- 將每個位置都做完計算後，可以得到一張完整的特徵圖
 - 此特徵圖代表原是圖片某部分的特徵(後面會更詳細說明)



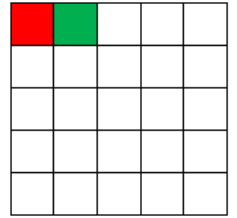
Convolution的參數

- 濾波器大小
- 步長
 - 每次滑動多少步
- 填充
 - 將原始圖片在做卷積前，照片周圍加上幾圈0

7 x 7 Input Volume

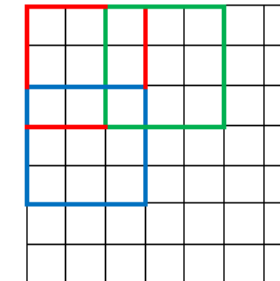


5 x 5 Output Volume

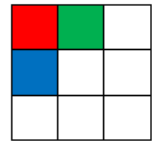


濾波器大小 = 3×3 步長 = 1

7 x 7 Input Volume



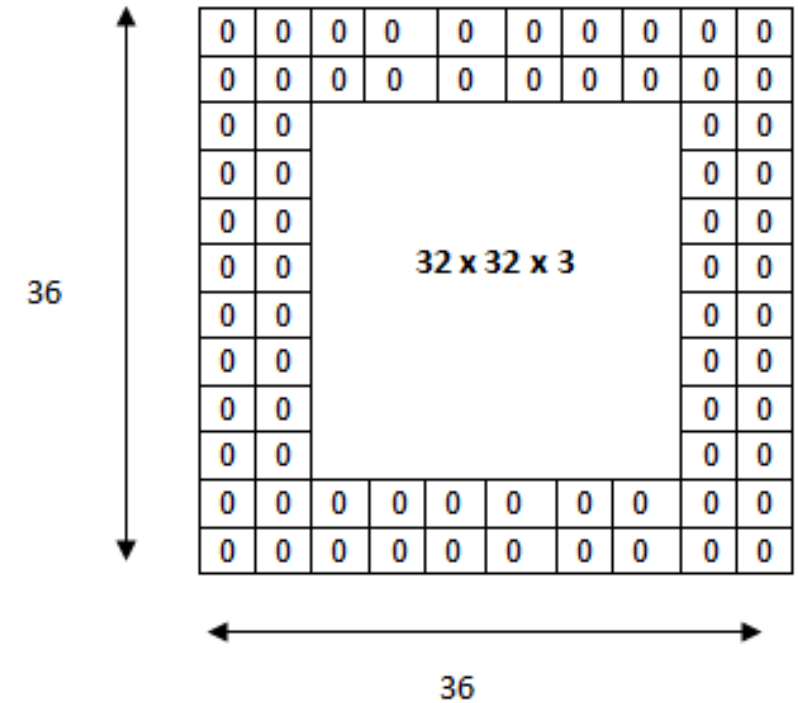
3 x 3 Output Volume



濾波器大小 = 3×3 步長 = 2

Convolution的參數

- 濾波器大小
- 步長
 - 每次滑動多少步
- 填充
 - 將原始圖片在做卷積前，照片周圍加上幾圈0



濾波器大小 = $5*5$, 步長 = 1, 填充=2

Convolution的參數

- 原始圖片以及特徵圖大小的關係可以用以下式子來計算

$$O = \frac{(W - K + 2P)}{S} + 1$$

O: 輸出特徵圖長/寬

W: 輸入照片長/寬

K: 濾波器大小

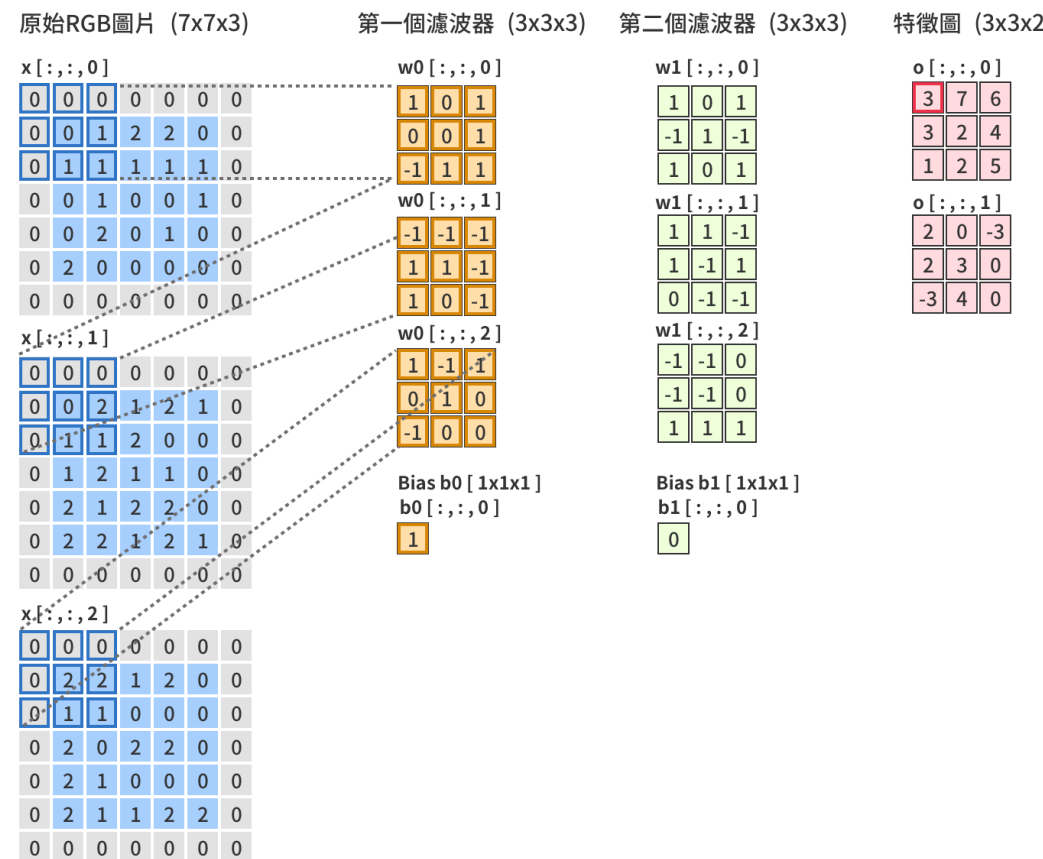
P: 填充

S: 步長

Convolution 運算方式

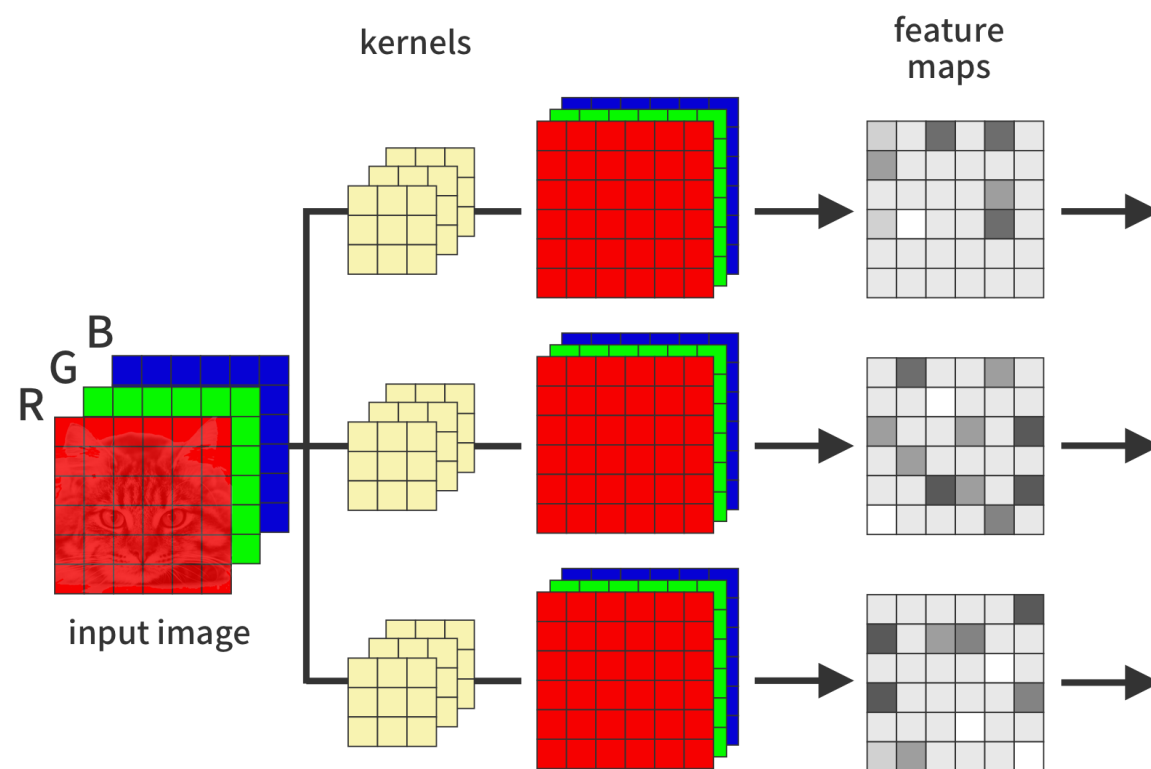
- RGB照片做卷積

- 每個濾波器是三維度數據組
- 設定多少個濾波器就會產生多少張特徵圖



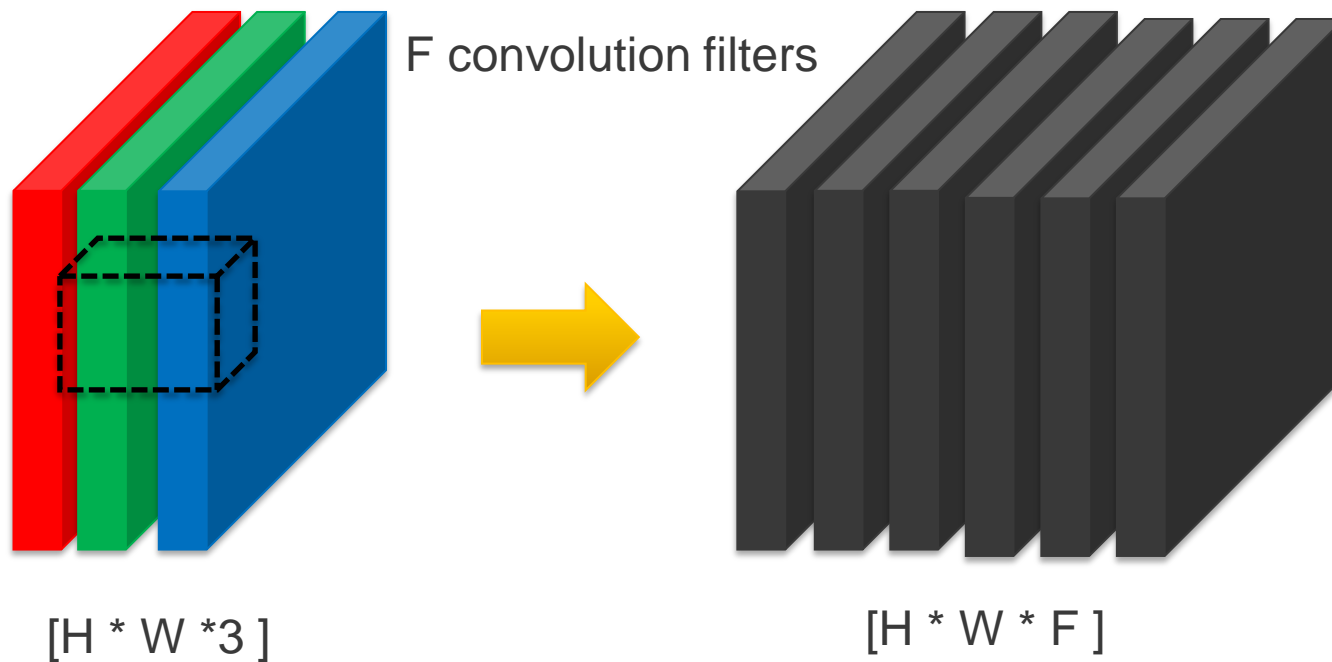
Convolution 運算方式

- 假設有一個RGB彩色照片，我們把它與5個濾波器做卷積
 - 會得到5張相對應之特徵圖



Convolution 運算方式

- RGB做卷積時，也有不少文獻會喜歡畫成以下的圖
 - RGB照片與F個卷積做完運算後，得到多個特徵圖並把他們疊起來得到[長, 寬, F]的數據組



Convolution 物理意義

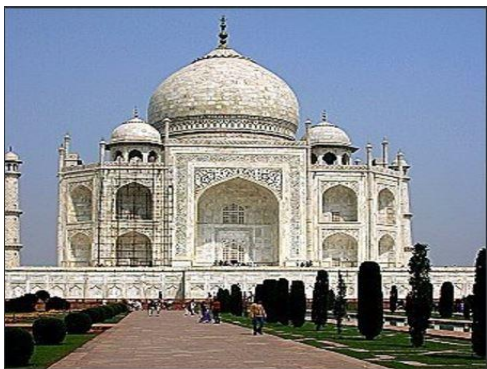
- 一張照片做卷積是有物理意義的
 - 卷積就像是抽取原始圖片某部分的特徵
 - 不同卷積的數值會有不同的效果

Convolution 物理意義

- RGB彩色照片與不同濾波器做卷積的結果

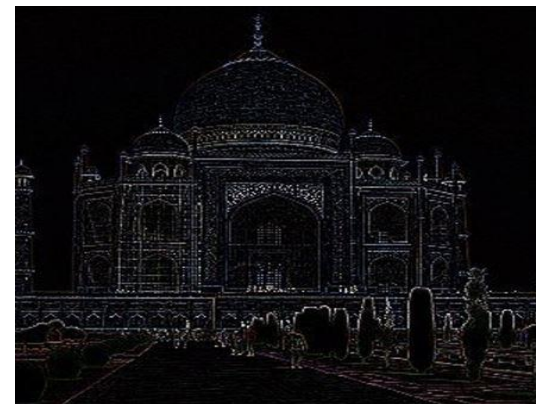
0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

尖化



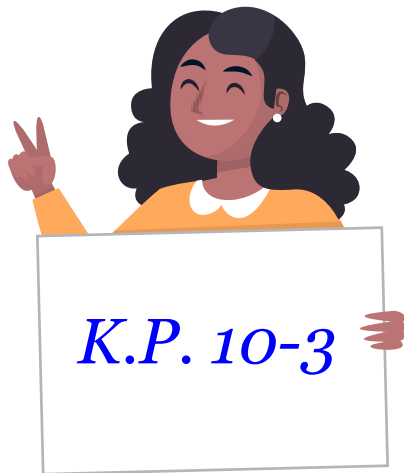
	0	1	0	
	1	-4	1	
	0	1	0	

邊緣偵測



10-3: Maxpooling

- Maxpooling運算
- Maxpooling物理意義



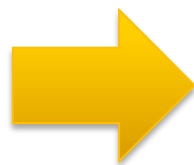
designed by freepik

Maxpooling運算

- Maxpooling運算方法跟convolution相似
 - 均需定義濾波器大小、步長，但其運算方法是將掃到的區域中提取數值最大的數字做保留
 - 可以讓原始圖片變小，讓整個CNN網路加速

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

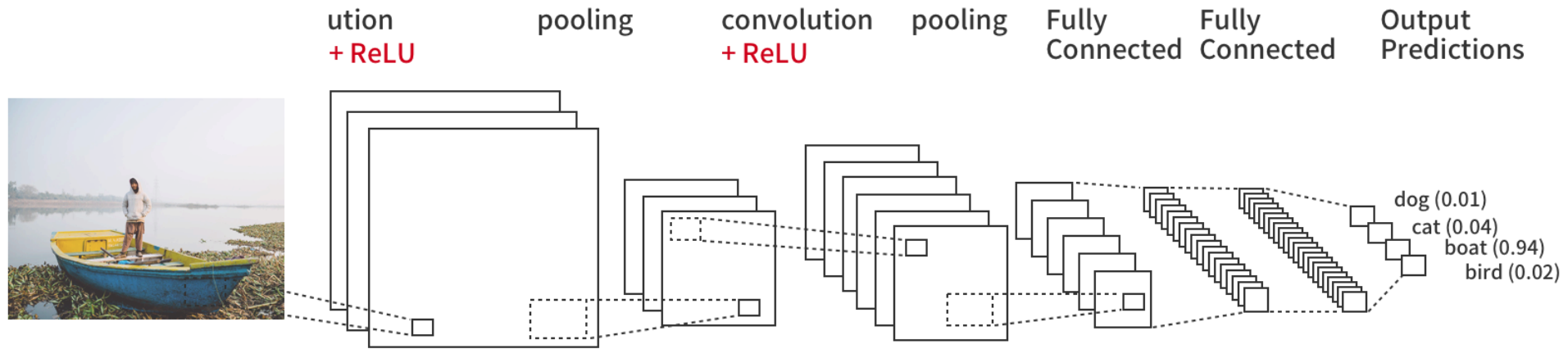
2*2濾波器且步長為2



6	8
3	4

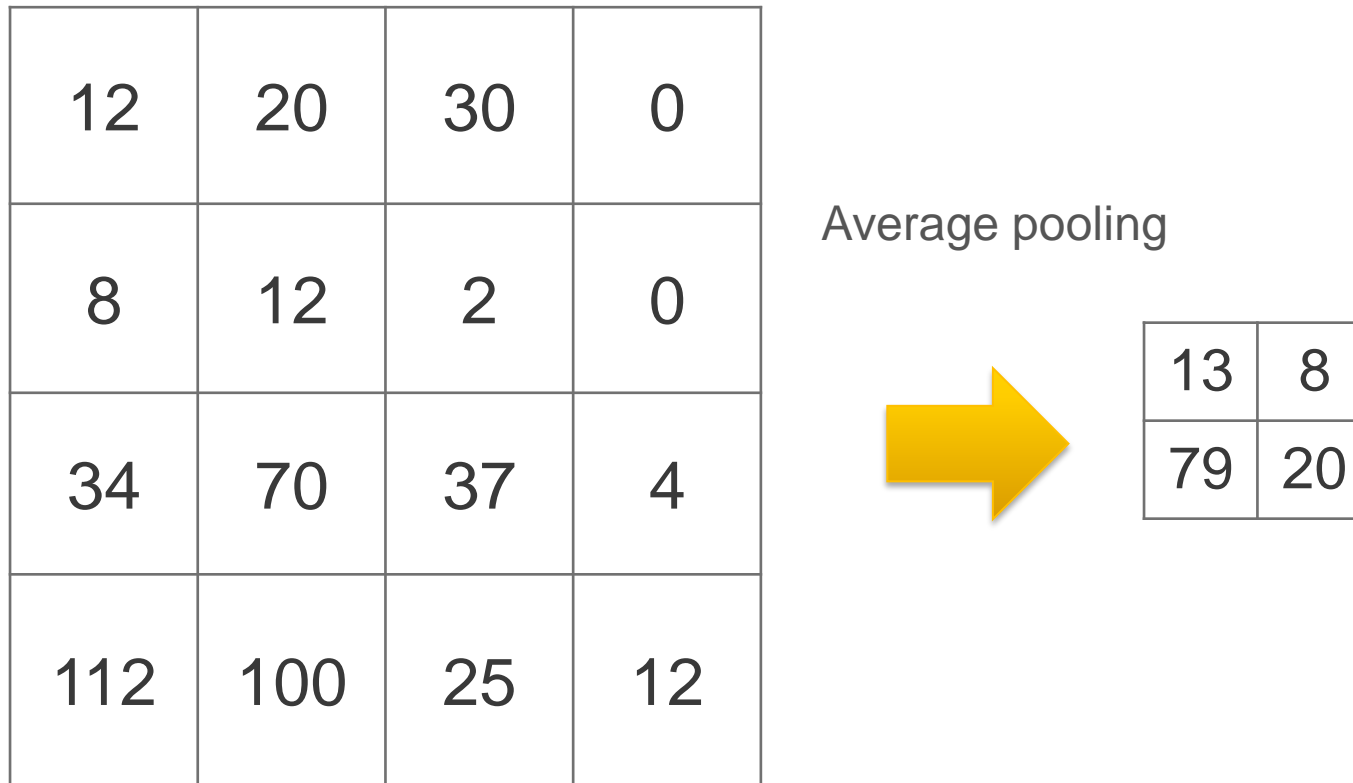
Maxpooling 運算

- 一個網路如果用太多Maxpooling，雖然有加速效果，但是準確度可以會往下掉，所以建議適當使用Maxpooling層就好



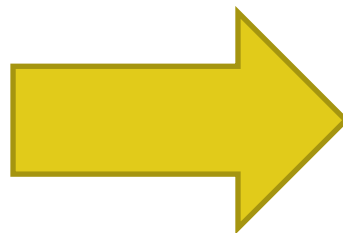
Maxpooling 運算

- Maxpooling 其實是這幾年大家習慣的做法，其實過去也有不少人喜歡用 Averagepooling



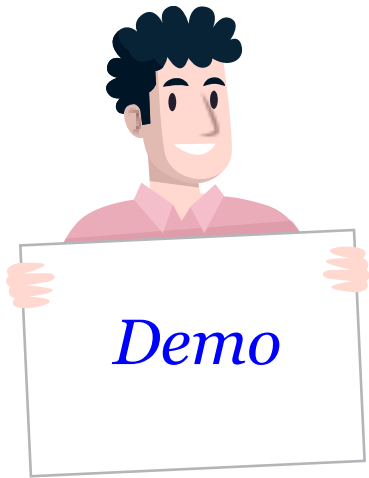
Maxpooling 物理意義

- Maxpooling的意義就是將原始圖片做下採樣
 - 圖片或特徵會變更模糊



Demo 10-3

- Convolution用法
- Maxpooling用法
- 建構CNN神經網路



designed by freepik

線上Corelab

- **題目1：Convolution的使用**
 - 請針對範例圖片用以下的矩陣做Convolution運算並輸出結果
- **題目2：Maxpooling的使用**
 - 請針對範例圖片做"4"次Maxpooling運算並輸出結果
- **題目3：建構與訓練CNN網路**
 - 請用以下的4個子程式建構出一個完整的CNN網路以MNIST資料集訓練，並輸出準確率

本章重點精華回顧

- **DNN網路的缺點**
- **圖片的組成**
- **Convolution運算及物理意義**
- **Maxpooling運算及物理意義**



Lab:Python 簡介

- **Lab01: Convolution用法**
- **Lab02: Maxpooling用法**
- **Lab03: 建構CNN神經網路**

Estimated time:

20 minutes

