

Day 36

深度學習與電腦視覺 學習馬拉松

cupay 陪跑專家：陳穗碧

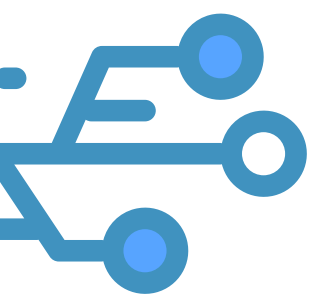


YOLO 細節理解 – 網路架構

重要知識點



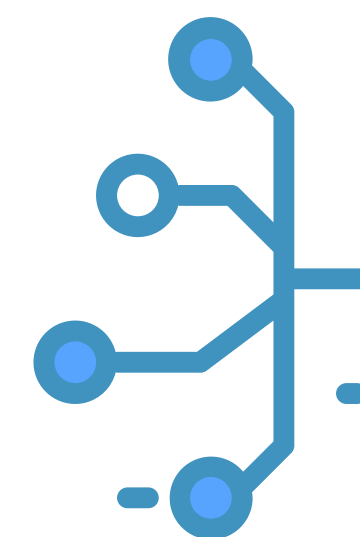
- 了解 YOLO 網絡架構的設計與原理

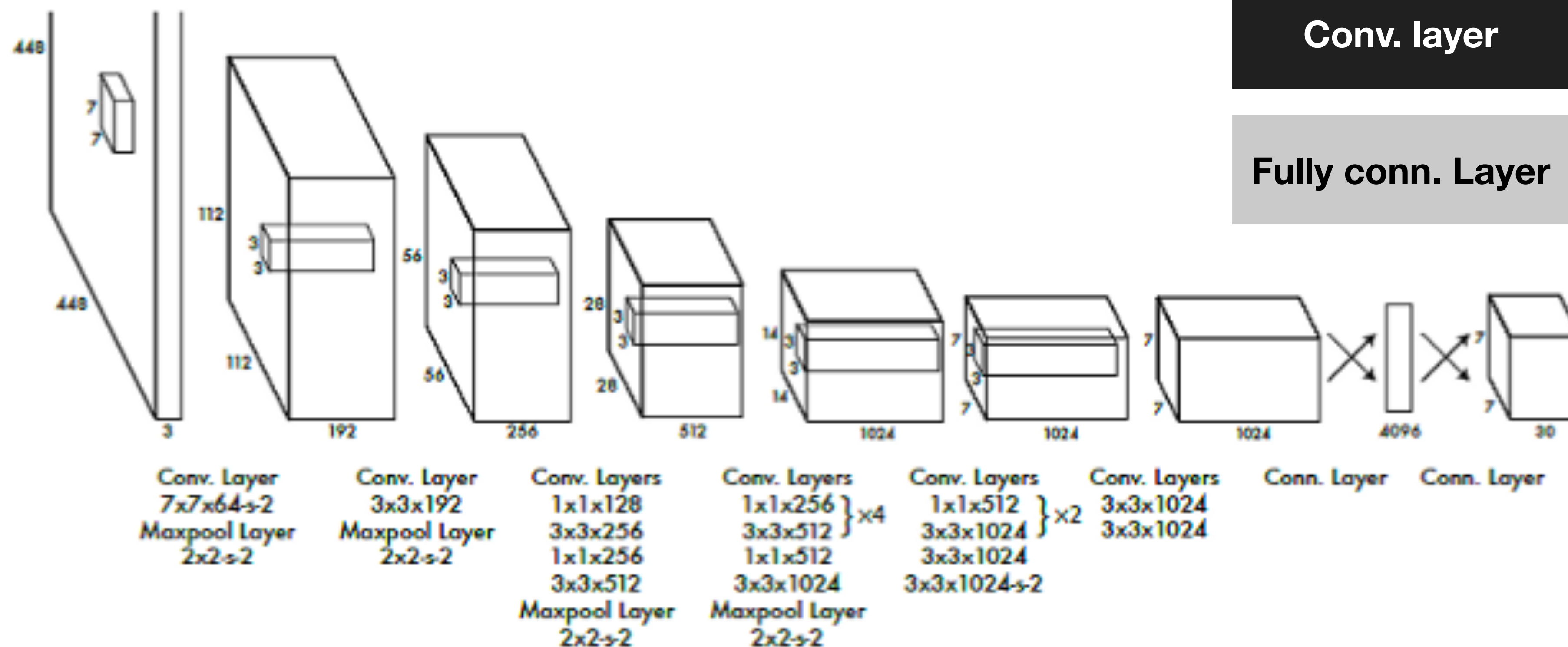


進入 YOLO 網絡架構的世界

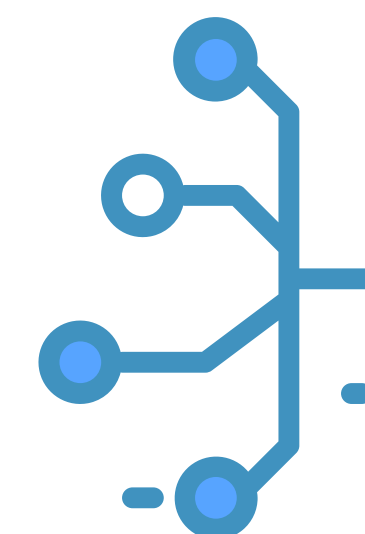


昨天我們介紹了 YOLO 架構的輸出設計，今天我們進入 YOLO 網絡架構的世界，了解每一層設計的目的。



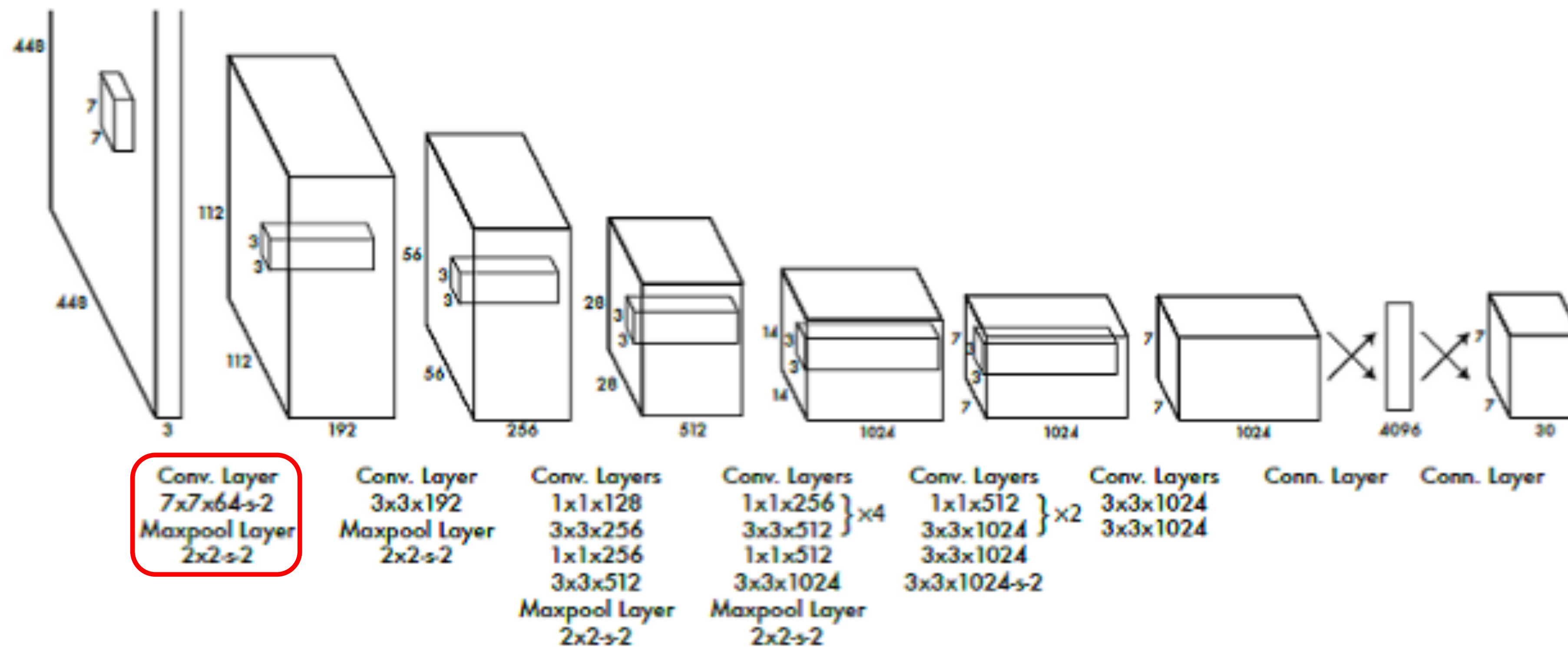


YOLO 的網絡架構，包含 24 層的卷積層(convolutional layer)和兩層的全連接層(fully connected layer)，24 層的卷積層的目的用於抽象圖像特徵擷取，兩個全連接層用於產生出對應的分類和定位的結果。

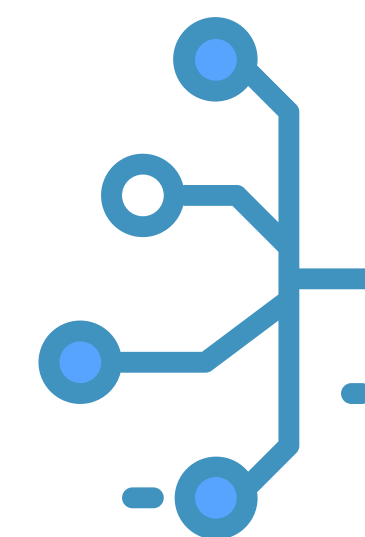




YOLO 網絡架構 - 細部拆解



第一層有 64 個 7*7 大小的 Filters，Stride=2。
搭載一個 池化層，在 2*2 的大小中取最大值當代表值，Stride=2。

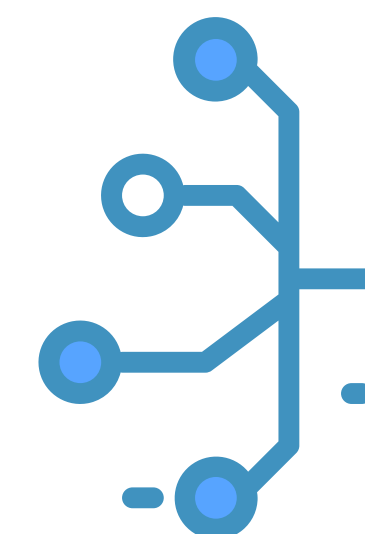




Filter 和 Output Dimension 的關係

Name	Filters	Output Dimension
Conv 1	7 x 7 x 64, stride=2	224 x 224 x 64
Max Pool 1	2 x 2, stride=2	112 x 112 x 64
Conv 2	3 x 3 x 192	112 x 112 x 192
Max Pool 2	2 x 2, stride=2	56 x 56 x 192
Conv 3	1 x 1 x 128	56 x 56 x 128
Conv 4	3 x 3 x 256	56 x 56 x 256
Conv 5	1 x 1 x 256	56 x 56 x 256
Conv 6	1 x 1 x 512	56 x 56 x 512
Max Pool 3	2 x 2, stride=2	28 x 28 x 512
Conv 7	1 x 1 x 256	28 x 28 x 256
Conv 8	3 x 3 x 512	28 x 28 x 512
Conv 9	1 x 1 x 256	28 x 28 x 256
Conv 10	3 x 3 x 512	28 x 28 x 512
Conv 11	1 x 1 x 256	28 x 28 x 256
Conv 12	3 x 3 x 512	28 x 28 x 512
Conv 13	1 x 1 x 256	28 x 28 x 256
Conv 14	3 x 3 x 512	28 x 28 x 512
Conv 15	1 x 1 x 512	28 x 28 x 512
Conv 16	3 x 3 x 1024	28 x 28 x 1024
Max Pool 4	2 x 2, stride=2	14 x 14 x 1024
Conv 17	1 x 1 x 512	14 x 14 x 512
Conv 18	3 x 3 x 1024	14 x 14 x 1024
Conv 19	1 x 1 x 512	14 x 14 x 512
Conv 20	3 x 3 x 1024	14 x 14 x 1024
Conv 21	3 x 3 x 1024	14 x 14 x 1024
Conv 22	3 x 3 x 1024, stride=2	7 x 7 x 1024
Conv 23	3 x 3 x 1024	7 x 7 x 1024
Conv 24	3 x 3 x 1024	7 x 7 x 1024
FC 1	-	4096
FC 2	-	7 x 7 x 30 (1470)

- 圖片大小 (W) : 448 x 448
- Filter大小 (F) : 7 x 7 共 64 個
- 步長 (S) : 2
- padding='same' , 會用 zero-padding 的手法, 讓輸入的圖不會受到 kernel map 的大小影響
- 第一層卷基層輸出的大小為
$$\frac{W}{S} = 224$$
- 輸出維度 = 224 x 224 x 64
- 觀察一下藍色的數字, 64, 發現 Filters 的個數等於輸出維度。



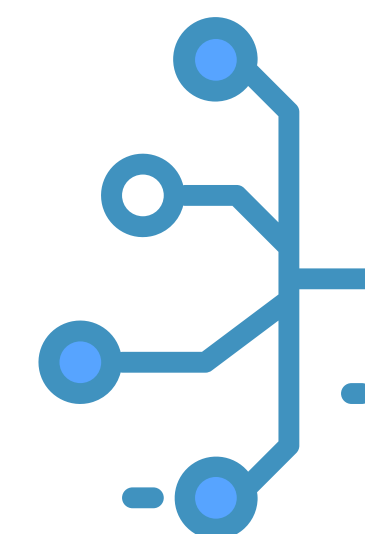


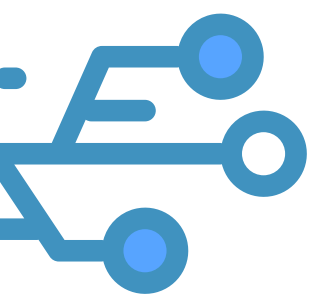
Filter 和 Output Dimension 的關係

Name	Filters	Output Dimension
Conv 1	7 x 7 x 64, stride=2	224 x 224 x 64
Max Pool 1	2 x 2, stride=2	112 x 112 x 64
Conv 2	3 x 3 x 192	112 x 112 x 192
Max Pool 2	2 x 2, stride=2	56 x 56 x 192
Conv 3	1 x 1 x 128	56 x 56 x 128
Conv 4	3 x 3 x 256	56 x 56 x 256
Conv 5	1 x 1 x 256	56 x 56 x 256
Conv 6	1 x 1 x 512	56 x 56 x 512
Max Pool 3	2 x 2, stride=2	28 x 28 x 512
Conv 7	1 x 1 x 256	28 x 28 x 256
Conv 8	3 x 3 x 512	28 x 28 x 512
Conv 9	1 x 1 x 256	28 x 28 x 256
Conv 10	3 x 3 x 512	28 x 28 x 512
Conv 11	1 x 1 x 256	28 x 28 x 256
Conv 12	3 x 3 x 512	28 x 28 x 512
Conv 13	1 x 1 x 256	28 x 28 x 256
Conv 14	3 x 3 x 512	28 x 28 x 512
Conv 15	1 x 1 x 512	28 x 28 x 512
Conv 16	3 x 3 x 1024	28 x 28 x 1024
Max Pool 4	2 x 2, stride=2	14 x 14 x 1024
Conv 17	1 x 1 x 512	14 x 14 x 512
Conv 18	3 x 3 x 1024	14 x 14 x 1024
Conv 19	1 x 1 x 512	14 x 14 x 512
Conv 20	3 x 3 x 1024	14 x 14 x 1024
Conv 21	3 x 3 x 1024	14 x 14 x 1024
Conv 22	3 x 3 x 1024, stride=2	7 x 7 x 1024
Conv 23	3 x 3 x 1024	7 x 7 x 1024
Conv 24	3 x 3 x 1024	7 x 7 x 1024
FC 1	-	4096
FC 2	-	7 x 7 x 30 (1470)

Layer 1

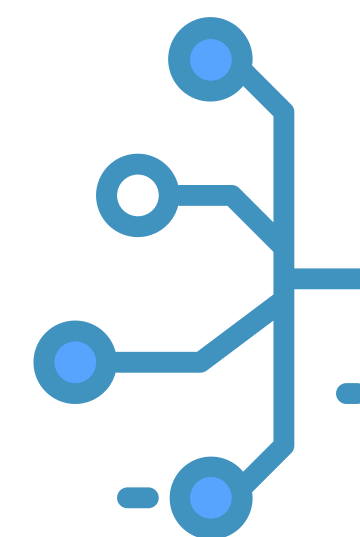
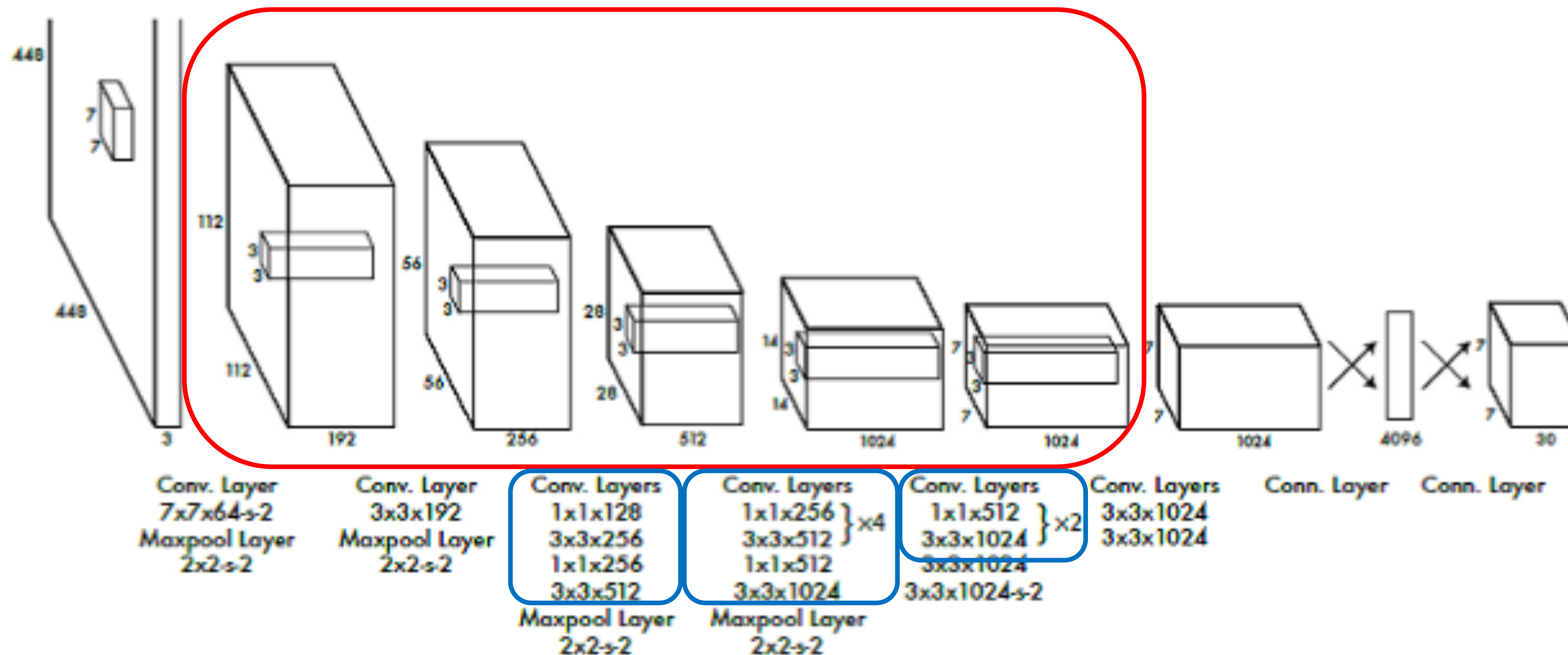
- Max pooling : 在 2*2 的大小中取最大值當代表值，
- Stride = 2
- Layer 1 輸出維度 = $\frac{224}{2} \times \frac{224}{2} \times 64$





YOLO 網絡架構 - 以 GoogLeNet 為核心

YOLO 的網絡架構，是以 **GoogLeNet** 模型為基礎發展，YOLO 借鑑 **Inception Module** 架構，在某些 3×3 的卷積層前面用 1×1 的卷積層，那麼這樣的架構有什麼好處？

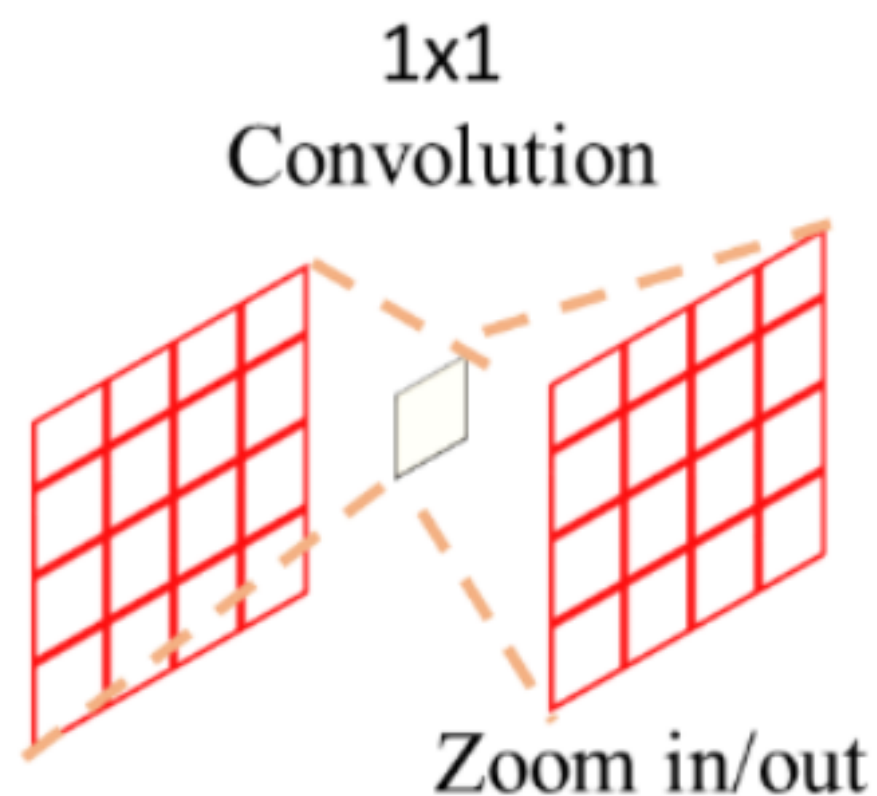




1x1 卷積的用處

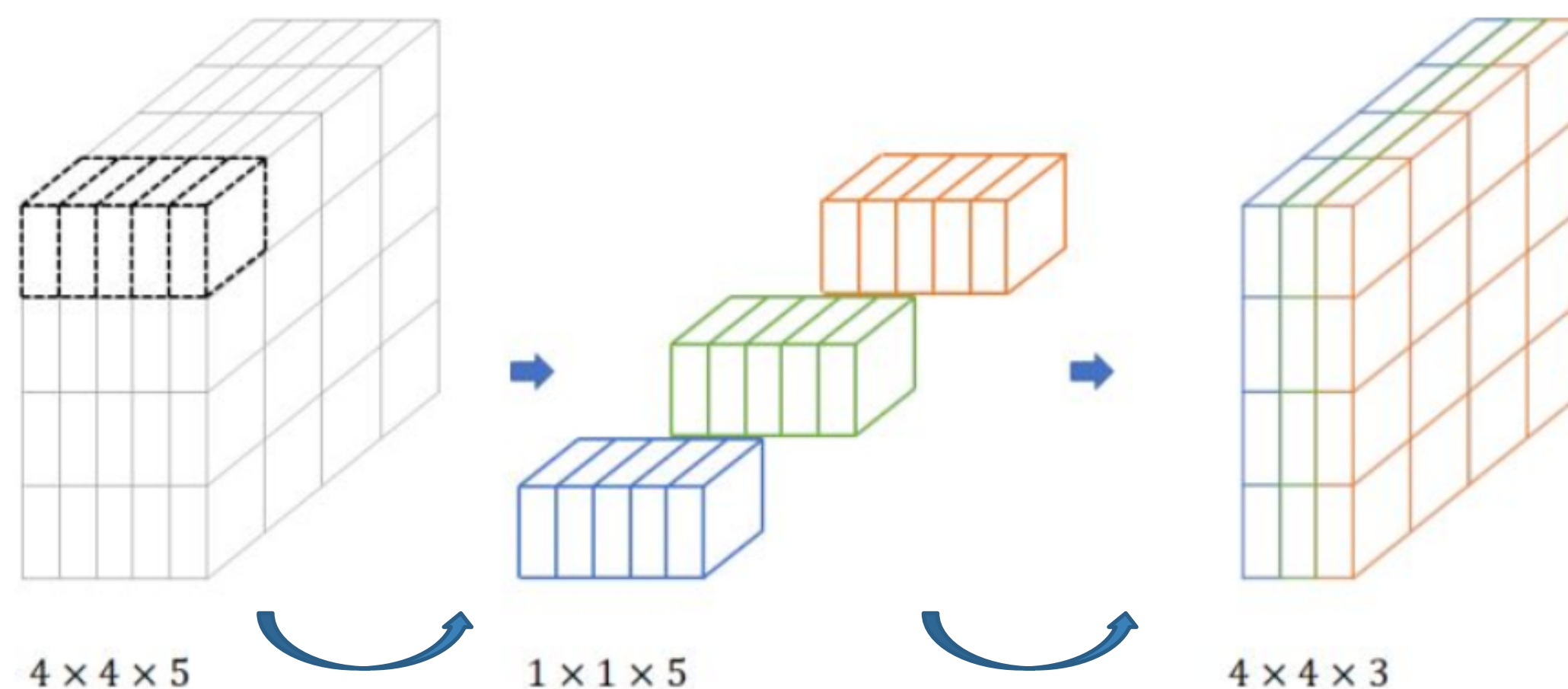
直觀

把原本的數值放大或縮小



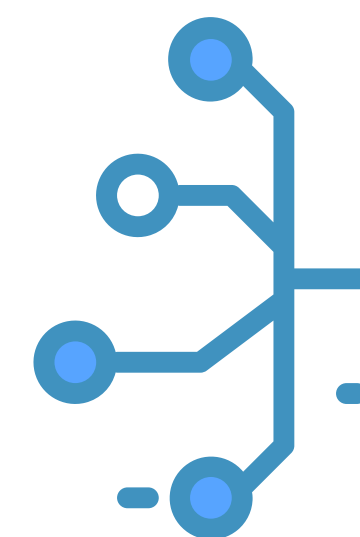
微觀:

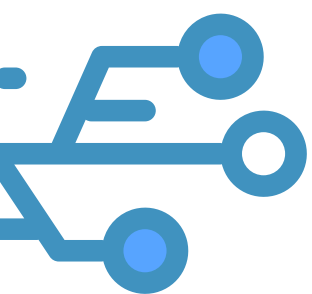
1. 跨通道(channel)信息整合
2. 空間關係不變下，達到降維或升維的效果



對應相乘的動作，就是把相同位置下的 channel 特徵融合再一起

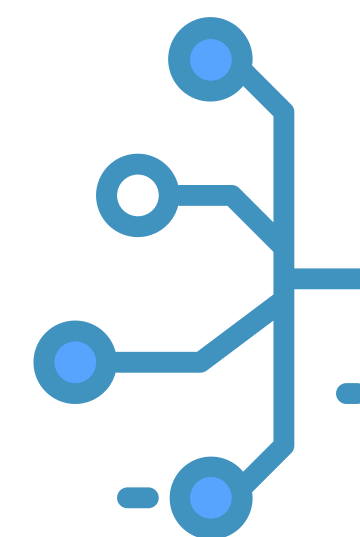
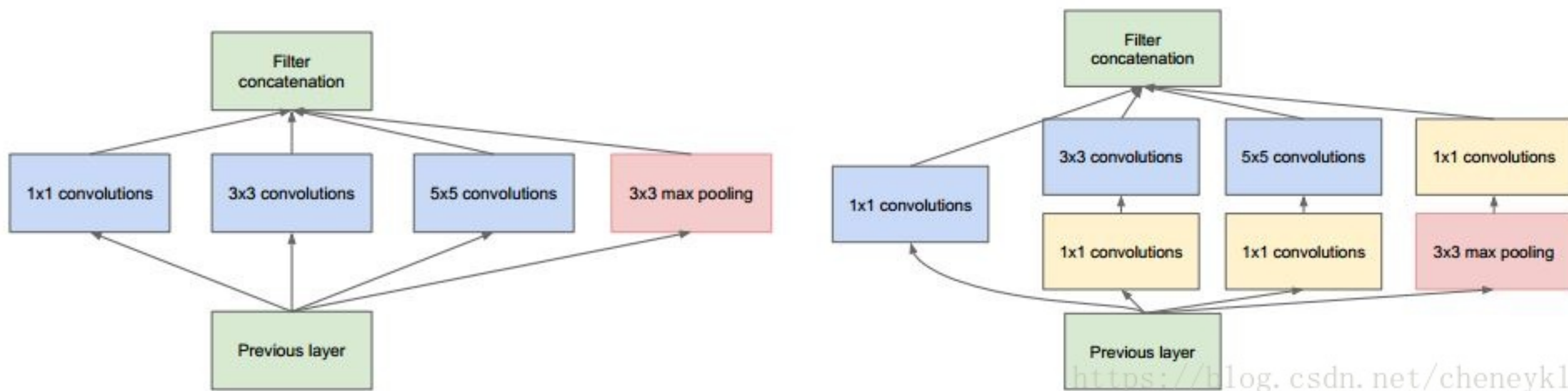
空間關係不變下，透過 Filters 個數，來達到降維或升維的效果，以這個圖為例從 5 channel→3 channel

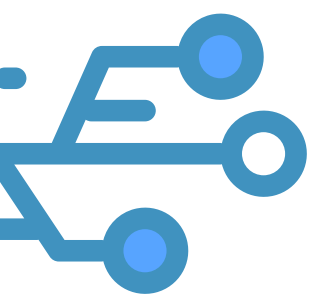




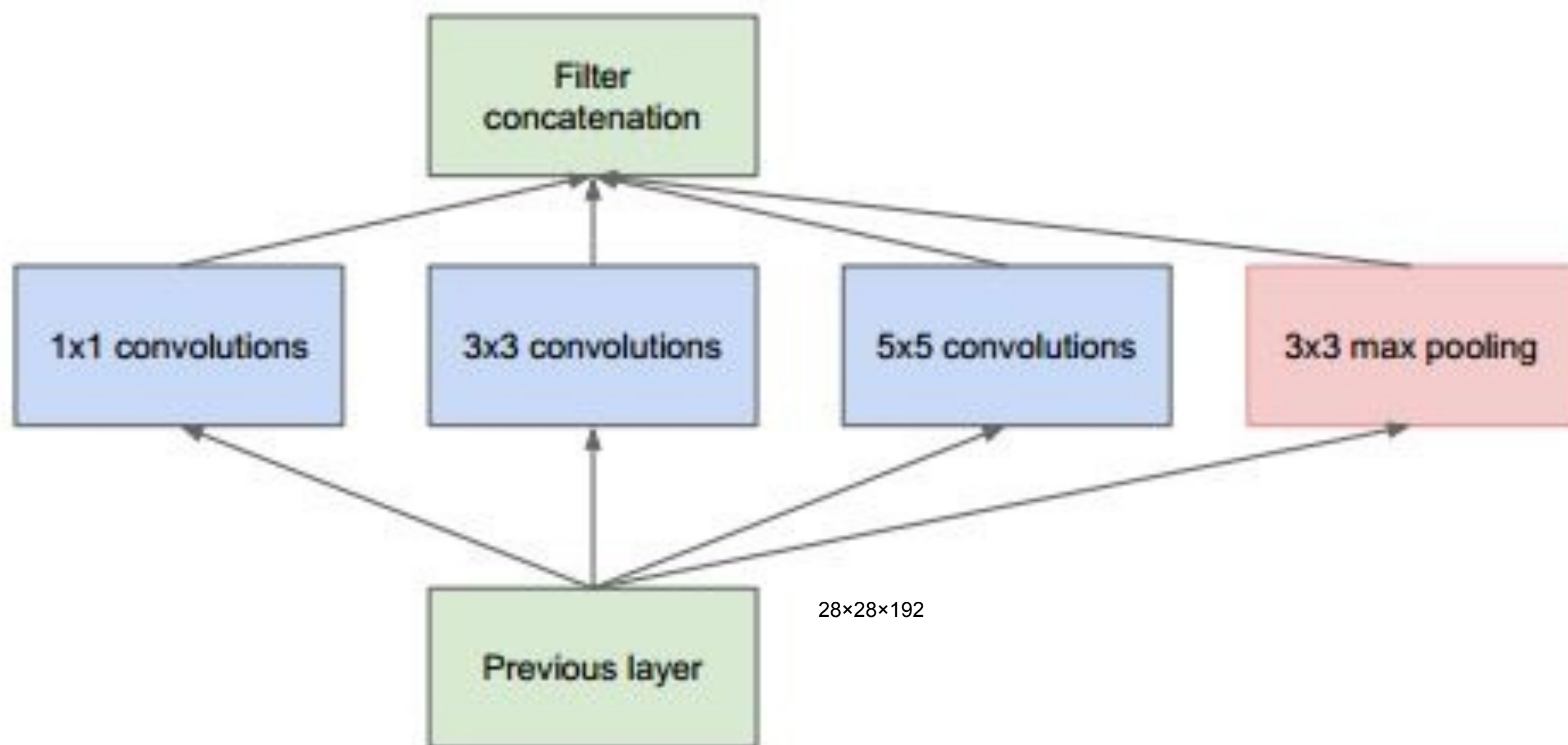
1×1 卷積的效益-降維達到降低參數量(1/3)

- 這邊我們以 GoogleNet 中的某一層架構來看，下方左邊的圖沒有搭載 1×1 卷積，右邊的突有搭載 1×1 卷積，比較兩者的參數，看 1×1 卷積如何達到降低參數量的過程。
- 首先，兩邊都輸入相同大小的 feature map 是 28×28×192





1×1 卷積的效益-降維達到降低參數量(2/3)



假設網絡架構中:

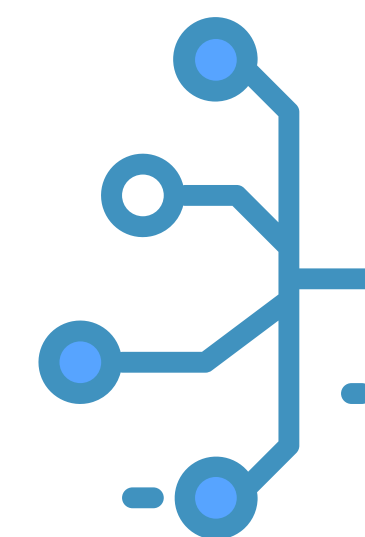
1×1 卷積Filter數量為64

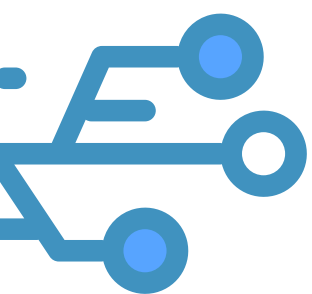
3×3 卷積Filter數量為128

5×5 卷積Filter數量為32

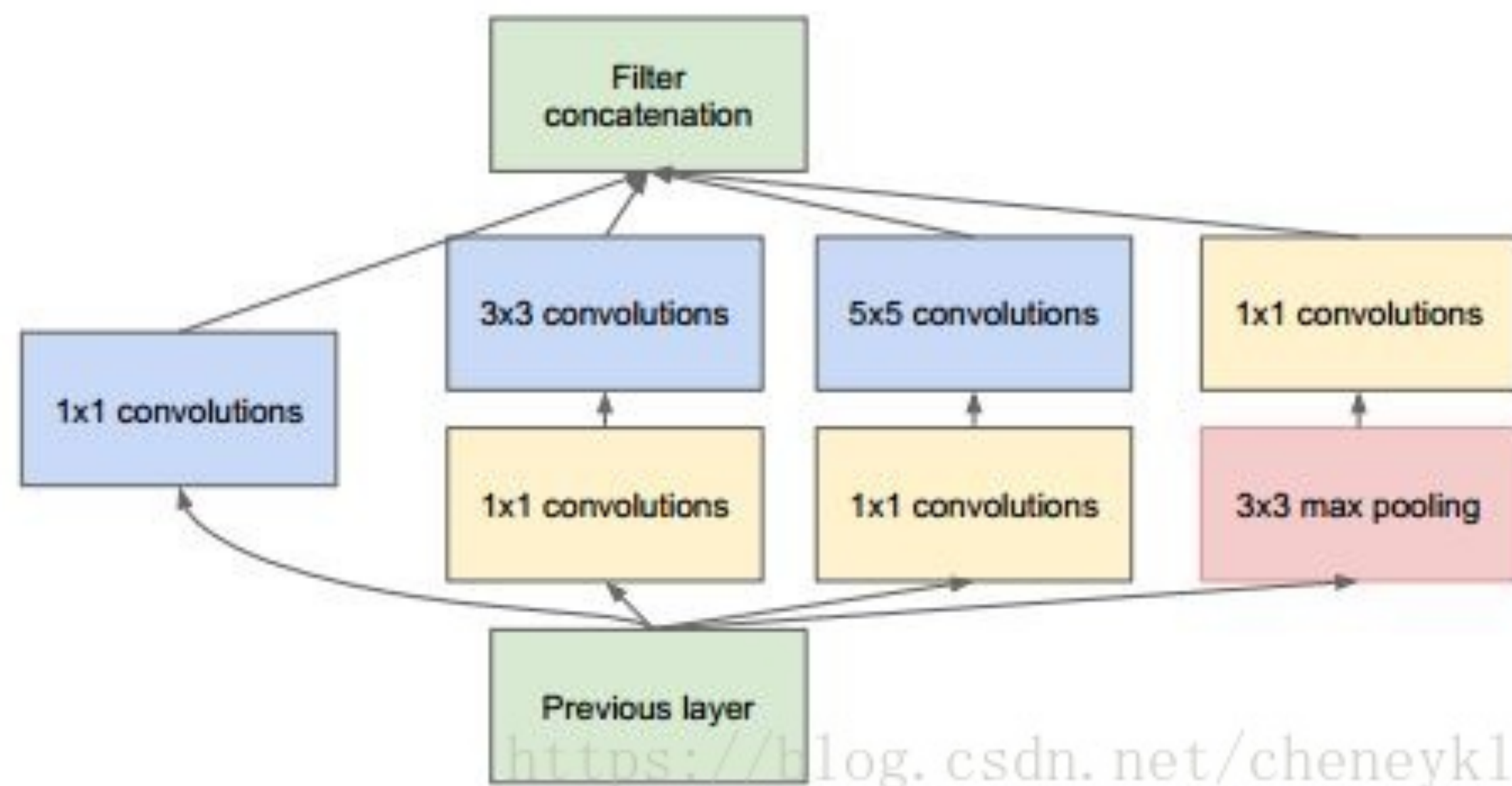
這張圖這一層的參數的數量為:

$$192 \times (1 \times 1 \times 64) + 192 \times (3 \times 3 \times 128) + 192 \times (5 \times 5 \times 32) = 387072$$





1×1 卷積的效益-降維達到降低參數量(3/3)



387,072→163,328

藍色的數字，就是降維，最終達到降低參數量的效果

假設網絡架構中：

1×1卷積 Filters 數量為64

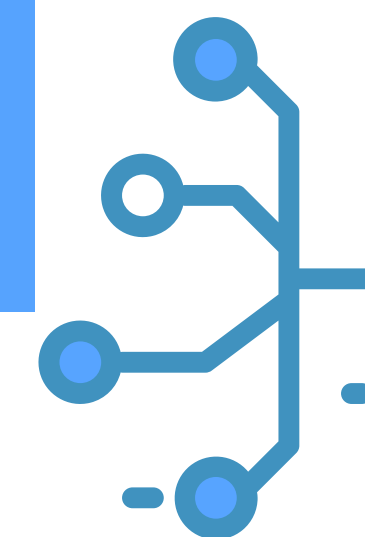
3×3卷積 Filters數量為128

5×5卷積 Filters 數量為32

對3×3、5×5卷積與3×3 池化層前(/後)分別加入了數量為96、16和32的1×1卷積Filters

這張圖這一層的參數的數量為：

$$192 \times (1 \times 1 \times 64) + (192 \times 1 \times 1 \times 96 + 96 \times 3 \times 3 \times 128) + (192 \times 1 \times 1 \times 16 + 16 \times 5 \times 5 \times 32) + (192 \times 1 \times 1 \times 32) = 163,328$$





1×1 與 3×3 卷積層組合的好處

Name	Filters	Output Dimension
Conv 1	7 x 7 x 64, stride=2	224 x 224 x 64
Max Pool 1	2 x 2, stride=2	112 x 112 x 64
Conv 2	3 x 3 x 192	112 x 112 x 192
Max Pool 2	2 x 2, stride=2	56 x 56 x 192
Conv 3	1 x 1 x 128	56 x 56 x 128
Conv 4	3 x 3 x 256	56 x 56 x 256
Conv 5	1 x 1 x 256	56 x 56 x 256
Conv 6	1 x 1 x 512	56 x 56 x 512
Max Pool 3	2 x 2, stride=2	28 x 28 x 512
Conv 7	1 x 1 x 256	28 x 28 x 256
Conv 8	3 x 3 x 512	28 x 28 x 512
Conv 9	1 x 1 x 256	28 x 28 x 256
Conv 10	3 x 3 x 512	28 x 28 x 512
Conv 11	1 x 1 x 256	28 x 28 x 256
Conv 12	3 x 3 x 512	28 x 28 x 512
Conv 13	1 x 1 x 256	28 x 28 x 256
Conv 14	3 x 3 x 512	28 x 28 x 512
Conv 15	1 x 1 x 512	28 x 28 x 512
Conv 16	3 x 3 x 1024	28 x 28 x 1024
Max Pool 4	2 x 2, stride=2	14 x 14 x 1024
Conv 17	1 x 1 x 512	14 x 14 x 512
Conv 18	3 x 3 x 1024	14 x 14 x 1024
Conv 19	1 x 1 x 512	14 x 14 x 512
Conv 20	3 x 3 x 1024	14 x 14 x 1024
Conv 21	3 x 3 x 1024	14 x 14 x 1024
Conv 22	3 x 3 x 1024, stride=2	7 x 7 x 1024
Conv 23	3 x 3 x 1024	7 x 7 x 1024
Conv 24	3 x 3 x 1024	7 x 7 x 1024
FC 1	-	4096
FC 2	-	7 x 7 x 30 (1470)

1

- 通過跨通道 (channel) 信息整合後，再取 3×3 的小範圍下找相似的特徵

3

- 在 YOLOv1 中共有 9 個這樣的組合

4

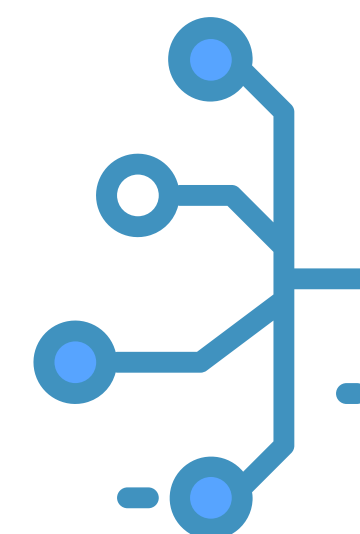
5

6

7

8

9

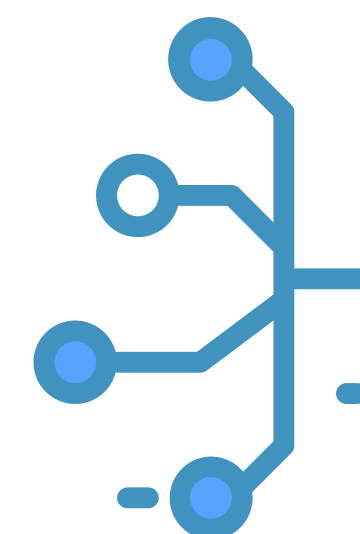




1×1 與 3×3 卷積層組合的好處

Name	Filters	Output Dimension
Conv 1	7 x 7 x 64, stride=2	224 x 224 x 64
Max Pool 1	2 x 2, stride=2	112 x 112 x 64
Conv 2	3 x 3 x 192	112 x 112 x 192
Max Pool 2	2 x 2, stride=2	56 x 56 x 192
Conv 3	1 x 1 x 128	56 x 56 x 128
Conv 4	3 x 3 x 256	56 x 56 x 256
Conv 5	1 x 1 x 256	56 x 56 x 256
Conv 6	1 x 1 x 512	56 x 56 x 512
Max Pool 3	2 x 2, stride=2	28 x 28 x 512
Conv 7	1 x 1 x 256	28 x 28 x 256
Conv 8	3 x 3 x 512	28 x 28 x 512
Conv 9	1 x 1 x 256	28 x 28 x 256
Conv 10	3 x 3 x 512	28 x 28 x 512
Conv 11	1 x 1 x 256	28 x 28 x 256
Conv 12	3 x 3 x 512	28 x 28 x 512
Conv 13	1 x 1 x 256	28 x 28 x 256
Conv 14	3 x 3 x 512	28 x 28 x 512
Conv 15	1 x 1 x 512	28 x 28 x 512
Conv 16	3 x 3 x 1024	28 x 28 x 1024
Max Pool 4	2 x 2, stride=2	14 x 14 x 1024
Conv 17	1 x 1 x 512	14 x 14 x 512
Conv 18	3 x 3 x 1024	14 x 14 x 1024
Conv 19	1 x 1 x 512	14 x 14 x 512
Conv 20	3 x 3 x 1024	14 x 14 x 1024
Conv 21	3 x 3 x 1024	14 x 14 x 1024
Conv 22	3 x 3 x 1024, stride=2	7 x 7 x 1024
Conv 23	3 x 3 x 1024	7 x 7 x 1024
Conv 24	3 x 3 x 1024	7 x 7 x 1024
FC 1	-	4096
FC 2	-	7 x 7 x 30 (1470)

- 拉平後
- 兩個全連接層用於產生出對應的分類和定位的結果。



知識點 回顧

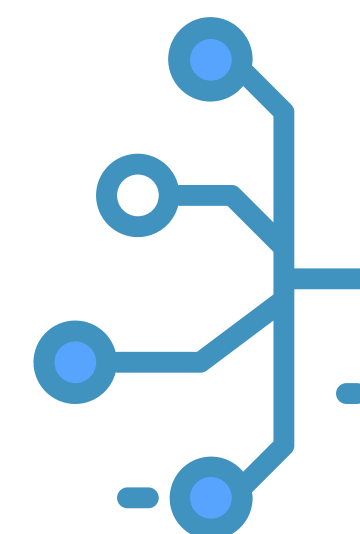
- YOLO 的網絡架構，是以 GoogLeNet 模型為骨架進行調整，透過 1×1 卷積層，進行跨通道(channel)信息整合，同時達到降低參數的效果，降低過度擬合的情形。
- 了解原理後，明天將進行 YOLO 的網絡架構程式碼講解，明天見



參考資料



- 今天的課程還意猶未盡嗎？想更深入從不同的角度探討 1×1 的實際效益，可以看參考資料中的網路討論，可以學習到更多。
- 卷積神經網絡中用 1×1 卷積有甚麼作用或好處？



學習卡關了 專家來幫助你

...

本課程提供線上專家問答服務，你可以在問答平台中觀看其他學員的提問，也可以主動詢問專家，專家跟學習助理將盡快回覆你的問題。

點此問問題



解題時間 Let's Crack It



請跳出 PDF 至官網 Sample Code & 作業開始解題