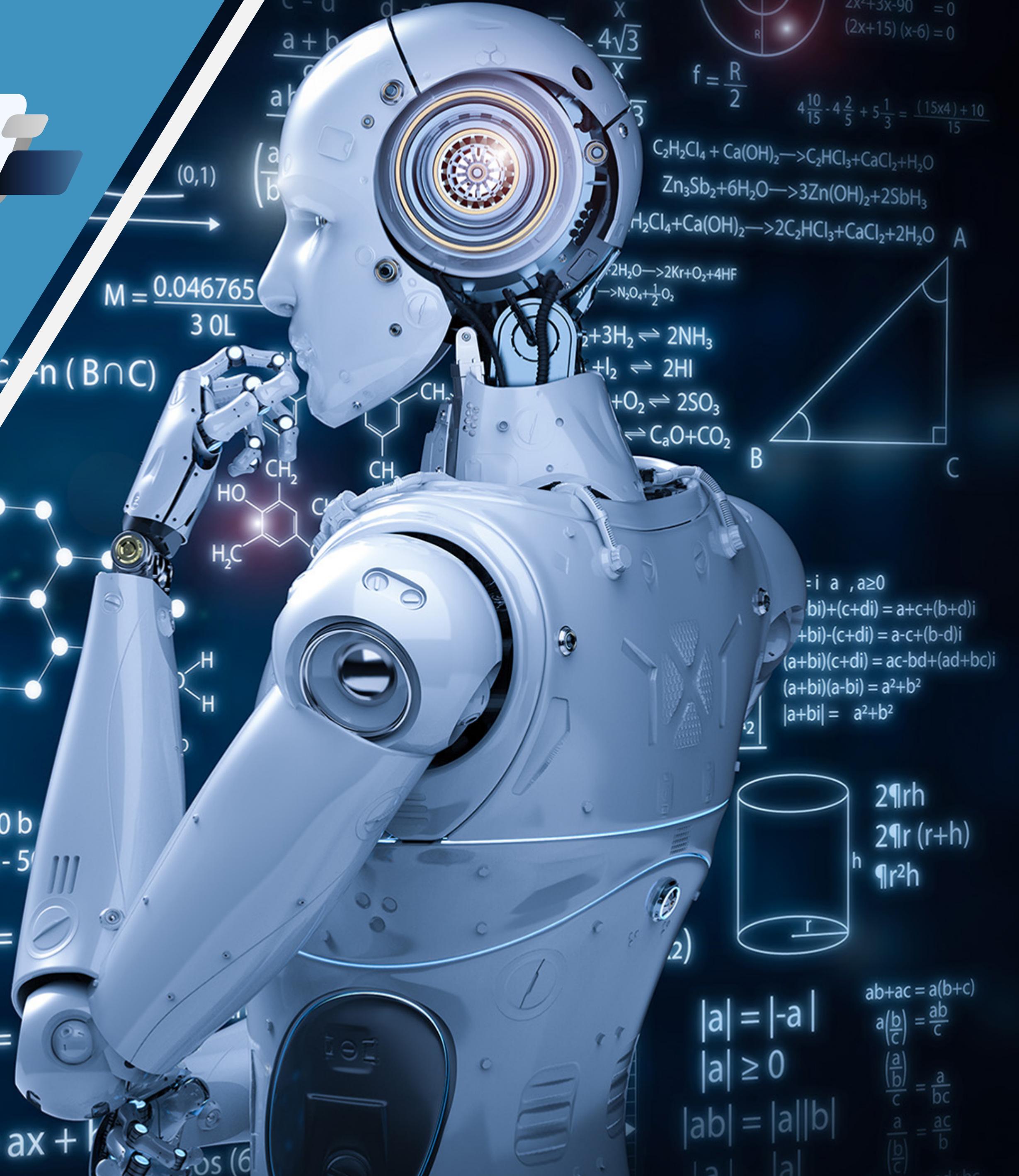


# Day 26

## 深度學習與電腦視覺 學習馬拉松

cupay 陪跑專家：楊哲寧





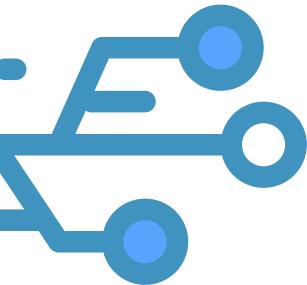
# 深度學習理論與實作

## Object Detection-RPN結構



# 重要知識點

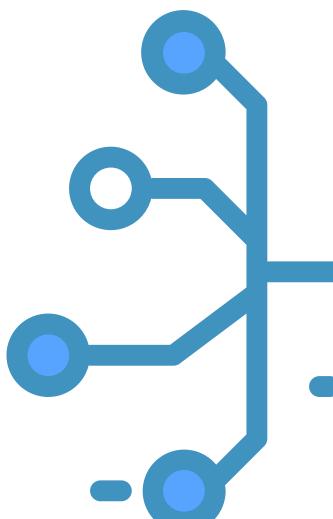
了解Faster R-CNN 中的Region proposal network (RPN)結構

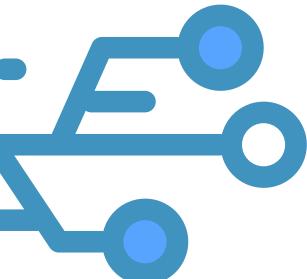


# Region Proposal Network 目的



- RPN 首次出現於 Faster R-CNN 中，用來取代過去的 Selective Search 提出 Region Proposal。
- RPN 的目標是先提出可能包含物體的 proposals，因此一個訓練好的 RPN 應該要有很高的 **Recall**。
- 相較於 Selective Search 是一個獨立的作業，RPN的優點在於可以放入模型中一起學習，因此可以打造一個 End-to-End 的模型。



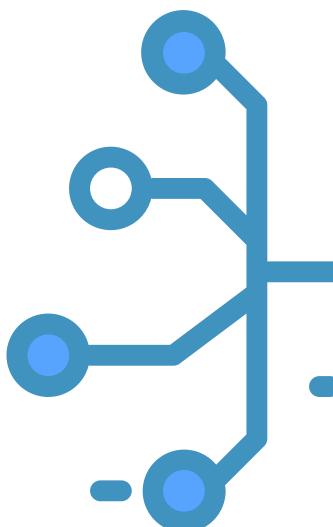
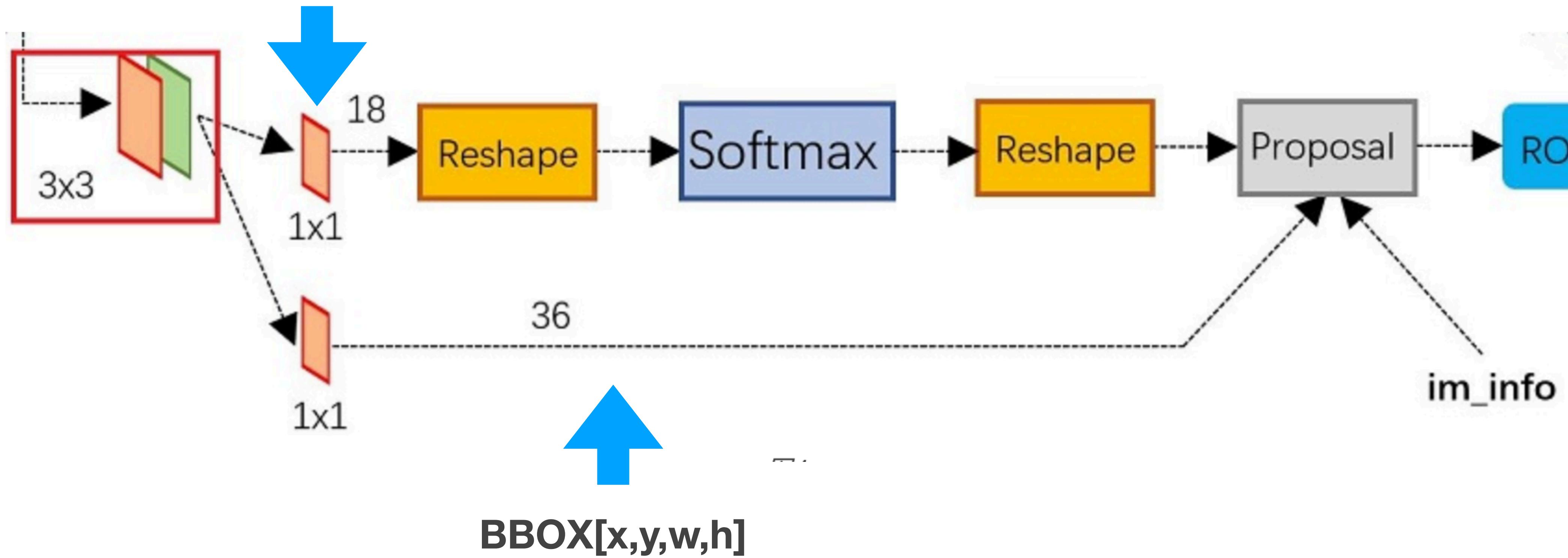


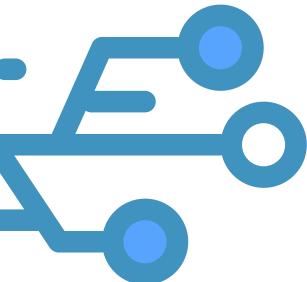
# RPN - 輸入與輸出



下圖為 RPN 的基本結構，RPN 主要由兩條支線組成，一條用來預測BBOX 的  $[x,y,w,h]$  另一條則是用來預測前景或背景。

前景或背景





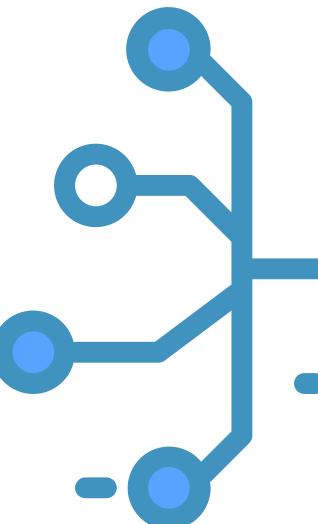
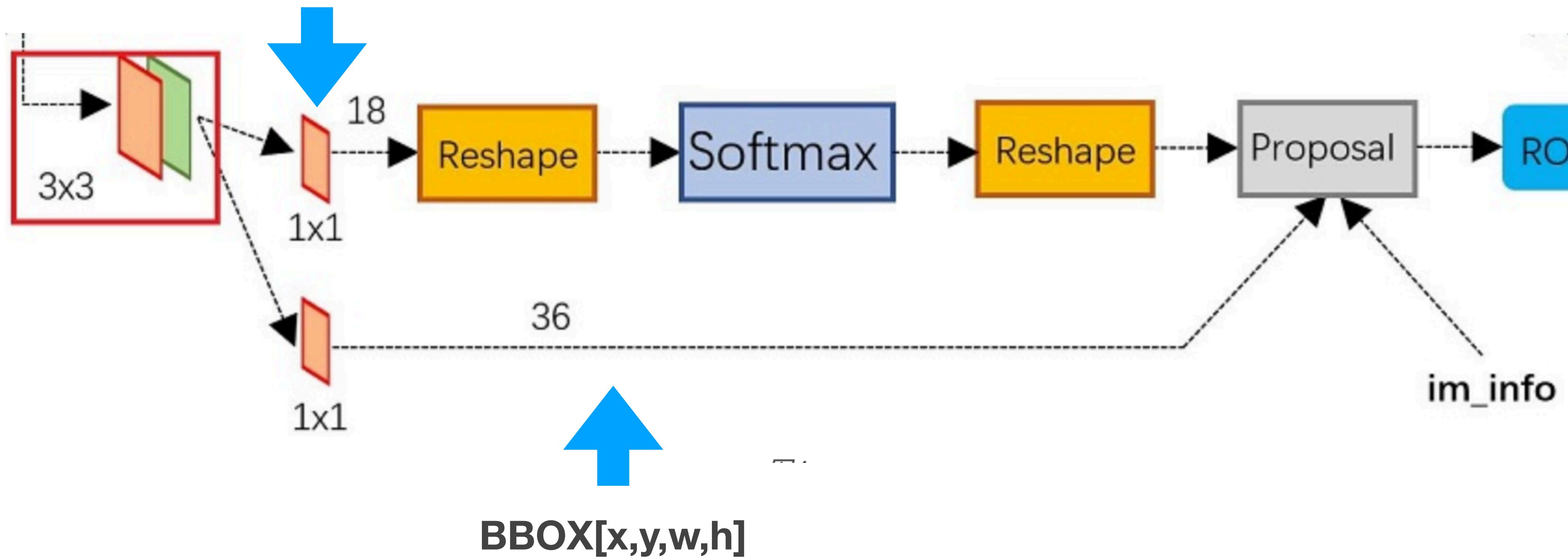
# RPN - 輸入與輸出

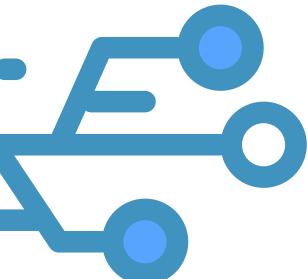


CUPOY

其輸入為經過 CNN Backbone 所獲得的特徵圖，以及預設生成的 anchors (可以想像成一堆預先設定的 Bounding Box)，輸出為 Bounding Box的資訊以及前景、背景的信心程度。

前景或背景

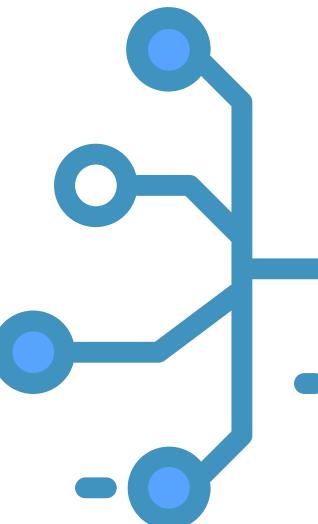
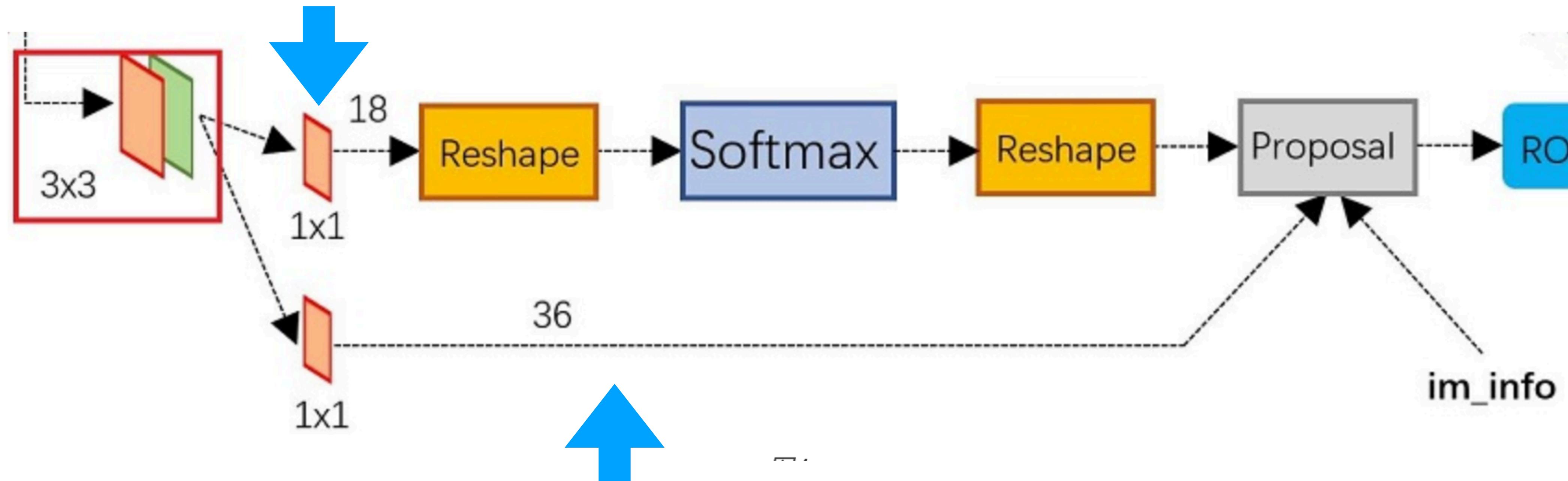


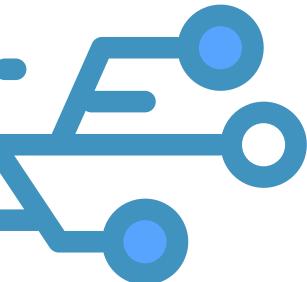


# RPN 架構



而圖中 前景或背景 支線的 18 與 BBOX 的支線的 36 分別帶表其深度





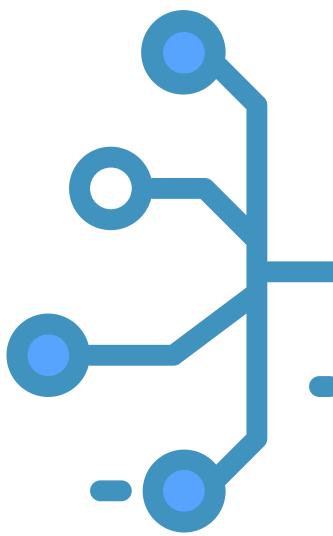
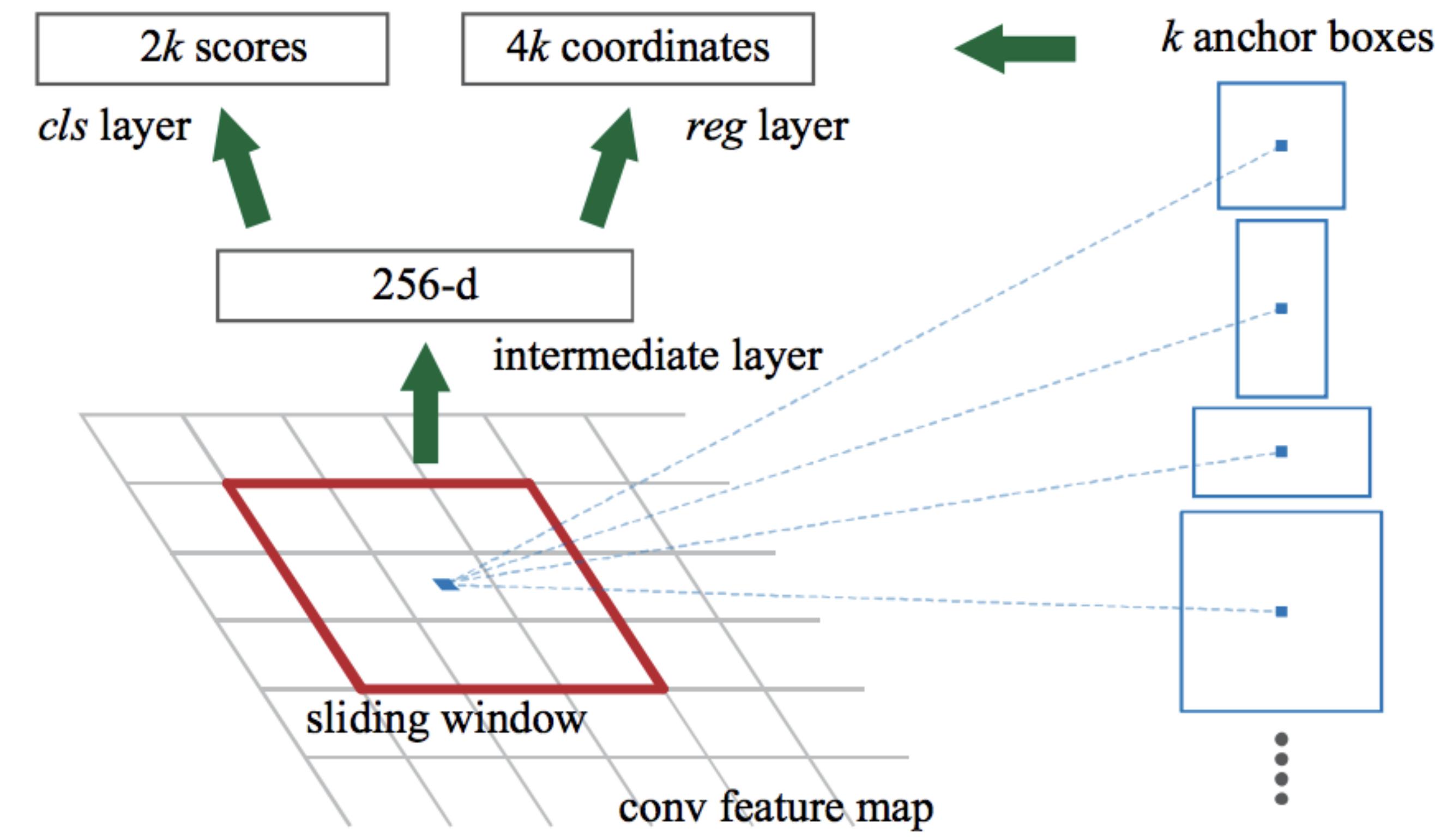
# RPN 架構

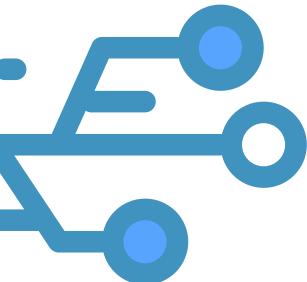


原文中作者在每個像素上設置了9個 default anchor boxes，每個 Box 要預測  $[x,y,w,h]$  與 [前景,背景]，因此 BBOX 支線就有  $9 \times 4 = 36$  的深度，另一條則為  $9 \times 2 = 18$  的深度。

$[x,y,w,h]$

[前景、背景]



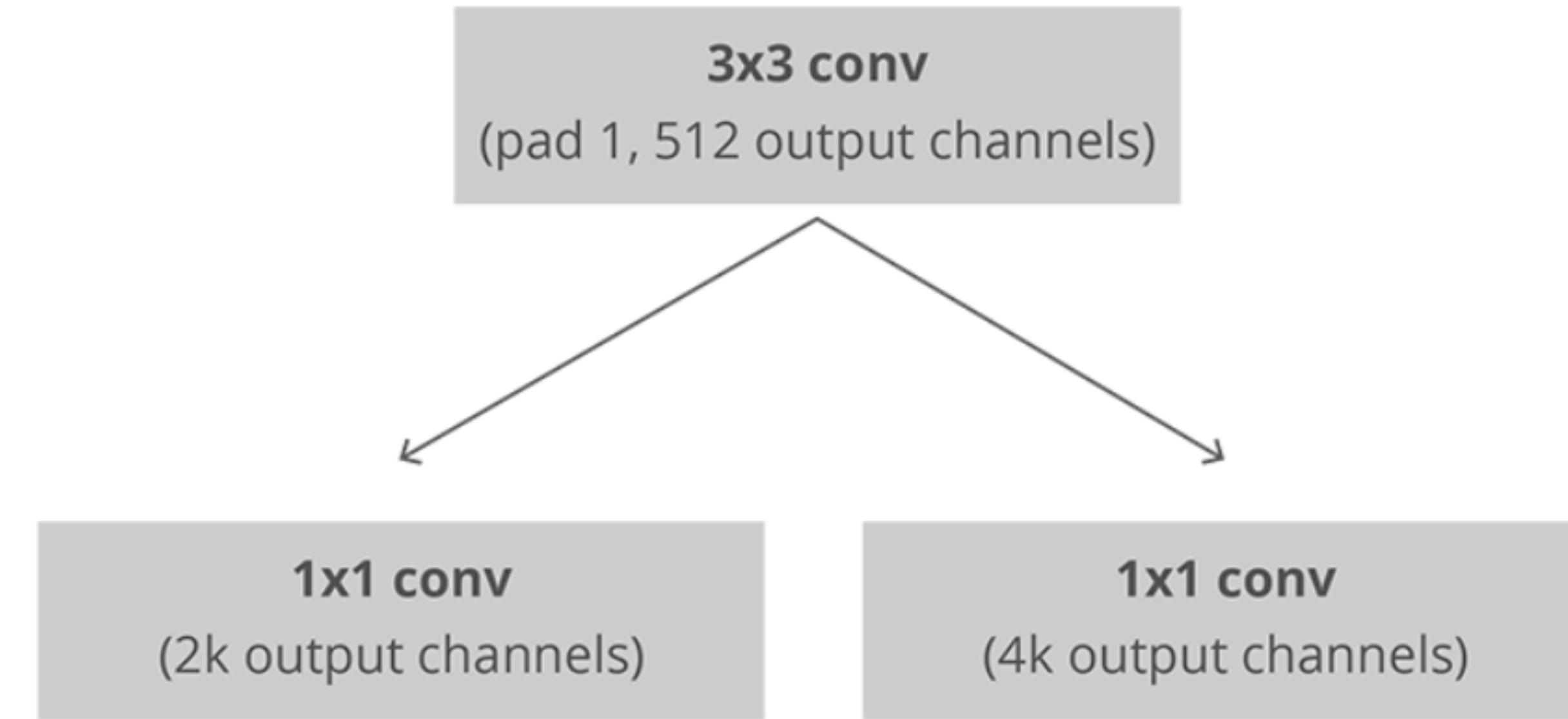


# 如何設計Default Anchor Box



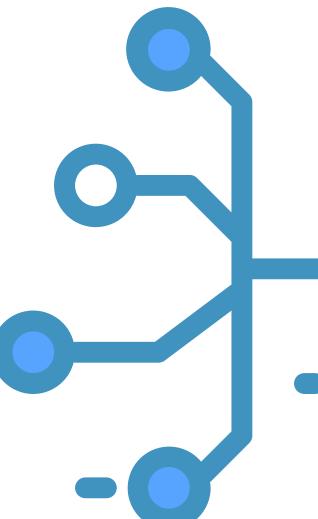
首先模型這邊要先設計好對應的值(Prediction)

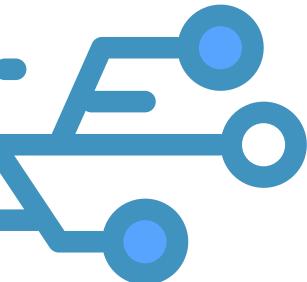
- 透過 3\*3 kernel，Padding 為 1，以 Sliding window 的方式掃過整張圖，再用 1\*1 kernel 壓到我們要的深度(18,36)。
- Output 值要與 Default Anchor Box 相對應**，Faster R-CNN 原文在每一個 Pixel 上設計9個 Boxes，因此 Output 深度要等於『預測值\* Boxes 數量(9)』。



參考來源：

[簡介 Faster R-CNN](#)  
[Faster R-CNN](#)



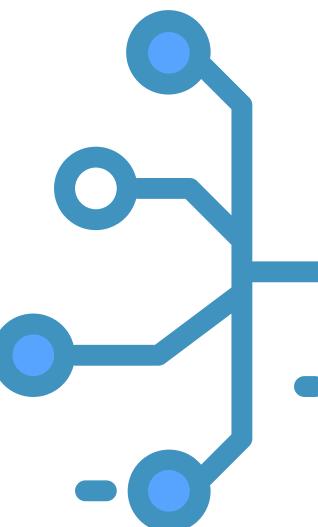
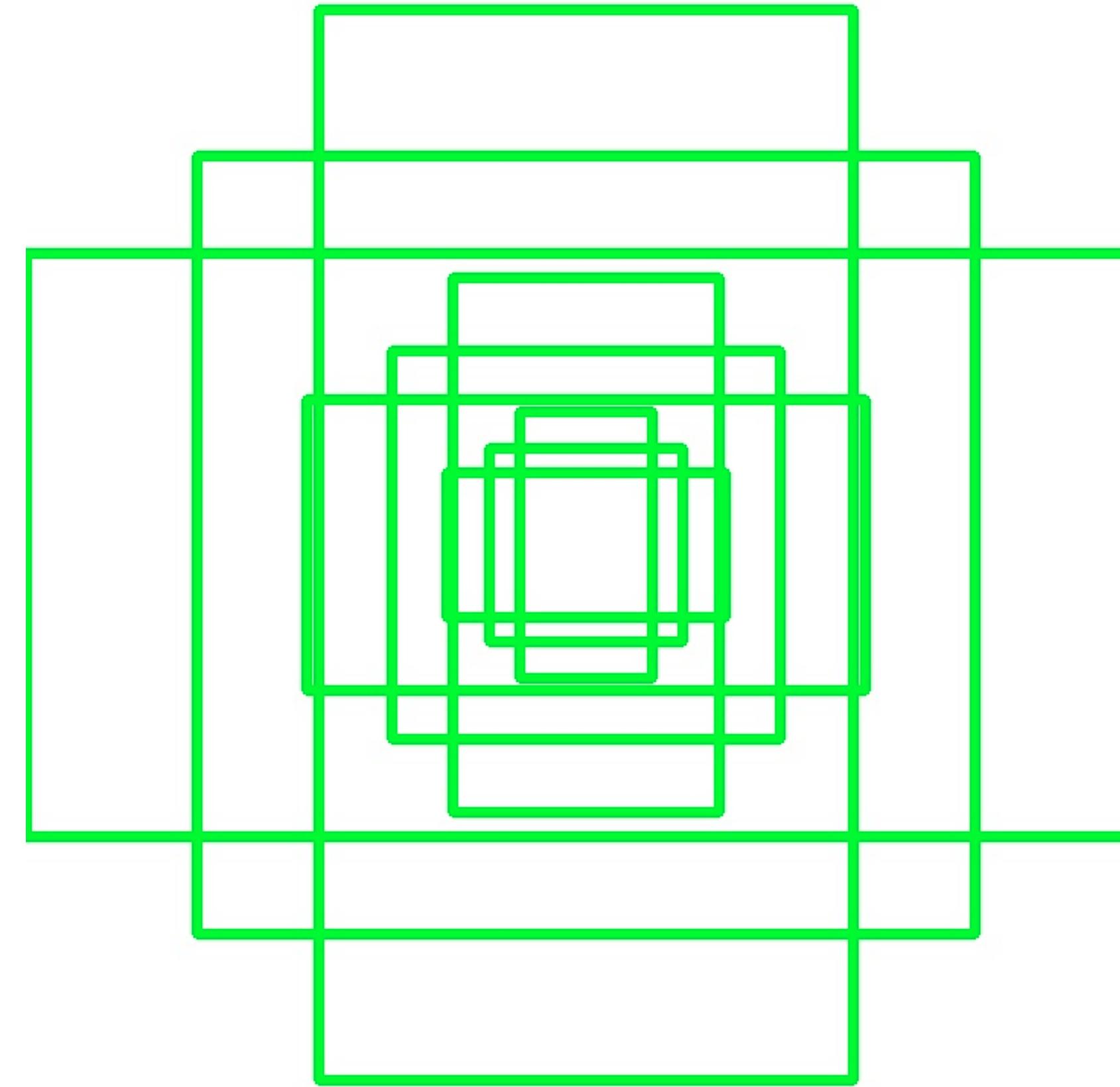


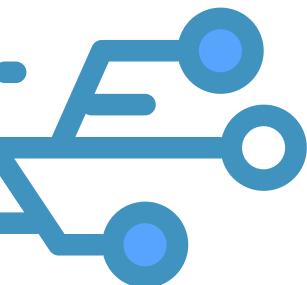
# 如何設計 Default Anchor Box



再來要在每個 Pixels 上設計 9 個 BBOX

- 這九個 BBOX 分別為三種尺度，每個尺度有三種長寬比的 BBOX。

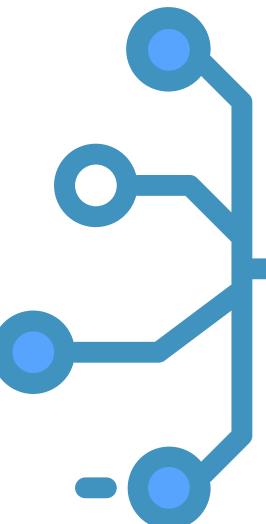
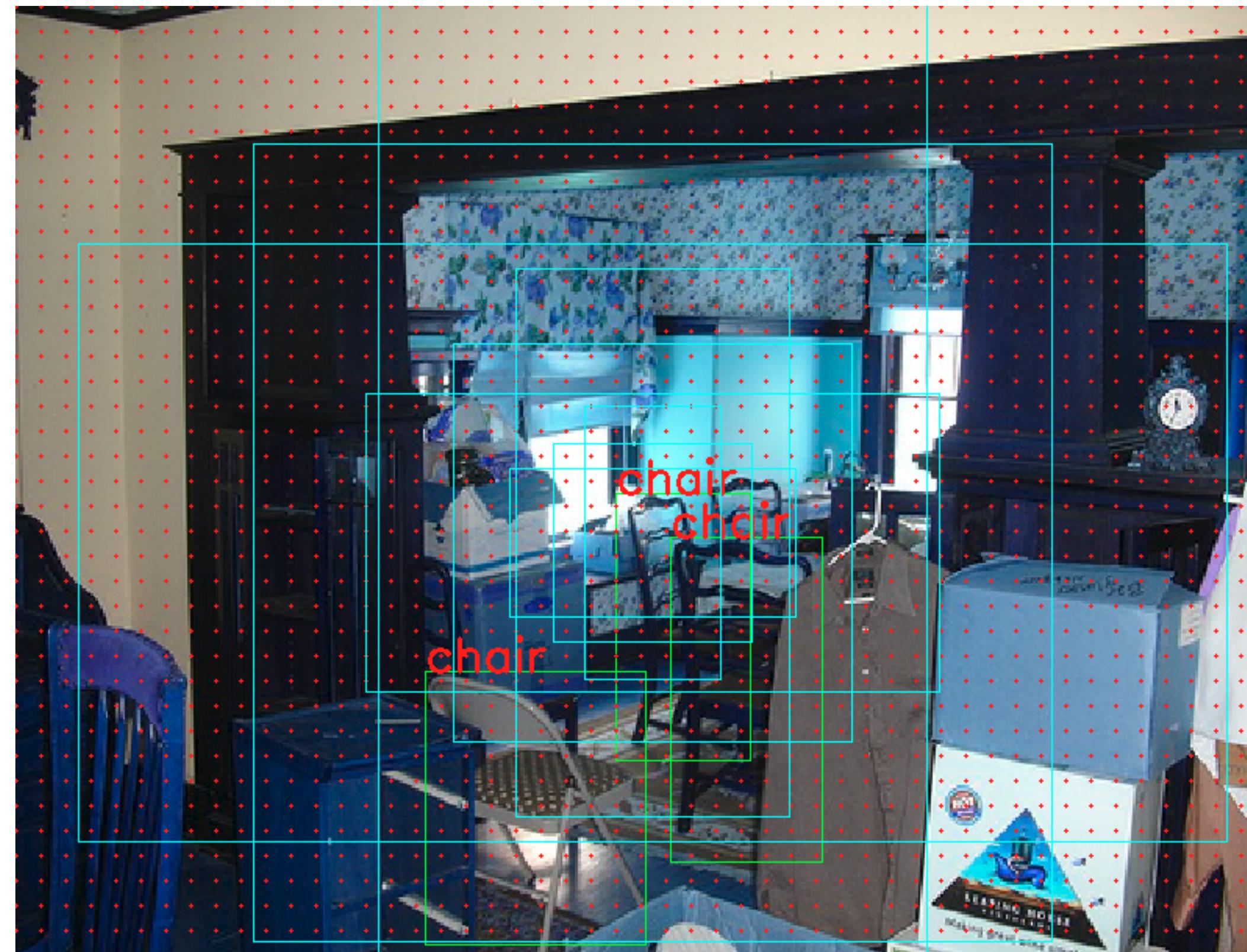


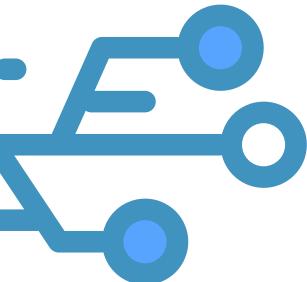


# 如何設計 Default Anchor Box



Default Anchor Box 運用 Feature maps 上的 **Pixels當作中心點**，並在每一個點上放入 **9 個不同大小、長寬比例**的 BBOX (看圖中藍色框部分)，回放為原圖就像是下圖所展示的樣子。



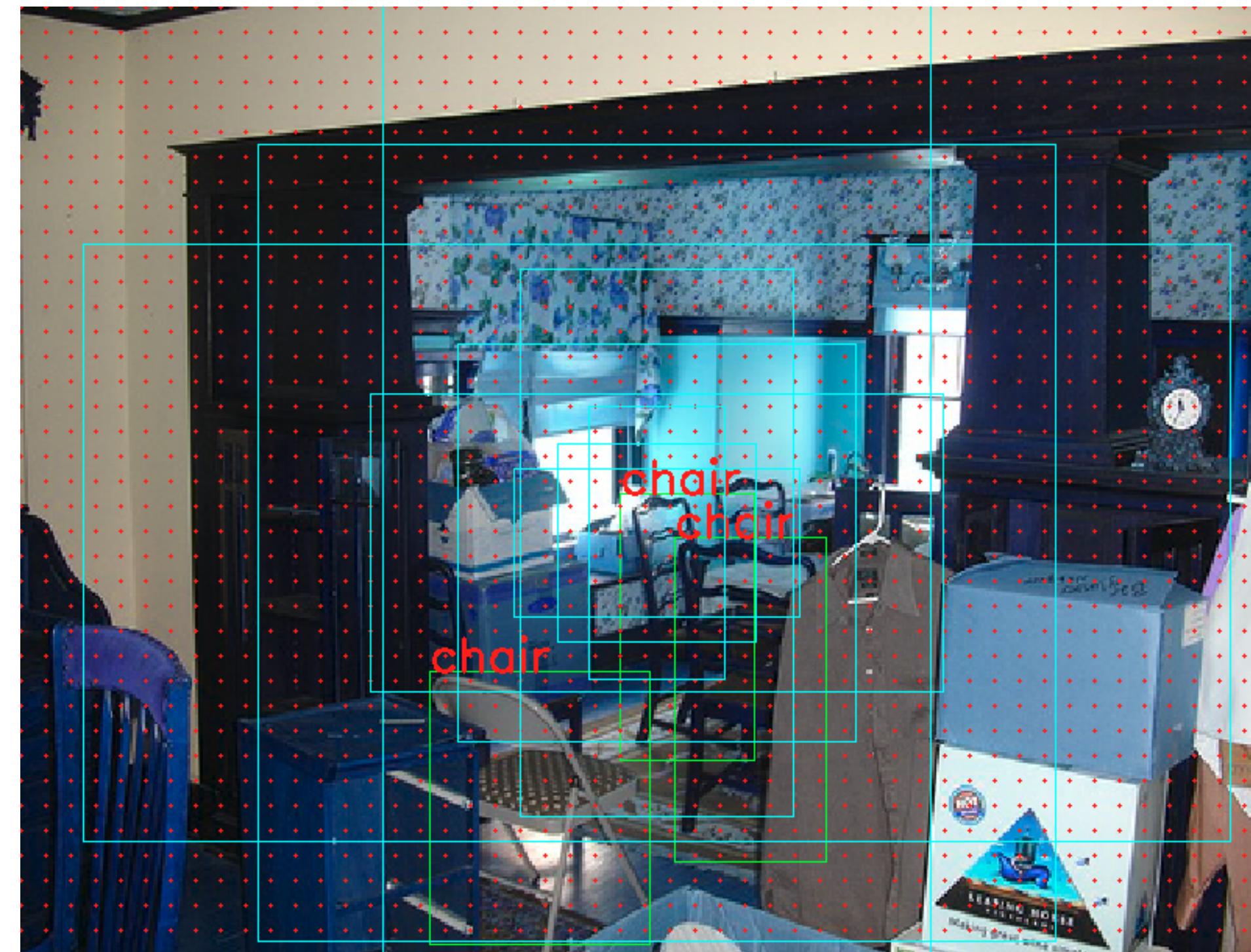


# 如何設計Default Anchor Box

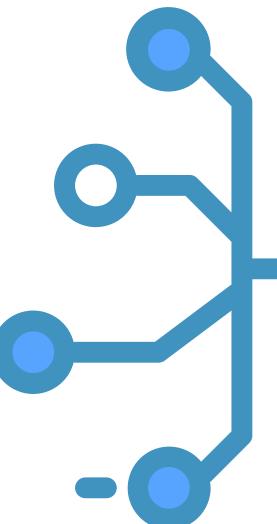


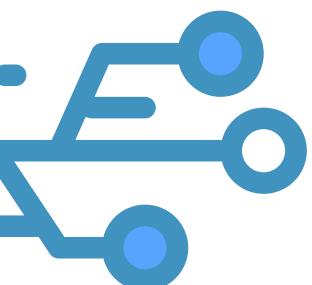
回放原圖原理是透過 Feature Map 大小與原圖大小比例換算，舉個例來說，原圖為 300\*300，在 30\*30 的 Feature Map 上畫了一個 1\*1 (pixel) 的 BBOX，其對應回原圖就是一個 (10,10) 的 BBOX，兩者擁有相同的 Receptive Field。

1\*(300/30)



參考來源：[RCNN works](#)

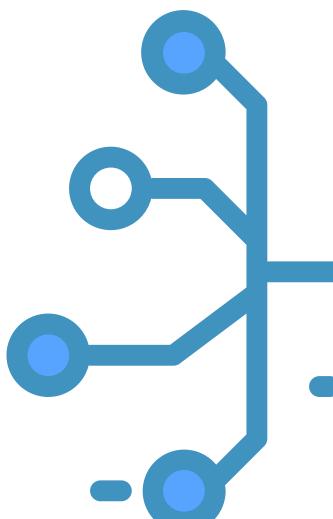
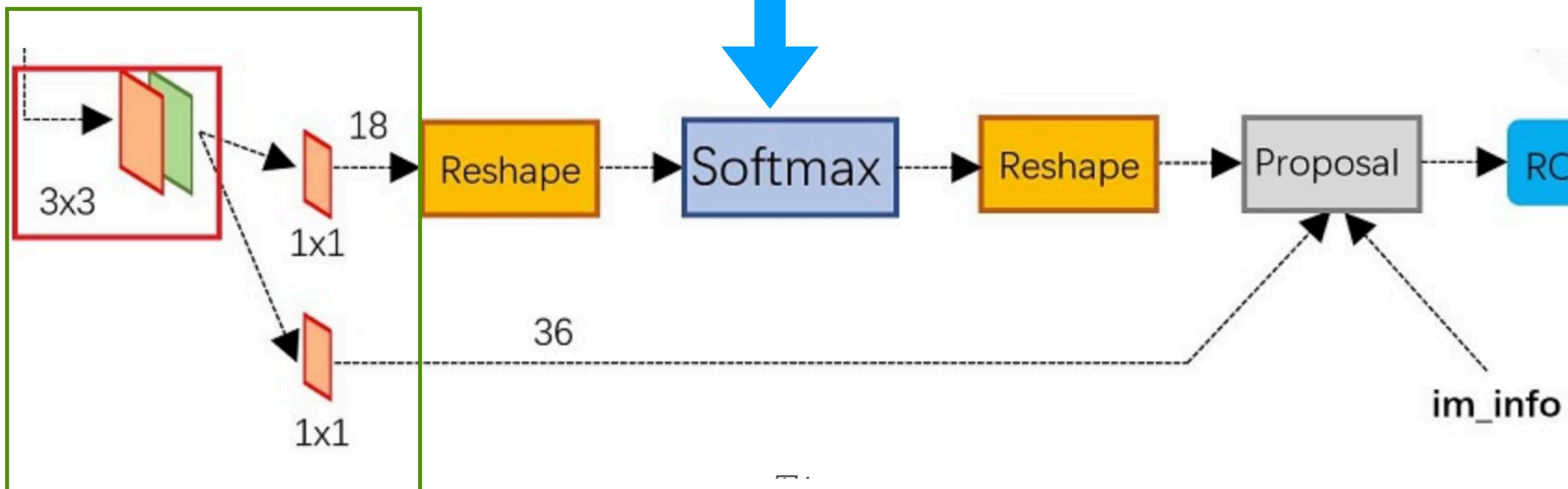


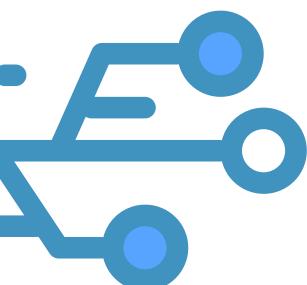


# Region Proposal Network



決定深度後，上方支線會學習每一個 Default BBOX 是前景還是背景，這個步驟主要是透過 **Softmax** 函數做分類。

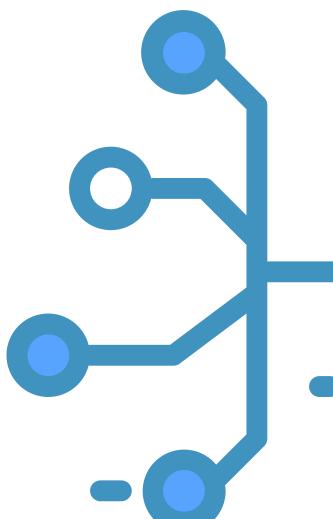
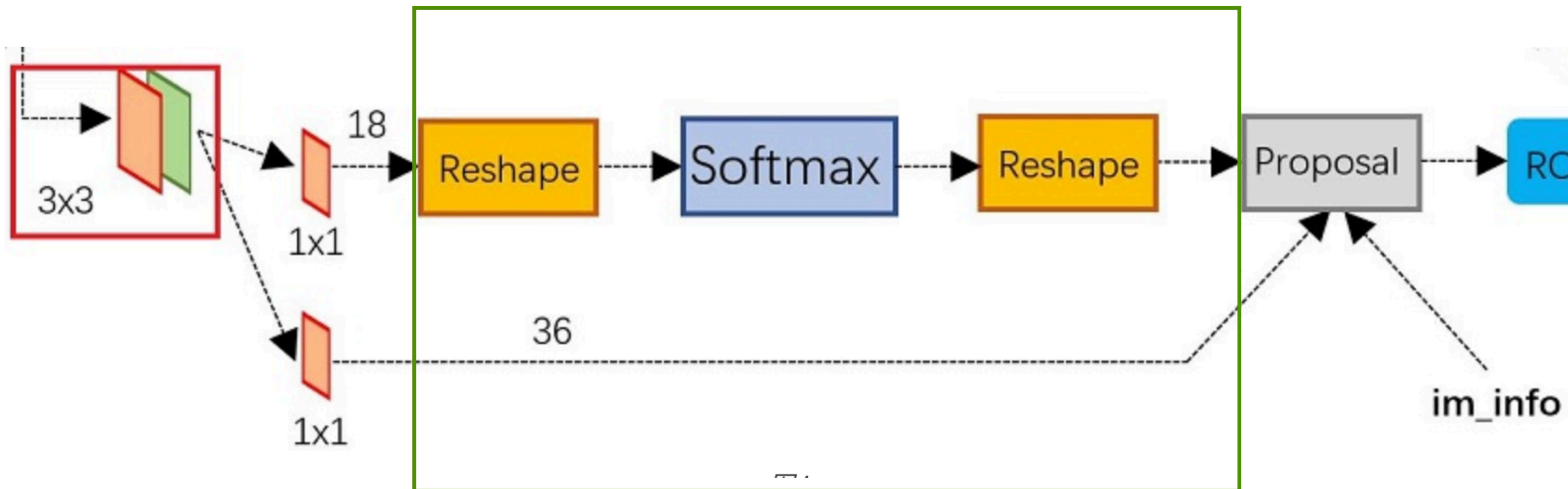


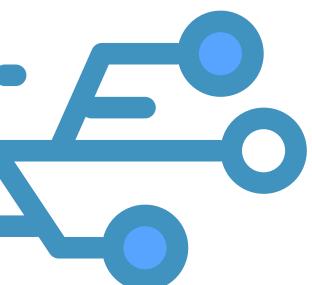


# Region Proposal Network



此階段為 RPN 的重點，我們透過『學習』讓模型能更精準地提出可能包含物件的『Region Proposal』。



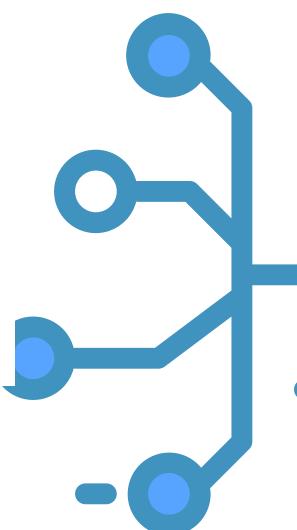
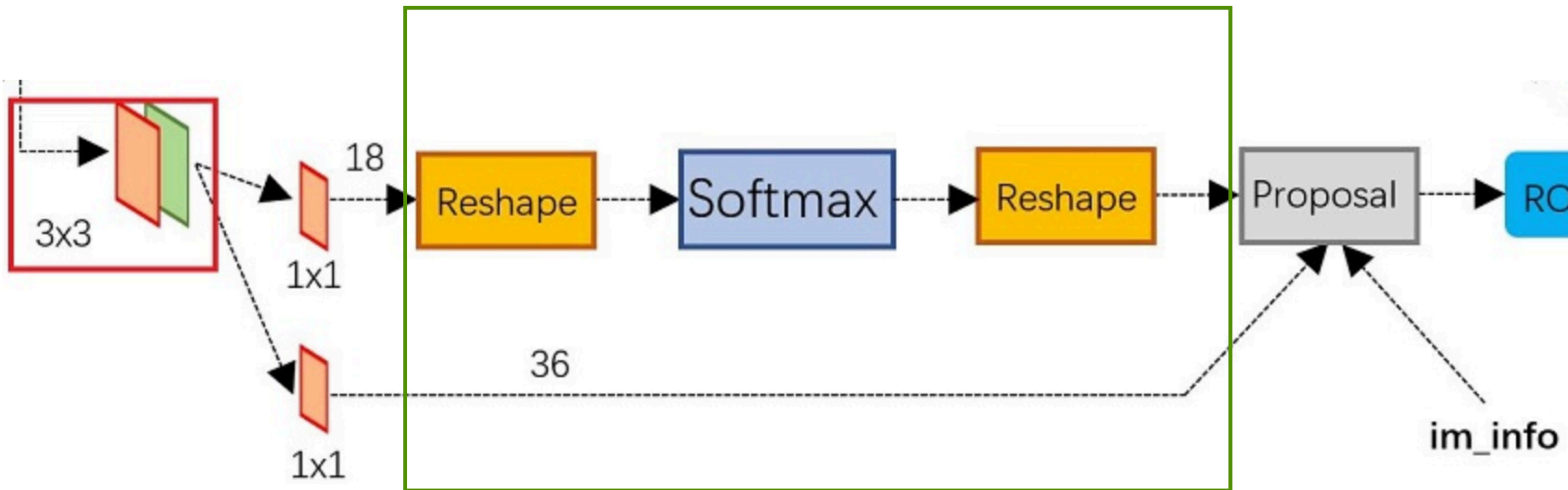


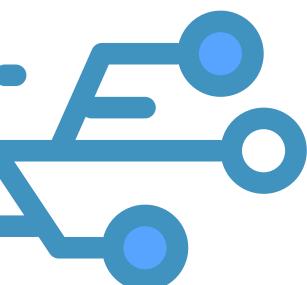
# Region Proposal Network



輸出預測框如與標註框的 IOU 大於一定的值我們就稱這個 BBOX 是前景，原文中是將  $\text{IOU} > 0.7$  設為是前景，藉此讓模型學習物件的特徵。

輸出預測框由 Default BBOX 與輸出  $[\text{delta } (x,y,w,h)]$  轉換所獲得，  
下一章會細講原理。

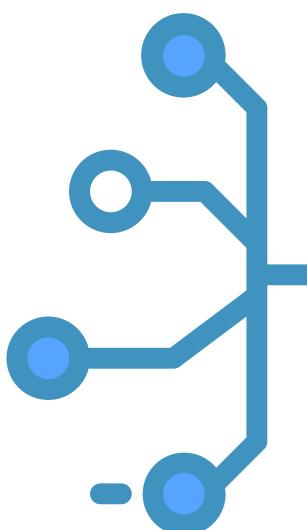
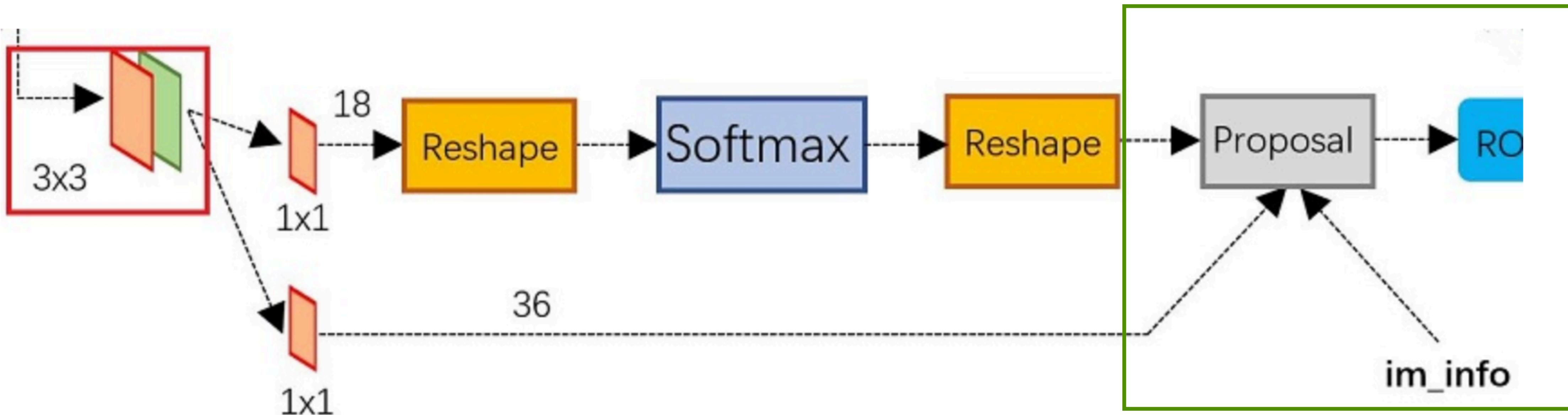


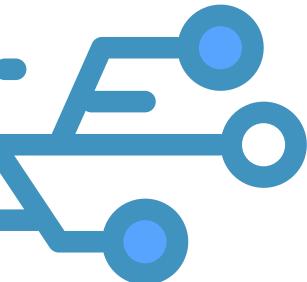


# Region Proposal Network



而 RPN 的輸出資訊就是 BBOX 的  $[x, y, w, h]$  的偏移、縮放量與前後景 Score 的資訊。

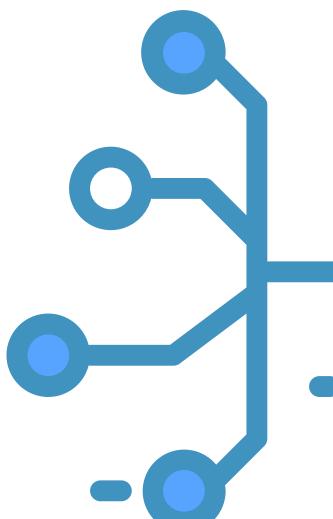
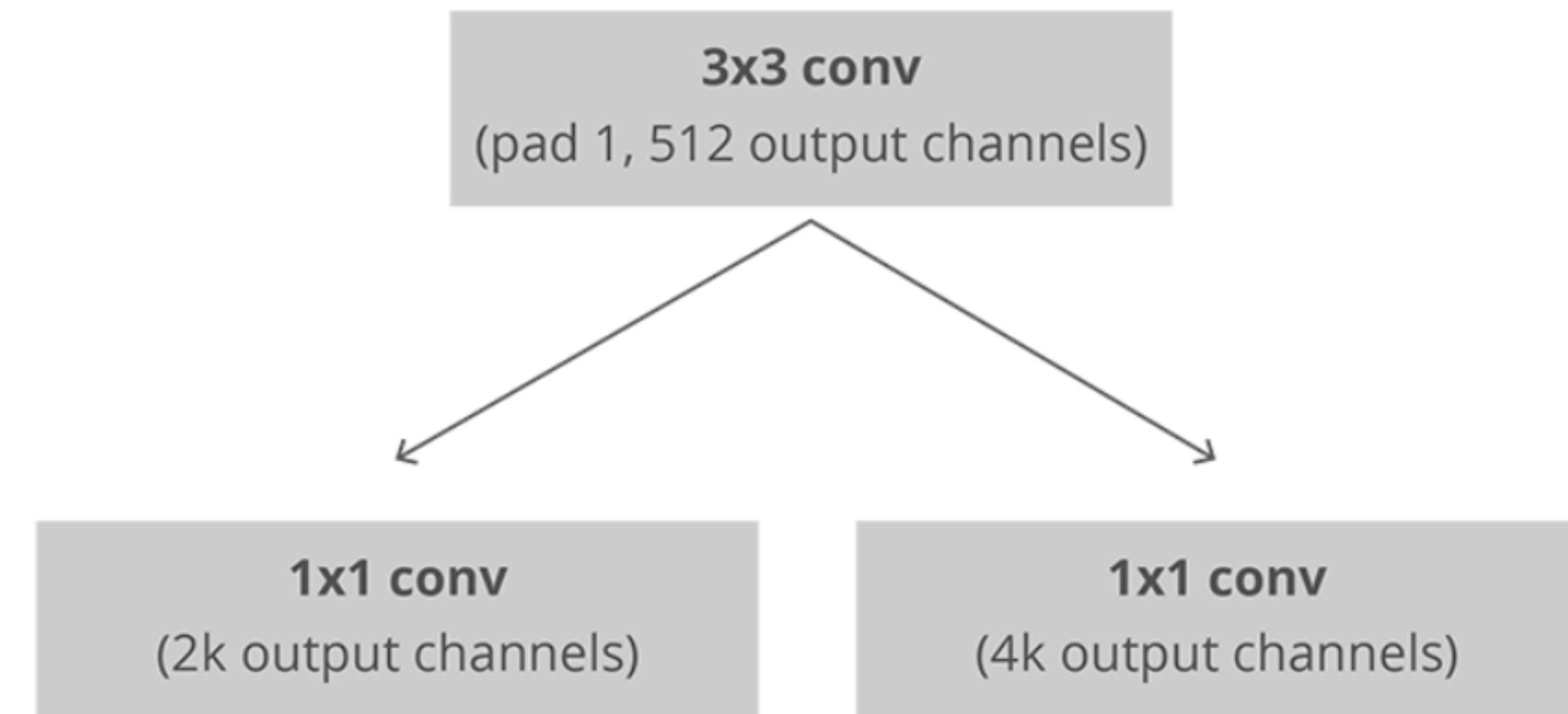


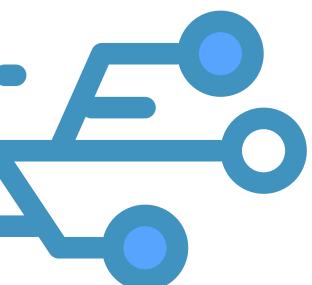


# Region Proposal Network



簡化的 RPN 示意圖如下，Input 先經過  $3 \times 3$  卷積，再通過兩條  $1 \times 1$  卷積，其中 Input 為 CNN backbone 最後一層的 Feature map。

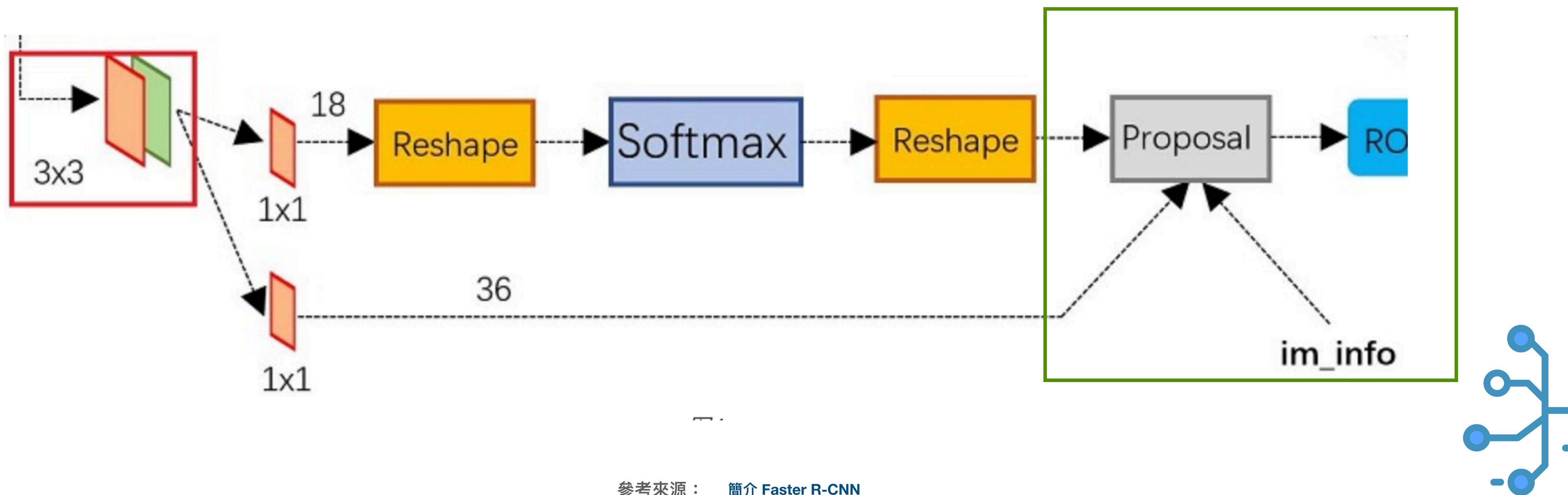


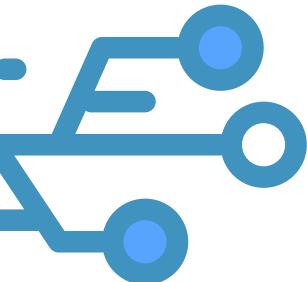


# Region Proposal Network



RPN 輸出前後景的 BBOX，會依照 **score排列**，在訓練階段時，最後輸出 256 個 BBOX 紿下一個階段(手動設置，可以變動)，128 個前景與 128 個背景，這麼做的原因是為了平衡前景、背景框數量，在預測時是輸出所有的 positive 的 proposals (前景)。

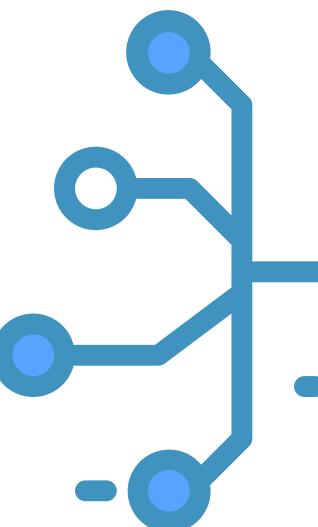


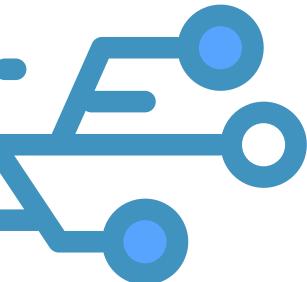


# Region Proposal Network



- 了解 RPN 後我們就可以發現，RPN是一個**可以學習的架構**，能更精準的提出 Region Proposal 。
- Faster R-CNN 有很高的準確率與 RPN 關係相當大，RPN 這一步等於先幫 **Proposal**做過一次**BBOX regression**，因此這個『Proposal』本身就已經有一定的準確率。





# 推薦延伸閱讀



## 2. Architecture#06

### RPN as reg. proposer

$$\text{IoU} = \frac{A \cap Gt}{A \cup Gt} \left\{ \begin{array}{l} > 0.7 = \text{object} \\ < 0.3 = \text{not object} \end{array} \right.$$

By Ardian Umam

12

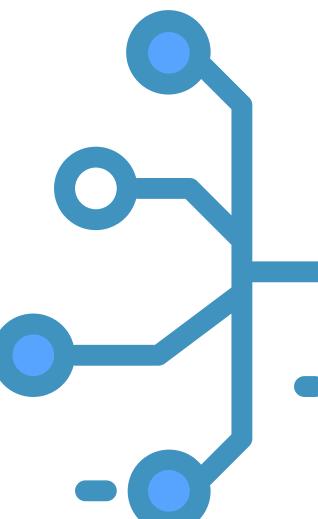
### 3. How RPN (Region Proposal Networks) Works

觀看次數：49,692次 · 2017年3月31日

1 290 74 分享 儲存 ...

## RPN是如何運作的？

連結



# 解題時間 Let's Crack It



請跳出 PDF 至官網 Sample Code & 作業開始解題