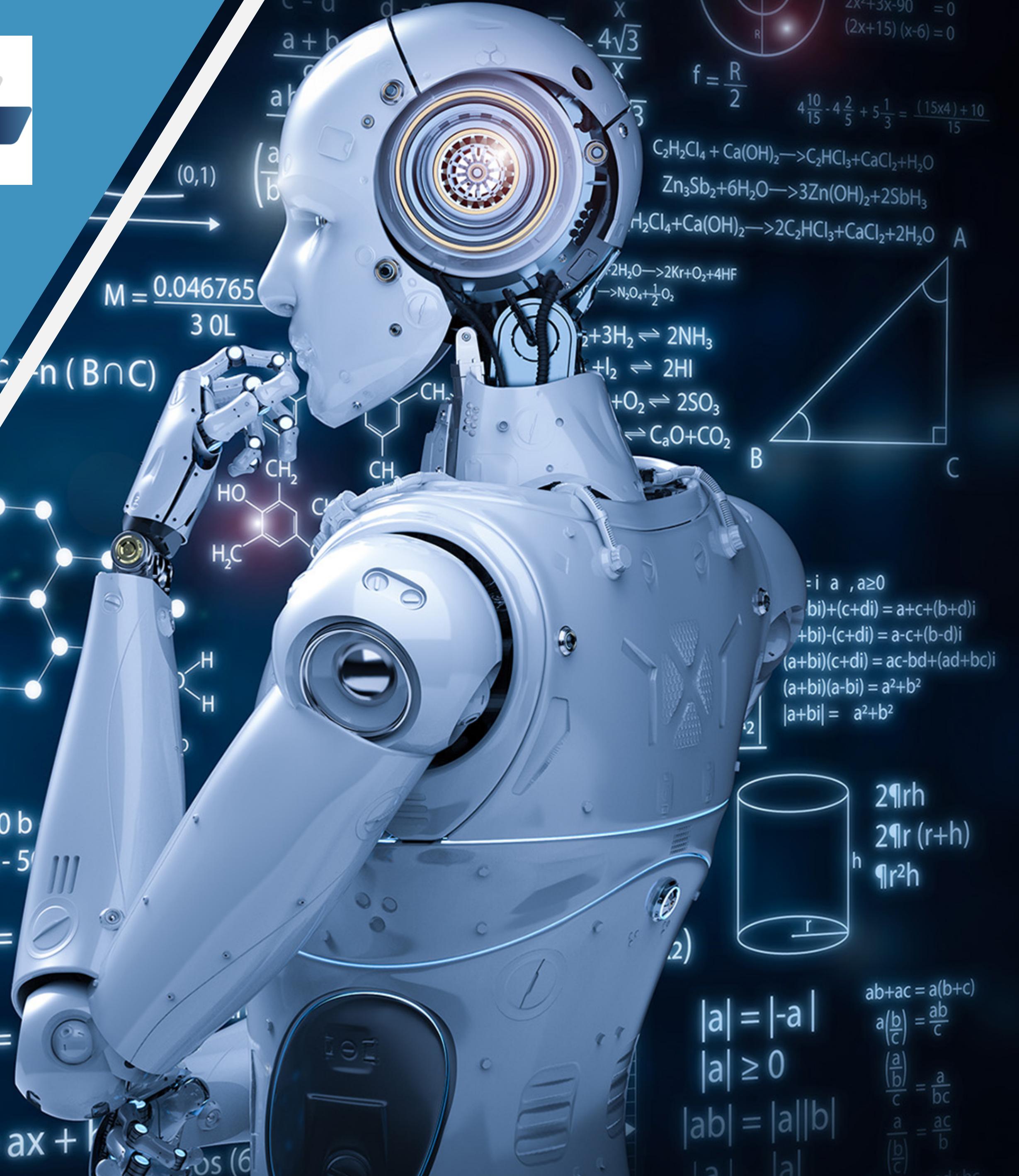




# Day 23

## 深度學習與電腦視覺 學習馬拉松

**Cupay** 陪跑專家：楊哲寧





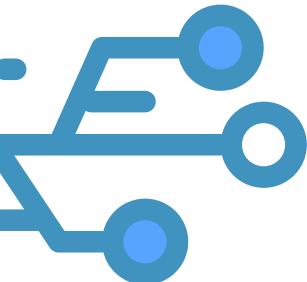
# 深度學習理論與實作

## Object Detection (物件偵測) 原理



# 重要知識點

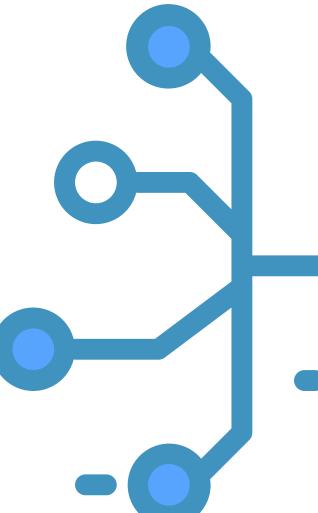
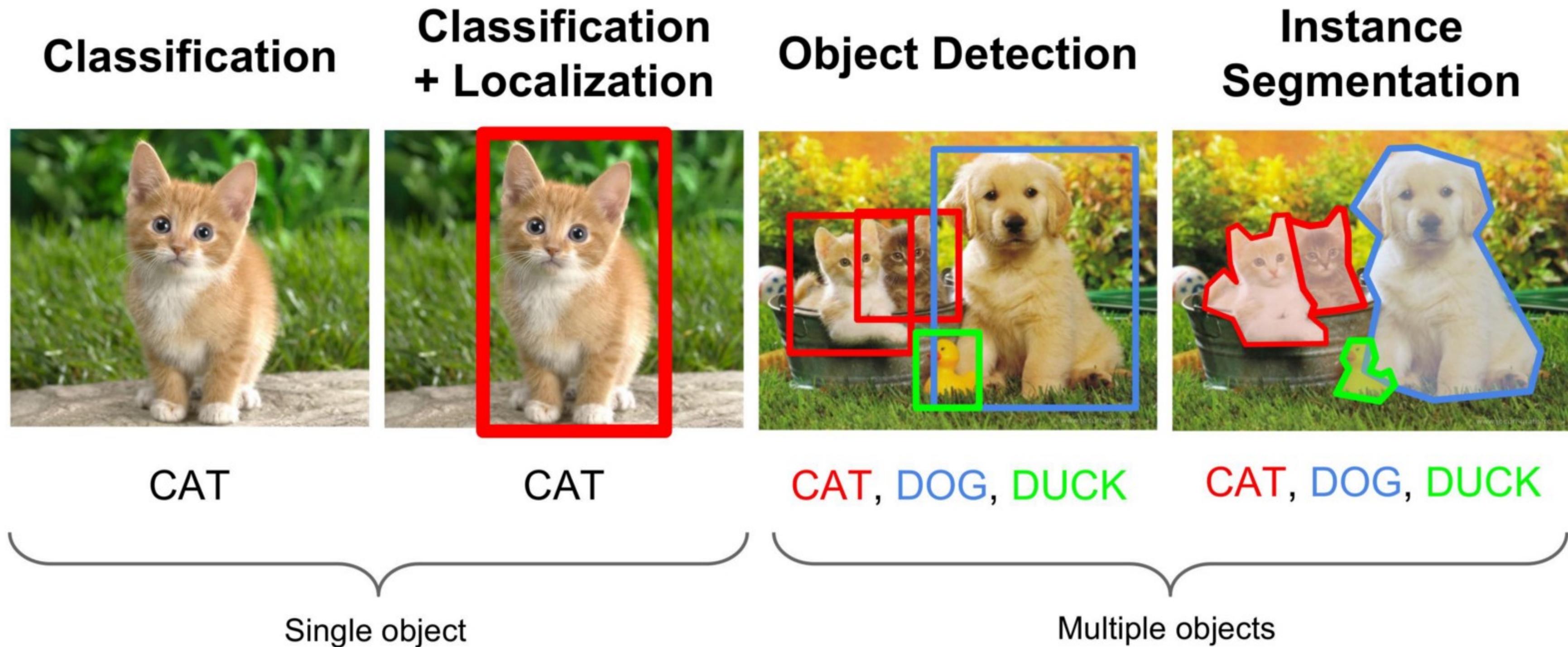
- Object Detection 用來解決怎麼樣的問題
- 如何設計 Object Detection 的 Loss
- One Stage 與 Two Stage 的差異

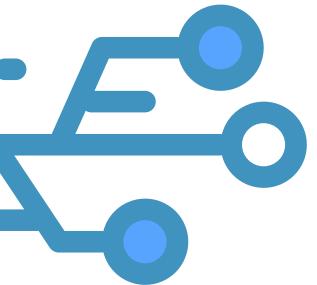


# 什麼是Object Detection



與一般基本的 CNN 分類器不同的是，Object Detection 能告訴你物件在照片中的位置，並同時能偵測多個不同的物件。

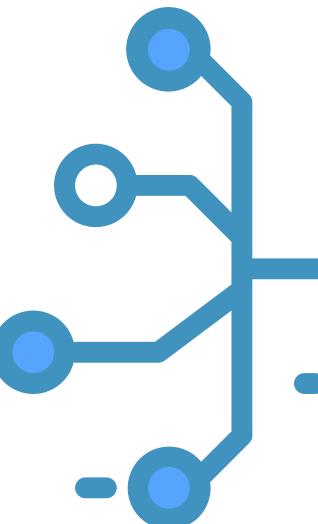


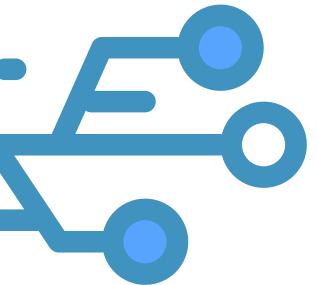


How?



如何做到偵測物件在照片中的位置並同時能偵測多個不同的物件？

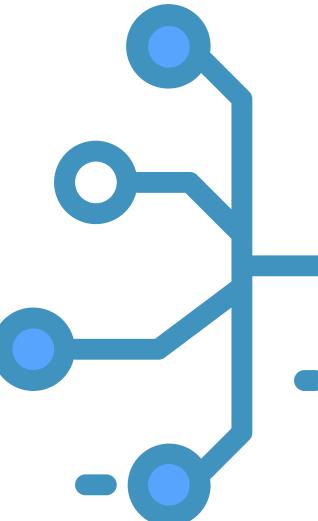




How?



透過模型架構與 Loss Function 設計使我們能同時定位物件並偵測物件



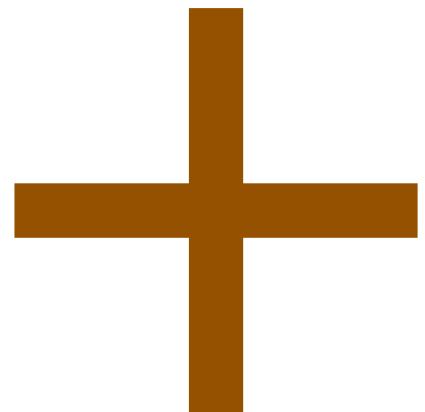


# Loss Function

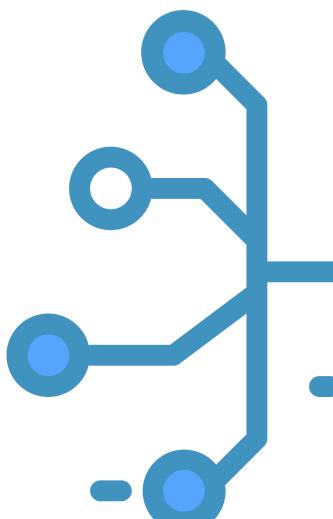


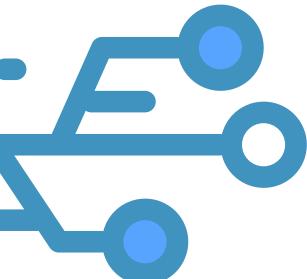
儘管 Object Detection 的演算法很多種，Loss Function 都不會脫離兩大核心。

## Bounding Box Regression



## Classification

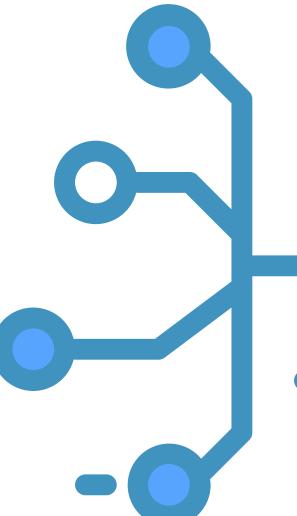
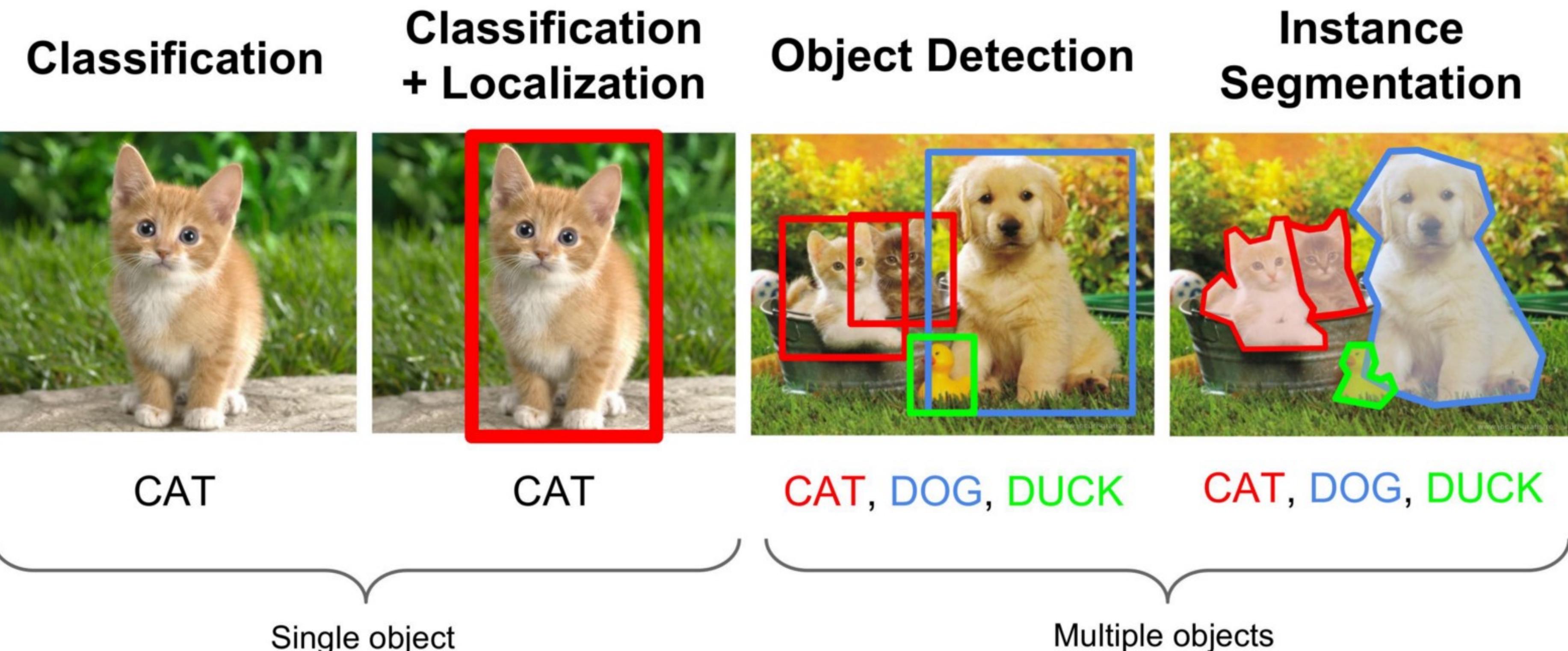


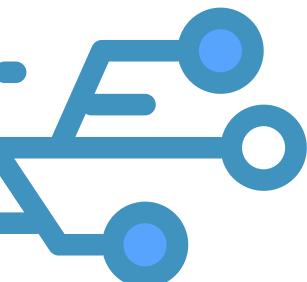


# Bounding Box Regression



Bounding Box 就是我們圖中看到的框，Loss 的其中一項就在學習怎麼將這個預測框精準的定位( 一種 Regression )，使其完整的包涵物件，細節如何辦到的在之後章節會說明。

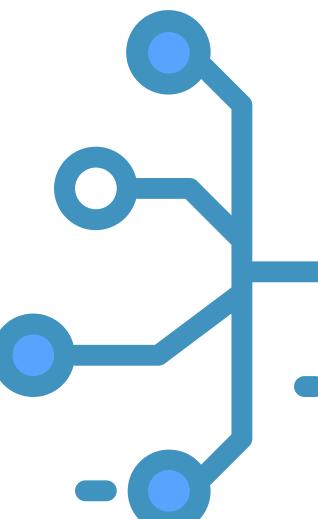
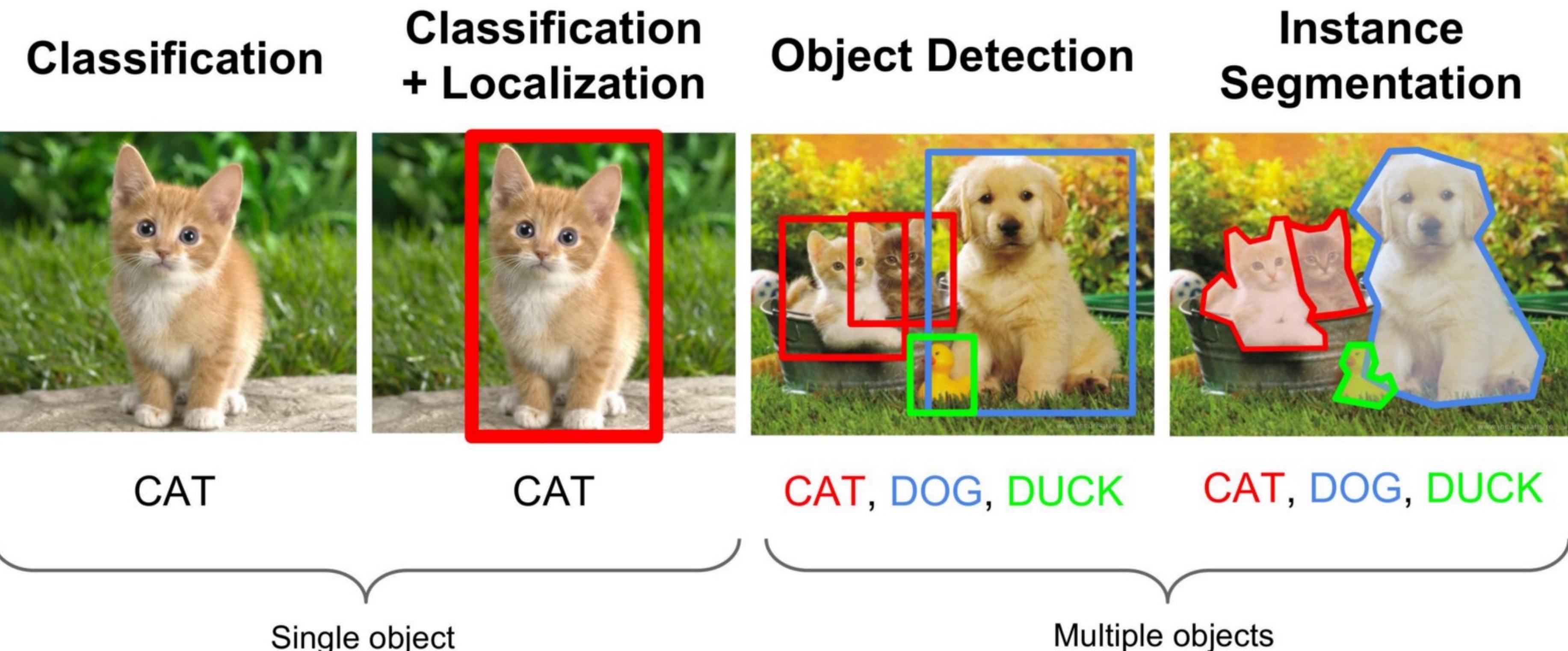


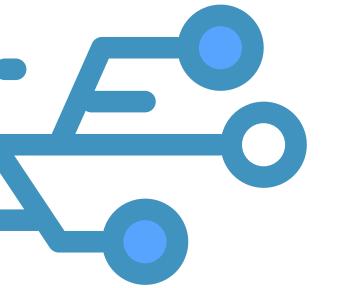


# Classification



Classification 就很好理解了，當我們成功定位物件後，使用與一般 CNN 分類器相同的Loss ( Softmax+Cross Entropy )即可。

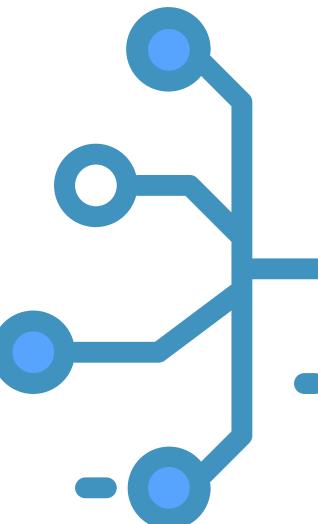


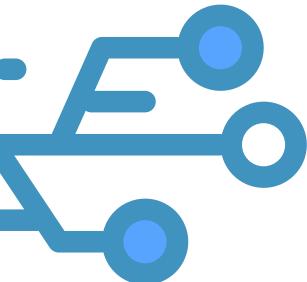


# Object Detection



其中值得注意的一點是，Object Detection 是同時在學習『定位』與『分類』這兩件事情，所以可以共享 CNN 層。





# Object Detection分類

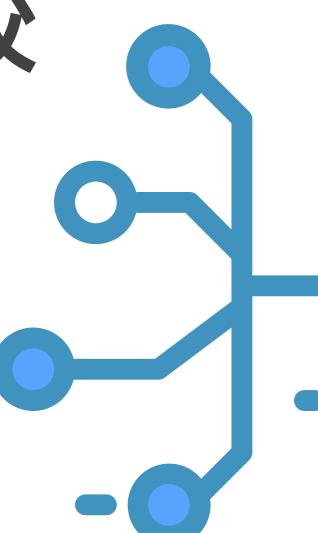


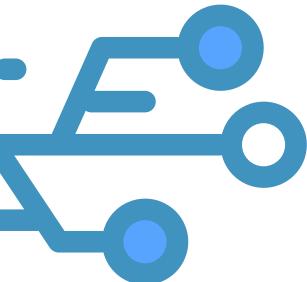
而 Object Detection 演算法中主要可以分為兩路，分別為

## One Stage 與 Two Stage

單階段      與      雙階段

兩者主要差異在於是否需要『透過特定運算方法』來提出可能包含物件的『框 ( Region Proposal )』，由於 One Stage 不需要這個步驟，通常運算速度更快，但精度也相對較低，後面章節會更詳細介紹其差異。





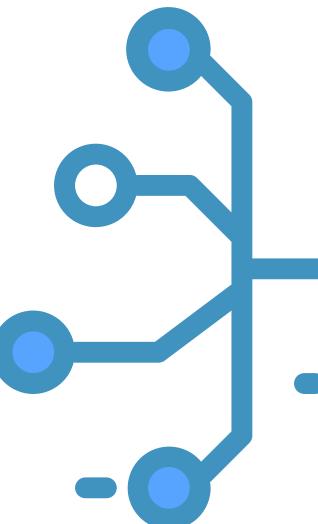
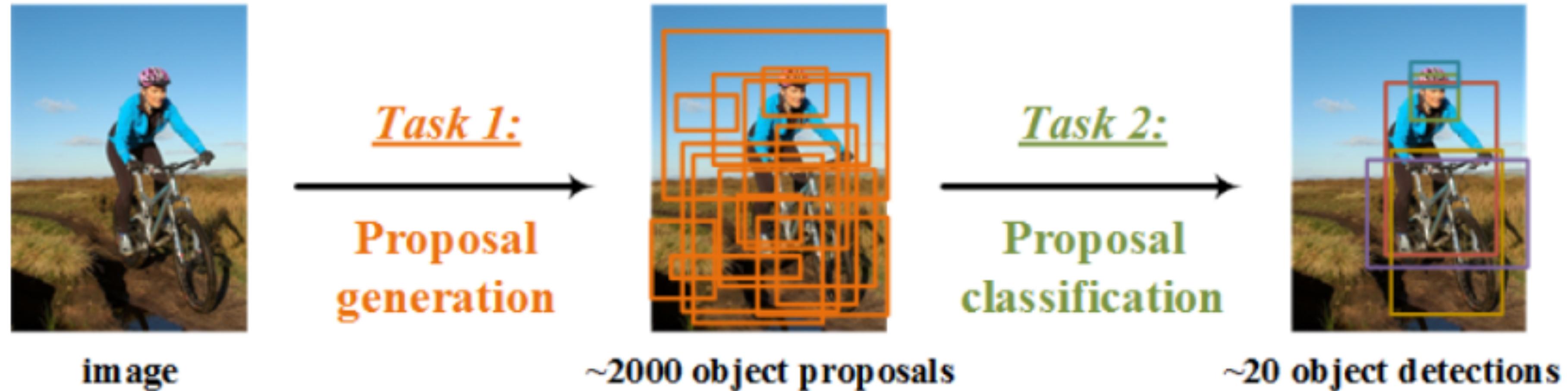
# Two Stage

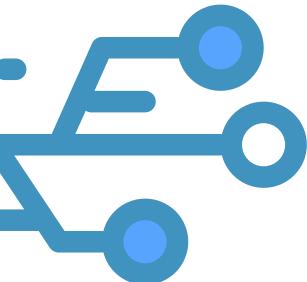


早期的 Object Detection 主要為 Two Stage，其重點為：

先找出 Region Proposal，再做回歸與分類

Region Proposal 就是可能有包含物件的 Bounding Box。





# Region Proposal

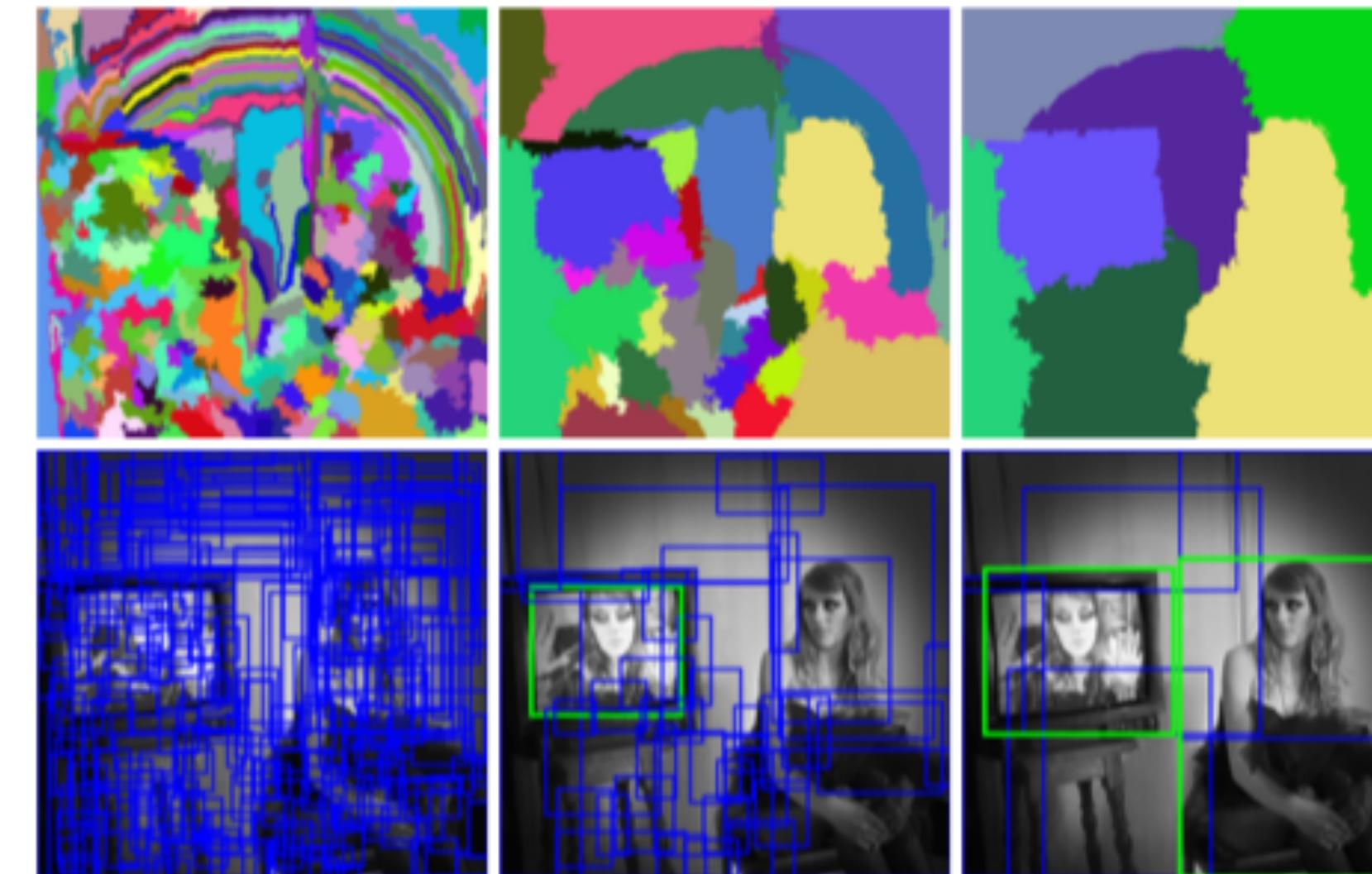


提出 Region Proposal 的方法很多，如：

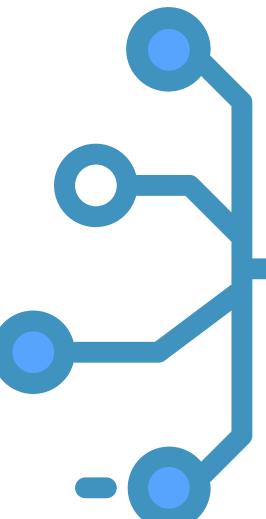
- Selective Search
- RPN

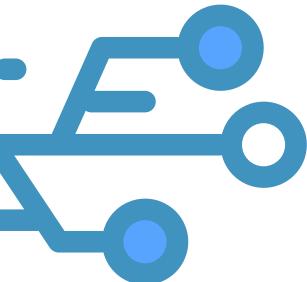
Selective search 是圖像 segmentation 的一種，其主要是基於圖像的顏色，紋理，大小和形狀兼容計算相似區域的分層分組。

Selective Search 切格圖像的結果



透過圖像切割後的結果提出候選框



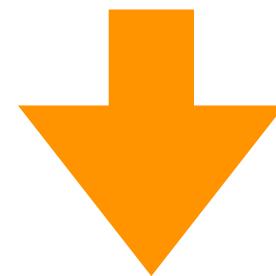


# Selective Search



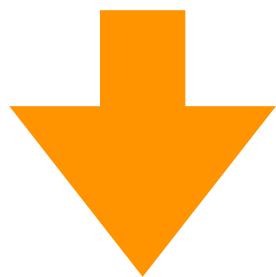
Selective Search 主要由兩部分組成，分別為

## Hierarchical Grouping Algorithm



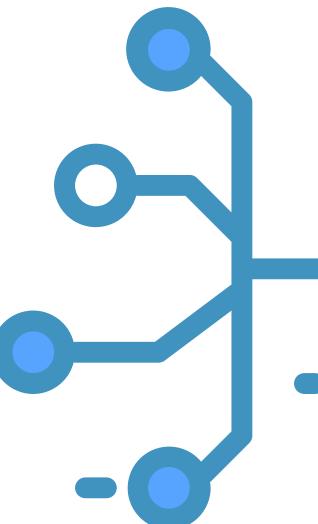
- (1) 計算所有鄰近區域之間的相似性
- (2) 兩個最相似的區域被組合在一起
- (3) 計算合併區域和相鄰區域的相似度
- (4) 重複 2、3 過程，直到整個圖像變為一個地區，由於 RCNN 預設是 1000-2000 個 proposal，所以分到一定數量左右就會停止。

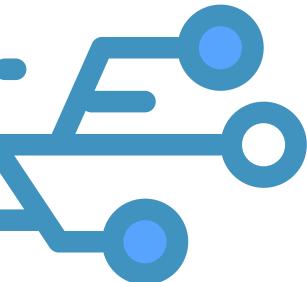
## Diversification Strategies



### 相似性

- (1) 利用各種不同不變性的色彩空間
- (2) 採用不同的相似性度量，如顏色、紋理、尺度等。

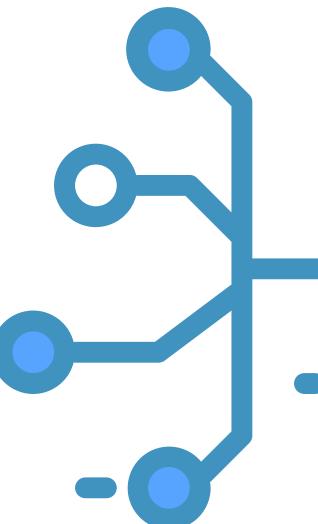
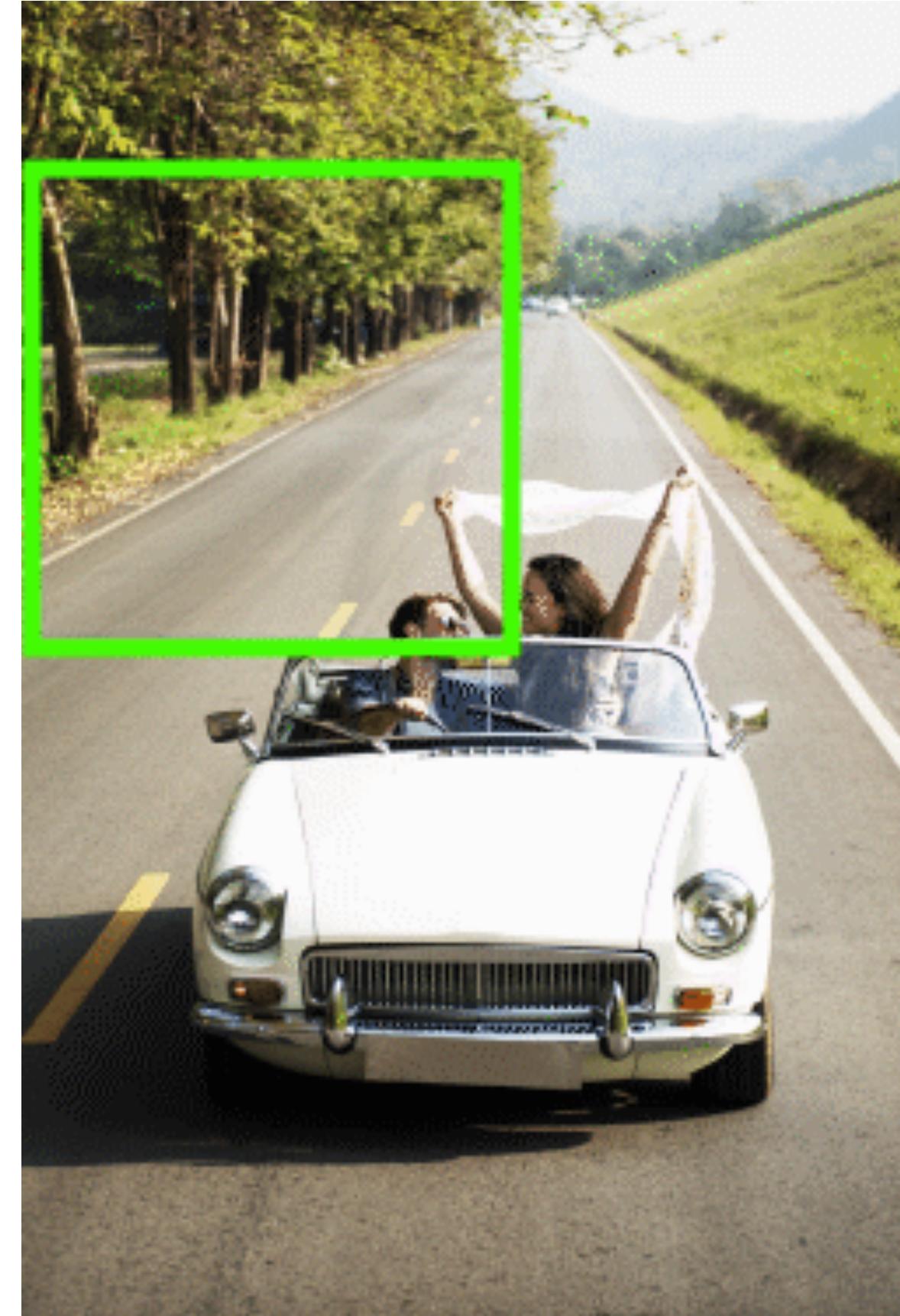


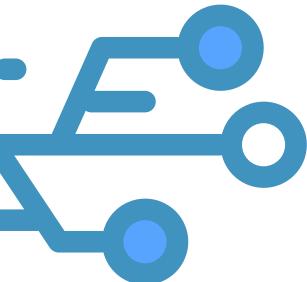


# RCNN-Selective Search



Sliding window 是早期的做法，等於將整張圖都看一遍，有興趣的學員們可以觀看參考資料的影片，而之後章節會再介紹 RPN 與 Selective Search。

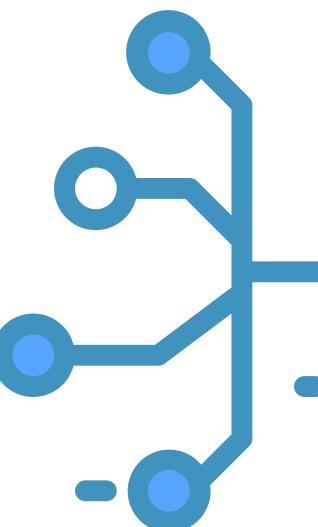
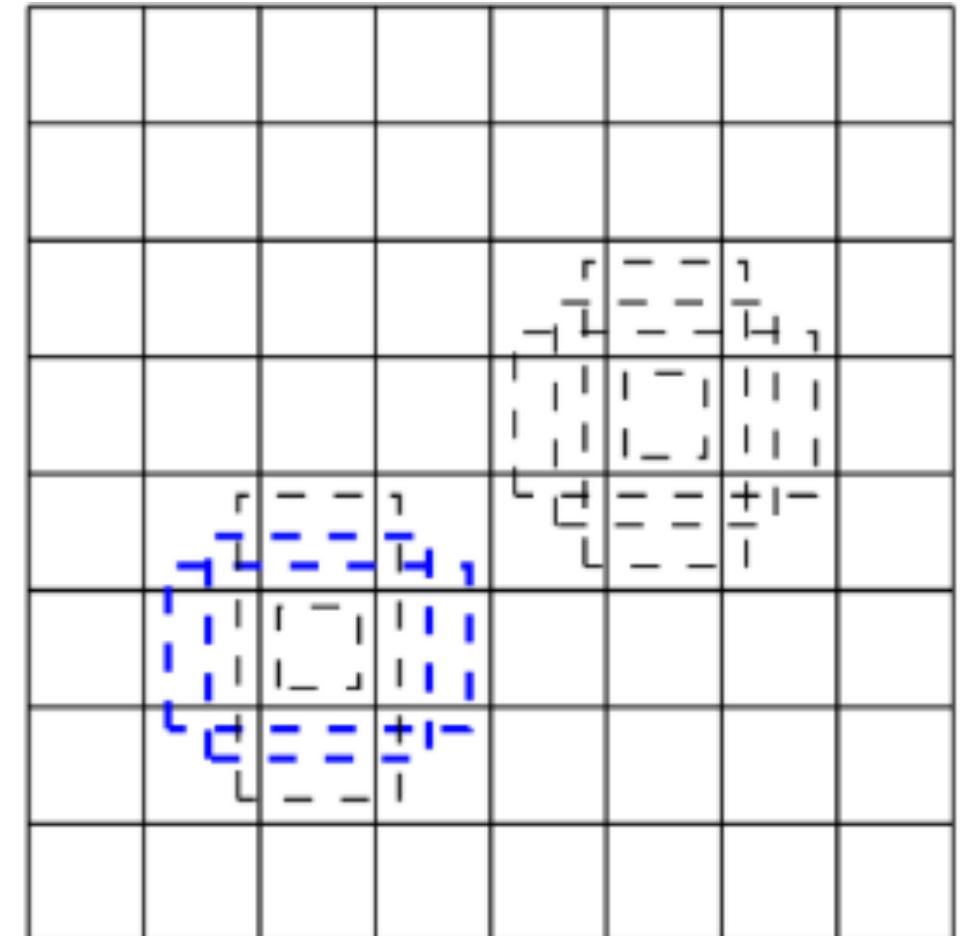


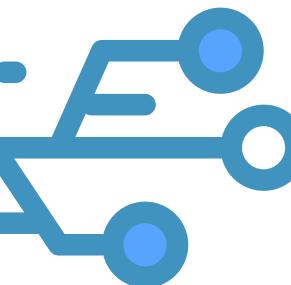


# One Stage



- One Stage 的核心理念就是一次到位。
- 直接對 **Default Bounding Box** 進行回歸與分類。
  - \* **Default Bounding Box** :直接以 Pixels 為中心點提出『Region Proposal』
- 不用再想辦法提出 Region Proposal ，因此通常速度上有較大的優勢，精度上在一開始提出來時明顯落後 Two Stage ，不過近期的演算法已經逐漸能追上 Two Stage 的精度。





# 推薦延伸閱讀

Sliding windows detection

0:00 / 5:48

C4W3L03 Object Detection  
觀看次數：26,377次 · 2017年11月7日

128 分享 儲存

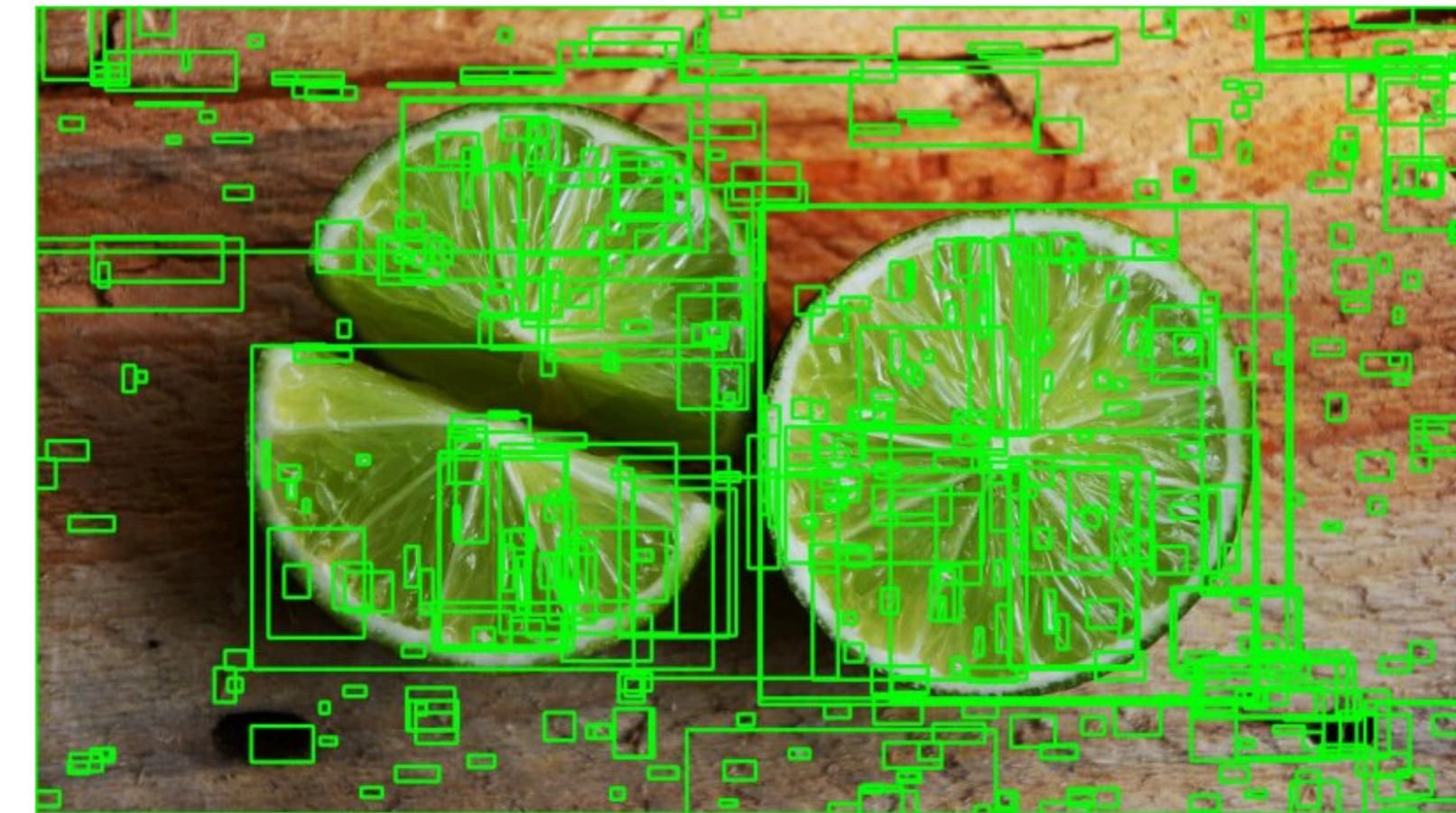
Deeplearning.ai  
6.09 萬 位訂閱者

**Object Detection Sliding Window**  
連結

## OpenCV 教學：實作 Selective Search 物體偵測候選區域演算法

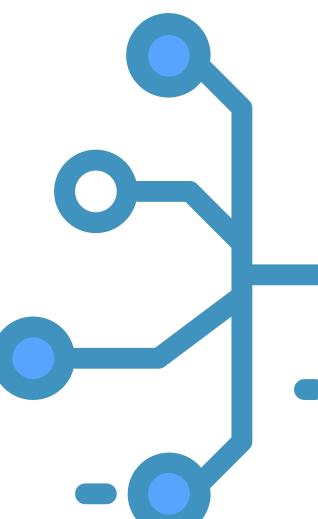
2017/10/26

4 則留言



本篇介紹如何在 OpenCV 中實作 Selective Search 物體偵測候選區域演算法。  
初版的 R-CNN 是將 Selective Search 所得到的候選區域，放進 CNN 中進行判斷，為了更清楚理解 Selective Search 的運作，以下我們直接使用 OpenCV 來撰寫一個 Selective Search 的實作版本，觀察該演算法實際執行的結果。

**Selective Search 原理與教學**  
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