



YOLO

David Chiu

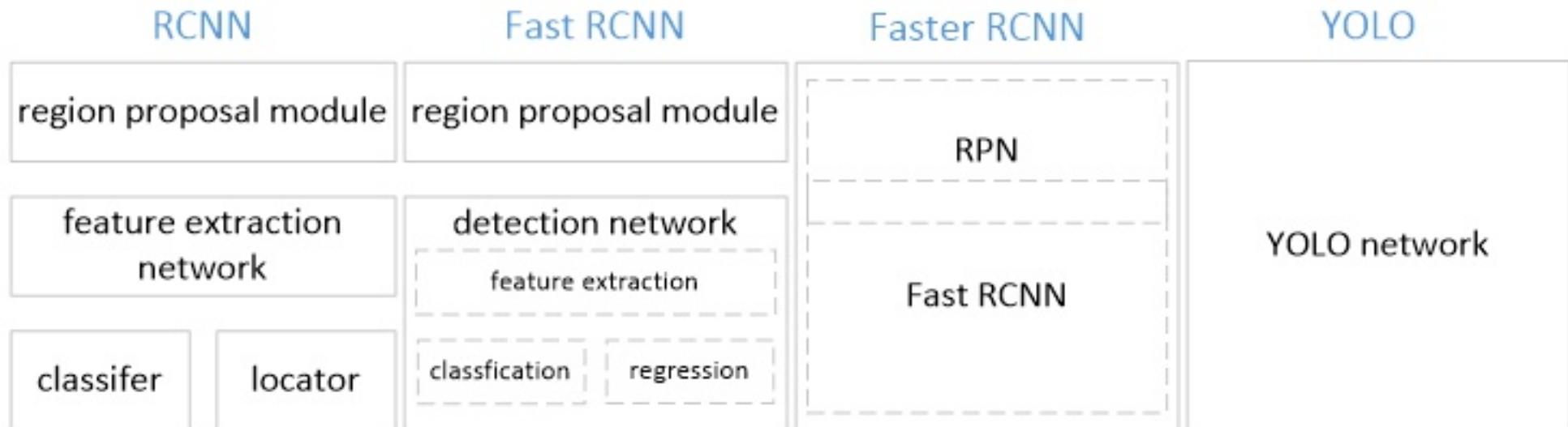
# YOLOV1

# You Only Look Once

■ rcnn / fast-rcnn / faster-rcnn 為兩步驟檢測演算法

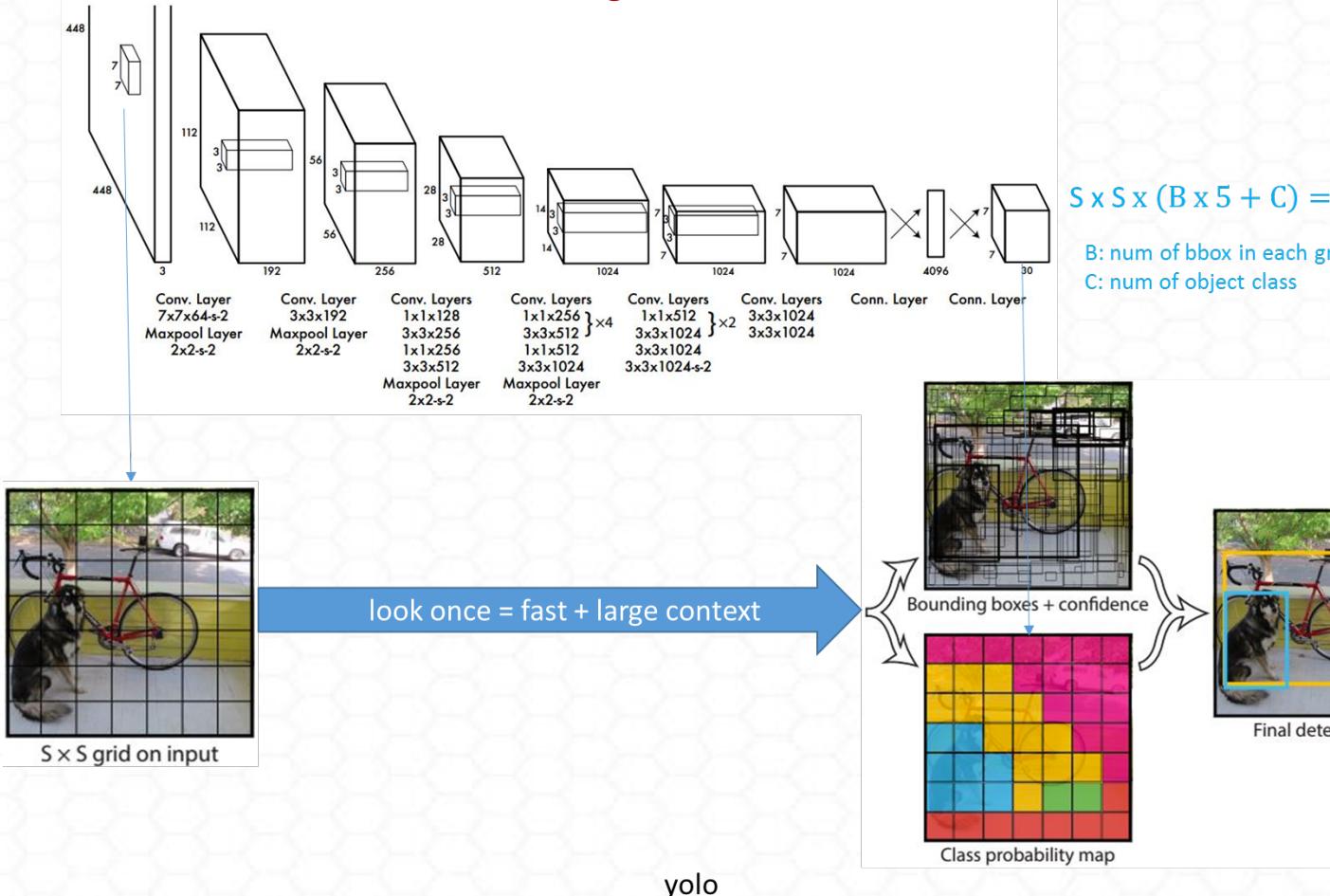
- 物體類別（分類問題）
- 物體位置（迴歸問題）

■ Yolo 只靠單一迴歸方法完成物體位置與類別識別



# YOLOv1

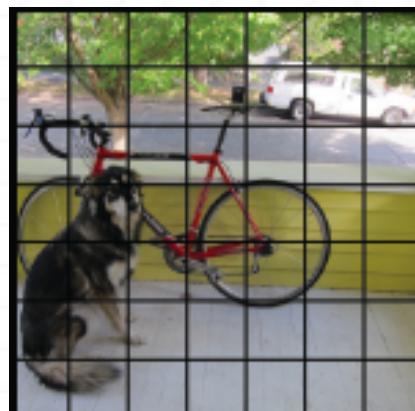
使用 GoogLeNet 作為底層



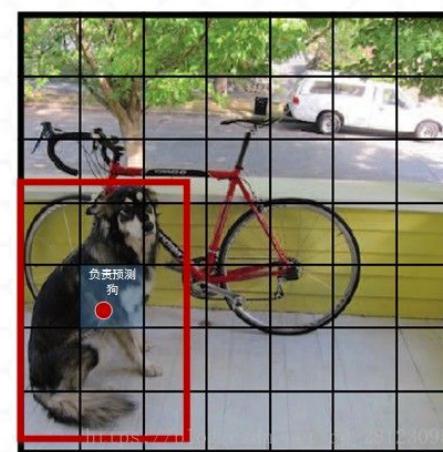
yolo

# YOLOv1

- 將輸入圖像劃分為 $S \times S$ 網格 ( grid )
- 如果目標的中心落入網格單元，則該網格單元負責檢測該目標



$5 \times 5$  grid on input



# Bbox

- 在每個格子上生成2個Bbox，Bbox信息包含5個數據值，分別是x，y，w，h和Confidence
  - x，y是指當前格子預測得到的物體的Bbox的中心位置的坐標。w，h是Bbox的寬度和高度。
  - Confidence包含物體的概率Pr ( Object ) 以及預測框的準是度IOU，其中，若bounding box包含物體，則P ( object ) = 1；否則P ( object ) = 0。IOU為預測Bbox與物體真實區域的交集與並集的比值。

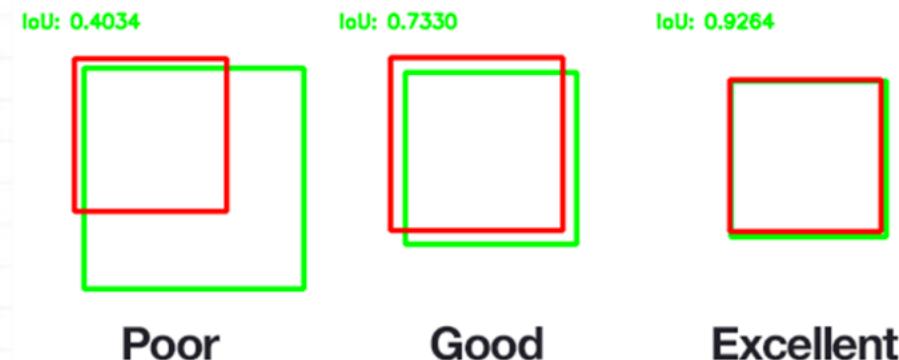
$$\text{Confidence} = \text{Pr}(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

# IOU(intersection over union)

- 衡量預測框與真實框之間相似度的方法。IOU數值介於0到1之間，越高則表示兩個框彼此位置與大小越相似

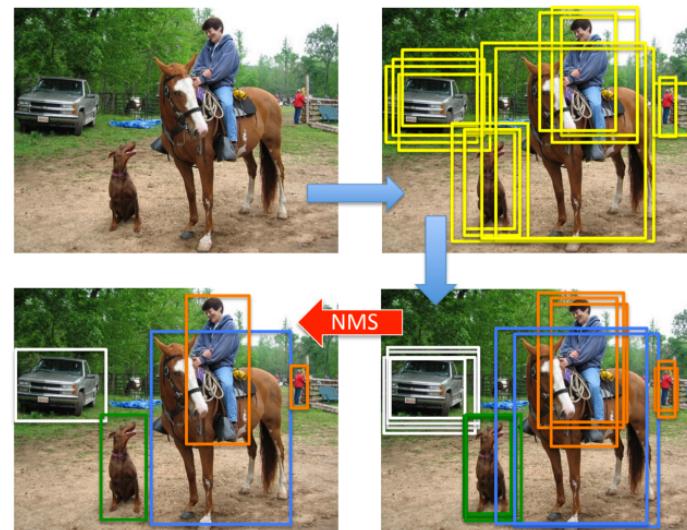
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

http://blog.csdn.net/lanchunhui



# NMS(Non-maximum suppression)

- 當有許多cell grid都預測出有物件時，則會得到許多預測框交互重疊
- 需要根據每個cell grid預測出來的Confidence以及對應的類別機率，保留最高數值的預測框
- 計算是否有機率較小的預測框與該預測框的IOU大於某個閥值 (如0.5)，大於該閥值則捨棄掉機率較小的預測框，小於該閥值則繼續保留機率較小的預測框



# YOLOv1 輸出

- YOLO網路最終的全連接層的輸出維度是 $S * S * (B * 5 + C)$
- 假設輸入圖像解析度是448x448， $S=7$ ， $B=2$ ；採用VOC 20類標注物體作為訓練資料， $C=20$ 。因此輸出向量為 $7 * 7 * (20+2 * 5)=1470$

$$S \times S \times (B \times 5 + C) = 7 \times 7 \times (2 \times 5 + 20)$$

B: num of bbox in each grid

C: num of object class

# YOLO-Loss

$$loss = \sum_{i=0}^{S^2} coordError + iouError + classError \quad [1]$$

- CoordError – IOU 與座標位置相關誤差
- iouError - IOU誤差時，針對包含物體的格子與不包含物體的格子計算二者的IOU誤差
- classError 類別預測誤差對於相等的誤差值，大物體誤差對檢測的影響應小於小物體誤差對檢測的影響。因為，相同的位置偏差占大物體的比例遠小於同等偏差占小物體的比例

# YOLOv1 到 YOLOv2

## ■ YOLOv1 缺點

- 如果有兩個物件中心點都分配在同一個cell grid，YOLOv1是無法同時預測出兩物件與類別，只能從類別預測值中找出最高機率的類別，認定該grid cell中只有該類別

## ■ YOLOv2

- YOLOv2在保持處理速度的基礎上，預測更準確（**Better**），速度更快（**Faster**），識別物件更多（**Stronger**）
- 識別更多物件也擴展到能夠檢測9000種不同物件，稱之為**YOLO9000**

# YOLOV2

# YOLOv2 特點

## ■ Batch Normalization

- 使用Batch Normalization對網路進行優化，讓網路提高了收斂性，同時還消除了對其他形式的正則化（regularization）的依賴

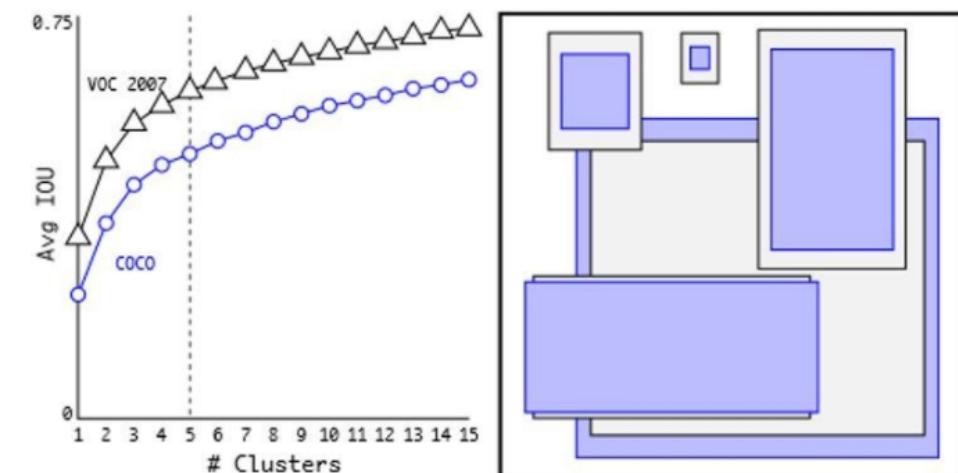
## ■ High resolution classifier

- YOLO v2的分類網路以448\*448的解析度在ImageNet上進行訓練，學習圖片特徵

# YOLOv2 特點

## ■ Convolution with anchor boxes

- YOLO一代包含有全連接層，能直接預測 Bounding Boxes的座標值。作者發現通過預測偏移量而不是座標值能夠簡化問題，讓神經網路學習起來更容易。所以去掉了全連接層，使用Anchor Boxes來預測 Bounding Boxes。作



## ■ Dimension clusters

- 之前Anchor Box的尺寸是手動選擇的，YOLOv2 在訓練集 ( training set ) Bounding Boxes上跑了一下k-means聚類，來找到一個比較好的尺寸值

Dimension clusters

# Anchor Box

- anchor box ( bounding box prior ) 是從訓練集的所有Ground Truth 中統計 ( 使用k-means ) 出來的在訓練集中，經常出現的幾個盒子形狀和尺寸。
- 或者，在某個訓練集中最經常出現的盒子形狀有扁長的，瘦高的和寬高比例差不多的正方形這三種形狀。我們可以預先將這些統計上的加入到模型中，有助於模型快速收斂了

# YOLOv2 特點

## ■ Fine-Grained Features

- 使用Passthrough Layer把高解析度特徵與低解析度特徵聯繫在一起，辨別不同解析度的圖片物件

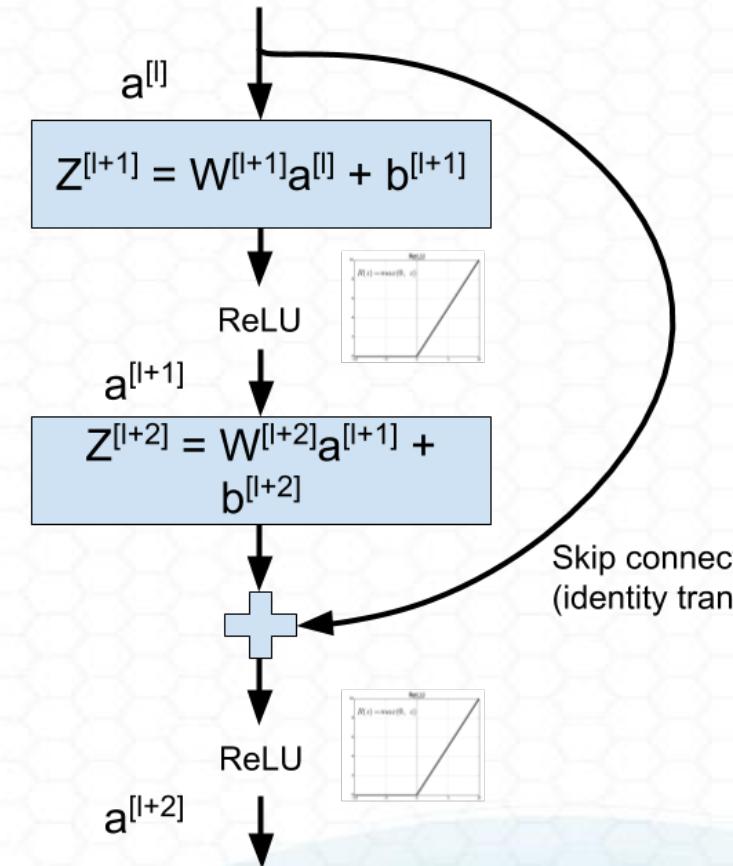
## ■ Multi-Scale Training

- YOLO v2每迭代幾次都會改變網絡參數。每10個Batch，網絡會隨機地選擇一個新的圖片尺寸，由於使用了下采樣參數是32，所以不同的尺寸大小也選擇為32的倍數，最小 $320*320$ ，最大 $608*608$ ，網絡會自動改變尺寸，繼續訓練的過程

# YOLOV3

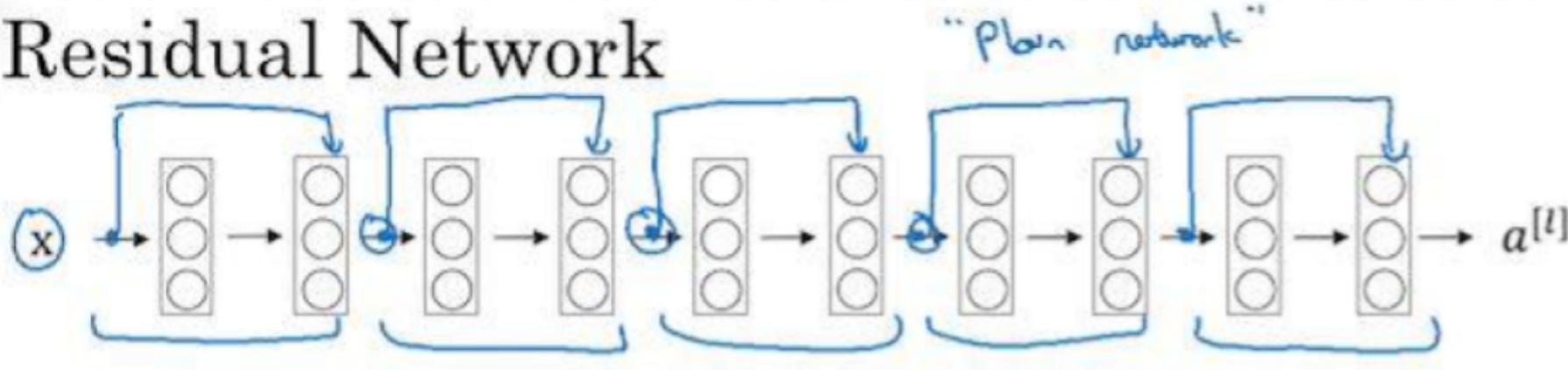
# Yolov3

- Residual Block 解決了當逐漸加深網路時，gradient 不能回流下一層的問題
- 透過 skip connection，而使上層作 back-propagation 時，如果此層的參數逼近於零時，仍可以選擇捷徑的路徑，跳過此層，直接回流至來源層

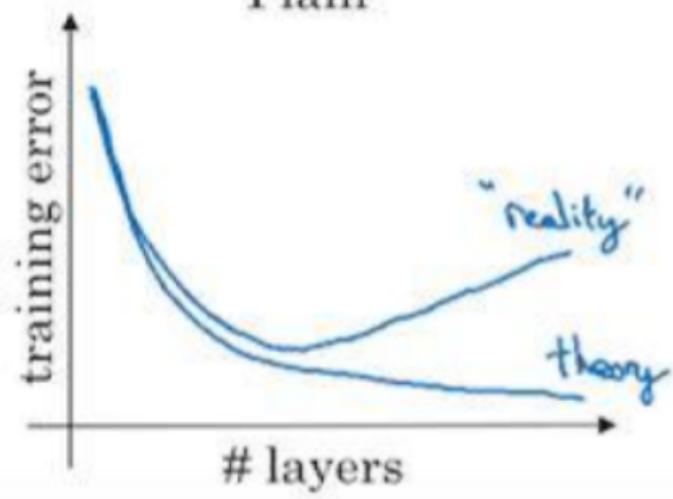


# Residual Block

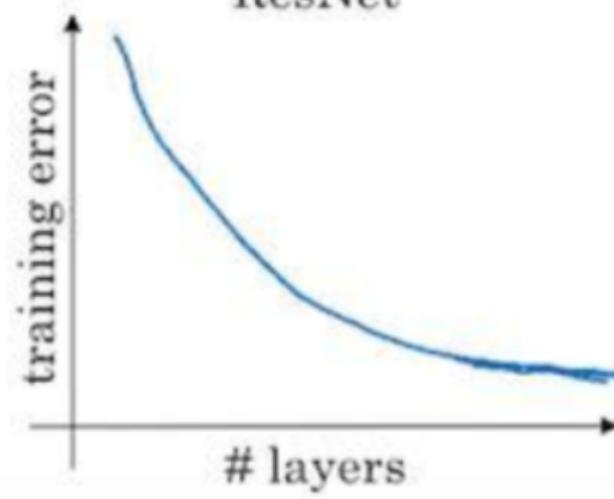
## Residual Network



Plain

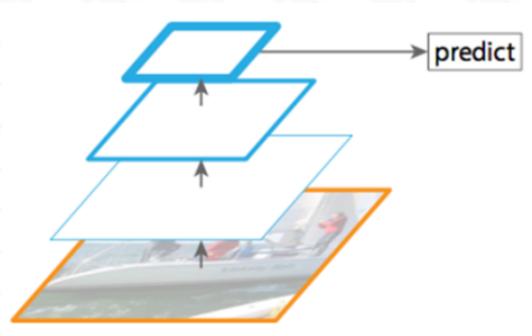


ResNet

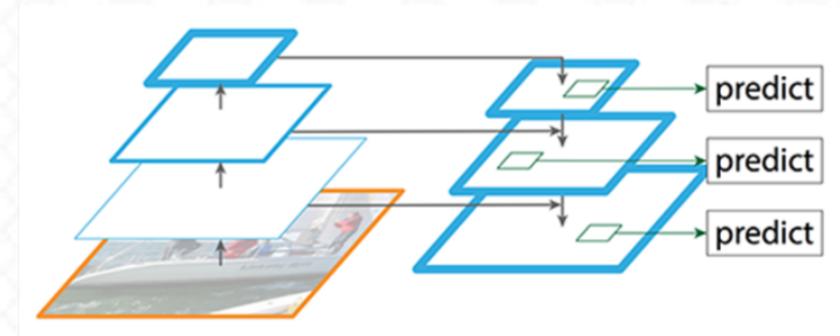


# Feature Pyramid network

- 傳統的方法是single feature map，圖片經由卷積層後做單一的輸出
- Feature pyramid network (FPN)是每層尺度的預測都因為綜合了不同尺度的特徵圖資訊。使得在做預測時，對於不同尺度的圖片語意與空間特徵擷取量較足夠，因而得出更好的預測結果



Single feature map



Feature Pyramid Network

- YOLOv3對於anchor box 的數量也從YOLOv2的5個增加到了9個，在最後的三個不同尺度輸出層分別用3個anchor box做預測，以增加預測準確率。



# **THANK YOU**