本課程為教育部109年度智慧聯網技術課程推廣計畫之補助課程
教材內容也自教育部智慧聯網技術重點模組(人工智慧視覺感知運算系統模組)教材改編

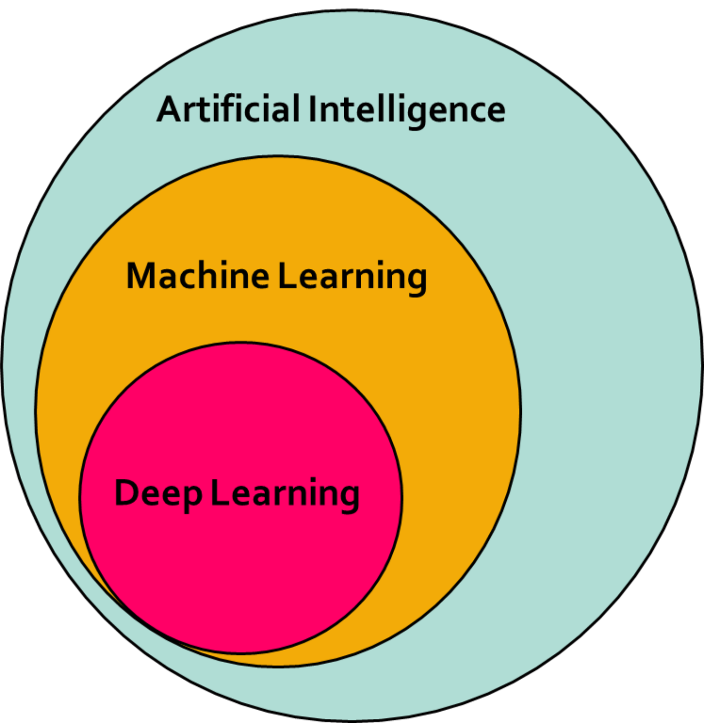
人工智慧視覺運算方法

謝東佑
可測及可靠系統實驗室
(Testable And Reliable Systems Lab., TARS)
國立中山大學電機系

Office: 工EC-7038
07-5252000 Ext. 4114
tyhsieh@mail.ee.nsysu.edu.tw

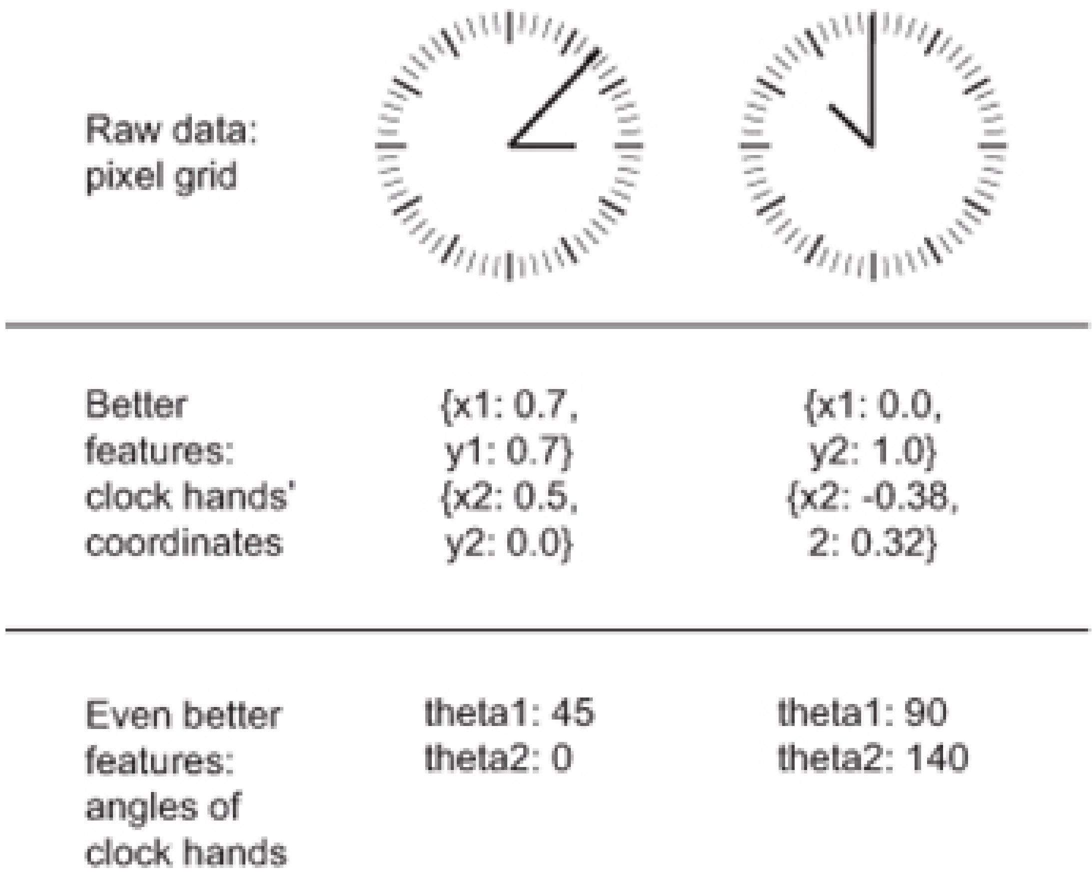
TARS
Keep feet on the ground

AI vs Machine Learning vs Deep Learning



- AI: 模擬人類智慧
  - 結果有智慧就算
  - 一個擁有非常詳盡的 rule-based 系統也可以是 AI
- Machine learning是達成 AI 的一種方法
  - 從資料當中學習出 rules
  - 找到一個夠好的 function 能解決特定的問題
- Deep learning 是machine learning的一種
  - 從feature engineering 走向architecture engineering
  - 不再人工萃取特徵
  - 深層網路萃取更抽象特徵

Deep Learning v.s. Feature Engineering



- 讓機器看時鐘報時
- 直接看圖
  - 要用CNN才行
  - 需要大量資料
- 放點工人智慧
  - 用指針座標
  - 簡單的ML就可以
  - 少量資料就可以
- 更多工人智慧
  - 用指針角度(像人看時鐘一樣)
  - 連ML都不用，查表就可以
  - 資料最少

對DL來講，好的特徵可以幫助你用較少資源與資料，反過來，若你的資料資源很少，你會需要比較好的特徵(aka.更多工人智慧)

影像處理 (Image Processing)

- 改變影像內容/本質，以方便
  - 人眼辨識
  - 機器辨識

加強影線的邊緣線條，呈現更銳利的影像。見圖1.1

圖1.1 影像銳利化 (a) 原始影像 (b) 銳利化結果

讓人看得更清晰

去除影像的雜訊。見圖1.2

圖1.2 去除影像雜訊 (a) 原始影像 (b) 去除雜訊結果

THOMSON

讓人看得更清晰

去除影像的動態模糊現象。見圖1.3

圖1.3 去除影像模糊現象 (a) 原始影像 (b) 去除模糊現象結果

THOMSON

讓機器方便看 (取得特徵)

取得影線邊緣線條，這個動作是為了測量影像中的物體。見圖1.4 (a與b)

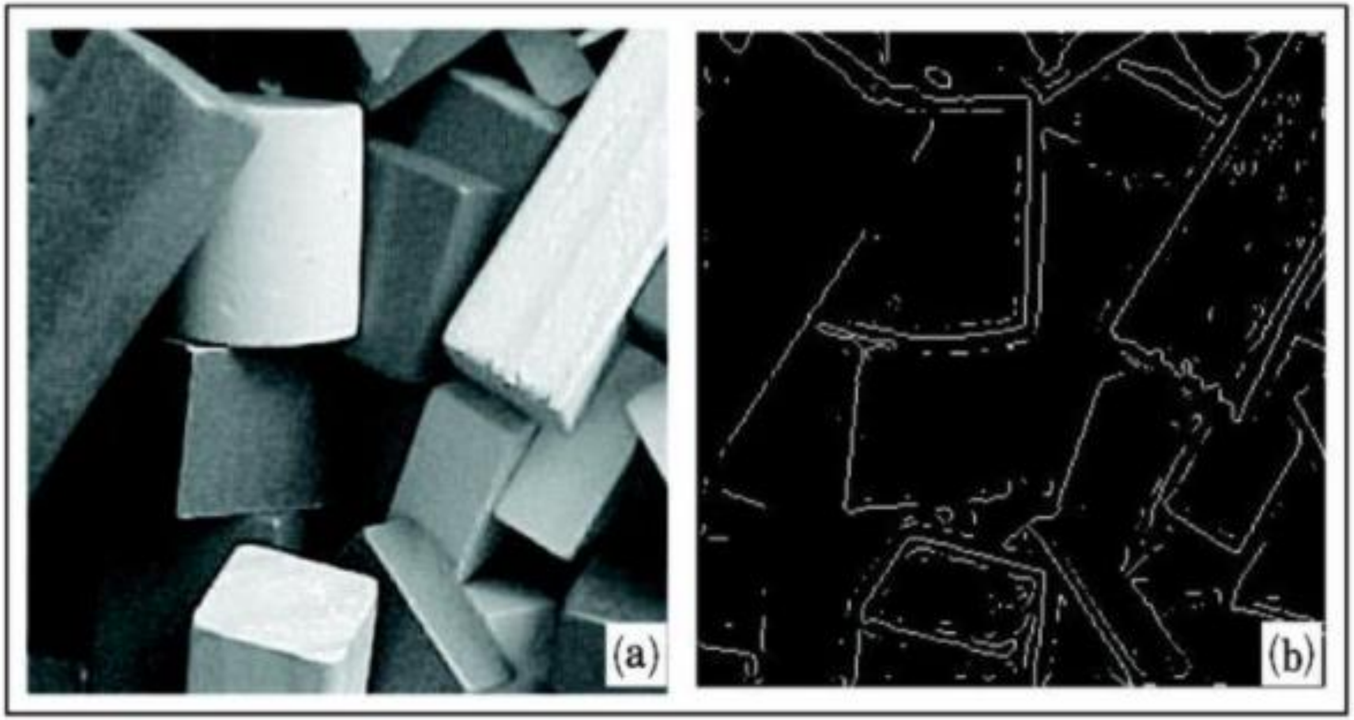
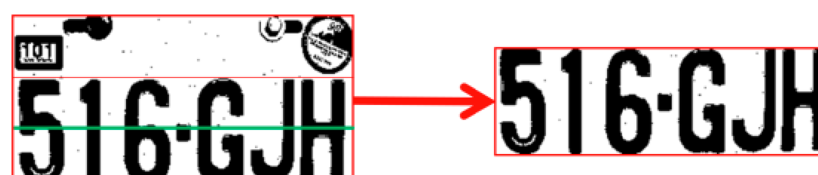


圖1.4 取得影像邊緣線條 (a) 原始影像 (b) 物體邊緣線條

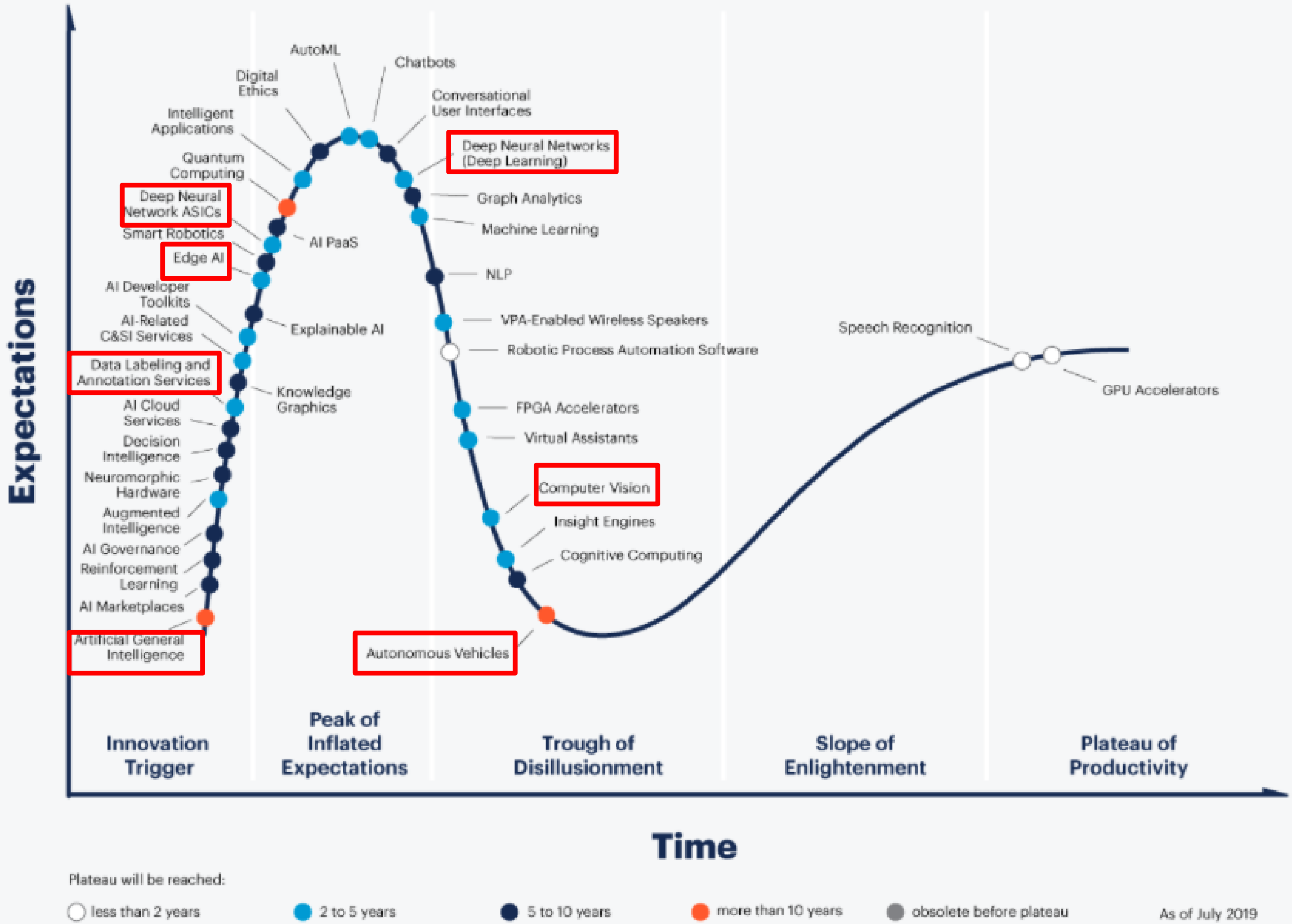
車牌辨識

Source: http://www.csie.chu.edu.tw/ezfiles/11/1011/bbs/22/bbs_119_1038141_85574.pdf

相機或手機的影像 → 將影像縮圖至 320x240 → 將影像轉為灰階影像 → 邊緣偵測(Sobel) → 灰階、Sobel 做二值化 → Sobel 影像做中值濾波 → 車牌定位 → 字元切割 → 字元辨識 → 顯示車牌號碼



Gartner Hype Cycle for Artificial Intelligence, 2019



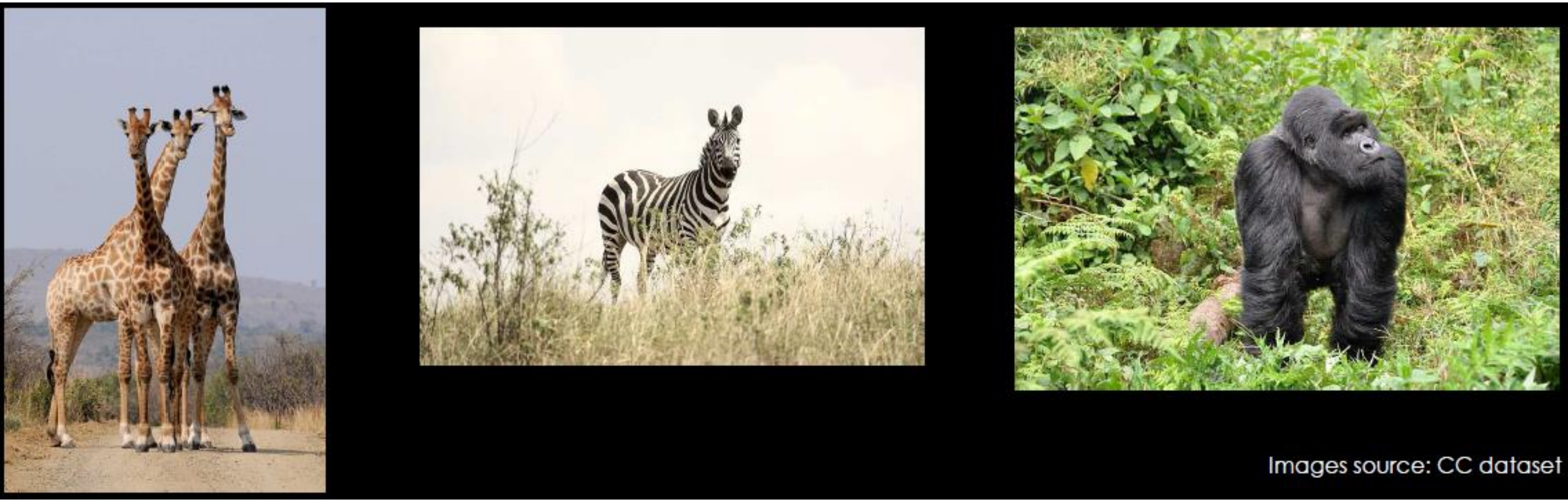
gartner.com/SmarterWithGartner

Source: Gartner

AI VISUAL ALGORITHMS

How Does A Computer Classify Pictures?

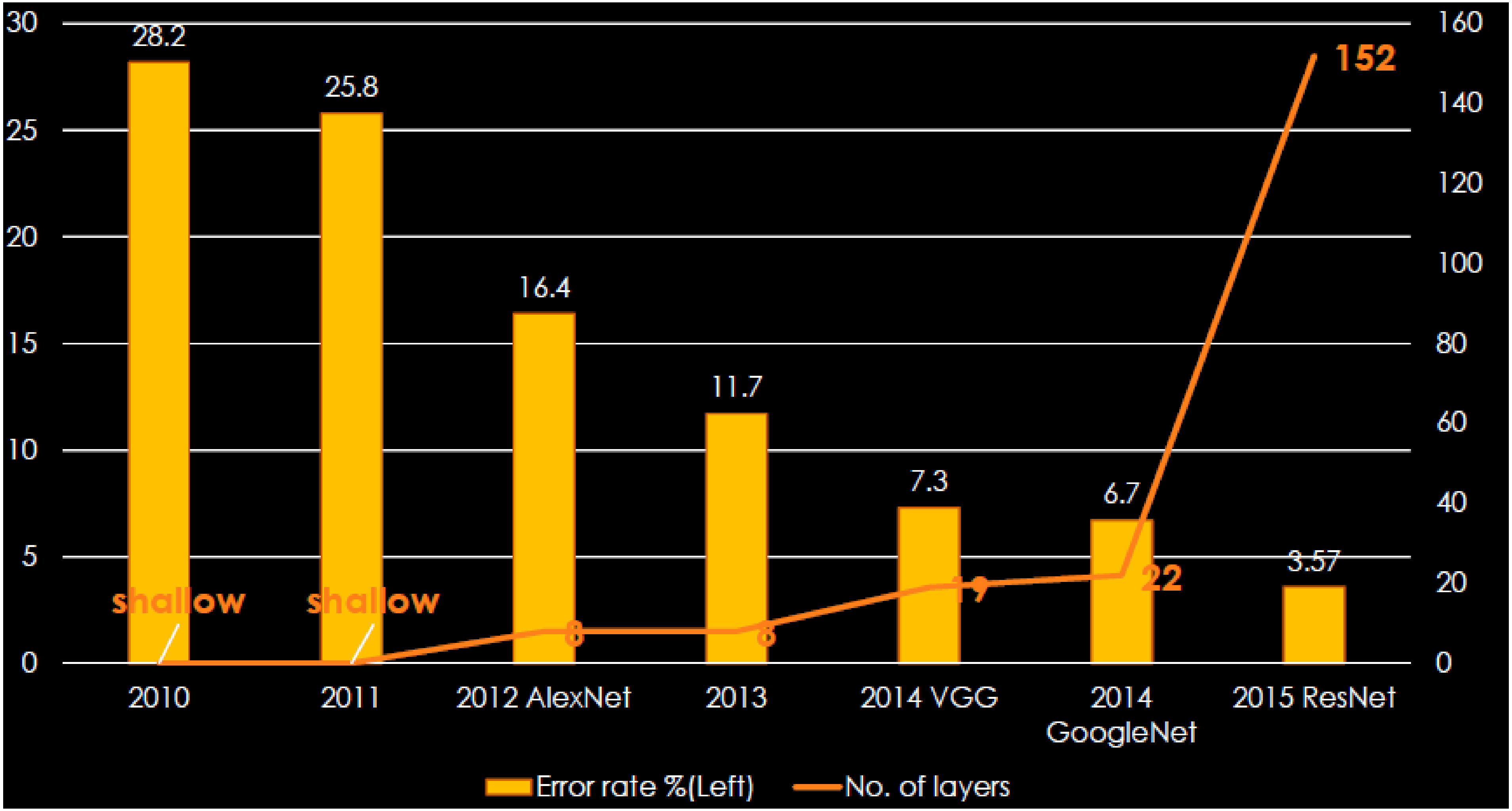
- **A picture is only a group of pixels for a computer.**
- **Modern AI nets learn features of objects.**



Object Classification

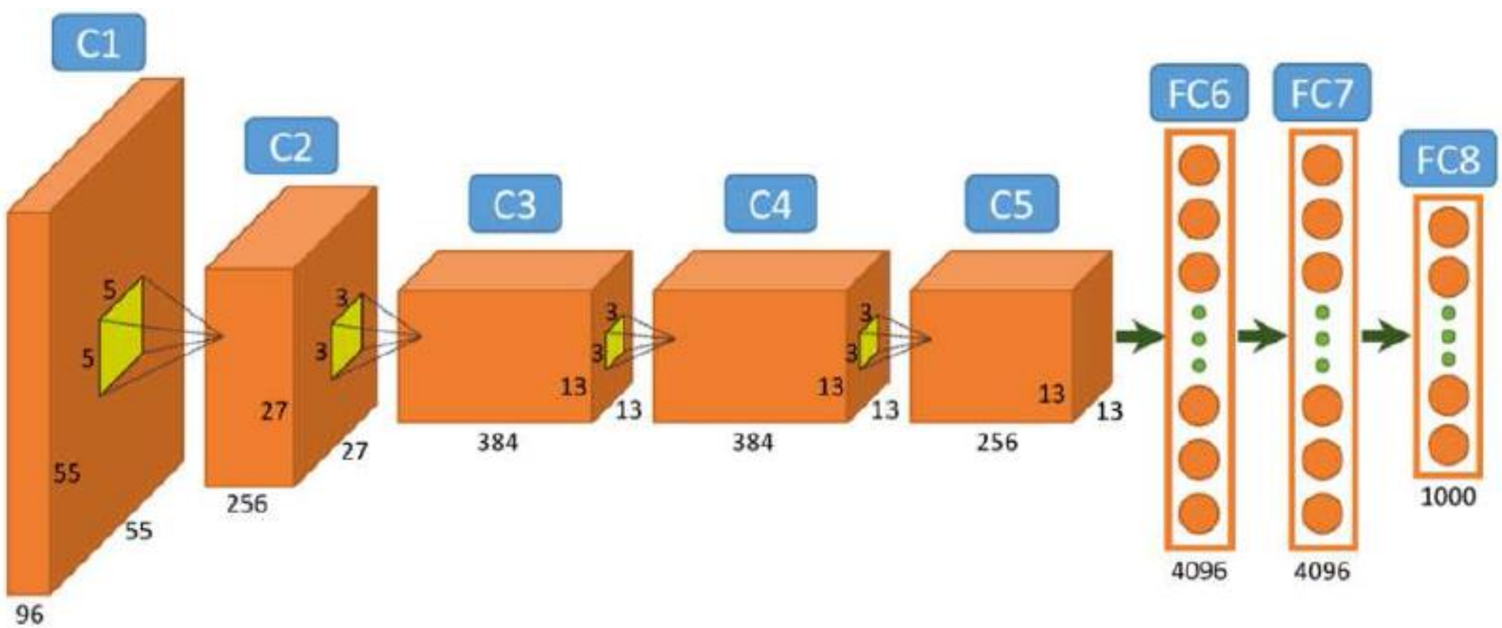
- Modern AI algorithms for object classification
  - AlexNet, 5 CNN layers and 3 FC layers, 2012
  - VGG, 16 CNN layers and 3 FC layers, 2014
  - GoogLenet, 21 CNN layers and 1 FC layer, 2014
  - ResNet, 151 CNN layers and 1 FC layer, 2015
- Foundation of object detection
- Limitation
  - One object in one picture, no localization

ILSVRC(IMAGENET Large Scale Visual Recognition Competition)



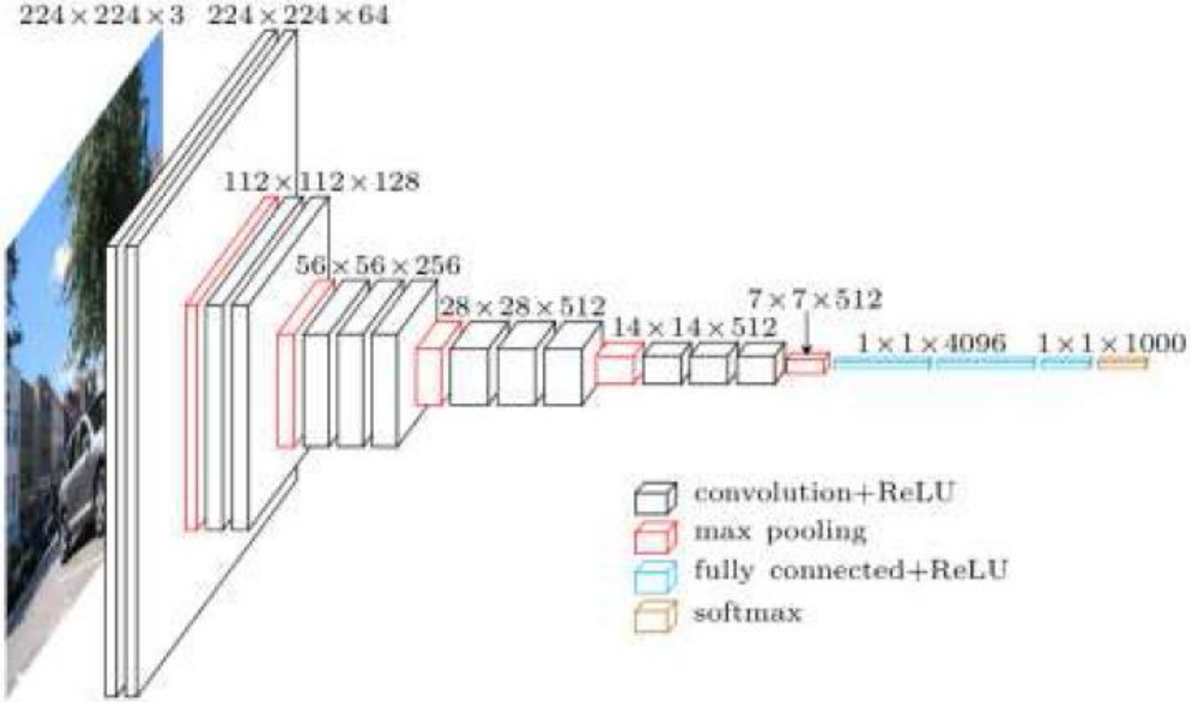
AlexNet

- CONV Layers: 5
- Fully Connected Layers: 3
- Weights: 61M
- MACs: 724M



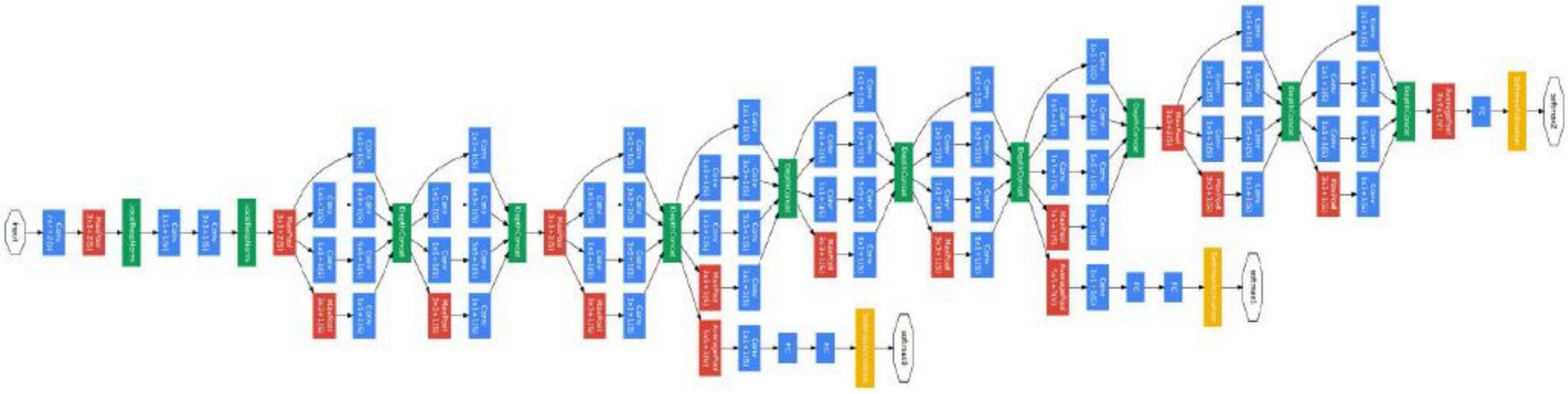
VGG

- **CONV Layers: 16**
- **Fully Connected Layers: 3**
- **Weights: 138M**
- **MACs: 15.5G**



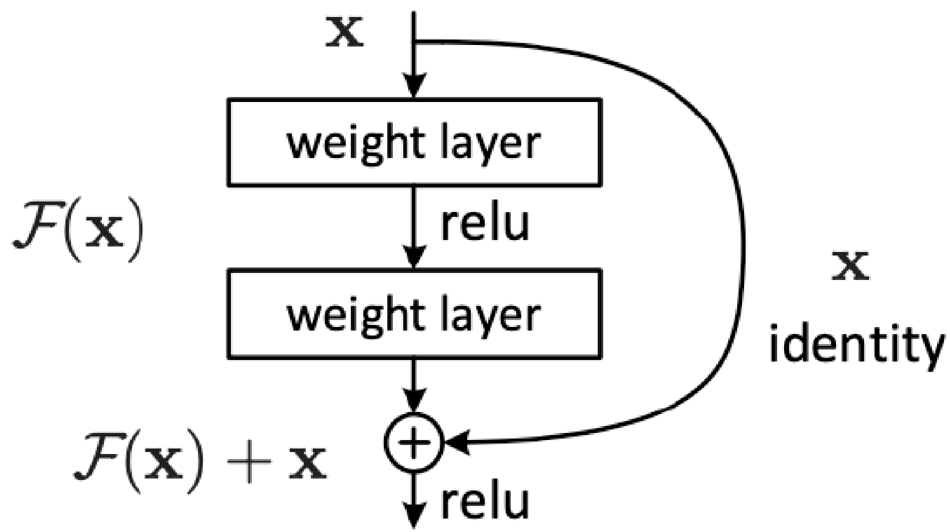
GoogLenet

- **CONV Layers: 21**
- **Fully Connected Layers: 1**
- **Weights: 7.0M**
- **MACs: 1.43G**

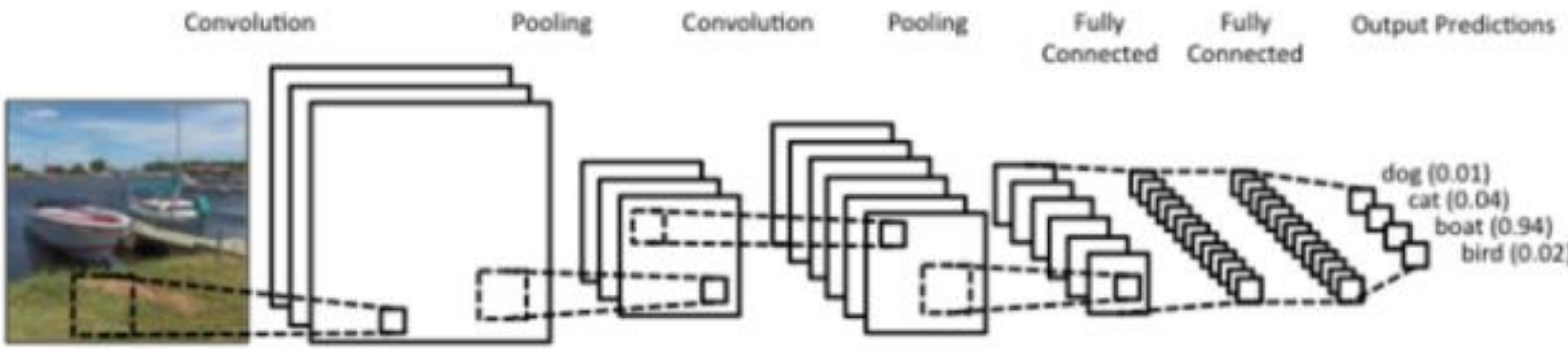


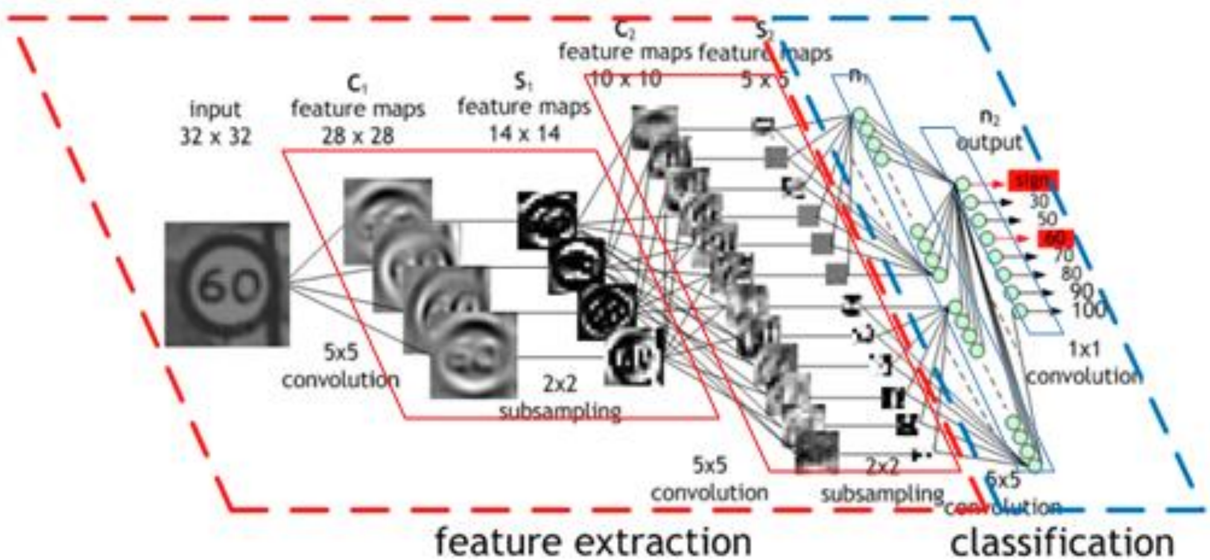
ResNet

- **Main idea**
  - **Residual layer**
- **CONV Layers: 151**
- **Fully Connected Layers: 1**
- **Weights: 25.5M**
- **MACs: 3.9G**



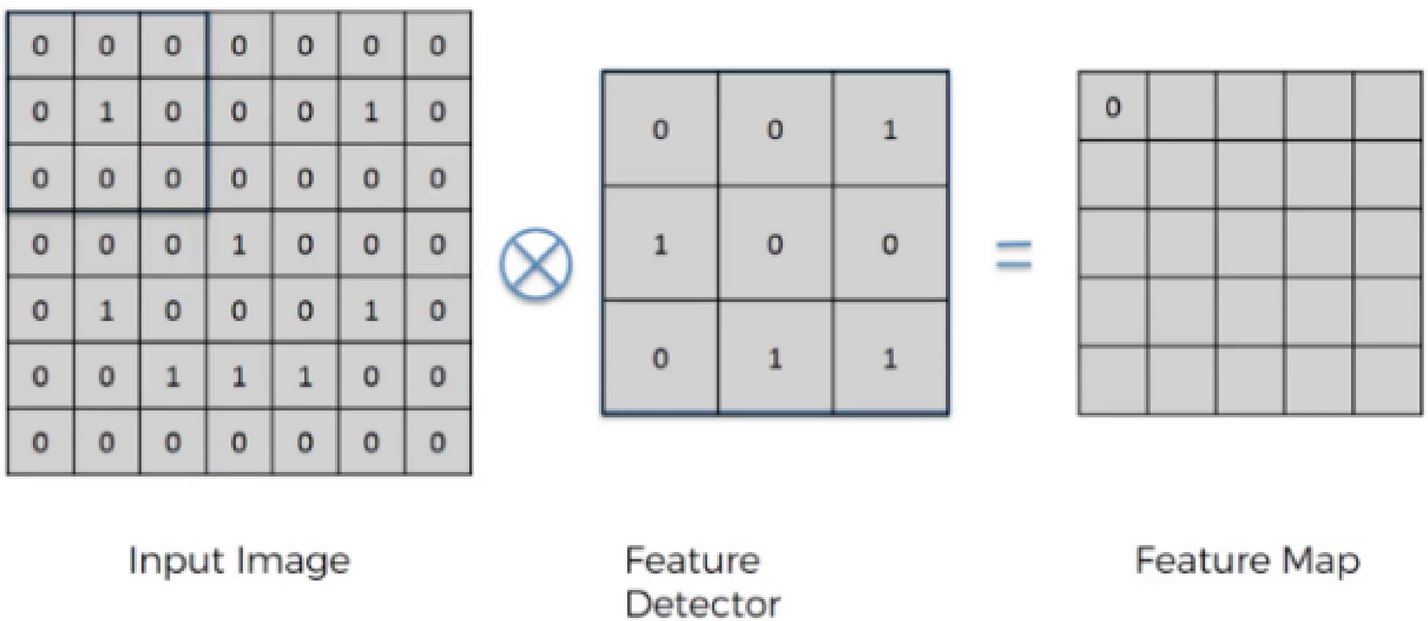
Idea of CNN (Convolutional Neural Network)





Source: https://medium.com/jameslearningnote/資料分析-機器學習-第5-1講-捲積神經網絡介紹-convolutional-neural-network-4f8249d65d4f

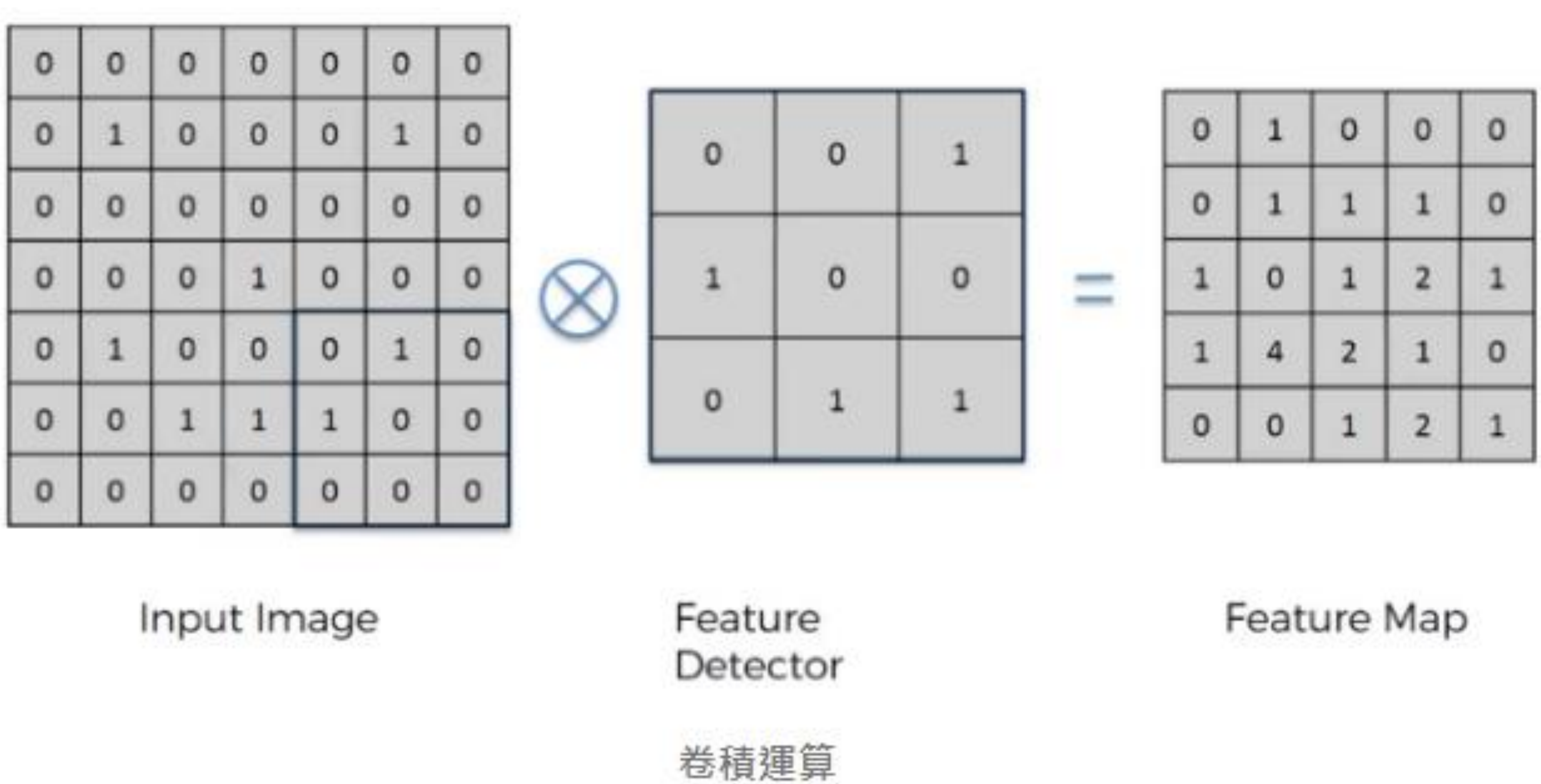
Convolution Layer



卷積運算

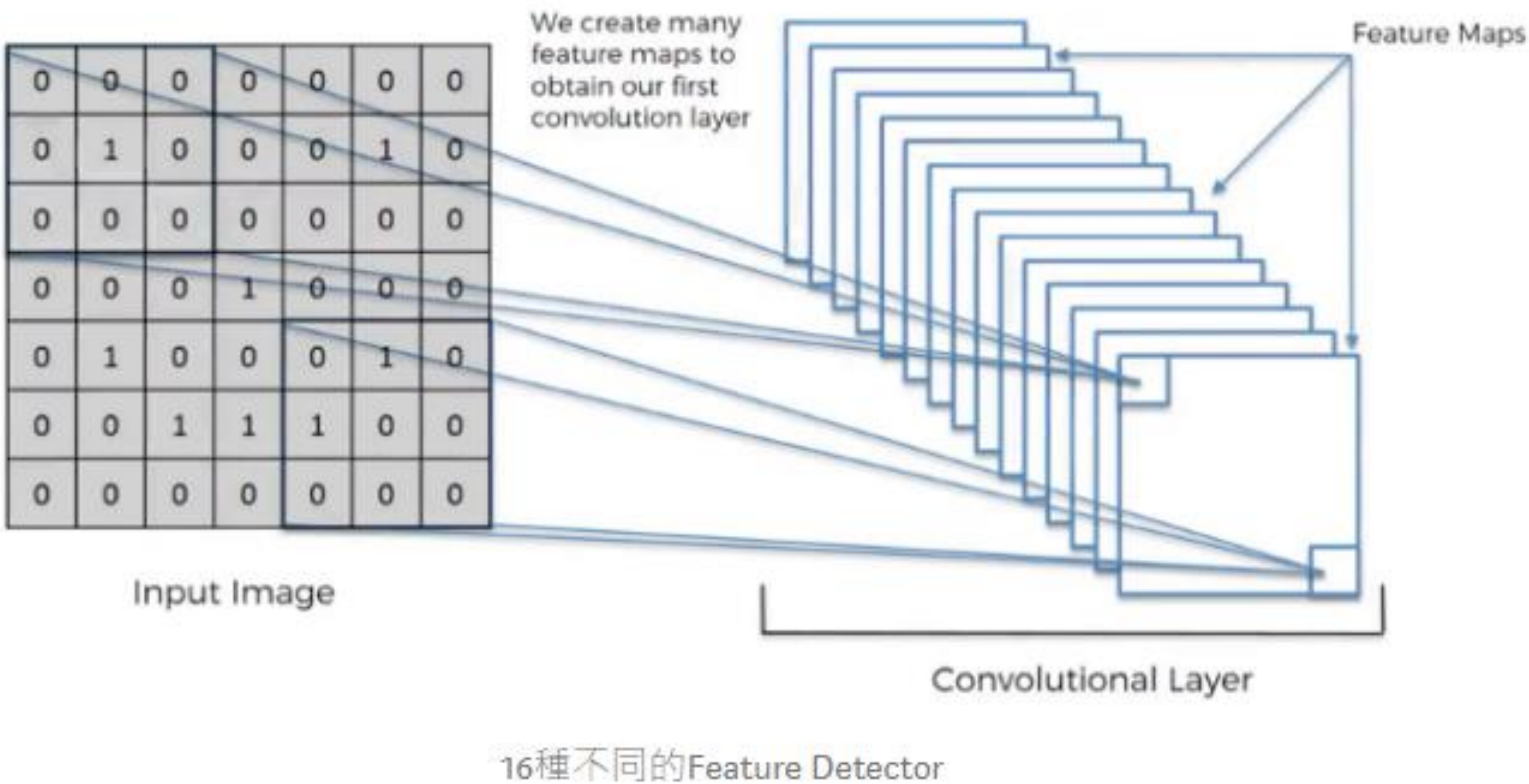
Source: https://medium.com/jameslearningnote/資料分析-機器學習-第5-1講-捲積神經網絡介紹-convolutional-neural-network-4f8249d65d4f

Convolution Layer



Source: https://medium.com/jameslearningnote/資料分析-機器學習-第5-1講-捲積神經網絡介紹-convolutional-neural-network-4f8249d65d4f

Convolution Layer



Source: https://medium.com/jameslearningnote/資料分析-機器學習-第5-1講-捲積神經網絡介紹-convolutional-neural-network-4f8249d65d4f

Convolution Layer

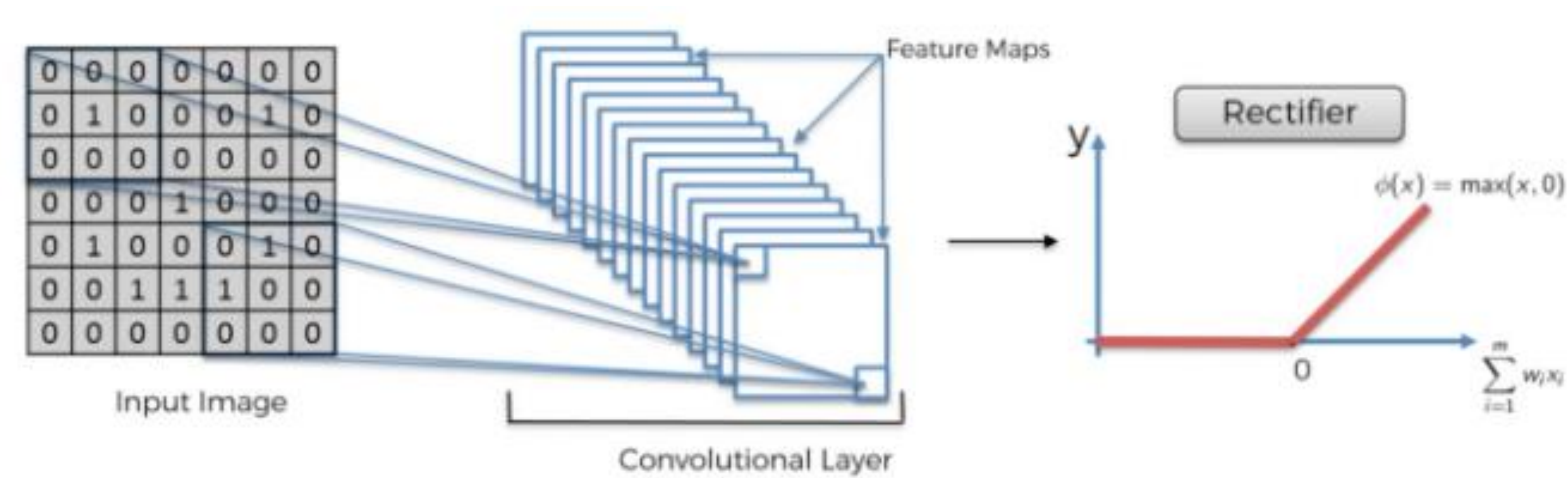
∗

1	0	-1
2	0	-2
1	0	-1

利用Feature Detector萃取出物體的邊界

Source: https://medium.com/jameslearningnote/資料分析-機器學習-第5-1講-捲積神經網絡介紹-convolutional-neural-network-4f8249d65d4f

使用Relu函數去掉負值，更能淬煉出物體的形狀



Source: https://medium.com/jameslearningnote/資料分析-機器學習-第5-1講-捲積神經網絡介紹-convolutional-neural-network-4f8249d65d4f

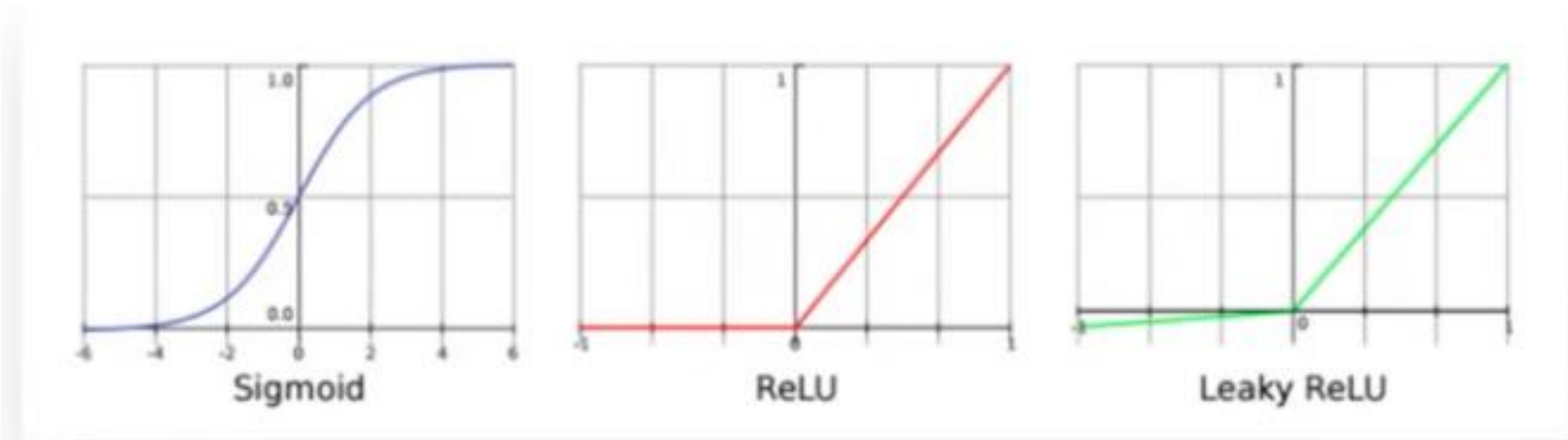
使用Relu函數去掉負值，更能淬煉出物體的形狀

Source: https://medium.com/jameslearningnote/資料分析-機器學習-第5-1講-捲積神經網絡介紹-convolutional-neural-network-4f8249d65d4f

使用Relu函數去掉負值，更能淬煉出物體的形狀

Source: https://medium.com/jameslearningnote/資料分析-機器學習-第5-1講-捲積神經網絡介紹-convolutional-neural-network-4f8249d65d4f

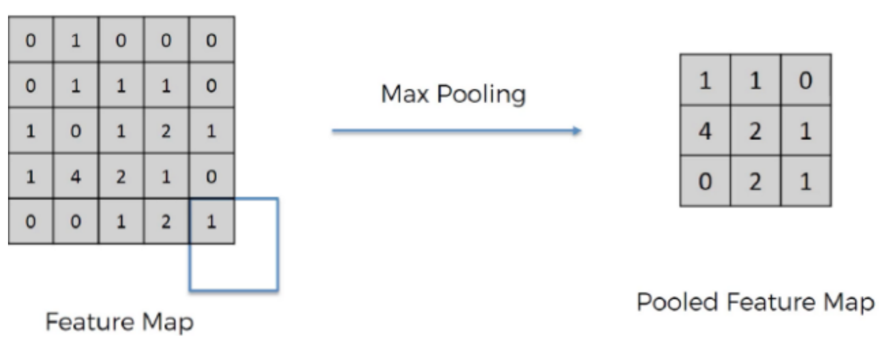
其他函數



Source: https://medium.com/jameslearningnote/資料分析-機器學習-第5-1講-捲積神經網絡介紹-convolutional-neural-network-4f8249d65d4f

Pooling Layer 池化層

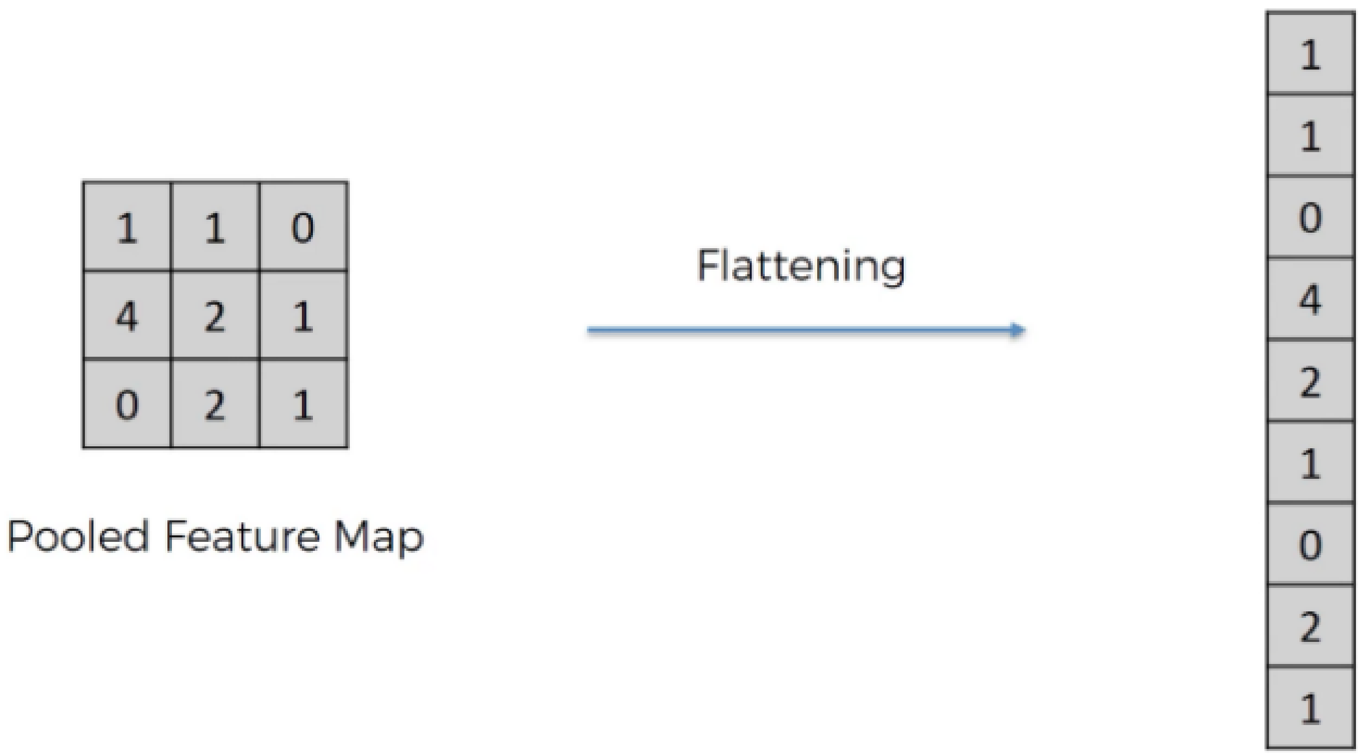
- **Max Pooling**
- 當圖片整個平移幾個Pixel的話對判斷上完全不會造成影響，以及有很好的抗雜訊功能



Source: https://medium.com/jameslearningnote/資料分析-機器學習-第5-1講-捲積神經網絡介紹-convolutional-neural-network-4f8249d65d4f

Fully Connected Layer 全連接層

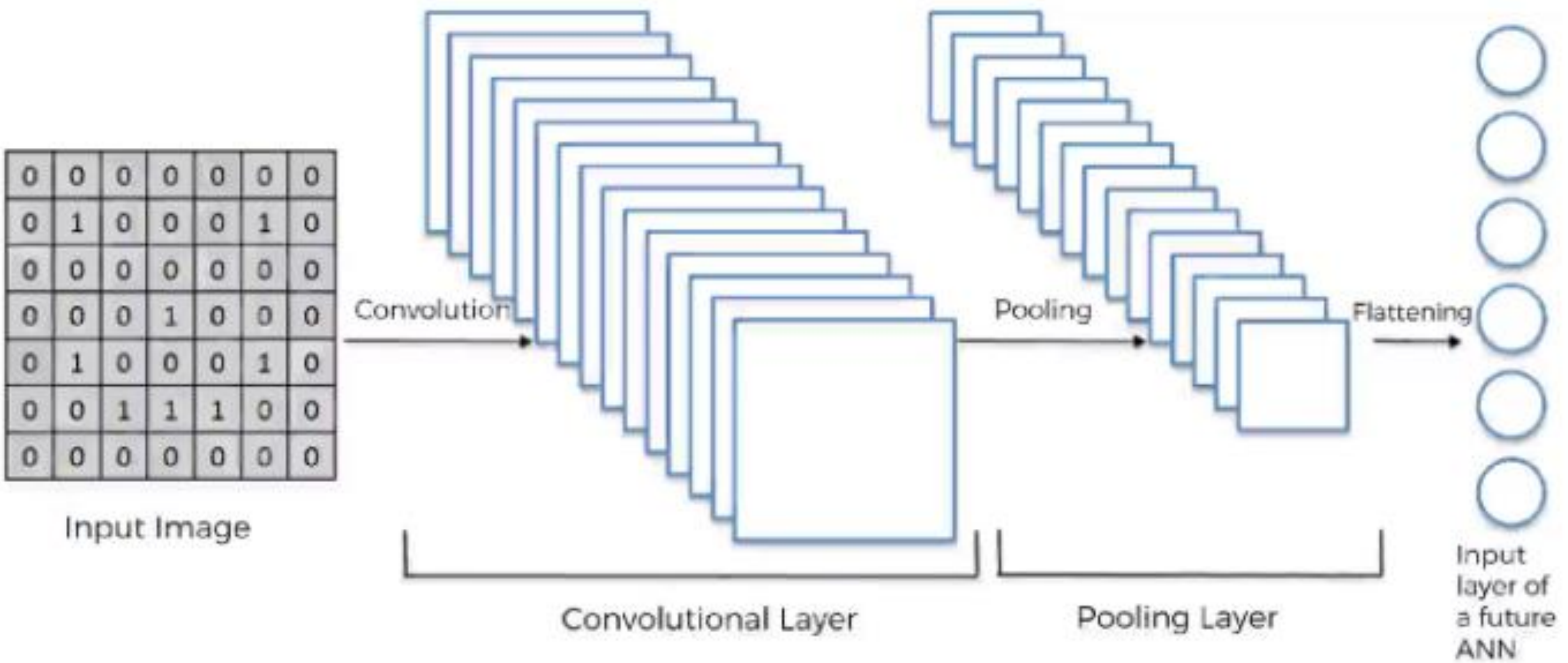
- 將之前的結果平坦化之後接到最基本的神經網絡



Source: https://medium.com/jameslearningnote/資料分析-機器學習-第5-1講-捲積神經網絡介紹-convolutional-neural-network-4f8249d65d4f

Fully Connected Layer 全連接層

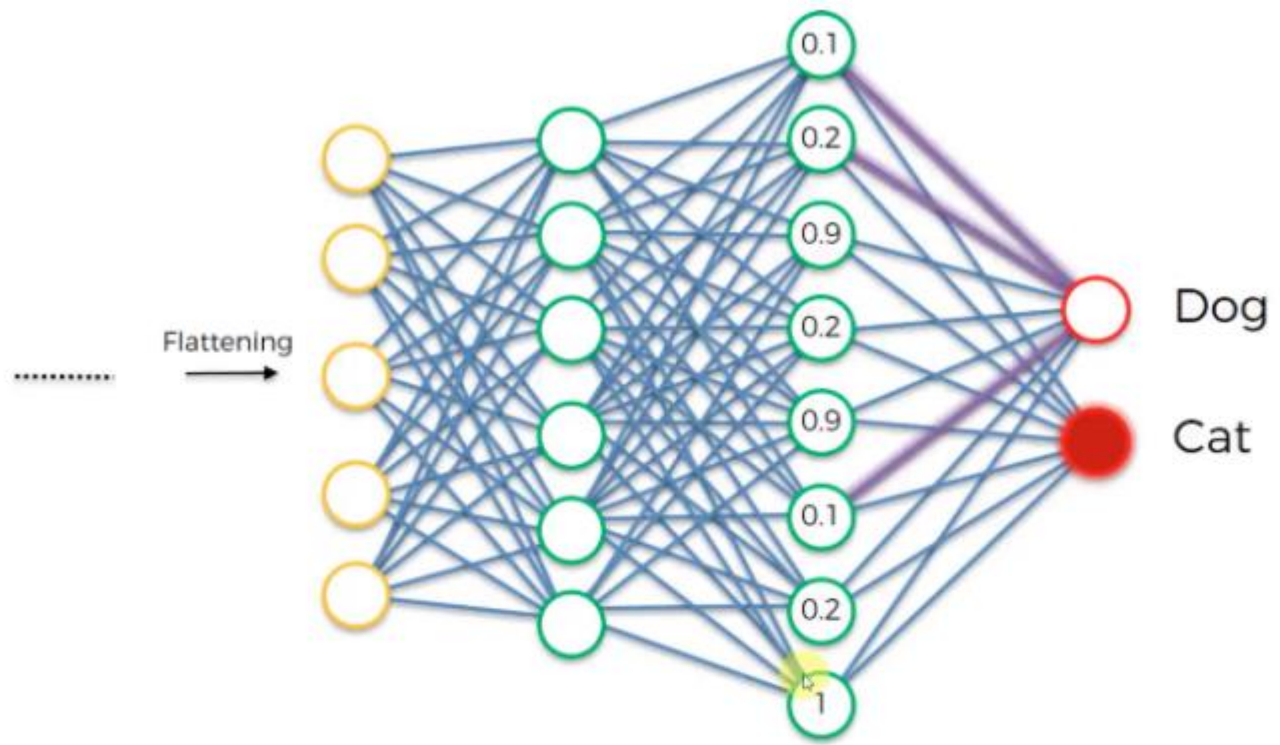
- 將之前的結果平坦化之後接到最基本的神經網絡



Source: https://medium.com/jameslearningnote/資料分析-機器學習-第5-1講-捲積神經網絡介紹-convolutional-neural-network-4f8249d65d4f

Fully Connected Layer 全連接層

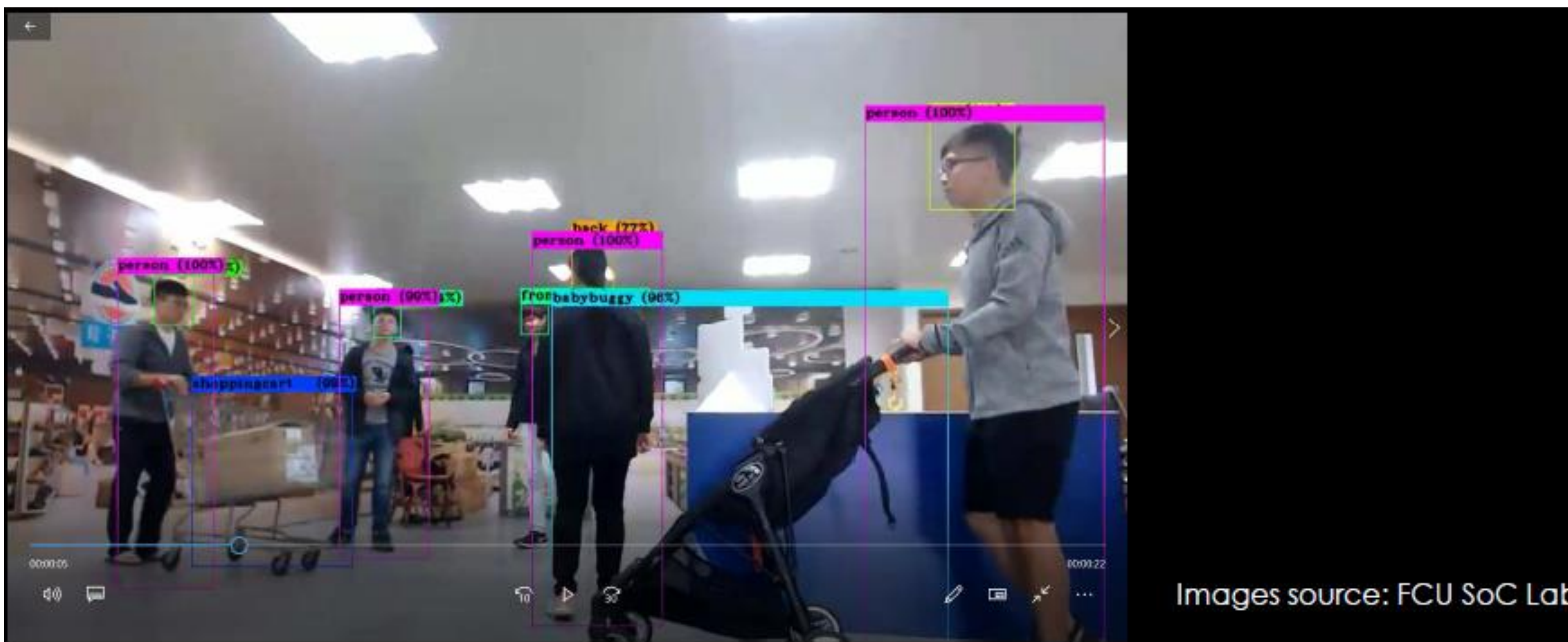
- 將之前的結果平坦化之後接到最基本的神經網絡



Source: https://medium.com/jameslearningnote/資料分析-機器學習-第5-1講-捲積神經網絡介紹-convolutional-neural-network-4f8249d65d4f

Objection Localization

- Besides class, the computer needs to know the location of each object.

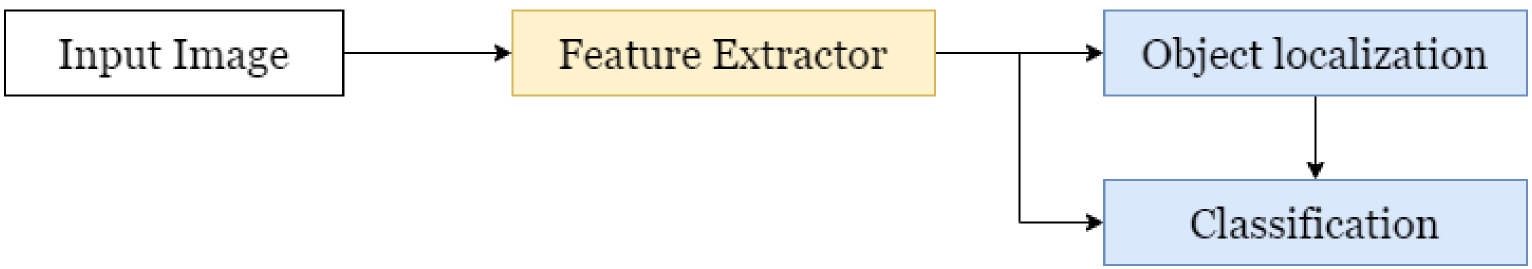


Modern AI Algorithms for Object Detection

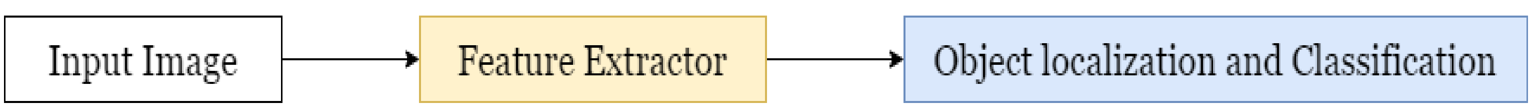
- **RCNN (Region-based CNN), fast RCNN, faster RCNN**
- **YOLO (You Only Look Once)**
- **SSD (Single Shot Detection)**

Object Detection

- Two-stage object detection
  - Good detection accuracy but slow operation
  - Ex: Faster R-CNN

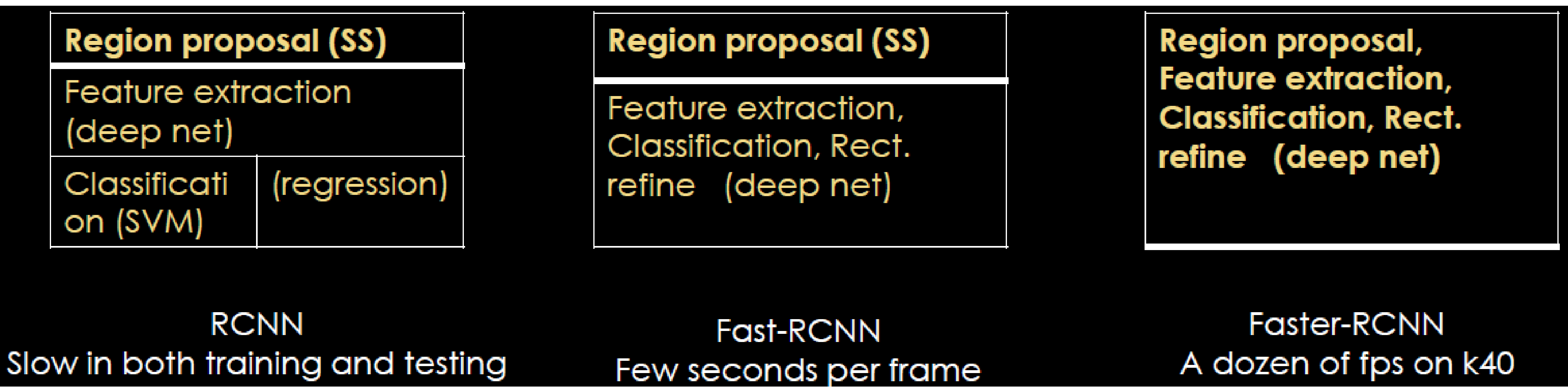


- One-stage object detection
  - Fast operation and acceptable detection accuracy
  - Ex: SSD, YOLO

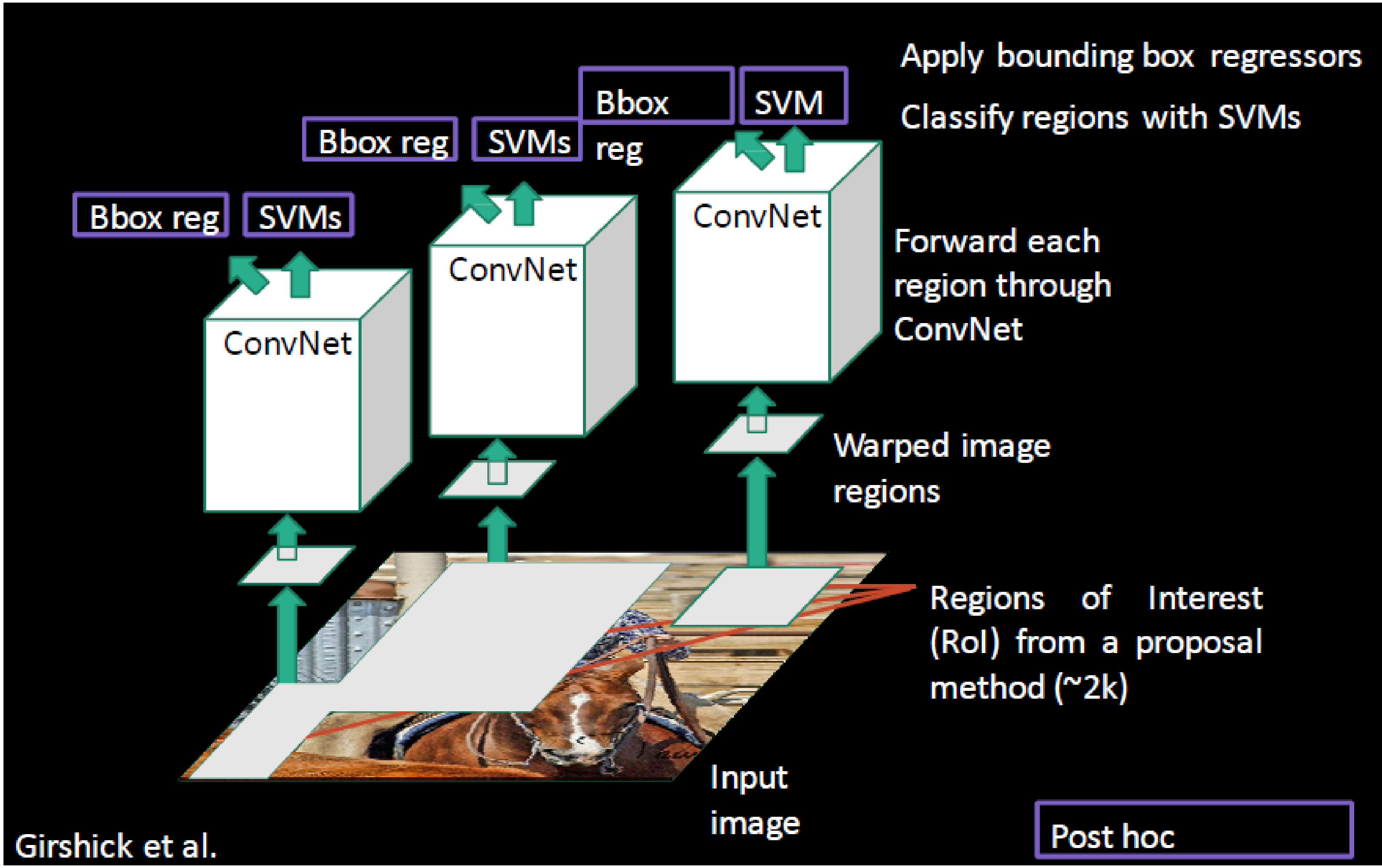


RCNN (Region-Based CNN), Fast RCNN, Faster RCNN

- **Two-stage ways**



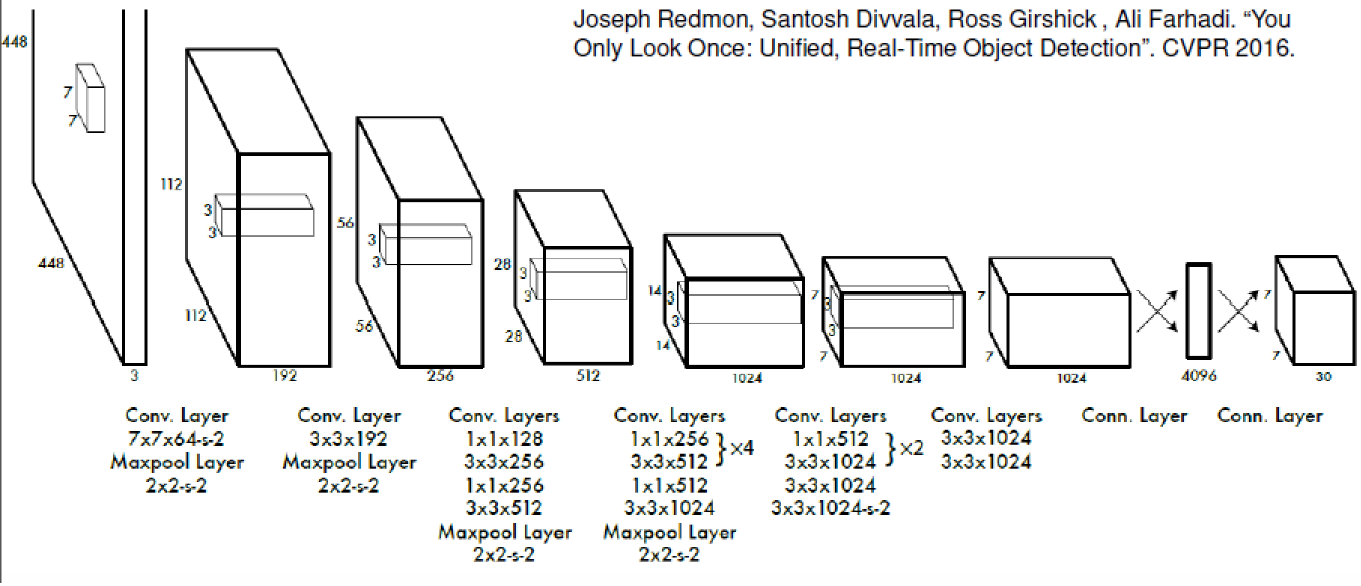
Two-Stage Ways

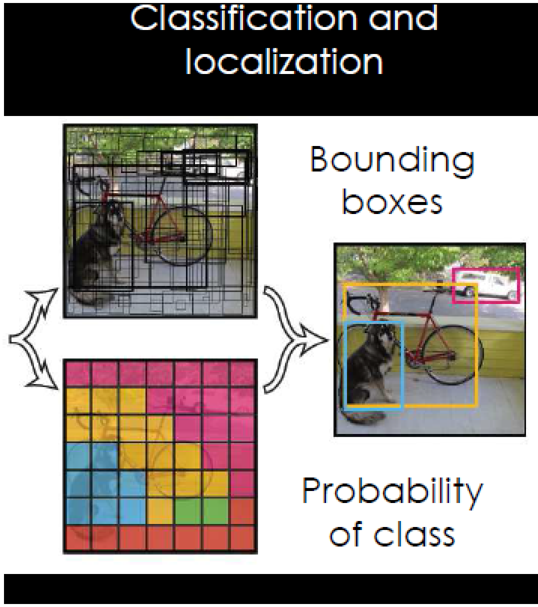


YOLO (YOU ONLY LOOK ONCE)

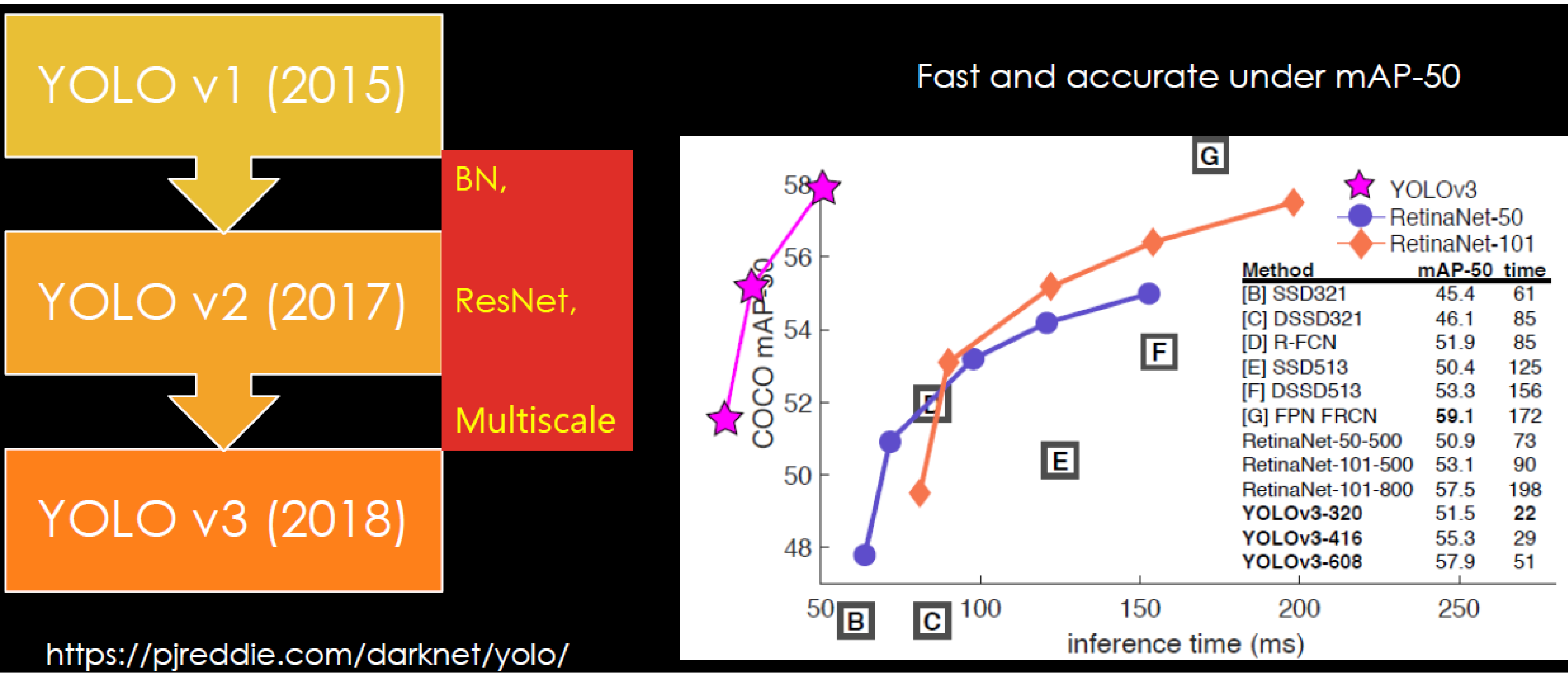
- **One-stage way**

Joseph Redmon, Santosh Divvala, Ross Girshick , Ali Farhadi. “You Only Look Once: Unified, Real-Time Object Detection”. CVPR 2016.





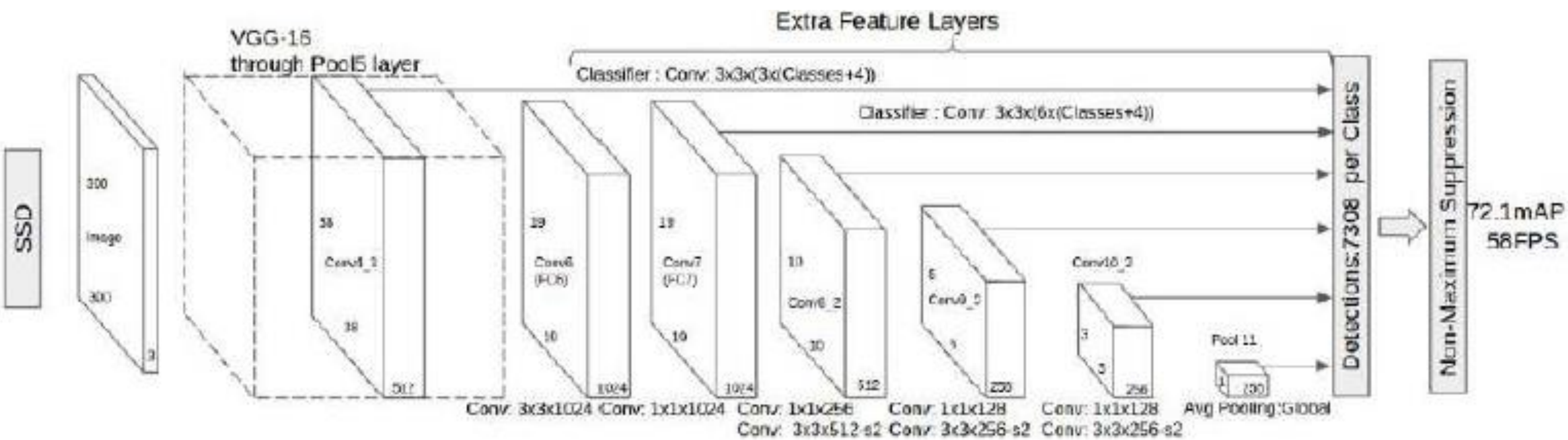
YOLO V3



YOLO v1 (2015)
YOLO v2 (2017)
YOLO v3 (2018)
BN,
ResNet,
Multiscale
Fast and accurate under mAP-50
YOLOv3
RetinaNet-50
RetinaNet-101
Method mAP-50 time
[B] SSD321 45.4 61
[C] DSSD321 46.1 85
[D] R-FCN 51.9 85
[E] SSD513 50.4 125
[F] DSSD513 53.3 156
[G] FPN FRCN 59.1 172
RetinaNet-50-500 50.9 73
RetinaNet-101-500 53.1 90
RetinaNet-101-800 57.5 198
YOLOv3-320 51.5 22
YOLOv3-416 55.3 29
YOLOv3-608 57.9 51
COCO mAP-50
inference time (ms)
https://pjreddie.com/darknet/yolo/

SSD (Single Shot Detection)

- **+Multi-scale feature maps**
- **- FC layers**

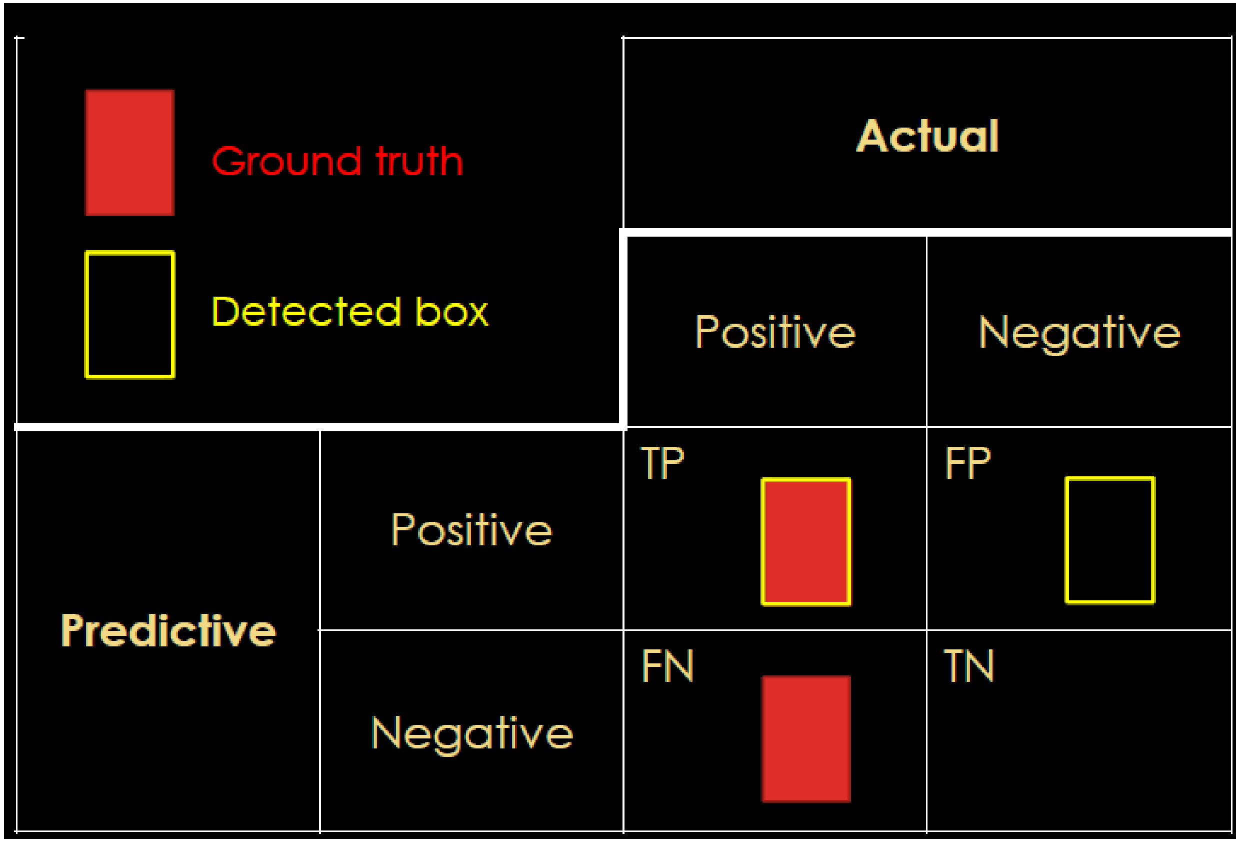


Performance Evaluation Indexes

- TP, FP, TN, FN
- Precision, Recall
- mAP (mean Average Precision)

TP, FP, TN, FN

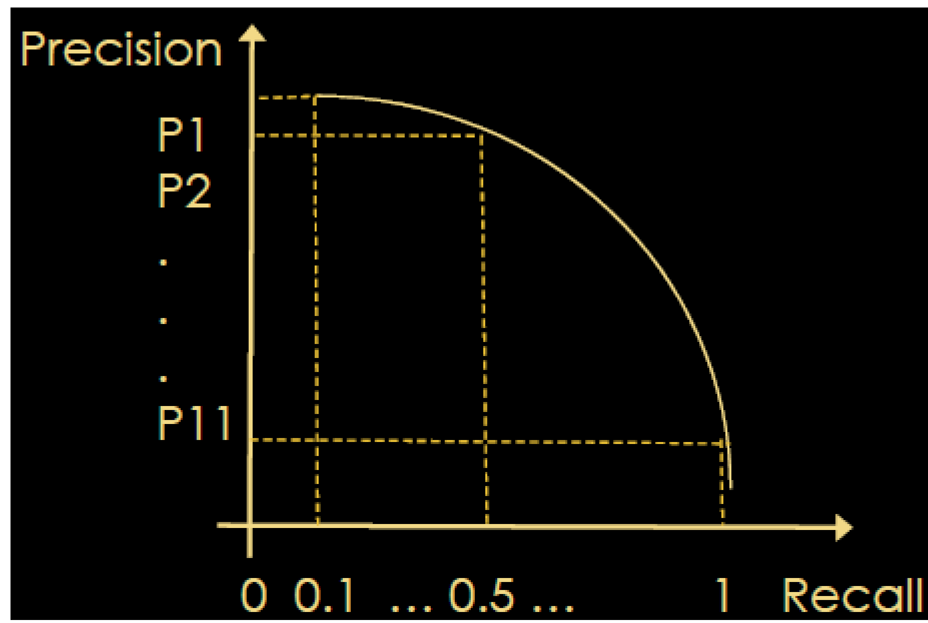
- TP: True Positive
- FP: False Positive
- TN: True Negative
- FN: False Negative



Precision, Recall

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

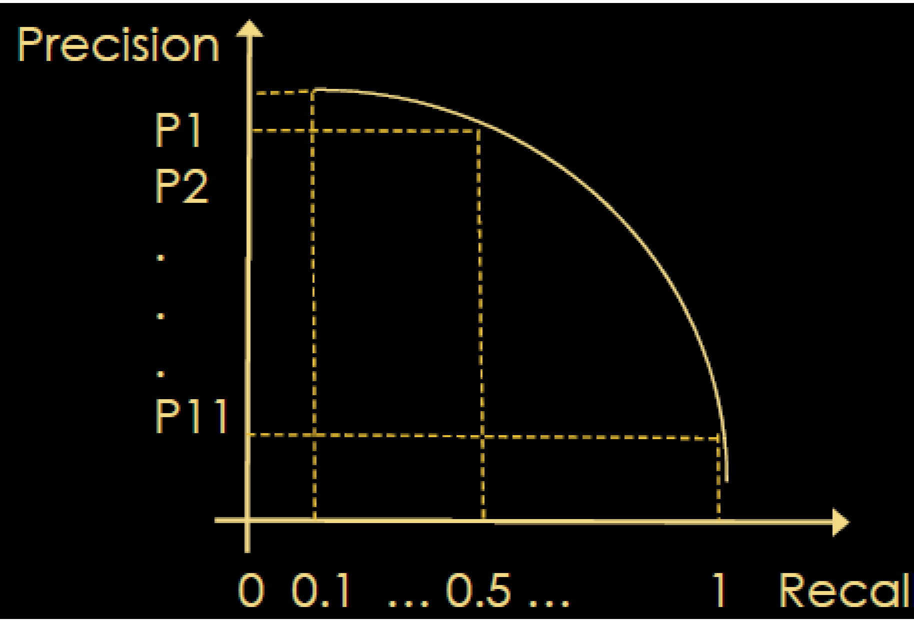


mAP (mean Average Precision)

- **AP: the average precision of precisions of different recalls**
- **mAP: the mean of APs of different kinds of objects**

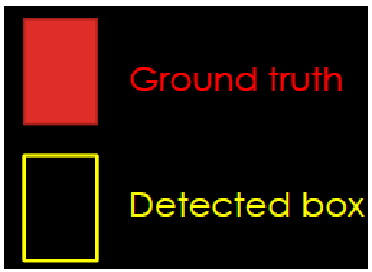
$$AP = \int_0^1 p(r)\,\mathrm{d}r$$

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n}$$

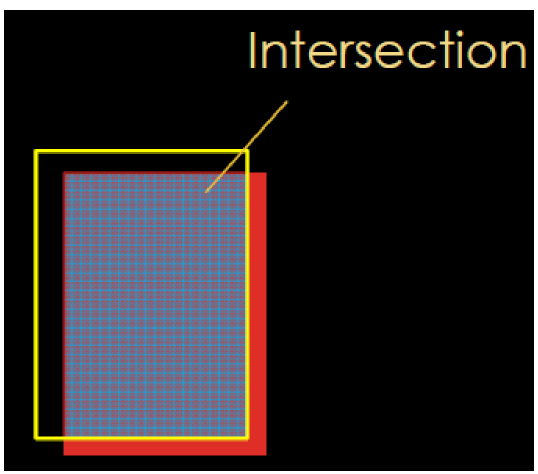


Important Parameters

- IoU (Intersection over Union)
  - 一般IoU>0.5時為預測成功
- Confidence threshold

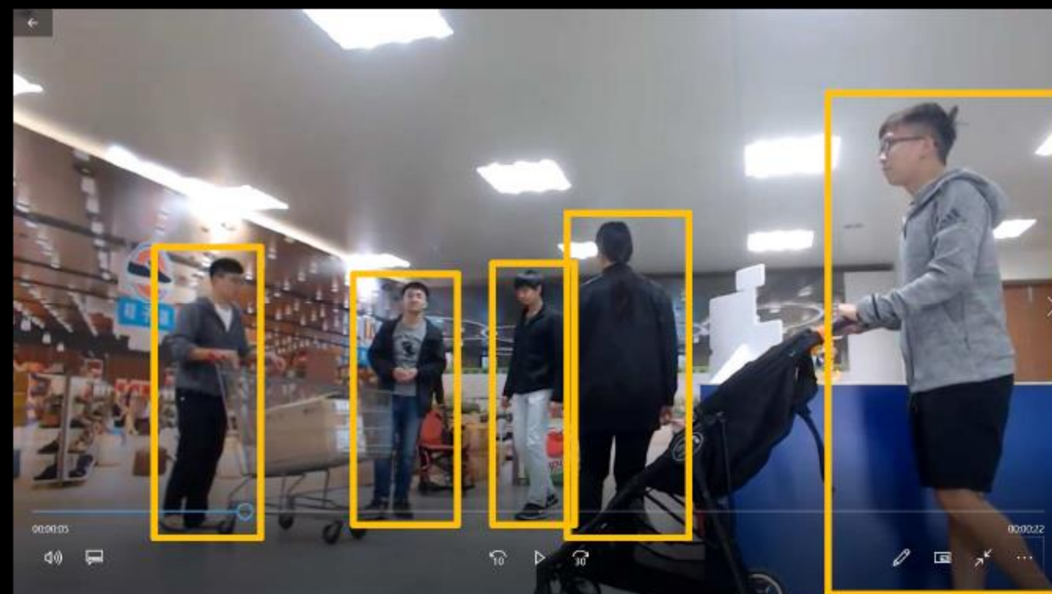


$$IoU(A,B) = \frac{A \cap B}{A \cup B}$$



Example 1

- Class: people
  - TP: 5
  - FP: 0
  - FN: 0
  - Precision: 5/5
  - Recall: 5/5



Example 2

- Class: people
  - TP: 4
  - FP: 0
  - FN: 1
  - Precision: 4/4
  - Recall: 4/5

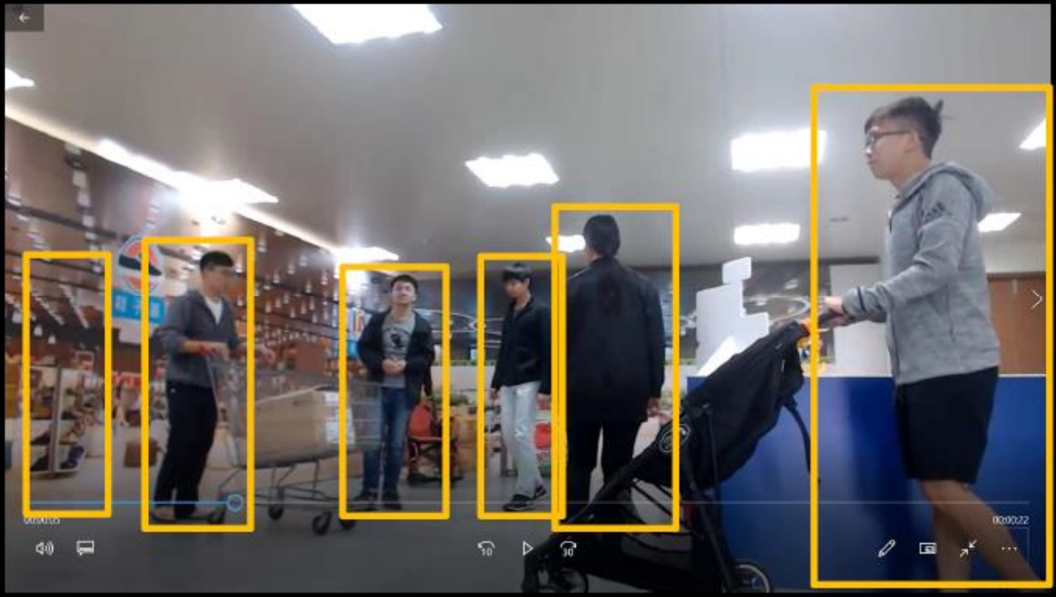
		Actual	
		Positive	Negative
Predictive	Positive	TP	FP
	Negative	FN	TN

$$\text{Precision} = \frac{TP}{TP+FP} \qquad \text{Recall} = \frac{TP}{TP+FN}$$

Example 3

- Class: people
  - TP: 5
  - FP: 1
  - FN: 0
  - Precision: 5/6
  - Recall: 5/5

		Actual	
		Positive	Negative
Predictive	Positive	TP	FP
	Negative	FN	TN



$$\text{Precision} = \frac{TP}{TP+FP} \qquad \text{Recall} = \frac{TP}{TP+FN}$$

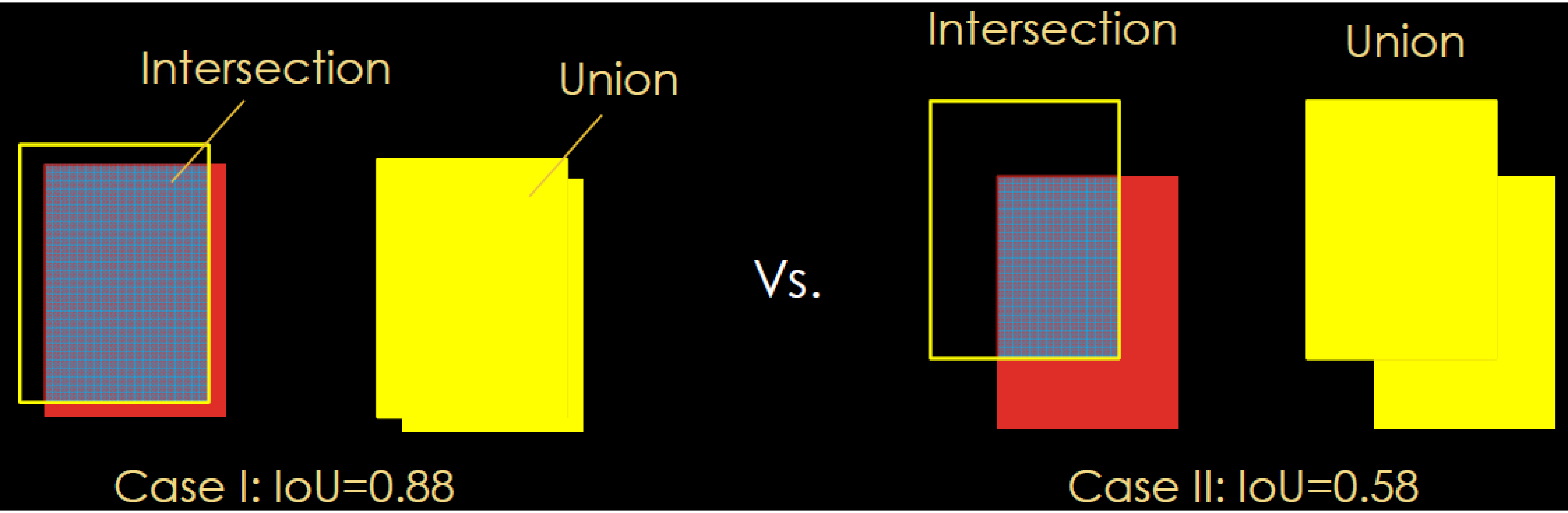
Example 4

- **Assume**
  - **Recall** 1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
  - **Precision** 0.70 0.74 0.78 0.82 0.85 0.89 0.93 0.96 0.98 0.99 1.00
  - **AP=(0.7+0.74+…+1)/11=0.88**

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n} \quad \text{for n classes}$$

How Does IoU Affect AP?

- **Judging criteria of a nice shot**



Commonly Used Indexes

- AP-50: IoU=0.5 as the threshold
  - Both case I (IoU=0.88) and case II (IoU=0.58) get 1 TP
- AP-75: IoU=0.75 as the threshold
  - Case I (IoU=0.88) is TP, but case II (IoU=0.58) is not
  - Besides losing 1 TP, case II generates 1 FP and 1 FN simultaneously
- AP@[0.5 : 0.95]: from IoU=0.5 to IoU=0.95 with a step size of 0.05 (adopted in COCO dataset)