Embedded Vision Intelligent Laboratory

嵌入式智慧影像分析與實境界面
Fall 2021

Instructor：Yen-Lin Chen(陳彥霖), Ph.D.

Professor

Dept. Computer Science and Information Engineering

National Taipei University of Technology

Lecture 4

Resnet神經網路介紹

Resnet介紹

ResNet

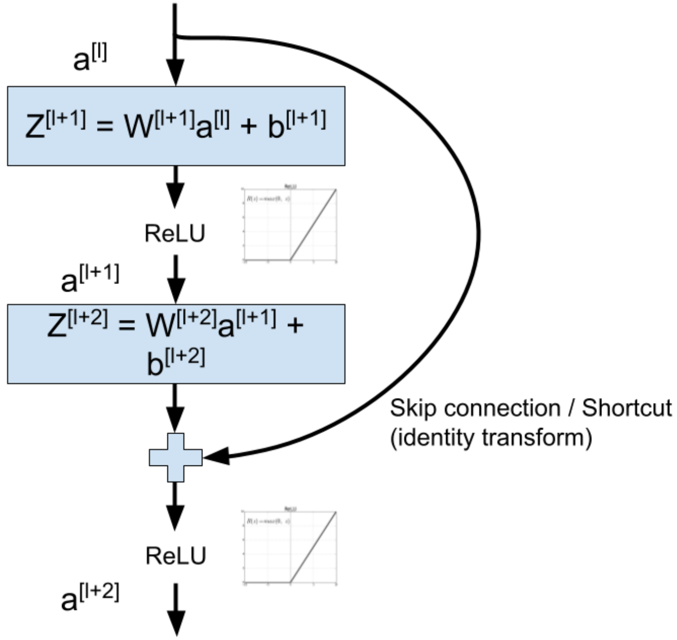
- ResNet是中國大陸計算機科學家何凱明在2015年提出的一種網路結構，在ILSVRC-2015獲得了分類任務的第一名，同時在ImageNet detection，ImageNet localization，COCO detection和COCO segmentation等任務中均獲得了第一名，在當時可謂是轟動一時。

ResNet

- ResNet又名為殘差神經網路，指的是在傳統卷積神經網路中加入殘差學習（residual learning）的思想，解決了深層網路中梯度消失和精度下降（訓練集）的問題，使網路能夠越來越深，既保證了精度，又控制了速度。

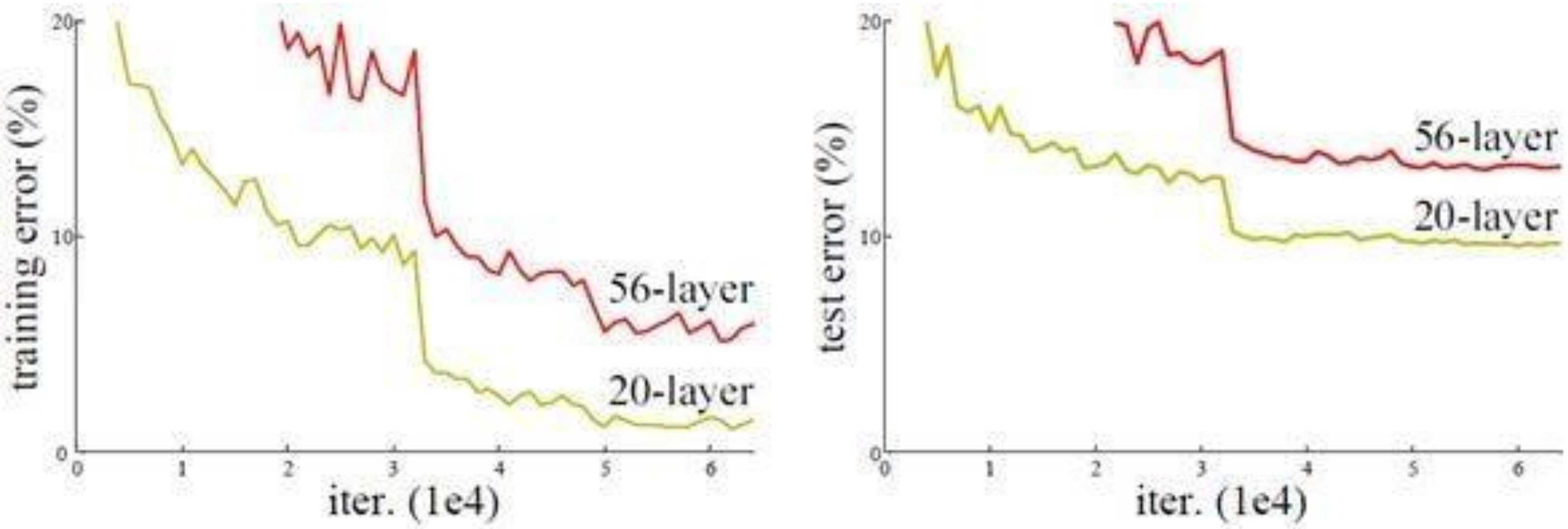
Residual / Bottleneck Block

- Residual Network 使用 Residual Block 的架構(如下圖)
- Skip connection跳過一層線性轉換和非線性輸出，直接成為上兩層中 activation function 的輸入



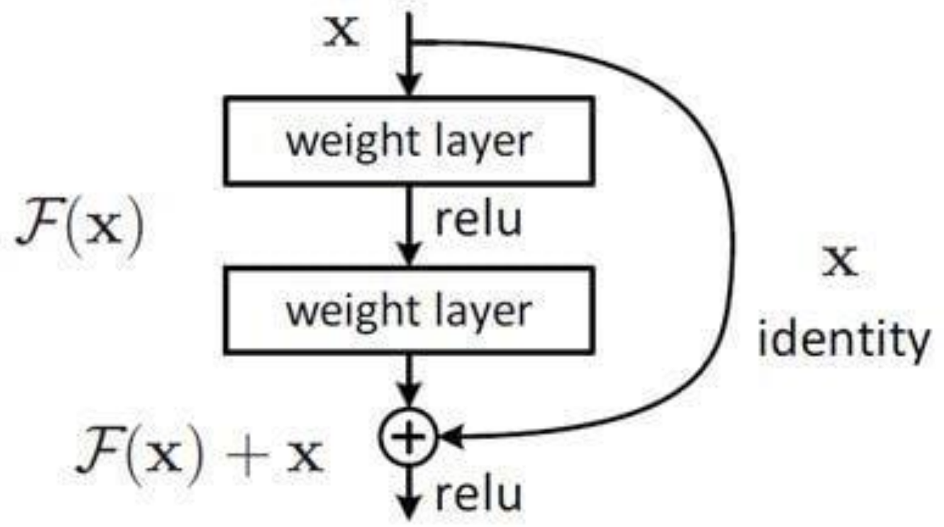
Why Residual Learning

- 神經網路層數多→ gradient vanishing與degradation問題(如下圖，layer越多training error越高，testing error也越高)



Residual Learning(殘差學習)

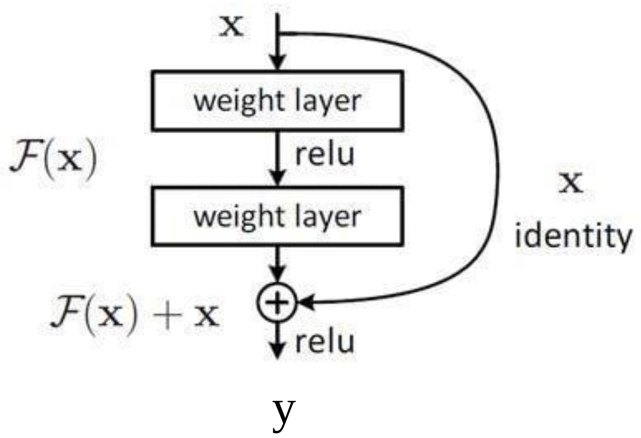
- $y = F(x, \{W_i\}) + x$
  - x: 本身的對映，也就是 x 本身(*identity mapping*)
  - *y: outputs to the layers*
  - $F(x, \{W_i\})$: 殘差對映，也就是$y - x$ (*the residual mapping to be learned*)



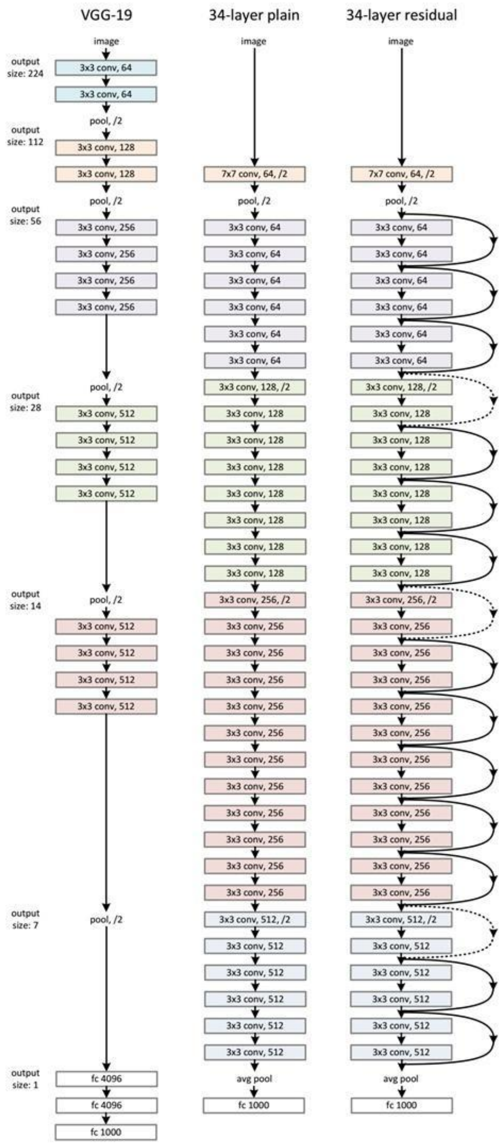
y

Residual Learning(殘差學習)

- 殘差單元可以表示為 (x_l, x_{l+1} 為第 l 個殘差單元的輸入和輸出, F是殘差函數, $h(x_l) = x_l$ 為恆等映射, f為ReLU):
  - $y_l = h(x_l) + F(x_l, W_l)$
  - $x_{l+1} = f(y_l)$
- 從淺層 l 到深層 L 的學習特徵為
  - $x_L = x_l + \sum_{i=1}^{L-1} F(x_i, W_i)$
- Back propagation
  - $\frac{\partial loss}{\partial x_l} = \frac{\partial loss}{\partial x_L} \cdot \frac{\partial x_L}{\partial x_l} = \frac{\partial loss}{\partial x_L} \cdot \left(1 + \frac{\partial}{\partial x_L} \sum_{i=1}^{L-1} F(x_i, W_i)\right)$



Plain Network vs Residual Network



- ResNet網絡參考了VGG19網絡，在其基礎上進行了修改，並通過短路機制加入了殘差單元
- 在ResNet使用stride=2的卷積做down sample，並且用global average pool層替換了fully connected layer
- feature map大小降低一半時，feature map的數量增加一倍，保持了layer複雜度

Plain Network vs Residual Network

- 擁有 Residual Blocks 的 Residual Network(右下) 在 training error 的表現會比起Plain Network(左下)更低

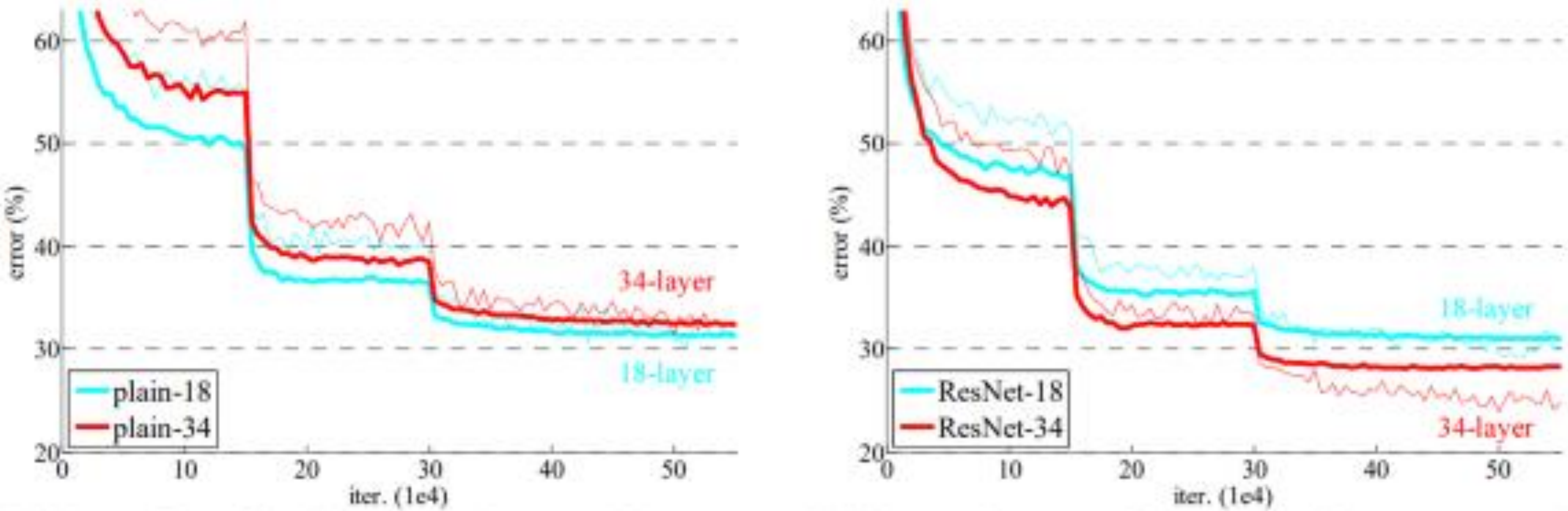
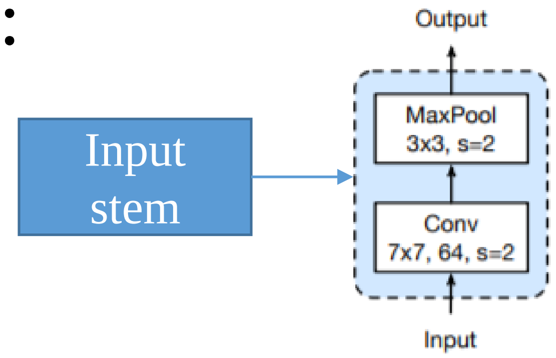


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

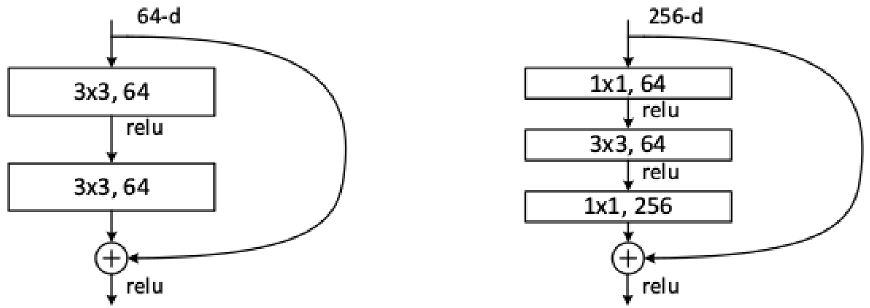
Resnet架構

- 不管是哪種ResNet，整體架構都可以分為三大部分：
  - Input stem:
    - 使用7*7 convolution layer, output channel of 64, stride = 2
    - 使用3*3 max pooling layer, stride = 2
  - Stage block:
    - ResNet共有4個Stage block，每個stage block都是由數個building block堆疊而成。不論是用stride或是pooling，每個stage一般都會先降低解析度並加大寬度 (channel)，再做一連串的residual learning
  - Output stem:
    - 依照任務，設計不同的輸出。一般來說這邊會隨著任務轉變，所以通常不算在ResNet的backbone裡



ResNet架構 cont.

- 針對不同深度的ResNet，作者提出了兩種Residual Block：



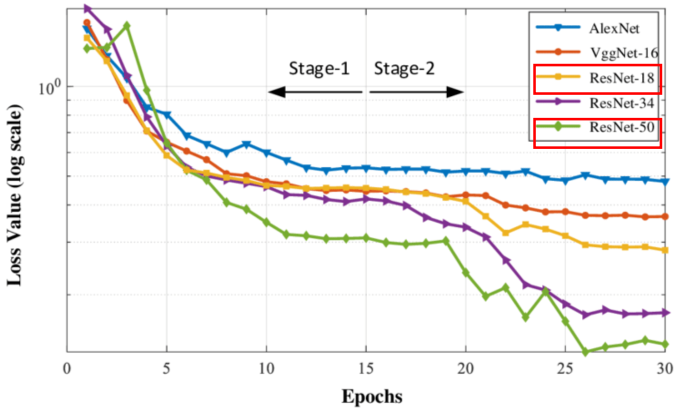
  - 1. 圖左為基本的residual block，residual mapping為兩個64通道的3x3卷積，輸入輸出均為64通道，可直接相加。此block主要使用在較淺層的網路，如ResNet-34
  - 2.圖右為針對深層網路提出的block，稱為"bottleneck" block，主要目的就是為了降維。首先通過一個1x1卷積將256維通道（channel）降到64通道，最後通過一個256通道的1x1卷積恢復。

Resnet架構比較

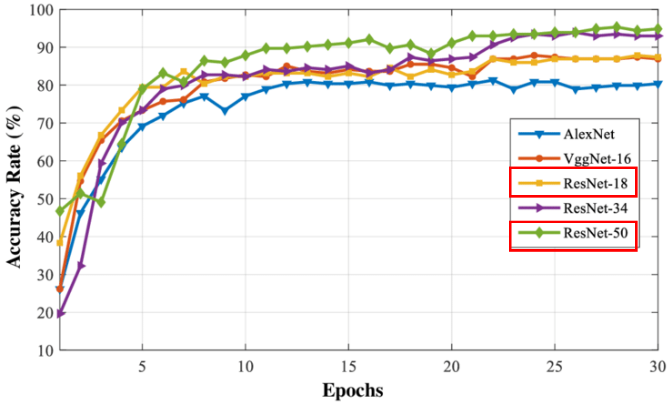
- ResNet18、ResNet34使用一般的residual block
- ResNet50、ResNet101、ResNet152使用了expansion為4的bottleneck block

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Resnet比較



(a)



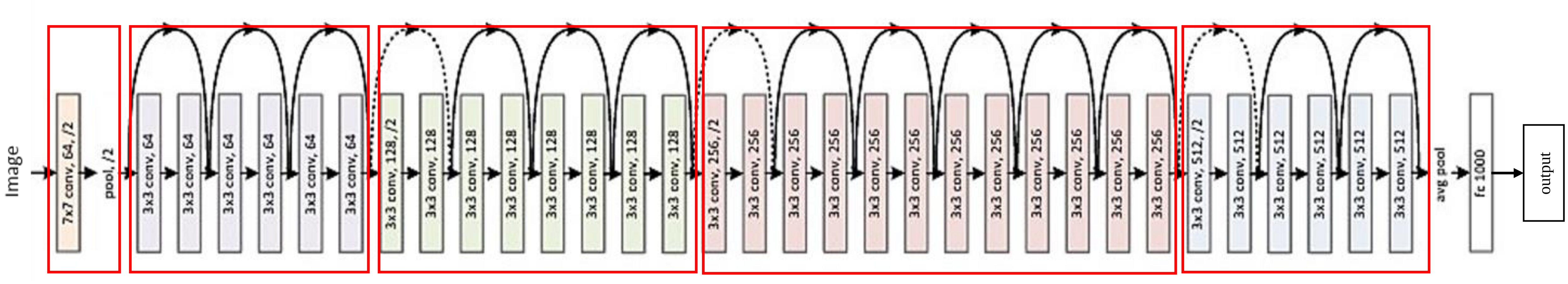
(b)

- Resnet層數越多，Accuracy越高、loss越低
- 但是同時層數越多，運算時間越長
- 可以根據裝置運算能力選擇不同層數的Resnet

Model	Top-1/Top-5 Error (%)	Pruned Para. (%)	FLOPs (%) Pruning	Speed-up× Pruning + Quantization
ResNet-18(Baseline)	29.36/10.02	0	100	1
ResNet-18	30.29/10.43	30.0	71.4	11.4
ResNet-18	30.65/11.93	50.0	44.2	16.0
ResNet-18	33.40/13.37	66.7	29.5	28.2
ResNet-50(Baseline)	24.87/6.95	0	100	1
ResNet-50	23.42/6.93	30.0	66.7	12.0
ResNet-50	24.21/7.65	50.0	47.6	16.0
ResNet-50	28.73/8.37	75.3	27.0	29.6

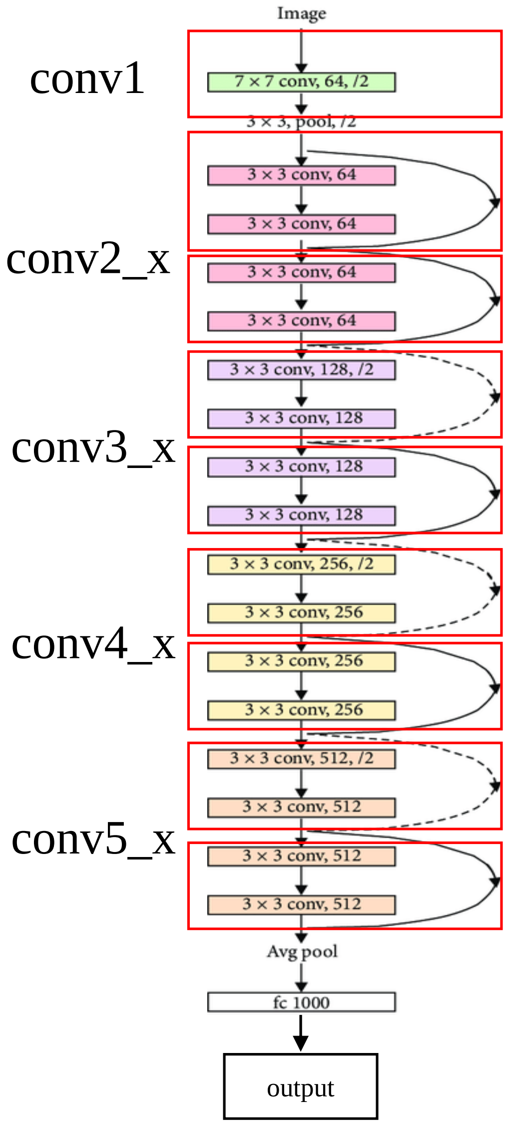
Resnet50

- Input stem (conv1)
- Stage block (conv2_x, conv3_x, conv4_x, conv5_x)
- Output stem
- 網路計算量為3.8x10^9FLOPs



conv1 Conv2_x Conv3_x Conv4_x conv_5_x

Resnet18



- Input stem (conv1)
- Stage block (conv2_x, conv3_x, conv4_x, conv5_x)
- Output stem
- 網路計算量為$1.8\text{x}10^9$FLOPs
- 本次project會使用Resnet18來進行道路辨識模型訓練。

參考資料

參考資料

- Resnet
  - https://ithelp.ithome.com.tw/articles/10204727
  - https://medium.com/%E8%BB%9F%E9%AB%94%E4%B9%8B%E5%BF%83/deep-learning-residual-leaning-%E8%AA%8D%E8%AD%98resnet%E8%88%87%E4%BB%96%E7%9A%84%E5%86%A0%E5%90%8D%E5%BE%8C%E7%B9%BC%E8%80%85resnext-resnest-6bedf9389ce
  - https://kknews.cc/zh-tw/code/xpaz689.html
  - https://arxiv.org/pdf/1812.01187.pdf