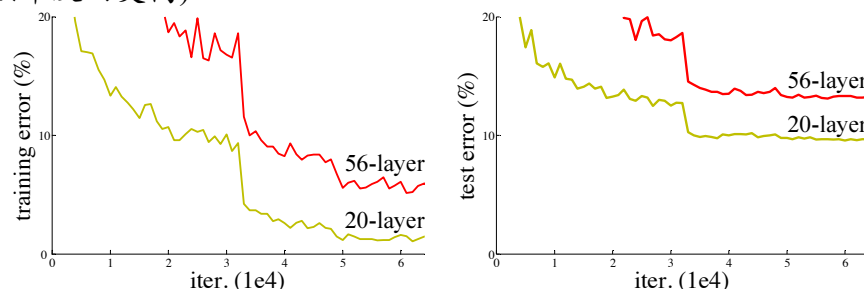


基於DenseNet研究之總結報告

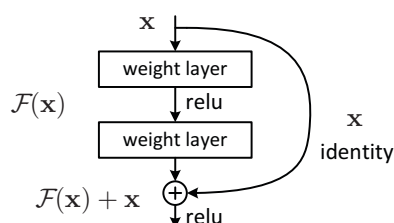
(一) 相關工作

1.1 ResNet

Kaiming He 在 2016 CVPR 上發表 [Deep Residual Learning for Image Recognition](#)，提出殘差學習(residual learning) 框架，緩解梯度消失(gradient vanish)問題(神經網路在增加層數後，錯誤率反而更高)。

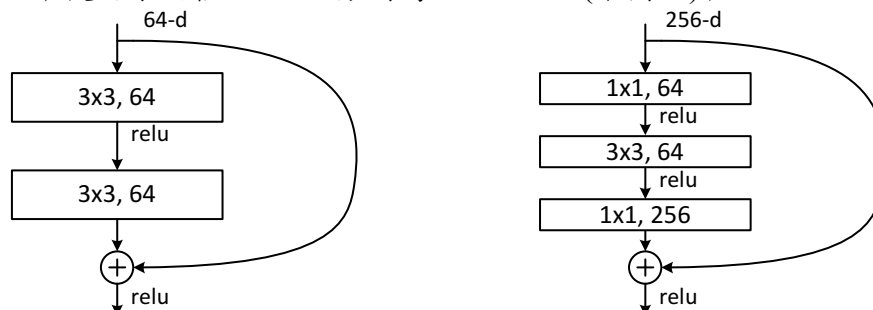


將原先的 $x \rightarrow H(x)$ (輸入 \rightarrow 特徵) 變成 $x \rightarrow F(x) + x$ (輸入 \rightarrow 輸入 + 殘差)：



$$H(x) = F(x) + x$$

ResNet 中使用了兩種 block，分別為 basic block(下圖左)和 bottleneck(下圖右)：



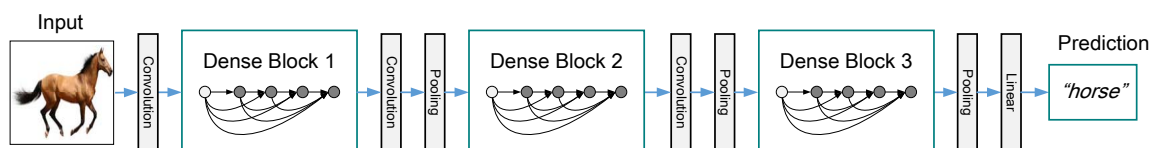
1.2 DenseNet

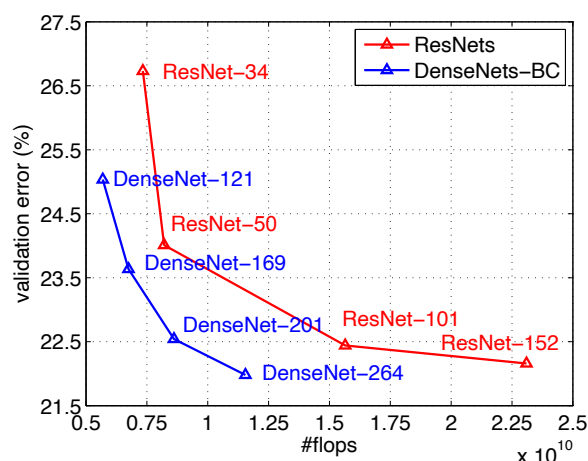
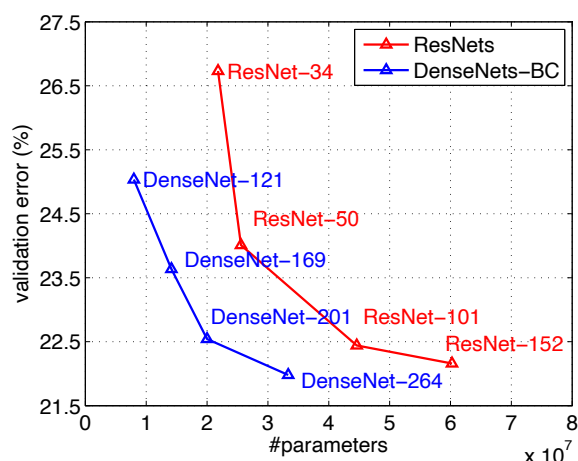
Gao Huang 在 2017 CVPR 上發表 [Densely Connected Convolutional Networks](#)，比 ResNet 更激進地連接方式，所有卷積層都相互連接。

因為在梯度下降法和反向傳播訓練神經網路時，梯度會越來越小，最終導致無法更新網路的權重(梯度消失)。DenseNet 連接所有的卷積層進行特徵重用，提高特徵利用率和模型收斂速度。

$$x = H(x_0, x_1, x_2, \dots, x_n)$$

DenseNet 的優點：(1)緩解梯度消失問題，(2)加強特徵傳播，鼓勵特徵重用，(3)大幅減少參數數量。

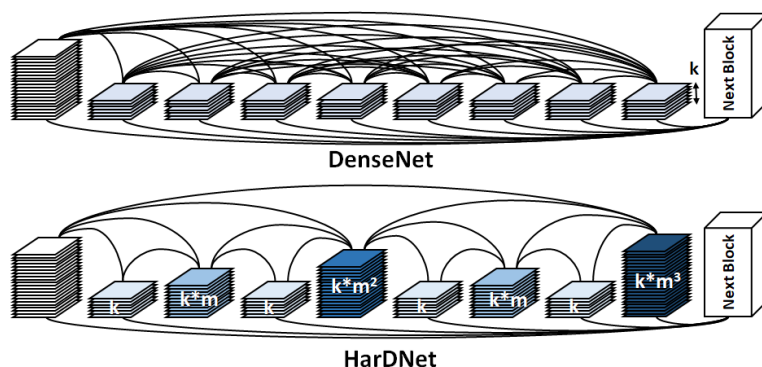




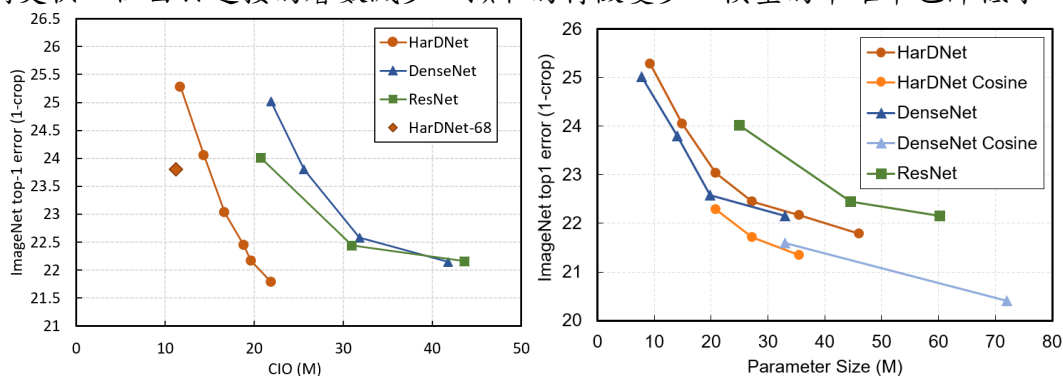
1.3 HarDNet

Ping Chao 在 2019 ICCV 上發表 [HarDNet: A Low Memory Traffic Network](#)，減少了 DenseNet 中的連接數量來降低網路的記憶體佔用量。

因為 DenseNet 使用了密集連接，每一層擷取的特徵都會保留到最後一層，保留下來的特徵也會在下一層擷取更細節的特徵，因此會導致資料搬運的次數過多。



HarDNet 和 DenseNet 的實驗結果比較：HarDNet 的記憶體佔用量更小，模型的推理時間更快。但由於連接的層數減少，擷取的特徵變少，模型的準確率也降低了。



(二) 研究成果

2.1 Rui-Yang Ju, Ting-Yu Lin, Jia-Hao Jian, Jen-Shiun Chiang, Wei-Bin Yang. [ThreshNet: An Efficient DenseNet Using Threshold Mechanism to Reduce Connections](#). *IEEE Access*, 2022.

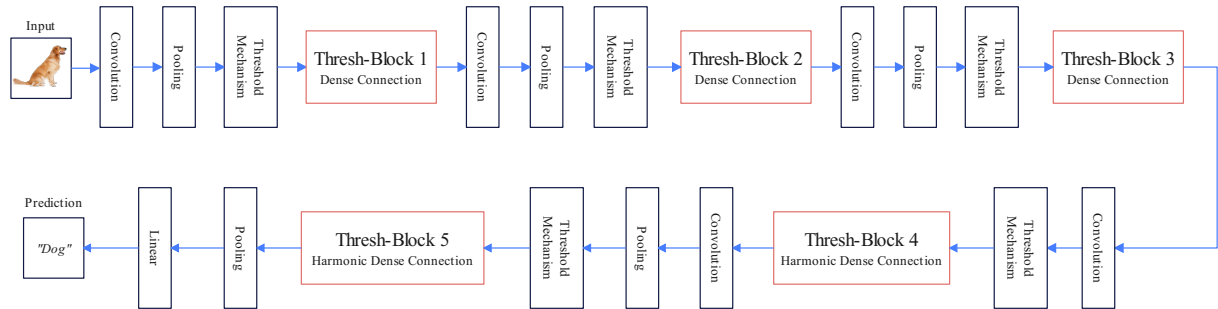
2.2 Rui-Yang Ju, Ting-Yu Lin, Jia-Hao Jian, Jen-Shiun Chiang. [Efficient Convolutional Neural Networks on Raspberry Pi for Image Classification](#). *Journal of Real-Time Image Processing (JRTIP)*, 2022.

2.3 Rui-Yang Ju, Jen-Shiun Chiang, Chih-Chia Chen, Yu-Shian Lin. [Connection Reduction of DenseNet for Image Recognition](#). *International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Penang, Malaysia, Nov. 2022.

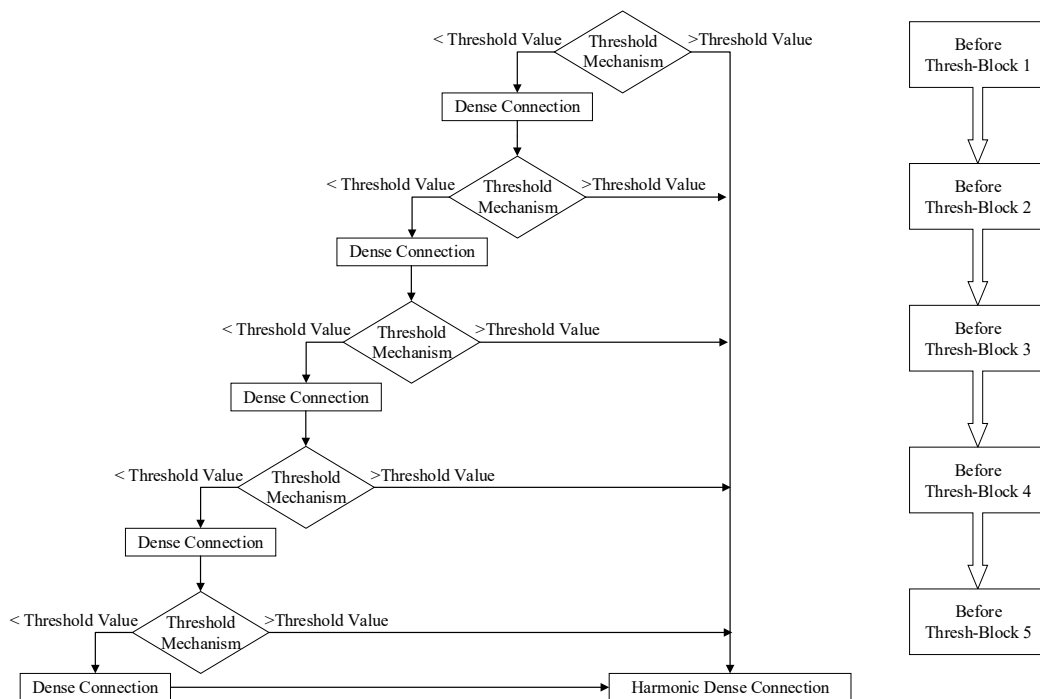
(三) 研究內容

3.1 ThreshNet

因為 DenseNet 使用密集連接(Dense Connection)鼓勵特徵重用，模型的參數量較小；HarDNet 使用諧波密集連接(Harmonic Dense Connection)，提高了網路的記憶體佔用量、縮短了模型的推理時間，但是模型的準確率下降了。為了平衡神經網路的記憶體佔用量和模型的準確率。

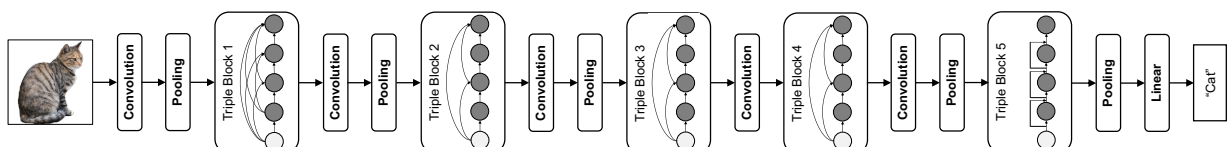


因為層數大的 block 如果使用密集連接會導致資料的搬運次數過大。因此我們設置一個 Threshold Mechanism，對於層數較小的 block 使用密集連接，對於層數較大的 block 使用諧波密集連接。



3.2 TripleNet

在卷積神經網路(CNN)，隨著卷積層的不斷堆疊，通道數(Channel)也會成倍增加。為了降低模型的大小，可以在邊緣運算平台(樹莓派 Raspberry Pi)上實現。因此我們在最後一個 block 中只使用 2 或 3 個層，並且使用 bottleneck 和殘差學習來進行連接。



我們設計的網路由 5 個 block 組成，第 1 個 block 使用密集連接，第 2、3、4 個 block 使用諧波密集連接，第 5 個 block 使用殘差學習進行連接。

3.3 Connection Reduction

根據對 ThreshNet 和 TripleNet 的研究，我們發現卷積層之間的不同連接方式直接影響模型的參數量、記憶體佔用量、推理時間和準確率。因此我們提出了兩種算法來進行層與層之間的連接。

Algorithm 1 *Shortnet₁*

Require: Stack 3×3 Convolution Layer in each block

Ensure: $BN + ReLU + 3 \times 3 Conv$

```

for layer  $n$  is odd do
  layer  $n$  connect to layer  $2^0$ 
  for  $i$  in  $2^1$  to  $2^5$  do
    let  $n$  connect to layer  $i$  ( $i \leq n$  and  $i$  is even)
  end for
end for
for layer  $n$  is even do
  layer  $n$  connect to layer  $2^0$ 
  for  $i$  in  $2^1$  to  $2^5$  do
    let  $n$  connect to layer  $i$  ( $i \leq n$  and  $i$  is odd)
  end for
end for

```

Algorithm 2 *Shortnet₂*

Require: Stack 3×3 Convolution Layer in each block

Ensure: $BN + ReLU + 3 \times 3 Conv$

```

for layer  $n$  is even do
  for  $i$  in 1 to 5 do
     $x = 2^i = 1$ 
     $y = n - x$  ( $x \leq n$ )
    layer  $n$  connect to layer  $y$ 
  end for
end for
for layer  $n$  is odd do
  layer  $n$  connect to layer  $y$ 
end for

```

使用算法 1 進行層與層之間的連接，連接的數量和諧波密集連接的數量相似；使用算法 2 進行層與層之間的連接，方法更激進，連接的數量更少。