國立雲林科技大學  
電子工程所

National Yunlin University of Science and Technology

Graduate School of Electronic Engineering

報告題目 :汽車駕駛者監控系統設計與實現

|  |  |
| --- | --- |
| 報告日期(Date)： | 2022/10/28 |
| 姓名(Name)： | 謝昕樺(碩一生) |
| 指導教授(Professor)： | 蘇慶龍副教授 |
| E-mail： | [M11113020 yuntech.edu.tw](mailto:M10813007@yuntech.edu.tw) |
| 學號 (Student ID)： | M11113020 |

**本週進度(Progress of this week)**

* 查看YOLOV5/YOLOV7架構

11/25 Fri 22:00~00:00 (2.0 個小時)

11/26 Sat 00:00~03:00 (3.0 個小時)

11/27 Sun 21:00~00:00 (3.0 個小時)

11/28 Mon 13:00~22:00 (9.0 個小時)

11/29 Tue 10:00~13:00 (3.0 個小時)

11/29 Tue 16:30~23:00 (7.5 個小時)

* 看學長給的CODE和學長架構

11/30 Wed 13:30~23:00 (10.5 個小時)

12/1 Thu 17:30~20:00 (2.5 個小時)

12/1 Thu 21:00~23:00 (2.0 個小時)

12/2 Fri 11:30~17:30 (6.0 個小時)

12/4 Sun 15:00~17:00 (2.0 個小時)

12/4 Sun 01:00~04:00 (4.0 個小時)

12/5 Mon 12:30~19:00 (7.5 個小時)

12/5 Mon 20:00~23:30 (3.5 個小時)

12/6 Tue 16:30~22:30 (6.0 個小時)

12/7 Wed 11:30~00:30 (13.0個小時)

12/8 Tue 16:30~22:00 (5.5 個小時)

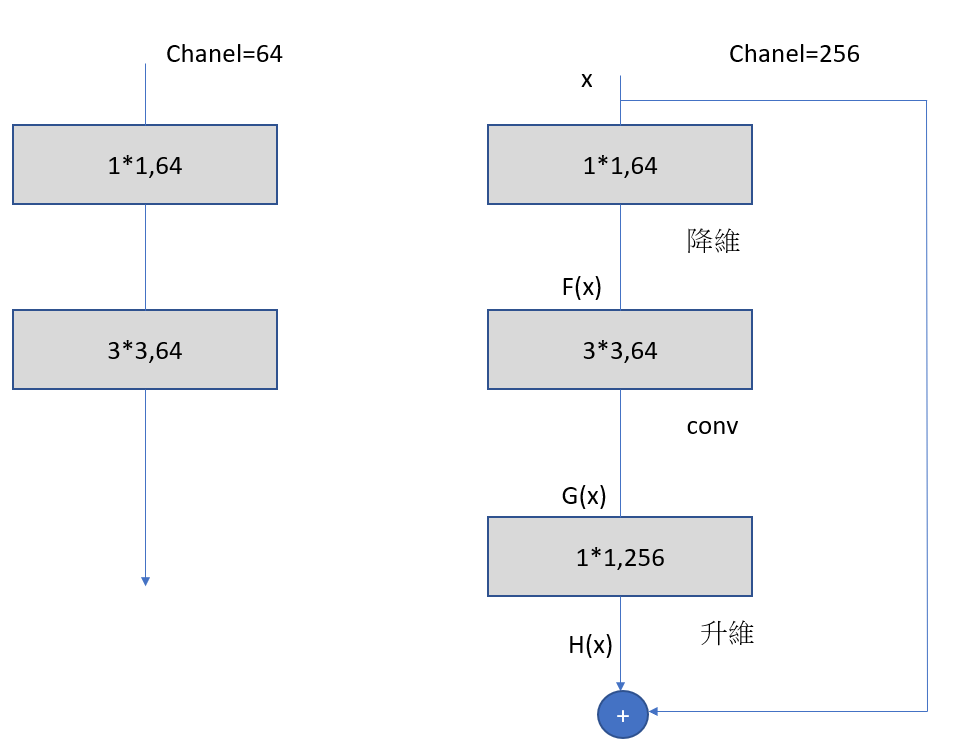
* 製作Meeting報告

共小時

|  |  |
| --- | --- |
| **11/25進度** | **12/9進度** |
| 1. 捲積加速運算 2. Xnor-NetWork運算 | 1. 看學長code架構 2. Yolov5/Yolov7架構 |
| **待辦事項** | |

* ResNet

當網路深度越來越深時，會有梯度消失問題，這種現象不是Overfitting問題，而是model疊的層數太深，導致Backpropagation偏微分到後來沒東西

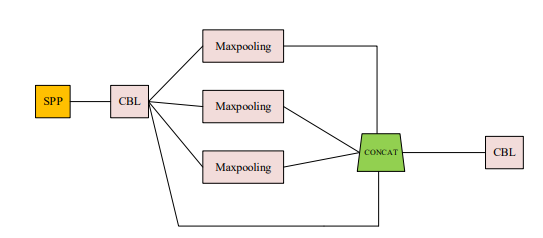


假設今天輸入X，經由一次Conv後，得到結果為F(x)，因此可以得出殘差網路為，那ResNet就是利用1\*1的Conv去進行升降維，提取完特徵後再和原本的輸入做殘差運算，這樣就會產生新的特徵，梯度就不會消失那麼快

* SPP

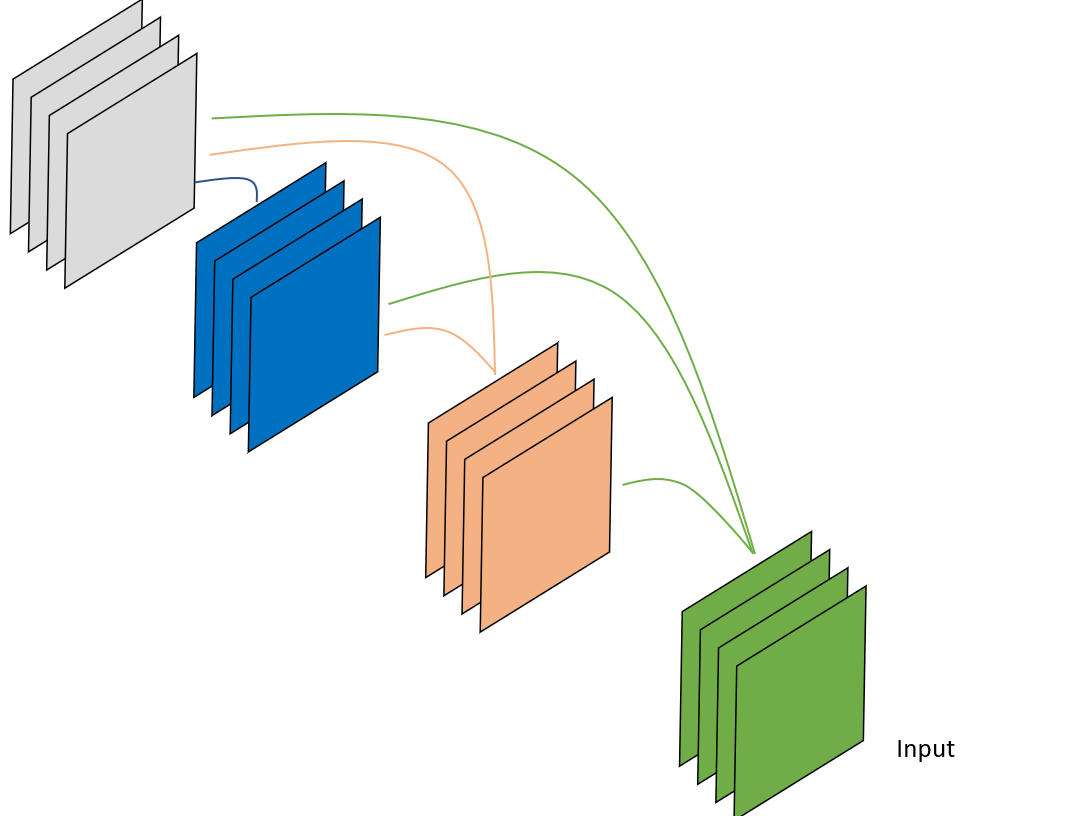
SPP又稱空間金字塔，他可以將任意大小的特徵圖轉換為固定大小特徵圖，YOLOV5中SPP三個Maxpooling的Kernel Size預設為5,9,13，將pooling過後的Chanel進行Concat，可以提取特徵同時又增加Chanel深度，那經過Pooling過後的Feature map size為式(1-1)

式(1-1)



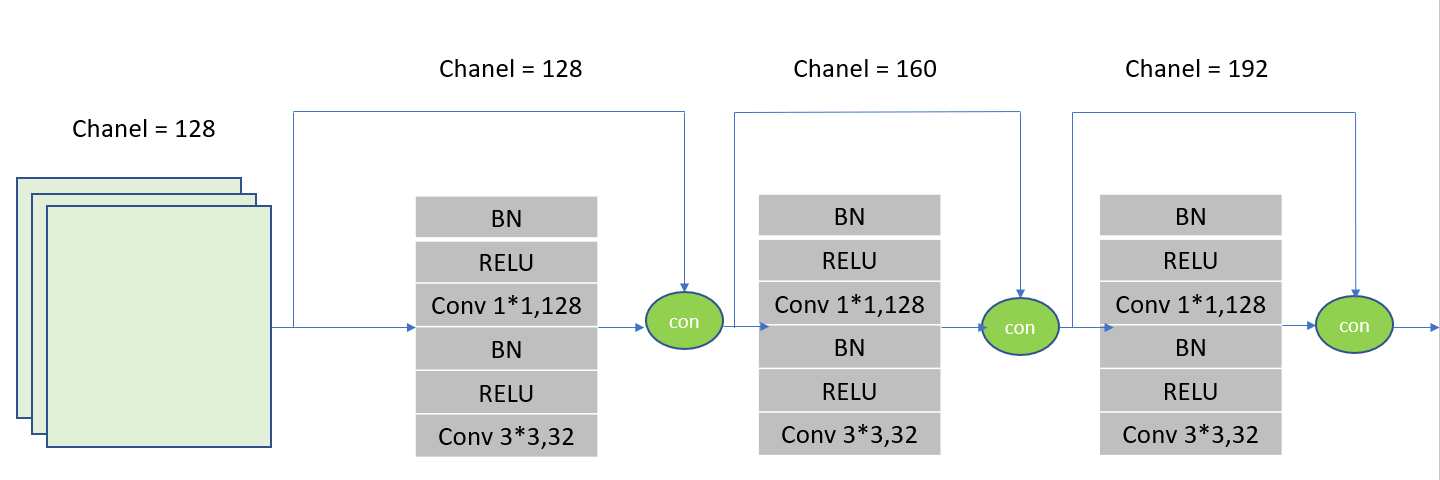
* DenseNet

DenseNet是ResNet的延伸，它比ResNet更可以減輕梯度消失，並且增強Feature之間的傳遞，也使用較少參數量，每層Block之間都與後面的Block進行連接



下圖為每個Block內部的捲積，Conv 1\*1,128為固定要讓輸入降維，假設10幾個block相連接沒有降維通道數會急遽增加，而最後的Conv 3\*3,32要讓模型Concat後以32的成長量進行增長

，Block內採取Activation在前，Conv在後，並且每層的Block都有Concat到下一層



DenseNet缺點為要進行多次的Concat，數據會一直保存在顯卡記憶體，因此需要對顯卡記憶體做優化

* VoVNet

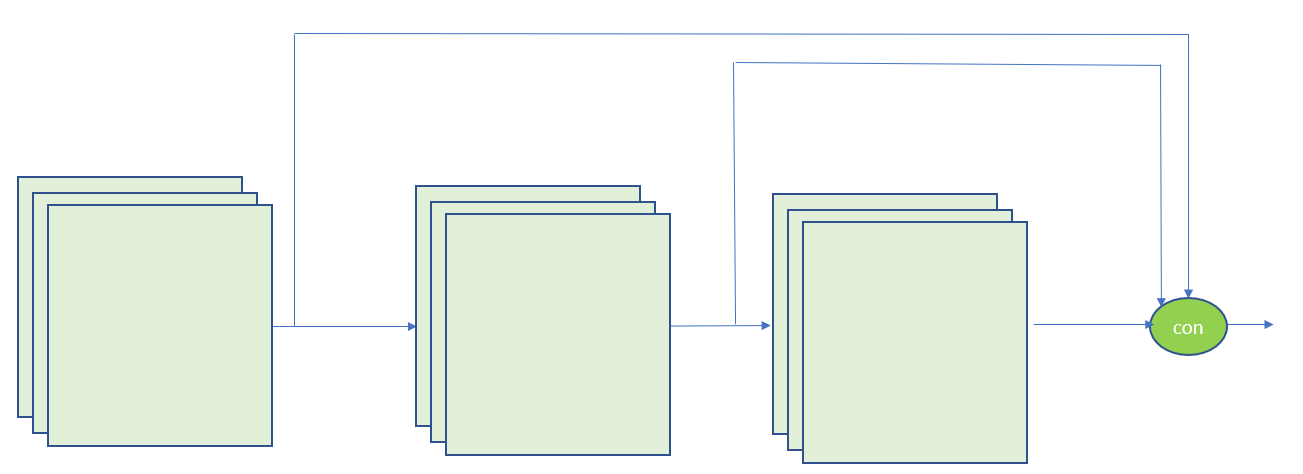
VoVNet改善了DenseNet的缺點，首先DenseNet的內存訪問成本太高(Memory Access Cost)，每層都採取密集連接，會導致Mac隨著網路深度增加，Mac計算方法如式(1-2)、式(1-3)、式(1-4)，當Chanel\_IN和Chanel\_OUT相等時，Mac為最小值，有較高效率

式(1-2)

式(1-3)

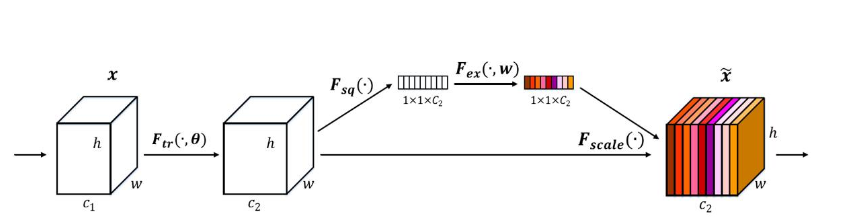
式(1-4)

接下來式GPU計算方面，GPU的優勢在於平行運算，若使用深度可分離捲積或者用1\*1Conv升降維，雖然可以減少FLOPs(浮點計算量，可以比喻為計算量)，但是由上面的Mac推出來結果得知相同的Chanel\_IN和Chanel\_OUT對GPU有較好的FLOPS(每秒浮點數計算次數，可以比喻為計算速度)，因此VOVNet他採取的策略是每一層的Chanel\_IN和Chanel\_OUT都使用相同維度，並且在最後一層Concat所有Block

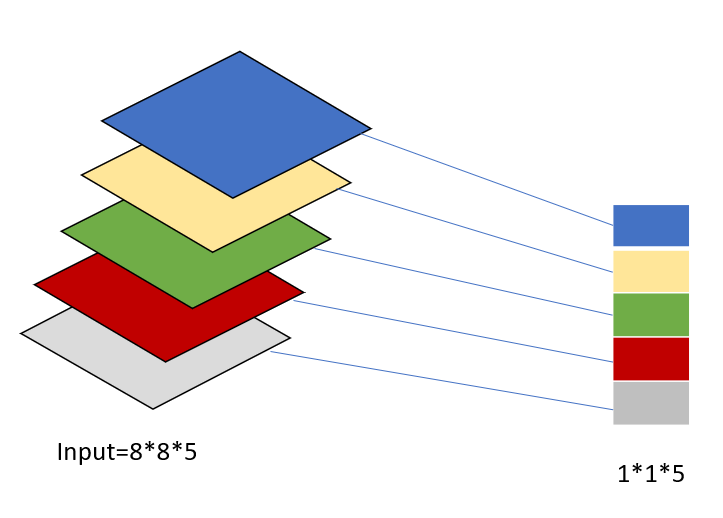


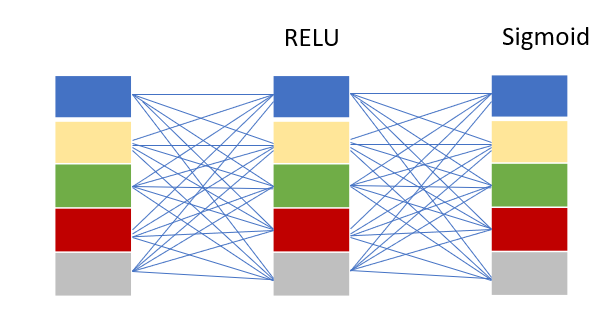
* SeNet

SeNet架構為下圖，通過學習方式來自動提取每個通道特徵的重要性，用這個特性去提升有用的特徵，抑制沒用的特徵



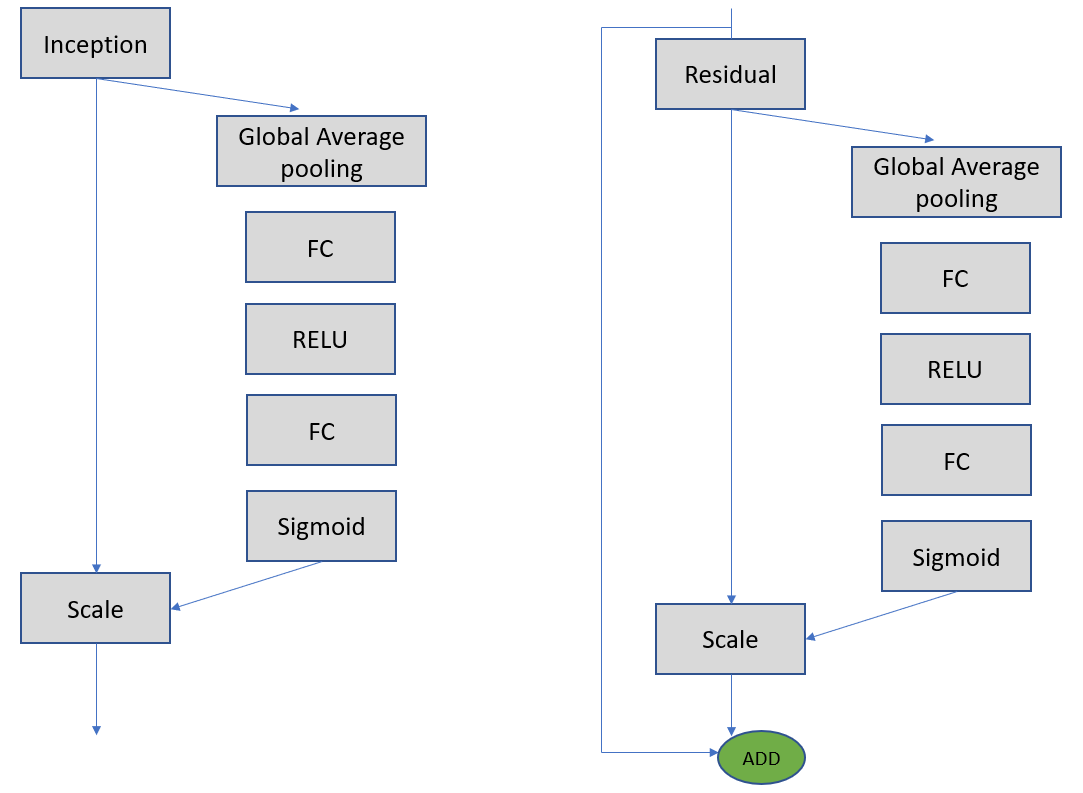
首先是部分，論文作者透過Global Average Pooling和Global Max Pooling做比較，發現Global Average Pooling效果比較好，Global Average Pooling原理為將每層各自的Chanel做Normolization，將原本Feature Map Size = h\*w\*c轉換為1\*1\*c



接下來是部分，通過兩層Fully Connected，第一層的Fully Connected的Activation採用Relu，第二層的Fully Connected的Activation採用Simoid

最後是部分，將最終Sigmoid得出的1\*1\*Chanel在和原本

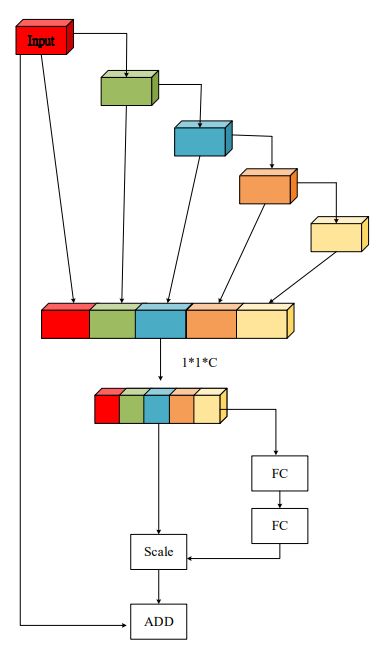
的做點積，最後的Actavation採用Sigmoid原因是要將數值控制在0~1之間做點積時可以讓模型自行學習通道重要性



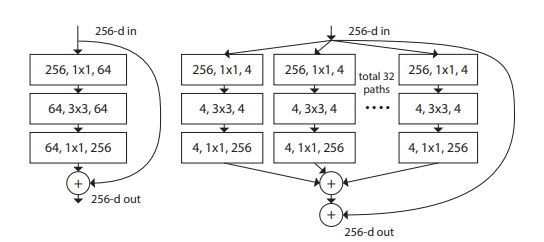
上圖左邊是沒有加入ResNet網路，右邊是加入ResNet網路的SeNet，SeNet並不是一個完整的網路架構，它只是一個子架構，因此可以插入其他模型或網路中

* VoVNetv2

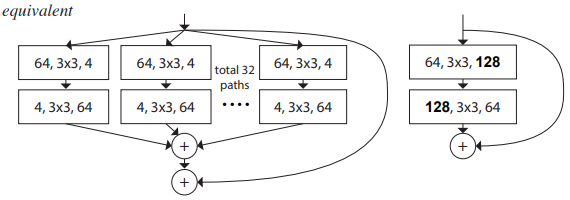
VoVNetv2採用了VoVNet原生的架構，再加入SeNet和ResNet，架構變為下圖



* ResNeXt

ResNeXt是將高維度卷積層分組(Cardinality)成相同的卷積層進行卷積運算，最後將卷積層進行融合

論文作者用相同的Parameter與ResNet和ResNeXt做比較發現獲得更高準確率



最後論文作者提供三種等價的ResNeXt架構，得出相同的結果

