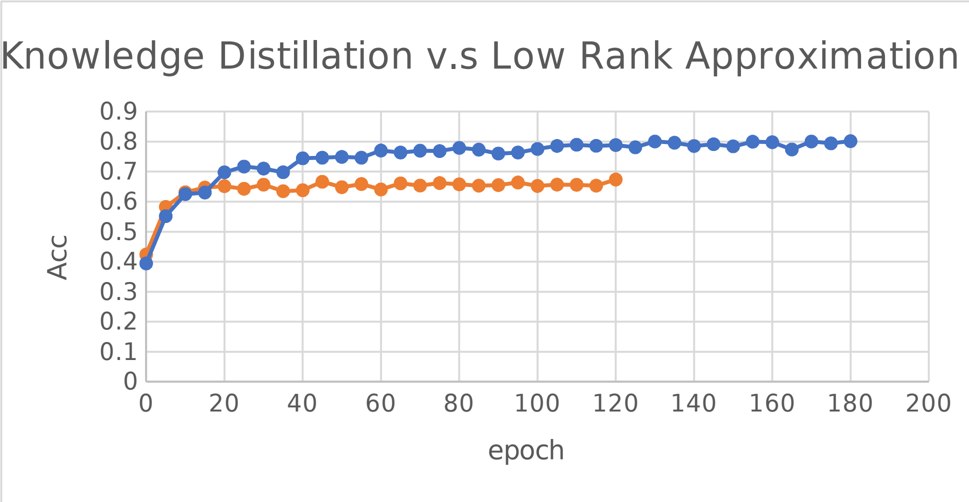
學號：b06502158　　系級：機械三　　姓名：陳柏元

1. 請從 Network Pruning/Quantization/Knowledge Distillation/Low Rank Approximation 選擇兩個方法(並詳述)，將同一個大 model 壓縮至同等數量級，並討論其 accuracy 的變化。 (2%)

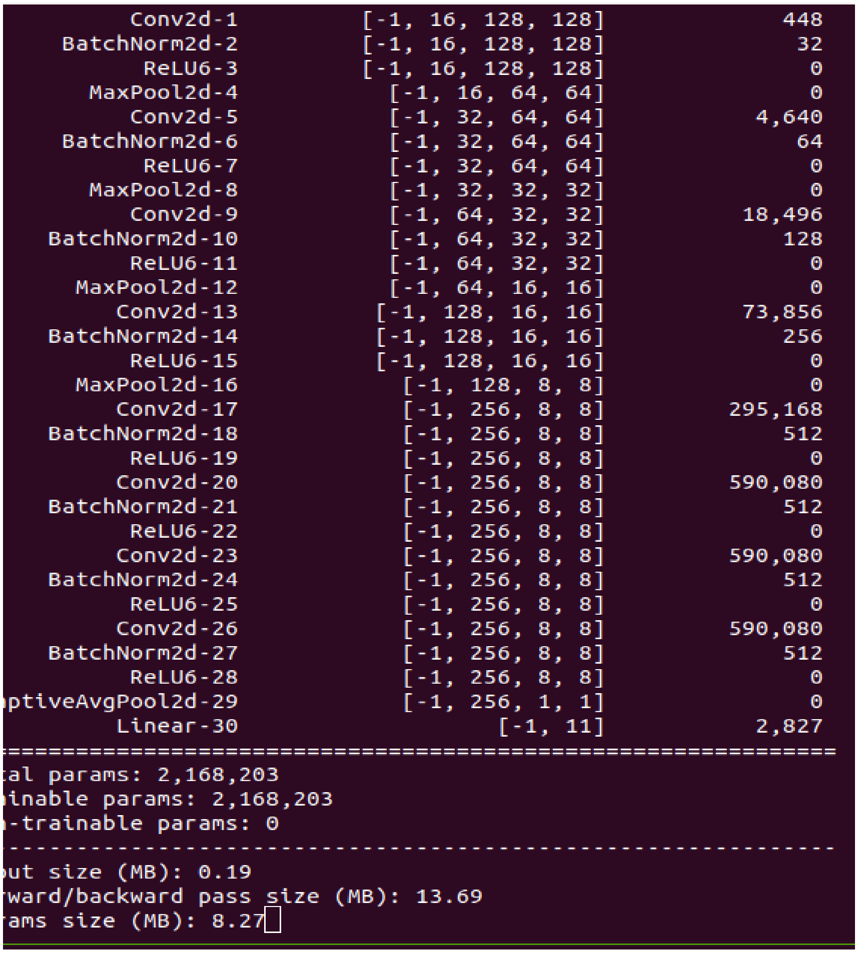


圖一、KD v.s Model Architecture 的 validation Accuracy

```
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1        [-1, 16, 128, 128]             448
       BatchNorm2d-2        [-1, 16, 128, 128]              32
             ReLU6-3        [-1, 16, 128, 128]               0
         MaxPool2d-4          [-1, 16, 64, 64]               0
            Conv2d-5          [-1, 16, 64, 64]             160
       BatchNorm2d-6          [-1, 16, 64, 64]              32
             ReLU6-7          [-1, 16, 64, 64]               0
            Conv2d-8          [-1, 32, 64, 64]             544
         MaxPool2d-9          [-1, 32, 32, 32]               0
           Conv2d-10          [-1, 32, 32, 32]             320
      BatchNorm2d-11          [-1, 32, 32, 32]              64
            ReLU6-12          [-1, 32, 32, 32]               0
           Conv2d-13          [-1, 64, 32, 32]           2,112
        MaxPool2d-14          [-1, 64, 16, 16]               0
           Conv2d-15          [-1, 64, 16, 16]             640
      BatchNorm2d-16          [-1, 64, 16, 16]             128
            ReLU6-17          [-1, 64, 16, 16]               0
           Conv2d-18         [-1, 128, 16, 16]           8,320
        MaxPool2d-19           [-1, 128, 8, 8]               0
           Conv2d-20           [-1, 128, 8, 8]           1,280
      BatchNorm2d-21           [-1, 128, 8, 8]             256
            ReLU6-22           [-1, 128, 8, 8]               0
           Conv2d-23           [-1, 256, 8, 8]          33,024
           Conv2d-24           [-1, 256, 8, 8]           2,560
      BatchNorm2d-25           [-1, 256, 8, 8]             512
            ReLU6-26           [-1, 256, 8, 8]               0
           Conv2d-27           [-1, 256, 8, 8]          65,792
           Conv2d-28           [-1, 256, 8, 8]           2,560
      BatchNorm2d-29           [-1, 256, 8, 8]             512
            ReLU6-30           [-1, 256, 8, 8]               0
           Conv2d-31           [-1, 256, 8, 8]          65,792
           Conv2d-32           [-1, 256, 8, 8]           2,560
      BatchNorm2d-33           [-1, 256, 8, 8]             512
            ReLU6-34           [-1, 256, 8, 8]               0
           Conv2d-35           [-1, 256, 8, 8]          65,792
AdaptiveAvgPool2d-36           [-1, 256, 1, 1]               0
           Linear-37                  [-1, 11]           2,827
================================================================
Total params: 256,779
Trainable params: 256,779
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.19
Forward/backward pass size (MB): 13.13
Params size (MB): 0.98
Estimated Total Size (MB): 14.29
```

圖二、KD&Model Architecture 架構

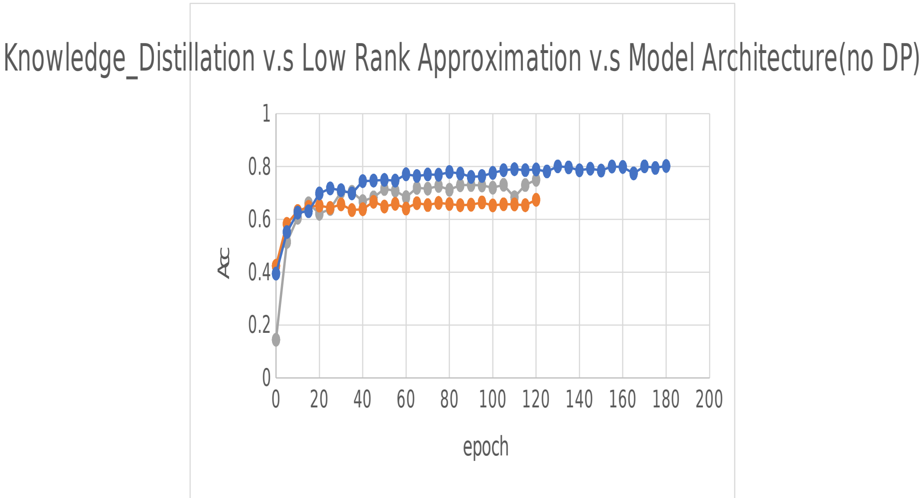
我選擇了 Low Rank Approximation 以及 Knowledge Distillation 兩種方法來進行比較。Low Rank Approximation / Model Architecture 當中以 Depthwise & Pointwise 來進行 training。Knowledge Distillation 當中以前者 Model Architecture 的架構當 Student Net，並以助教提供的大 model，ResNet18 (ImageNet pretrained & fine-tune)，來 train。由於在此兩者 pruning 的過程當中使用的架構相同，所以參數量皆為 256,779，大小也皆為 0.98MB。從圖一可以明顯觀察到，如果給了小 model，也就是上述的 Student Net 一個表現優異的大 model 參數來學習，明顯的提升了不少準確率。而在 Depthwise & Pointwise 的 pruning 下，從原先的 2,168,203 龐大的參數量，降為 1/8 的 256,779，大小則也差不多降了 8 倍，8.27MB->0.98MB，如下圖三。



```
          Conv2d-1         [-1, 16, 128, 128]             448
     BatchNorm2d-2         [-1, 16, 128, 128]              32
           ReLU6-3         [-1, 16, 128, 128]               0
       MaxPool2d-4           [-1, 16, 64, 64]               0
          Conv2d-5           [-1, 32, 64, 64]           4,640
     BatchNorm2d-6           [-1, 32, 64, 64]              64
           ReLU6-7           [-1, 32, 64, 64]               0
       MaxPool2d-8           [-1, 32, 32, 32]               0
          Conv2d-9           [-1, 64, 32, 32]          18,496
    BatchNorm2d-10           [-1, 64, 32, 32]             128
          ReLU6-11           [-1, 64, 32, 32]               0
      MaxPool2d-12           [-1, 64, 16, 16]               0
         Conv2d-13          [-1, 128, 16, 16]          73,856
    BatchNorm2d-14          [-1, 128, 16, 16]             256
          ReLU6-15          [-1, 128, 16, 16]               0
      MaxPool2d-16            [-1, 128, 8, 8]               0
         Conv2d-17            [-1, 256, 8, 8]         295,168
    BatchNorm2d-18            [-1, 256, 8, 8]             512
          ReLU6-19            [-1, 256, 8, 8]               0
         Conv2d-20            [-1, 256, 8, 8]         590,080
    BatchNorm2d-21            [-1, 256, 8, 8]             512
          ReLU6-22            [-1, 256, 8, 8]               0
         Conv2d-23            [-1, 256, 8, 8]         590,080
    BatchNorm2d-24            [-1, 256, 8, 8]             512
          ReLU6-25            [-1, 256, 8, 8]               0
         Conv2d-26            [-1, 256, 8, 8]         590,080
    BatchNorm2d-27            [-1, 256, 8, 8]             512
          ReLU6-28            [-1, 256, 8, 8]               0
ptiveAvgPool2d-29            [-1, 256, 1, 1]               0
         Linear-30                   [-1, 11]           2,827
==============================================================
al params: 2,168,203
inable params: 2,168,203
-trainable params: 0
--------------------------------------------------------------
ut size (MB): 0.19
ward/backward pass size (MB): 13.69
ams size (MB): 8.27
```

圖三、Model Architecture(no DP)架構

下圖中的灰色曲線 Model Architecture(no DP)，雖然相較於加了 Depthwise & Pointwise 來進行 training 的 validation 高了一些 5-7%左右，圖四中灰線，但縮減了八倍的參數量後，其實仍然是值得的，畢竟不是所有裝置都有能力跑大 model 的。比較有趣的是可以觀察到下圖，如果再加入 Knowledge Distillation，validation 的 accuracy 甚至超越了原先未加 Depthwise & Pointwise 的 Model Architecture，代表著，除了縮小了大量的參數量，準確率還提高，表現甚為優異。



圖四、KD(藍) v.s Low Rank Approximation(橘) v.s Model Architecture(no DP)(灰)

2. [Knowledge Distillation] 請嘗試比較以下 validation accuracy (兩個 Teacher Net 由助教提供)以及 student 的總參數量以及架構，並嘗試解釋為甚麼有這樣的結果。你的 Student Net 的參數量必須要小於 Teacher Net 的參數量。(2%)

   x. Teacher net architecture and # of parameters: torchvision’s ResNet18, with 11,182,155 parameters.

   y. Student net architecture and # of parameters:

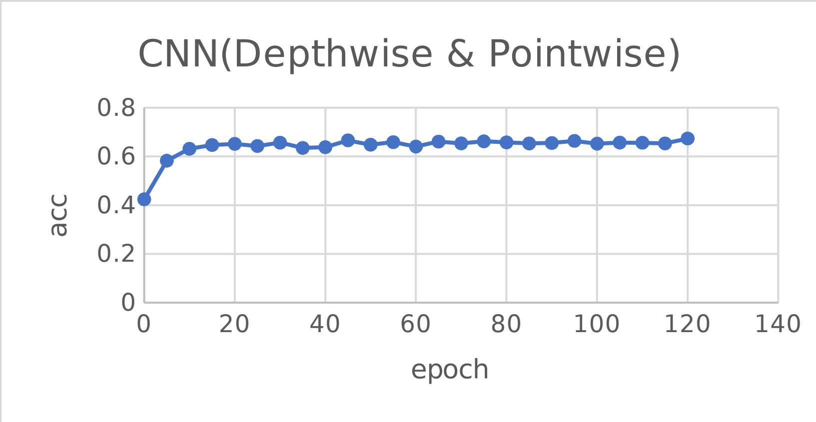
```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 16, 128, 128]             448
       BatchNorm2d-2         [-1, 16, 128, 128]              32
             ReLU6-3         [-1, 16, 128, 128]               0
         MaxPool2d-4           [-1, 16, 64, 64]               0
            Conv2d-5           [-1, 16, 64, 64]             160
       BatchNorm2d-6           [-1, 16, 64, 64]              32
             ReLU6-7           [-1, 16, 64, 64]               0
            Conv2d-8           [-1, 32, 64, 64]             544
         MaxPool2d-9           [-1, 32, 32, 32]               0
           Conv2d-10           [-1, 32, 32, 32]             320
      BatchNorm2d-11           [-1, 32, 32, 32]              64
            ReLU6-12           [-1, 32, 32, 32]               0
           Conv2d-13           [-1, 64, 32, 32]           2,112
        MaxPool2d-14           [-1, 64, 16, 16]               0
           Conv2d-15           [-1, 64, 16, 16]             640
      BatchNorm2d-16           [-1, 64, 16, 16]             128
            ReLU6-17           [-1, 64, 16, 16]               0
           Conv2d-18          [-1, 128, 16, 16]           8,320
        MaxPool2d-19            [-1, 128, 8, 8]               0
           Conv2d-20            [-1, 128, 8, 8]           1,280
      BatchNorm2d-21            [-1, 128, 8, 8]             256
            ReLU6-22            [-1, 128, 8, 8]               0
           Conv2d-23            [-1, 256, 8, 8]          33,024
           Conv2d-24            [-1, 256, 8, 8]           2,560
      BatchNorm2d-25            [-1, 256, 8, 8]             512
            ReLU6-26            [-1, 256, 8, 8]               0
           Conv2d-27            [-1, 256, 8, 8]          65,792
           Conv2d-28            [-1, 256, 8, 8]           2,560
      BatchNorm2d-29            [-1, 256, 8, 8]             512
            ReLU6-30            [-1, 256, 8, 8]               0
           Conv2d-31            [-1, 256, 8, 8]          65,792
           Conv2d-32            [-1, 256, 8, 8]           2,560
      BatchNorm2d-33            [-1, 256, 8, 8]             512
            ReLU6-34            [-1, 256, 8, 8]               0
           Conv2d-35            [-1, 256, 8, 8]          65,792
daptiveAvgPool2d-36            [-1, 256, 1, 1]               0
           Linear-37                   [-1, 11]           2,827
================================================================
otal params: 256,779
rainable params: 256,779
on-trainable params: 0
----------------------------------------------------------------
nput size (MB): 0.19
orward/backward pass size (MB): 13.13
arams size (MB): 0.98
```

圖五、Student net architecture and # of parameters

a. Teacher net (ResNet18) from scratch: 80.09%

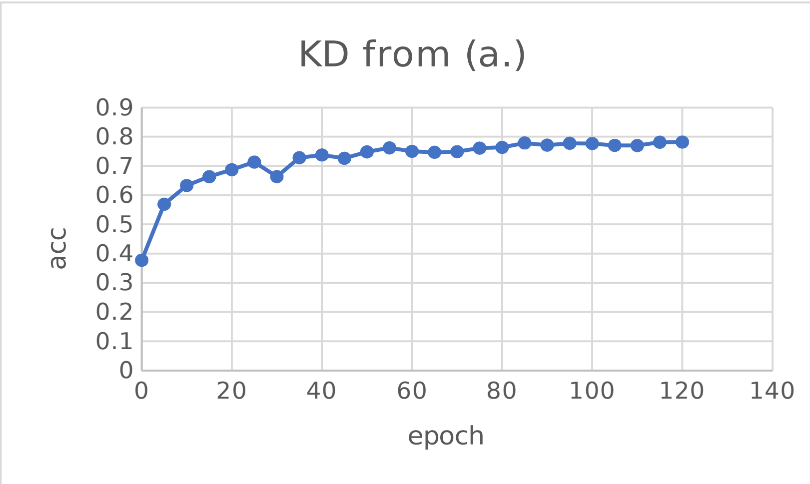
b. Teacher net (ResNet18) ImageNet pretrained & fine-tune: 88.41%

c. Your student net from scratch:



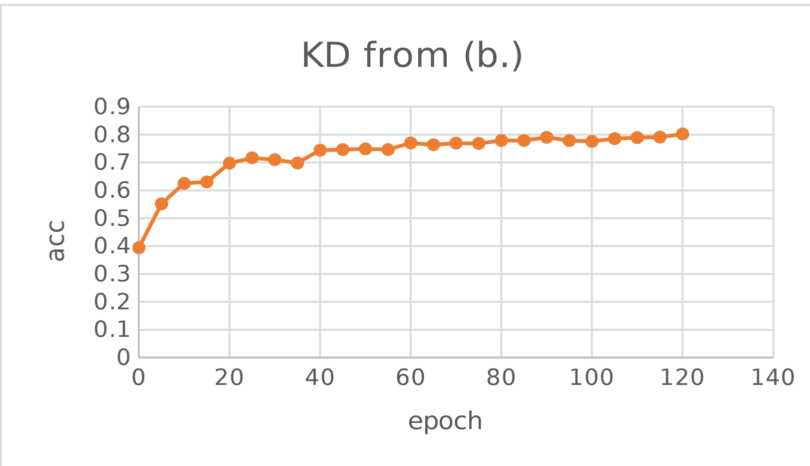
圖六、student net from scratch

d. Your student net KD from (a.):

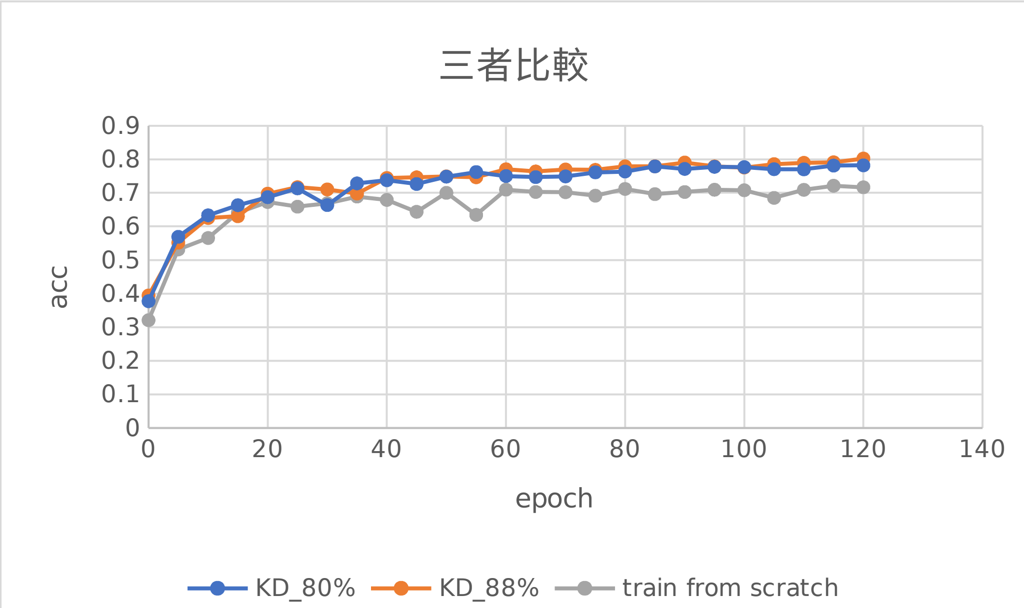


圖七、student net from KD(Teacher net (ResNet18) from scratch :80.09%)

e. Your student net KD from (b.):



圖八、student net from KD(Teacher net (ResNet18) ImageNet pretrained & fine-tune: 88.41%)



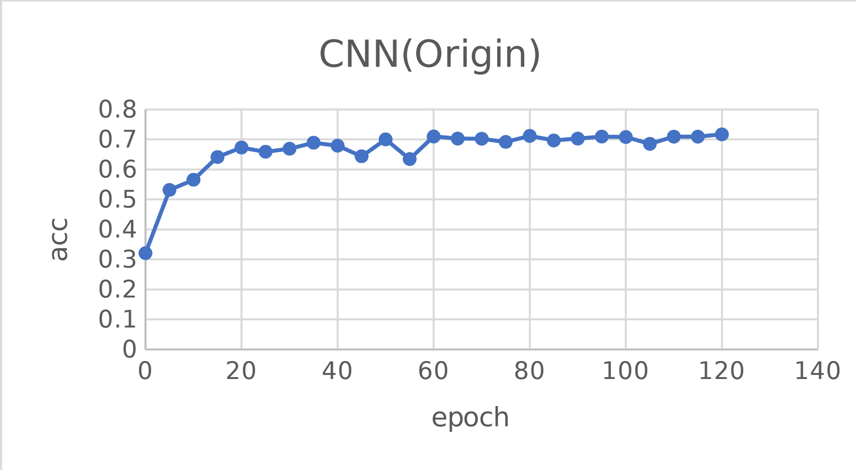
圖九、三者比較

由圖九可發現，即使進行了 Fine-Tune 後的 Teacher Net Accuracy 明顯的高於 From Scratch 後的 Teacher Net Accuracy，但在 Student Net 的 Accuracy 卻未上升多少，可能的原因是 Fine-Tune 的學習資訊在 Student Net 的學習過程中，並未能有效學習，或許 Fine-Tune 的學習資訊量過於龐大，無法讓較小的 Student Net 完全的學習，也或許 Student Net 的架構本身因與 ResNet18 From Scratch 相像，故學會了 From Scratch 大部分後便無法再進行以外的知識學習。而

Student Net From Scratch 則因為缺少了大 model 的許多資訊，故無法學習到大 model 中許多大量參數底下的更多資訊。

4. [Low Rank Approximation / Model Architecture] 請嘗試比較以下 validation accuracy，並且模型大小須接近 1 MB。 (2%)

a. 原始 CNN model (用一般的 Convolution Layer) 的 accuracy

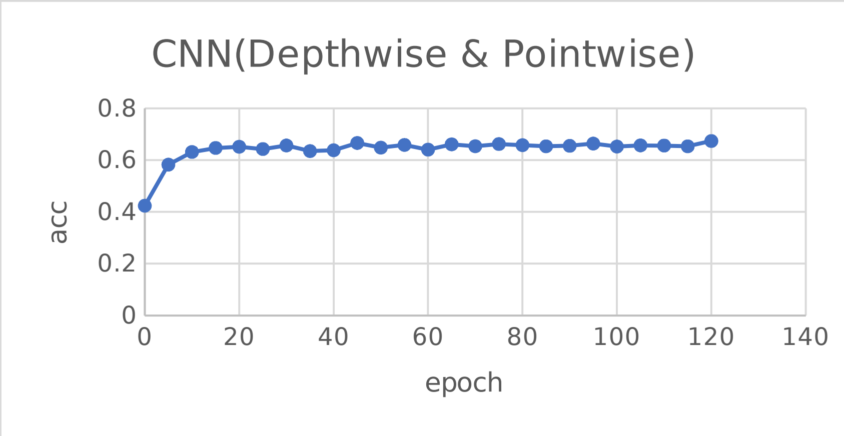


圖十、CNN(Origin) Validation Accuracy

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 16, 128, 128]             448
       BatchNorm2d-2         [-1, 16, 128, 128]              32
             ReLU6-3         [-1, 16, 128, 128]               0
         MaxPool2d-4           [-1, 16, 64, 64]               0
            Conv2d-5           [-1, 32, 64, 64]           4,640
       BatchNorm2d-6           [-1, 32, 64, 64]              64
             ReLU6-7           [-1, 32, 64, 64]               0
         MaxPool2d-8           [-1, 32, 32, 32]               0
            Conv2d-9           [-1, 32, 32, 32]           9,248
      BatchNorm2d-10           [-1, 32, 32, 32]              64
            ReLU6-11           [-1, 32, 32, 32]               0
        MaxPool2d-12           [-1, 32, 16, 16]               0
           Conv2d-13           [-1, 64, 16, 16]          18,496
      BatchNorm2d-14           [-1, 64, 16, 16]             128
            ReLU6-15           [-1, 64, 16, 16]               0
        MaxPool2d-16             [-1, 64, 8, 8]               0
           Conv2d-17             [-1, 64, 8, 8]          36,928
      BatchNorm2d-18             [-1, 64, 8, 8]             128
            ReLU6-19             [-1, 64, 8, 8]               0
           Conv2d-20             [-1, 64, 8, 8]          36,928
      BatchNorm2d-21             [-1, 64, 8, 8]             128
            ReLU6-22             [-1, 64, 8, 8]               0
           Conv2d-23             [-1, 96, 8, 8]          55,392
      BatchNorm2d-24             [-1, 96, 8, 8]             192
            ReLU6-25             [-1, 96, 8, 8]               0
           Conv2d-26            [-1, 128, 8, 8]         110,720
      BatchNorm2d-27            [-1, 128, 8, 8]             256
            ReLU6-28            [-1, 128, 8, 8]               0
AdaptiveAvgPool2d-29            [-1, 128, 1, 1]               0
           Linear-30                   [-1, 11]           1,419
================================================================
Total params: 275,211
Trainable params: 275,211
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.19
Forward/backward pass size (MB): 11.49
Params size (MB): 1.05
Estimated Total Size (MB): 12.72
```

圖十一、CNN(Origin) Classifier structure

b. 將 CNN model 的 Convolution Layer 換成參數量接近的 Depthwise & Pointwise 後的 accuracy

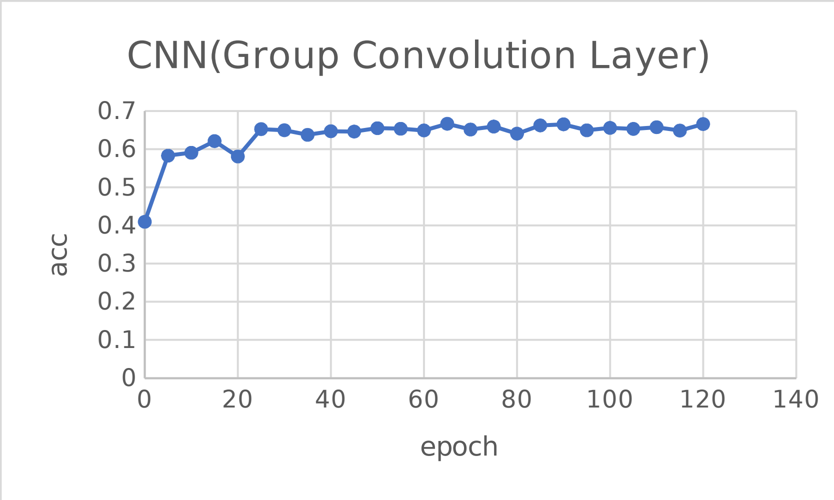


圖十二、CNN(Depthwise & Pointwise) Validation Accuracy

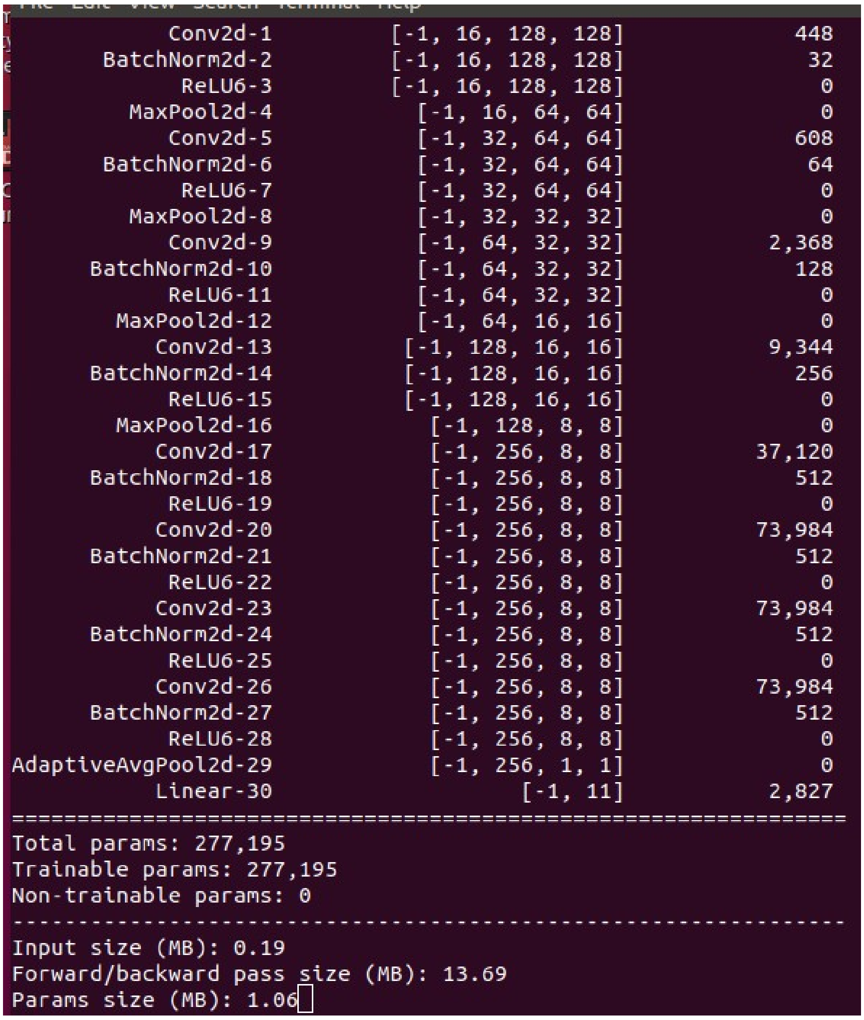
```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1        [-1, 16, 128, 128]             448
       BatchNorm2d-2        [-1, 16, 128, 128]              32
             ReLU6-3        [-1, 16, 128, 128]               0
         MaxPool2d-4          [-1, 16, 64, 64]               0
            Conv2d-5          [-1, 16, 64, 64]             160
       BatchNorm2d-6          [-1, 16, 64, 64]              32
             ReLU6-7          [-1, 16, 64, 64]               0
            Conv2d-8          [-1, 32, 64, 64]             544
         MaxPool2d-9          [-1, 32, 32, 32]               0
           Conv2d-10          [-1, 32, 32, 32]             320
      BatchNorm2d-11          [-1, 32, 32, 32]              64
            ReLU6-12          [-1, 32, 32, 32]               0
           Conv2d-13          [-1, 64, 32, 32]           2,112
        MaxPool2d-14          [-1, 64, 16, 16]               0
           Conv2d-15          [-1, 64, 16, 16]             640
      BatchNorm2d-16          [-1, 64, 16, 16]             128
            ReLU6-17          [-1, 64, 16, 16]               0
           Conv2d-18         [-1, 128, 16, 16]           8,320
        MaxPool2d-19           [-1, 128, 8, 8]               0
           Conv2d-20           [-1, 128, 8, 8]           1,280
      BatchNorm2d-21           [-1, 128, 8, 8]             256
            ReLU6-22           [-1, 128, 8, 8]               0
           Conv2d-23           [-1, 256, 8, 8]          33,024
           Conv2d-24           [-1, 256, 8, 8]           2,560
      BatchNorm2d-25           [-1, 256, 8, 8]             512
            ReLU6-26           [-1, 256, 8, 8]               0
           Conv2d-27           [-1, 256, 8, 8]          65,792
           Conv2d-28           [-1, 256, 8, 8]           2,560
      BatchNorm2d-29           [-1, 256, 8, 8]             512
            ReLU6-30           [-1, 256, 8, 8]               0
           Conv2d-31           [-1, 256, 8, 8]          65,792
           Conv2d-32           [-1, 256, 8, 8]           2,560
      BatchNorm2d-33           [-1, 256, 8, 8]             512
            ReLU6-34           [-1, 256, 8, 8]               0
           Conv2d-35           [-1, 256, 8, 8]          65,792
AdaptiveAvgPool2d-36           [-1, 256, 1, 1]               0
           Linear-37                  [-1, 11]           2,827
================================================================
Total params: 256,779
Trainable params: 256,779
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.19
Forward/backward pass size (MB): 13.13
Params size (MB): 0.98
```

圖十三、CNN(Depthwise & Pointwise) Classifier structure

c. 將 CNN model 的 Convolution Layer 換成參數量接近的 Group Convolution Layer (Group 數量自訂，但不要設為 1 或 in_filters)

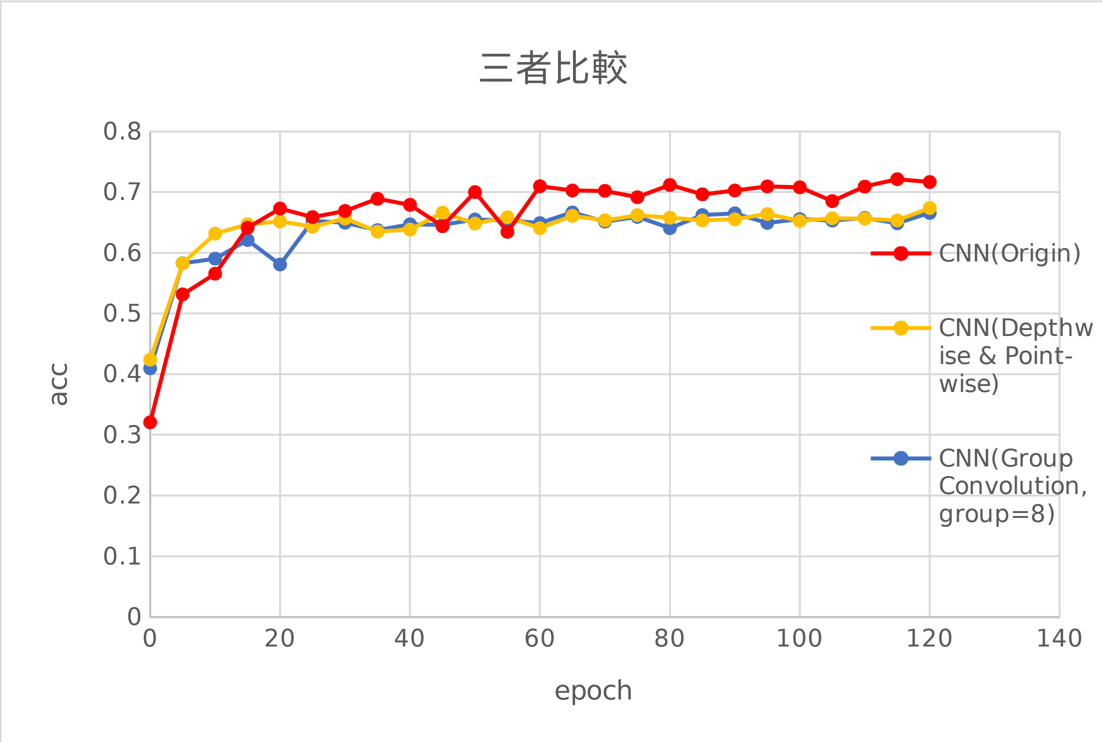


圖十四、CNN(Group Convolution Layer, group = 8) Validation Accuracy



```
          Conv2d-1        [-1, 16, 128, 128]             448
     BatchNorm2d-2        [-1, 16, 128, 128]              32
           ReLU6-3        [-1, 16, 128, 128]               0
       MaxPool2d-4          [-1, 16, 64, 64]               0
          Conv2d-5          [-1, 32, 64, 64]             608
     BatchNorm2d-6          [-1, 32, 64, 64]              64
           ReLU6-7          [-1, 32, 64, 64]               0
       MaxPool2d-8          [-1, 32, 32, 32]               0
          Conv2d-9          [-1, 64, 32, 32]           2,368
    BatchNorm2d-10          [-1, 64, 32, 32]             128
          ReLU6-11          [-1, 64, 32, 32]               0
      MaxPool2d-12          [-1, 64, 16, 16]               0
         Conv2d-13         [-1, 128, 16, 16]           9,344
    BatchNorm2d-14         [-1, 128, 16, 16]             256
          ReLU6-15         [-1, 128, 16, 16]               0
      MaxPool2d-16           [-1, 128, 8, 8]               0
         Conv2d-17           [-1, 256, 8, 8]          37,120
    BatchNorm2d-18           [-1, 256, 8, 8]             512
          ReLU6-19           [-1, 256, 8, 8]               0
         Conv2d-20           [-1, 256, 8, 8]          73,984
    BatchNorm2d-21           [-1, 256, 8, 8]             512
          ReLU6-22           [-1, 256, 8, 8]               0
         Conv2d-23           [-1, 256, 8, 8]          73,984
    BatchNorm2d-24           [-1, 256, 8, 8]             512
          ReLU6-25           [-1, 256, 8, 8]               0
         Conv2d-26           [-1, 256, 8, 8]          73,984
    BatchNorm2d-27           [-1, 256, 8, 8]             512
          ReLU6-28           [-1, 256, 8, 8]               0
AdaptiveAvgPool2d-29         [-1, 256, 1, 1]               0
         Linear-30                  [-1, 11]           2,827
================================================================
Total params: 277,195
Trainable params: 277,195
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.19
Forward/backward pass size (MB): 13.69
Params size (MB): 1.06
```

圖十五、CNN(Group Convolution Layer, group = 8) Classifier structure



圖十六、三者 Validation Accuracy

由此圖可以觀察到，在 Depthwise & Pointwise 以及 Group Convolution 當中，得到了相近的結果，可能原因應該是 Group Convolution 和 Depthwise & Pointwise 所做的事情相似，假設上一層的 feature map 總共有 N 個，先將 channel 分成 M 份。每一個 group 對應 N/M 個 channel，與之獨立相連。然後上層 group 卷積完成後將輸出疊在一起（concatenate），作為這一層的輸出 channel。達成減少參數量，≒1/M 倍。故所得到的 Validation Accuracy 相近。

而比較讓人意外的是 CNN (Origin) train from scratch，在相近的參數下，層數與 Group Convolution 相近甚至低於 Depthwise & Pointwise 的情況下，Validation Accuracy 卻高於另外兩者，可能原因是，在我寫的 CNN (Origin) 架構底下，參數量仍然很龐大，所以即使和大 model 的 Low Rank Approximation 相比，他本身就仍有好表現的的機會，又或者，如同樂透一樣，就那麼剛好 train 到一個較好的小 model。除此之外，和作業三比較後，亦觀察到瘦高形的 model 會有不錯的結果。

最後，再附上丟到 kaggle 的值。

	第一題		第四題		
	Low Rank Approximation	Knowledge Distillation	CNN(Group Convolution Layer)	CNN(Depthwise & Pointwise)	CNN (Origin)
acc	0.71010	0.83801	0.71607	0.76867	0.77047