

1. 請比較你本次作業的架構，參數量、結果和原 HW3 作業架構、參數量、結果做比較。(1%)

本次作業的模型主要由 **conv_bn** 和 **conv_dw** 構成，這兩個模塊的架構如下：

conv_bn(input, output, stride)

```
(0): Conv2d(input, output, kernel_size=(3, 3), stride=stride, padding=(1, 1), bias=False)
(1): BatchNorm2d(output, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(2): ReLU(inplace)
```

conv_dw(input, output, stride)

```
(0): Conv2d(input, input, kernel_size=(3, 3), stride=stride, padding=(1, 1), groups=input, bias=False)
(1): BatchNorm2d(input, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(2): ReLU(inplace)
(3): Conv2d(input, output, kernel_size=(1, 1), stride=(1, 1), bias=False)
(4): BatchNorm2d(output, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(5): ReLU(inplace)
```

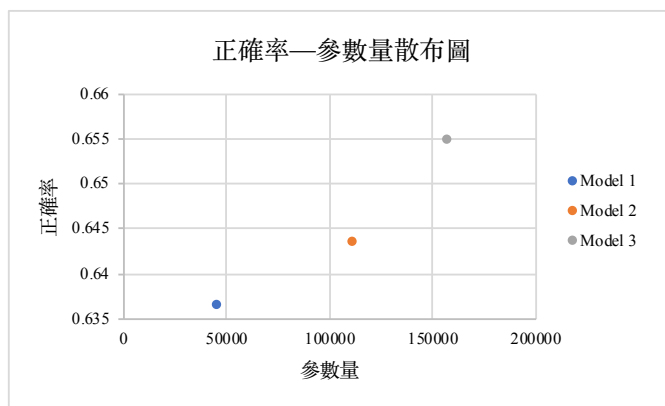
	本次作業	HW3
模型架構	<p>使用 MobileNet</p> <pre>conv_bn(1, 32, 1) conv_bn(32, 32, 1) MaxPool2d(2) Dropout(0.25) conv_dw(32, 64, 1) conv_dw(64, 64, 1) conv_dw(64, 80, 2) conv_dw(80, 80, 1) conv_dw(80, 80, 2) conv_dw(80, 80, 1) AvgPool2d(6) Dropout(0.5) Linear(80, 7)</pre>	<p>使用簡化版的 VGG</p> <pre>Conv2D(64, (3, 3), padding = 'same') BatchNormalization Activation('relu') Conv2D(64, (3, 3)) BatchNormalization Activation('relu') MaxPooling2D(2, 2) Dropout(0.25) Conv2D(128, (3, 3), padding = 'same') Activation('relu') Conv2D(128, (3, 3)) Activation('relu') MaxPooling2D(2, 2) Dropout(0.25) Conv2D(128, (3, 3), padding = 'same') BatchNormalization Activation('relu') Conv2D(128, (3, 3)) BatchNormalization Activation('relu') MaxPooling2D(2, 2) Dropout(0.25) Flatten Dense(512) BatchNormalization Activation('relu') Dropout(0.35) Dense(512) BatchNormalization Activation('relu') Dropout(0.35) Dense(7) Activation('softmax')</pre>
參數量	45,959	1,875,143
正確率	Public 0.63109, Private 0.62803	Public 0.68487, Private 0.68208

HW3 使用的模型架構較複雜，參數量約為本次作業的 40 倍，正確率也較高。本次作業使用簡單的模型，用相當少的參數量就能達到一定的正確率。

2. 請使用 MobileNet 的架構，畫出參數量-acc 的散布圖（橫軸為參數量，縱軸為 accuracy，且至少 3 個點，參數量選擇時儘量不要離的太近，結果選擇只要大致收斂，不用 train 到最好沒關係。）(1%)

	參數量	正確率
Model 1	45959	0.636364
Model 2	111751	0.643330
Model 3	157831	0.654824

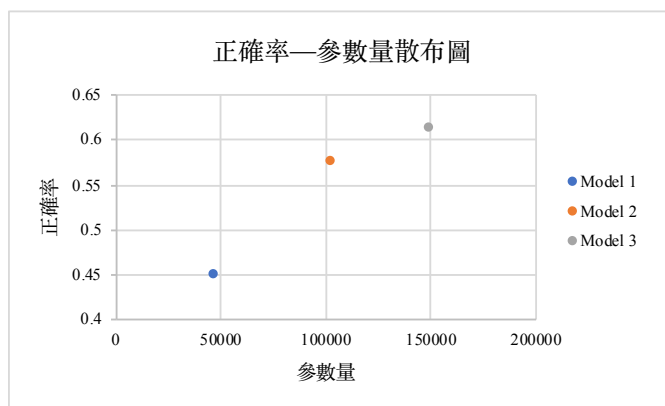
正確率為自己切的 validation set 上的正確率。



3. 請使用一般 CNN 的架構，畫出參數量-acc 的散布圖（橫軸為參數量，縱軸為 accuracy，且至少 3 個點，參數量選擇時儘量不要離的太近，結果選擇只要大致收斂，不用 train 到最好沒關係。）(1%)

	參數量	正確率
Model 1	47511	0.449321
Model 2	103079	0.576106
Model 3	149575	0.612679
Model 4	851911	0.655869

正確率為自己切的 validation set 上的正確率。



Model 1 ~ 3 的參數量和第 2 題接近，為方便比較，散布圖只畫出這三個 model 的分佈。Model 4 是為了比較達到和第 2 題接近的正確率所需要的參數量，因參數量遠大於 Model 1 ~ 3，所以並未畫在散布圖上。

4. 請你比較題 2 和題 3 的結果，並請針對當參數量相當少的時候，如果兩者參數量相當，兩者的差異，以及你認為為什麼會造成這個原因。(2%)

(1) 第 2 題和第 3 題的結果都顯示：參數量越多，正確率越高。

(2) 比較第 2 題和第 3 題的結果，Model 1~3 的參數量接近，使用 MobileNet 的正確率較高。當參數量相當少的時候（Model 1），使用 MobileNet 仍有一定的正確率（0.636364），但使用 CNN 的正確率相當低（0.449321）。

(3) 第 3 題的 Model 4 的正確率和第 2 題的 Model 3 相當，但前者的參數量（851911）遠多於後者（157831）。

MobileNet 的參數量少，在同樣參數量的限制下，可以做較多層的 convolution；而 CNN 的參數量較多，若限制使用相當少的參數，必須使用比 MobileNet 更少的 filter，linear layer 的 neuron 數目也必須更少，因此正確率較低。