學號：B06902030　　系級：資工二 姓名：邱譯

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：

(1) 架構與參數量

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 48, 48, 1)	0
conv2d_1 (Conv2D)	(None, 48, 48, 32)	288
dropout_1 (Dropout)	(None, 48, 48, 32)	0
batch_normalization_1 (Batch	(None, 48, 48, 32)	128
activation_1 (Activation)	(None, 48, 48, 32)	0
depthwise_conv2d_1 (Depthwis	(None, 48, 48, 32)	320
dropout_2 (Dropout)	(None, 48, 48, 32)	0
batch_normalization_2 (Batch	(None, 48, 48, 32)	128
activation_2 (Activation)	(None, 48, 48, 32)	0
conv2d_2 (Conv2D)	(None, 48, 48, 64)	2048
batch_normalization_3 (Batch	(None, 48, 48, 64)	256
activation_3 (Activation)	(None, 48, 48, 64)	0
depthwise_conv2d_2 (Depthwis	(None, 24, 24, 64)	640
dropout_3 (Dropout)	(None, 24, 24, 64)	0
batch_normalization_4 (Batch	(None, 24, 24, 64)	256
activation_4 (Activation)	(None, 24, 24, 64)	0
conv2d_3 (Conv2D)	(None, 24, 24, 64)	4096
batch_normalization_5 (Batch	(None, 24, 24, 64)	256
activation_5 (Activation)	(None, 24, 24, 64)	0
depthwise_conv2d_3 (Depthwis	(None, 24, 24, 64)	640
dropout_4 (Dropout)	(None, 24, 24, 64)	0
batch_normalization_6 (Batch	(None, 24, 24, 64)	256
activation_6 (Activation)	(None, 24, 24, 64)	0
conv2d_4 (Conv2D)	(None, 24, 24, 64)	4096
batch_normalization_7 (Batch	(None, 24, 24, 64)	256
activation_7 (Activation)	(None, 24, 24, 64)	0
depthwise_conv2d_4 (Depthwis	(None, 12, 12, 64)	640
dropout_5 (Dropout)	(None, 12, 12, 64)	0
batch_normalization_8 (Batch	(None, 12, 12, 64)	256
activation_8 (Activation)	(None, 12, 12, 64)	0
conv2d_5 (Conv2D)	(None, 12, 12, 64)	4096
batch_normalization_9 (Batch	(None, 12, 12, 64)	256
activation_9 (Activation)	(None, 12, 12, 64)	0
depthwise_conv2d_5 (Depthwis	(None, 12, 12, 64)	640
dropout_6 (Dropout)	(None, 12, 12, 64)	0
batch_normalization_10 (Batc	(None, 12, 12, 64)	256
activation_10 (Activation)	(None, 12, 12, 64)	0
conv2d_6 (Conv2D)	(None, 12, 12, 128)	8192
batch_normalization_11 (Batc	(None, 12, 12, 128)	512
activation_11 (Activation)	(None, 12, 12, 128)	0
depthwise_conv2d_6 (Depthwis	(None, 6, 6, 128)	1280
dropout_7 (Dropout)	(None, 6, 6, 128)	0
batch_normalization_12 (Batc	(None, 6, 6, 128)	512
activation_12 (Activation)	(None, 6, 6, 128)	0
conv2d_7 (Conv2D)	(None, 6, 6, 128)	16384
batch_normalization_13 (Batc	(None, 6, 6, 128)	512
activation_13 (Activation)	(None, 6, 6, 128)	0
depthwise_conv2d_7 (Depthwis	(None, 6, 6, 128)	1280
dropout_8 (Dropout)	(None, 6, 6, 128)	0
batch_normalization_14 (Batc	(None, 6, 6, 128)	512
activation_14 (Activation)	(None, 6, 6, 128)	0
conv2d_8 (Conv2D)	(None, 6, 6, 128)	16384
batch_normalization_15 (Batc	(None, 6, 6, 128)	512
activation_15 (Activation)	(None, 6, 6, 128)	0
global_max_pooling2d_1 (Glob	(None, 128)	0
dense_1 (Dense)	(None, 56)	7224
dense_2 (Dense)	(None, 7)	399

Total params: 73,511
Trainable params: 71,079
Non-trainable params: 2,432

(2) 正確率

Private: 0.63973

Public: 0.64781

HW3：

(1) 架構與參數量

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 128)	1280
conv2d_2 (Conv2D)	(None, 48, 48, 128)	147584
batch_normalization_1 (Batch	(None, 48, 48, 128)	512
max_pooling2d_1 (MaxPooling2	(None, 24, 24, 128)	0
dropout_1 (Dropout)	(None, 24, 24, 128)	0
conv2d_3 (Conv2D)	(None, 24, 24, 256)	295168
conv2d_4 (Conv2D)	(None, 24, 24, 256)	590080
batch_normalization_2 (Batch	(None, 24, 24, 256)	1024
max_pooling2d_2 (MaxPooling2	(None, 12, 12, 256)	0
dropout_2 (Dropout)	(None, 12, 12, 256)	0
conv2d_5 (Conv2D)	(None, 12, 12, 512)	1180160
conv2d_6 (Conv2D)	(None, 12, 12, 512)	2359808
batch_normalization_3 (Batch	(None, 12, 12, 512)	2048
max_pooling2d_3 (MaxPooling2	(None, 6, 6, 512)	0
dropout_3 (Dropout)	(None, 6, 6, 512)	0
conv2d_7 (Conv2D)	(None, 6, 6, 768)	3539712
conv2d_8 (Conv2D)	(None, 6, 6, 768)	5309184
batch_normalization_4 (Batch	(None, 6, 6, 768)	3072
max_pooling2d_4 (MaxPooling2	(None, 3, 3, 768)	0
dropout_4 (Dropout)	(None, 3, 3, 768)	0
flatten_1 (Flatten)	(None, 6912)	0
dense_1 (Dense)	(None, 1024)	7078912
dropout_5 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
dropout_6 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 1024)	1049600
dropout_7 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 7)	7175

Total params: 22,614,919
Trainable params: 22,611,591
Non-trainable params: 3,328

(2) 正確率

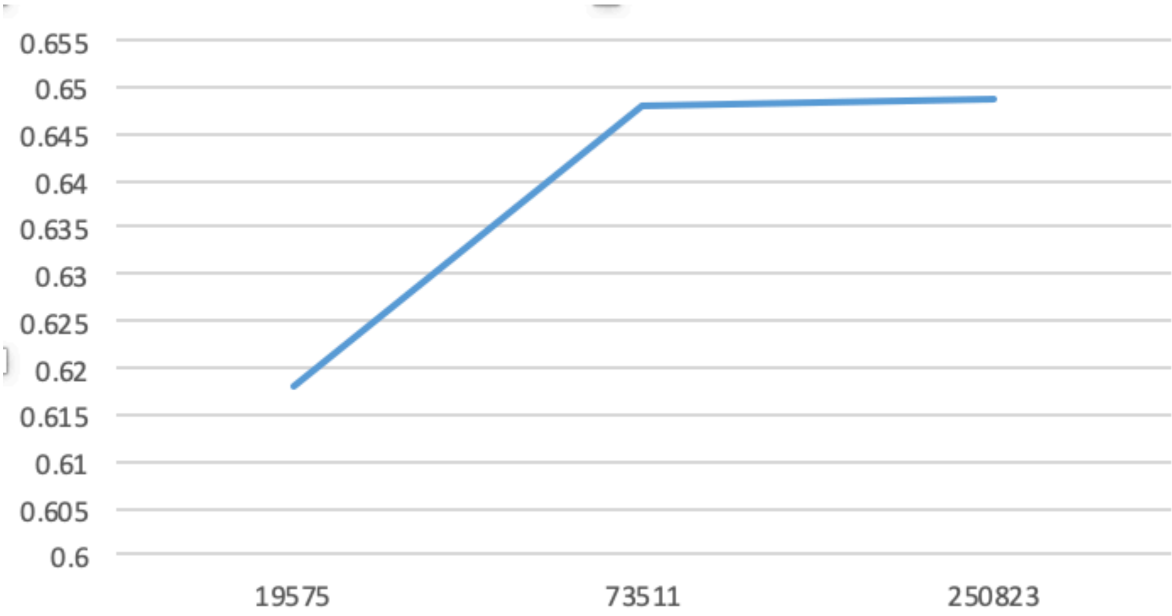
Private: 0.68487

Public: 0.68208

可以發現我在HW3的參數量比HW8多上非常多，約為300倍，但正確率卻只高了約4%，可見model compressing是有效的。

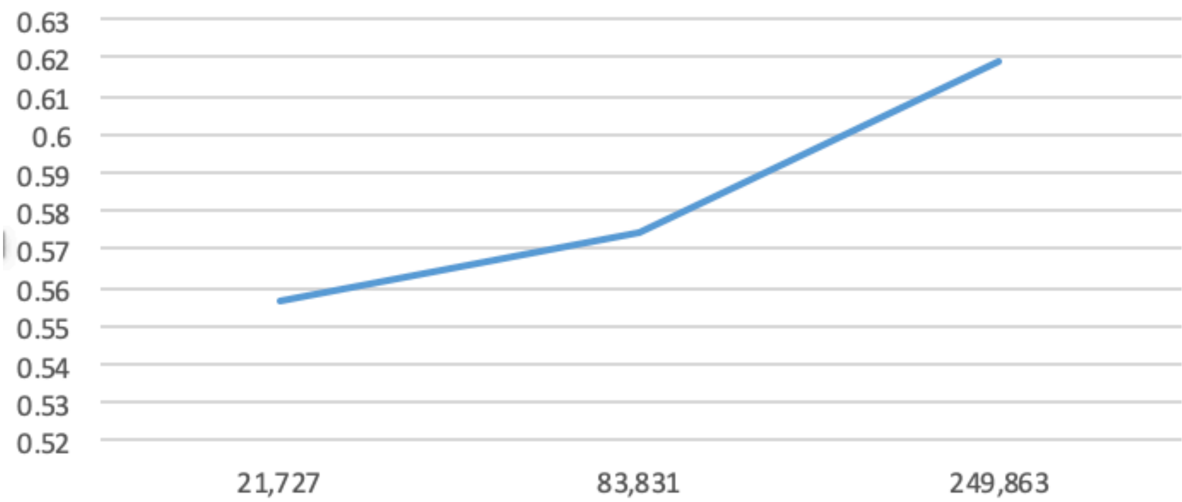
2. 請使用MobileNet的架構，畫出參數量-acc的散布圖（橫軸為參數量，縱軸為accuracy，且至少3個點，參數量選擇時儘量不要離的太近，結果選擇只要大致收斂，不用train到最好沒關係。）(1%)

   橫軸為參數量，縱軸為在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%)

   橫軸為參數量，縱軸為在validation set上的正確率



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%)

   當參數量相當少時，MobileNet能有比一般CNN高的正確率，我認為是因為MobileNet是使用Depthwise Separable Convolution，如此能透過在某些地方共用參數來減少參數量，雖然會使得取出的特徵較接近，但能夠在參數量少的狀況下取出許多的特徵，相比於一般CNN，在參數量少時，只能夠取出少量特徵，因此正確率會下降很多。

   此外，可以看到當我們增加參數量時，一般CNN的正確率逐漸上升且有越來越好的趨勢，但MobileNet的正確率在一定程度上升後開始停滯，可見MobileNet比較適合在參數量少的狀況，再繼續增加參數量也沒有辦法明顯提高正確率。