學號：B06901087 系級：電機二 姓名：翁瑋襄

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%)

HW8架構:

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 46, 46, 16)	160
separable_conv2d_1 (Separabl	(None, 46, 46, 32)	688
batch_normalization_1 (Batch	(None, 46, 46, 32)	128
conv2d_2 (Conv2D)	(None, 46, 46, 48)	13872
separable_conv2d_2 (Separabl	(None, 46, 46, 64)	3568
batch_normalization_2 (Batch	(None, 46, 46, 64)	256
max_pooling2d_1 (MaxPooling2	(None, 23, 23, 64)	0
conv2d_3 (Conv2D)	(None, 23, 23, 96)	55392
separable_conv2d_3 (Separabl	(None, 23, 23, 128)	13280
batch_normalization_3 (Batch	(None, 23, 23, 128)	512
max_pooling2d_2 (MaxPooling2	(None, 11, 11, 128)	0
separable_conv2d_4 (Separabl	(None, 11, 11, 128)	17664
batch_normalization_4 (Batch	(None, 11, 11, 128)	512
max_pooling2d_3 (MaxPooling2	(None, 5, 5, 128)	0
dropout_1 (Dropout)	(None, 5, 5, 128)	0

```
flatten_1 (Flatten)          (None, 3200)            0
_________________________________________________________________
dense_1 (Dense)              (None, 7)               22407
=================================================================
Total params: 128,439
Trainable params: 127,735
Non-trainable params: 704
```

HW8 model參數: 128439
HW8準確率: public: 0.64112 / private: 0.64279

HW3架構:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 44, 44, 64)        1664
_________________________________________________________________
batch_normalization_1 (Batch (None, 44, 44, 64)        256
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 44, 44, 64)        102464
_________________________________________________________________
batch_normalization_2 (Batch (None, 44, 44, 64)        256
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 22, 22, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 22, 22, 128)       73856
_________________________________________________________________
batch_normalization_3 (Batch (None, 22, 22, 128)       512
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 22, 22, 128)       147584
_________________________________________________________________
batch_normalization_4 (Batch (None, 22, 22, 128)       512
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 11, 11, 128)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 11, 11, 128)       0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 11, 11, 128)       147584
```

batch_normalization_5 (Batch	(None, 11, 11, 128)	512
conv2d_6 (Conv2D)	(None, 11, 11, 128)	147584
batch_normalization_6 (Batch	(None, 11, 11, 128)	512
max_pooling2d_3 (MaxPooling2	(None, 5, 5, 128)	0
dropout_2 (Dropout)	(None, 5, 5, 128)	0
flatten_1 (Flatten)	(None, 3200)	0
dense_1 (Dense)	(None, 512)	1638912
batch_normalization_7 (Batch	(None, 512)	2048
dropout_3 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 512)	262656
batch_normalization_8 (Batch	(None, 512)	2048
dropout_4 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 7)	3591

===

Total params: 2,532,551
Trainable params: 2,529,223
Non-trainable params: 3,328

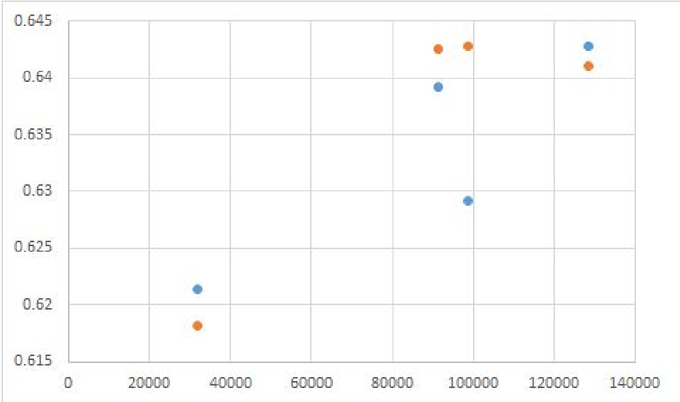
HW3 model參數: 2532551
HW3 準確率: public: 0.68765 / private: 0.68292

從這兩個model的架構跟參數量觀察，當model參數量較多時，比較容易fit training set，在predict testing set上也會有較好的表現(kaggle上預測準確率較高)

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

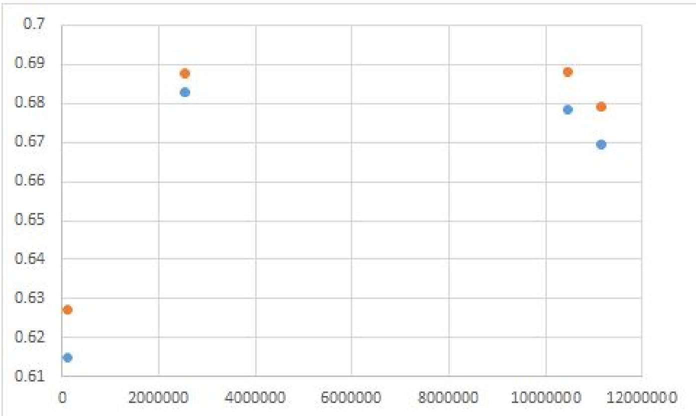
收斂，不用train到最好沒關係。）(1%)

圖中橘點為kaggle上public分數，藍點為private分數，當整體model參數量不太足夠的時候，model很難fit training data，所以準確率會下降。



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%)

圖中橘點為kaggle上public分數，藍點為private分數，可以觀察到當一般CNN架構參數量較少的時候，model很難fit training set，預測的準確率也相對較低；當一般CNN架構參數量滿多的時候，幾乎都足以fit training data，甚至可以發現不同的架構下參數量越多的預測準確率不見得比較高，因此推論一般CNN架構的model在參數量較多時，預測準確率和參數數量較無關。



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%)

mobilenet:

參數量: 128439　準確率: public: 0.64112 / private: 0.64279

CNN:

參數量: 115959　準確率: public: 0.61493 / private: 0.62727

上面結果顯示在參數量相當的時候，一般CNN架構的model比mobilenet的model預測準確率較低，我認為原因是每一層Conv2D所需的參數量比SeparableConv2D還要更多，在接近的參數量情況下，CNN能疊的Conv2D層數比mobilenet的SeparableConv2D層數少，CNN每一層filter的數量也會比較少，因此比較難fit training data，預測準確率也會相對較低。