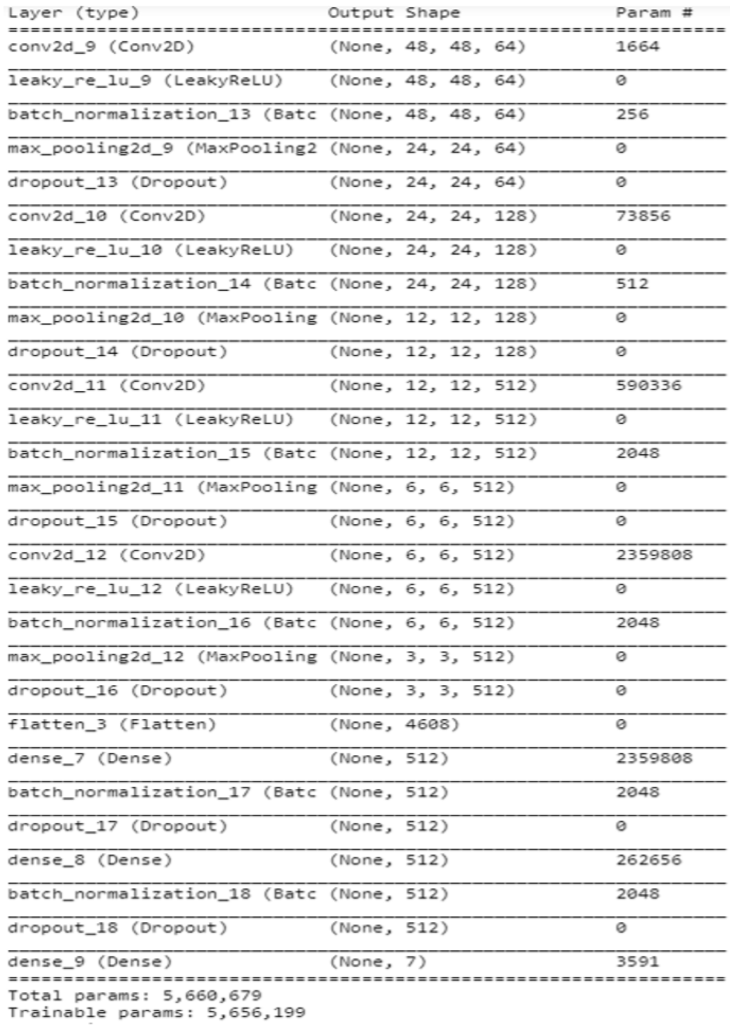
學號：B06502149 系級：資工二 姓名：張琦琛

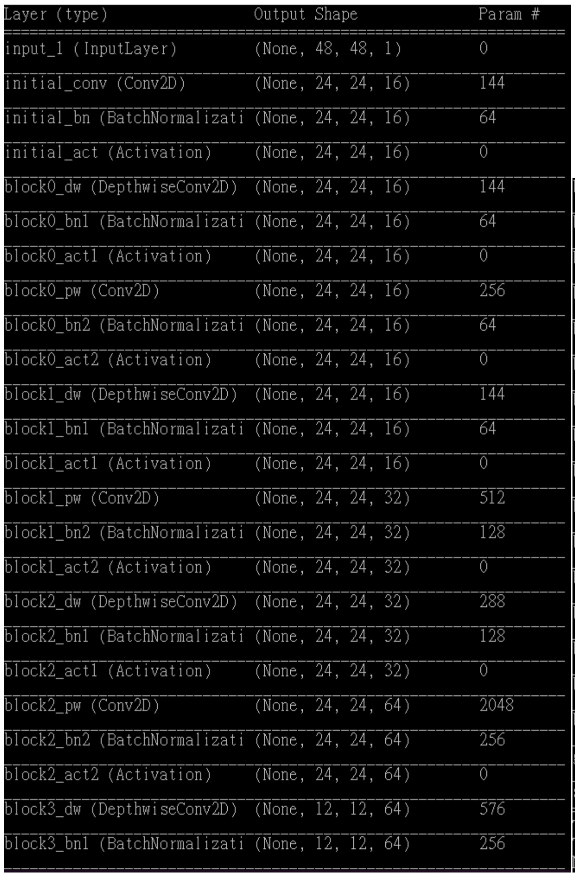
1. 請比較你本次作業的架構，參數量、結果和原 HW3 作業架構、參數量、結果做比較。(1%)

   HW3 Model 架構

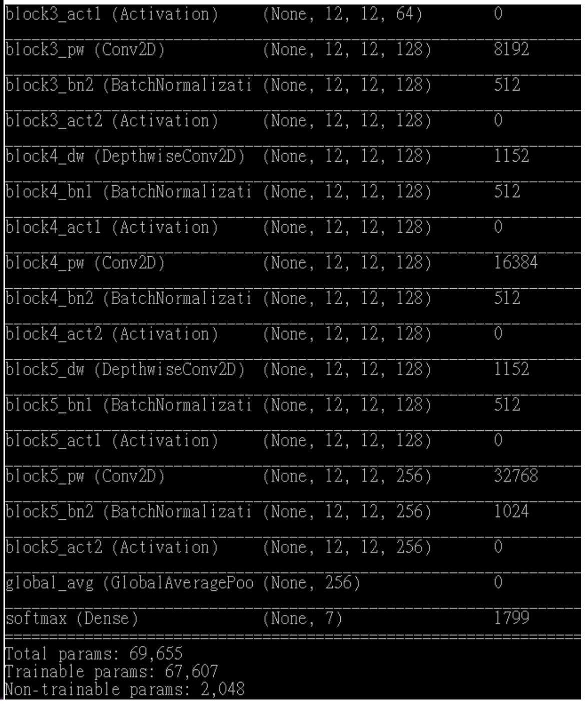


```
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_9 (Conv2D)            (None, 48, 48, 64)        1664      
_________________________________________________________________
leaky_re_lu_9 (LeakyReLU)    (None, 48, 48, 64)        0         
_________________________________________________________________
batch_normalization_13 (Batc (None, 48, 48, 64)        256       
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 24, 24, 64)        0         
_________________________________________________________________
dropout_13 (Dropout)         (None, 24, 24, 64)        0         
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 24, 24, 128)       73856     
_________________________________________________________________
leaky_re_lu_10 (LeakyReLU)   (None, 24, 24, 128)       0         
_________________________________________________________________
batch_normalization_14 (Batc (None, 24, 24, 128)       512       
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 12, 12, 128)       0         
_________________________________________________________________
dropout_14 (Dropout)         (None, 12, 12, 128)       0         
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 12, 12, 512)       590336    
_________________________________________________________________
leaky_re_lu_11 (LeakyReLU)   (None, 12, 12, 512)       0         
_________________________________________________________________
batch_normalization_15 (Batc (None, 12, 12, 512)       2048      
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 6, 6, 512)         0         
_________________________________________________________________
dropout_15 (Dropout)         (None, 6, 6, 512)         0         
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 6, 6, 512)         2359808   
_________________________________________________________________
leaky_re_lu_12 (LeakyReLU)   (None, 6, 6, 512)         0         
_________________________________________________________________
batch_normalization_16 (Batc (None, 6, 6, 512)         2048      
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 3, 3, 512)         0         
_________________________________________________________________
dropout_16 (Dropout)         (None, 3, 3, 512)         0         
_________________________________________________________________
flatten_3 (Flatten)          (None, 4608)              0         
_________________________________________________________________
dense_7 (Dense)              (None, 512)               2359808   
_________________________________________________________________
batch_normalization_17 (Batc (None, 512)               2048      
_________________________________________________________________
dropout_17 (Dropout)         (None, 512)               0         
_________________________________________________________________
dense_8 (Dense)              (None, 512)               262656    
_________________________________________________________________
batch_normalization_18 (Batc (None, 512)               2048      
_________________________________________________________________
dropout_18 (Dropout)         (None, 512)               0         
_________________________________________________________________
dense_9 (Dense)              (None, 7)                 3591      
=================================================================
Total params: 5,660,679
Trainable params: 5,656,199
```

HW8 Model 架構



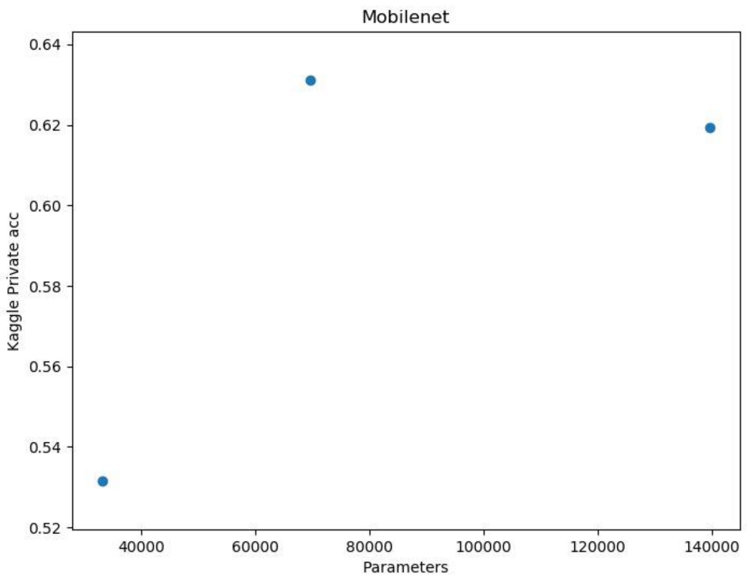
```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 48, 48, 1)         0
_________________________________________________________________
initial_conv (Conv2D)        (None, 24, 24, 16)        144
_________________________________________________________________
initial_bn (BatchNormalizati (None, 24, 24, 16)        64
_________________________________________________________________
initial_act (Activation)     (None, 24, 24, 16)        0
_________________________________________________________________
block0_dw (DepthwiseConv2D)  (None, 24, 24, 16)        144
_________________________________________________________________
block0_bn1 (BatchNormalizati (None, 24, 24, 16)        64
_________________________________________________________________
block0_act1 (Activation)     (None, 24, 24, 16)        0
_________________________________________________________________
block0_pw (Conv2D)           (None, 24, 24, 16)        256
_________________________________________________________________
block0_bn2 (BatchNormalizati (None, 24, 24, 16)        64
_________________________________________________________________
block0_act2 (Activation)     (None, 24, 24, 16)        0
_________________________________________________________________
block1_dw (DepthwiseConv2D)  (None, 24, 24, 16)        144
_________________________________________________________________
block1_bn1 (BatchNormalizati (None, 24, 24, 16)        64
_________________________________________________________________
block1_act1 (Activation)     (None, 24, 24, 16)        0
_________________________________________________________________
block1_pw (Conv2D)           (None, 24, 24, 32)        512
_________________________________________________________________
block1_bn2 (BatchNormalizati (None, 24, 24, 32)        128
_________________________________________________________________
block1_act2 (Activation)     (None, 24, 24, 32)        0
_________________________________________________________________
block2_dw (DepthwiseConv2D)  (None, 24, 24, 32)        288
_________________________________________________________________
block2_bn1 (BatchNormalizati (None, 24, 24, 32)        128
_________________________________________________________________
block2_act1 (Activation)     (None, 24, 24, 32)        0
_________________________________________________________________
block2_pw (Conv2D)           (None, 24, 24, 64)        2048
_________________________________________________________________
block2_bn2 (BatchNormalizati (None, 24, 24, 64)        256
_________________________________________________________________
block2_act2 (Activation)     (None, 24, 24, 64)        0
_________________________________________________________________
block3_dw (DepthwiseConv2D)  (None, 12, 12, 64)        576
_________________________________________________________________
block3_bn1 (BatchNormalizati (None, 12, 12, 64)        256
_________________________________________________________________
```



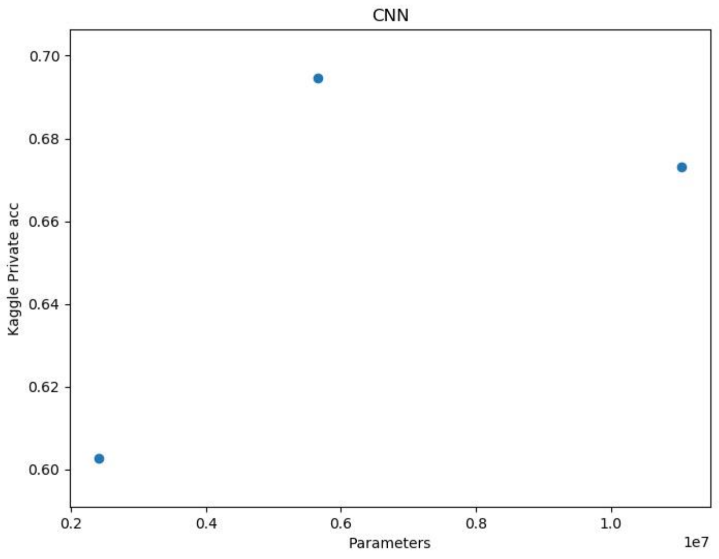
```
block3_act1 (Activation)     (None, 12, 12, 64)        0
_________________________________________________________________
block3_pw (Conv2D)           (None, 12, 12, 128)       8192
_________________________________________________________________
block3_bn2 (BatchNormalizati (None, 12, 12, 128)       512
_________________________________________________________________
block3_act2 (Activation)     (None, 12, 12, 128)       0
_________________________________________________________________
block4_dw (DepthwiseConv2D)  (None, 12, 12, 128)       1152
_________________________________________________________________
block4_bn1 (BatchNormalizati (None, 12, 12, 128)       512
_________________________________________________________________
block4_act1 (Activation)     (None, 12, 12, 128)       0
_________________________________________________________________
block4_pw (Conv2D)           (None, 12, 12, 128)       16384
_________________________________________________________________
block4_bn2 (BatchNormalizati (None, 12, 12, 128)       512
_________________________________________________________________
block4_act2 (Activation)     (None, 12, 12, 128)       0
_________________________________________________________________
block5_dw (DepthwiseConv2D)  (None, 12, 12, 128)       1152
_________________________________________________________________
block5_bn1 (BatchNormalizati (None, 12, 12, 128)       512
_________________________________________________________________
block5_act1 (Activation)     (None, 12, 12, 128)       0
_________________________________________________________________
block5_pw (Conv2D)           (None, 12, 12, 256)       32768
_________________________________________________________________
block5_bn2 (BatchNormalizati (None, 12, 12, 256)       1024
_________________________________________________________________
block5_act2 (Activation)     (None, 12, 12, 256)       0
_________________________________________________________________
global_avg (GlobalAveragePoo (None, 256)               0
_________________________________________________________________
softmax (Dense)              (None, 7)                 1799
=================================================================
Total params: 69,655
Trainable params: 67,607
Non-trainable params: 2,048
```

在 HW3 的 CNN 模型中我使用了 4 層 Convolution layer，在每一組都用了 LeakyReLU， BatchNormalization，DropOut，Filter 數量是越來越多個，為了避免 OverFitting， 我的 DropOut rate 也有慢慢增加，分別是 0.25、0.3、 0.35、0.35。最後再將圖壓平，進入 2 層的 Fully-connected-network，DropOut rate 設為 0.5，再利用 softmax 當作 Activation function，產生 7 種分類，總參數量為 5660679。而在 HW8 我參考 Mobilenet 的模型架構，一開始先用 16 個 filter 作一般的 convolution，之後慢慢增加 filter 數量作 Depthwise convolution，而每個 Depthwise convolution 後面都接一個 Pointwise convolution，總共 6 組。最後一層 filter 數量最多，共 256 個，總參數量為 69655。而我同樣也是用 softmax 當作 Activation function、Adam 當作 optimizer、categorical_crossentropy 計算 loss。我認為兩者最不一樣的地方在於 Mobilenet 用 Depthwise 加 Pointwise convolution 減少參數量，而且為了避免超過大小限制，我每層的 filter 數量相較於 HW3 的 CNN，每層少了將近 2-3 倍。HW3 在 Kaggle 的分數為 0.70075，HW8 為 0.63109。

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%)



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%)



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%)

從題 2 和題 3 的結果可以發現，HW3 的 CNN 模型看似有較好的準確率，雖然兩者最佳準確率的差距為 0.06353，但是兩者的參數量差距約為 560 萬，以 560 萬的代價換來的準確率似乎不算高。雖然兩張圖是不同的模型，但是從散佈圖的趨勢可以發現當參數量超過某個大小，準確率不升反降，必須不斷嘗試才能找出最好的模型架構。當兩者參數量皆約為 15 萬時，一般 CNN 模型的分數為 0.60406，Mobilenet 架構的模型為 0.61939;當參數量再下降到約 7 萬時，差距又更加明顯，一般 CNN 模型的分數為 0.56422，Mobilenet 架構的模型為 0.63109，並且我發現在 train 一般 CNN 時有 underfitting 的情況。我認為儘管兩者參數量相當，但對於普通 CNN 模型，filter 數量不能設太大且 model 深度較受限制，但對於 Mobilenet 這種是作 separable convolution 而言，可以增加多一點 layer 及 filter 數量，整體架構是高瘦型，而普通 CNN 大致是矮胖型，我在查詢網路上的資料後也發現高瘦型的 model 通常有較好的表現，因此我認為主要差異是模型的深度，導致 underfitting，若模型不夠深，就如線性函數無法準確的擬合一個以高次多項式函數產生的資料一樣。