學號：B02123456 系級： 電機四 姓名：涼宮春日

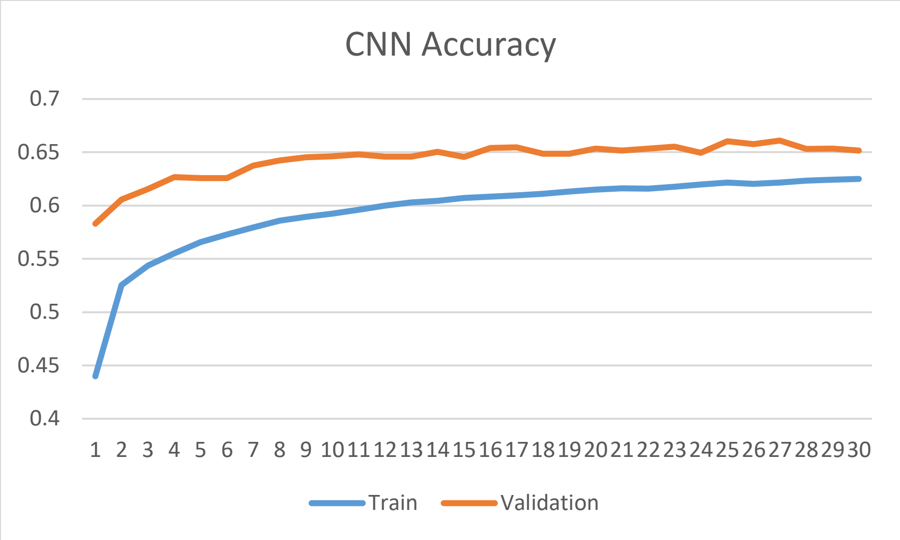
1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？
   答：

模型架構：

Layer (type)	Output Shape	Param #
batch_normalization_1 (Batch	(None, 48, 48, 1)	4
conv2d_1 (Conv2D)	(None, 46, 46, 32)	320
activation_1 (Activation)	(None, 46, 46, 32)	0
max_pooling2d_1 (MaxPooling2	(None, 23, 23, 32)	0
dropout_1 (Dropout)	(None, 23, 23, 32)	0
batch_normalization_2 (Batch	(None, 23, 23, 32)	128
conv2d_2 (Conv2D)	(None, 21, 21, 64)	18496
activation_2 (Activation)	(None, 21, 21, 64)	0
max_pooling2d_2 (MaxPooling2	(None, 10, 10, 64)	0
dropout_2 (Dropout)	(None, 10, 10, 64)	0
batch_normalization_3 (Batch	(None, 10, 10, 64)	256
conv2d_3 (Conv2D)	(None, 8, 8, 128)	73856
activation_3 (Activation)	(None, 8, 8, 128)	0
max_pooling2d_3 (MaxPooling2	(None, 4, 4, 128)	0
dropout_3 (Dropout)	(None, 4, 4, 128)	0
batch_normalization_4 (Batch	(None, 4, 4, 128)	512
conv2d_4 (Conv2D)	(None, 2, 2, 256)	295168
activation_4 (Activation)	(None, 2, 2, 256)	0
max_pooling2d_4 (MaxPooling2	(None, 1, 1, 256)	0
dropout_4 (Dropout)	(None, 1, 1, 256)	0
flatten_1 (Flatten)	(None, 256)	0
dense_1 (Dense)	(None, 2000)	514000
dropout_5 (Dropout)	(None, 2000)	0
dense_2 (Dense)	(None, 7)	14007

Total params: 916,747
Trainable params: 916,297
Non-trainable params: 450

訓練過程和準確率：

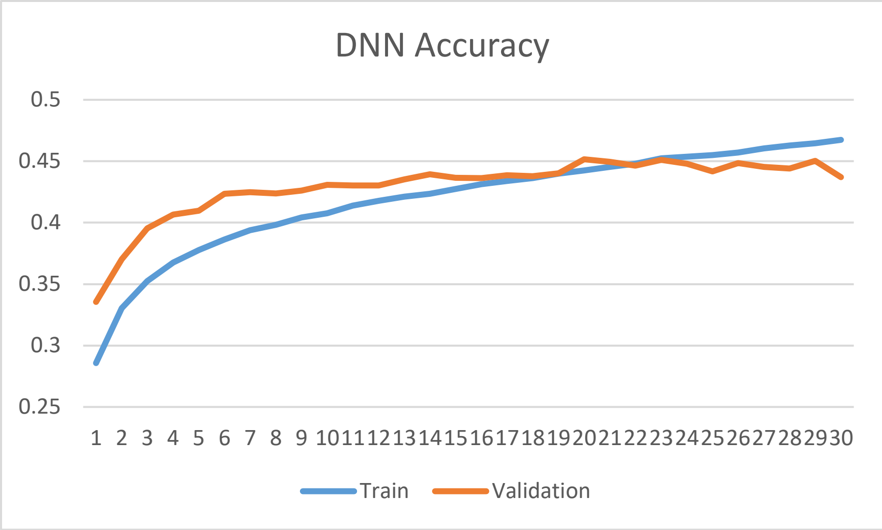


2. (1%) 承上題，請用與上述 CNN 接近的參數量，實做簡單的 DNN model。其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？
答：

DNN model 的模型架構：

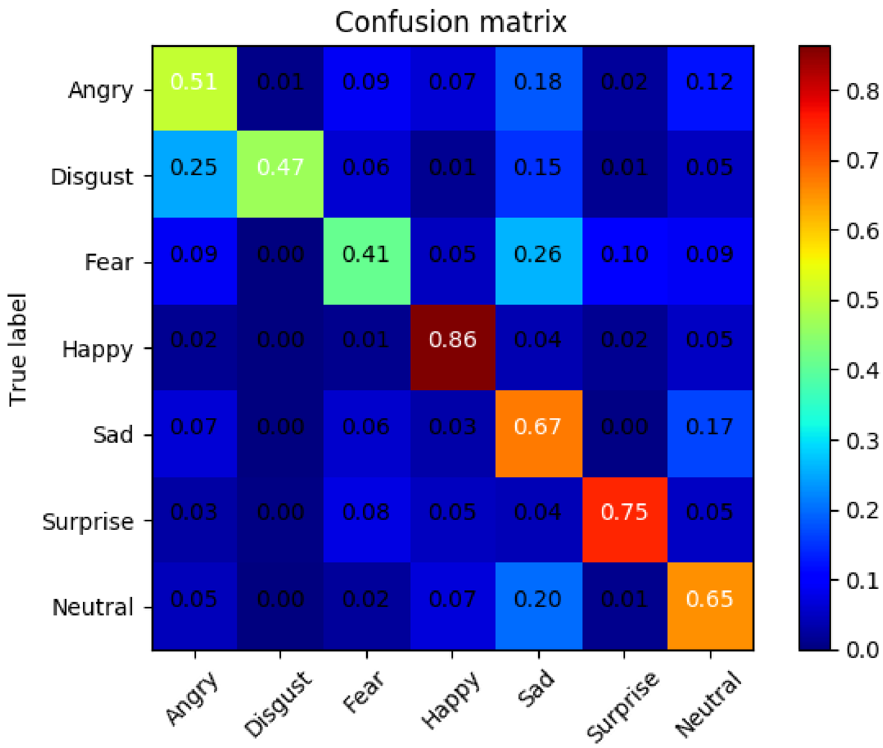
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
batch_normalization_1 (Batch (None, 48, 48, 1)         4
_________________________________________________________________
flatten_1 (Flatten)          (None, 2304)              0
_________________________________________________________________
dense_1 (Dense)              (None, 270)               622350
_________________________________________________________________
dropout_1 (Dropout)          (None, 270)               0
_________________________________________________________________
dense_2 (Dense)              (None, 270)               73170
_________________________________________________________________
dropout_2 (Dropout)          (None, 270)               0
_________________________________________________________________
dense_3 (Dense)              (None, 270)               73170
_________________________________________________________________
dropout_3 (Dropout)          (None, 270)               0
_________________________________________________________________
dense_4 (Dense)              (None, 270)               73170
_________________________________________________________________
dropout_4 (Dropout)          (None, 270)               0
_________________________________________________________________
dense_5 (Dense)              (None, 270)               73170
_________________________________________________________________
dropout_5 (Dropout)          (None, 270)               0
_________________________________________________________________
dense_6 (Dense)              (None, 7)                 1897
=================================================================
Total params: 916,931
Trainable params: 916,929
Non-trainable params: 2
_________________________________________________________________
```

訓練過程和準確率：



比較之後會發現 DNN 所做出來的效果在這次的測試資料中比 CNN 所做出的結果還要差，不過他在接近參數量和 CNN 很接近的情況下，Training 過程的速度卻比 CNN 還要快，很有可能是因為 DNN 的結構比較適合平行運算。

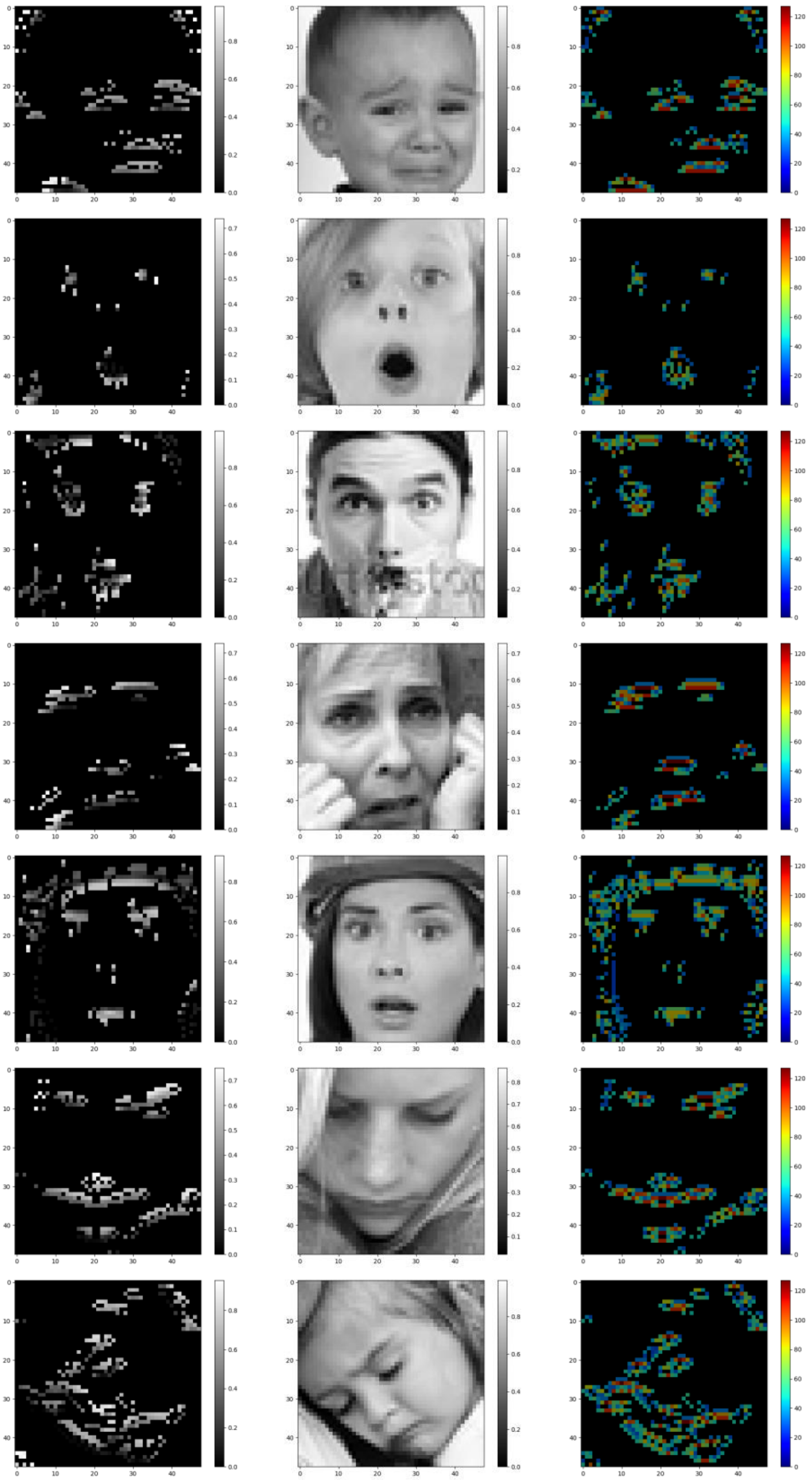
3. (1%) 觀察答錯圖片中，哪些 class 彼此間容易用混?[繪出 confusion matrix 分析]
   答：

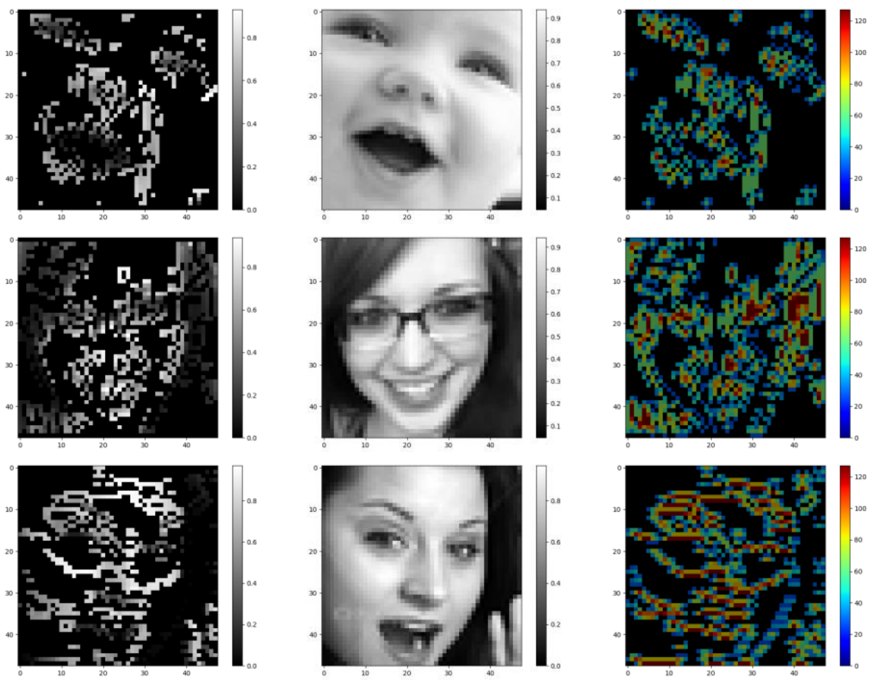


從圖片中我們會發現 Fear 比較容易和 Sad 搞混(大約四分之一的機率)
而 Disgust 也很容易和 Angry 搞混(也是大約四分之一的機率)。
Modle 對 Happy 的辨識率最好。

4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？

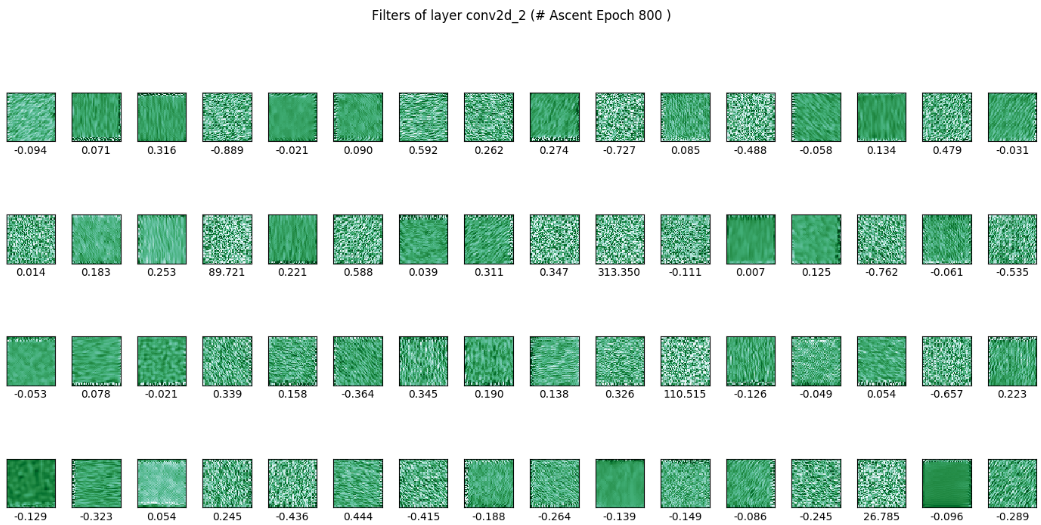
答：



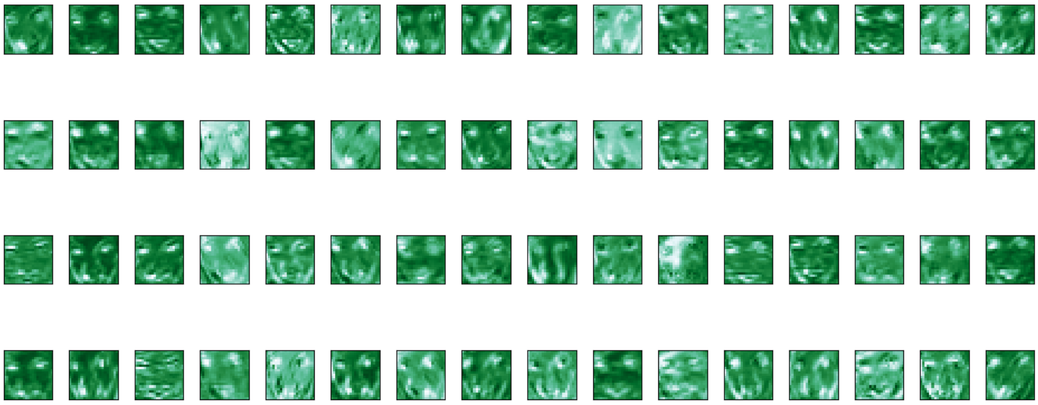


他主要會抓到的特徵是眼睛、眉毛、嘴巴、臉頰等這些臉部比較顯著的特徵。

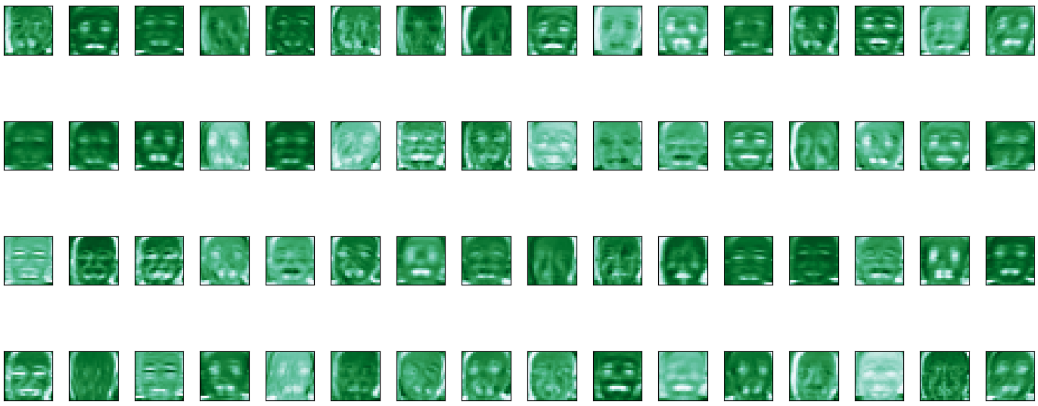
5. (1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的 filter 最容易被哪種圖片 activate。
答：



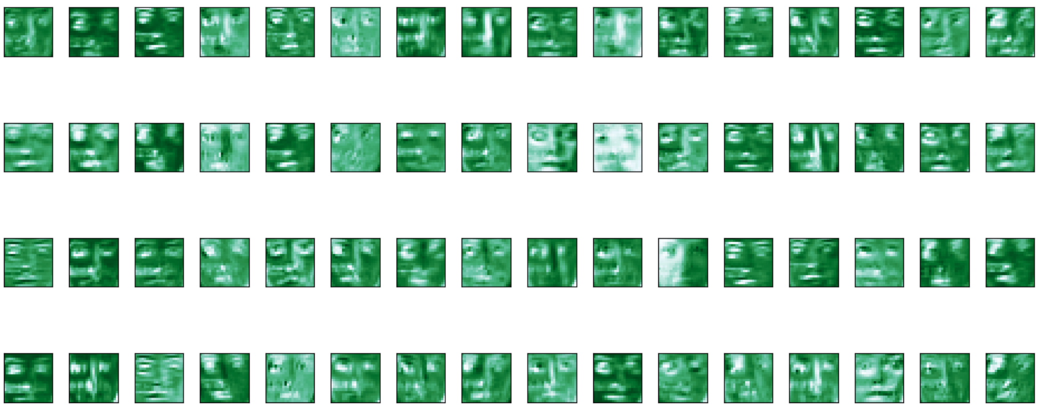
Output of layer0 (Given image91)



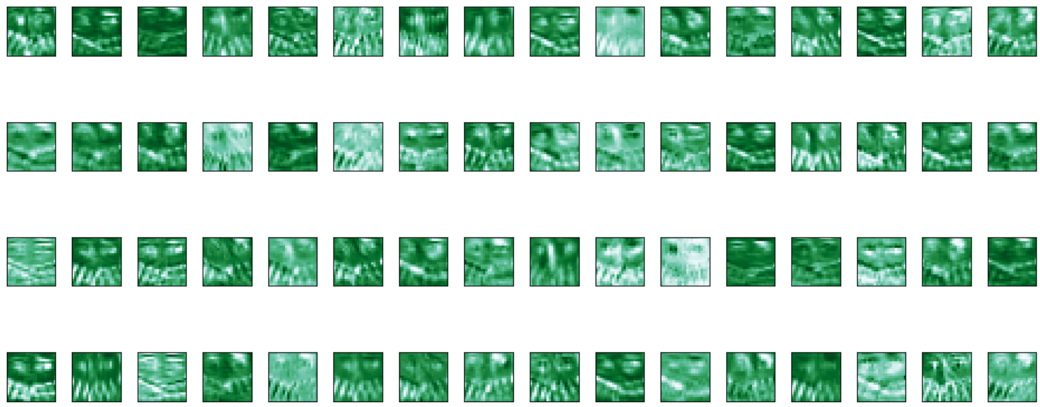
Output of layer0 (Given image87)



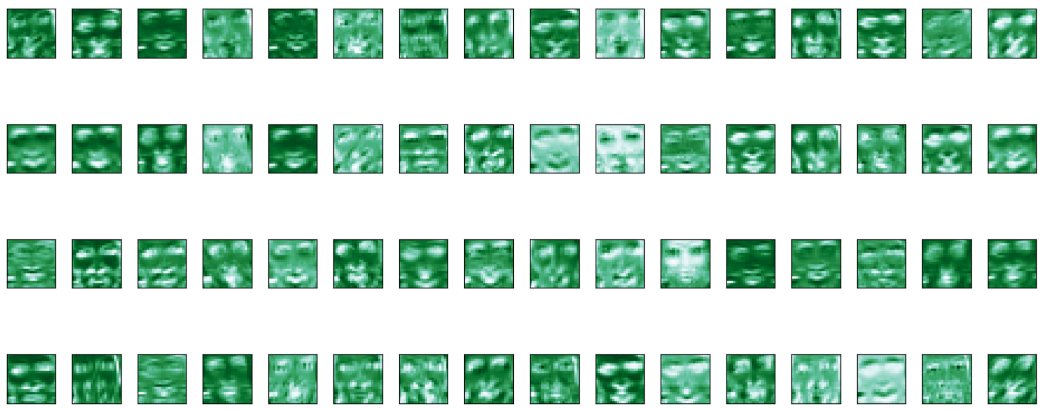
Output of layer0 (Given image66)



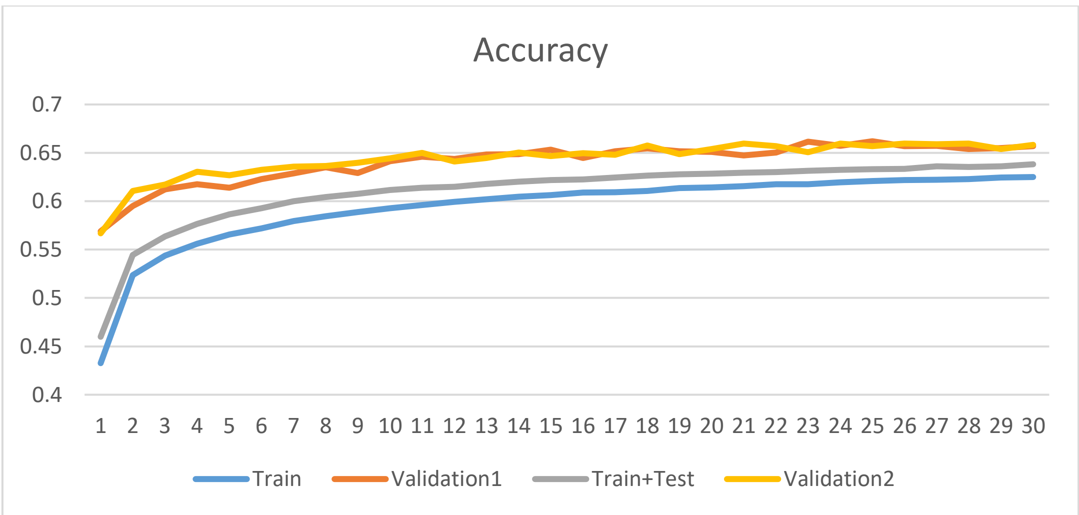
Output of layer0 (Given image56)



Output of layer0 (Given image104)



[Bonus] (1%) 從 training data 中移除部份 label，實做 semi-supervised learning



我這裡的做法是把 Training data 切出一塊當 Validation 後，用正常的方式拿去 train 後，拿 Test data 去 predict 結果，然後再把用 Test data predict 出的結果和一開始的 Training data 合在一起後拿去重新 fit 一次 model。上圖是過程中的準確度。然後我們會發現用 semi-supervised learning 所做出來的效果雖然在 Validation 上不顯著，但是在 Training 過程的可以有效地提升準確度。

[Bonus] (1%) 在 Problem 5 中，提供了 3 個 hint，可以嘗試實作及觀察 (但也可以不限於 hint 所提到的方向，也可以自己去研究更多關於 CNN 細節的資料)，並說明你做了些什麼？ [完成 1 個: +0.4%, 完成 2 個: +0.7%, 完成 3 個: +1%]