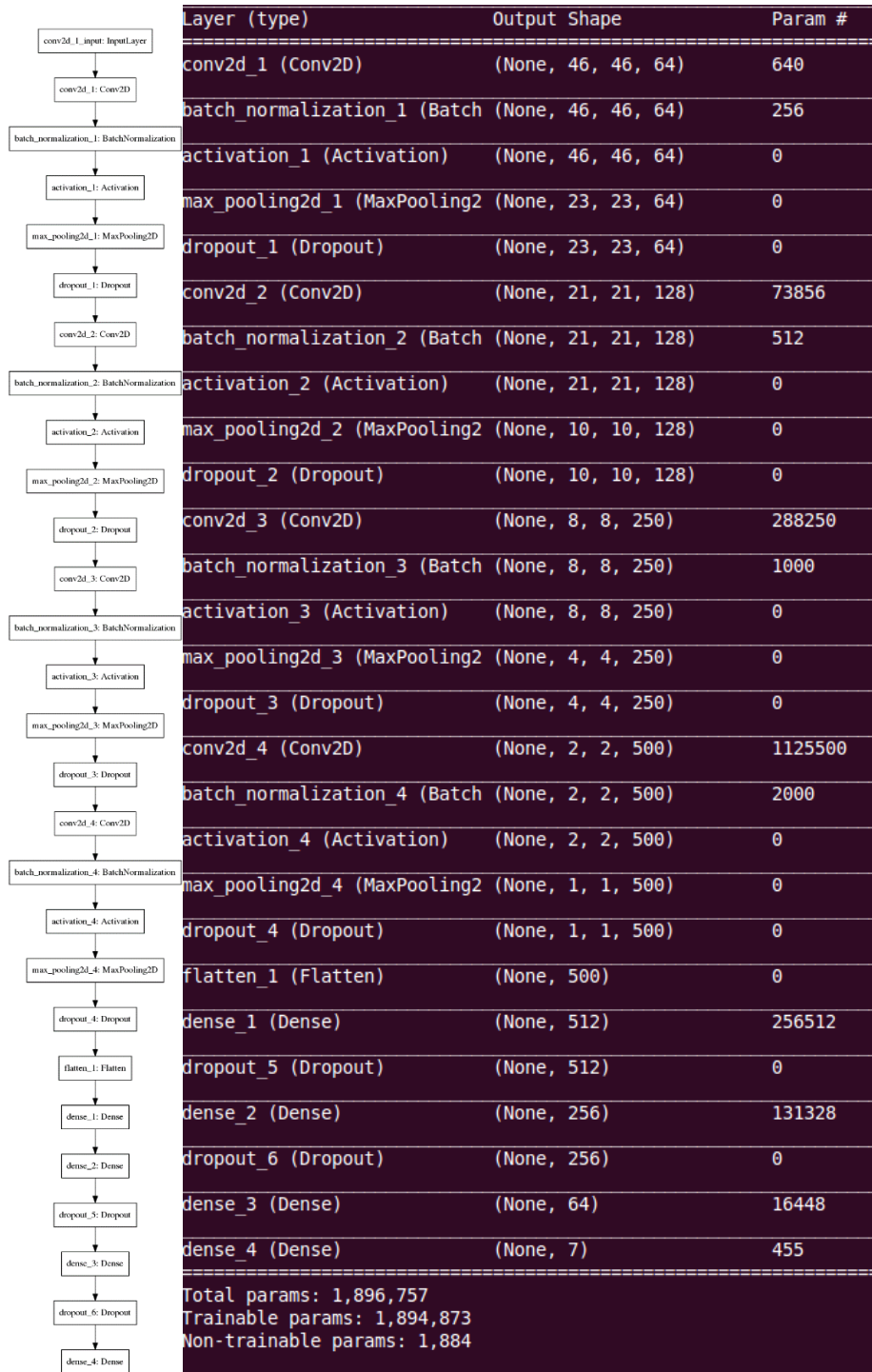


學號: R05921069 系級: 電機碩一 姓名: 黃武昱

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?

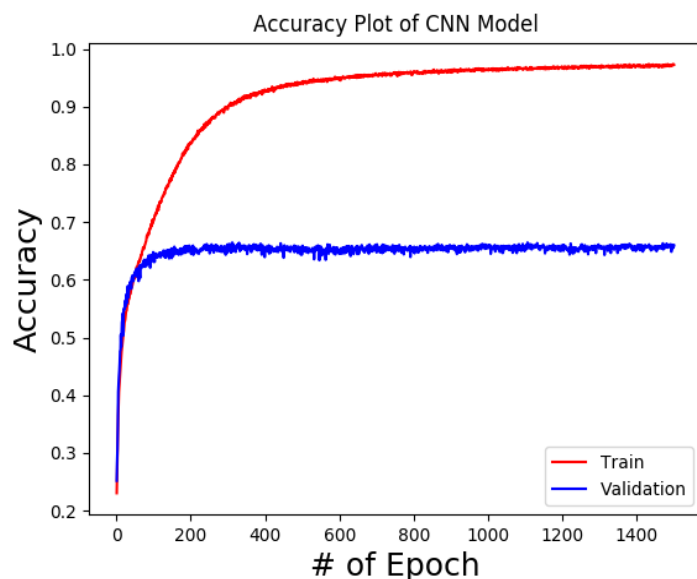


答:

在做

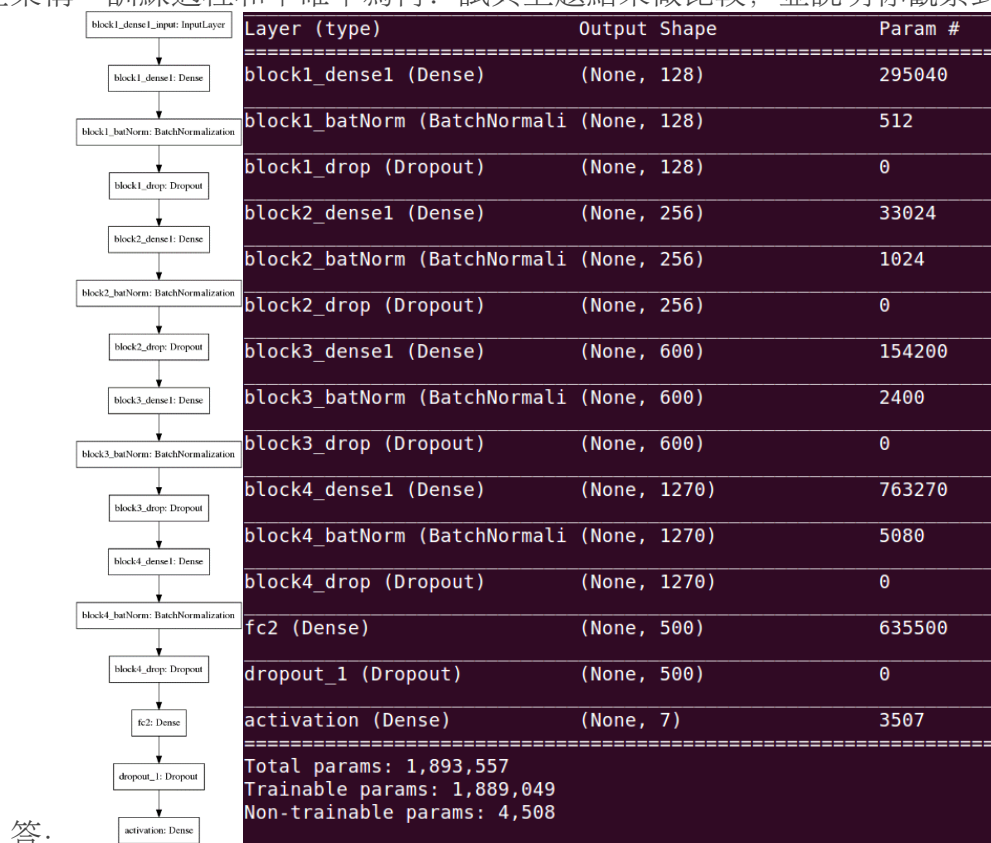
CNN 的部分, 我是用了 4 層的 convolution2D layer 然後都有分別加上 Batchnormalization, 每一層我選用 relu 當作我的 Activation function, 之後也有用 Maxpooling 去縮減圖片的大小和 Dropout 去除不好的 feature, 最後在使用 2 層的 Dense 也是選用 relu 跟最後一層 Dense 選用 softmax 去得到篩選的結果。

這是 CNN 根據 train data 跟 validation data 得到的 accuracy 圖:



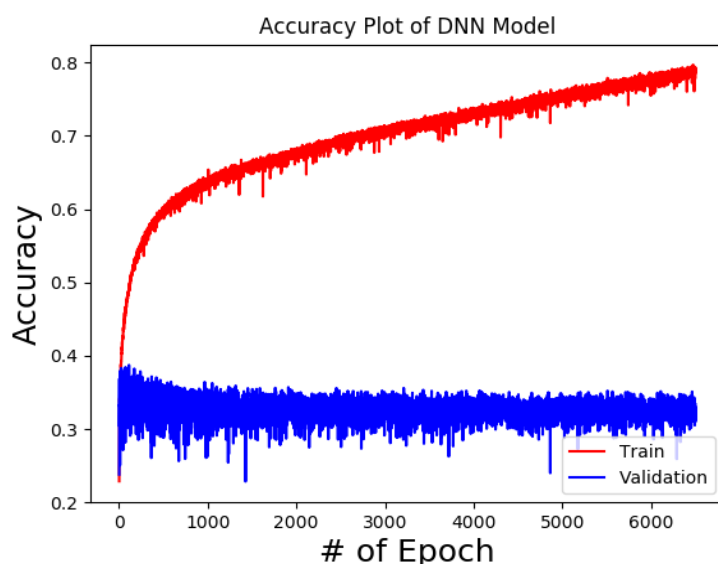
從圖表可得知，Validation 在大約 150 次 epoch 後就幾乎維持在 64%~66% 之間震盪，然後 Train 的還持續上升中，這也表示發生了 overfit，這部分雖然已經加入 Batchnormalize 去防止了，所以這部分還需要繼續研究一下。

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？



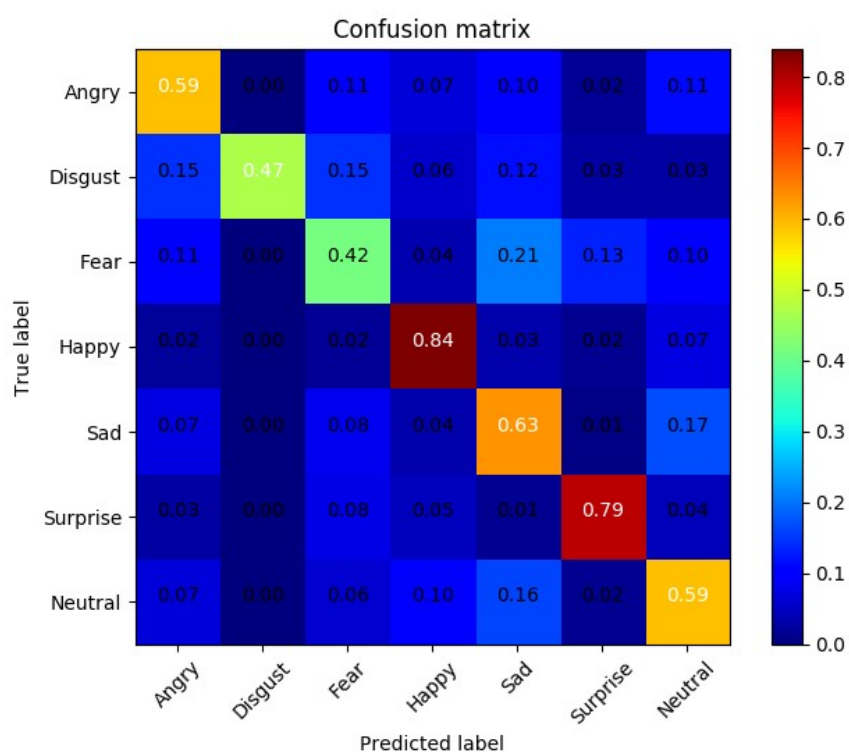
答:

在 DNN 的部分，我是使用了 5 層的 Dense 的和一層輸出 Dense 層，架構跟 CNN 是很類似的，然後，參數也是很接近，在與 CNN 最大不同之處在於，CNN 需要訓練的時間，遠大於 DNN 所需要的時間，畢竟 CNN 是需要針對一張圖片做很多次 mark 分成很多 filter，這方面的處理量是遠大於 Dense 的處理量，雖然 DNN 的速度很快，但精準度的部分，還是 CNN 遠大於 DNN。



從上圖可以發現，Validation 的 accuracy 一直持續在 30%~40%之間震盪，雖然 Train 也有在上升，但在半途中，震盪幅度也大於 CNN。

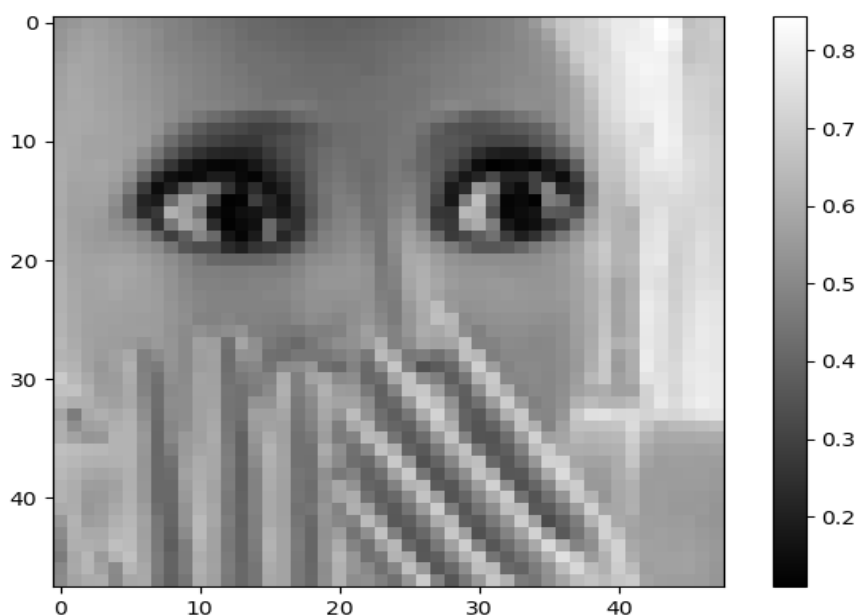
3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]



答: _____

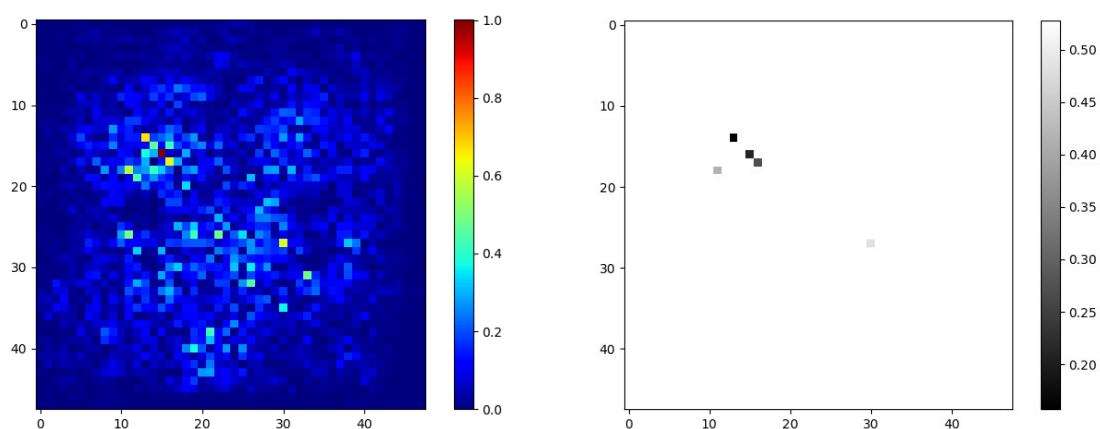
這是用 20% 的 Validation 來測試訓練好的 model，並且繪製出來的分類圖，由上圖可得知，Happy 是最好被辨識出來的，然後 Fear 反而是最難被辨識出來了，從圖片分析可以推測，Happy 的部分較誇大，所以可以得到較多的 feature，然後 Fear 的部分，幅度較小，得到的 feature 較小，故在分辨時，較少的參考數據，會被跟 Sad 混淆而出錯。

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？

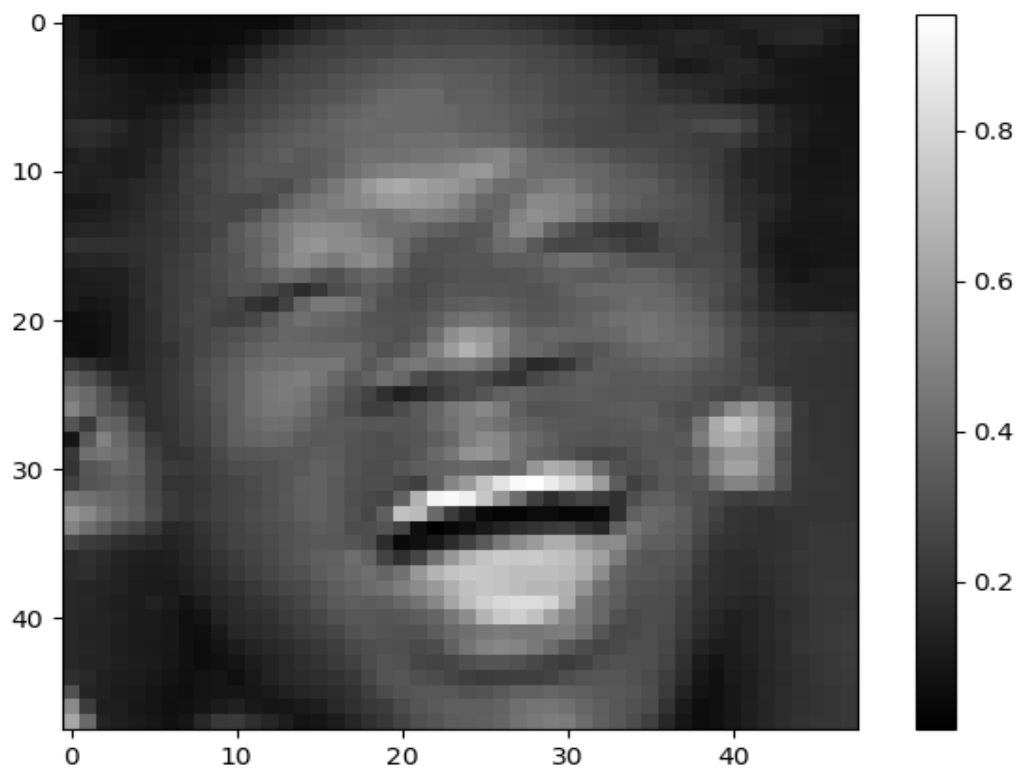


答：

首先這是原始驚訝的圖片，雖然有手擋到臉部表情，我們卻仍然能直觀的感覺它是驚訝，但是，電腦卻無法直觀的分辨，我們人類之所以能直觀的辨識，是因為長時間看過類似的表情，才能推測出來，但對於沒有預訓練過大量資料的 model 要辨別這個是極度的困難。



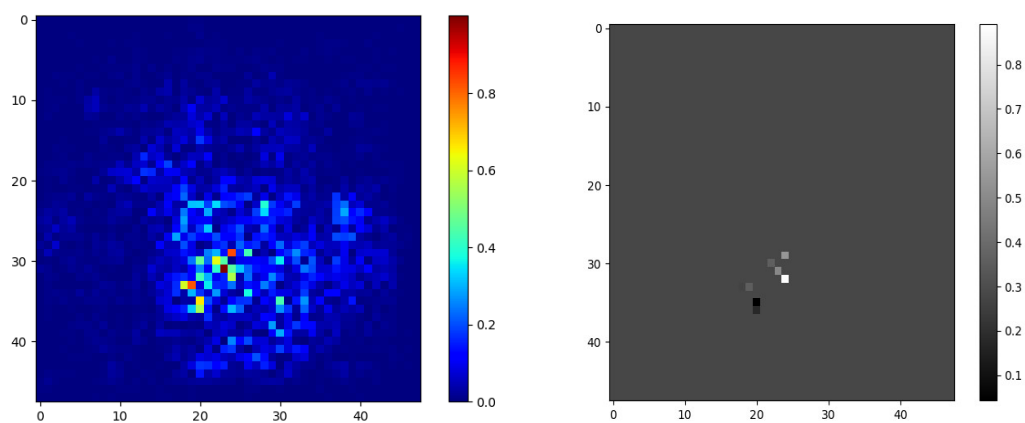
由這兩張圖可得知，電腦是直接 focus 在眼睛的部分，是因為眼睛部分相對於其他部分是比較容易辨別的。



再來我們換另一張表情較容易辨識的圖案來比對一下。

從這張圖，可以得知是笑臉表情的圖案，相對於上圖表情來說，這張圖有很明顯的特徵。

有較多的特徵，那麼電腦在分辨時，就有較多的資料能辨識了。



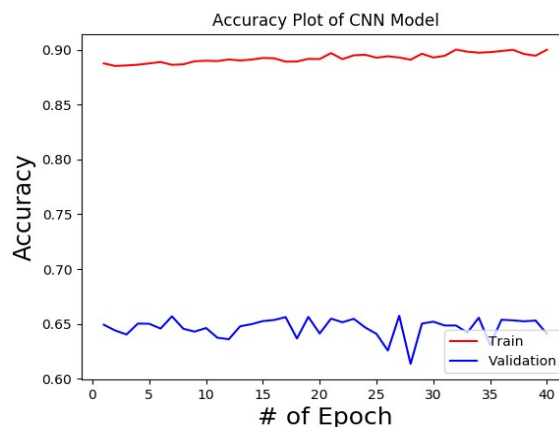
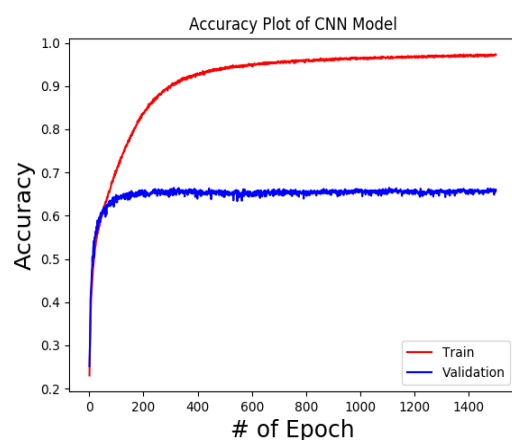
(1%) 承(1)(2)，利用上課所提到的 **gradient ascent** 方法，觀察特定層的 **filter** 最容易被哪種圖片 **activate**。



這是選取第一層 **convolution layer** 的 **Activation relu layer** 後，並且經過 **gradient ascent** 出來得圖片，從這裡面可以發現，有機張圖是比較明顯看得出特徵 (**loss value** 較低)，有機張圖則是還全看不出來是什麼 (**loss value** 較高)，對於我們訓練 **model** 而言，只以保留那些比較能看得出臉型 (**loss value** 低的)，其餘的都是屬於可以去掉的部分。

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

我把 test data 丟入訓練好的 model，先使用 `predict_classes` 去預測這些 data 的數值，再用 `predict_proba` 去預測這個 data 在 0~6 之間所有可能的機率，並使這兩個資料相乘，看是在哪個數值的機率最高 (EX:1 的機率是 80%)，在把這些 data 跟從 data 得到的 label 加入倒 train data 中，增加數據量，使精準度上升。



但

右圖為原本的 CNN，左圖為加入 semi-supervised 後的結果，比較之下，CNN 的 model 比較穩定，但理論上有加入 semi-supervised 應該會比較好，也許我設置的加入 ublabel 部份的參數過大，需要再調整。