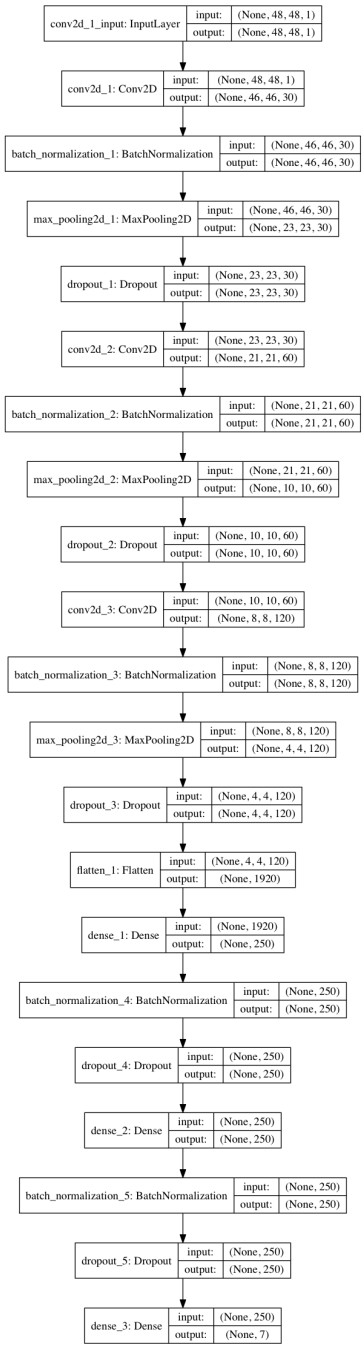
學號：B02404002 系級：醫技四 姓名：葛竑志

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：如右圖

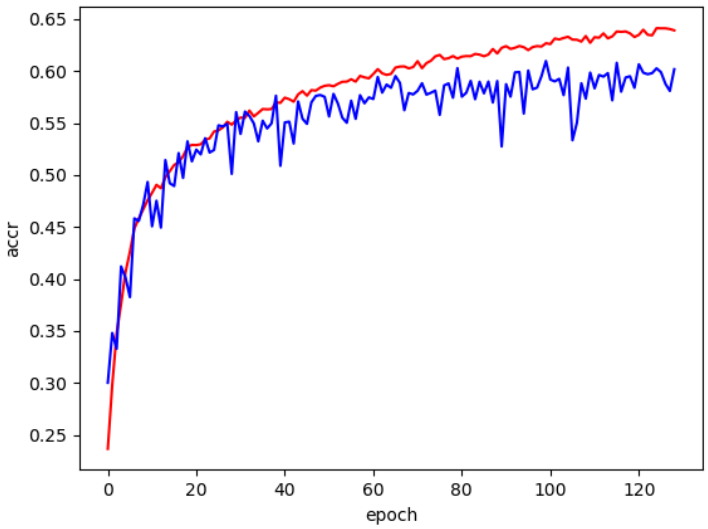


1. 參數量
Total params: 629,077
Trainable params: 627,657
Non-trainable params: 1,420

2. 前處理
有使用 imageDataGenerator 做 pre-processing。
ImageDataGenerator(width_shift_range=0.1,height_shift_range= 0.1, rotation_range= 35,preprocessing_function= lambda X:X + normal(scale= 0.05, size=X.shape))

3. 訓練量
使用 fit_generator 做訓練，mini_batch= 75，epoch=20，callback 加有 earlystopping。外加一個 for loop 重複此訓練過程。

＊訓練過程：
（紅線：training accuracy, 藍線： validation accuracy）



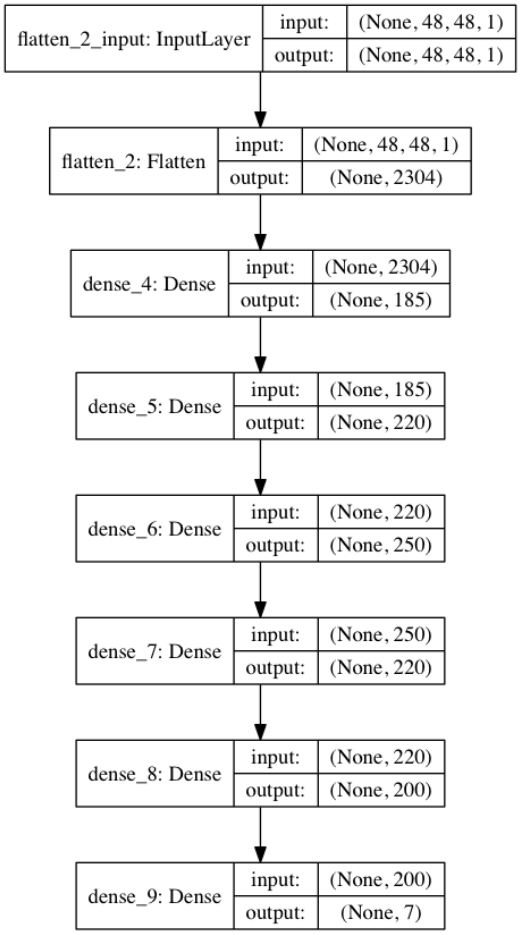
＊準確率：
1. Validation accuracy 可以達到 0.59，訓練過程中剛開始可以得到顯著的提升，然而隨著 epoch 數後期的上升會使 accuracy 的上升程度趨緩，為了避免 overfitting，不可過度

train。
2. Training accuracy 可以達到 0.64， 同樣隨著 epoch 數上升，其上升幅度也會下降。

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：

＊模型架構：如右圖



1. 參數量

Total params: 623,422

Trainable params: 623,422

Non-trainable params: 0

2. 前處理

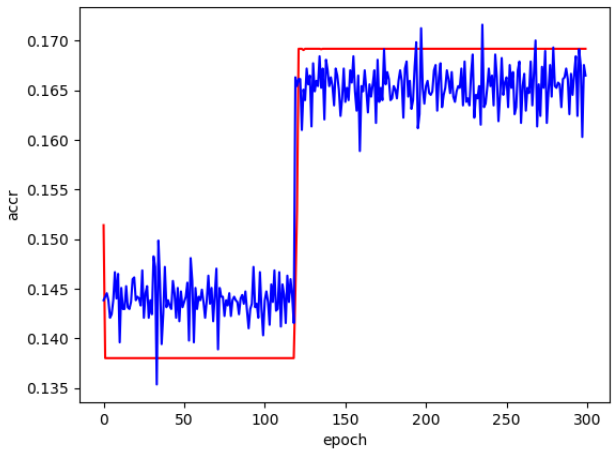
與第一題同。

3. 訓練量

欲第一題同。但因為此 model 無法讓 accuracy 有穩定上身的趨勢，為了避免訓練 epoch 數太少，所以沒有加上 early_stopping 的限制。

＊訓練過程：

（紅線：training accuracy, 藍線： validation accuracy）



＊準確率：

1. Validation 與 Training accuracy 皆只有不到 20%。
2. 訓練 epoch 數增加也無法使 accuracy 獲得明顯提升。

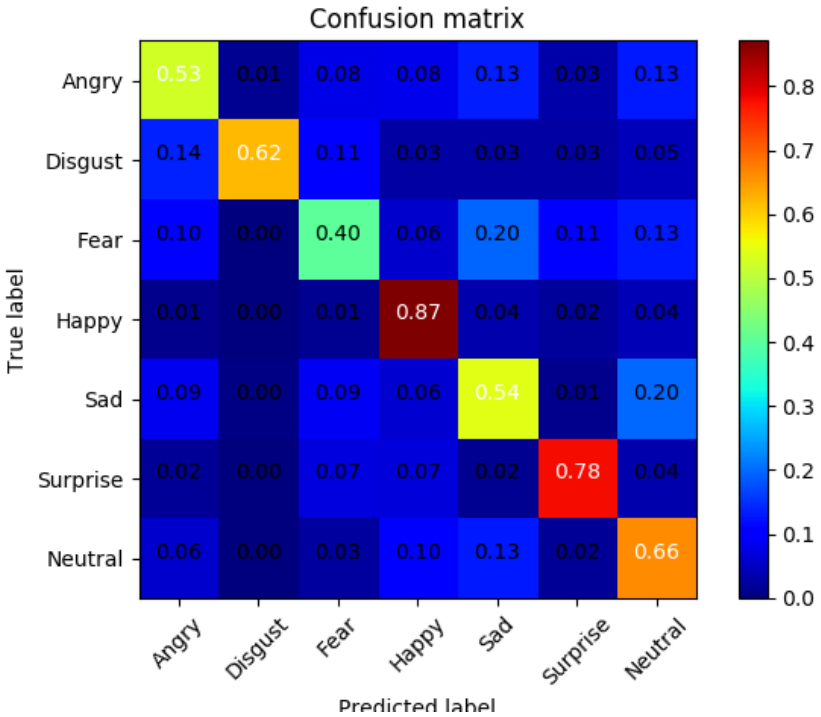
＊與第一題 model 的比較：

1. 經過接近的參數量的下去 train，此 DNN 無法使 accuracy 得到大量提升。
2. DNN 的準確率顯然遠低於 CNN 的 model。

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]

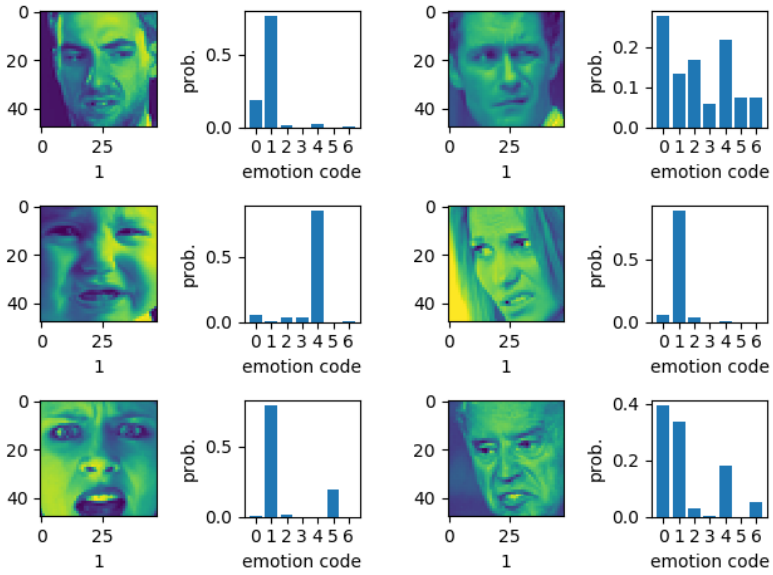
答：

A. Confusion matrix over a Kaggle-scored 0.65478 model.



B. 此處根據上述 Confusion matrix 挑出一些 error distribution 大於等於 0.14 的部分作討論，從該 True label 的圖片中隨機挑選圖片，並呈現經過 model.predict_proba()得出的 distribution probability。

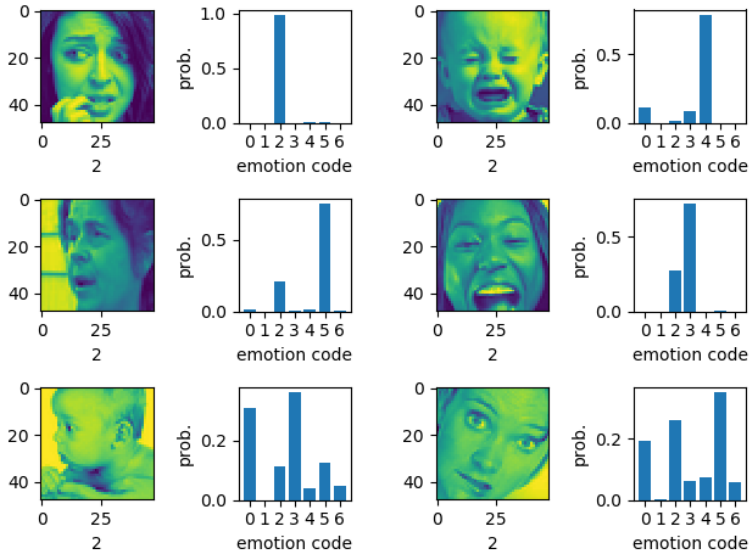
a. Disgust 判為 Angry：（以下圖為 True label= disgust）



左中：雖然 label 給的是 disgust，但主觀上會覺得這是嬰兒要哭的樣子，和 predict 出來的 sad 不謀而合。

右上、右下：predict 出來的結果皆為 angry，主觀上看起來若是不知該表情產生的情境，確實容易造成誤判。

b. Fear 判為 Sad（以下圖為 True label= fear）

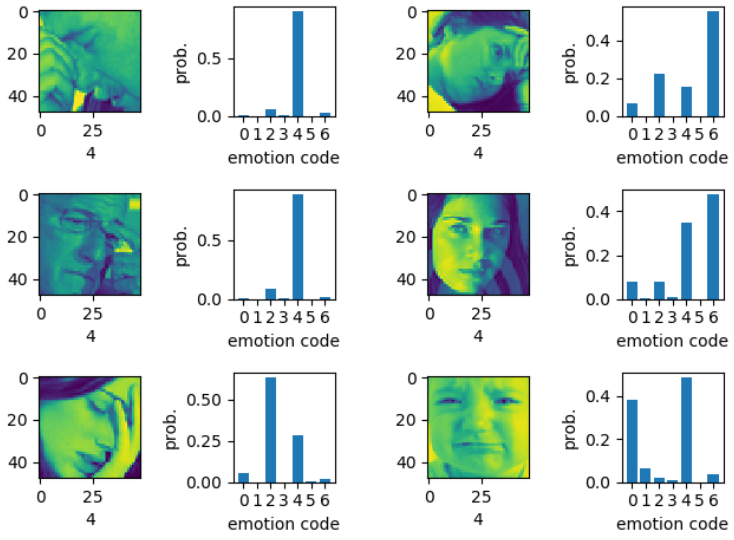


左下：是小嬰兒的側臉，可能是無法呈現重要的 feature（五官）供機器判讀，因此容易誤判為不相干的情緒。

右上：雖然 label 為 fear，主觀上認為像是小嬰兒哭，與 predict 出來的結果一致，但對小孩而言 fear 與 sad 也只是一線之隔罷了。

右中：因為嘴巴和眉毛的關係，主觀上容易覺得是 happy，與 predict 出來的結果一致。

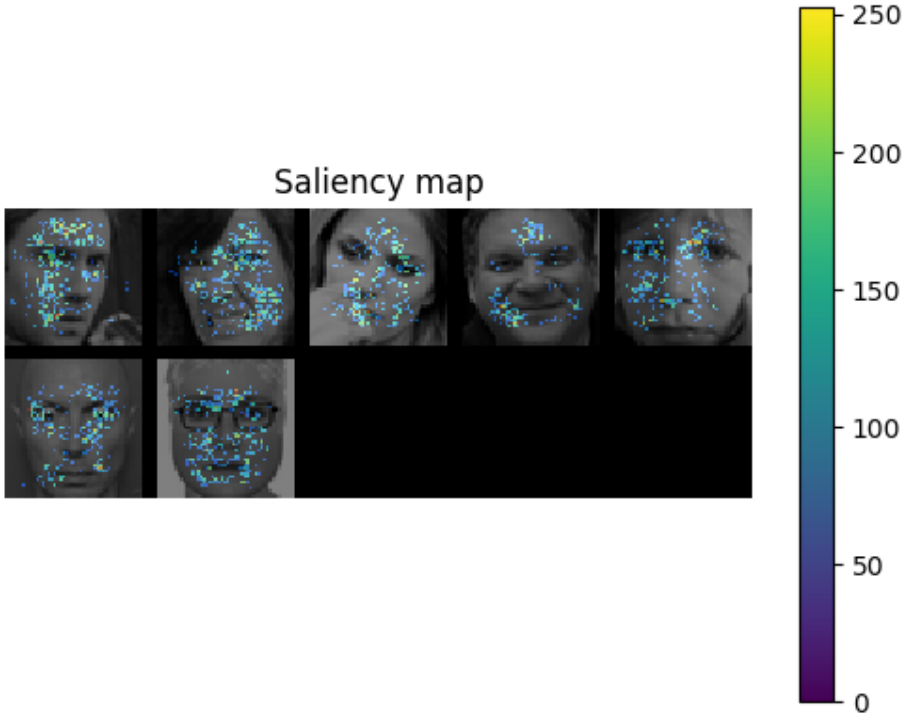
c. Sad 判為 Neutral（以下圖為 True label= sad）



這邊的結果主觀上認為都和其 label 很符合，然而確實也容易發生誤判的情形。（也許是很多 Sad 圖常常有手在圖片內干擾重要的 feature）

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：

＊從 dense_3（根據第一題的模型）layer 做 saliency map 的結果。

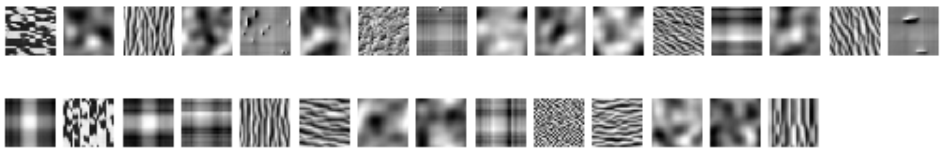


此處依序選了 label 為 0 至 6 不同情緒的影像做分析，大部分 heat spots 可以很好的落在臉部。然而，並不是所有點都落在與情緒最有關的眼睛、眉毛、嘴巴上面，有些部分如額頭、嘴角的位置也是此模型 focus 的地方。

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：

＊以下 layer 皆是根據第一題的模型而測試

＊conv2d_1



此 layer 為第一層 layer，可以看到這一層最能 activate filter 的圖片很多是由重複的線條在不同方向上的排列所組成，可能也是要對圖片初步的做分型而有這種結果。

＊dense_1

此 layer 為 flatten 後的第一層 fully connected layer，可以看到根據這一層 filter 產生的結果更複雜化，例如對於線條的扭曲、重複的特殊 pattern 等，可能代表 model 在對於圖片做更細部化的分類。

＊dense_3

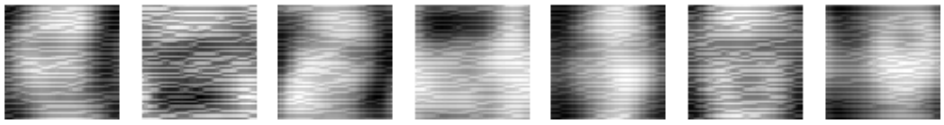
此 layer 為最後一層 layer，output dimension 為 7。可以在由左數來第四張圖（Happy）看到類似開口大笑的嘴巴和眼睛的線條，而這也是在 confusion matrix 中看來預測精準度最高的 label。其餘根據 filter 產生的結果有些可以看到類似眼睛或是臉部上較為明顯線條的組合，然而也有些則呈現扭曲的波紋或重複的 pattern，有可能是因為 model 在預測這類 label 時的精準度尚無法在此達到較容易看出人臉的程度。

＊接下來將特定圖片送入 model 後 plot 出某些 layer 的結果

＊original

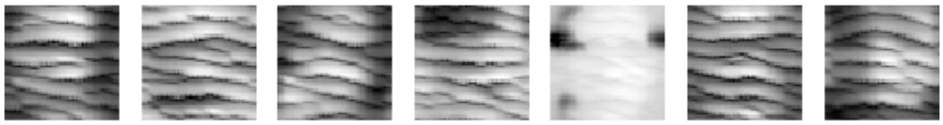
這是原圖

＊conv2d_1



這是經過第一次 convolution 的結果，可以看到許多橫向的線條。

＊conv2d_3



這是經過三次 convolution 與一些 normalization 及 dropout 後的結果，線條更複雜化。

＊dense_1

可以看到看起來像人臉的圖案，並且大多凸顯在眼睛鼻子嘴巴等足夠判斷五官的位置。

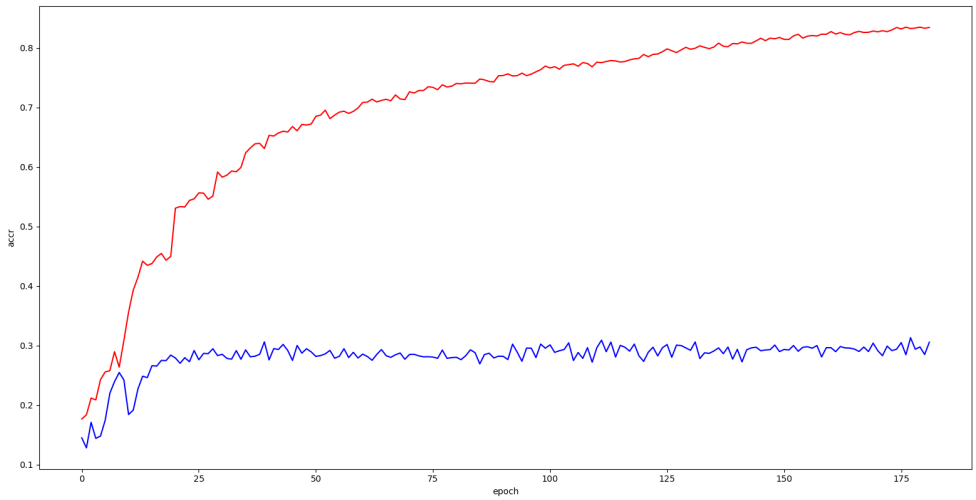
[Bonus] (1%) 從 training data 中移除部份 label，實做 semi-supervised learning

self_learning.py

＊設計架構:
1.從所有的 training data 中切出約 20%的資料作為 validation set。
2.剩下的 training data 以 1000 筆資料為單位切出各個 batch，除了第一個 batch 因為要放進 model 訓練要保留 label，剩下資料的 label 皆忽略掉。
3.送到 model 中 training。Train 完後用 validation set 驗證準確率。
4.以訓練好的 model 去 predict 下一個 batch，並將其 feature 及 predicted label 加入 training batch 中。
5.重複 3,4 步驟直到 training data 都被訓練過。

＊模型架構：
與第一題同。

＊訓練過程：
（紅線：training accuracy, 藍線： validation accuracy）

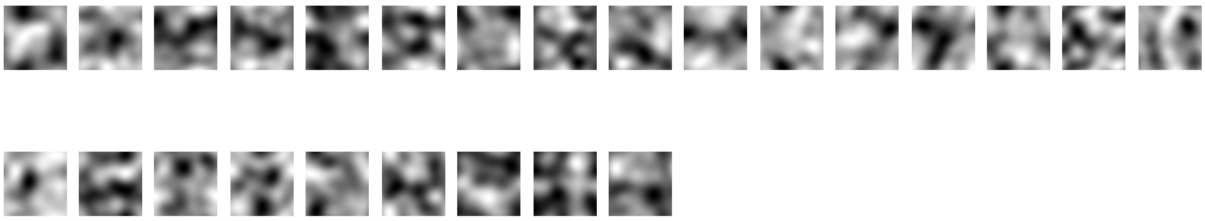


＊觀察與比較：
1. 經過 Initial batch 做訓練後，即便加上 predicted label 的資料做訓練，仍無法使 val_acc 得到明顯的上升。相對於 val_acc，tra_acc 可以隨著訓練 epoch 數上升，但上升幅度將逐漸趨緩。這種模型可能可以讓一開始對 Initial batch 的假設逐漸增強，因此在 tra_acc 上可以穩定上升。然而對外部的 data 的預測性能則不高。
2. 相對於第一題的模型，此模型由於下去 train 的 true labeled 資料量不多，因此顯然無法訓練出足夠強韌的模型。

[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%]

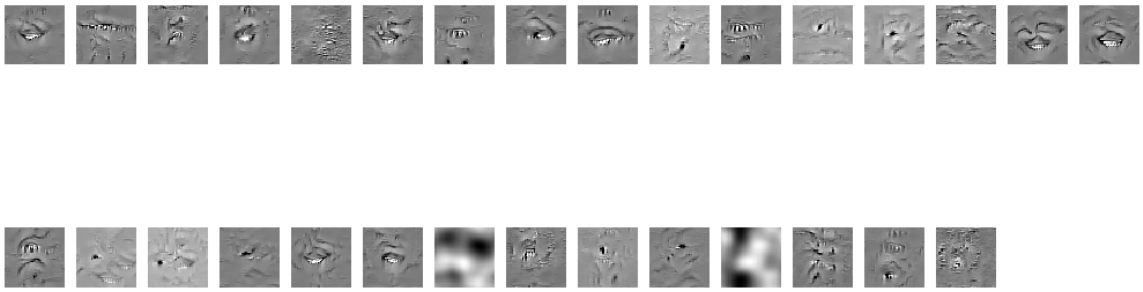
＊test on poor model:
此處選擇了一個 validation accuracy 只有 0.59 的模型，模型架構如下頁圖。

＊conv2d_1：



可以看到相對於較好的 model，這一層的 filter 多半沒有一個固定的 pattern（如重複條紋）而比較像模糊雜訊，代表在第一層可能不具備對圖片做一種初步系統性分類的能力。

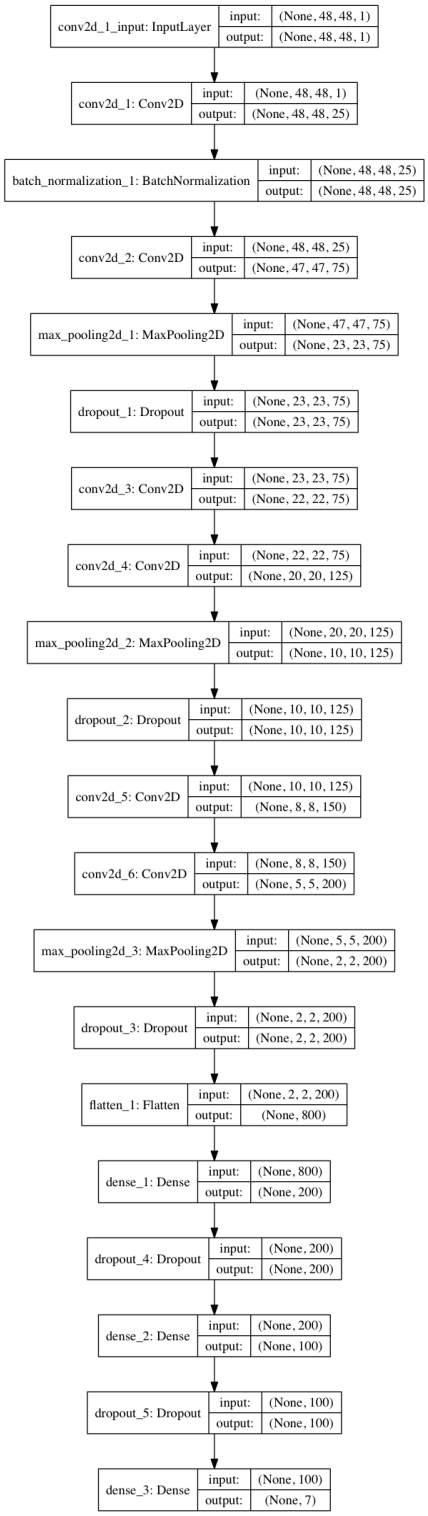
＊dense_1:



可以看到這一層呈現的影像是眼睛或是嘴巴經過任意重複堆疊的圖形，與較好的模型比較起來，這裡似乎是根據圖片上的特定位置上的特定 feature 去辨識，而與藉由複雜線條的 pattern 做分類的好 model 不同。

＊dense_3:

這是這個 model 的 output layer，相對於好 model，這層的圖幾乎無法看起來像個人臉。


conv2d_1_input: InputLayer
input: (None, 48, 48, 1)
output: (None, 48, 48, 1)
conv2d_1: Conv2D
input: (None, 48, 48, 1)
output: (None, 48, 48, 25)
batch_normalization_1: BatchNormalization
input: (None, 48, 48, 25)
output: (None, 48, 48, 25)
conv2d_2: Conv2D
input: (None, 48, 48, 25)
output: (None, 47, 47, 75)
max_pooling2d_1: MaxPooling2D
input: (None, 47, 47, 75)
output: (None, 23, 23, 75)
dropout_1: Dropout
input: (None, 23, 23, 75)
output: (None, 23, 23, 75)
conv2d_3: Conv2D
input: (None, 23, 23, 75)
output: (None, 22, 22, 75)
conv2d_4: Conv2D
input: (None, 22, 22, 75)
output: (None, 20, 20, 125)
max_pooling2d_2: MaxPooling2D
input: (None, 20, 20, 125)
output: (None, 10, 10, 125)
dropout_2: Dropout
input: (None, 10, 10, 125)
output: (None, 10, 10, 125)
conv2d_5: Conv2D
input: (None, 10, 10, 125)
output: (None, 8, 8, 150)
conv2d_6: Conv2D
input: (None, 8, 8, 150)
output: (None, 5, 5, 200)
max_pooling2d_3: MaxPooling2D
input: (None, 5, 5, 200)
output: (None, 2, 2, 200)
dropout_3: Dropout
input: (None, 2, 2, 200)
output: (None, 2, 2, 200)
flatten_1: Flatten
input: (None, 2, 2, 200)
output: (None, 800)
dense_1: Dense
input: (None, 800)
output: (None, 200)
dropout_4: Dropout
input: (None, 200)
output: (None, 200)
dense_2: Dense
input: (None, 200)
output: (None, 100)
dropout_5: Dropout
input: (None, 100)
output: (None, 100)
dense_3: Dense
input: (None, 100)
output: (None, 7)