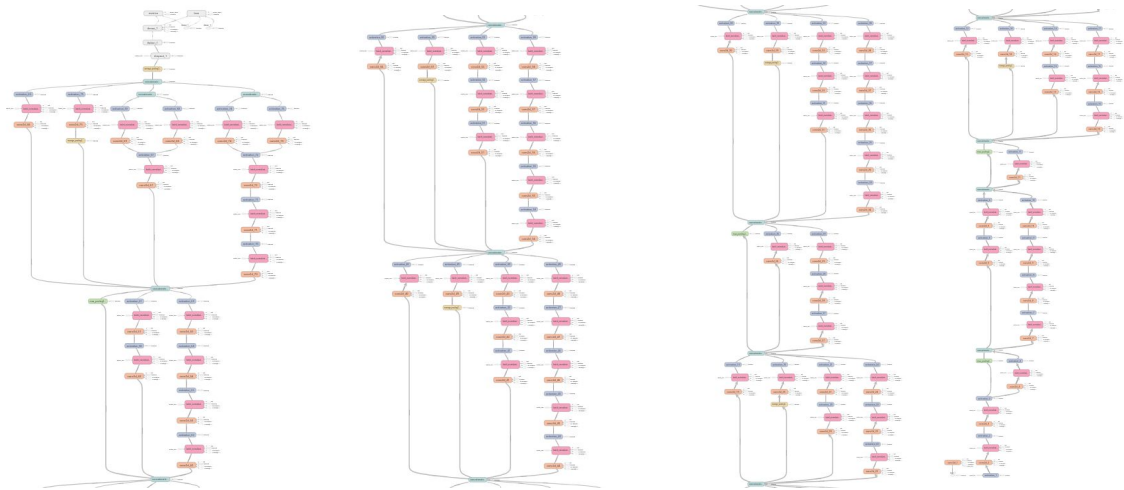


1. (1%) 請說明你實作的 CNN model, 其模型架構、訓練過程和準確率為何？
(Collaborators:)

架構：

參考 Inception-v3 並少用了幾層的block和把一些stride調成1



過程：

input 做小幅度的 scale/shear/horizontal flip

總共訓練了160 epoch 但超過 60 就有一點 overfit training data

準確率：

train	validation	public	private
0.892	0.675	0.676	0.682

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？
(Collaborators:)

	CNN	DNN
Total Params	16,129,383	15,928,199
Trainable	16,100,007	15,928,199
Non trainable	29,376	0

架構：

2048 * 2 -> 1024 * 2 -> 512 * 2 -> 256 * 2 -> 128 * 2 -> 64 * 2 -> 32 * 2 -> 64 -> 128 -> 256
-> 512 -> 1024 -> 2048 -> softmax(7)

過程：

input 做小幅度的 scale/shear/horizontal flip

總共訓練了160 epoch, model一直學不會

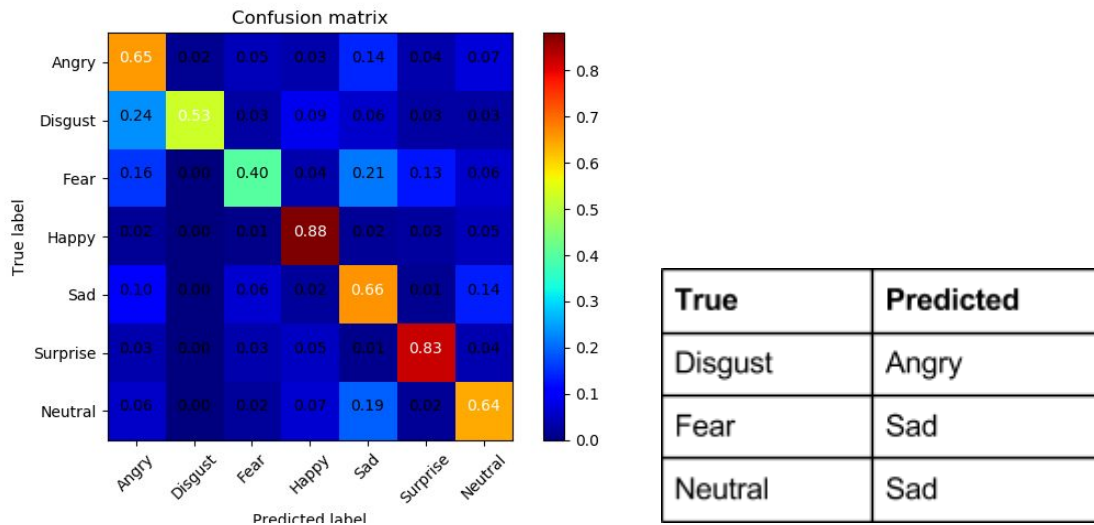
準確率：

train	validation	public	private
0.259	0.253	0.260	0.260

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]

(Collaborators:)

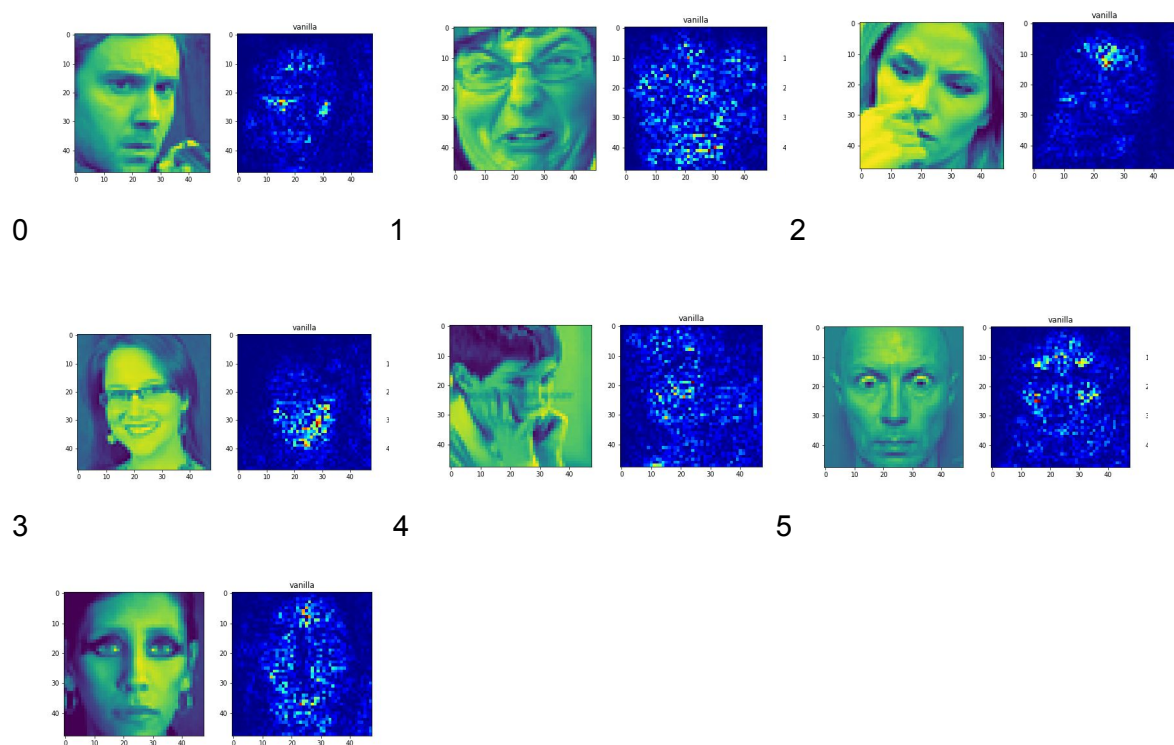
答：



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？

(Collaborators:)

[original, vanilla, guided, relu]

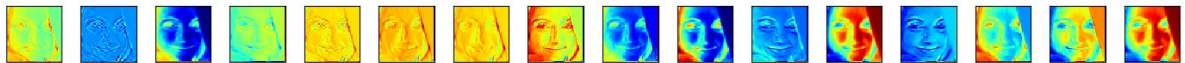
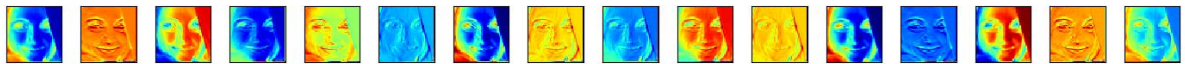


5. (1%) 承(1)(2), 利用上課所提到的 gradient ascent 方法, 觀察特定層的filter最容易被哪種圖片 activate。

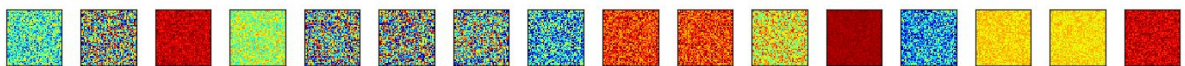
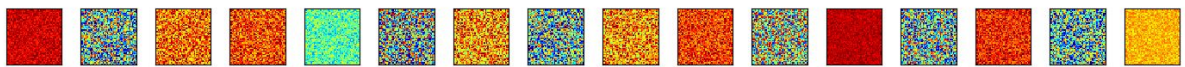
(Collaborators:)

答：

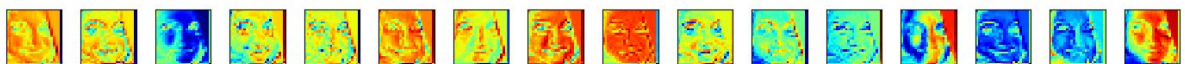
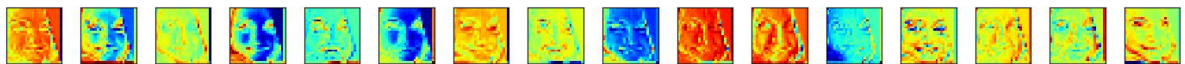
conv2d_1 activations



conv2d_1 filters



conv2d_2



filters were not generated because there is a batch normalization layer in between and I haven't figure out how to feed it what it needs